Proceedings of the
16th International Workshop
on Non-Monotonic Reasoning
(NMR 2016)
April 22 – 24, 2016
Cape Town, South Africa

Editors:
Gabriele Kern-Isberner, Lehrstuhl Informatik 1, Technische Universität Dortmund
Renata Wassermann, Computer Science Department, University of São Paulo

Number: 852
Nonmonotonicity is crucial for any formal approach to modelling human reasoning – most of the conclusions we draw in our everyday lives and on which we base (sometimes important) decisions are defeasible, prone to be given up if further information arrives. This is in clear contrast to classical – e.g., propositional or first-order – logics which are monotonic, i.e., their deductive conclusions are preserved for eternity. It is due to monotonicity that all (correct) proofs in mathematics are still valid, regardless of whatever new theories are being developed, but also that robots based on classical logics fail in uncertain, incompletely specified environments.

Therefore, nonmonotonic reasoning (NMR) deals with important issues in Artificial Intelligence, and has strong connections to other areas of knowledge representation, in particular, to belief revision, action logics, argumentation, logic programming, preference handling, and uncertain reasoning. The NMR workshops are the premier forum for presenting results in this broad subfield of knowledge representation and reasoning (KR). Their aim is to bring together active researchers, and foster discussions and collaborations on theoretical foundations, applications, and system development.

NMR has a long history – it started in 1984, and is held every two years. Recent previous NMR workshops took place in Vienna (2014), Rome (2012), Toronto (2010), and Sydney (2008). Following established and fruitful traditions, NMR 2016 was co-located with the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR 2016) and the 29th International Workshop on Description Logics (DL 2016). In particular, NMR 2016 shared a joint session with DL 2016. We were happy to welcome Laura Giordano (Università del Piemonte Orientale) and Leon van der Torre (University of Luxembourg) as invited speakers, Laura’s talk was also invited by DL 2016.

This volume contains most of the accepted papers of NMR 2016. Some papers had been already published, or are meant to be published elsewhere, so we could only provide URL’s in those cases. This collection of NMR papers illustrate impressively both the depth and the breadth of NMR by dealing with theoretical issues as well as connecting different subfields of knowledge representation and reasoning.

In Studies on Brutal Contraction and Severe Withdrawal: Preliminary Report, Marco Garapa, Eduardo Fermé, and Maurício Reis study different classes of contraction operators and provide axiomatic characterizations for them. Zhiqiang Zhuang, James Delgrande, Abhaya Nayak, and Abdul Sattar deal with interleaving two kinds of nonmonotonicity: in their paper A New Approach for Revising Logic Programs, they also allow the logic underlying belief revision operations to be nonmonotonic. Also Aaron Hunter’s paper Ordinal Conditional Functions for Nearly Counterfactual Revision considers an interesting and often neglected issue in belief revision: How can revision by (nearly) counterfactual conditionals be carried out? Marc van Zee and Dragan Doder present results on AGM-Style Revision of Beliefs and Intentions from a Database Perspective.
Two papers study belief revision in a probabilistic environment: Gavin Rens proposes a unified model of quantitative belief change in his paper On Stochastic Belief Revision and Update and their Combination, and together with Thomas Meyer and Giovanni Casini, he also contributes to this volume by Revising Incompletely Specified Convex Probabilistic Belief Bases. Moreover, another two papers deal with connections between nonmonotonic reasoning and belief revision, on the one hand, and ontological reasoning, on the other hand: Özgür Özçep considers Iterated Ontology Revision by Reinterpretation, and Valentina Gliozzi focusses on typicality operators in her paper A strengthening of rational closure in DLs: reasoning about multiple aspects.

One of the most basic ideas of nonmonotonic reasoning is to order models or states by preference relations. In Preferential Modalities Revisited, Katarina Britz and Ivan Varzinczak apply such semantic preference relations to modal accessibility relations, while Kristijonas Čyras and Francesca Toni consider preferences in assumption-based argumentation (ABA) frameworks in their paper Properties of ABA+ for Non-Monotonic Reasoning. Jesse Heyninck and Christian Straßer also emphasize the strong Relations between assumption-based approaches in nonmonotonic logic and formal argumentation. Ringo Baumann, Thomas Linsbichler, and Stefan Woltran deal with Verifiability of Argumentation Semantics by elaborating on which specific information some well-known semantics of abstract argumentation systems can be based. Jean-Guy Mailly’s paper Using Enthymemes to Fill the Gap between Logical Argumentation and Revision of Abstract Argumentation Frameworks is a first approach to tackle the problem that agents cannot decode arguments correctly due to missing or different (background) knowledge. Thomas Linsbichler, Jörg Pührer, and Hannes Strass aim at Characterizing Realizability in Abstract Argumentation by presenting algorithmic approaches that are apt to build up knowledge bases of arguments from a given set of interpretations. A somehow dual problem is considered in the context of belief merging in the paper Distributing Knowledge into Simple Bases by Adrian Haret, Jean-Guy Mailly, and Stefan Woltran: How can a knowledge base arise by merging simpler knowledge bases in a given fragment of classical logic?

Zeynep Saribatur and Thomas Eiter present a high-level representation formalism that can be applied to model Reactive Policies with Planning for Action Languages. In his extended abstract, Zoltán Ésik studies Equational properties of stratified least fixed points associated with logic programs. Adrian Paschke and Tara Athan show in their paper Law Test Suites for Semantically-Safe Rule Interchange how basic principles of nonmonotonic reasoning can be adapted to verify a particular semantics. In Static and Dynamic Views on the Algebra of Modular Systems, Eugenia Ternovska elaborates on properties of a knowledge representation framework called Algebra of Modular Systems; in particular, she uses the algebra for a high-level encoding of problem solving on graphs.
ACKNOWLEDGMENTS

First of all, we would like to thank the members of our Program Committee who provided us with excellent reviews on time. We are also grateful to the South African Department of Science and Technology (DST), the Council for Scientific and Industrial Research (CSIR) in South Africa, The South African Centre for Artificial Intelligence Research (CAIR), Principles of Knowledge Representation and Reasoning, Incorporated (KR, Inc.), and the European Association for Artificial Intelligence (EurAI, formerly ECCAI) for supporting our workshop by donations. The team of local organizers – Tommie Meyer in the first place – did a perfect job to help us making NMR 2016 a great event, thanks to all of them.

Last, but not least, we wish to thank Christian Eichhorn who took care of the NMR 2016 website and set up these proceedings, and the Technische Universität Dortmund where the proceedings were published as a technical report of the Faculty of Computer Science.

PROGRAM COMMITTEE OF NMR 2016

Gabriele Kern-Isberner, Technische Universität Dortmund (Co-Chair)
Renata Wassermann, University of São Paulo (Co-Chair)
Christoph Beierle, University of Hagen
Alexander Bochman, Holon Institute of Technology
Gerhard Brewka, Leipzig University
Jan Broersen, Utrecht University
Marina De Vos, University of Bath
Marc Denecker, K.U.Leuven
Juergen Dix, Clausthal University of Technology
Paul Dunne, University of Liverpool
Wolfgang Faber, University of Huddersfield
Eduardo Fermé, Universidade da Madeira
Martin Gebser, University of Potsdam
Michael Gelfond, Texas Tech University
Valentina Gliozzi, Università di Torino
Lluis Godo, Artificial Intelligence Research Institute, Barcelona
Sven Ove Hansson, Royal Institute of Technology, Stockholm
Andreas Herzig, Université Paul Sabatier, Toulouse
Anthony Hunter, University College London
Katsumi Inoue, National Institute of Informatics, Japan
Tomi Janhunen, Aalto University
Sébastien Konieczny, Université d’Artois
Gerhard Lakemeyer, Aachen University of Technology
Thomas Lukasiewicz, University of Oxford
Maria Vanina Martinez, Universidad Nacional del Sur (Bahia Blanca)
Thomas Meyer, University of Cape Town
Nir Oren, University of Aberdeen
Maurice Pagnucco, The University of New South Wales
Odile Papini, Aix-Marseille Université
Pavlos Peppas, University of Technology, Sydney
Laurent Perrussel, Université de Toulouse
Ramon Pino Perez, Universidad de Los Andes
Henri Prade, Université Paul Sabatier, Toulouse
Ken Satoh, National Institute of Informatics, Japan
Luigi Sauro, University of Naples "Federico II"
Steven Schockaert, Cardiff University
Gerardo Simari, Universidad Nacional del Sur (Bahia Blanca)
Guillermo Simari, Universidad Nacional del Sur (Bahia Blanca)
Hannes Strass, Leipzig University
Heiner Stuckenschmidt, University of Mannheim
Evgenia Ternovska, Simon Fraser University
Matthias Thimm, Universität Koblenz-Landau
Mirek Truszczynski, University of Kentucky
Ivan Varzinczak, Universidade Federal do Rio de Janeiro
Joost Vennekens, K.U. Leuven
Serena Villata, INRIA Sophia Antipolis
Kewen Wang, Griffith University
Emil Weydert, University of Luxembourg
Stefan Woltran, TU Wien
CONTENTS

Invited Talks

Laura Giordano: Reasoning about typicality in preferential description logics ........................................... 1

Leon van der Torre: Arguing about obligations and permissions ...... 3

Regular Papers

Ringo Baumann, Thomas Linsbichler and Stefan Woltran: Verifiability of Argumentation Semantics ......................... 5

Katarina Britz, Ivan Varzinczak: Preferential Modalities Revisited .... 15

Kristijonas Čyras, Francesca Toni: Properties of ABA+ for Non-Monotonic Reasoning ........................................ 25

Zoltan Esik: Equational properties of stratified least fixed points .... 35

Marco Garapa, Eduardo Fermé, Maurício D. L. Reis: Studies on Brutal Contraction and Severe Withdrawal: Preliminary Report ............ 37

Valentina Gliozzi: A strengthening of rational closure in DLs: reasoning about multiple aspects ........................................ 47

Adrian Haret, Jean-Guy Mailly and Stefan Woltran: Distributing Knowledge into Simple Bases .............................. 55

Jesse Heyninck, Christian Straßer: Relations between assumption-based approaches in nonmonotonic logic and formal argumentation ..... 65

Aaron Hunter: Ordinal Conditional Functions for Nearly Counterfactual Revision ................................................ 77

Thomas Linsbichler, Jörg Pührer, Hannes Strass: Characterizing Realizability in Abstract Argumentation ...................... 85

Jean-Guy Mailly: Using Enthymemes to Fill the Gap between Logical Argumentation and Revision of Abstract Argumentation Frameworks ........................................... 95

Özgür L. Özçep: Iterated Ontology Revision by Reinterpretation .... 105
Adrian Paschke and Tara Athan: Law Test Suites for Semantically-Safe Rule Interchange ................................................................. 115

Gavin Rens: On Stochastic Belief Revision and Update and their Combination .................................................. 123

Gavin Rens, Thomas Meyer, Giovanni Casini: Revising Incompletely Specified Convex Probabilistic Belief Bases ......................... 133

Zeynep G. Saribatur and Thomas Eiter: Reactive Policies with Planning for Action Languages ............................................................. 143

Eugenia Ternovska: Static and Dynamic Views on the Algebra of Modular Systems ................................................................. 153

Marc van Zee and Dragan Doder: AGM-Style Revision of Beliefs and Intentions from a Database Perspective (Preliminary Version) .... 163

Zhiqiang Zhuang, James Delgrande, Abhaya Nayak, Abdul Sattar: A New Approach for Revising Logic Programs ............................. 171
Reasoning about typicality in preferential description logics

Laura Giordano
Dipartimento di Scienze e Innovazione Tecnologica
Istituto di Informatica
Università del Piemonte Orientale
laura.giordano@uniupo.it

Abstract
The talk presents an approach for reasoning about typicality in description logics (DLs), based on Kraus, Lehmann and Magidor’s preferential semantics. It describes an extension to DLs of Lehmann and Magidor’s rational closure as well as a semantic characterization for it through a minimal model semantics. For expressive description logics, the computation of rational closure can exploit a polynomial encoding of preferential entailment into standard DLs. The talk aims at discussing to what extent preferential reasoning can be considered as a building block for dealing with exceptions in OWL ontologies, considering that dealing with exceptions is a long standing problem of nonmonotonic reasoning, and at comparing this approach with other approaches to defeasible reasoning in DLs.
Arguing about obligations and permissions

Leon van der Torre
University of Luxembourg

Abstract

In this talk I apply a theory of structured argumentation to normative reasoning. In an ASPIC+ style setting, I discuss the definition of argument and attack, the role of constitutive and permissive norms, and hierarchical normative systems. Based on joint work with Beishui Liao, Nir Oren, Gabriella Pigozzi and Serena Villata.
Verifiability of Argumentation Semantics∗

Ringo Baumann
Leipzig University
Germany

Thomas Linsbichler and Stefan Woltran
TU Wien
Austria

Abstract

Dung’s abstract argumentation theory is a widely used formalism to model conflicting information and to draw conclusions in such situations. Hereby, the knowledge is represented by so-called argumentation frameworks (AFs) and the reasoning is done via semantics extracting acceptable sets. All reasonable semantics are based on the notion of conflict-freeness which means that arguments are only jointly acceptable when they are not linked within the AF. In this paper, we study the question which information on top of conflict-free sets is needed to compute extensions of a semantics at hand. We introduce a hierarchy of so-called verification classes specifying the required amount of information. We show that well-known standard semantics are exactly verifiable through a certain such class. Our framework also gives a means to study semantics lying inbetween known semantics, thus contributing to a more abstract understanding of the different features argumentation semantics offer.

Introduction

In the late 1980s the idea of using argumentation to model nonmonotonic reasoning emerged (see (Loui 1987; Pollock 1987) as well as (Prakken and Vreeswijk 2002) for excellent overviews). Nowadays argumentation theory is a vibrant subfield of Artificial Intelligence, covering aspects of knowledge representation, multi-agent systems, and also philosophical questions. Among other approaches which have been proposed for capturing representative patterns of inference in argumentation theory (Besnard et al. 2014), Dung’s abstract argumentation frameworks (AFs) (Dung 1995) play an important role within this research area. At the heart of Dung’s approach lie the so-called argumentation semantics (cf. (Baroni, Caminada, and Giacomin 2011) for an excellent overview). Given an AF $F$, which is set-theoretically just a directed graph encoding arguments and attacks between them, a certain argumentation semantics $\sigma$ returns acceptable sets of arguments $\sigma(F)$, so-called $\sigma$-extensions. Each of these sets represents a reasonable position w.r.t. $F$ and $\sigma$.

Over the last 20 years a series of abstract argumentation semantics were introduced. The motivations of these semantics range from the desired treatment of specific examples to fulfilling a number of abstract principles. The comparison via abstract criteria of the different semantics available is a topic which emerged quite recently in the community ((Baroni and Giacomin 2007b) can be seen as the first paper in this line). Our work takes a further step towards a comprehensive understanding of argumentation semantics. In particular, we study the following question: Do we really need the entire AF $F$ to compute a certain argumentation semantics $\sigma$? In other words, is it possible to unambiguously determine acceptable sets w.r.t. $\sigma$, given only partial information of the underlying framework $F$.

In order to solve this problem let us start with the following reflections:

1. As a matter of fact, one basic requirement of almost all existing semantics is that of conflict-freeness, i.e. arguments within a reasonable position are not allowed to attack each other. Consequently, knowledge about conflict-free sets is an essential part for computing semantics.

2. The second step is to ask the following: Which information on top on conflict-free sets has to be added? Imagine the set of conflict-free sets given by $\{\emptyset, \{a\}, \{b\}\}$. Consequently, there has to be at least one attack between $a$ and $b$. Unfortunately, this information is not sufficient to compute any standard semantics (except naive extensions, which are defined as $\subseteq$-maximal conflict-free sets) since we know nothing precise about the neighborhood of $a$ and $b$. The following three AFs possess exactly the mentioned conflict-free sets, but differ with respect to other

$$F : a \rightarrow b \quad G : a \quad b \quad H : a \rightarrow b$$

3. The final step is to try to minimize the added information. That is, which kind of knowledge about the neighborhood is somehow dispensable in the light of computation? Clearly, this will depend on the considered semantics. For instance, in case of stage semantics (Verheij 1996), which requests conflict-free sets of maximal range, we do not need any information about incoming attacks. This information can not be omitted in case of admissible-based semantics since incoming attacks require counterattacks.

The above considerations motivate the introduction of so-called verification classes specifying a certain amount of

∗This research has been supported by DFG (project BR 1817/7-1) and FWF (projects I1102 and P25521).

1See (Jakobovits and Vermeir 1999; Arieli 2012; Grossi and Modgil 2015) for exemptions.
information. In a first step, we study the relation of these classes to each other. We therefore introduce the notion of being more informative capturing the intuition that a certain class can reproduce the information of an other. We present a hierarchy w.r.t. this ordering. The hierarchy contains 15 different verification classes only. This is due to the fact that many syntactically different classes collapse to the same amount of information.

We then formally define the essential property of a semantics \( \sigma \) being verifiable w.r.t. a certain verification class. We present a general theorem stating that any rational semantics is exactly verifiable w.r.t. one of the 15 different verification classes. Roughly speaking, a semantics is rational if attacks inbetween two self-loops can be omitted without affecting the set of extensions. An important aside hereby is that even the most informative class contains indeed less information than the entire framework by itself.

In this paper we consider a representative set of standard semantics. All of them satisfy rationality and thus, are exactly verifiable w.r.t. a certain class. Since the theorem does not provide an answer to which verification class perfectly matches a certain rational semantics we study this problem one by one for any considered semantics. As a result, only 6 different classes are essential to classify the considered standard semantics.

In the last part of the paper we study an application of the concept of verifiability. More precisely, we address the question of strong equivalence for semantics lying inbetween known semantics, so-called intermediate semantics. Strong equivalence is the natural counterpart to ordinary equivalence in monotonic theories (see (Oikarinen and Woltran 2011; Baumann 2016) for abstract argumentation and (Maher 1986; Lifschitz, Pearce, and Valverde 2001; Turner 2004; Truszczynski 2006) for other nonmonotonic theories). We provide characterization theorems relying on the notion of verifiability and thus, contributing to a more abstract understanding of the different features argumentation semantics offer. Besides these main results, we also give new characterizations for strong equivalence with respect to naive extensions and strong admissible sets.

### Preliminaries

An argumentation framework (AF) \( F = (A, R) \) is a directed graph whose nodes \( A \subseteq U \) (with \( U \) being an infinite set of arguments, so-called universe) are interpreted as arguments and whose edges \( R \subseteq A \times A \) represent conflicts between them. We assume that all AFs possess finitely\(^2\) many arguments only and denote the collection of all AFs by \( \mathcal{A} \). If \((a, b) \in R\) we say that \( a \) attacks \( b \). Alternatively, we write \( a \rightarrow b \) as well as, for some \( S \subseteq A \), \( a \rightarrow S \) or \( S \rightarrow b \) if there is some \( c \in S \) attacked by \( a \) or attacking \( b \), respectively. An argument \( a \in A \) is defended by a set \( S \subseteq A \) if for each \( b \in A \) with \( b \rightarrow a \), \( S \rightarrow b \). We define the range of \( S \) in \( F \) as \( S^+ = S \cup \{a \mid S \rightarrow a\} \). Similarly, we use \( S^- \) to denote the anti-range of \( S \) (in \( F \)) as \( S \cup \{a \mid a \rightarrow S\} \). Furthermore, we say that a set \( S \) is conflict-free (in \( F \)) if there is no argument \( a \in S \) s.t. \( S \rightarrow a \). The set of all conflict-free sets of an AF \( F \) is denoted by \( \text{cf}(F) \).

For an AF \( F = (B, S) \) we use \( A(F) \) and \( R(F) \) to refer to \( B \) and \( S \), respectively. Furthermore, we use \( L(F) = \{a \mid (a, a) \in R(F)\} \) for the set of all self-defeating arguments. Finally, we introduce the union of AFs \( F \) and \( G \) as \( F \cup G = (A(F) \cup A(G), R(F) \cup R(G)) \).

#### Semantics

A semantics \( \sigma \) assigns to each \( F = (A, R) \) a set \( \sigma(F) \subseteq 2^A \) where the elements are called \( \sigma \)-extensions. Numerous semantics are available. Each of them captures different intuitions about how to reason about conflicting knowledge. We consider \( \sigma \in \{\text{ad, na, stb, pr, co, gr, ss, stg, id, eq}\} \) for admissible, naive, stable, preferred, complete, grounded, semi-stable, stage, ideal, and eager semantics (Dung 1995; Caminada, Carnielli, and Dunne 2012; Verheij 1996; Dung, Mancarella, and Toni 2007; Caminada 2007).

**Definition 1.** Given an AF \( F = (A, R) \) and let \( S \subseteq A \).

1. \( S \in \text{ad}(F) \) iff \( S \in \text{cf}(F) \) and each \( a \in S \) is defended by \( S \).
2. \( S \in \text{na}(F) \) iff \( S \in \text{cf}(F) \) and there is no \( S' \in \text{cf}(F) \) s.t. \( S \subseteq S' \).
3. \( S \in \text{stb}(F) \) iff \( S \in \text{cf}(F) \) and \( S^+_F = A \).
4. \( S \in \text{pr}(F) \) iff \( S \in \text{ad}(F) \) and there is no \( S' \in \text{ad}(F) \) s.t. \( S \subseteq S' \).
5. \( S \in \text{co}(F) \) iff \( S \in \text{ad}(F) \) and for any \( a \in A \) defended by \( S, a \in S \).
6. \( S \in \text{gr}(F) \) iff \( S \in \text{co}(F) \) and there is no \( S' \in \text{co}(F) \) s.t. \( S' \subseteq S \).
7. \( S \in \text{ss}(F) \) iff \( S \in \text{ad}(F) \) and there is no \( S' \in \text{ad}(F) \) s.t. \( S^+_F \subseteq S^+_F \).
8. \( S \in \text{stg}(F) \) iff \( S \in \text{cf}(F) \) and there is no \( S' \in \text{cf}(F) \) s.t. \( S^+_F \subseteq S^+_F \).
9. \( S \in \text{id}(F) \) iff \( S \in \text{ad}(F), S \subseteq \bigcap \text{pr}(F) \) and there is no \( S' \in \text{ad}(F) \) satisfying \( S' \subseteq \bigcap \text{pr}(F) \) s.t. \( S \subseteq S' \).
10. \( S \in \text{eg}(F) \) iff \( S \in \text{ad}(F), S \subseteq \bigcap \text{ss}(F) \) and there is no \( S' \in \text{ad}(F) \) satisfying \( S' \subseteq \bigcap \text{ss}(F) \) s.t. \( S \subseteq S' \).

For two semantics \( \sigma, \tau \) we use \( \sigma \subseteq \tau \) to indicate that \( \sigma(F) \subseteq \tau(F) \) for each AF \( F \in \mathcal{A} \). If we have \( \rho \subseteq \sigma \subseteq \tau \) for semantics \( \rho, \sigma, \tau \), we say that \( \rho \) is \( \sigma, \tau \)-intermediate. Well-known relations between semantics are \( \text{stb} \subseteq \text{ss} \subseteq \text{pr} \subseteq \text{co} \subseteq \text{ad} \), meaning, for instance, that \( \text{ss} \) is \( \text{stb-pr} \)-intermediate.

**Definition 2.** We call a semantics \( \sigma \) rational if self-loop-chains are irrelevant. That is, for every AF \( F \) it holds that \( \sigma(F) = \sigma(F^1) \), where \( F^1 = (A(F), R(F) \setminus \{(a, b) \in R(F) \mid (a, a) \in R(F)\}) \). Indeed, all semantics introduced in Definition 1 are rational. A prominent semantics that is based on conflict-free sets, but is not rational is the \( \text{cf}2 \)-semantics (Baroni, Giacomin, and Guida 2005), since here chains of self-loops can have an influence on the SCCs of an AF (see also (Gaggl and Woltran 2013)).
Equivalence and Kernels

The following definition captures the two main notions of equivalence available for non-monotonic formalisms, namely ordinary (or standard) equivalence and strong (or expansion) equivalence. A detailed overview of equivalence notions including their relations to each other can be found in (Baumann and Brewka 2013; 2015).

**Definition 3.** Given a semantics $\sigma$. Two AFs $F$ and $G$ are

- **standard equivalent** w.r.t. $\sigma$ ($F \equiv^E G$) iff $\sigma(F) = \sigma(G)$,
- **expansion equivalent** w.r.t. $\sigma$ ($F \equiv^E_G G$) iff for all AFs $H$: $F \cup H \equiv^E_G G \cup H$.

Expansion equivalence can be decided syntactically via so-called kernels (Oikarinen and Woltran 2011). A kernel is a function $k : \mathcal{A} \rightarrow \mathcal{A}$ mapping each AF $F$ to another AF $k(F)$ (which we may also denote as $F^k$). Consider the following definitions.

**Definition 4.** Given an AF $F = (A, R)$ and a semantics $\sigma$. We define $\sigma$-kernels $F^{k(\sigma)} = (A, R^{k(\sigma)})$ whereby

- $R^{k(\sigma)}(a,b) = R \setminus \{(a,b) | a \neq b, (a,a) \in R\}$,
- $R^{k(ad)}(a,b) = R \setminus \{(a,b) | a \neq b, (a,a) \in R, (b,b) \not\in R\}$,
- $R^{k(pr)}(a,b) = R \setminus \{(a,b) | a \neq b, (a,b) \in R \land B \neq \emptyset\}$,
- $R^{k(co)}(a,b) = R \setminus \{(b,a) | a \neq b, (b,b) \not\in R\}$.

Expansion equivalence can be decided syntactically via so-called kernels (Oikarinen and Woltran 2011). A kernel is a function $k : \mathcal{A} \rightarrow \mathcal{A}$ mapping each AF $F$ to another AF $k(F)$ (which we may also denote as $F^k$). Consider the following self-referential definition taken from (Caminada 2014).

In accordance with Definition 5 we observe that both AFs possess the same naive kernel $H = F^{k(\mu)} = G^{k(\mu)}$.

The following theorem proves that possessing the same kernels is necessary as well as sufficient for being strongly equivalent, i.e. $F \equiv^E G$.

**Theorem 2.** For all AFs $F, G$,

$F \equiv^E G \iff F^{k(\mu)} = G^{k(\mu)}$.

**Proof.** In (Baumann and Woltran 2014) it was already shown that $F \equiv^E G$ iff jointly $A(F) = A(G)$ and $\text{na}(F) = \text{na}(G)$. Consequently, it suffices to prove that $F^{k(\mu)} = G^{k(\mu)}$ implies $A(F) = A(G)$ as well as $\text{na}(F) = \text{na}(G)$ and vice versa.

($\Rightarrow$) Given $F^{k(\mu)} = G^{k(\mu)}$. By Definition 5 we immediately have $A(F) = A(G)$. Assume now that $\text{na}(F) \neq \text{na}(G)$ and without loss of generality let $S \subseteq \text{na}(F) \setminus \text{na}(G)$. Obviously, for any AF $H$, $c_f(H) = c_f(H^{k(\mu)})$. Hence, there is an $S'$, s.t. $S \subseteq S' \subseteq c_f(G) \subseteq c_f(F)$. Thus, there are $a,b \in S' \setminus S$, s.t. $(a,b) \not\in R(F) \land R(G)$. Furthermore, $(a,b) \not\in R(G)$ and since for any AF $H$, $L(H) = L(H^{k(\mu)})$ we obtain $(a,b) \not\in R(F)$. Consequently, we have to consider $a \neq b$. Since $(a,b) \not\in R(F) \land R(G)$, we obtain $(a,b) \not\in R(F^{k(\mu)})$. Since $F^{k(\mu)} = G^{k(\mu)}$ is assumed we derive $(a,b), (b,a) \in R(G^{k(\mu)})$. By Definition 5 we must have $(a,b) \in R(G)$ contradicting the conflict-freeness of $S'$. In conclusion, $S \subseteq S'$.

($\Leftarrow$) We show the contrapositive, i.e. $F^{k(\mu)} \neq G^{k(\mu)}$ implies $A(F) \neq A(G)$ or $\text{na}(F) \neq \text{na}(G)$. Observe that for any AF $H, A(H) = A(H^{k(\mu)})$. Consequently, if $A(F^{k(\mu)}) \neq A(G^{k(\mu)})$, then $A(F) \neq A(G)$. Assume now $R(F^{k(\mu)}) \neq R(G^{k(\mu)})$. Without loss of generality let $(a,b) \in R(F^{k(\mu)}) \setminus R(G^{k(\mu)})$. Since for any AF $H$, $L(H) = L(H^{k(\mu)})$ we obtain $a \neq b$. Furthermore, $(a,b) \in R(F^{k(\mu)})$ implies $(a,b), (b,a), (a,a), (b,b) \in R(F) \setminus (F^{k(\mu)}) \land (F^{k(\mu)}) \setminus (F) \neq \emptyset$ and consequently, for any $S \subseteq \text{na}(F)$, $(a,b) \not\subseteq S$. Since $(a,b) \not\subseteq S$ we deduce $(a,b), (b,a), (a,a) \not\in R(F)$ and consequently, for any $S \subseteq \text{na}(F)$, $(a,b) \not\subseteq S$. Hence, we have $S \subseteq \text{na}(F)$ and thus, there exists a set $S \subseteq \text{na}(G)$, s.t. $(a,b) \not\subseteq S$ (compare (Baumann and Spanring 2015, Lemma 3)) witnessing $\text{na}(F) \neq \text{na}(G)$.

We turn now to strongly admissible sets (for short, sad) (Baroni and Giacomin 2007b). We will show that, beside grounded (Oikarinen and Woltran 2011) and resolution based grounded semantics (Baroni, Dunne, and Giacomin 2011; Dvorák et al. 2014), strongly admissible sets are characterizable through the grounded kernel. Consider the following self-referential definition taken from (Caminada 2014).

### Complementing Previous Results

In order to provide an exhaustive analysis of intermediate semantics (confer penultimate section) we provide missing kernels for naive semantics as well as strongly admissible sets. We start with the so-called naive kernel characterizing expansion equivalence w.r.t. naive semantics. As an aside, the following kernel is the first one which adds attacks to the former attack relation.

**Definition 5.** Given an AF $F = (A, R)$. We define the naive kernel $F^{k(\mu)} = (A, R^{k(\mu)})$ whereby $R^{k(\mu)} = R \cup \{(a,b) | a \neq b, (a,a), (b,b) \not\in R\}$.

The following example illustrates the definition above.

**Example 1.** Consider the AFs $F$ and $G$. Note that $\text{na}(F) = na(G) = \{a,c\}$ and consequently, $F \equiv^E G$.

\[
F : \begin{array}{ccc}
  & a & b & c & d \\
  a & & & & \\
  b & & & & \\
  c & & & & \\
  d & & & & \\
\end{array}
\]

\[
G : \begin{array}{ccc}
  & a & b & c & d \\
  a & & & & \\
  b & & & & \\
  c & & & & \\
  d & & & & \\
\end{array}
\]

In accordance with Definition 5 we observe that both AFs possess the same naive kernel $H = F^{k(\mu)} = G^{k(\mu)}$. 

In accordance with Definition 5 we observe that both AFs possess the same naive kernel $H = F^{k(\mu)} = G^{k(\mu)}$.
Definition 6. Given an AF $F = (A, R)$. A set $S \subseteq A$ is strongly admissible, i.e. $S \in \text{sad}(F)$ iff any $a \in S$ is defended by a strongly admissible set $S' \subseteq S \setminus \{a\}$.

The following properties are needed to prove the characterization theorem. The first two of them are already shown in (Baroni and Giacomin 2007a).

Proposition 1. Given two AFs $F$ and $G$, then
1. $\text{gr}(F) \subseteq \text{sad}(F) \subseteq \text{ad}(F)$.
2. If $S \in \text{gr}(F)$ we have: $S' \subseteq S$ for all $S' \in \text{sad}(F)$, and
3. $\text{sad}(F) = (G) \iff \text{gr}(F) = (G)$.

The following definition provides us with an alternative criterion for being strongly admissible set. In contrast to the former it allows one to construct strong admissible sets step by step. Thus, a construction method is given.

Definition 7. Given an AF $F = (A, R)$. A set $S \subseteq A$ is strongly admissible, i.e. $S \in \text{sad}(F)$ iff there are finitely many and pairwise disjoint sets $A_1, \ldots, A_n$, s.t. $S = \bigcup_{1 \leq i \leq n} A_i$, $A_i \not\subseteq \bigcup_{1 \leq j \leq n} A_j$防守s $A_j$ for $1 \leq j \leq n - 1$ and $\bigcup_{1 \leq i \leq n} A_i$ defends $A_{n+1}$.

Proof. For the proof we use $S \in \text{sad}(F)$ as a shorthand for $S \in \text{sad}(F)$ in the sense of Definition $k$.

$(\Rightarrow)$ Given $S \in \text{sad}(F)$. Hence, there is a finite partition, s.t. $S = \bigcup_{1 \leq i \leq n} A_i$, $A_i \not\subseteq \bigcup_{1 \leq j \leq n} A_j$防守s $A_j$ for $1 \leq j \leq n - 1$. Observe that $\bigcup_{1 \leq i \leq n} A_i$ defends $A_j$ for any $j \leq n$. Consequently, there is an index $i^*$, s.t. $a \in A_{i^*}$. Furthermore, since $\bigcup_{1 \leq i \leq n} A_i$ defends $A_{i^*}$ by definition, we deduce that $\bigcup_{1 \leq i \leq n} A_i \subseteq S \setminus \{a\}$ defends $a$. We have to show now that (the smaller set w.r.t) $\bigcup_{1 \leq i \leq n} A_i \subseteq \text{sad}(F)$. Note that $\bigcup_{1 \leq i \leq n} A_i \subseteq \text{sad}(F)$. Since we are dealing with finite AFs we may iterate our construction. Hence, no matter which elements are chosen we end up with a $\leq$-chain, s.t. $\emptyset \subseteq \bigcup_{1 \leq i \leq n} A_i \subseteq S \setminus \{a\}$ and $\emptyset \not\subseteq \bigcup_{1 \leq i \leq n} A_i$ defends $a$ for some index $i$, set $S_i$ and element $a_i$. This means, the question whether $S \in \text{sad}(F)$ can be decided positively by proving $\emptyset \not\subseteq \text{sad}(F)$. Since the empty set does not contain any elements we find $\emptyset \not\subseteq \text{sad}(F)$ concluding $\text{sad}(F) \subseteq \text{sad}(F)$.

$(\Leftarrow)$ Given $S \in \text{sad}(F)$, consider the following sets $S_i$: $S_1 = (\Gamma(\emptyset) \setminus \emptyset) \cap S$, $S_2 = (\Gamma(S_1) \setminus S_1) \cap S$, ..., $S_n = (\Gamma(S_{n-1}) \setminus S_{n-1}) \cap S$. We are dealing with finite AFs there has to be a natural $n \in \mathbb{N}$, s.t. $S_n = S_{n+1} = S_{n+2} = \ldots$. Consider now the union of these sets, i.e. $\bigcup_{1 \leq i \leq n} S_i$. We show now that $\bigcup_{1 \leq i \leq n} S_i \in \text{sad}(F)$ and $\bigcup_{1 \leq i \leq n} S_i$ defends $S$. By construction we have $S_1 \subseteq \Gamma(\emptyset)$. Moreover, $\bigcup_{1 \leq i \leq n} S_i$ defends $S_{i+1}$ for $1 \leq j \leq n - 1$. This can be seen as follows. By definition $S_{j+1} = \left(\Gamma(\bigcup_{1 \leq i \leq j} S_i) \setminus \bigcup_{1 \leq i \leq j} S_i \right) \cap S$. This means, $S_{j+1} \subseteq \Gamma(\bigcup_{1 \leq i \leq j} S_i)$.

The following example shows how to use the new construction method.

Example 2. Consider the following AF $F$. 

$$ F = \{a, b, c, d, e, f\} $$ 

We have $\Gamma(\emptyset) = \{a, d\}$, hence, for all $S \subseteq \{a, d\}$, $S \in \text{sad}(F)$.

Furthermore, $\Gamma(\{a\}) = \{a, c, f\}$, $\Gamma(\{d\}) = \{d, f\}$ and $\Gamma(\{a, d\}) = \{a, d, c, f\}$. This means, additionally $\Gamma(\{a\}, \{d\}) = \{a, d, e, c, f\}$,

Corollary 1. Given an AF $F$ and two sets $B, B' \subseteq A(F)$. If $B$ defends $B'$, then $B \cup B'$ is strongly admissible if $B$. If $B$ does not defend $B'$, then $B \cup B'$ is not strongly admissible.

The following lemma shows that the grounded kernel is insensitive w.r.t. strong admissible sets.

Lemma 1. For any AF $F$, $\text{sad}(F) = \text{sad}(F^{k(\Gamma)})$.

Proof. The grounded kernel is node- and loop-preserving, i.e. $A(F) = A(F^{k(\Gamma)})$ and $L(F) = L(F^{k(\Gamma)})$. Furthermore, $\text{cf}(F) = \text{cf}(F^{k(\Gamma)})$ and $\Gamma(\emptyset) = \Gamma_{f+k(\Gamma)}(\emptyset)$ as shown in (Oikarinen and Woltran 2011, Lemma 6).

$(\subseteq)$ Given $S \in \text{sad}(F)$. The proof is by induction on $n$ indicating the number of sets forming a suitable (according to Definition 7) partition of $S$. Let $n = 1$. In consideration of the grounded kernel we observe $\Gamma(\emptyset) = \Gamma_{f+k(\Gamma)}(\emptyset)$, i.e. the set of unattacked arguments does not change. Since $S \subseteq \Gamma(\emptyset)$ assumed are done. Assume now that the assertion is proven for any $k$-partition. Let $S$ be a $(k+1)$-partition, i.e. $S = \bigcup_{1 \leq i \leq k+1} A_i$. According to induction hypothesis as well as Corollary 1 it suffices to prove $\bigcup_{1 \leq i \leq k+1} A_i$ defends $A_{k+1}$ in $F^{k(\Gamma)}$. Assume not, i.e. there are arguments $b \in A(F) \setminus S$, $c \in A_{k+1}$ s.t. $R(F^{k(\Gamma)}) \not\subseteq R(F)$ and for all $a \in A_k$, $(a, b) \not\in R(F^{k(\Gamma)})$. Since $\bigcup_{1 \leq i \leq k} A_i$ defends $A_{k+1}$ in $F$ we deduce the existence of an argument $a \in \bigcup_{1 \leq i \leq k} A_i$ s.t. $(a, b) \in R(F)$. Thus, $(a, b)$ is redundant w.r.t. the grounded kernel. According to Definition 4 and due to
the conflict-freeness of $\bigcup_{i=1}^{k} A_i$ we have $(a, a) \notin R(F)$ and $(b, a), (b, b) \in R(F)$. Consequently, $(b, a) \in F_{k(gr)}$. Hence there has to be an argument $a \in \bigcup_{i=1}^{k} A_i$, s.t. $(a, b) \in R(F_{k(gr)})$, contradicting $(*)$.

(2) Assume $S \in sad(F_{k(gr)})$. We show $S \in sad(F)$ by induction on $n$ indicating that $S$ is a $n$-partition in $F_{k(gr)}$.

Due to $\Gamma_{F}(\emptyset) = \Gamma_{F_{k(gr)}}(\emptyset)$ the base case is immediately clear. For the induction step let $S$ be a $(k+1)$-partition, i.e. $S = \bigcup_{i=1}^{k+1} A_i$. By induction hypothesis we may assume that $\bigcup_{i=1}^{k} A_i$ is strongly admissible in $F$. Using Corollary 1 it suffices to prove $\bigcup_{i=1}^{k} A_i$ defends $A_{k+1}$ in $F$. Assume not, i.e. there are arguments $b \in A(F) \setminus S, c \in A_{k+1}$ s.t. $(b, c) \in R(F)$ and for all $a \in \bigcup_{i=1}^{k} A_i$, $(a, b) \notin R(F)$. We even have $(a, b) \notin R(F_{k(gr)})$ since $R(F_{k(gr)}) \subseteq R(F)$. Consequently, $(b, c)$ has to be deleted in $F_{k(gr)}$. Definition 4 requires $(c, c) \in R(F_{k(gr)})$ contradicting the conflict-freeness of $S$ in $F_{k(gr)}$.

\[\square\]

**Theorem 3.** For any two AFs $F$ and $G$ we have,

\[F \equiv_{sad} G \iff F_{k(gr)} = G_{k(gr)}\]

**Proof.** $(\Rightarrow)$ We show the contrapositive, i.e. $F_{k(gr)} \neq G_{k(gr)} \Rightarrow F \not\equiv_{sad} G$. Assuming $F_{k(gr)} \neq G_{k(gr)}$ implies $F \not\equiv_{E_{k}} G$ (Theorem 1). This means, there is an AF $H$, s.t. $gr(F \cup H) \neq gr(G \cup H)$. Due to statement 3 of Proposition 1, we deduce $sad(F \cup H) \neq sad(G \cup H)$ proving $F \not\equiv_{sad} G$.

$(\Leftarrow)$ Given $F_{k(gr)} = G_{k(gr)}$. Since expansion equivalence is a congruence w.r.t. $\cup$ we obtain $(F \cup H)^{k(gr)} = (G \cup H)^{k(gr)}$ for any AF $H$. Consequently, $sad((F \cup H)^{k(gr)}) = sad((G \cup H)^{k(gr)})$. Due to Lemma 1 we deduce $sad(F \cup H) = sad(G \cup H)$, concluding the proof. \[\square\]

**Verifiability**

In this section we study the question whether we really need the entire AF $F$ to compute the extensions of a given semantics. Let us consider naïve semantics. Obviously, in order to determine naïve extensions it suffices to know all conflict-free sets. Conversely, knowing $cf(F)$ only does not allow to reconstruct $F$ unambiguously. This means, knowledge about $cf(F)$ is indeed less information than the entire AF by itself.

In fact, most of the existing semantics do not need information of the entire framework. We will categorize the amount of information by taking the conflict-free sets as a basis and distinguish between different amounts of knowledge about the neighborhood, that is range and anti-range, of these sets.

**Definition 8.** We call a function $r^{x} : 2^{U_{1}} \times 2^{U_{2}} \rightarrow (2^{U})^{n}$ ($n > 0$) which is expressible via basic set operations only neighborhood function. A neighborhood function $r^{x}$ induces the verification class mapping each AF $F$ to

\[\tilde{F^{x}} = \{(S, r^{x}(S^{+}_{F}, S^{-}_{F})) \mid S \in cf(F)\}\]

We coined the term neighborhood function because the induced verification classes apply these functions to the neighborhoods, i.e. range and anti-range of conflict-free sets. The notion of expressible via basic set operations simply means that (in case of $n = 1$) the expression $r^{x}(A, B)$ is in the language generated by the following BNF:

\[X ::= A | B | (X \cup U) | (X \cap U) | (X \setminus U)\]

Consequently, in case of $n = 1$, we may distinguish eight set theoretically different neighborhood functions, namely $r^{x}(S, S') = 0$

\[r^{+}(S, S') = S\]

\[r^{x}(S, S') = S' \setminus S\]

\[r^{\Delta}(S, S') = (S \cap S') \setminus (S \cup S')\]

A verification class encapsulates a certain amount of information about an AF, as the following example illustrates.

**Example 3.** Consider the following AF $F$:

\[F : a \quad b \quad c\]

Now take, for instance, the verification class induced by $r^{+}$, that is $\tilde{F^{+}} = \{(S, r^{+}(S^{+}_{F}, S^{-}_{F})) \mid S \in cf(F)\} = \{(S, S^{+}_{F}) \mid S \in cf(F)\}$, storing information about conflict-free sets together with their associated ranges w.r.t. $F$. It contains the following tuples: $(\emptyset, \emptyset), (\{a\}, \{b\}), (\{c\}, \{b\})$, and $(\{a, c\}, \{b\})$. The verification class induced by $r^{\Delta}$ contains the same tuples but instead of $(\{a\}, \emptyset)$ instead of $(\{a\}, \emptyset)$. Nevertheless, it should be clear that the set $\tilde{F^{+}}$ suffices to compute stage extensions (i.e., range-maximal conflict-free sets) of $F$. This intuitive understanding of verifiability will be formally specified in Definition 10. Note that a neighborhood function $r^{x}$ may return $n$-tuples. Consequently, in consideration of the eight listed basic function we obtain (modulo reordering, duplicates, empty set) $2^{9}$ + syntactically different neighborhood functions and therefore the same number of verification classes. As usual, we will denote the $n$-ary combination of basic functions $(r^{x}(S, S'), \ldots, r^{x}(S, S'))$ as $r^{x}(S, S')$ with $x = x_{1} \ldots x_{n}$.

With the following definition we can put neighborhood functions into relation w.r.t. their information. This will help us to show that actually many of the induced classes collapse to the same amount of information.

**Definition 9.** Given neighborhood functions $r^{x}$ and $r^{y}$ returning $n$-tuples and $m$-tuples, respectively, we say that $r^{x}$ is more informative than $r^{y}$, for short $r^{x} \geq r^{y}$, iff there is a function $\delta : (2^{U})^{n} \rightarrow (2^{U})^{m}$ such that for any two sets of arguments $S, S' \subseteq U$, we have $\delta(r^{x}(S, S')) = r^{y}(S, S')$. 


We will denote the strict part of \( \leq \) by \( \prec \), i.e., \( \tau^x \succ \tau^y \) iff \( \tau^x \succeq \tau^y \) and \( \tau^y \nprec \tau^x \). Moreover \( \tau^x \cong \tau^y \) in case \( \tau^x \succeq \tau^y \) and \( \tau^y \succeq \tau^x \), we say that \( \tau^x \) represents \( \tau^y \) and vice versa.

**Lemma 2.** All neighborhood functions are represented by the ones depicted in Figure 1 and the \( \prec\)-relation represented by arcs in Figure 1 holds.

**Proof.** We begin by showing that all neighborhood functions are represented in Figure 1. Clearly, each neighborhood function \( \rho^x \) represents itself, i.e. \( \tau^x \cong \tau^x \). All neighborhood functions for \( n = 1 \) are depicted in Figure 1. We turn to \( n = 2 \). Consider the neighborhood functions \( \tau^{x \pm}, \tau^{y \pm}, \) and \( \tau^\pm \), defined as \( \tau^{x \pm} (S, S') = (S, S \land S') \), \( \tau^{y \pm} (S, S') = (S, S \lor S') \), and \( \tau^\pm (S, S') = (S \land S') \lor (S \lor S') \) for \( S, S' \subseteq \mathcal{U} \). Observe that \( S = (S, S') \cup (S \land S') \) and \( (S \land S') \cup (S \lor S') \). Hence, we can easily define functions in the spirit of Definition 9 mapping the images of the function to one another:

\[
\begin{align*}
\delta_1 (\tau^{x \pm} (S, S')) &= \delta_1 (S, S \setminus S') =_{\text{def}} (S, S \setminus (S \setminus S')) = (S, S \land S') = \tau^{\pm \land} (S, S'); \\
\delta_2 (\tau^{y \pm} (S, S')) &= \delta_2 (S, S \land S') =_{\text{def}} (S \setminus (S \land S'), S \land S') = (S, S' \land S) = \tau^{\pm \land} (S, S'); \\
\delta_3 (\tau^\pm (S, S')) &= \delta_3 (S \setminus (S \land S'), (S \land S')) =_{\text{def}} ((S \setminus S') \cup (S \land S'), (S \land S')) = \tau^{\pm \lor} (S, S').
\end{align*}
\]

Therefore, \( \tau^{x \pm} \cong \tau^{y \pm} \cong \tau^\pm \). In particular, they are all represented by \( \tau^\pm \). We can apply the same reasoning to other combinations of neighborhood functions and get the following equivalences w.r.t. information content: \( \tau^{x \pm \pm} \cong \tau^{y \pm \pm} \), \( \tau^{x \pm \pm} \cong \tau^\pm \), \( \tau^{y \pm \pm} \cong \tau^\pm \), and \( \tau^{x \pm \pm} \cong \tau^{y \pm \pm} \). Consequently, the functions stated first acting as representatives in Figure 1 by \( \tau^\pm \).

For the remaining functions returning 2-tuples we get \( \tau^{x \pm} \cong \tau^\pm \) by Figure 1:

\[
\begin{align*}
\delta_4 (\tau^{x \pm} (S, S')) &= \delta_4 (S, S \setminus S') =_{\text{def}} (S, (S \cup S') \setminus (S \setminus S')) = \tau^{\pm \Delta} (S, S'); \\
\delta_5 (\tau^\pm (S, S')) &= \delta_5 (S, (S \cup S') \setminus (S \setminus S')) =_{\text{def}} ((S \setminus (S \cup S') \setminus (S \setminus S')) \cup ((S \cup S') \setminus (S \setminus S')) \setminus S, (S \cup S') \setminus (S \setminus S')) = \tau^\pm (S, S'); \\
\delta_6 (\tau^{x \pm} (S, S')) &= \delta_6 (S, S \setminus (S \setminus S')) =_{\text{def}} ((S \setminus (S \setminus S')) \cup (S \setminus (S \setminus S')) \setminus S, (S \setminus (S \setminus S')) \setminus S, (S \setminus (S \setminus S')) \setminus S, (S \setminus S')) = \tau^{x \pm} (S, S').
\end{align*}
\]

Finally, every neighborhood function \( \tau^{x_{1 \cdots n}} \) with \( n \geq 3 \) is represented by \( \tau^{x_{1 \cdots n}} \) since we can compute all possible sets from \( S \) and \( S' \).

Now consider two functions \( \tau^x \) and \( \tau^y \) such that there is an arrow from \( x \) to \( y \) in Figure 1. It is easy to see that \( \tau^y \succ \tau^x \) since, for sets of arguments \( S \) and \( S' \), \( \tau^x (S, S') \) is either contained in \( \tau^y (S, S') \) or obtainable from \( \tau^y (S, S') \) by basic set operations. The fact that \( \tau^x \nprec \tau^y \), entails \( \tau^y \succ \tau^x \), follows from the impossibility of finding a function \( \delta \) such that \( \delta (\tau^x (S, S')) = \tau^y (S, S') \).

If the information provided by a neighborhood function is sufficient to compute the extensions, we say the semantics is verifiable by the class induced by the neighborhood function.

**Definition 10.** A semantics \( \sigma \) is verifiable by the verification class induced by the neighborhood function \( \tau^x \) returning \( n \)-tuples (or simply, \( x \)-verifiable) iff there is a function (also called criterion) \( \gamma_\sigma : (2^\mathcal{U})^n \times 2^\mathcal{U} \rightarrow 2^\mathcal{U} \) s.t. for every AF \( F \in \mathcal{A} \) we have:

\[
\gamma_\sigma (\bar{F}_\sigma, A(F)) = \sigma (F).
\]

Moreover, \( \sigma \) is exactly \( x \)-verifiable iff \( \sigma \) is \( x \)-verifiable and there is no verification class induced by \( \tau^x \) such that \( \sigma \) is \( y \)-verifiable.

Observe that if a semantics \( \sigma \) is \( x \)-verifiable then for any two AFs \( F \) and \( G \) with \( \bar{F}_\sigma = \bar{G}_\sigma \) and \( A(F) = A(G) \) it must hold that \( \sigma (F) = \sigma (G) \).

We proceed with a list of criteria showing that any semantics mentioned in Definition 1 is verifiable by a verification class induced by a certain neighborhood function. In the following, we abbreviate the tuple \((\bar{F}_\sigma, A(F))\) by \( \bar{F}_\sigma^{A} \).

\[
\begin{align*}
\gamma_{\text{max}} (\bar{F}_\sigma^{A}) &= \{S \mid S \in \bar{F}_\sigma, S is \ \subseteq \ -\text{maximal in } \bar{F}\}; \\
\gamma_{\text{ref}} (\bar{F}_\sigma^{A}) &= \{S \mid (S, S') \in \bar{F}^+, S' is \ \subseteq \ -\text{maximal in } (C^++ (C, C^+) \in \bar{F}^+\}; \\
\gamma_{\text{ad}} (\bar{F}_\sigma^{A}) &= \{S \mid S \in \bar{F}^+, S^+ = \{A\}; \\
\gamma_{\text{pr}} (\bar{F}_\sigma^{A}) &= \{S \mid (S, S') \in \bar{F}^+, S^T = \emptyset\}; \\
\gamma_{\text{ad}} (\bar{F}_\sigma^{A}) &= \{S \mid S \in \text{ad}(\bar{F}_\sigma^{A}), S is \ \subseteq \ -\text{maximal in } \text{ad}(\bar{F}_\sigma^{A})\}; \\
\gamma_{\text{ad}} (\bar{F}_\sigma^{A}) &= \{S \mid S \in \text{ad}(\bar{F}_\sigma^{A}), S^+ is \ \subseteq \ -\text{maximal in } (C^+ (C, C^+, C^+) \in \bar{F}^+\}, C \in \text{ad}(\bar{F}_\sigma^{A})\}; \\
\gamma_{\text{ad}} (\bar{F}_\sigma^{A}) &= \{S \mid C \in \text{ad}(\bar{F}_\sigma^{A}), C \subseteq \bigcap_{\text{ad}} (\bar{F}_\sigma^{A})\}; \\
\gamma_{\text{ad}} (\bar{F}_\sigma^{A}) &= \{S \mid S is \ \subseteq \ -\text{maximal in } (C \mid C \in \text{ad}(\bar{F}_\sigma^{A}), C \subseteq \bigcap_{\text{ad}} (\bar{F}_\sigma^{A})\); \\
\gamma_{\text{ad}} (\bar{F}_\sigma^{A}) &= \{S \mid S, S^+ \in \bar{F}^+\} \subseteq \emptyset, \forall (S_0, S_0^+, S_1^+, \ldots, S_n, S_n^+) \in \bar{F}^+; \\
\gamma_{\text{ad}} (\bar{F}_\sigma^{A}) &= (0 = S_0 \land \cdots \land S_n = S) \land \\
\forall i \in \{1, \ldots, n\} : S_i \subseteq S_i^+);)
\end{align*}
\]
\[ \gamma_\sigma(\tilde{\mathcal{F}}_A^+) = \{ S \mid S \in \gamma_\text{sad}(\tilde{\mathcal{F}}_A^+) \}, \\\n\gamma_\sigma(\tilde{\mathcal{F}}_A^-) = \{ S \mid (S, S^+, S^-) \in \tilde{\mathcal{F}}^+, \quad (S^- \setminus S^+) \neq \emptyset \}; \\\n\gamma_\text{co}(\tilde{\mathcal{F}}_A^+) = \{ S \mid (S, S^+, S^-) \in \tilde{\mathcal{F}}^+, \quad (S^- \setminus S^+) \neq \emptyset \}. \]

Instead of a formal proof, we give the following explanations. First of all, it is easy to see that the naive semantics is verifiable by the verification class induced by \( \tau^+ \) since the naive extensions can be determined by the conflict-free sets. Stable and stage semantics, on the other hand, utilize the range of each conflict-free set in addition. Hence they are verifiable by the verification class induced by \( \tau^+ \). Now consider admissible sets. Recall that a conflict-free \( S \) set is admissible if and only if it attacks all attackers. This is captured exactly by the condition \( S^+ = \emptyset \), hence admissible sets are verifiable by the verification class induced by \( \tau^+ \). The same holds for preferred semantics, since we just have to determine the maximal conflict-free sets with \( S^+ = \emptyset \). Semi-stable semantics, however, needs the range of each conflict-free set in addition, see \( \gamma_\text{ss} \), which makes it verifiable by the verification class induced by \( \tau^+ \). Finally consider the criterion \( \gamma_\text{co} \).

The first two conditions for a set of arguments \( S \) stand for conflict-freeness and admissibility, respectively. Now assume the third condition does not hold, i.e., there exists a tuple \((S, S^+, S^-) \in \tilde{\mathcal{F}}^-\) with \( S \supset S \) and \( S^- \setminus S^+ = \emptyset \). This means that every argument attacking \( S \) is attacked by \( S \), i.e., \( S \) is defended by \( S \). Hence \( S \) is not a complete extension, showing that \( \gamma_\text{co}(\tilde{\mathcal{F}}_A^+) = \text{co}(F) \) for each \( F \not\in \mathcal{A}' \). One can verify that all criteria from the list are adequate in the sense that they describe the extensions of the corresponding semantics.

We show now that the formal concepts of verifiability and being more informative behave correctly in the sense that the use of more informative neighborhood functions do not lead to a loss of verification capacity.

**Proposition 3.** If a semantics \( \sigma \) is \( x \)-verifiable, then \( \sigma \) is verifiable by all verification classes induced by some \( \tau^0 \) with \( \tau^0 \succeq \tau^+ \).

**Proof.** As \( \sigma \) is verifiable by the verification class induced by \( \tau^+ \) it holds that there is some \( \sigma \) such that for all \( F \in \mathcal{A}' \),
\[ \gamma_\sigma(\tilde{\mathcal{F}}_F^+, A(F)) = \sigma(F). \]
Now let \( \tau^0 \succeq \tau^+ \), meaning that there is some \( \delta \) such that \( \delta(\tau^0(S, S')) = \tau^0 \). We define
\[ \gamma_\sigma(\tilde{\mathcal{F}}_F^+, A(F))' = \gamma_\sigma(\{(S, \delta(S)) \mid (S, S') \in \tilde{\mathcal{F}}_F^+, A(F)\}) \]
and observe that \( \{(S, \delta(S)) \mid (S, S') \in \tilde{\mathcal{F}}_F^+, A(F)\} = \tilde{\mathcal{F}}_F^+, \) hence \( \gamma_\sigma(\tilde{\mathcal{F}}_F^+, A(F))' = \sigma(F) \) for each \( F \not\in \mathcal{A}' \). \( \Box \)

In order to prove unverifiability of a semantics \( \sigma \) w.r.t. a class induced by a certain \( \tau^0 \) it suffices to present two AFs \( F \) and \( G \) such that \( \sigma(F) \neq \sigma(G) \) but, \( \tilde{\mathcal{F}}^0 = \tilde{\mathcal{F}}^0 \) and \( A(F) = A(G) \). Then the verification class induced by \( \tau^0 \) does not provide enough information to verify \( \sigma \).

In the following we will use this strategy to show exact verifiability. Consider a semantics \( \sigma \) which is verifiable by a class induced by \( \tau^+ \). If \( \sigma \) is unverifiable by all verification classes induced by \( \tau^0 \) with \( \tau^0 \prec \tau^+ \) we have that \( \sigma \) is exactly verifiable by \( \tau^0 \). The following examples study this issue for the semantics under consideration.

**Example 4.** The complete semantics is \( ++ \)-verifiable as seen before. The following AFs show that it is even exactly verifiable by that class.

\[
F_1: a \quad b \\
F_1': a \\
F_2: a \\
F_2': a \\
F_3: a \\
F_3': a \\
F_4: a \\
F_4': a \\
F_5: a \\
F_5': a \\
F_6: a \\
F_6': a \\
F_7: a \\
F_7': a \\
F_8: a \\
F_8': a
\]

First consider the AFs \( F_1 \) and \( F_1' \), and observe that \( \tilde{\mathcal{F}}_1^+ = \tilde{\mathcal{F}}_1^- = \{(\emptyset, \emptyset, \emptyset), \{(a), \emptyset, \emptyset\}\} \). On the other hand \( F_1 \) and \( F_1' \) differ in their complete extensions since \( \text{co}(F_1) = \emptyset \) but \( \text{co}(F_1') = \{(a)\} \).

Therefore complete semantics is unverifiable by the verification class induced by \( \tau^+ \). Likewise, this can be shown for the classes induced by \( \tau^-, \tau^\pm, \tau^-\), \( \tau^\mp \), and \( \tau^{+-} \), respectively:

\[
F_2: a \\
F_2': a \\
F_3: a \\
F_3': a \\
F_4: a \\
F_4': a \\
F_5: a \\
F_5': a \\
F_6: a \\
F_6': a \\
F_7: a \\
F_7': a \\
F_8: a \\
F_8': a
\]

Hence the complete semantics is exactly verifiable by the verification class induced by \( \tau^\mp \).

**Example 5.** Consider the semi-stable and eager semantics and recall that they are \( +\mp \)-verifiable. In order to show exact verifiability it suffices to show unverifiability by the classes induced by \( \tau^+, \tau^- \), and \( \tau^\mp \) (cf. Figure 1); \( F_1 \) and \( F_6 \) are taken from Example 4 above:

\[
F_1: a, b \\
F_1': a \\
F_6: a, b \\
F_6': a \\
F_7: a, b, c \\
F_7': a, b, c
\]

Hence, both the semi-stable and eager semantics are exactly verifiable by the verification class induced by \( \tau^\mp \).
Example 6. Now consider the grounded and strong admissibility semantics and recall that they are \( \tau^{+} \)-verifiable. In order to show exact verifiability we have to show unverifiability by the classes induced by \( r^{+}, r^{-} \), and \( r^{\omega} \) (cf. Figure 1); again, the AFs from Example 4 can be reused.

- \( F_{1}^{\pm} = \overline{F_{1}}^{\pm} \), but \( gr(F_{1}) = \emptyset \neq \{a\} = gr(F_{1}^{'}) \) and \( sad(F_{1}) = \emptyset \neq \emptyset, \{a\} = sad(F_{1}) \).
- \( F_{2}^{\pm} = \overline{F_{2}}^{\pm} \), but \( gr(F_{2}) = \{a\} \neq \emptyset, \{a, c\} = gr(F_{2}) \) and \( sad(F_{2}) = \emptyset, \{a\}, \{a, c\} = sad(F_{2}) \).
- \( F_{6}^{\pm} = \overline{F_{6}}^{\pm} \), but \( gr(F_{6}) = \{a\} \neq \emptyset = gr(F_{6}) \) and \( sad(F_{6}) = \emptyset, \{a\} \neq \emptyset = sad(F_{6}) \).

Hence, both the grounded and strong admissible semantics are exactly verifiable by the verification class induced by \( \tau^{+} \).

Example 7. Finally consider stable, stage, admissible, preferred and ideal semantics. They are either \( +r \)-verifiable (\( stb \) and \( stg \)) or \( -r \)-verifiable (\( ad, pr \), and \( id \)). In order to show that these verification classes are exact we have to show unverifiability w.r.t. the verification class induced by \( r^{+} \). Consider, for instance, the AFs \( F_{4}^{1} \) and \( F_{4}^{2} \) from Example 4. We have \( \overline{F_{4}}^{1} = \overline{F_{4}}^{2} \), but \( ad(F_{4}) = \emptyset \neq \{a\} = ad(F_{4}^{'}) \), \( stb(F_{4}) = \{a\} \neq \emptyset = stb(F_{4}^{'}) \), and \( sad(F_{4}) = \{\{a\}\} \neq \emptyset = sad(F_{4}^{'}) \) for \( \sigma \in \{stb, pr, id\} \), showing exactness of the respective verification classes.

The insights obtained through Examples 4, 5, 6, and 7 show that the verification classes obtained from the criteria given above are indeed exact. Figure 2 shows the relation between the semantics under consideration with respect to their exact verification classes.

We turn now to the main theorem stating that any rational semantics (recall that all semantics we consider in this paper are rational) is exactly verifiable by one of the 15 different verification classes.

**Theorem 4.** Every semantics which is rational is exactly verifiable by a verification class induced by one of the neighborhood functions presented in Figure 1.

**Proof.** First of all note that by Lemma 2, \( r^{\omega} \) is the least informative neighborhood function and for every other neighborhood function \( r^{i} \) it holds that \( r^{i} \preceq r^{\omega} \). Therefore, if a semantics is verifiable by the verification class induced by any \( r^{i} \) then it is exactly verifiable by a verification class induced by some \( r^{\omega} \) with \( r^{i} \preceq r^{\omega} \preceq r^{\omega} \). Moreover, if a semantics is exactly verifiable by a class, then it is by definition also verifiable by this class. Hence it remains to show that every semantics which is rational is verifiable by a verification class presented in Figure 1.

We show the contrapositive, i.e., if a semantics is not verifiable by a verification class induced by one of the neighborhood functions presented in Figure 1 then it is not rational.

Assume a semantics \( \sigma \) is not verifiable by one of the verification classes. This means \( \sigma \) is not verifiable by the verification class induced by \( r^{+} \). Hence there exist two AFs \( F \) and \( G \) such that \( F^{+} = \overline{G^{+}} \) and \( A(F) = A(G) \), but \( \sigma(F) \neq \sigma(G) \). For every argument \( a \) which is not self-attacking, a tuple \( \{\{a\}\} \) is contained in \( F^{+} \) (and in \( G^{+} \)). Hence \( F \) and \( G \) have the same not-self-attacking arguments and, moreover these arguments have the same ingoing and outgoing attacks in \( F \) and \( G \). This, together with \( A(F) = A(G) \) implies that \( F^{+} = G^{+} \) (see Definition 2) holds. But since \( \sigma(F) \neq \sigma(G) \) we get that \( \sigma \) is not rational, which was to show.

Note that the criterion gives evidence for verifiability of a semantics by a certain class has access to the set of arguments of a given framework. In fact, only the criterion for stable semantics makes use of that. Indeed, stable semantics needs this information since it is not verifiable by any class when using a weaker notion of verifiability, which rules out the usage of \( A(F) \).

**Intermediate Semantics**

A type of semantics which has aroused quite some interest in the literature (see e.g. (Baroni and Giacomin 2007a) and (Nieves, Osorio, and Zepeda 2011)) are intermediate semantics, i.e. semantics which yield results lying between two existing semantics. The introduction of \( \sigma^{-} \)-intermediate semantics can be motivated by deleting undesired (or add desired) \( \tau \)-extensions while guaranteeing all reasonable positions w.r.t. \( \sigma \). In other words, \( \sigma^{-} \)-intermediate semantics can be seen as sceptical or credulous acceptance shifts within the range of \( \sigma \) and \( \tau \).

A natural question is whether we can make any statements about compatible kernels of intermediate semantics. In particular, if semantics \( \sigma \) and \( \tau \) are compatible with some kernel \( k \), is then every \( \sigma^{-} \)-intermediate semantics \( k \)-compatible. The following example answers this question negatively.

**Example 8.** Recall from Theorem 1 that both stable and stage semantics are compatible with \( k(stb) \), i.e. \( F = k(stb) G \Rightarrow F = k(stb) G \Rightarrow F^{k(ab)} = G^{k(ab)} \). Now we define the following \( stb-stg \)-intermediate semantics, say stage semantics: Given an AF \( F = (A, R), S \in sta(F) \) iff \( S \in cf(F) \), \( S^{+}_{x} \cup S^{-}_{x} = A \) and for every \( T \in cf(F) \) we have \( S^{+}_{x} \subseteq T^{+}_{x} \). Obviously, it holds that \( stb \subseteq sta \subseteq stg \) and \( stb \neq sta \) as well as \( sta \neq stg \), as witnessed by the following AF \( F \):

\[
\begin{align*}
F : \quad & a \\
& b \\
& c
\end{align*}
\]

It is easy to verify that \( stb(F) = \emptyset \subseteq sta(F) = \{b\} \subseteq stg(F) = \{\{b\}, \{c\}\} \). We proceed by showing that stage semantics is not compatible with \( k(stb) \). To this end consider \( F^{k(ab)} \), which is depicted below.

![Diagram](https://example.com/diagram.png)
Now, \( sta \{ F^{k(ab)} \} = \{ \{ b \}, \{ c \} \} \) witnesses \( F \not\equiv_{stb} F^{k(ab)} \) and therefore, \( F \not\equiv_{stb} F^{k(ab)} \). Since \( F^{k(ab)} = \{ F^{k(ab)} \}^{k(ab)} \), we are done, i.e. stagle semantics is indeed not compatible with the stable kernel.

It is the main result of this section that compatibility of intermediate semantics w.r.t. a certain kernel can be guaranteed if verifiability w.r.t. a certain class is presumed. The provided characterization theorems generalize former results presented in (Oikarinen and Woltran 2011). Moreover, due to the abstract character of the theorems the results are applicable to semantics which may be defined in the future.

Before turning to the characterization theorems we state some implications of verifiability. In particular, under the assumption that \( \sigma \) is verifiable by a certain class, equality of certain kernels implies expansion equivalence w.r.t. \( \sigma \).

**Proposition 4.** For any +\-verifiable semantics \( \sigma \) we have

\[
F^{k(ab)} = G^{k(ab)} \Rightarrow F \equiv_{G}^{\sigma}.
\]

**Proof.** In (Oikarinen and Woltran 2011) it was shown that \( F^{k(ab)} = G^{k(ab)} \Rightarrow (F \cup H)^{k(ab)} = (G \cup H)^{k(ab)} \) (i).

Consider now a +\-verifiable semantics \( \sigma \). In order to show \( \sigma(F) = \sigma(F^{k(ab)}) \) (ii) we prove \( F^{k(ab)} = F^{k(ab)} \) (first).

It is easy to see that \( S \in cf(F) \) iff \( S \in cf(F^{k(ab)}) \).

Furthermore, since \( k(stb) \) deletes an attack \((a, b)\) only if \( a \) is self-defeating we deduce that ranges does not change as long as conflict-free sets are considered. Thus, \( \sigma(F) = (\sigma(F^{k(ab)}))^{+} \).

Now assume that \( F^{k(ab)} = F^{k(ab)} \) \( \Leftrightarrow \) \( \gamma_{\sigma}(F^{k(ab)}) \) \( \Leftrightarrow \) \( \gamma_{\sigma}(F^{k(ab)}) \).

Applying (ii) we obtain \( S \in \sigma(F \cup H) \) for some AF \( H \). We have to show that \( S \in \sigma(G \cup H) \).

Applying (i) we deduce \( S \in \sigma(G \cup H) \).

Furthermore, using (i) we deduce \( S \in \sigma(G \cup H) \).

Finally, \( S \in \sigma(G \cup H) \) by applying (ii), which concludes the proof.

The following results can be shown in a similar manner.

**Proposition 5.** For any +\+\-verifiable semantics \( \sigma \) we have

\[
F^{k(ad)} = G^{k(ad)} \Rightarrow F \equiv_{G}^{\sigma}.
\]

**Proposition 6.** For any +\-\-verifiable semantics \( \sigma \) we have

\[
F^{k(co)} = G^{k(co)} \Rightarrow F \equiv_{G}^{\sigma}.
\]

**Proposition 7.** For any +\-\+\-verifiable semantics \( \sigma \) we have

\[
F^{k(gr)} = G^{k(gr)} \Rightarrow F \equiv_{G}^{\sigma}.
\]

**Proposition 8.** For any +\-\+\-verifiable semantics \( \sigma \) we have

\[
F^{k(ua)} = G^{k(ua)} \Rightarrow F \equiv_{G}^{\sigma}.
\]

We proceed with general characterization theorems. The first one states that stb-stg-Intermediate semantics are compatible with stable kernel if +\-verifiability is given. Consequently, stagle semantics as defined in Example 8 can not be +\-verifiable.

**Theorem 5.** Given a semantics \( \sigma \) which is +\-verifiable and stb-stg-intermediate, it holds that

\[
F^{k(ab)} = G^{k(ab)} \Leftrightarrow F \equiv_{G}^{\sigma}.
\]

**Proof.** \((\Rightarrow)\) Follows directly from Proposition 4.

\((\Leftarrow)\) We show the contrapositive, i.e. \( F^{k(ab)} \neq G^{k(ab)} \Rightarrow F \not\equiv_{G}^{\sigma} \). Assuming \( F^{k(ab)} \neq G^{k(ab)} \) implies \( F \not\equiv_{G}^{\sigma} \), i.e. there exists an AF \( H \) such that \( stg(F \cup H) \neq stg(G \cup H) \) and therefore, \( stb(F \cup H) \neq stb(G \cup H) \). Let \( B = A(F \cup H) \cup A(G) \cup A(H) \) and \( H' = (B \cup \{ a \}, \{ (a, b), (b, a) \mid b \in B \}) \).

It is easy to see that \( stb(F \cup H') = stb(F \cup H) \cup \{ \{ a \} \} \) and \( stb(G \cup H') = stb(G \cup H) \cup \{ \{ a \} \} \). Since now both \( stb(F \cup H') \neq \emptyset \) and \( stb(G \cup H') \neq \emptyset \) it holds that \( stb(F \cup H') = stg(F \cup H') \) and \( stb(G \cup H') = stg(G \cup H') \).

Hence \( \sigma(F \cup H') \neq \sigma(F \cup H) \), showing that \( F \not\equiv_{G}^{\sigma} \).

The following theorems can be shown in a similar manner.

**Theorem 6.** Given a semantics \( \sigma \) which is +\+\-verifiable and ad-intermediate with \( p \in \{ \{ s, \ id, eq \} \), it holds that

\[
F^{k(ad)} = G^{k(ad)} \Leftrightarrow F \equiv_{G}^{\sigma}.
\]

Remember that complete semantics is a ss-ad-intermediate semantics. Furthermore, it is not characterizable by the admissible kernel as already observed in (Oikarinen and Woltran 2011). Consequently, complete semantics is not +\+\-verifiable (as we have shown in Example 4 with considerable effort).

**Theorem 7.** Given a semantics \( \sigma \) which is -\+\-verifiable and gr-sad-intermediate, it holds that

\[
F^{k(gr)} = G^{k(gr)} \Leftrightarrow F \equiv_{G}^{\sigma}.
\]

**Conclusions**

In this work we have contributed to the analysis and comparison of abstract argumentation semantics. The main idea of our approach is to provide a novel categorization in terms of the amount of information required for testing whether a set of arguments is an extension of a certain semantics. The resulting notion of verifiability classes allows us to categorize any new semantics (given it is “rational”) with respect to the information needed and compare it to other semantics. Thus our work is in the tradition of the principle-based evaluation due to Baroni and Giacomin (2007b) and paves the way for a more general view on argumentation semantics, their common features, and their inherent differences.

Using our notion of verifiability, we were able to show kernel-compatibility for certain intermediate semantics. Concerning concrete semantics, our results yield the following observation: While preferred, semi-stable, ideal and eager semantics coincide w.r.t. strong equivalence, verifiability of these semantics differs. In fact, preferred and ideal semantics manage to be verifiable with strictly less information.

For future work we envisage an extension of the notion of verifiability classes in order to categorize semantics not captured by the approach followed in this paper, such as cf2 (Baroni, Giacomin, and Guida 2005).
References


Preferential Modalities Revisited

Katarina Britz
CSIR-SU Centre for AI Research
Stellenbosch University, South Africa
abritz@sun.ac.za

Ivan Varzinczak
Centre de Recherche en Informatique de Lens
Université d’Artois, France
varzinczak@cril.fr

Abstract

We venture beyond the customary semantic approach in NMR, namely that of placing orderings on worlds (or valuations). In a modal-logic setting, we motivate and investigate the idea of ordering elements of the accessibility relations in Kripke frames, i.e., world pairs \((w, w')\) (or ‘arrows’). The underlying intuition is that some world pairs may be seen as more normal (or typical, or expected) than others. We show this delivers an elegant and intuitive semantic construction, which gives a new perspective on present notions of defeasible necessity. From a modeler’s perspective, the new framework we propose is more intuitively appealing. Technically, though, the revisited logic happens to not substantively increase the expressive power of the previously defined preferential modalities. This conclusion follows from an analysis of both semantic constructions via a generalisation of bisimulations to the preferential case. Let this be seen as a negative result, it essentially means that reasoners based on the previous semantics (which have been shown to preserve the computational complexity of the underlying classical modal language) suffice for reasoning over the new semantics. Finally, we show that the kind of construction we here propose has many fruitful applications, notably in a description-logic context, where it provides the foundations on which to ground useful notions of defeasibility in ontologies yet to be explored.

Introduction

Accounts of normality (or typicality), plausibility and alike traditionally have an underlying semantics built on a notion of preference on worlds. Such is the case of non-monotonic entailment (Shoham 1988; Kraus, Lehmann, and Magidor 1990; Makinson 2005), conditionals (Lehmann and Magidor 1992; Boutilier 1994), belief revision (Katsuno and Mendelzon 1991; Baltag and Smets 2006; 2008), counterfactuals (Stalnaker 1968; Lewis 1973; 1974), obligations (Hanson 1969) and many others, as known from the literature on non-monotonic reasoning, conditional and deontic logics, and related areas. Roughly speaking, the usual approach consists in selecting some worlds (or propositional valuations) as being more normal (alias typical, alias desirable) and carrying out the reasoning relative to an underlying normality ordering on worlds.

A typical representative of these different yet interrelated threads of investigation is the well-known preferential approach (Shoham 1988) and its derivatives (Kraus, Lehmann, and Magidor 1990; Lehmann and Magidor 1992). There, a preference relation is defined on the set of possible worlds with the (tacit) assumption that these contain all is needed to reason about what is normal or expected. A case can indeed be made for such an assumption in a propositional setting. However, in logics with more structure, it is reasonable to say that the normality ‘spotlight’ should not be confined to worlds, but rather (also) be put on (possibly) whatever structure one has at one’s disposal in the respective underlying semantics. To witness, in a modal logic context, it makes sense to ask whether some links between worlds in a frame are (relatively) more normal (or preferred) than others—irrespective of whether the worlds involved are by any means comparable in that way or another amongst themselves. In other words, one can be interested in the normality of the transition from one world to another one. This point is better illustrated with some well-known concrete applications of modal logics as given below.

Let us assume a very simplistic scenario in which we have only one propositional atom, on, of which the intuition is that a particular light-bulb is on. Moreover, let us assume there is only one action at one’s disposal, namely toggle (hereafter abbreviated t), of which the intuition is that of changing the state of the light switch. Figure 1 below depicts a possible-worlds model for this scenario.

![Figure 1: A possible-worlds model for one action (toggle) and one atom (on).](image)

Intuitively, normal (or typical, or expected) executions of the toggle action are given by the t-transitions from \(w_1\) to \(w_2\) and back, whereas the reflexive arrows are, in a sense, less expected in the given scenario. Therefore, it becomes important to single out those executions of the action that are deemed more normal from those that are not. In Figure 1,
this would amount to enriching the semantic structure (in a way still to be defined) with information specifying that the pairs \((w_1, w_2)\) and \((w_2, w_1)\) somehow take ‘precedence’ over \((w_1, w_1)\) and \((w_2, w_2)\) when reasoning about possible executions of the action.

Let us now consider a variant of the above scenario (although just as simple), in which we have only one atomic proposition, correct (which we shall abbreviate as \(c\)), the intuition of which is that a proposed proof for a mathematical statement is correct. Furthermore, let us assume there is a good mathematician, \(M\), whose knowledge about the correctness of the proof is of interest to us. Figure 2 below depicts one possible configuration of this scenario.

![Figure 2: A possible-worlds model for one agent (the good mathematician \(M\)) and one atom (correct).](image)

As a good mathematician, Agent \(M\) should know whether the proof is correct. Nevertheless, in the model of Figure 2, \(M\) does not know (in the classical sense) whether the proof is correct or not, since, also by virtue of being a good mathematician, \(M\) admits the (unlikely) possibility of being wrong (at least until the proof has been submitted to peer-reviewed scrutiny). In this case, we would say that Agent \(M\) defeasibly knows whether the proof is correct or not, an epistemic stance that can be adopted by ‘focusing’ on the most normal (or expected) of the epistemic possibilities held by the agent, namely \((w_1, w_1)\) and \((w_2, w_2)\) in Figure 2, which, in this example, are more normal than \((w_1, w_2)\) and \((w_2, w_1)\). (It is not hard to see that the motivation above also holds in a doxastic context, as certain beliefs may be more entrenched than others.)

In order to motivate the foregoing ideas in a deontic context, let us assume a language with a single propositional atom, namely fair-play, henceforth abbreviated \(f\) and of which the intuition is that, in a competition, the players abide by an established standard of ‘decency’ or an ‘honorable conduct’. In this context, adopting a fair-play stance is not to be seen as an obligation in the usual (strict) meaning of the term. It is rather a matter of best practice in that it corresponds to the expected, though not enforceable (even if, in some cases, liability-biding), attitude. Figure 3 below depicts a possible-worlds model for this scenario.

![Figure 3: A possible-worlds model for one atom (fair-play) in a deontic-context.](image)

A case can be made that envisioning an \(f\)-world as a better alternative (to the current one) is more appropriate than the contemplation of a \(! f\)-one. Semantically, this requirement would be translated as setting the pairs \((w_1, w_1)\) and \((w_2, w_1)\) as more preferred than \((w_2, w_2)\) and \((w_1, w_2)\). In this specific example, it happens that we could also model the underlying preference as an ordering on worlds, with \(f\)-worlds preferred to the \(! f\)-worlds. However, in the preceding examples, ordering worlds rather than pairs of worlds is neither intuitive nor is it immediately clear whether this is even possible.

In this work, we address precisely these issues. We shall start by shifting the normality spotlight from possible worlds to transitions amongst them, i.e., to accessibility relations in Kripke frames. The justification for doing so stems from a comparison with the classical (monotonic) case: In classical Kripkean semantics, modalities are primarily about accessibility, only secondarily about worlds’ contents. Hence, we contend that accounts of a notion of defeasibility in modalities (like those illustrated above) should primarily focus on normality of the accessibility relations rather than (or at least prior to) that of the (accessible) worlds. With that we hope to pave the way for further explorations of non-monotonicity in modal logics, in particular in extensions of the preferential approach therein (Britz, Meyer, and Varzinczak 2011a).

**Preliminaries**

In this section, we provide the required formal background for the rest of this work. In particular, we set up the notation and conventions that shall be followed in the upcoming sections. (The reader conversant with modal logic can safely skip the first subsection below.)

**Modal Logic**

We work in a set of atomic propositions \(P\), using the logical connectives \(\land\) (conjunction), \(\lnot\) (negation), and a set of modal operators \(\Box\), \(1 \leq i \leq n\). Propositions are denoted by \(p, q, \ldots\), and sentences by \(\alpha, \beta, \ldots\) constructed in the usual way according to the rule \((1 \leq i \leq n)\):

\[
\alpha ::= p \mid \lnot \alpha \mid \alpha \land \alpha \mid \Box_i \alpha
\]

All the other truth-functional connectives (\(\lor, \rightarrow, \leftrightarrow, \ldots\)) are defined in terms of \(\lnot\) and \(\land\) in the usual way. Given \(\Box\), \(1 \leq i \leq n\), with \(\circ_i\) we denote its dual modal operator, i.e., for any \(\alpha, \circ_i \alpha ::= \lnot \Box_i \lnot \alpha\). We use \(T\) as an abbreviation for \(p \lor \lnot p\) and \(\bot\) as an abbreviation for \(p \land \lnot p\), for some \(p \in P\). With \(L^n\) we denote the set of all sentences of the modal language.

The semantics is the standard possible-worlds one:

**Definition (Kripke Model)** A Kripke model is a tuple \(M := \langle W, R, V \rangle\) where \(W\) is a (non-empty) set of possible worlds, \(R := \langle R_1, \ldots, R_n \rangle\), where each \(R_i \subseteq W \times W\) is an accessibility relation on \(W\), \(1 \leq i \leq n\), and \(V : W \rightarrow \{0, 1\}^P\) is a valuation function mapping possible worlds into propositional valuations.
As an example, Figure 4 depicts the Kripke model $M_1 = \langle W_1, R_1, V_1 \rangle$, where $W_1 := \{w_i \mid 1 \leq i \leq 4\}$, $R_1 := \langle R_a, R_b \rangle$, with $R_a := \{(w_1, w_2), (w_1, w_3), (w_4, w_3)\}$, and $R_b := \{(w_1, w_2), (w_2, w_3)\}$, and $V_1$ is the obvious valuation function.

In our pictorial representations of models, we represent propositional valuations as sequences of $0$s and $1$s, and with the obvious implicit ordering of atoms. Thus, for the logic generated from $p$ and $q$, the valuation in which $p$ is true and $q$ is false will be represented as $10$.

We shall use $w, u, v, \ldots$ (possibly decorated with primes) to denote possible worlds. Moreover, where it aids readability, we shall henceforth sometimes write tuples of the form $(w, w')$ as $ww'$.

Sentences of $L^2$ are true or false relative to a possible world in a given Kripke model.

Definition 2 (Truth Conditions) Let $M = \langle W, R, V \rangle$ and $w \in W$:

- $M, w \models p$ if and only if $V(w)(p) = 1$;
- $M, w \models \neg \alpha$ if and only if $M, w \models \alpha$;
- $M, w \models \alpha \land \beta$ if and only if $M, w \models \alpha$ and $M, w \models \beta$;
- $M, w \models \Box \alpha$ if and only if $M, w' \models \alpha$ for all $w'$ such that $(w, w') \in R$.

Given $\alpha \in L^2$ and $M = \langle W, R, V \rangle$, we say that $M$ satisfies $\alpha$ if there is at least one world $w \in W$ such that $M, w \models \alpha$. We say that $M$ is a model of $\alpha$ (alias $\alpha$ is true in $M$), denoted $M \models \alpha$, if $M, w \models \alpha$ for every world $w \in W$. Given a class (i.e., a collection) of models $M$, we say that $\alpha$ is valid in $M$, denoted $\models_M \alpha$, if and only if every Kripke model $M \in M$ is a model of $\alpha$. Given $K \subseteq L^2$ and $\alpha \in L^2$, we say that $K$ locally entails $\alpha$ in the class of models $M$, denoted $K \models_M \alpha$, if and only if for every Kripke model $M \in M$ and every $w \in M$, if $M, w \models \beta$ for every $\beta \in K$, then $M, w \models \alpha$. (When the class of models we are working with is clear from the context, we shall dispense with subscripts and just write $\models \alpha$ and $K \models \alpha$.)

Here we shall assume the system of normal modal logic $K$, of which all the other normal modal logics are extensions. Semantically, $K$ is characterised by the class of all Kripke models (Definition 1). Syntactically, $K$ corresponds to the smallest set of sentences containing all propositional tautologies, all instances of the axiom schema $K : \Box \alpha \rightarrow (\Box \alpha \rightarrow \Box \beta)$, $1 \leq i \leq n$, and closed under the rule of necessitation below:

\[
\text{(RN) } \alpha \rightarrow \Box \alpha
\]

For more details on modal logic, we refer the reader to the handbook by Blackburn et al. (2006).

**Preferential Modalities**

In previous work (Britz, Meyer, and Varzinczak 2011a; Britz and Varzinczak 2013), we have investigated the fruitfulness of extending the standard Kripke semantics with a preference relation on the set of possible worlds. This gives rise to the following semantic structure, of which the underlying motivation is similar to that behind Boutilier’s (1994) CT40 models and the plausibility models of Baltag and Smets (2006; 2008).

**Definition 3 (W-Ordered Model)** A W-ordered model is a tuple $W := \langle W, R, V \rangle$ where $W$ is as in Definition 1 and $\mathcal{W} \subseteq W \times W$ is a well-founded strict partial order on $w$, i.e., $\mathcal{W}$ is irreflexive, transitive and every non-empty $W' \subseteq W$ has minimal elements w.r.t. $\mathcal{W}$ (see Definition 4).

The intuition behind the preference relation $\prec$ in a W-ordered model $W$ is that the worlds lower down in the ordering are deemed more preferred (or more normal) than those higher up.

**Definition 4 (Minimality w.r.t. $\prec$)** Let $W = \langle W, \mathcal{W}, \prec \rangle$ be a W-ordered model and let $X \subseteq W$. Then $\min \prec X := \{w \in X \mid w \prec w' \text{ for no } w' \in X\}$, i.e., $\min \prec X$ denotes the minimal elements of $X$ with respect to the preference relation $\prec$.

As an example, Figure 5 below depicts the W-ordered model $W_1 = \langle W_1, R_1, V_1, \prec_1 \rangle$, where $\langle W_1, R_1, V_1 \rangle$ is as in Figure 4 and $\prec_1 := \{(w_1, w_2), (w_2, w_3), (w_1, w_3), (w_4, w_3)\}$.

![Figure 4: A Kripke model for $P = \{p, q\}$ and two modalities, namely $a$ and $b$.](image1)

![Figure 5: A W-ordered model for $P = \{p, q\}$ and two modalities ($a$ and $b$). The preference relation $\prec_1$ is represented by the dashed arrows, which point from more preferred to less preferred worlds.](image2)
the extended language are then recursively defined by:
\[
\alpha ::= p \mid \neg \alpha \mid \alpha \wedge \alpha \mid \Box_i \alpha \mid \Diamond_i \alpha
\]
As before, the other connectives are defined in terms of \(\neg\) and \(\wedge\) in the usual way, \(\top\) and \(\bot\) are seen as abbreviations, and \(\Box_i\) is the dual of \(\Diamond_i\). Moreover, with \(\Box_i\) (called ‘flame’) we denote the dual of \(\Diamond_i\). We shall use \(\mathcal{L}^\cdot\) to denote the set of all sentences of such a richer language.

**Definition 5 (Truth Conditions for \(\mathcal{L}^\cdot\))** Let a \(W\)-ordered model \(\mathcal{W} = \langle W, R, V, \mathcal{L} \rangle\) and let \(w \in W\).

- \(\mathcal{L}^\cdot\)-sentences are evaluated as usual (Definition 2);
- \(\mathcal{W}, w \models \Box_i \alpha\) if and only if for all \(w'\), if \(w' \in \min_R(w)\), then \(\mathcal{W}, w' \models \alpha\).

The notions of satisfaction, truth (in a model), and local entailment are generalised to \(\mathcal{L}^\cdot\)-sentences and \(W\)-ordered models in the obvious way.

Informally, a sentence of the form \(\Box_i \alpha\) holds in a world if \(\alpha\) holds in all the most preferred amongst its \(i\)-accessible worlds. It is easy to see that \(\Box\) is weaker than \(\Box\), i.e., the following is a validity (\(1 \leq i \leq n\)):

\[
\models \Box_i \alpha \rightarrow \Box_i \alpha
\]
Hence, intuitively, flag can be read as **defeasible necessity**.

As an example, considering the \(W\)-ordered model \(\mathcal{W}_1\) from Figure 5, we have that \(\mathcal{W}_1, w_1 \models \Box_\alpha \neg p\) (but note that \(\mathcal{W}_1, w_1 \not\models \Box_\neg p\)).

**Revisiting Preferential Modal Logics**
In spite of its gain in expressiveness when checked against traditional approaches to defeasible reasoning, \(\Box\) does not quite seem to allow us to formalise the type of reasoning motivated in the Introduction inasmuch as it relies on orderings on worlds. In this section, we shall revisit the framework for preferential modalities, in particular its semantic constructions.

**R-Ordered Models**
We start by giving a formal account of the semantic ideas put forward in the Introduction.

**Definition 6 (R-Ordered Model)** An R-ordered model is a tuple \(\mathcal{R} := \langle W, R, V, \mathcal{L} \rangle\) where \(W\) is a (non-empty and possibly infinite) set of possible worlds, \(R := \langle R_1, \ldots, R_n \rangle\), where each \(R_i \subseteq W \times W\) is an accessibility relation on \(W\), for \(1 \leq i \leq n\), \(V : W \rightarrow \{0, 1\}\) is a valuation function assigning each world to a valuation on \(P\), and \(\mathcal{L} := \langle \Box_1, \ldots, \Box_n \rangle\), where each \(\Box_i \subseteq R_i \times R_i\) for \(1 \leq i \leq n\), is a well-founded strict partial order on the respective \(R_i\), i.e., each \(\Box_i\) is a well-founded strict partial order on \(R_i\), i.e., each \(\Box_i\) is irreducible, transitive, and every non-empty \(R_i \setminus \Box_i\) has minimal elements w.r.t. \(\Box_i\) (see Definition 7).

Given \(\mathcal{R} := \langle W, R, V, \mathcal{L} \rangle\), the intuition of \(W\), \(R\), and \(V\) is the same as that in a standard Kripke model. The intuition of each \(\Box_i\) in \(\mathcal{L}\) is that the pairs \((w, w')\) that are lower down in the ordering \(\Box\) are deemed as the most normal (or typical, or expected) in the context of \(R_i\).

**Definition 7 (Minimality w.r.t. \(\Box_i\))** Let \(\mathcal{R} = \langle W, R, V, \mathcal{L} \rangle\) be an R-ordered model and let \(X \subseteq W\), for some \(1 \leq i \leq n\). Then \(\min_{\Box_i} X := \{(w, w') \in X \mid \text{there is no } (u, v) \in X\) such that \((u, v) \models (w, w')\}, i.e., \(\min_{\Box_i} X\) denotes the minimal elements of \(X\) with respect to the preference relation \(\Box_i\) associated to \(R_i\).

Since we assume each \(\Box_i\) to be a well-founded strict partial order on the respective \(R_i\), we are guaranteed that for every \(X \subseteq W\) such that \(X \neq \emptyset\), \(\min_{\Box_i} X\) is well defined.

As an example, Figure 6 below depicts the R-ordered model \(\mathcal{R}_1 := \langle W_1, R_1, V_1, \mathcal{L}_1 \rangle\), where \(\mathcal{R}_1, V_1\) is as in Figure 4, and \(\mathcal{L}_1 := \langle \Box_{a_1}, \Box_{a_2} \rangle\), where \(\Box_{a_1} := \{(w_1 w_2, w_1 w_3), (w_1 w_3, w_4 w_3), (w_1 w_2, w_4 w_3)\}\) and \(\Box_{a_2} := \{(w_1 w_4, w_2 w_3)\}\), respectively, by the dashed and the dotted arcs in the picture. (Note the direction of the \(\Box\)-arrows, which point from more preferred to less preferred transitions.) For the sake of readability, in our pictorial representations of R-ordered models, we shall omit the transitive \(\Box\)-arrows.

**A New Logic of Defeasible Modalities**
We shall now enrich our underlying modal language with a family of additional modal operators \(\Box_i\), \(1 \leq i \leq n\), where \(n\) is the number of classical modalities in the language. (For lack of a better term, we shall call \(\Box\) the ‘banner’.) The sentences of the extended modal language are recursively defined as follows:

\[
\alpha ::= p \mid \neg \alpha \mid \alpha \wedge \alpha \mid \Box_i \alpha \mid \Diamond_i \alpha
\]

With \(\mathcal{L}^\cdot\) we shall denote the set of all sentences of the banner language.

**Definition 8** Let \(\mathcal{R} = \langle W, R, V, \mathcal{L} \rangle\). For every \(w \in W\) and every \(R_i \subseteq W \times W\), we define:

\[
R_i^w := \{(u, v) \mid (u, v) \in R_i\text{ and }u = w\}
\]

**Definition 9 (\(\mathcal{L}^\cdot\)-Truth Conditions)** Let \(\mathcal{R} = \langle W, R, V, \mathcal{L} \rangle\) be an R-ordered model and \(w \in W\).

- \(\mathcal{L}^\cdot\)-sentences are evaluated as usual;
- \(\mathcal{R}, w \models \Box \alpha\) if and only if for every \(w'\), if \(\langle w, w' \rangle \in \min_{\Box_i} R_i^w\) then \(\mathcal{R}, w' \models \alpha\).
The notions of satisfaction, truth (in a model), validity (in a class of models) and local entailment are also generalised to \(L^{\alpha}\)-sentences and \(R\)-ordered models in the usual way.

Informally, a sentence of the form \(\Xi_i \alpha\) holds in a world if \(\alpha\) holds in all its most normally \(i\)-accessible worlds. As an example, in the \(R\)-ordered model \(\mathcal{R}_1\) of Figure 6, we have that \(\mathcal{R}_1, w_1 \models \Xi_a \neg p\) (but, of course, \(\mathcal{R}_1, w_1 \not\models \Xi_a p\)).

Incidentally, \(\Xi\) too is weaker than \(\square\), as witnessed by the validity below (\(1 \leq i \leq n\)):

\[
\models \square_i \alpha \rightarrow \Xi_i \alpha
\]

Hence, \(\Xi\) provides an alternative perspective on the notion of defeasible necessity as formalised by \(\square\). For instance, in an action context, some executions (which refer to transitions) of a given action are deemed as more normal than others. A priori, this is different from saying that some effects (which refer to target worlds) are normal. Indeed, an abnormal execution may still lead to the expected (normal) effect, just as a normal execution may produce an abnormal effect. (We shall come back to this issue later on.)

The definitions of \(R\)-ordered models and \(\Xi\), alongside the comment right above, raise the question as to how \(L^{\alpha}\) and \(L^{\alpha}\) compare to each other in terms of expressive power. This is what we address in the next section.

### Preferential Bisimulations

Standard bisimulations are used to determine whether two Kripke models have the same modal properties, and to reason about modal expressivity. Here, we extend the definition of bisimulations to \(W\)-ordered and \(R\)-ordered models, and use it to make precise the connection between these notions, and the resulting modalities and modal languages.

**Definition 10** Let \(\mathcal{M} = \langle W, R, V \rangle\) and \(\mathcal{M}' = \langle W', R', V' \rangle\). A bisimulation between \(\mathcal{M}\) and \(\mathcal{M}'\) is a non-empty binary relation \(E\) between their domains (that is, \(E \subseteq W \times W'\)) such that, whenever \(wEw'\),

1. For every \(p \in \mathcal{P}\), \(w \models p\) if and only if \(w' \models p\);
2. if \(wRvw'\), then there exists a world \(v'\) in \(W'\) such that \(v' \models \forall v'\) and \(w'R'v'\), and
3. if \(w'R'v'\), then there exists a world \(v\) in \(W\) such that \(v \models \forall v'\) and \(wRv\).

Informally, two worlds are bisimilar if they satisfy the same atomic information, and their modal accessibility structures match. Two pointed models \((\mathcal{M}, w)\) and \((\mathcal{M}', w')\) are bisimilar if there exists a bisimulation \(E\) between \(\mathcal{M}\) and \(\mathcal{M}'\) such that \(wEw'\). It then follows that:

**Lemma 1** (Bisimulation invariance lemma) If \(E\) is a bisimulation between \(\mathcal{M} = \langle W, R, V \rangle\) and \(\mathcal{M}' = \langle W', R', V' \rangle\), \(w \in W\) and \(w' \in W'\), and \(wEw'\), then \(w\) and \(w'\) satisfy the same basic modal sentences.

The next definition and lemma generalise bisimulations to take account of a preference order on worlds, as defined on models of \(L^{\alpha}\). Informally, two worlds are bisimilar if they satisfy the same atomic information and their modal accessibility structures match, both with respect to accessible worlds and with respect to most preferred relative accessible worlds. Bisimilar worlds then also satisfy the same preferential modal sentences.

**Definition 11** (\(W\)-ordered bisimulation) Let \(W\)-ordered models \(\mathcal{W} = \langle W, R, V, \prec \rangle\) and \(\mathcal{W}' = \langle W', R', V', \prec' \rangle\). A bisimulation between \(\mathcal{W}\) and \(\mathcal{W}'\) is a non-empty binary relation \(E \subseteq W \times W'\) such that, whenever \(wEw'\), we have that:

1. For every \(p \in \mathcal{P}\), \(w \models p\) if and only if \(w' \models p\);
2. if \(wRvw'\), then there exists a world \(v'\) in \(W'\) such that \(v' \models \forall v'\) and \(w'R'v'\), and
   - if \(v \models p\) \(\min \prec R(w)\), then \(v' \models p\) \(\min \prec' R'(w')\);
3. if \(w'R'v'\), then there exists a world \(v\) in \(W\) such that \(v \models \forall v'\) and \(wRv\), and
   - if \(v' \models p\) \(\min \prec' R'(w')\), then \(v \models p\) \(\min \prec R(w)\).

**Lemma 2** (\(W\)-ordered bisimulation invariance lemma) If \(E\) is a bisimulation between \(\mathcal{W} = \langle W, R, V, \prec \rangle\) and \(\mathcal{W}' = \langle W', R', V', \prec' \rangle\), and \(wEw'\), then \(w\) and \(w'\) satisfy the same modal sentences in the extended modal language \(L^{\alpha}\).

**Proof:**

The lemma is proved by structural induction on \(\alpha \in L^{\alpha}\). We show that, for any \(w \in W\) and \(w' \in W'\), if \(wEw'\), then \(w', w' \models \alpha\) iff \(w', w \models \alpha\). For atomic propositions, and when \(\alpha = \neg \beta\) or \(\alpha = \beta_1 \lor \beta_2\), the proof is immediate. We consider the remaining two cases, namely when \(\alpha = \square \beta\) or \(\alpha = \alpha \circ \beta\).

Assume \(\alpha = \square \beta\) and let \(w', w \models \square \beta\). The proof is as for basic modal logic: Suppose \(v' \in R'(w')\). Since \(wEw'\), there is some \(v \in R(w)\) with \(v \models v'\). Therefore \(w, v \models \beta\), and hence \(w', v' \models \beta\) by the induction hypothesis. It follows that \(\mathcal{W}', w' \models \square \beta\). A symmetric argument applies if \(\mathcal{W}', w' \models \square \beta\).

Assume \(\alpha = \alpha \circ \beta\) and let \(w, w \models \alpha \circ \beta\). Suppose \(v' \in \min \prec R'(w')\). Since \(wEw'\), there is some \(v \in \min \prec R(w)\) with \(v \models v'\). Therefore \(w, v \models \beta\), and hence \(w', v' \models \beta\) by the induction hypothesis. It follows that \(\mathcal{W}', w' \models \alpha \circ \beta\). A symmetric argument applies if \(\mathcal{W}', w' \models \alpha \circ \beta\).

We now turn to bisimulations between \(R\)-ordered models. As above, two worlds are bisimilar if they satisfy the same atomic information and their modal accessibility structures match, both in terms of accessible worlds and in terms of preference of accessibility.

**Definition 12** (\(R\)-ordered bisimulation) Let \(R\)-ordered models \(\mathcal{R} = \langle W, R, V, \prec \rangle\) and \(\mathcal{R}' = \langle W', R', V', \prec' \rangle\). A bisimulation between \(\mathcal{R}\) and \(\mathcal{R}'\) is a non-empty binary relation \(E \subseteq W \times W'\) such that, whenever \(wEw'\), we have that:

1. For every \(p \in \mathcal{P}\), \(w \models p\) if and only if \(w' \models p\);
2. if \(wRvw'\), then there exists a world \(v'\) in \(W'\) such that \(v' \models \forall v'\) and \(w'R'v'\), and
   - if \(v \models p\) \(\min \prec R(w)\), then \(v' \models p\) \(\min \prec' R'(w')\);
3. if \(w'R'v'\), then there exists a world \(v\) in \(W\) such that \(v \models \forall v'\) and \(wRv\), and
   - if \(v' \models p\) \(\min \prec' R'(w')\), then \(v \models p\) \(\min \prec R(w)\).
Lemma 3 (R-ordered bisimulation invariance lemma)  
If $E$ is a bisimulation between $R = \langle W, R, V, \prec \rangle$ and $R' = \langle W', R', V', \prec' \rangle$, $w \in W$ and $w' \in W'$, then $w$ and $w'$ satisfy the same modal sentences in the extended language $L^e$.

Proof:
The proof is by structural induction on $\alpha \in L^e$ and is similar to that of Lemma 2. We show that, for any $w, w' \in W$, if $w E w'$, then $R, w \models \alpha$ iff $R', w' \models \alpha$. We only prove the case when $\alpha = \beta_1 \beta_2$.

Assume $\alpha = \beta_1 \beta_2$ and let $R, w \models \beta_1 \beta_2$. Suppose $w' v' \in \min R_v^w$. Since $w E w'$, there is some $w v \in \min R_v^w$ with $v E v'$. Therefore $R, v \models \beta_1$, and hence $R', v' \models \beta_2$ by the induction hypothesis. It follows that $R', w' \models \beta_1 \beta_2$. A symmetric argument applies if $R', w' \models \beta_1 \beta_2$.

The relationship between $L^e$ and $L^o$, and between R-ordered and W-ordered models, can be made precise using bisimulations. We first show that $L^o$ is at least as expressive as $L^e$. Given a sentence $\alpha \in L^e$, let $\alpha^o$ be the sentence obtained by replacing all occurrences of $\beta_1 \beta_2$ in $\alpha$ with $\beta_1 \beta_2$.

Definition 13 Let $W = \langle W, R, V, \prec \rangle$ be a W-ordered model. For any $u, v, w \in W$ such that $w R u$ and $w R v$ and $u \prec v$, let $w u \prec w v$. Then $W' = \langle W, R, V, \prec \rangle$ is the R-ordered model induced by $W$.

Lemma 4 For any $\alpha \in L^e$, $W = \langle W, R, V, \prec \rangle$ and $w \in W$, $w \models \alpha$ if and only if in the R-ordered model $W' = \langle W, R, V, \prec \rangle$ induced by $W$, $w \models \alpha^o$.

Proof:
The proof is simple and proceeds by structural induction on the sentence $\alpha$.

Lemma 4 shows that, if $\alpha$ and $\beta$ are not equivalent in $L^e$, then their translations $\alpha^o$ and $\beta^o$ are also not equivalent in $L^o$. Further, if $(W', w)$ and $(W', v)$ are distinguishable by some $\alpha \in L^o$, say, $W', w \models \alpha$ and $W', v \not\models \alpha$, then $R' w$ and $R' v$ are distinguishable by $\alpha^o \in L^e$. Hence, $L^o$ is at least as expressive as $L^e$.

The converse of this result may not be as obvious to see, and translating R-ordered models to W-ordered models requires more care. The light switch example (Figure 1) shows that, even in the case of a single modality, there is no direct translation of a preference order on $R$ to a preference order on $W$. There is no order on the two worlds $w_1$ and $w_2$ such that $w_2$ is the preferred result of toggling the light switch when the light is off, but $w_2$ is the preferred result when the light is on. A further problematic aspect is that R-ordered models allow for a preference order on each accessibility relation, whereas a W-ordered semantics assigns a single common preference order on worlds.

Definition 14 Let $R = \langle W, R, V, \prec \rangle$ be an R-ordered model with single accessibility relation $R$. Let $W' = W \times W$; let $V'(w v) = V(u)$; let $w R_1 v$ whenever $w R v$, and let $w v < w' v$ whenever $w \prec u \prec w' v$. Then $W = \langle W', R', V', \prec' \rangle$ is the W-ordered model induced by $R$.

As an example, we apply Definition 14 to obtain the W-ordered models induced by the models of Figures 1 and 2, and depicted in Figure 7 and Figure 8 respectively. Note that in Figure 7, $w_1 w_2 < w_1 w_1$ and $w_2 w_1 < w_2 w_2$, reflecting the intuition of normal execution of the action as an order on worlds. In Figure 8, the order on worlds is reversed, with $w_1 w_2 < w_1 w_2$ and $w_2 w_1 < w_2 w_1$, depicting the intuition of defeasible knowledge of the agent as an order on worlds.
3. Suppose \( u \mathrel{R_1 w} \). It again follows from the construction of \( \mathcal{R}_w \) that \( v \mathrel{R_1 w} \) and \( w \mathrel{E v w} \). Further, if \( vw \in \min_{\preceq_1} (R_1)^w \), then \( vw \in \min_{\preceq_1} R_1(w) \), and hence \( vw \in \min_{\preceq_1} R_1 \).

We illustrate the construction of Theorem 1 by applying Definition 13 to the induced \( \mathcal{R}_w \)-ordered model in Figure 7 to obtain the \( \mathcal{R}_w \)-ordered model of Figure 9. In Figure 9, the dashed arrows represent the preference order \( \preceq_1 \). Theorem 1 then states that the \( \mathcal{R}_w \)-ordered model of Figure 1 (with the order as described in the Introduction) is bisimilar to the \( \mathcal{R}_w \)-ordered model of Figure 9. The construction is via the \( \mathcal{R}_w \)-ordered model of Figure 7.

Similarly, the \( \mathcal{R}_w \)-ordered model of Figure 2 (again, with the order as described in the Introduction) is bisimilar to the \( \mathcal{R}_w \)-ordered model of Figure 10, which is constructed via the \( \mathcal{R}_w \)-ordered model of Figure 8.

**Corollary 1** \( L^\preceq \) and \( L^E \) can distinguish between the same modal propositions when restricted to a single modality.

**Proof:**
The bisimulation result of Theorem 1 shows that any \( \mathcal{R}_w \)-ordered model is bisimilar to some \( \mathcal{R}_w \)-ordered model induced by a \( \mathcal{R}_w \)-ordered model. Lemma 3 ensures that bisimilar worlds satisfy the same modal sentences, and that bisimilar models can distinguish between the same modal properties. We need therefore consider only \( \mathcal{R}_w \)-ordered models induced by some \( \mathcal{R}_w \)-ordered model when reasoning about expressivity. The result then follows from Lemma 4.

Corollary 1 may be seen as a negative result in the sense that, at least in the monomodal case, no richer language is obtained when substituting a preference order on the accessibility relation for the preference order on worlds. It is also clear that the results of Theorem 1 and Corollary 1 can be generalised to multi-modal languages if multiple preference relations on \( W \) are allowed.

What, then, has been gained? As we have argued, there are a number of contexts in which an order on the accessibility relation has an intuitive appeal. The induced \( \mathcal{R}_w \)-ordered models of Definition 13 are technically useful, but intuitively hard to motivate. However, from an implementation perspective, we now know that a reasoner based on a \( \mathcal{R}_w \)-ordered semantics suffices also for reasoning over \( \mathcal{R}_w \)-ordered models. This, together with our previous results (Britz and Varzinczak 2013), establish the following:

**Corollary 2** Satisfiability checking for monomodal \( L^\preceq \) is \( \text{PSPACE-complete} \).

**Discussion and Related Work**

It might be worth emphasising that the logics we have investigated here do not aim at providing a formal account of the notion of most, as addressed in the study of generalised quantifiers (Lindström 1966) and, more recently, in a modal context by Veloso et al. (2009) and Askounis et al. (2012). Clearly, they are not about degrees of truth, as it has been studied in fuzzy logics, nor about degrees of possibility and necessity, as addressed by possibilistic logics (Dubois, Lang, and Prade 1994). Instead, here we have investigated a rather complementary notion to those ones, namely that of normal, expected, practical necessity, which need not rely on majority or degrees of likelihood.

In a sense, the notions we investigated here can be seen as the qualitative counterpart of possibilistic modalities (Liau 1999; Liau and Lin 1996). (We thank an anonymous referee for pointing this out to us.) There, each possible world \( w \) is associated with a possibility distribution \( \pi_w : W \rightarrow [0, 1] \), the intuition of which is to capture the degree of likelihood (in terms of belief) of all possible worlds w.r.t. \( w \). In that setting, the pairs \( (w, w') \) for which \( \pi_w(w') \) is maximal correspond here to the most preferred pairs in a single accessibility relation. In this sense, there are strong links between monomodal \( \equiv \) and the preferential possibilistic semantics for epistemic reasoning.

Currently, the definition of \( \mathcal{R}_w \)-ordered model (Definition 6) allows only for elements of the same accessibility relation \( R_i \) to be ordered (via the respective \( \preceq_i \)). More generally, we could have defined \( \preceq \) as a relation on \( \bigcup_{1 \leq i \leq n} R_i \times \)
\( \bigcup_{1 \leq i \leq n} R_i \), so that we allow pairs \((w, w')\) belonging to different \( R \)-components to be compared as well. An investigation of the philosophical and practical ramifications of this alternative definition is left for future work.

We have seen that one can obtain \( R \)-ordered models from \( W \)-ordered models by inducing an ordering on edges from the ordering on worlds. The result is an ‘embedding’ of \( \mathbb{D} \) into \( \mathcal{L}^\alpha \). Conversely, in the monomodal case, we can obtain \( W \)-ordered models from \( R \)-ordered models by inducing an ordering on worlds from an ordering on edges. If we allow multiple preferences on worlds, the latter result can easily be generalised, thereby establishing that \( \mathcal{L}^\alpha \) and \( \mathcal{L}^\beta \) are equally expressive. This would have an interesting consequence, namely that the notions of ‘normal effects’ and ‘normal executions’ of actions are one and the same. This a priori counter-intuitive claim is easily justifiable. It turns out the effects of an action (the worlds one ‘lands’ in) depend to a large extent on what the current state of the world (the ‘departing’ points) is. In other terms, talking about effects (tacitly) amounts to talking about pairs \((w, w')\), linking both a context of execution and the action’s outcome. This feature just carries over when normality is considered.

In this work, we have not addressed the question as to what an appropriate notion of entailment for \( \mathcal{L}^\alpha \) is and have contented ourselves with the standard (Tarskian) definition, which is monotonic (and therefore inappropriate in many contexts). The recent results by Booth et al. (2015) in a propositional setting may provide us with a springboard to investigate this matter in more expressive languages such as those we are interested in here.

**Outlook on Further Work**

We shall now briefly discuss about possible ideas for exploration stemming from the present work.

**\( R \)-based Conditionals**

A framework for representing and reasoning with defeasibility would not be complete without an account of (defeasible) conditionals. Here we catch a glimpse of two versions thereof which can both be defined in our \( R \)-ordered models semantically.

Given an \( R \)-ordered model \( \mathcal{R} \), for every propositional sentence \( \alpha \), let \( R_\alpha := \{(w, w') \mid \mathcal{R}, w \models \alpha \text{ and } \mathcal{R}, w' \models \alpha \} \) and \( \prec_\alpha \), its corresponding preference relation. (Of course, if we work in a finite propositional language, then there are finitely many of such \( R_\alpha \)s and \( \prec_\alpha \)s.) We can then define a conditional statement as a macro in \( \mathcal{L}^\alpha \) as follows:

\[ \alpha \sim_{\mathcal{R}} \beta \text{ if and only if } \exists_{\mathcal{R}} \beta \]

Such a definition, of course, has its limitations, as it only allows for propositional sentences in the antecedent of the conditional. A generalisation to the case where \( \alpha \in \mathcal{L}^\beta \) would hardly improve matters, as we would end up with an infinite number of accessibility relations in the \( R \)-component of our \( R \)-ordered models.

Fortunately, we can do better than this. First, we need to define an extra, identity relation \( id \) on \( W \) and order its elements in the same way as for the other \( R \)-components. The intuition of doing so is that the most normal \( id \)-arrows correspond to the most normal worlds, i.e., we get an ordering on worlds induced by the ordering on the elements of the identity relation. With this, we can define our second candidate for a conditional in the following way. First, for every \( \alpha \in \mathcal{L}^\alpha \), let \( Id^\alpha := \{(w, w) \mid \mathcal{R}, w \models \alpha \} \). Then

\[ \mathcal{R}, w \models \beta \]

We shall leave an investigation of the appropriateness of \( \sim_{\mathcal{R}} \) as a defeasible conditional for future work.

**Next Steps in Preferential Reasoning for DLs**

In the context of formal ontologies specified in Description Logics (Baader et al. 2007), placing a preference order on binary relations as we proposed here has a natural appeal. As an example, consider the role name hasChild: ‘Normal’ tuples in this relation may be biological or adopted parent-child tuples, while an ‘exceptional’ tuple may be an appointed legal guardian parent-child tuple. In this example, there is nothing exceptional about either the legal guardian or the child—the exceptionality lies in the nature of their relationship.

To make things more precise, given a DL interpretation \( I = (\Delta^I, \cdot, \cdot) \), we can enrich it with a collection of preference relations \( \sim_I := \langle \sim_{\mathcal{R}}^I, \ldots, \sim_{\mathcal{R}}^I \rangle \), one for each role name and each of which satisfying the conditions in Definition 6. Armed with this semantic construction, it becomes possible to:

- Define defeasible value restrictions (Britz et al. 2013) of the form \( \forall r, C \), like \( \forall \text{ hasChild}. \text{Male} \), which refers to those individuals whose most normal parenting relations are of male children;
- State defeasible role inclusions of the form \( r_1 \subseteq r_2 \), as in e.g. parentOf \( \preceq \) progenitorOf, which stipulates that the role of being a parent is usually that of also being the progenitor;
- State typicality-based role instances in the ABox of the form \( \bullet(a, b) \), where \( \bullet \) is the extension of a typicality operator (Booth, Meyer, and Varzinczak 2012; Giordano et al. 2007) to roles, like \( \text{hasChild}(\text{john}, \text{anne}) \), conveying the information that, under the interpretation of role hasChild, the tuple (john, anne) is to be regarded as a typical one;
- State defeasible role properties like in saying that role marriedTo is normally functional and that partOf is normally transitive, while allowing for exceptions, i.e., less normal tuples failing the relation’s property under consideration.

Moreover, definitions analogous to those in the preceding subsection would allow us to:

- State defeasible concept subsumptions (Britz, Heidema, and Meyer 2008; Britz, Meyer, and Varzinczak 2011b; Casini and Straccia 2010; Giordano et al. 2007) of the form \( C \subseteq D \), as in \( \text{Mother} \subseteq \exists \text{marriedTo} \), of which the intuition is that usually, mothers are married.
It is an open question whether a result similar to that obtained in Theorem 1 holds in a DL context. Roles can be reified, similar to the reification of n-ary relations in DLs (Sattler, Calvanese, and Motik 2007), as a workaround to model preferences on tuples as preferences on objects in a DL enriched with a preferential subsumption relation \( \preceq \). Nevertheless, it is not immediately clear how the addition of preferential roles to a DL with preferential subsumption would affect its expressivity.

**Defeasible Comparative Epistemic Logic**

By placing a preference relation on the accessibility relations, we can get to a generalisation of Comparative Epistemic Logic (CEL) (Ditmarsch, Hoek, and Kooi 2012).

In CEL, a statement of the form \( a \geq b \) intuitively means “agent \( b \) knows at least as much as agent \( a \)”. The corresponding semantics is given by:

\[ \models M, w \models a \geq b \text{ if and only if } R^a_w \subseteq R^b_w. \]

In the context of our enriched semantic framework, we could envisage making statements of the form “agent \( b \) normally knows as much as agent \( a \)”, of which a semantics can be given by the condition \( \min_{<_a} R^b_w \subseteq R^a_w \).

**Summary and Conclusion**

The contributions of the present paper can be summarised as follows: (i) the motivation for and the definition of a semantic structure allowing for the ordering of pairs of worlds (instead of worlds tout court, as is customary in traditional NMR formalisms) and (ii) a generalisation of bisimulation to the preferential case together with a result relating our new semantics to that we studied in previous work and showing that, in the monomodal case, they are equivalent.

We have introduced a logic allowing for modal operators the intuition of which is to capture the idea of some transitions being more normal than others. As we have seen, our \( R \)-ordered models can be used to provide the extended language with an intuitive and elegant semantics. The resulting framework provides for an alternative formalisation for the notion of defeasible necessity we studied previously.

We have given examples, in an action, epistemic and deontic contexts, of what this semantic structure, as simple as it is, would allow us to represent (or give a meaning to) that one cannot do with standard Kripkean semantics. Likewise, we have briefly illustrated the fruitfulness of our definitions in other formalisms, in particular in a DL setting.

**Acknowledgements**

This work is based upon research supported in part by the Brazilian National Council for Scientific and Technological Development (CNPq) under grant number 302002/2014-6. This work was also partially funded by the National Research Foundation of South Africa (UIDs 81225 and 85482, IFR1202160021 and IFR2011032700018).

**References**


Properties of ABA$^+$ for Non-Monotonic Reasoning

Kristijonas Čyras and Francesca Toni

Imperial College London, UK

Abstract

We investigate properties of ABA$^+$, a formalism that extends the well studied structured argumentation formalism Assumption-Based Argumentation (ABA) with a preference handling mechanism. In particular, we establish desirable properties that ABA$^+$ semantics exhibit. These pave the way to the satisfaction by ABA$^+$ of some (arguably) desirable principles of preference handling in argumentation and non-monotonic reasoning, as well as non-monotonic inference properties of ABA$^+$ under various semantics.

1 Introduction

Recent decades have seen a number of non-monotonic reasoning (NMR) formalisms advanced (see e.g. (Brewka, Nielmå, and Truszczynski 2007) for an overview). Since preferences are ubiquitous in common-sense reasoning, there has been a considerable effort to integrate preference information within NMR formalisms (cf. e.g. (Brewka, Truszczynski, and Niemela 2008; Delgrande et al. 2004; Domshlak et al. 2011; Kaci 2011)). To evaluate distinct formalisms, various properties of both non-monotonic inference and preference handling have been proposed, see e.g. (Makinson 1988; Kraus, Lehmann, and Magidor 1990; Brewka and Eiter 1999; Brewka, Truszczynski, and Woltran 2010; Šimko 2014).

Meanwhile, argumentation (as overviewed in (Rahwan and Simari 2009)) has become an established branch of AI widely used for NMR (see e.g. (Dung 1995; Bondarenko et al. 1997; Modgil and Prakken 2013)). Broadly speaking, information in argumentation is represented via arguments, while attacks among them indicate conflicts. Procedures, known as argumentation semantics, are employed to select extensions, i.e. sets of collectively acceptable arguments. Preferences in argumentation also play a significant role (cf. e.g. (Simari and Loui 1992; Kaci 2011)), by allowing to, for instance, discriminate among arguments or extensions. Over the years, numerous formalisms of argumentation with preferences have been presented (see Section 7) and some properties for argumentation with preferences indicated (e.g. (Brewka, Truszczynski, and Woltran 2010; Modgil and Prakken 2013; Amgoud and Vesic 2014; Dung 2016)).

NMR properties are also adaptable to argumentation setting. For example, the well known non-monotonic inference properties of Cautious Monotonicity and Cumulative Transitivity (cf. (Makinson 1988; Kraus, Lehmann, and Magidor 1990)) concern what happens when a conclusion reached through a reasoning process is added to the knowledge base to reason with anew. These properties have been cast with respect to extensions in argumentation, in e.g. (Cyras and Toni 2015; Dung 2016).

Preference handling properties for NMR can be phrased in terms of extensions in argumentation too. For instance, the well known Principle I from (Brewka and Eiter 1999) regarding preferred answer sets can be applied to argumentation semantics thus: if two extensions $E_1$ and $E_2$ coincide except for two arguments $A \in E_1 \setminus E_2$ and $B \in E_2 \setminus E_1$ such that $A$ is preferred over $B$, then $E_2$ should not be chosen as a ‘preferable’ extension. Likewise, a common property of NMR says that, in the absence of preference information, a formalism extended with a preference handling mechanism should return the same extensions as the preference-free version of the formalism (cf. (Brewka, Truszczynski, and Woltran 2010; Šimko 2014)).

In this paper, drawing from the above mentioned works, we investigate various properties of a recently proposed NMR formalism ABA$^+$ (Cyras and Toni 2016). ABA$^+$ extends with a preference handling mechanism a well established argumentation formalism, Assumption-Based Argumentation (ABA) (Bondarenko et al. 1997; Toni 2014). Whereas a common way to approach preferences in argumentation is to use preference information to discard the attacks from arguments that are less preferred than the ones they attack (see e.g. (Amgoud and Cayrol 2002; Bench-Capon 2003; Kaci and van der Torre 2008; Brewka et al. 2013; Besnard et al. 2014)), ABA$^+$ instead reverses such attacks. We show that ABA$^+$’s method of accounting for preferences satisfies (arguably) desirable properties.

On the one hand, we consider preference handling properties from (Brewka and Eiter 1999; Brewka, Truszczynski, and Woltran 2010; Amgoud and Vesic 2014) and show their satisfaction under various ABA$^+$ semantics. On the other hand, building on the investigations of Cumulative Transitivity and Cautious Monotonicity for ABA (Cyras and Toni 2015), we analyse ABA$^+$ in the light of these non-monotonic inference properties, and show that results obtained for ABA carry over to ABA$^+$. In addition, we make use of the well known principle of Contraposition of rules.
(see e.g. (Modgil and Prakken 2013)) and prove it guarantees that ABA+ semantics satisfy desirable properties akin to those in e.g. (Dung 1995; Bondarenko et al. 1997; Modgil and Prakken 2013).

The paper is organized as follows. Sections 2 and 3 give preliminaries on ABA and ABA+. In Section 4 ABA+ semantics are analysed. Preference handling properties of ABA+ are studied in Section 5, while Section 6 concerns ABA+ and non-monotonic inference properties. After discussing related work (Section 7), we conclude in Section 8.

2 Preliminaries

We base the following ABA background on (Toni 2014).

Definition 1. An ABA framework is a tuple $(L, R, A, \rightarrow)$, where:

- $(L, R)$ is a deductive system with a language $L$ and a set $R$ of rules of the form $\varphi_0 \leftarrow \varphi_1, \ldots, \varphi_m$ with $m \geq 0$ and $\varphi_i \in L$ for $i \in \{0, \ldots, m\}$; $\varphi_0$ is referred to as the head of the rule, and $\varphi_1, \ldots, \varphi_m$ is referred to as the body of the rule; if $m = 0$, then the rule $\varphi_0 \leftarrow \varphi_1, \ldots, \varphi_m$ is written as $\varphi_0 \leftarrow \varphi_1, \ldots, \varphi_m$ and is said to have an empty body;
- $A \subseteq L$ is a non-empty set, whose elements are referred to as assumptions;
- $\neg : A \rightarrow L$ is a total map: for $\alpha \in A$, the $L$-formula $\pi$ is referred to as the contrary of $\alpha$.

We focus on flat ABA frameworks, where no assumption is the head of any rule. Flat ABA frameworks are very common, and capture, as instances, widely used paradigms of non-monotonic reasoning, such as Logic Programming and Default Logic (see e.g. (Bondarenko et al. 1997)).

Definition 2. A deduction for $\varphi \in L$ supported by $S \subseteq L$ and $R \subseteq R$, denoted by $S \vdash^R \varphi$, is a finite tree with the root labelled by $\varphi$, leaves labelled by $\top$ or elements from $S$, the children of non-leaf nodes $\psi$ labelled by the elements of the body of some rule from $R$ with head $\psi$, and $R$ being the set of all such rules. For $E \subseteq L$, the conclusions $Cn(E)$ of $E$ is the set of elements with deductions supported by $S \subseteq E$ and some $R \subseteq R$, i.e. $Cn(E) = \{ \varphi \in L : \exists S \vdash^R \varphi, S \subseteq E, R \subseteq R \}$. Assumption-level attacks in ABA are defined thus.

Definition 3. A set $A \subseteq A$ attacks a set $B \subseteq A$, denoted $A \rightsquigarrow B$, if there is a deduction $A \vdash^R \beta$, for some $\beta \in B$, supported by some $A' \supseteq A$ and $R \subseteq R$. For $E \subseteq A$, also called an extension, we say that:

- $E$ is conflict-free if $E \not\vdash E$;
- $E$ defends $\alpha \in A$ if for all $B \rightsquigarrow \{ \alpha \}$ it holds that $E \rightsquigarrow B$;
- $E$ is admissible if $E$ is conflict-free and defends all $\alpha \in E$.

The most standard ABA semantics are as follows.

Definition 4. A conflict-free set $E \subseteq A$ is:

- stable, if $E \rightsquigarrow \{ \beta \}$ for every $\{ \beta \} \subseteq A \setminus E$;
- complete if $E$ is admissible and contains every assumption it defends;
- preferred if $E$ is $\subseteq$-maximally admissible;
- grounded if $E$ is $\subseteq$-minimally complete;
- ideal if $E$ is $\subseteq$-maximal such that $E$ is admissible and contained in all preferred extensions.

Example 5. Let $L = \{ \alpha, \beta, \pi \}$, $R = \{ \pi \leftarrow \beta \}$ and $A = \{ \alpha, \beta \}$. In $(L, R, A, \rightarrow)$, $\{ \beta \}$ attacks both $\{ \alpha \}$ and $\{ \alpha, \beta \}$, while $\{ \alpha, \beta \}$ attacks itself and $\{ \alpha \}$. $(L, R, A, \rightarrow)$ can be graphically represented via its assumption framework, pictured below (in illustrations of assumption frameworks, nodes hold sets of assumptions while directed edges indicate attacks):

This $(L, R, A, \rightarrow)$ has a unique complete extension $\{ \beta \}$, which is also grounded, ideal, preferred and stable, and has conclusions $Ch(\{ \beta \}) = \{ \pi, \beta \}$.

3 ABA+

ABA+ (Čyras and Toni 2016) extends ABA with preferences as follows.

Definition 6. An ABA+ framework is any tuple $(L, R, A, \rightarrow, \ll)$, where $(L, R, A, \rightarrow)$ is an ABA framework and $\ll$ is a preorder (i.e. a transitive and reflexive binary relation) on $A$.

Differently from e.g. (Modgil and Prakken 2013; 2014; García and Simari 2014), ABA+ considers preferences on assumptions rather than (defeasible) rules. This is not, however, a conceptual difference, since assumptions are the only defeasible component in ABA+.

Unless stated differently, we consider a fixed, but otherwise arbitrary ABA+ framework $(L, R, A, \rightarrow, \ll)$, and implicitly assume $(L, R, A, \rightarrow)$ to be its underlying ABA framework. The strict counterpart $\prec$ of $\ll$ is defined as $\alpha \prec \beta$ iff $\alpha \ll \beta$ and $\beta \ll \alpha$, for any $\alpha$ and $\beta$.

ABA+ attack relation is given thus.

Definition 7. A set $A \subseteq A$ of assumptions $\ll$-attacks a set $B \subseteq A$ of assumptions, written as $A \ll B$, if:

- either there is a deduction $A' \vdash^R \beta$, for some $\beta \in B$, supported by $A' \subseteq A$ and $\exists \beta' \in A'$ with $\alpha' \ll \beta'$; or there is a deduction $B' \vdash^R \pi$, for some $\pi \in A$, supported by $B' \subseteq B$, and $\exists \pi' \in B'$ with $\beta' \ll \pi'$.

The first type of attack is called normal, and the second one reverse.

ABA+ requires a standard ABA attack to be reversed whenever the attacker has an assumption less preferred than the one attacked. The following example illustrates.

Example 8. Recall $(L, R, A, \rightarrow)$ from Example 5. Suppose $\beta \ll \alpha$. In the ABA+ framework $(L, R, A, \rightarrow, \ll)$, $\{ \beta \}$ ‘tries’ to attack $\{ \alpha \}$, but is prevented by the preference $\beta \ll \alpha$. Instead, $\{ \alpha \} \ll\ll \{ \beta \}$, and likewise $\{ \alpha, \beta \}$, via reverse attack, and the latter $\ll$-attacks both itself and $\{ \beta \}$ via reverse attack. $(L, R, A, \rightarrow, \ll)$ can be represented graphically as follows (reverse attacks in assumption frameworks will be denoted by dotted arrows):

In contrast with the ABA framework, where $\{ \beta \}$ is unattacked and generates an attack on $\{ \alpha \}$, in the
ABA\textsuperscript{+} framework, \(\{\alpha\}\) is \(<\)-unattacked and \(<\)-attacks all sets of assumptions that contain \(\beta\). This concords with the intended meaning of the preference \(\beta < \alpha\), that the conflict should be resolved in favour of \(\alpha\).

This concept of \(<\)-attack reflects the interplay between deductions, contraries and preferences, by representing inherent conflicts among sets of assumptions while accounting for preference information. Normal attacks follow the standard notion of attack in ABA, additionally, preventing the attack to succeed when the attacker uses assumptions less preferred than the one attacked. Reverse attacks, meanwhile, resolve the conflict between two sets of assumptions by favouring the one containing an assumption whose contrary is deduced, over the one which uses less preferred assumptions to deduce that contrary.

The notions of conflict-freeness and defence w.r.t. \(\sim<\), and ABA\textsuperscript{+} semantics are given as follows.

**Definition 9.** For \(E \subseteq A\) we say that:

- \(E\) is \(<\)-conflict-free if \(E \not\sim< E\);
- \(E\) \(<\)-defends \(\alpha \in A\) if for all \(B \sim< \{\alpha\}\) it holds that \(E \sim< B\); and
- \(E\) is \(<\)-admissible if \(E\) is \(<\)-conflict-free and \(<\)-defends every \(\alpha \in E\).

In Example 8, \(\emptyset\), \(\{\alpha\}\) and \(\{\beta\}\) are conflict-free in \((L, R, A, \sim<)\) and \(<\>-conflict-free in \((L, R, A, \sim<, \subseteq)\), whereas \(\{\alpha, \beta\}\) is not \(<\>-conflict-free in either framework.

**Definition 10.** A \(<\>-conflict-free extension \(E \subseteq A\) is:

- \(<\>-stable if \(E \sim< \{\alpha\}\) for every \(\{\alpha\} \subseteq A \setminus E\);
- \(<\>-complete if \(E\) is \(<\>-admissible and contains every assumption it \(<\>-defends;
- \(<\>-preferred if \(E\) is \(\subseteq\)-maximally \(<\>-admissible;
- \(<\>-grounded if \(E\) is \(\subseteq\)-minimally \(<\>-complete;
- \(<\>-ideal if \(E\) is \(\subseteq\)-maximal such that \(E\) is \(<\>-admissible and contained in all \(<\>-preferred extensions.

In Example 8, \(\{\alpha\}\) is a unique \(<\>-stable, \(<\>-complete, \(<\>-preferred, \(<\>-grounded and \(<\>-ideal extension.

Henceforth, we assume \(\sigma \in \{\)stable, complete, preferred, grounded, ideal\} and use \(<\>-\sigma\) to denote any ABA\textsuperscript{+} semantics.

We recall several features that ABA\textsuperscript{+} possesses and that will be used later.

**Lemma 1.** Let \(A' \subseteq A \subseteq A \subseteq B \subseteq B' \subseteq A\) be given. If \(A' \sim< B',\) then \(A \sim< B\).

**Lemma 2.** For any \(A, B \subseteq A:\)

- if \(A \sim< B,\) then either \(A \sim< B\) or \(B \sim< A\);
- if \(A \sim< B,\) then either \(A \sim< B\) or \(B \sim< A\).

4 Properties of ABA\textsuperscript{+} Semantics

To ensure that the familiar relations between semantics carry from ABA over to ABA\textsuperscript{+}, we want to guarantee the so-called Fundamental Lemma (Dung 1995; Bondarenko et al. 1997) (see below). To this end, we follow the well-established structured argumentation formalism ASPIC\textsuperscript{+} (Modgil and Prakken 2013; 2014) and impose the principle of Contraposition, reformulated for ABA\textsuperscript{+} as follows.

**Axiom 11.** \((L, R, A, \sim<, \subseteq)\) satisfies the Axiom of Contraposition if for all \(A \subseteq A, R \subseteq R\) and \(\beta \in A\) it holds that if \(A \vdash R_\beta\), then for every \(\alpha \in A\), there is \(R_\alpha \subseteq R\) with \((A \setminus \{\alpha\}) \cup \{\beta\} \vdash R_\alpha\).

This axiom requires that if an assumption plays a role in deriving the contrary of another assumption, then it should contrapositively be possible for the latter to induce a derivation of the contrary of the former assumption too. The following example illustrates the effect Contraposition has in ABA\textsuperscript{+}.

**Example 12.** Let \(R = \{\beta \leftarrow \alpha, \gamma\}, A = \{\alpha, \beta, \gamma\}\) and \(\alpha < \beta, \alpha < \gamma\). (The language and the contrary mapping are implicit from \(R \) and \(A\).) This ABA\textsuperscript{+} framework \((L, R, A, \sim<, \subseteq)\) does not satisfy the Axiom of Contraposition. Its assumption framework (omitting \(\emptyset, A\) and \(<\>-attacks to and from \(A\)) is shown below:

![Diagram](https://via.placeholder.com/150)

There are no extensions under, for instance, \(<\>-complete semantics, because all the singletons \(\{\alpha\}, \{\beta\}\) and \(\{\gamma\}\) are \(<\>-unattacked, but \{\alpha, \beta, \gamma\}\) is not \(<\>-conflict-free. If the rules \(\bar{\pi} \leftarrow \beta, \gamma\) and \(\bar{\gamma} \leftarrow \alpha, \beta\) are added to \(R\) to constitute \(R'\), then the resulting \((L, R', A, \sim<, \subseteq)\) satisfies the Axiom of Contraposition and its assumption framework looks as follows (\(<\>-attacks that are both normal and reverse are depicted as solid directed edges):

![Diagram](https://via.placeholder.com/150)

Here, \(\{\beta, \gamma\}\) is a unique \(<\>-complete extension.

We prove next that in the presence of Contraposition, the Fundamental Lemma is guaranteed to hold in ABA\textsuperscript{+}.

**Lemma 3.** Suppose that \((L, R, A, \sim<, \subseteq)\) satisfies the Axiom of Contraposition. Let \(S \subseteq A\) be \(<\>-admissible and assume that \(S\) \(<\>-defends \alpha, \alpha' \in A\). Then \(S \cup \{\alpha\}\) is \(<\>-admissible and \(<\>-defends \alpha'\).

**Proof.** Note that if \(\alpha \in S\), then \(S \cup \{\alpha\}\) is trivially \(<\>-admissible. So assume \(\alpha \notin S\) and suppose for a contradiction that \(S \cup \{\alpha\}\) is not \(<\>-admissible. Then it is either not \(<\>-conflict-free, or does not \(<\>-defend itself. Suppose first \(S \cup \{\alpha\} \sim< S \cup \{\alpha\}\) via either (1) normal or (2) reverse attack. We show that either leads to a contradiction.

1. \(S \cup \{\alpha\} \sim< S \cup \{\alpha\}\) via normal attack. As \(S\) is \(<\>-conflict-free and \(<\>-defends \alpha, \) this \(<\>-attack must involve \(\alpha.\) I.e. \(S' \cup \{\alpha\} \vdash R_\beta\) for some \(S' \subseteq S\) and \(\beta \in S \cup \{\alpha\},\) and \(\forall S' \subseteq S \cup \{\alpha\}\) we find \(S' \neq \beta.\) If \(\beta = \alpha,\) then \(S' \cup \{\alpha\} \sim< \{\alpha\},\) and so \(S \sim< S' \cup \{\alpha\}.\) Else, if \(\beta \in S',\) then
$S' \cup \{\alpha\} \rightsquigarrow< S$, and so $S \rightsquigarrow< S' \cup \{\alpha\}$ as well. We show that we can similarly obtain $S \rightsquigarrow< S' \cup \{\alpha\}$ in case (2) too.

2. $S \cup \{\alpha\} \rightsquigarrow< S \cup \{\alpha\}$ via reverse attack. As in 1., this $<$-attack must involve $\alpha$, i.e. $S' \cup \{\alpha\} \vdash R S'$ for some $S' \subseteq S$ and $\beta \in S \cup \{\alpha\}$, and $\exists s' \in S' \cup \{\alpha\}$ such that $s' \not< \beta$. If $\beta \in S$, then $S \rightsquigarrow< S' \cup \{\alpha\}$. Else, if $\beta = \alpha$, then $s' \not< \alpha$ (by asymmetry of $<$), and using the Axiom of Contraction we find $A \vdash R \gamma$ for $A \subseteq (S' \cup \{\alpha\}) \setminus \{s'\}$, so that $S' \cup \{\alpha\} \rightsquigarrow S$. Then, by Lemma 2, either $S' \cup \{\alpha\} \rightsquigarrow S$ or $S \rightsquigarrow< S' \cup \{\alpha\}$, which yields $S \rightsquigarrow< S' \cup \{\alpha\}$ in any case.

In either (1) or (2), $S \rightsquigarrow< S' \cup \{\alpha\}$, and as $S$ is $<$-conflict-free and $<$-defends $\alpha$, this $<$-attack must be reverse and involve $\alpha$: $A_1 \cup \{\alpha\} \vdash R_1 \gamma$, $\gamma \in S$, $A_1 \subseteq S'$, and $\exists s'_1 \in A_1$ with $s'_1 \not< \gamma$. Without loss of generality take $s'_1$ to be $<$-minimal such. By the Axiom of Contraction, there is $S_1 \cup \{\alpha\} \vdash R_1 \gamma$ with $S_1 \subseteq (A_1 \setminus \{s'_1\}) \cup \{\gamma\}$ and $\forall \gamma \in S_1 \; x \not< s'_1$ (by $<$-minimality of $s'_1$). That is, $S_1 \subseteq S_2 \cup \{\alpha\}$, So we find $S \rightsquigarrow< S_1 \cup \{\alpha\}$, again via reverse attack involving $\alpha$: $A_2 \cup \{\alpha\} \vdash R_2 \gamma$, $\gamma \in S$, $A_2 \subseteq S_1$, and $\exists s'_2 \in A_2$ with $s'_2 \not< \gamma$. We again impose $<$-minimality on $s'_2$ and by the Axiom of Contraction get $S_2 \cup \{\alpha\} \vdash R_2 \gamma$, $\gamma \in S_2 \subseteq (A_2 \setminus \{s'_2\}) \cup \{\gamma\}$ and $\forall \gamma \in S_2 \; x \not< s'_2$.

As deductions are finite and $<$ asymmetric, the procedure described above will eventually exhaust pairs of $s'_1 \in A_2$ and $s_2 \in S_2$ such that $s'_1 \not< s_2$, so that $S \rightsquigarrow< S_2 \cup \{\alpha\}$ will have to be a normal attack, for some $k$. But this leads to a contradiction to $S$ being $<$-admissible and $<$-defending $\alpha$.

Hence, by contradiction, $S \cup \{\alpha\}$ is $<$-conflict-free.

We now want to show that $S \cup \{\alpha\}$ $<$-defends itself. So let $B \rightsquigarrow< S \cup \{\alpha\}$. As $S$ is $<$-admissible and $<$-defends $\alpha$, we consider this $<$-attack to be reverse and involving $\alpha$: $S' \cup \{\alpha\} \vdash R_3 \gamma$, $\gamma \in S$, $B \subseteq S'$, and $\exists s' \in S' \cup \{\alpha\}$ with $s' \not< \beta$. By the Axiom of Contraction, $S_1 \vdash R_3 \gamma$, $\gamma \in S$, $S_1 \subseteq ((S' \cup \{\alpha\}) \setminus \{s'\}) \cup \{\beta\}$, whence $S \cup \{\gamma\} \rightsquigarrow< S_1$. This $<$-attack cannot be normal on $(S' \cup \{\alpha\}) \setminus \{s'\}$, due to $<$-conflict-freeness of $S' \cup \{\alpha\}$; while, if it is normal on $\beta$, then $S_1 \rightsquigarrow< \gamma$, as required. Instead, $S \cup \{\alpha\} \rightsquigarrow< \gamma$, via reverse attack: $B_1 \vdash R_3 \gamma$, $\gamma \in S \cup \{\alpha\}$, $B_1 \subseteq S_1$, and $\exists s'_1 \in B_1$ with $s'_1 \not< \gamma$. Due to $<$-conflict-freeness of $S' \cup \{\alpha\}$, we find $\gamma \in B_1$. Then, again, by the Axiom of Contraction, we find $S_2 \vdash R_3 \gamma$, $\gamma \in S_2 \subseteq (B_1 \setminus \{s'_1\}) \cup \{\gamma\}$, and $\exists s'_2 \in S_2$. With the proof of $<$-conflict-freeness, this process must terminate with a normal attack $S \cup \{\alpha\} \rightsquigarrow< B$, so that $S \cup \{\alpha\}$ eventually $<$-defends itself.

Finally, given that $S$ $<$-defends $\alpha'$ to begin with, using Lemma 1 we conclude that $S \cup \{\alpha\}$ $<$-defends $\alpha'$ too.

For the rest of this section, we assume that $(L, R, A, \leq)$ satisfies the Axiom of Contraction.

We can now define the $<$-defence operator $Def$, inspired by (Dung 1995).

**Definition 13.** $Def : \varphi(A) \rightarrow \varphi(A)$ is defined as follows: for $A \subseteq A$, $Def(A) = \{\alpha \in A : A \not< \alpha \}$.

By Lemma 1, $Def$ is monotonic: if $A \subseteq B \subseteq A$, then $Def(A) \subseteq Def(B)$. Hence, $Def$ has a unique least fixed point, which is in addition a unique $<$-grounded extension of $(L, R, A, \leq)$, as shown next.

**Proposition 4.** $(L, R, A, \leq)$ admits a unique $<$-grounded extension.

**Proof.** First, observe that $\emptyset$ is $<$-admissible in $(L, R, A, \leq)$. The least fixed point $G$ can be given as $\bigcup_{i \in \mathbb{N}} Def^i(\emptyset)$. By Lemma 3, $G$ is $<$-admissible. It is clearly $<$-complete (as $G = Def(G)$) and unique $<$-minimal such (as the least fixed point). Hence, $G$ is a unique $<$-grounded extension of $(L, R, A, \leq)$.

As a consequence of Proposition 4, we get the following.

**Corollary 5.** $(L, R, A, \leq)$ admits a $<$-complete extension.

Using Lemma 3, we can prove the following results.

**Proposition 6.** $(L, R, A, \leq)$ admits a $<$-preferred extension.

**Proof.** By Lemma 3, the collection of $<$-admissible supersetsof $\emptyset$ is partially ordered by subset inclusion $\subseteq$, so any sequence $\emptyset = A_1 \subseteq \ldots \subseteq A_n \subseteq \ldots$ of $<$-admissible sets of assumptions (for $n$ an ordinal) has an upper bound $A = \bigcup_{i \in \mathbb{N}} A_i$. Then $A \subseteq A$ is $<$-admissible: if it were not $<$-conflict-free, then some $A_n$ would not be either; and for any $B \not< A$ we have $B \not< A_n$, for some $n$, so that $A_n \not< B$, and hence $A \not< B$ too. Since every chain $\emptyset \subseteq A_1 \subseteq \ldots \subseteq A_n \subseteq \ldots$ admits an $<$-admissible upper bound, every such chain has a $<$-maximally $<$-admissible set of assumptions, according to Zorn’s Lemma. As $\emptyset$ is $<$-admissible, $(L, R, A, \leq)$ admits at least one $<$-maximally $<$-admissible—i.e. a $<$-preferred—extension.

**Proposition 7.** Every $<$-preferred extension of $(L, R, A, \leq)$ is a $<$-complete extension too.

**Proof.** Let $E$ be a $<$-preferred extension of $(L, R, A, \leq)$ and suppose for a contradiction that it is not $<$-complete. Let $E \not< \text{defend} some \alpha \in A \setminus E$. As $E$ is $<$-admissible, $E \cup \{\alpha\}$ is $<$-admissible, by Lemma 3. But then $E$ is not $<$-maximally $<$-admissible, contrary to $E$ being $<$-preferred. Hence, by contradiction, $E$ must be $<$-complete.

Further, as in ABA, $<$-stable semantics is subsumed by both $<$-preferred and $<$-complete semantics, as shown next.

**Proposition 8.** Any $<$-stable extension of $(L, R, A, \leq)$ is a $<$-preferred extension too.

**Proof.** Let $E$ be a $<$-stable extension of $(L, R, A, \leq)$, As $E$ $<$-attacks every $\{\beta\} \not< E$, it must be $<$-maximally $<$-admissible. Hence, $E$ is $<$-preferred.

**Proposition 9.** Any $<$-stable extension of $(L, R, A, \leq)$ is a $<$-complete extension too.

**Proof.** Let $E$ be a $<$-stable extension of $(L, R, A, \leq)$. For any $\beta \not< E$, $<$-stability of $E$ means that $E \not< \{\beta\}$, and if $E \not< \text{defended} \beta$ as well, it would mean that $E \not< E$, contradicting its $<$-conflict-freeness. Hence, $E$ contains every assumption it $<$-defends, and so is $<$-complete.
Finally, we consider $<\text{-ideal semantics}$.

**Proposition 10.** $(L, R, A, \bar{\bar{\bar{\text{-}}}}, \leq)$ admits a unique $<\text{-ideal extension}$.  

**Proof.** From Proposition 6 we know that $(L, R, A, \bar{\bar{\bar{\text{-}}}}, \leq)$ admits $<\text{-preferred extensions}$, so let $S$ be their intersection. If $S = \emptyset$, then it is $<\text{-admissible}$, and so an $<\text{-ideal extension}$ (unique). If $S \neq \emptyset$ it is $<\text{-admissible}$, then it is an $<\text{-ideal extension}$ (unique as well). Else, suppose $S \neq \emptyset$ is not $<\text{-admissible}$. Then its $\subseteq$-maximally $<\text{-admissible subsets}$ $I \subseteq S$ are $<\text{-ideal extensions}$ of $(L, R, A, \bar{\bar{\bar{\text{-}}}}, \leq)$. Suppose $I$ and $I'$ are two distinct $<\text{-admissible subsets}$ of $S$. Then their union $I \cup I'$ is a subset of $S$ too, and so $<\text{-conflict-free}$. By Lemma 3, $I \cup I$ $<\text{-defends}$ its assumptions, so must be $<\text{-admissible}$. Consequently, there can be only one $<\text{-maximally}$ $<\text{-admissible subset}$ of $S$, i.e. $(L, R, A, \bar{\bar{\bar{\text{-}}}}, \leq)$ has a unique $<\text{-ideal extension}$. \hfill \square  

**Proposition 11.** Any $<\text{-ideal extension}$ of $(L, R, A, \bar{\bar{\bar{\text{-}}}}, \leq)$ is a $<\text{-complete extension}$ too.  

**Proof.** By Proposition 10, it has a unique $<\text{-ideal extension}$ $I$. Suppose for a contradiction that $I$ is not $<\text{-complete}$. Then there is $\alpha \in A \setminus I$ $<\text{-defended}$ by $I$. Such $\alpha$ must be contained in the intersection $S$ of $<\text{-preferred extensions}$ of $(L, R, A, \bar{\bar{\bar{\text{-}}}}, \leq)$, because $I \subseteq S$ $<\text{-defends}$ any $\alpha$ and every $<\text{-preferred extension}$ $E$ of $(L, R, A, \bar{\bar{\bar{\text{-}}}}, \leq)$ is $<\text{-complete}$ (by Proposition 7). But then, $I \cup \{\alpha\}$ is $<\text{-admissible}$, according to Lemma 3, so that $I$ is not $<\text{-ideal}$---a contradiction. Therefore, $I$ must be $<\text{-complete}$. \hfill \square  

These properties that $ABA^+$ exhibits in the presence of Contraposition will be used to show, in the coming sections, that $ABA^+$ satisfies certain principles of preference handling and non-monotonic reasoning.

## 5 Preference Handling Properties

Referring to (Amgoud and Vesic 2009), in (Brewka, Truszczynski, and Woltran 2010) the authors hinted at two (arguably) desirable properties of argumentation formalisms dealing with preferences, that concern conflict preservation and the absence of preferences. In the next two subsections we indicate that $ABA^+$ satisfies those properties, and in the following subsections show that (arguably) desirable properties of preference handling are too satisfied by $ABA^+$.  

### 5.1 Conflict Preservation

The first property insists that extensions returned after accounting for preferences should be conflict-free with respect to attack relation not taking into account preferences. We formulate it as a principle applicable to $ABA^+$ as follows.

**Definition 14.** $(L, R, A, \bar{\bar{\bar{\text{-}}}}, \leq)$ fulfills the Principle of Conflict Preservation for $<\sigma$ semantics if for all $<\sigma$ extensions $E \subseteq A$ of $(L, R, A, \bar{\bar{\bar{\text{-}}}}, \leq)$, for any $\alpha, \beta \in A$, $\{\alpha\} \rightarrow \{\beta\}$ implies that either $\alpha \notin E$ or $\beta \notin E$.  

In (Cyras and Toni 2016) it was shown that Lemma 2 guarantees the following result.

**Proposition 12.** $E \subseteq A$ is conflict-free in $(L, R, A, -)$ iff $E$ is $<\text{-conflict-free}$ in $(L, R, A, \bar{\bar{\bar{\text{-}}}}, \leq)$.  

Consequently, $ABA^+$ ensures conflict preservation:

**Proposition 13.** $(L, R, A, \bar{\bar{\bar{\text{-}}}}, \leq)$ fulfills the Principle of Conflict Preservation for any semantics $<\sigma$.  

**Proof.** Let $E$ be a $<\sigma$ extension of $(L, R, A, \bar{\bar{\bar{\text{-}}}}, \leq)$. If $\alpha, \beta \in E$ and $\{\alpha\} \rightarrow \{\beta\}$, then $\{\alpha, \beta\}$ is not conflict-free, and hence not $<\sigma$-conflict-free, by Proposition 12. But then $E$ is not $<\sigma$-conflict-free either, which is a contradiction. Thus, either one of $\alpha$ and $\beta$ does not belong to $E$. \hfill \square  

### 5.2 Empty Preferences

The second property insists that if there are no preferences, then the extensions returned using a preference handling mechanism should be the same as those obtained without accounting for preferences. We formulate it as a principle applicable to $ABA^+$ as follows.

**Definition 15.** Suppose that the preference relation $\leq$ in $(L, R, A, \bar{\bar{\bar{\text{-}}}}, \leq)$ is the strict empty ordering $\emptyset$. Then $(L, R, A, \emptyset, \emptyset)$ fulfills the Principle of Empty Preferences for $\emptyset$-sigma semantics if for all $\emptyset$-sigma extensions $E \subseteq A$ of $(L, R, A, \emptyset, \emptyset)$, $E$ is a $\sigma$-extension of $(L, R, A, -)$.  

In (Cyras and Toni 2016) the following result was shown to hold.

**Theorem 14.** $E \subseteq A$ is a $\sigma$-extension of $(L, R, A, -)$ iff $E$ is an $\emptyset$-$\sigma$-extension of $(L, R, A, \emptyset)$.  

This theorem, in addition to saying that $ABA^+$ is a conservative extension of $ABA$, immediately yields the satisfaction of the principle in question:

**Proposition 15.** $(L, R, A, \emptyset, \emptyset)$ fulfills the Principle of Empty Preferences for any semantics $\emptyset$-$\sigma$.  

### 5.3 Maximal Elements

(Amgoud and Vesic 2014) proposed a property concerning inclusion in extensions of the ’strongest’ arguments, i.e. arguments that are maximal w.r.t. preference ordering. We next reformulate the property to be applicable to $ABA^+$.

**Definition 16.** Suppose the preference ordering $\leq$ of $(L, R, A, \bar{\bar{\bar{\text{-}}}}, \leq)$ is total and further assume that the set $M = \{\alpha \in A : \exists \beta \in A$ with $\alpha < \beta\}$ is $<\text{-conflict-free}$. Then $(L, R, A, \bar{\bar{\bar{\text{-}}}}, \leq)$ fulfills the Principle of Maximal Elements for $<\sigma$ semantics if for all $<\sigma$ extensions $E \subseteq A$ of $(L, R, A, \bar{\bar{\bar{\text{-}}}}, \leq)$, it holds that $M \subseteq E$.  

As an illustration, in Example 8, $\alpha$ is a unique $\leq$-maximal element in $A$, and $\{\alpha\}$ is a unique $<\sigma$-extension of $(L, R, A, \bar{\bar{\bar{\text{-}}}}, \leq)$, whence $(L, R, A, \bar{\bar{\bar{\text{-}}}}, \leq)$ fulfills the Principle of Maximal Elements for any semantics $<\sigma$.  

Our next result shows that in general, $ABA^+$ satisfies this principle under $<\text{-stable}$ and $<\text{-complete}$ semantics.

**Proposition 16.** $(L, R, A, \bar{\bar{\bar{\text{-}}}}, \leq)$ fulfills the Principle of Maximal Elements for $<\text{-stable}$ and $<\text{-complete}$ semantics.  

**Proof.** Let the preference ordering $\leq$ of $(L, R, A, \bar{\bar{\bar{\text{-}}}}, \leq)$ be total and suppose $M = \{\alpha \in A : \exists \beta \in A$ with $\alpha < \beta\}$ is $<\text{-conflict-free}$. We first show that $M$ is not $<\text{-attacked}$.  

Fix $\alpha \in M$ and suppose for a contradiction that for some $S \subseteq A$ it holds that $S <\text{-}\{\alpha\}$. So either (i) $\exists B +\bar{\bar{\bar{\text{R}}}} \bar{\bar{\bar{\text{I}}}}$ with
$B \subseteq S$ and $\forall \beta \in B$ $\alpha \leq \beta$ or $\beta \not\leq \alpha$, or
(ii) $\{\alpha\} \vdash^R \beta$ for some $\beta \in S$ with $\alpha < \beta$. Note that the case (ii) cannot
happen, because $\alpha$ is $\leq$-maximal. So consider case (i). Since
$\leq$ is total, it follows that $\alpha \leq \beta \forall \beta \in B$. But as $\alpha$ is $\leq$-
maximal, it must also hold that $\beta \leq \alpha$, for any $\beta \in B$. From
here, we show $B \subseteq M$. Indeed, fix $\beta \in B$ and assume for
a contradiction that $\beta \notin M$. Then $\exists \gamma \in A$ such that $\beta < \gamma$.
By transitivity, $\alpha < \gamma$, contradicting $\alpha$'s $\leq$-maximality. So
we must have $\beta \in M$, and consequently, $B \subseteq M$.

But now, since $\alpha \in M$, $B \subseteq M$ and $B \not\sim < \{\alpha\}$, this
contradicts $\leq$-conflict-freeness of $M$. Therefore, by contra-
diction, $S \not\sim < \{\alpha\}$, for any $S \subseteq A$. Since $\alpha \in M$ was
arbitrary, we have $M$ $<$-unattacked, as required.

If $(L, R, A, -)$ admits no $<$-stable or $<$-complete ex-
tensions, then the principle is fulfilled trivially. Otherwise,
let $E \subseteq A$ be $<$-stable in $(L, R, A, -)$. Pick $\alpha \in M$ and
suppose for a contradiction that $\alpha \notin E$. Then $E \not\sim < \{\alpha\}$,
which is a contradiction. Thus, $\alpha \in S$, and hence $M \subseteq S$.

Now let $E$ be a $<$-complete extension of $(L, R, A, -)$
and suppose for a contradiction $M \notin E$. Then $E$ does not
$<$-defend some $\alpha \in M$. This means that $S \not\sim < M$ for some
$S \subseteq A$, which is a contradiction. Hence, $M \subseteq E$. $\Box$

This principle may, however, be violated under, say,$<$-
preferred semantics: in Example 12, the framework
$(L, R, A, -)$ to begin with, admits $\{\alpha, \beta\}$ as a $<$-
preferred extension, while $\gamma \notin \{\alpha, \beta\}$ is a $\leq$-
maximal element. However, assuming Contraction, the Principle of
Maximal Elements is satisfied under the remaining seman-
tics too.

**Corollary 17.** If $(L, R, A, -)$ satisfies the Axiom of Con-
traction, then it fulfills the Principle of Maximal Ele-
ments for $<$-preferred/$<$-ideal/$<$-grounded semantics.

**Proof.** Follows from Propositions 4, 7, 11 and 16. $\Box$

### 5.4 Principle I

(Brewka and Eiter 2000) formulated a principle for sound
extension-based default reasoning with preferences, which
we reformulate for ABA$^+$ next.

**Definition 17.** $(L, R, A, -)$ fulfills Principle I for
$<\sigma$ semantics if for all $E, E' \subseteq A$ such that $E = E_0 \cup \{\alpha\}$
and $E' = E_0 \cup \{\alpha'\}$ for some $E_0 \subseteq A$, with $\alpha, \alpha' \notin E_0$ and
$\alpha < \alpha'$, it holds that if $E$ is a $<\sigma$ exten-
sion of $(L, R, A, -)$, then $E'$ is not a $<\sigma$ extension of
$(L, R, A, -)$.

This principle insists that if two coherent viewpoints of a
situation differ only in that each of them contains a single as-
sumption not contained in the other, then the viewpoint with
the more preferred assumption should be chosen. ABA$^+$ satis-
fies this principle under $<$-stable semantics.

**Proposition 18.** $(L, R, A, -)$ fulfills Principle I for
$<\sigma$ semantics.

**Proof.** Suppose for a contradiction that both $E = E_0 \cup \{\alpha\}$
and $E' = E_0 \cup \{\alpha'\}$, where $\alpha' < \alpha$, are $<\sigma$ extensions
of $(L, R, A, -)$, $E'$ is $<\sigma$-stable and $\alpha \notin E'$, then
$E' \not\sim < \{\alpha\}$. As $E$ is $<\sigma$-conflict-free, we find $E_0 \not\sim < \{\alpha\}$.

so (from $E' \not\sim < \{\alpha\}$ we get that): (i) either there is $E'' \cup \{\alpha\} \vdash^R \beta$ with
$E'' \subseteq E_0$ and $\epsilon \notin \alpha \forall \epsilon \in E'' \cup \{\alpha\}$;
(ii) or $\{\alpha\} \vdash^R \beta$ is such that $\alpha < \alpha'$. As $\alpha' < \alpha$, both
cases lead to a contradiction, so that $E'$ is not a $<\sigma$-stable
extension, provided $E$ is.

In Example 8, $E = \{\alpha\}$ is a unique $<\sigma$-stable extension
of $(L, R, A, -)$, which illustrates the principle as follows: take
$E_0 = \emptyset$ so that $E = \{\alpha\}$ and $E' = \{\beta\}$, where $\beta < \alpha$.
It is important that Principle I is satisfied under $<$-stable seman-
tics, because (Brewka and Eiter 1999) investigated (pre-
ferred) answer sets of logic programs, and answer sets in
Logic Programming correspond to stable extensions in ABA
(Bondarenko et al. 1997). Satisfaction of the principle gives
hope that preferred answer set semantics can be captured in
ABA$^+$, as answer set semantics is captured in ABA.

Principle I, however, may be violated under $<$-preferred seman-
tics: in Example 12, $(L, R, A, -)$ has two $<\sigma$-
preferred extensions $\{\alpha, \beta\}$ and $\{\beta, \gamma\}$, and yet $\alpha < \gamma$.
Note, though, that $(L, R, A, -)$ satisfies the Axiom of Contra-
position and has a unique $<\sigma$ extension $\{\beta, \gamma\}$, and
thus fulfills Principle I for any semantics $<\sigma$. Based on our
investigations, we conjecture that assuming Contraction,
ABA$^+$ frameworks fulfill the principle for the remaining
semantics as well. Verifying this is left as future work.

### 6 Non-Monotonic Reasoning Properties

(Cyras and Toni 2015) proposed and studied the well
known non-monotonic inference properties of
Cautious Monotonicity (MON henceforth) and Cumulative Transitivity
(CUT henceforth) for ABA. Here, we investigate some
of those properties for ABA$^+$. We first recall (some of) the
properties considered and results obtained.

Assume as given a fixed, but otherwise arbitrary (flat)
ABA framework $F = (L, R, A, -)$. Let $E$ be a $\sigma$ ex-
ension of $F$. In what follows, $E'$ will denote a $\sigma$ extension of
a newly constructed ABA framework $F'$. To avoid trivialities,
we consider cases only where each of $F$ and $F'$ has at least
one $\sigma$ extension—$E$ and $E'$ respectively.

We first recall the **STRICT setting** regarding strength-
ing of information. Given $\psi \in Cn(E) \setminus A$, define
$F' = (L, R \cup \{\psi \leftarrow \top\}, A, -)$. There are four properties:

**SCEPTICAL STRICT CUT:**

For all extensions $E'$ of $F'$ we have $Cn(E') \subseteq Cn(E)$;

**CREDULOUS STRICT CUT:**

There is an extension $E'$ of $F'$ with $Cn(E') \subseteq Cn(E)$;

**SCEPTICAL STRICT MON:**

For all extensions $E'$ of $F'$ we have $Cn(E) \subseteq Cn(E')$;

**CREDULOUS STRICT MON:**

There is an extension $E'$ of $F'$ with $Cn(E) \subseteq Cn(E')$.

Table 1 summarizes results pertaining to ABA (sceptical
and credulous versions coincide under grounded and ideal

---

1 In (Cyras and Toni 2015), instead of sceptical/credulous (see
below) the words strong/weak were used, respectively; we have
altered the names to adhere to the more common terminology.
semantics, and for other semantics the status of the credulous property is indicated in parentheses).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>STRICT CUT</td>
<td>✓</td>
<td>✓</td>
<td>✓ (✓)</td>
<td>✓ (✓)</td>
<td>✓ (✓)</td>
</tr>
<tr>
<td>STRICT MON</td>
<td>✓</td>
<td>✓</td>
<td>✓ (✓)</td>
<td>✓ (✓)</td>
<td>✓ (✓)</td>
</tr>
</tbody>
</table>

Table 1: STRICT CUT/MON for standard ABA

We now recall the ASM setting, where conclusions that are themselves assumptions are being confirmed. Given \( \psi \notin Cn(E) \cap A \), define \( F^3 = (L, R \cup \{ \psi \leftarrow T \}, A \setminus \{ \psi \}, \preceq) \).

The properties are as follows:

SCEPTICAL ASM CUT:
For all extensions \( E' \) of \( F^3 \) we have \( Cn(E') \subseteq Cn(E) \);

CREDULOUS ASM CUT:
There is an extension \( E' \) of \( F^3 \) with \( Cn(E') \subseteq Cn(E) \);

SCEPTICAL ASM MON:
For all extensions \( E' \) of \( F^3 \) we have \( Cn(E) \subseteq Cn(E') \);

CREDULOUS ASM MON:
There is an extension \( E' \) of \( F^3 \) with \( Cn(E) \subseteq Cn(E') \).

Table 2 summarizes results regarding ABA in the ASM setting (notation as before).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ASM CUT</td>
<td>✓</td>
<td>✓</td>
<td>✓ (✓)</td>
<td>✓ (✓)</td>
<td>✓ (✓)</td>
</tr>
<tr>
<td>ASM MON</td>
<td>✓</td>
<td>✓</td>
<td>✓ (✓)</td>
<td>✓ (✓)</td>
<td>✓ (✓)</td>
</tr>
</tbody>
</table>

Table 2: ASM CUT/MON for standard ABA

The non-monotonic inference properties CUT and MON can be readily applied to ABA\(^+\). Take \( F \) to be an ABA\(^+\) framework \( (L, R, A, \preceq, \leq) \), let \( E \) be its \(<\sigma\)-extension, and given \( \psi \notin Cn(E) \), define \( F' \) as follows:

- **strict setting**: \( F' = (L, R \cup \{ \psi \leftarrow T \}, A, \preceq, \leq) \);
- **ASM setting**: \( F' = (L, R \cup \{ \psi \leftarrow T \}, A \setminus \{ \psi \}, \preceq, \leq) \), where \( \leq \) is a restriction of \( \preceq \) to \( A \setminus \{ \psi \} \).

We can then analyze whether the non-monotonic inference properties in question are satisfied in ABA\(^+\). Trivially, as ABA\(^+\) is a conservative extension of ABA (cf. Theorem 14), properties violated in ABA will remain violated in ABA\(^+\). Therefore, we will focus on those that are satisfied in ABA; in particular, the credulous versions except for MON under ideal semantics.

Example 18. As an illustration of the properties, recall Example 12. The ABA\(^+\) framework \( F = (L, R', A, \preceq, \leq) \) (that satisfies the Axiom of Contraposition) has a unique \(<\sigma\)-extension \( \{ \beta, \gamma \} \) with \( Cn(\{ \beta, \gamma \}) = \{ \pi, \beta, \gamma \} \).

For brevity reasons, the same symbol "inverse" is used for both contrary mappings, and in the new framework \( F' \), the contrary mapping is implicitly restricted to a diminished set of assumptions.

- **strict setting**: take \( \pi \) and let \( F' = (L, R \cup \{ \pi \leftarrow T \}, A, \preceq, \leq) \). Then \( F' \) has a unique \(<\sigma\)-extension \( \{ \beta, \gamma \} \).
- **ASM setting**: take \( \beta \) and let \( F' = (L, R \cup \{ \beta \leftarrow T \}, A \setminus \{ \beta \}, \preceq, \leq) \) with \( \alpha \prec \beta \). Then \( F' \) likewise has a unique \(<\sigma\)-extension \( \{ \beta, \gamma \} \).

As conclusions of extensions of both \( F \) and \( F' \) are actually the same, the credulous versions of the properties are indeed satisfied in both settings.

In what follows, we assume that \( (L, R, A, \preceq, \leq) \) satisfies the Axiom of Contraposition and show that ABA\(^+\) retains the same satisfaction results of CUT and MON from ABA in both STRICT and ASM settings.

Proposition 19. \(<\sigma\)-complete semantics satisfies CREDULOUS STRICT and CREDULOUS STRICT MON.

Proof. Let \( E \) be a \(<\sigma\)-complete extension of \( F = (L, R, A, \preceq, \leq) \), fix \( \psi \in Cn(E) \setminus A \), and let \( F' = (L, R \cup \{ \psi \leftarrow T \}, A, \preceq, \leq) \). For ease of reference, we will denote by \( \leadsto \) and \( \leadsto' \) the \(<\sigma\)-attack relations in \( F \) and \( F' \) respectively. We claim that \( E \) is a \(<\sigma\)-extension of \( F \) too.

First, \( E \) is clearly \(<\sigma\)-conflict-free in \( F \). Second, let \( \alpha \in E \) and suppose that \( B' \leadsto' \{ \alpha \} \), for some \( B' \subseteq A \) with \( B' \cap E \). There are two possibilities:

1. \( \alpha \in E \), as we split into cases.

2. \( \alpha \not\in E \), take \( \varepsilon < \alpha \), minimal such that \( En \setminus \{ \varepsilon \} \) is an \(<\sigma\)-extension of \( F \) and \( E \). We have \( E \setminus En \subseteq \{ \varepsilon \} \setminus \{ \alpha \} \). Hence, we are back in the first case above.

In any case, \( E \leadsto B \). Possibility 2: the \(<\sigma\)-attack \( B' \leadsto' \{ \alpha \} \) does not involve the rule \( \psi \leftarrow T \). That is, we actually have \( B' \leadsto' \{ \alpha \} \). Then, \( E \leadsto B \), hence \( E \leadsto' B' \).

In any event, \( E \leadsto B \). For any \( \alpha \in E \) arbitrary, we conclude that \( E \leadsto B \).

It now suffices to show that \( E \) contains every assumption it \(<\sigma\)-defends in \( F \). To this end, suppose \( E \leadsto B \) in \( F \), and suppose for a contradiction that \( \alpha \notin E \). Then \( E \) does not \(<\sigma\)-defend \( \alpha \) in \( F \). That is, there is \( B \leadsto \{ \alpha \} \) such that \( E \not\leadsto B \). But now, we also have \( B \leadsto' \{ \alpha \} \), so that \( E \leadsto' B \), whence it must be that \( E \leadsto' B \) is a normal attack that does not use some assumption \( \varepsilon \in E \) (which is used to deduce \( \psi \), i.e. \( En \setminus \{ \varepsilon \} \subseteq E \)).
\( R_0 \subseteq R \) such that \( \varepsilon < \beta \) for some \( \beta \in B \). Taking \( \leq \)-minimal such \( \varepsilon \) (and accordingly some \( \beta \) in \( B \)), the Axiom of Contraposition guarantees that \( (E \setminus \{ \varepsilon \}) \cup \{ \beta \} \sim < \{ \varepsilon \} \) via normal attack, and since \( \varepsilon \in E \), it must be that \( E \sim < \{ \beta \} \), giving \( E \sim < \beta \), which is a contradiction. Hence, \( \alpha \in E \) after all, and so \( E \) is \( < \)-complete in \( F' \), as required.

**Proposition 20.** \(<\)-preferred semantics satisfies CREDOUS STRICT CUT and CREDOUS STRICT MON.

**Proof.** Like in the proof of Proposition 19, we claim that a \(<\)-preferred extension \( E \) of \( F \) is a \(<\)-preferred extension of \( F' \). Indeed, if \( E \) were not \( \leq \)-maximally \(<\)-admissible in \( F' \), then for some \( \beta \in A \setminus E \), \( E \cup \{ \beta \} \) would be \(<\)-admissible in \( F' \). Verbatim to the proof for \(<\)-complete semantics, we could show that \( E \cup \{ \beta \} \) is \(<\)-admissible in \( F \), contradicting \( E \) being \(<\)-preferred in \( F \).

**Proposition 21.** \(<\)-stable semantics satisfies CREDOUS STRICT CUT and CREDOUS STRICT MON.

**Proof.** Like in the proof of Proposition 19, we claim that a \(<\)-stable extension \( E \) of \( F \) is a \(<\)-stable extension of \( F' \). Indeed, let \( \beta \notin E \). Then \( E \sim \prec \{ \beta \} \). Whether it is a normal or reverse attack, we clearly have \( E \sim \prec \{ \beta \} \) too. Hence, \( E \) is \(<\)-stable in \( F' \), provided \( E \) is \(<\)-stable in \( F \).

**Proposition 22.** \(<\)-grounded semantics satisfies CREDOUS STRICT CUT and CREDOUS STRICT MON.

**Sketch.** Using the argument as in Proposition 19, it can be proven by induction on the construction of the \(<\)-grounded extension \( G \) of \( F \) (cf. Proposition 4) that \( G \) is the \(<\)-grounded extension of \( F' \).

**Corollary 23.** \(<\)-ideal semantics satisfies CREDOUS STRICT CUT.

**Proof.** This follows by definition of the \(<\)-ideal extension and Proposition 20.

Now, in the ASM setting, the same results can be obtained as in the STRICT setting, with essentially the same proofs.

**Proposition 24.** \(<\)-complete/\(<\)-preferred/\(<\)-stable/\(<\)-grounded semantics satisfies CREDOUS ASM CUT and CREDOUS ASM MON, and \(<\>-ideal semantics satisfies CREDOUS ASM CUT.

**Sketch.** Let \( E \) be a \(<\)-complete extension of \( F = (\mathcal{L}, R, A, \leq, \preceq) \), fix \( \psi \in CN(E) \cap A \), and let \( F' = (\mathcal{L}, R \cup \{ \psi \}, \preceq', A \setminus \{ \psi \}, \sim', \leq') \), where \( \preceq' \) is a restriction of \( \leq \) to \( A \setminus \{ \psi \} \). It can be shown that \( E \) is a \(<\)-complete extension of \( F' \), by replacing, in the proof of Proposition 19, \( E_0 \) and \( E_0 \vdash R_0 \psi \) with \( \{ \psi \} \) and \( \{ \psi \} \vdash \psi \) respectively. Other claims follow the same line of reasoning as for the proofs in the STRICT setting.

Table 3 summarizes this section’s results (sceptical and credulous versions coincide under \(<\)-grounded and \(<\>-ideal semantics; for other semantics the credulous version is indicated in parentheses.)

<table>
<thead>
<tr>
<th>Property</th>
<th>(&lt;)-g.</th>
<th>(&lt;&gt;-id.</th>
<th>(&lt;)-stb.</th>
<th>(&lt;&gt;-pr.</th>
<th>(&lt;&gt;-cpl.</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRICT and ASM CUT</td>
<td>✓</td>
<td>✓</td>
<td>X((\psi))</td>
<td>X((\psi))</td>
<td>X((\psi))</td>
</tr>
<tr>
<td>STRICT and ASM MON</td>
<td>✓</td>
<td>X</td>
<td>X((\psi))</td>
<td>X((\psi))</td>
<td>X((\psi))</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>(&lt;)-g.</th>
<th>(&lt;&gt;-id.</th>
<th>(&lt;&gt;-stb.</th>
<th>(&lt;&gt;-pr.</th>
<th>(&lt;&gt;-cpl.</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRICT and ASM CUT</td>
<td>✓</td>
<td>✓</td>
<td>X((\psi))</td>
<td>X((\psi))</td>
<td>X((\psi))</td>
</tr>
<tr>
<td>STRICT and ASM MON</td>
<td>✓</td>
<td>X</td>
<td>X((\psi))</td>
<td>X((\psi))</td>
<td>X((\psi))</td>
</tr>
</tbody>
</table>

7 Related and Future Work

The principle of Contraposition of (strict) rules (see e.g. (Caminada and Amgoud 2007; Modgil and Prakken 2013)) is notably employed in the well studied structured argumentation formalism ASPIC\(^+\) (Modgil and Prakken 2013; 2014). The principle as such is also inherently present in classical logic-based approaches to structured argumentation such as (Gorogiannis and Hunter 2011; Besnard and Hunter 2014). Similarly as in ASPIC\(^+\), ABA\(^+\) utilizes Contraposition to ensure the Fundamental Lemma (cf. Lemma 3). As a consequence, Contraposition paves way to satisfaction of desirable properties of ABA\(^+\) semantics, as well as preference handling and non-monotonic inference properties discussed in Sections 5 and 6. Whether the Axiom of Contraposition can be relaxed for ABA\(^+\) to obtain the same results is a line of future research.

The preference handling principle discussed in Section 5.4 was originally proposed, along with some other properties, by (Brewka and Eiter 1999) for answer set programming (ASP) with preferences. To the best of our knowledge, reformulation of Principle I for ABA\(^+\) is the first application of this principle to argumentation with preferences. Building on (Brewka and Eiter 1999), (ˇSimko 2014) discussed an extended set of principles for ASP with preferences, most of which focus on preferences over rules. Whether those principles can be applied to ABA\(^+\) is a future work direction.

Regarding preference handling in argumentation, along with the Principle of Maximal Elements discussed in Section 5.3, (Amgoud and Vesimal 2014) suggested several arguably desirable properties of argumentation with preferences. Those properties are exhibited in ABA\(^+\) as Proposition 12 and Theorem 14. Referring to those properties, (Brewka, Truszczyński, and Woltran 2010) and (Amgoud and Vesimal 2014) also hinted at other properties regarding selection among extensions, as possible principles of preference handling in argumentation.

Relating those principles to ABA\(^+\) is left for future work.

In terms of non-monotonic reasoning properties, Cautious Monotonicity and Cumulative Transitivity (studied in Section 6) are traced to (Makinson 1988; Kraus, Lehmann, and Magidor 1990) and fall into the well studied area of analysing non-monotonic reasoning with respect to information change (cf. (Rott 2001)). In argumentation setting, the latter is also known as argumentation dynamics, and has recently been a topic of interest in the argumentation community (see e.g. (Cayrol, de Saint-Cyr, and Lagasquie-Schiex 2010; Falappa et al. 2011; Baroni et al. 2014; Coste-Marquis et al. 2014; Booth et al. 2014; Diller et al. 2015; Baumann and Brewka 2015)). In particular, non-monotonic inference properties were investigated in (Hunter 2010) with
respect to argument–claim entailment in logic-based argumentation systems; in (ˇCyras and Toni 2015) for ABA; and with regards to ASPIC+–type-of argumentation systems in (Dung 2016). Only the latter of the three works concerns argumentation with preferences. In addition to considering different structured argumentation setting and different preference handling mechanisms, it diverges from our analysis in Section 6 in that (Dung 2016) regards Cumulative Transitivity plus Cautious Monotonicity as a single property of Cumulativity and studies it only for stable and complete semantics. Other argumentation-related properties from (Dung 2016) will be studied for ABA+ in the future.

Several other topics of interest are left for future work. For instance, integrating dynamic preferences (see e.g. (Prakken and Sartor 1999; Zhang and Foo 1997; Brewka and Woltran 2010)) within ABA+ and studying their interaction with the properties of preference handling as well as of non-monotonic inference. Also, relation of ABA+ to Logic Programming with preferences (e.g. (Sakama and Inoue 1996; Zhang and Foo 1997; Brewka and Eiter 1999)) and non-monotonic reasoning formalisms equipped with preferences in general (e.g. (Brewka 1989; Baader and Hollunder 1995; Rintanen 1998; Brewka and Eiter 2000; Delgrande and Schaub 2000; Stolzenburg et al. 2003; Kakas and Moraitis 2003)) is left for future research.

There are as well numerous approaches to integrating reasoning with preferences within argumentation, e.g. (Amgoud and Cayrol 2002; Bench-Capon 2003; Kaci and van der Torre 2008; Modgil 2009; Modgil and Prakken 2010; Baroni et al. 2011; Dunne et al. 2011; Brewka et al. 2013; Amgoud and Vesci 2014; Besnard and Hunter 2014; García and Simari 2014; Wakaki 2014; Modgil and Prakken 2013; 2014; Brewka and Woltran 2016). It would be interesting to study these formalisms with respect to the properties considered in this paper, where it has not already been done. We leave this as future work.

8 Conclusions

We investigated various properties of a recently proposed non-monotonic reasoning formalism ABA+ (ˇCyras and Toni 2016) that deals with preferences in structured argumentation. In particular, we first established that assuming the principle of Contraposition (see e.g. (Modgil and Prakken 2013)), ABA+ semantics exhibit desirable properties akin to those of other existing argumentation formalisms, such as (Dung 1995). We then showed that ABA+ satisfies some (arguably) desirable principles of preference handling in argumentation and non-monotonic reasoning, e.g. (Brewka and Eiter 1999). Finally, we analysed non-monotonic inference properties (as in (ˇCyras and Toni 2015)) of ABA+ under various semantics. We believe our work contributes to the understanding of preferences within argumentation in particular, and in non-monotonic reasoning at large.

References


Čyras, K., and Toni, F. 2016. ABA+: Assumption-Based Argumentation with Preferences. In KR.


Equational properties of stratified least fixed points

Zoltan Esik
Dept. of Computer Science
University of Szeged
Hungary

Studies on Brutal Contraction and Severe Withdrawal: Preliminary Report

Marco Garapa
Universidade da Madeira
CIMA - Centro de Investigação em Matemática e Aplicações*
marco@uma.pt

Eduardo Fermé
Universidade da Madeira
NOVA Laboratory for Computer Science and Informatics (NOVA LINCS)†
ferme@uma.pt

Maurício D. L. Reis
Universidade da Madeira
CIMA - Centro de Investigação em Matemática e Aplicações*
m_reis@uma.pt

Abstract
In this paper we study the class of brutal base contractions that are based on a bounded ensconcement and also the class of severe withdrawals which are based on bounded epistemic entrenchment relations that are defined by means of bounded ensconcements (using the procedure proposed by Mary-Anne Williams). We present axiomatic characterizations for each one of those classes of functions and investigate the interrelation among them.

1 Introduction
The central goal underlying the research area of logic of theory change is the study of the changes which can occur in the belief state of a rational agent when he receives new information.

The most well known model of theory change was proposed by Alchourrón, Gärdenfors, and Makinson (1985) and is, nowadays, known as the AGM model. Assuming that the belief state of an agent is modelled by a belief set (i.e. a logically closed set of sentences), this framework essentially provides a definition for contractions — i.e. functions that receive a sentence (representing the new information received by the agent), and return a belief set which is a subset of the original one that does not contain the received sentence. In the mentioned paper, the class of partial meet contractions was introduced and axiomatically characterized. Subsequently several constructive models have been presented for the class of function contractions proposed in the AGM framework (such as the system of spheres-based contractions (Grove 1988), safe/kernel contractions (Alchourrón and Makinson 1985; Hansson 1994), and the epistemic entrenchment-based contractions (Gärdenfors 1988; Gärdenfors and Makinson 1988)). Also several adaptations and variations of those constructive models have been presented and studied in the literature as it is the case, for example, of severe withdrawals (or mild contractions or Rott's contractions) (Rott 1991; Rott and Pagnucco 1999) which results of simplifying the definition of epistemic entrenchment-based contractions.

Although the AGM model has quickly acquired the status of standard model of theory change, several researchers (for an overview see (Fermé and Hansson 2011)) have pointed out its inadequateness in several contexts and proposed several extensions and generalizations to that framework. One of the most relevant of the proposed extensions of the AGM model of contraction is to use sets of sentences not (necessarily) closed under logical consequence — which are designated belief sets — rather than belief sets to represent belief states.

Hence, several of the existing models (of AGM contractions) were generalized to the case when belief states are represented by belief bases instead of belief sets. Among those we emphasize the ensconcement-based contractions and the brutal base contractions (of belief bases) proposed in (Williams 1995), which can be seen as adaptations to the case of belief bases of the epistemic entropy-based contractions and of the severe withdrawals, respectively. In fact, the definitions of those operations are both based on the concept of ensconcement, which is an adaptation of the concept of epistemic entrenchment relation to the case of belief bases. In the mentioned paper Mary-Anne Williams has also presented a method for constructing an epistemic entrenchment from an ensconcement relation.

In the present paper we will study the interrelation among brutal base contractions (of belief bases) and severe withdrawals (of belief sets). More precisely, we will devote special attention to the class of brutal base contractions which are based on bounded ensconcements — the so-called bounded brutal base contractions — and also to the class of the so-called ensconcement-based severe withdrawals, which is formed by the severe withdrawals that are based on an epistemic entrenchment relation defined from a bounded ensconcement using Mary-Anne William’s method. We shall provide axiomatic characterizations to each one of those classes of functions and study the interrelation among them.

This paper is organized as follows: Firstly we provide the notation and background needed for the rest of the paper. After that we provide axiomatic characterizations for the classes of bounded brutal base contractions and of ensconcement-based severe withdrawals. Furthermore we show how to define a bounded brutal base contraction from an ensconcement-based severe withdrawal and vice-versa.
Finally, we briefly summarize the main contributions of the paper. In the appendix we provide proofs for the theorems. Proofs for all the remaining results are available at http://www.cee.una.pl/ferme/GFR16-full.pdf.

2 Background

2.1 Formal preliminaries

We will assume a language \( \mathcal{L} \) that is closed under truth-functional operations and a consequence operator \( Cn \) for \( \mathcal{L} \). \( Cn \) satisfies the standard Tarskian properties, namely inclusion (\( A \subseteq Cn(A) \)), monotony (if \( A \subseteq B \), then \( Cn(A) \subseteq Cn(B) \)), and iteration (\( Cn(A) = Cn(Cn(A)) \)). It is supraclassical and compact, and satisfies deduction (if \( \beta \in Cn(A \cup \{ \alpha \}) \), then \( (\alpha \rightarrow \beta) \in Cn(A) \)). \( A \vdash \alpha \) will be used as an alternative notation for \( \alpha \in Cn(A) \), \( A \vdash \alpha \) for \( \alpha \in Cn(\emptyset) \) and \( Cn(\{ \alpha \}) \) for \( Cn(\{ \alpha \}) \). Upper-case letters denote subsets of \( \mathcal{L} \). Lower-case Greek letters denote elements of \( \mathcal{L} \).

A well-ranked preorder on a set \( X \) is a preorder such that every nonempty subset of \( X \) has a minimal member, and similarly an inversely well-ranked preorder on a set \( X \) is a preorder such that every nonempty subset of \( X \) has a maximal member. A total preorder on \( X \) is bounded if and only if it is both well-ranked and inversely well-ranked.1

2.2 AGM

The AGM model of belief change was proposed by Alchourrón, Gärdenfors, and Makinson (1985) and acquired the status of standard model of belief change. In this model beliefs are represented by a set of sentences closed under logical consequence. In the AGM framework there are three operations to be considered, namely expansion, contraction and revision. Expansion, consists of adding new information (represented by sentences) in the original set preserving logical closure. Contraction, consists of eliminating sentences from a belief set, in such a way that the remaining set does not imply a specified sentence. Revision, consists in incorporating a sentence in the original set, but (eventually) eliminating some sentences in order to retain consistency of the revised set. AGM has been characterized in at least five, different ways: Postulates, partial meet functions, epistemic entrenchment, safe/kernel contraction and Grove’s sphere-systems (for an overview see (Fermé and Hansson 2011)).

One of the Postulates included in the axiomatic characterization of the contraction operator is recovery:

\[( \text{Recovery} ) \quad K \subseteq (K - \alpha) + \alpha \]

\( \text{Recovery} \) is based in the principle that “it is reasonable to require that we get all of the beliefs [...] back again after first contracting and then expanding with respect to the same belief” (Gärdenfors 1982). Nevertheless, the recovery postulate have been criticized by several authors (Fuhrmann 1991; Hansson 1991; Levi 1991; Niederée 1991) as a general principle that contractions should hold. Alternative contraction models were proposed in which the recovery postulate does not hold, for instance: Levi Contraction (Levi 1991), Severe Withdrawal (Rott 1991; Rott and Pagnucco 1999) and Semi-contraction (Fermé 1998).

2.3 Epistemic Entrenchment

Epistemic entrenchment was introduced in (Gärdenfors 1988; Gärdenfors and Makinson 1988) and relies on the idea that contractions on a belief set \( K \) should be based on an ordering of its sentences according to their epistemic entrenchment. When a belief set \( K \) is contracted it is preferred to give up beliefs with lower entrenchment over others with a higher entrenchment. Gärdenfors proposed the following set of axioms that an epistemic entrenchment order \( \leq \) related to a belief set \( K \) should satisfy:

\[ (\text{EE1}) \quad \text{If } \alpha \leq \beta \text{ and } \beta \leq \gamma, \text{ then } \alpha \leq \gamma \quad (\text{Transitivity}) \]
\[ (\text{EE2}) \quad \text{If } \alpha \vdash \beta, \text{ then } \alpha \leq \beta \quad (\text{Dominance}) \]
\[ (\text{EE3}) \quad (\alpha \wedge \beta) \leq (\alpha \wedge \gamma) \quad (\text{Conjunctiveness}) \]
\[ (\text{EE4}) \quad \text{If } K \not\vdash \bot, \text{ then } \alpha \notin K \text{ if and only if } \alpha \leq \beta \text{ for all } \beta \quad (\text{Minimality}) \]
\[ (\text{EE5}) \quad \text{If } \beta \leq \alpha \text{ for all } \beta, \text{ then } \vdash \alpha \quad (\text{Maximality}) \]

If \( \leq \) is well-ranked and inversely well-ranked, then the epistemic entrenchment is well-ranked and inversely well-ranked, and therefore is a bounded epistemic entrenchment. The relation \( \leq \) of epistemic entrenchment is independent of the change functions in the sense that it does not refer to any contraction or revision function. In addition to stating the axioms of entrenchment, Gärdenfors proposed the following entailment-based contraction functions:

\[ (G \bowtie) \quad \beta \in K - \alpha \text{ if and only if } \beta \in K \text{ and, either } \vdash \alpha \text{ or } \alpha \bowtie (\alpha \wedge \beta) \]

The crucial clause of \((G \bowtie)\) is \( \alpha \bowtie (\alpha \vee \beta) \). This clause can be justified with reference to the recovery postulate (Gärdenfors and Makinson 1988).

\textbf{Severe withdrawal:} Rott (1991) proposed a more intuitive alternative definition, later called Severe withdrawal (or mild contraction or Rott’s contraction) (Rott and Pagnucco 1999):

\[ (R \bowtie) \quad \beta \in K - \alpha \text{ if and only if } \beta \in K \text{ and, either } \vdash \alpha \text{ or } \alpha \bowtie \beta \]

Arló-Costa and Levi (2006) have analyzed it in terms of minimal loss of informational value. It has been shown to satisfy the implausible postulate of expansiveness. (If \( \nolnot \alpha \vdash \alpha \) and \( \nolnot \beta \vdash \beta \), then either \( \alpha \notin K \bowtie \beta \) or \( \beta \notin K \bowtie \alpha \) (Hansson 1999b). Lindström and Rabinowicz (1991) abstained from recommending either a particularly expansive contraction (severe withdrawal) or a particularly retentive one (AGM contraction). They argued that these extremes should be taken as “upper” and “lower” bounds and that any “reasonable” contraction function should be situated.

---

1In (Williams 1994a) a preorder in these conditions is designated by \textit{finite}, however we think it is more adequate to use the denomination \textit{bounded}. 
between them. This condition was called the Lindström’s and Rabinowicz’s interpolation thesis (Rott 1995). Severe withdrawal was axiomatized independently by Rott and Pagnucco (1999) and by Fermé and Rodriguez (1998). The following set of postulates characterize severe withdrawals (Rott and Pagnucco 1999):

1. \( K \div \alpha = Cn(K \div \alpha) \)
2. \( K \div \alpha \subseteq K \)
3. If \( \alpha \not\in K \) or \( \vdash \alpha \), then \( K \subseteq K \div \alpha \)
4. If \( \not\vdash \alpha \), then \( \alpha \not\in K \div \alpha \)
5. If \( Cn(\alpha) = Cn(\beta) \), then \( K \div \alpha = K \div \beta \)
6. If \( \not\vdash \alpha \), then \( K \div \alpha \subseteq K \div (\alpha \land \beta) \)
7. If \( \alpha \not\in K \div (\alpha \land \beta) \), then \( K \div (\alpha \land \beta) \subseteq K \div \alpha \)

Severe withdrawal also satisfies the following postulates:

10. If \( \not\vdash \alpha \) and \( \alpha \in K \div \beta \), then \( K \div \alpha \subseteq K \div \beta \).

11. (Linearity) Either \( K \div \alpha \subseteq K \div \beta \) or \( K \div \beta \subseteq K \div \alpha \).

12. (Expulsiveness) If \( \not\vdash \alpha \) and \( \vdash \beta \), then either \( \alpha \not\in K \div \beta \) or \( \beta \not\in K \div \alpha \).

Rott and Pagnucco (1999) showed that an alternative axiomatization of severe withdrawals consists of the postulates (1) to (4) and (6) and:

9. If \( \alpha \not\in K \div \beta \), then \( K \div \beta \subseteq K \div \alpha \).

2.4 Ensconce

Williams (1992; 1995) defines an ensconce relation on a belief base \( A \) as a transitive and connected relation \( \leq \) that satisfies the following three conditions:\footnote{\( \alpha \prec \beta \) means \( \alpha \leq \beta \) and \( \beta \not\in \alpha \). \( \alpha = \leq \beta \) means \( \alpha \leq \beta \) and \( \beta \leq \alpha \).}

(1) If \( \beta \in A \setminus Cn(\emptyset) \), then \( \{ \alpha \in A : \beta \prec \alpha \} \not\vdash \beta \)
(2) If \( \not\vdash \alpha \) and \( \vdash \beta \), then \( \alpha \prec \beta \), for all \( \alpha, \beta \in A \)
(3) If \( \vdash \alpha \) and \( \not\vdash \beta \), then \( \alpha \not\prec \beta \), for all \( \alpha, \beta \in A \)

(1) says that the formulae that are strictly more ensconced than \( \alpha \) do not (even conjointly) imply \( \alpha \). Conditions (2) and (3) say that tautologies are the most ensconced formulae. If \( \leq \) is well-ranked/inversely well-ranked, then the ensconce \((A, \leq)\) is well-ranked/inversely well-ranked. If \( \leq \) is both well-ranked and inversely well-ranked then it is a bounded ensconce.

Given an ensconce relation, a cut operator for \( \alpha \in Cn(A) \) is defined by:

\[ cut_{\leq}(\alpha) = \{ \beta \in A : \gamma \in A : \beta \leq \gamma \} \not\vdash \alpha \].

A proper cut for \( \alpha \in L \) is defined by:

\[ cut_{\leq}(\alpha) = \{ \beta \in A : \gamma \in A : \beta \leq \gamma \} \not\vdash \alpha \].

Observation 1 (Williams 1994a)

If \( \alpha \in A \), \( cut_{\leq}(\alpha) = \{ \beta \in A : \beta \prec \alpha \} \)

The previous observation says that when \( \alpha \) is an explicit belief, its proper cut is the subset of \( A \) such that its members are strictly more ensconced than \( \alpha \). Other properties of proper cut are:

Observation 2 Let \((A, \leq)\) be a bounded ensconce and \( \alpha, \beta \in Cn(A) \), then:

(a) Let \( \not\vdash \beta \). If \( cut_{\leq}(\alpha) \subseteq cut_{\leq}(\beta) \), then \( cut_{\leq}(\alpha) \subseteq cut_{\leq}(\beta) \).

(b) If \( \vdash \beta \) and \( \not\vdash \alpha \), then \( cut_{\leq}(\beta) \subseteq cut_{\leq}(\alpha) \).

Intuitively, an ensconce is to belief bases as epistemic entrenchment is to belief sets. Williams explores this relation:

Definition 3 (Williams 1994b) Let \((A, \leq)\) be an ensconce. For \( \alpha, \beta \in L \), define \( \alpha \leq_{\leq} \beta \) to be given by: \( \alpha \leq_{\leq} \beta \) if and only if either:

1. \( \alpha \not\in Cn(A) \), or ii) \( \alpha, \beta \in Cn(A) \) and \( cut_{\leq}(\beta) \subseteq cut_{\leq}(\alpha) \).

Observation 4 (Williams 1994b) If \((A, \leq)\) is an ensconce, then \( \leq_{\leq} \) is an epistemic entrenchment related to \( Cn(A) \).

Observation 5 (Williams 1994b) Given an ensconce \((A, \leq)\), \( \leq_{\leq} \) is well-ranked (inversely well-ranked, bounded) if and only if \( \leq_{\leq} \) is well-ranked (inversely well-ranked, bounded).

2.5 Brutal Contraction

Mary-Anne Williams (Williams 1994b) defines two operators for base contraction: The first one inspired in AGM contraction (ensconce-based contraction) and the second one inspired in severe withdraw (brutal contraction). In this paper we will focus in the second one. Brutal contraction, as Mary-Anne Williams says, “retains as little as necessary of the theory base”.

Definition 6 (Williams 1994b) Let \( A \) be a belief base. An operation \( \phi \) is a brutal base contraction on \( A \) if and only if there is an ensconce relation \( \leq_{\leq} \) on \( A \) such that:

\[ \beta \in A - \alpha \) if and only if \( \beta \in A \text{ and either (i) } \alpha \in Cn(\emptyset) \text{ or (ii) } \beta \in cut_{\leq}(\alpha) \).

In (Garapa, Fermé, and Reis 2016) the following axiomatic characterization for brutal base contractions was presented:

Observation 7 (Garapa, Fermé, and Reis 2016) Let \( A \) be a belief base. An operator \( \neg \) of \( A \) is a brutal base contraction on \( A \) if and only if it satisfies:

(Success) If \( \not\vdash \alpha \), then \( A - \alpha \not\vdash \alpha \)
(Inclusion) \( A - \alpha \subseteq A \)
(Vacuity) If \( A \not\vdash \alpha \), then \( A \subseteq A - \alpha \)
(Failure) If \( \vdash \alpha \), then \( A - \alpha = A \)
(Relative Closure) \( A \cap Cn(A - \alpha) \subseteq A - \alpha \)
(Strong Inclusion) If \( A - \beta \not\vdash \alpha \), then \( A - \beta \subseteq A - \alpha \)
(Uniform Behaviour) If $\beta \in A$, $A \vdash \alpha$ and $A - \alpha = A - \beta$, then $\alpha \in Cn(A - \beta \cup \{\gamma \in A : A - \beta = A - \gamma\})$

The following observation lists some other well-known postulates which are satisfied by the brutal base contraction functions.

Observation 8 (Garapa, Fermé, and Reis 2016) Let $A$ be a belief base and $-\alpha$ an operator on $A$ that satisfies success, inclusion, failure, relative closure, strong inclusion and uniform behaviour. Then it satisfies:

(a) If $\alpha \in A \setminus A - \beta$, then $A - \beta \subseteq A - \alpha$.
(b) If $A - \alpha \subseteq A - \beta$, then $A - \beta \vdash \alpha$.
(c) If $\vdash \alpha$ and $\alpha \in A$, then $\alpha \in A - \beta$.
(d) If $\vdash \alpha \leftrightarrow \beta$, then $A - \alpha = A - \beta$. (Extensionality)

3 Bounded Brutal Base Contraction Functions

In this subsection we introduce the bounded brutal base contractions and obtain an axiomatic characterization for that class of functions.

Definition 9 Let $A$ be a belief base. An operation $-\alpha$ is a bounded brutal base contraction on $A$ if and only if it is a brutal base contraction based on a bounded ensconcement.

We introduce the following postulates:

(Upper Bound) For every non-empty set $X \subseteq A$ of nontautological formulae, there exists $\alpha \in X$ such that $A - \beta \subseteq A - \alpha$ for all $\beta \in X$.

(Lower Bound) For every non-empty set $X \subseteq A$ of nontautological formulae, there exists $\alpha \in X$ such that $A - \alpha \subseteq A - \beta$ for all $\beta \in X$.

(Clustering) If $\beta \in A$, then there exists $\alpha \in A \cup Cn(\emptyset)$ such that $A - \alpha = A - \beta \cup \{\gamma \in A : A - \beta = A - \gamma\}$.

Upper Bound (respectively Lower Bound) states that every non-empty set of nontautological formulae of $A$ contains an element which is such that the result of contracting $A$ by that sentence is a superset (respectively a subset) of any set which results of contracting $A$ by one of the remaining sentences of that subset.

Clustering asserts that for any sentence $\beta$ in $A$ there exists some sentence $\alpha$ in $A \cup Cn(\emptyset)$ such that the result of the contraction of $\alpha$ from $A$ is the set consisting of the union of the result of contracting $A$ by $\beta$ with the set formed by all the sentences of $A$ which are such that the result of contracting it from $A$ coincides with the result of contracting $A$ by $\beta$.

The two following observations present some interrelations among the above proposed postulates and some of the of the postulates included in the axiomatic characterization that was obtained for the class of brutal base contraction.

Observation 10 Let $A$ be a belief base and $-\alpha$ an operator on $A$ that satisfies success, inclusion, failure, relative closure, strong inclusion and lower bound. Then it satisfies clustering.

Observation 11 Let $A$ be a belief base and $-\alpha$ an operator on $A$ that satisfies failure, success, strong inclusion and clustering. Then it satisfies uniform behaviour.

We are now in a position to present an axiomatic characterization for the class of bounded brutal base contractions.

Theorem 12 (Axiomatic characterization of bounded brutal base contraction functions) Let $A$ be a belief base. An operator $-\alpha$ on $A$ is a bounded brutal base contraction on $A$ if and only if it satisfies success, inclusion, failure, relative closure, lower bound, upper bound and strong inclusion.

The following observation exposes another relevant property of the bounded brutal base contractions which will be useful further ahead. More precisely, it asserts that for any non-tautological sentence $\alpha$ which is deducible from $A$ it holds that the result of contracting $A$ by $\alpha$ coincides with the result of the contraction of $A$ by some sentence explicitly included in $A$.

Observation 13 Let $A$ be a belief base and $-\alpha$ an operator on $A$ that satisfies success, inclusion, failure, relative closure, strong inclusion and lower bound. Then it satisfies:

For all $\alpha \in Cn(A) \setminus Cn(\emptyset)$ there exists $\beta \in A$ such that $A - \alpha = A - \beta$.

4 Relation between Bounded Brutal Base Contraction and Ensconcement-based Severe Withdrawal

In this section we will define and axiomatically characterize a particular kind of severe withdrawals which we will show to be the contraction functions that correspond to the bounded brutal base contractions in the context of belief set contractions.

We start by noticing that, given a bounded ensconcement $(A, \preceq)$, we can combine Definitions 3 and $(R_\preceq)$ in order to define a contraction function on the belief set $Cn(A)$. This kind of functions is formally introduced in the following definition.

Definition 14 $\div$ is an ensconcement-based withdrawal related to $(A, \preceq)$ if and only if $(A, \preceq)$ is a bounded ensconcement such that $Cn(A) \div \alpha = Cn(A) \div \preceq \alpha$, where $\preceq$ is the epistemic entrenchment with respect to $Cn(A)$ defined by Definition 3 and $\preceq \leq$ is the severe withdrawal on $Cn(A)$ defined by $(R_\leq)$.

Comparing the above definition with Definitions 6 and 9 it becomes clear that there is a strong interrelation among the ensconcement-based severe withdrawals and the (bounded) brutal base contractions. That interrelation is explicitly presented in the two following theorems. More precisely, given a bounded ensconcement $(A, \preceq)$, these two results expose how the $\preceq$-based brutal contraction on $A$ can be defined from the ensconcement-based withdrawal related to $(A, \preceq)$ and, vice-versa, how the latter can be defined by means of the former.

Theorem 15 Let $(A, \preceq)$ be a bounded ensconcement, $-\alpha$ be the $\preceq$-based brutal contraction, and $\div \preceq \leq$ be the ensconcement-based severe withdrawal related to $(A, \preceq)$, then $A - \alpha = (Cn(A) \div \preceq \leq \alpha) \cap A$. 
Theorem 16 Let \((A, \preceq)\) be a bounded ensconcement, \(\preceq\) be the \(\preceq\)-based brutal contraction, and \(\div\preceq\) be the ensconcement-based severe withdrawal related to \((A, \preceq)\), then \(\text{Cn}(A) \div \alpha = \text{Cn}(A - \alpha)\).

4.1 Axiomatic Characterization of ensconcement-based severe withdrawals

In this subsection we will present an axiomatic characterization for the class of ensconcement-based severe withdrawals. To do that we must start by introducing the following postulate:

**(Base-reduction)** If \(\text{Cn}(A) \div \alpha \vdash \beta\), then \((\text{Cn}(A) \div \alpha) \cap A \vdash \beta\)

This postulate essentially states that that the result of contracting the belief set \(\text{Cn}(A)\) by any sentence \(\alpha\) coincides with the logical closure of some subset of \(A\). Indeed, it is not hard to see that base-reduction is equivalent to the following postulate: \(\forall \alpha \forall A' \subseteq A : \text{Cn}(A') = \text{Cn}(A) \div \alpha\) (which is very similar to the postulate of finitude proposed by Hansson (1999a)).

The following observation highlights that for a severe withdrawal that satisfies the postulates of base-reduction and lower bound it also holds that for any non-tautological sentence \(\alpha\) in \(\text{Cn}(A)\) the result of the contraction of \(\text{Cn}(A)\) by \(\alpha\) coincides with the result of the contraction of \(\text{Cn}(A)\) by some sentence in \(A\).

**Observation 17** Let \(\div\) be an operator on \(\text{Cn}(A)\) that satisfies \((\vdash 1), (\vdash 2), (\div 4), (\div 9)\), base-reduction and lower bound, then for all \(\alpha \in \text{Cn}(A) \setminus \text{Cn}(\emptyset)\) there exists \(\beta \in A\) such that \(\text{Cn}(A) \div \alpha = \text{Cn}(A) \div \beta\).

We are now in a position to present the following axiomatic characterization for the ensconcement-based severe withdrawals.

**Theorem 18** Let \(A\) be a belief base and \(\div\) be an operator on \(\text{Cn}(A)\). \(\div\) satisfies \((\vdash 1)\) to \((\vdash 4), (\div 6), (\div 9)\), base-reduction, upper bound and lower bound if and only if there exists a bounded ensconcement such that \(\div\) is an ensconcement-based withdrawal related to \((A, \preceq)\).

Theorems 15 and 16 expose how a base contraction function can be defined from a belief set contraction function and vice-versa. Combining those two results with the axiomatic characterizations presented in Theorems 12 and 18 we can obtain the following results which highlight the correspondence among sets of postulates for base contraction and sets of postulates for belief set contraction.

**Corollary 19** An operator \(\div\) on \(A\) satisfies success, inclusion, vacuity, failure, relative closure, strong inclusion, upper bound and lower bound if and only if there exists an operator \(\div\) on \(\text{Cn}(A)\) that satisfies \((\vdash 1)\) to \((\vdash 4), (\div 6), (\div 9)\), base-reduction, upper bound and lower bound such that: \(A - \alpha = \text{Cn}(A \div \alpha) \cap A\).

**Corollary 20** An operator \(\div\) on \(\text{Cn}(A)\) satisfies \((\vdash 1)\) to \((\vdash 4), (\div 6), (\div 9)\), base-reduction, upper bound and lower bound if and only if there exists an operator \(\div\) on \(A\) that satisfies success, inclusion, vacuity, failure, relative closure, strong inclusion, upper bound and lower bound such that: \(\text{Cn}(A) \div \alpha = \text{Cn}(A - \alpha)\).

The two following observations consist of a slight refinement of the right to left part of Corollary 20. More precisely these results specify more precisely which properties of the belief base contraction are needed in order to assure that the belief set contraction obtained from it as exposed in Theorem 16 satisfies certain postulates.

**Observation 21** Let \(A\) be a belief base and \(\div\) be an operator on \(A\) that satisfies success, inclusion, vacuity, failure, relative closure and strong inclusion. If \(\div\) is an operator on \(\text{Cn}(A)\) defined by \(\text{Cn}(A) \div \alpha = \text{Cn}(A - \alpha)\) then \(\div\) satisfies \((\vdash 1)\) to \((\vdash 4), (\vdash 6), (\vdash 9)\) and base-reduction.

**Observation 22** Let \(A\) be a belief base and \(\div\) be an operator on \(A\) that satisfies success, inclusion, failure, relative closure, upper bound, lower bound and strong inclusion. If \(\div\) is an operator on \(\text{Cn}(A)\) defined by \(\text{Cn}(A) \div \alpha = \text{Cn}(A - \alpha)\) then \(\div\) satisfies upper bound and lower bound.

5 Conclusions

We have presented an axiomatic characterizations for the subclass of brutal base contractions formed by the brutal contractions that are based on a bounded ensconcement relation. We have also introduced and axiomatically characterized the class of ensconcement-based severe withdrawals which is formed by the severe withdrawals that are based on epistemic entrenchment relations which are obtained from an ensconcement relation using the construction proposed by Mary-Anne Williams. Some results were presented concerning the interrelation among the classes of bounded brutal base contractions and of ensconcement-based severe withdrawals. Finally we presented some results relating base contraction postulates and belief set contraction postulates by means of explicit definitions of belief set contractions from base contractions and vice-versa.

Acknowledgements

We wish to thank the three reviewers for their comments which have contributed to the improvement of this paper.

Appendix: Proofs

Previous Lemmas

**Lemma 23** (Fermé, Krevneris, and Reis 2008)

(a) If \(\nvdash\alpha, cut_\prec(\alpha) \nvdash\alpha\).

(b) If \(A \nvdash\alpha, cut_\prec(\alpha) = A\).

(c) If \(\beta \vdash\alpha\), then \(cut_\prec(\alpha) \subseteq cut_\prec(\beta)\).

(d) If \(\alpha \preceq\beta\), then \(cut_\prec(\beta) \subseteq cut_\prec(\alpha)\).

(e) If \(cut_\prec(\alpha) \vdash\beta\), then \(cut_\prec(\alpha \land \beta) = cut_\prec(\alpha)\).

(f) If \(cut_\prec(\alpha) \nvdash\beta\), then \(cut_\prec(\alpha \land \beta) = cut_\prec(\beta)\).

**Lemma 24** (Rott and Pagnucco 1999, Observation 19(ii)) If \(\div\) is a severe withdrawal function, then \(\div\) can be represented as an ensconcement-based withdrawal where the relation \(\preceq\) on which \(\div\) is based is obtained by (Def \(\preceq\) from \(\div\)) \(\alpha \preceq\beta\) if and only if \(\alpha \not\in K\div\beta\) or \(\vdash\beta\) and \(\preceq\) satisfies (EE1) to (EE5).
Lemma 25 Let \((A, \preceq)\) be a bounded ensconcement and \(\text{cut}_\preceq(\alpha) \neq \emptyset\). Then there exists \(\beta \in \text{cut}_\preceq(\alpha)\) such that \(\text{cut}_\preceq(\beta) = \text{cut}_\preceq(\alpha)\).

Lemma 26 Let \((A, \preceq)\) be a bounded ensconcement and \(\alpha \in Cn(A)\). Then \(\text{cut}_\preceq(\alpha) \vdash \alpha\).

Lemma 27 Let \((A, \preceq)\) be a bounded ensconcement and \(\alpha, \beta \in Cn(A)\). If \(\text{cut}_\preceq(\alpha) \subseteq \text{cut}_\preceq(\beta)\), then \(\text{cut}_\preceq(\alpha) \subseteq \text{cut}_\preceq(\beta)\).

Proofs

Proof of Theorem 12

From bounded brutal base contraction to postulates

Let \(-\) be a bounded brutal base contraction operator on \(A\).

By Observation 7 \(-\) satisfies success, inclusion, vacuity, failure, relative closure and strong inclusion. It remains to show that \(-\) satisfies upper bound and lower bound.

Upper Bound Let \(X \subseteq A\) be a non empty set of non-tautological formulae. Since \(\preceq\) is well ranked there exists \(\beta \in X\) such that \(\beta \preceq \alpha\) for all \(\alpha \in X\). Hence, by Lemma 23 (d), there exists \(\beta \in X\) for all \(\alpha \in X\) such that \(\text{cut}_\preceq(\alpha) \subseteq \text{cut}_\preceq(\beta)\). Therefore, by definition of \(-\) there exists \(\beta \in X\) for all \(\alpha \in X\) such that \(\alpha \preceq \beta\) for all \(\alpha \in X\).

Lower Bound Analogous to upper bound.

From postulates to bounded brutal base contraction

Let \(-\) be an operator on \(A\) that satisfies success, inclusion, vacuity, failure, relative closure, lower bound, upper bound and strong inclusion. From Observation 10 and Observation 11 it follows that \(-\) satisfies uniform behaviour. Let \(\preceq\) be defined by:

\[
\alpha \preceq \beta \text{ if } \left\{ \begin{array}{ll}
A - \beta \subseteq A - \alpha \text{ and } \not\vdash \alpha & \\
\vdash \beta & 
\end{array} \right.
\]

According to the Postulates to Construction part of the proof of Observation 7 \(-\) satisfies \((\preceq 1) \cdot (\preceq 3)\) and is such that

\[
A - \alpha = \left\{ \begin{array}{ll}
\text{cut}_\preceq(\alpha) & \text{if } \not\vdash \alpha \\
\text{otherwise} & 
\end{array} \right.
\]

It remains to prove that \(\preceq\) is bounded. To do so we must prove that \(\preceq\) is well-ranked and inversely well-ranked.

(\(\preceq\) is well-ranked) Let \(X \neq \emptyset\) and \(X \subseteq A\). We will prove by cases:

Case 1) All formulae in \(X\) are tautologies. Let \(\beta\) be one of those formulas. Hence by \((\preceq 3)\) \(\beta \preceq \alpha\) for all \(\alpha \in X\).

Case 2) All formulae in \(X\) are non-tautological. By upper bound there exists \(\beta \in X\) such that \(A - \alpha \subseteq A - \beta\). Hence, by definition of \(\preceq\), there exists \(\beta \in X\) such that \(\beta \preceq \alpha\) for all \(\alpha \in X\).

(\(\preceq\) is inversely well-ranked) Let \(X \neq \emptyset\) and \(X \subseteq A\). We will prove by cases:

Case 1) There are some \(\beta \in X\) such that \(\vdash \beta\). Then, by definition of \(\preceq\), \(\alpha \preceq \beta\) for all \(\alpha \in X\).

Case 2) All formulae in \(X\) are non-tautological. By lower bound there exists \(\beta \in X\) such that \(A - \beta \subseteq A - \alpha\) for all \(\alpha \in X\). Hence, by definition of \(\preceq\), there exists \(\beta \in X\) such that \(\alpha \preceq \beta\) for all \(\alpha \in X\).

Proof of Theorem 15

We will prove by cases:

Case 1) \(\vdash \alpha\). It follows that \(A - \alpha = A\) and \((Cn(A) \preceq \leq \alpha) \cap A = A\).

Case 2) \(A \not\vdash \alpha\). It follows that \((Cn(A) \preceq \leq \alpha) \cap A = A\) and that \(A - \alpha = \text{cut}_\preceq(\alpha)\). By Lemma 23 (b), it follows that \(\text{cut}_\preceq(\alpha) = A\).

Case 3) \(A \vdash \alpha\) and \(\not\vdash \alpha\).

We will prove that \(A - \alpha = (Cn(A) \preceq \leq \alpha) \cap A\) by double inclusion.

(\(\subseteq\)) Let \(\beta \in A - \alpha\). It follows that \(\beta \in A\). It remains to prove that \(\beta \in Cn(A) \preceq \leq \alpha\), i.e. that \(\beta \in \{ \psi \in Cn(A) : \text{cut}_\preceq(\psi) \subseteq \text{cut}_\preceq(\alpha)\}\).

\(\supseteq\) Let \(\beta \in \{ \psi \in Cn(A) : \text{cut}_\preceq(\psi) \subseteq \text{cut}_\preceq(\alpha)\}\). We will prove that \(\beta \in Cn(A) \preceq \leq \alpha\) and \(\not\vdash \alpha\).

Proof of Theorem 16

We will prove by cases:

Case 1) \(\vdash \alpha\). Then \(Cn(A) \preceq \leq \alpha = Cn(A)\) and \(A - \alpha = A\).

Hence \(Cn(A - \alpha) = Cn(A)\) and \(\text{cut}_\preceq(\alpha) = Cn(A)\) and, by Lemma 23 (b), \(A - \alpha = \text{cut}_\preceq(\alpha) = A\). Hence \(Cn(A - \alpha) = Cn(A)\). \(\preceq \leq \alpha\)

Case 3) \(A \vdash \alpha\) and \(\not\vdash \alpha\). Hence \(Cn(A) \preceq \leq \alpha = \{ \psi \in Cn(A) : \alpha \not\leq \psi\} = \{ \psi \in Cn(A) : \text{cut}_\preceq(\psi) \subseteq \text{cut}_\preceq(\alpha)\}\).

We will prove that \(Cn(A - \alpha) = Cn(A)\) \(\preceq \leq \alpha\) by double inclusion.

(\(\subseteq\)) Let \(\beta \in Cn(A - \alpha)\). If \(\vdash \beta\), then \(\beta \in Cn(A)\) and, by Observation 2 (b), \(\text{cut}_\preceq(\beta) \subseteq \text{cut}_\preceq(\alpha)\). Hence \(\beta \in Cn(A)\) \(\preceq \leq \alpha\).

Assume now that \(\not\vdash \beta\), then \(\beta \not\in \text{cut}_\preceq(\alpha)\). Hence \(\text{cut}_\preceq(\beta) \subseteq \text{cut}_\preceq(\alpha)\) and \(\beta \not\in \text{cut}_\preceq(\alpha)\).

Therefore, since \(\psi \in Cn(A) : \gamma \not\leq \psi\) \(\vdash \beta\), it follows that \(\{ \psi \in Cn(A) : \gamma \not\leq \psi\} \vdash \beta\) which contradicts \((\preceq 1)\).
α). Therefore cut≺(α) ⊄ β. By Lemma 23 (f) it follows that
cut≺(α ∧ β) = cut≺(β). From α ∧ β ⊄ α, by Lemma 23
(c), it follows that cut≺(α) ⊆ cut≺(β). From Observation 2
(a) it follows that cut≺(α) ⊆ cut≺(β). Contradiction. ■

Proof of Theorem 18

(⇒) Let ⊸ be an ensnacement-based withdrawal related to
(A, ≤) and let ≤ ≤ ≤. Hence ⊸ satisfies the postulates
for severe withdrawals. It remains to show that ⊸ satisfies:
base-reduction, upper bound and lower bound.

Upper Bound: Let ⊸ be an ensnacement-based withdrawal related to
(A, ≤). Let X = /\ 0 and X ⊆ Cn(A) \ Cn(∅). From Observation 5,
so (A, ≤) is a bounded ensnacement, it follows that ≤ is
bounded. Hence, there exists x ∈ X such that ≤ x /\ for all
x ∈ X. We will prove that Cn(A) x /\ Cn(x) x β x for all
x ∈ X. Let γ ∈ Cn(A) x /\ α. Hence, by definition of ⊃, γ
x Cn(A) and α ≤ γ. By EE1, since ≤ x α and ≤ γ it
follows that γ < γ. Hence, γ ∈ Cn(A) x /\ β. Therefore
Cn(A) x /\ α ⊆ Cn(A) x /\ β.

Lower Bound: Analogous to upper bound.

Base-reduction: Let Cn(A) x /\ α ⊂ β. We will prove that
(Cn(A) x /\ α) \ A ⊂ β by cases:

Case 1) β ⊂ A. Follows trivially.

Case 2) α ⊂ Cn(A) or ⊸ α. Follows trivially by (R_c).

Case 3) β ⊂ Cn(A) and /\ α. From
Case 3) /\ β, α ∈ Cn(A) and α /\ β, it follows by (R_c), that X ⊂ β
where X = {ψ ∈ Cn(A) : cut≺(ψ) ⊂ cut≺(α)} \ X \ Cn(∅) = /\, since
β. Let ψ ∈ X \ Cn(∅). Assume that cut≺(ψ) = /\ and let θ ∈ Cn(∅). Hence, by EE5, it follows that
ψ < θ. Hence, by Definition 3, cut≺(θ) ⊂ cut≺(ψ) = /\.

Contradiction. Hence cut≺(ψ) /\ α = /\, 0. From Lemma
25, and since ≤ is bounded, it follows that there
exists δ ∈ cut≺(ψ) such that cut≺(δ) = cut≺(ψ). Let
Y = {μ ∈ A : cut≺(μ) ⊂ cut≺(α)}. Let μ ∈ Y
such that μ ≤ μ for all μ ∈ Y. Let λ ∈ cut≺(μ). Hence
cut≺(λ) ⊂ cut≺(μ), from which follows that
cut≺(λ) ⊂ cut≺(α). Therefore λ ∈ Y. Let φ ∈ Y. It
follows that μ ≤ φ. Hence φ ∈ cut≺(μ). Therefore
Y = cut≺(μ). By Lemma 26 cut≺(ψ) ⊂ Y. Hence,

so Y ⊂ β. Hence, for all ψ ∈ Cn(A) x /\ it follows that
Y ⊂ β. Therefore, since Cn(A) x /\ α it follows that
Y ⊂ β, Y ⊆ Cn(A) x /\ A. Hence (Cn(A) x /\ α) \ A ⊂ β.

(⇒) Let A be a belief base and ⊸ an operator on
(Cn(A), ≤) satisfies (1) to (4), (6), (9), base-
reduction, upper bound and lower bound. Let ≤ be a binary
relation on A defined by:

α ≤ β if and only if α /\ Cn(A) x /\ β, x
We will prove that ≤ is a bounded ensnacement.

(1) Let γ ∈ A \ Cn(∅) we must show that
H = {α ∈ A : γ < α} /\ γ. It is enough to show that
H \ Cn(∅) γ. Let α ∈ A \ Cn(∅) and γ < α. Then,
γ ≤ α and α ≤ γ, Hence, by definition of ≤, it follows that
γ /\ Cn(A) < α, α /\ Cn(A) γ and γ \ γ, H \ Cn(A) γ
where, γ γ. Hence, since by (3) Cn(A) γ /\ γ it follows that
H γ γ.

(2) Let α, β ∈ A be such that γ /\ α and γ β. From γ β
it follows, by definition of ≤, that α ≤ β. Assume by
reductio that γ /\ α and β ≤ α. Hence, by definition
of ≤, β /\ Cn(A) α or αγ. Contradiction, since γ /\ α and by
(1) β ⊂ Cn(A) x /\ α.

(3) Follows trivially by definition of ≤.

(4) is transitive Let α ≤ β and β ≤ γ. Hence, by
definition of ≤, it follows that (α ≤ Cn(A) x /\ β) or (β ≤ Cn(A) x /\ γ). Hence, α ≤ Cn(A) x /\ β
and (β ≤ Cn(A) x /\ γ or (β ≤ Cn(A) x /\ γ). Hence, we have four cases to consider:

Case 1) α /\ Cn(A) x /\ β and β /\ Cn(A) x /\ γ. From
(1) it follows that Cn(A) x /\ α ⊂ Cn(A) x /\ β, Hence,
α /\ Cn(A) x /\ γ. Therefore α ≤ γ, by definition of ≤.

Case 2) α /\ Cn(A) x /\ β and tan γ ≤ ≤ γ follows trivially
by definition of ≤.

Case 3) β /\ Cn(A) x /\ γ. Contradicts (1).

Case 4) γ /\ α and γ /\ β follows trivially by definition of ≤.

(5) is connected Let α /\ Cn(A) x /\ β. Hence, α /\ Cn(A) x /\ β and
γ /\ β. We will consider two cases:

Case 1) γ /\ α. Hence β ≤ α, by definition of ≤.

(6) is expulsiveness, β /\ Cn(A) x /\ α. Therefore,
by definition of ≤, we must have β ≤ α.

(7) is well-ranked Let X ⊂ A a non empty set. We
will prove by cases:

Case 1) X ⊂ Cn(∅). Trivial.

Case 2) X ⊂ Cn(A). Let X’ = X \ Cn(∅). Hence, by
\ upper bound there exists β ∈ X’ such that
Cn(A) x /\ Cn(A) x /\ β for all α ∈ X’. By (4)
β /\ Cn(A) x /\ α for all α ∈ X’. Hence, by definition of ≤,
there exists β ∈ X’ such that β ≤ α for all α ∈ X’. If
X = X’ trivial. Assume now that X = X’. Then γ ∈ X’ \ X’.
Hence ⊸ γ and by (7) it follows that β ≤ γ. Therefore,
there exists β ∈ X such that ≤ β for all α ∈ X.

(8) is inversely well-ranked Let X ⊂ A non empty set.
We will consider two cases:

Case 1) X \ Cn(A) = /\, 0. Let β ∈ X \ Cn(A) hence, by
definition of ≤, we must have β ≤ α for all α ∈ X.

Case 2) X \ Cn(A) = /\, 0. Hence, by \ lower bound, there
exists α ∈ X such that α /\ Cn(A) x /\ β for all α ∈ X.
By (3) α /\ Cn(A) x /\ β, for all α ∈ X. Hence, by definition of ≤
there exists β ∈ X such that ≤ β for all α ∈ X.

We have proved that ≤ is a bounded ensnacement.

Let ≤ be as in Definition 3. According to Observation
4 and Observation 5 ≤ is a bounded episentnic
entrenchment related to Cn(A). It remains to show that
Cn(A) x /\ α = Cn(A) x /\ α, where x is defined (in
(R_c)) by:

\begin{align*}
Cn(A) x /\ α & = \left\{ \begin{array}{ll}
Cn(A) & \text{if } α \in Cn(A) \text{ and } γ /\ α \\
Cn(A) & \text{otherwise}
\end{array} \right.
\end{align*}

According to Lemma 24 and since ≤ is a severe withdrawal
function, the episentnic entrenchment ≤ on which ≤
is based on is such that: α ≤ β if and only if α /\ Cn(A) x /\ β
or γ β. Thus to prove that Cn(A) x /\ α = Cn(A) x /\ α it
is enough to show that:

\begin{align*}
\end{align*}
\( \alpha \leq \beta \) if and only if \( \alpha \notin Cn(A) \div \beta \) or \( \vdash \beta \).

\((\Rightarrow)\) Let \( \alpha \leq \beta \). Hence, by definition of \( \leq \), \( \alpha \leq \beta \) if and only if:

i) \( \alpha \notin Cn(A) \), or

ii) \( \beta \in Cn(A) \) and \( cut_{\leq}(\beta) \subseteq cut_{\leq}(\alpha) \).

We will prove by cases:

Case 1) \( \alpha \notin Cn(A) \). Then, by \((\div 2)\), \( \alpha \notin Cn(A) \div \beta \).

Case 2) \( \alpha, \beta \in Cn(A) \) and \( cut_{\leq}(\beta) \subseteq cut_{\leq}(\alpha) \).

Case 2.1) \( \vdash \beta \). Trivial.

Case 2.2) \( \nvdash \beta \). 

\( \{ \gamma \in A : \{ \delta \in A : \gamma < \delta \} \cap \beta \} \subseteq \{ \gamma \in A : \{ \delta \in A : \gamma < \delta \} \cap \alpha \} \).

Hence, \( \{ \gamma \in A : \{ \delta \in A : \gamma < \delta \} \cap \beta \} \subseteq \{ \gamma \in A : \{ \delta \in A : \gamma < \delta \} \cap \alpha \} \).

From \( \alpha \in Cn(A) \div \beta \) it follows, by base-reduction, that \( Cn(A) \div \beta \cap A \cap \alpha \). By compactness, there exists a finite subset of \( Cn(A) \div \beta \cap A \) such that \( H \vdash \alpha \). Let us assume that \( H \cap Cn(\emptyset) = \emptyset \). For all \( \alpha \in H \), \( \alpha \in Cn(A) \div \beta = Cn(A) \div \beta ^{\prime} \), for some \( \beta ^{\prime} \in A \) (by observation 17). Hence, by expulsiveness, \( \beta ^{\prime} \notin Cn(A) \div \alpha \).

Therefore \( \beta ^{\prime} \notin Y \), since \( H \subseteq Z = \{ \delta \in A : (\beta ^{\prime} \notin Cn(A) \div \delta \) and \( \delta \in Cn(A) \div \beta ^{\prime} \div \beta ^{\prime} \) or \( (\delta \div \beta ^{\prime} \div \beta ^{\prime} ) \}. \)

On the other hand \( \beta ^{\prime} \notin Y \), since \( Z \subseteq Cn(A) \div \beta ^{\prime} \), and by \((\div 4)\) \( Cn(A) \div \beta ^{\prime} \nvdash \beta ^{\prime} \). Hence \( X \subseteq Y \). Contradiction.

\((\Leftarrow)\) Let \( \alpha \notin Cn(A) \div \beta \). We will prove by cases:

Case 1) \( \alpha \notin Cn(A) \). Trivial.

Case 2) \( \alpha \in Cn(A) \).

Case 2.1) \( \vdash \beta \). Then \( \alpha, \beta \in Cn(A) \) and \( cut_{\leq}(\beta) \subseteq cut_{\leq}(\alpha) \).

Case 2.2) \( \alpha \notin Cn(A) \) \div \beta \) and \( \nvdash \beta \). Hence, it follows that \( \beta \in Cn(A) \), \( \alpha \) and \( Cn(A) \div \beta \subseteq Cn(A) \div \alpha \), by \((\div 3)\), \((\div 1)\) and \((\div 0)\), respectively.

Let us assume by reductio that \( cut_{\leq}(\beta) \subseteq cut_{\leq}(\alpha) \). Hence there exists \( \psi \in \alpha \) such that \( \psi \notin cut_{\leq}(\beta) \) and \( \psi \notin cut_{\leq}(\alpha) \). From which follows that \( \nvdash \psi \) and \( C \subseteq \{ \delta \in A : (\psi \notin Cn(A) \div \delta \) and \( \delta \in Cn(A) \div \psi \} \div \beta \) and \( \nvdash \beta \). Then \( Cn(A) \div \psi \nvdash \alpha \). Hence, by \((\div 4)\) and linearity, it follows that \( Cn(A) \div \alpha \subseteq Cn(A) \div \psi \). From \( Cn(A) \div \beta \subseteq Cn(A) \) it follows that \( Cn(A) \div \beta \subseteq Cn(A) \div \psi \). By \((\div 0)\), \( \beta \notin Cn(A) \div \psi \). Therefore, by base-reduction, \( Cn(A) \div \psi \nvdash \alpha \). On the other hand \( Cn(A) \div \psi \nvdash \alpha \subseteq C \).

Hence \( \nvdash \beta \). Contradiction.

References


A strengthening of rational closure in DLs: reasoning about multiple aspects

Valentina Gliozzi

Center for Logic, Language and Cognition
Dipartimento di Informatica - Università di Torino - Italy
valentina.gliozzi@unito.it

Abstract

We propose a logical analysis of the concept of typicality, central in human cognition (Rosch, 1978). We start from a previously proposed extension of the basic Description Logic \(ALC\) with a typicality operator \(\mathcal{T}\) that allows to consistently represent the attribution to classes of individuals of properties with exceptions (as in the classic example (i) typical birds fly, (ii) penguins are birds but (iii) typical penguins don’t fly). We then strengthen this extension in order to separately reason about the typicality with respect to different aspects (e.g., flying, having nice feather: in the previous example, penguins may not inherit the property of flying, for which they are exceptional, but can nonetheless inherit other properties, such as having nice feather).

Introduction

In (Giordano et al. 2015) it is proposed a rational closure strengthening of \(ALC\). This strengthening allows to perform non monotonic reasoning in \(ALC\) in a computationally efficient way. The extension, as already the related logic \(ALC + \mathcal{T}_{\min}\) proposed in (Giordano et al. 2013a) and the weaker (monotonic) logic \(ALC + \mathcal{T}\) presented in (Giordano et al. 2009), allows to consistently represent typical properties with exceptions that could not be represented in standard \(ALC\).

For instance, in all the above logics one can say that:

**SET 1:**
- Typical students don’t earn money
- Typical working students do earn money
- Typical apprentice working students don’t earn money

without having to conclude that there cannot exist working students nor apprentice working students. On the contrary, in standard \(ALC\) typicality cannot be represented, and these three propositions can only be expressed by the stronger ones:

**SET 2:**
- Students don’t earn money \((\text{Student} \subseteq \neg \text{EarnMoney})\)
- Working students do earn money \((\text{Worker} \cap \text{Student} \subseteq \text{EarnMoney})\)
- Apprentice working students don’t earn money \((\text{Worker} \cap \text{Apprentice} \subseteq \neg \text{EarnMoney})\)

These propositions are consistent in \(ALC\) only if there are no working students nor apprentice working students.

In all the extensions of \(ALC\) mentioned above one can represent the set of propositions in **SET 1** by means of a typicality operator \(\mathcal{T}\) that, given a concept \(C\) (e.g., Student) singles out the most typical instances of \(C\): so, for instance, \(\mathcal{T}(\text{Student})\) refers to the typical instances of the concept Student. The semantics of \(\mathcal{T}\) is given by means of a preference relation \(<\) that compares the typicality of two individuals: for any two \(x\) and \(y\), \(x < y\) means that \(x\) is more typical than \(y\). Typical instances of a concept \(C\) are those minimal with respect to \(<\) (formally, as we will see later, \((\mathcal{T}(C))^I = \min_x(C)^I\), where \(\min_x(C)^I = \{x \in C^I : \forall y \in C^I \text{ s.t. } y < x\}\)).

The operator \(\mathcal{T}\) has all the properties that, in the analysis of Kraus Lehmann and Magidor (Kraus, Lehmann, and Magidor 1990) any non monotonic entailment should have. For instance, \(\mathcal{T}\) satisfies the principle of cautious monotonicity, according to which if \(\mathcal{T}(\text{Student}) \subseteq \text{Young}\), then \(\mathcal{T}(\text{Student}) = \mathcal{T}(\text{Student} \cap \text{Young})\). The precise relations between the properties of \(\mathcal{T}\) and preferential entailment are established in (Giordano et al. 2009).

Although the extensions of \(ALC\) with the typicality operator \(\mathcal{T}\) allow to express **SET 1** of propositions, the resulting logic is monotonic, and it does not allow to perform some wanted, non monotonic inferences. For instance, it does not allow to deal with irrelevance which is the principle that from the fact that typical students are young, one would want to derive that typical blond students also are young, since being blond is irrelevant with respect to youth. As another example, when knowing that an individual, say John, is a student, and given **SET 1** of propositions, one would want to conclude that John is a typical student and therefore does not earn money. On the other hand, when knowing that John is a working student, one would want to conclude that he is a typical working student and therefore does earn money. In other words one would want to assume that an individual is a typical instance of the most specific class it belongs to, in the absence of information to the contrary.

These stronger inferences all hold in the strengthening of \(ALC + \mathcal{T}\) presented in (Giordano et al. 2013a; 2015). In particular, (Giordano et al. 2015) proposes an adaptation to \(ALC\) of the well known mechanism of rational closure, first proposed by Lehman and Magidor in (Lehmann and Magidor 1992). From a semantic point of view, this strengthening of \(ALC + \mathcal{T}\) corresponds to restricting one’s
attention to minimal models, that minimize the height (rank) of all domain elements with respect to < (i.e. that minimize the length of the < -chains starting from all individuals). Under the condition that the models considered are canonical, the semantic characterization corresponds to the syntactical rational closure. This semantics supports all the above wanted inferences, and the nice computational properties of rational closure guarantee that whether the above inferences are valid or not can be computed in reasonable time.

The main drawback of rational closure is that it is an all-or-nothing mechanism: for any subclass $C'$ of $C$ it holds that either the typical members of $C'$ inherit all the properties of $C$ or they don’t inherit any property. Once the typical members of $C'$ are recognized as exceptional with respect to $C$ for a given aspect, they become exceptional for all aspects.

Consider the classic birds/penguins example, expressed by the following propositions:

**Propositions**

- Typical birds fly
- Penguins are birds
- Typical penguins do not fly

In this case, since penguins are exceptional with respect to the aspect of flying, they are non-typical birds, and for this reason they do not inherit any of the typical properties of birds.

On the contrary, given SET3 of propositions, one wants to separately reason about the inheritance of different properties. This is to say that one wants to separately reason about the property of flying, they are non-typical birds, and for this reason they do not inherit any of the typical properties of birds.

- **Typical penguins have nice feather**

This is to say that one wants to separately reason about the different aspects: the property of flying is not related to the property of having nice feather, hence we want to separately reason on the two aspects.

Here we propose a strengthening of the semantics used for rational closure in ALC (Giordano et al. 2015) that only used a single preference relation $<$ by allowing, beside $<$, several preference relations that compare the typicality of individuals with respect to a given aspect. Obtaining a strengthening of rational closure is the purpose of this work. This puts strong constraints on the resulting semantics, and defines the horizon of this work. In this new semantics we can express the fact that, for instance, $x$ is more typical than $y$ with respect to the property of flying but $y$ is more typical that $x$ with respect to some other property, as the property of having nice feather. To this purpose we consider preference relations indexed by concepts that stand for the above mentioned aspects under which we compare individuals. So we will write that $x <_A y$ to mean that $x$ is preferred to $y$ for what concerns aspect $A$: for instance $x <_{FLy} y$ means that $x$ is more typical than $y$ with respect to the property of flying.

We therefore proceed as follows: we first recall the semantics of the extension of ALC with a typicality operator $\mathcal{T}$ which was at the basis of the definition of rational closure and semantics in (Giordano et al. 2013b; 2015). We then expand this semantics by introducing several preference relations, that we then minimize obtaining our new minimal models’ mechanism. As we will see this new semantics leads to a strengthening of rational closure, allowing to separately reason about the inheritance of different properties.

### The Operator $\mathcal{T}$ and the General Semantics

Let us briefly recall the logic $\mathcal{ALC} + \mathcal{T}_R$ which is at the basis of a rational closure construction proposed in (Giordano et al. 2015) for $\mathcal{ALC}$. The intuitive idea of $\mathcal{ALC} + \mathcal{T}_R$ is to extend the standard $\mathcal{ALC}$ with concepts of the form $\mathcal{T}(C)$, whose intuitive meaning is that $\mathcal{T}(C)$ selects the typical instances of a concept $C$, to distinguish between the properties that hold for all instances of concept $C (C \subseteq D)$, and those that only hold for the typical such instances ($\mathcal{T}(C) \subseteq D$). The $\mathcal{ALC} + \mathcal{T}_R$ language is defined as follows: $C_R := A | \top | \bot | \neg C_R \; | \; C_R \cap C_R \; | \; C_R \cup C_R | \forall R.C_R | \exists R.C_R$, and $C_L := C_R \cup \mathcal{T}(C_R)$, where $A$ is a concept name and $R$ a role name. A KB is a pair $(TBox, ABox)$. TBox contains a finite set of concept inclusions $C_L \subseteq C_R$. ABox contains a finite set of assertions of the form $C_L(a)$ and $R(a, b)$, where $a, b$ are individual constants.

The semantics of $\mathcal{ALC} + \mathcal{T}_R$ is defined in terms of rational models: ordinary models of $\mathcal{ALC}$ are equipped with a preference relation $<$ on the domain, whose intuitive meaning is to compare the “typicality” of domain elements: $x < y$ means that $x$ is more typical than $y$. Typical members of a concept $C$, instances of $\mathcal{T}(C)$, are the members $x$ of $C$ that are minimal with respect to $<$ (such that there is no other member of $C$ more typical than $x$). In rational models $<$ is further assumed to be modular: for all $x, y, z \in \Delta$, if $x < y$ then either $x < z$ or $z < y$. These rational models characterize $\mathcal{ALC} + \mathcal{T}_R$.

#### Definition 1 (Semantics of $\mathcal{ALC} + \mathcal{T}_R$ (Giordano et al. 2015))

A model $M$ of $\mathcal{ALC} + \mathcal{T}_R$ is any structure $(\Delta, \prec, I)$ where:

- $\Delta$ is the domain; $\prec$ is an irreflexive, transitive, and modular relation over $\Delta$ that satisfies the finite chain condition (there is no infinite $\prec$ -descending chain, hence if $S \neq \emptyset$, then $\min_{\prec}(S) \neq \emptyset$).
- $I$ is the extension function that maps each concept name $C$ to $C^I \subseteq \Delta$, each role name $R$ to $R^I \subseteq \Delta^I \times \Delta^I$ and each individual constant $a \in \Delta$ to $a^I \in \Delta$. For concepts of $\mathcal{ALC}$, $C^I$ is defined in the usual way. For the $\mathcal{T}$ operator, we have $(\mathcal{T}(C))^I = \min_{\prec}(C^I)$.

As shown in (Giordano et al. 2015), the logic $\mathcal{ALC} + \mathcal{T}_R$ enjoys the finite model property and a finite $\mathcal{ALC} + \mathcal{T}_R$ models can be equivalently defined by postulating the existence of a function $k_M : \Delta \rightarrow \mathbb{N}$, where $k_M$ assigns a finite rank to each world: the rank $k_M$ of a domain element $x \in \Delta$ is the length of the longest chain $x_0 < \cdots < x$ from $x$ to a minimal $x_0$ (s. t. there is no $x' < x$).

A model $M$ satisfies a knowledge base $K$ if its TBox, when $K$ extends $\mathcal{ALC}$, or if its ABox when $K$ extends $\mathcal{ALC}$.

Although the typicality operator $\mathcal{T}$ itself is nonmonotonic (i.e. $\mathcal{T}(C) \subseteq D$ does not imply $\mathcal{T}(C \cap E) \subseteq D$), the logic $\mathcal{ALC} + \mathcal{T}_R$ is monotonic: what is logically entailed by $K$ is still entailed by any $K'$ with $K \subseteq K'$.
which extends to DLs the notion of rational closure proposed in the propositional context by Lehmann and Magidor (Lehmann and Magidor 1992). The definition is based on the notion of exceptionality. Roughly speaking \( T(C) \subseteq D \) holds (is included in the rational closure) of \( K \) if (indeed, \( C \cap D \)) is less exceptional than \( C \cap \neg D \). We briefly recall this construction and we refer to (Giordano et al. 2013b; 2015) for full details. Here we only consider rational closure of TBox, defined as follows.

**Definition 2 (Exceptionality of concepts and inclusions)** Let \( T_B \) be a TBox and \( C \) a concept. \( C \) is said to be exceptional for \( T_B \) if and only if \( T_B \models_{\text{ALC} + \text{T}_B} T(C) \subseteq \neg C \). A \( T \)-inclusion \( T(C) \subseteq D \) is exceptional for \( T_B \) if \( C \) is exceptional for \( T_B \). The set of \( T \)-inclusions of \( T_B \) which are exceptional in \( T_B \) will be denoted as \( E(T_B) \).

Given a DL TBox, it is possible to define a sequence of non increasing subsets of TBox ordered according to the exceptionality of the elements \( E_0 \supseteq E_1 \supseteq E_2 \ldots \) by letting \( E_0 = \text{TBox} \) and, for \( i > 0 \), \( E_i = \{ E_{i-1} \} \cup \{ C \subseteq D \in \text{TBox} \, \text{s.t.} \, T \) does not occur in \( C \} \). Observe that, being KB finite, there is a \( n \geq 0 \) such that, for all \( m > n \), \( E_m = E_n \) or \( E_n = \emptyset \). A concept \( C \) has rank \( i \) (denoted \( \text{rank}(C) = i \)) for TBox, if \( i \) is the least natural number for which \( C \) is not exceptional for \( E_i \). If \( C \) is exceptional for all \( E_i \) then \( \text{rank}(C) = \infty \) (\( C \) has no rank).

Rational closure builds on this notion of exceptionality:

**Definition 3 (Rational closure of TBox)** Let \( KB = \{ \text{TBox}, \text{ABox} \} \) be a DL knowledge base. The rational closure of TBox \( \text{TBox} = \{ T(C) \subseteq D \mid \text{either} \, \text{rank}(C) < \text{rank}(C \cap \neg D) \) or \( \text{rank}(C) = \infty \} \cup \{ C \subseteq D \mid KB \models_{\text{ALC} + \text{T}_B} C \subseteq D \} \), where \( C \) and \( D \) are ALC concepts.

As a very interesting property, in the context of DLs, the rational closure has a very interesting complexity: deciding if an inclusion \( T(C) \subseteq D \) belongs to the rational closure of TBox is a problem in \( \text{ExpTime} \) (Giordano et al. 2015).

In (Giordano et al. 2015) it is shown that the semantics corresponding to rational closure can be given in terms of minimal canonical \( \text{ALC} + \text{T}_n \) models. With respect to standard \( \text{ALC} + \text{T}_n \) models, in these models the rank of each domain element is as low as possible (each domain element is assumed to be as typical as possible). This is expressed by the following definition.

**Definition 4 (Minimal models of \( K \) (with respect to \( \text{TBox} \)))** Given \( M = (\Delta, <, I) \) and \( M' = (\Delta', <', I') \), we say that \( M \) is preferred to \( M' \) (\( M < M' \)) if: \( \Delta = \Delta' \), \( C^I = C'^I \) for all concepts \( C \), for all \( x \in \Delta \), it holds that \( k_M(x) \leq k_M'(x) \) whereas there exists \( y \in \Delta \) such that \( k_M(y) < k_M'(y) \).

Given a knowledge base \( K = \{ \text{TBox}, \text{ABox} \} \), we say that \( M \) is a minimal model of \( K \) (with respect to TBox) if it is a model satisfying \( K \) and there is no \( M' \) model satisfying \( K \) such that \( M' < M \).

Furthermore, the models corresponding to rational closure are canonical. This property, expressed by the following definition, is needed when reasoning about the (relative) rank of the concepts: it is important to have them all represented.

**Definition 5 (Canonical model)** Given \( K = (\text{TBox}, \text{ABox}) \), a model \( M = (\Delta, <, I) \) satisfying \( K \) is canonical if for each set of concepts \( \{ C_1, C_2, \ldots, C_n \} \) consistent with \( K \), there exists (at least) a domain element \( x \in \Delta \) such that \( x \in (C_1 \cap C_2 \cap \cdots \cap C_n) \).

**Definition 6 (Minimal canonical models (with respect to TBox))** \( M \) is a minimal canonical model of \( K \) minimal with respect to TBox if it satisfies \( K \), it is minimal with respect to TBox (Definition 4) and it is canonical (Definition 5).

The correspondence between minimal canonical models and rational closure is established by the following key theorem.

**Theorem 1 (Giordano et al. 2015)** Let \( K = (\text{TBox}, \text{ABox}) \) be a knowledge base and \( C \subseteq D \) a query. We have that \( C \subseteq D \in \text{TBox} \) if and only if \( C \subseteq D \) holds in all minimal canonical models of \( K \) with respect to TBox (Definition 6).

**Semantics with several preference relations**

The main weakness of rational closure, despite its power and its nice computational properties, is that it is an all-or-nothing mechanism that does not allow to separately reason on single aspects. To overcome this difficulty, we here consider models with several preference relations, one for each aspect we want to reason about. We assume this is any concept occurring in \( K \); we call \( A \) the set of these aspects (observe that \( A \) may be non-atomic). For each aspect \( A, <_A \) expresses the preference for aspect \( A : <_{\text{fly}} \) expresses the preference for flying, so if we know that \( T(Bird) \subseteq \text{fly} \), birds that do fly will be preferred to birds that do not fly, with respect to aspect fly, i.e. with respect to \( <_{\text{fly}} \). All these preferences, as well as the global preference relation \( < \), satisfy the properties in Definition 7 below. We now enrich the definition of an \( \text{ALC} + \text{T}_n \) model given above (Definition 1) by taking into account preferences with respect to all of the aspects. In the semantics we can express that for instance \( x <_{A_1} y \), whereas \( y <_{A_2} x \) (\( x \) is preferred to \( y \) for aspect \( A_1 \) but \( y \) is preferred to \( x \) for aspect \( A_2 \)).

This semantic richness allows to obtain a strengthening of rational closure in which typicality with respect to every aspect is maximized. Since we want to compare our approach to rational closure, we keep the language the same than in \( \text{ALC} + \text{T}_n \). In particular, we only have one single typicality operator \( T \). However, the semantic richness could motivate the introduction of several typicality operators \( T_{A_1}, \ldots, T_{A_n} \), by which one might want to explicitly talk in the language about the typicality w.r.t. aspect \( A_1 \), or \( A_2 \), and so on. We leave this extension for future work.

**Definition 7 (Enriched rational models)** Given a knowledge base \( K \), we call an enriched rational model a structure \( M = (\Delta, <, <_{A_1}, \ldots, <_{A_n}, I) \) where \( \Delta, I \) are defined as in Definition 1, and \( <_{A_1}, \ldots, <_{A_n} \) are preference relations over \( \Delta \), with the properties of being irreflexive, transitive, satisfying the finite chain condition, modular (for all \( x, y, z \in \Delta \) if \( x <_{A_1} y \) then either \( x <_{A_1} z \) or \( z <_{A_1} y \)).

For all \( <_{A_1} \) and for \( < \) it holds that \( \text{min}_{<_{A_1}}(S) = \{ x \in S \} \) and \( \text{min}_{<}(S) = \)
\{x \in S \text{ s.t. there is no } x_1 \in S \text{ s.t. } x_1 < x\} \text{ and } (T(C))^I = \min_{<}(C^I).

< \text{ satisfies the further conditions that } x < y \text{ if:}
(a) there is } A_i \text{ such that } x < A_i, \text{ y, and there is no } A_j \text{ such that } y < A_j \text{ or } x;
(b) \text{ there is } T(C_i) \subseteq A_k \text{ s.t. } y \in (C_i \cap \neg A_i)^I, \text{ and for all } T(C_j) \subseteq A_j \text{ s.t. } x \in (C_j \cap \neg A_j)^I, \text{ there is } T(C_k) \subseteq A_k \text{ s.t. } y \in (C_k \cap \neg A_k)^I \text{ and } k_M(C_j) < k_M(C_k).

In this semantics the global preference relation < is related to the various preference relations <_{A_i} \text{ relative to single aspects } A_i. \text{ Given (a) } x < y \text{ when } x \text{ is preferred to } y \text{ for a single aspect } A_i, \text{ and there is no aspect } A_j \text{ for which } y \text{ is preferred to } x. \text{ (b) captures the idea that case two individuals are preferred with respect to different aspects, preference (for the global preference relation) is given to the individual that satisfies all typical properties of the most specific concept (if } C_k \text{ is more specific than } C_j, \text{ then } k_M(C_j) < k_M(C_k)), \text{ as illustrated by Example 1 below.}

We insist in highlighting that this semantics somewhat complicated is needed since we want to provide a strengthening of rational closure. For this, we have to respect the constraints imposed by rational closure. One might think in the future to study a semantics in which only (a) holds. We have not considered such a simpler semantics since it would no longer be a strengthening of the semantics corresponding to rational closure, and is therefore out of the focus of this work.

In order to be a model of } K \text{ an } \mathcal{ALC}^\text{RT}_E \text{ model must satisfy the following constraints.}

**Definition 8 (Enriched rational models of } K)\text{ Given a knowledge base } K, \text{ an enriched rational model for } K \text{ is a model of } K \text{ if it satisfies both its } TBox \text{ and its } ABox, \text{ where } M \text{ satisfies } TBox \text{ if for all inclusions } C \subseteq A_i \in TBox: \text{ if } C \text{ does not occur in } C, \text{ then } C^I \subseteq A_i^I \text{ if } C \text{ occurs in } C, \text{ and } C = T(C'), \text{ then both (i) } \min_{<}(C^I) \subseteq A_i^I \text{ and (ii) } \min_{<}(C^I) \subseteq A_i^I, \text{ M satisfies } ABox \text{ if (i) for all } C(a) \text{ in } ABox, A_i^I \subseteq C^I, \text{ (ii) for all } aRb \text{ in } ABox, A_i^I \subseteq (aI, bI) \subseteq R^I.\text{ Example 1 Let } K = \{\text{Penguin} \subseteq \text{Bird}, T(\text{Bird}) \subseteq \text{HasNiceFeather}, T(\text{Bird}) \subseteq \text{Fly}, T(\text{Penguin}) \subseteq \neg \text{Fly}\}, \text{ } L_A = \{\text{HasNiceFeather}, \text{Fly}, \neg \text{Fly}, \text{Bird}, \text{Penguin}\}. \text{ We consider an } \mathcal{ALC}^\text{RT}_E \text{ model } M \text{ of } K, \text{ that we don’t fully describe but which we only use to observe the behavior of two Penguins } x, y \text{ with respect to the properties of (not) flying and having nice feather. In particular, let us consider the three preference relations: } \prec, \preceq_{\text{Fly}}, \prec_{\text{HasNiceFeather}}. \text{ Suppose } x \prec_{\text{Fly}} y \text{ (because } x, \text{ as all typical penguins, does not fly whereas } y \text{ exceptionally does) and there is no other aspect } A_i \text{ such that } y < A_i, x, \text{ and in particular it does not hold that } y <_{\text{HasNiceFeather}} x \text{ (because for instance both have a nice feather). In this case, obviously it holds that } x < y \text{ (since } (a) \text{ is satisfied).}

Consider now a more tricky situation in which again } x \prec_{\text{Fly}} y \text{ holds (because for instance } x \text{ does not fly whereas } y \text{ flies), } (x \text{ is a typical penguin for what concerns } \text{Flying}) \text{ but this time } y <_{\text{HasNiceFeather}} x \text{ holds (because for instance } y \text{ has a nice feather, whereas } x \text{ has not). So } x \text{ is preferred to } y \text{ for a given aspect whereas } y \text{ is preferred to } x \text{ for another aspect. However, } x \text{ enjoys the typical properties of penguins, and violates the typical properties of birds, whereas } y \text{ enjoys the typical properties of birds and violates those of penguins. Being concept Penguin more specific than concept Bird, we prefer } x \text{ to } y, \text{ since we prefer the individuals that inherit the properties of the most specific concepts of which they are instances. This is exactly what we get: by (b) } x < y \text{ holds.}

Logical entailment for } \mathcal{ALC}^\text{RT} \text{ is defined as usual: a query (with form } C_L(a) \text{ or } C_E \subseteq C_R) \text{ is logically entailed by } K \text{ if it holds in all models of } K, \text{ as stated by the following definition. The following theorem shows the relations between } \mathcal{ALC}^\text{RT} \text{ and } \mathcal{ALC} + \text{ Tn}. \text{ Proofs are omitted due to space limitations.}

**Theorem 2** If } K \models_{\mathcal{ALC}^\text{RT}} T_E \text{ then also } K \models_{\mathcal{ALC} + \text{ Tn}} F. \text{ If } T \text{ does not occur in } F \text{ the other direction also holds: If } K \models_{\mathcal{ALC} + \text{ Tn}} F \text{ then also } K \models_{\mathcal{ALC}^\text{RT}} T_E.

The following example shows that } \mathcal{ALC}^\text{RT} \text{ alone does not allow us to perform the stronger inferences with respect to rational closure mentioned in the Introduction (and in particular, it does not allow to infer (**)), that typical penguins have a nice feather.}

**Example 2** Consider the above Example 1. As said in the Introduction, in rational closure we are not able to reason separately about the property of flying or not flying, and the property of having or not having a nice feather. Since penguins are extraordinary birds with respect to the property of flying, in rational closure which is an all-or-nothing mechanism, they do not inherit any of the properties of typical birds. In particular, they do not inherit the property of having a nice feather, even if this property and the fact of flying are independent from each other and there is no reason why being exceptional with respect to one property should block the inheritance of the other one. Does our enriched semantics enforce the separate inheritance of independent properties? Consider a model } M \text{ in which we have } \Delta = \{x, y, z\}, \text{ where } x \text{ is a bird (not a penguin) that flies and has a nice feather } (x \in \text{Bird}, x \in \text{Fly}, y \notin \text{HasNiceFeather}), \text{ y is a penguin that does not fly and has a nice feather } (y \in \text{Penguin}, y \in \text{Bird}, y \notin \text{Fly}, y \in \text{HasNiceFeather}), \text{ z is a penguin that does not fly and has no nice feather } (z \in \text{Penguin}, z \in \text{Bird}, z \notin \text{Fly}, z \notin \text{HasNiceFeather}). \text{ Suppose it holds that } x <_{\prec} y, x <_{\preceq_{\text{Fly}}} z, x <_{\prec_{\text{HasNiceFeather}}} y, y <_{\prec_{\text{HasNiceFeather}}} z, z <_{\prec} y, z <_{\preceq_{\text{Fly}}} x, y <_{\preceq_{\text{HasNiceFeather}}} z. \text{ It can be verified that this is an } \mathcal{ALC}^\text{RT} \text{ model, satisfying } T(\text{Penguin}) \subseteq \text{HasNiceFeather} \text{ (since the only typical Penguin is } y, \text{ instance of HasNiceFeather).}

Unfortunately, this is not the only } \mathcal{ALC}^\text{RT} \text{ model of } K. \text{ For instance there can be } M \text{ equal to } M \text{ except from the fact that } y <_{\text{HasNiceFeather}} z \text{ does not hold, nor } y < z \text{ holds. It can be easily verified that this is also an } \mathcal{ALC}^\text{RT} \text{ model of } K \text{ in which } T(\text{Penguin}) \subseteq \text{HasNiceFeather.}
does not hold (since now also \( z \) is a typical Penguin, and \( z \) is not an instance of HasNiceFeather).

This example shows that although there are \( \mathcal{ALC}^{\mathcal{R}_E} \) models satisfying well suited inclusions, the logic is not strong enough to limit our attention to these models. We would like to constrain our logic in order to exclude models like \( \mathcal{M}' \). Roughly speaking, we want to eliminate \( \mathcal{M}' \) because it is not minimal: although the model as it is satisfies \( K \), so \( y \) does not need to be preferred to \( z \) to satisfy \( K \) (neither with respect to \( < \) nor with respect to \(<_{\text{HasNiceFeather}} \)), intuitively we would like to prefer \( y \) to \( z \) (with respect to the property HasNiceFeather, whence in this case with respect to the global \( < \)), since \( y \) does not falsify any of the inclusions with HasNiceFeather, whereas \( z \) does. This is obtained by imposing the constraint of considering only models minimal with respect to all relations \( <_A \), as defined in Definition 10 below. Notice that the wanted inference does not hold in \( \mathcal{ALC} + \mathcal{T}_E \) minimal canonical models corresponding to rational closure: in these models \( y < z \) does never hold (the two elements have the same rank) and this semantics does not allow us to prefer \( y \) to \( z \). By adopting the restriction to minimal canonical models, we obtain a semantics which is stronger than rational closure (and therefore enforces all conclusions enforced by rational closure) and, furthermore, separately allows to reason on different aspects.

Before we end the section, similarly to what done above, let us introduce a rank of a domain element with respect to an aspect. We will use this notion in the following section.

**Definition 9** The rank \( k_{A_i M} (x) \) of a domain element \( x \) with respect to \( <_{A_i} \) in \( M \) is the length of the longest chain \( x_0 <_{A_i} x_1 <_{A_i} \cdots <_{A_i} x_n = x \) from \( x \) to a minimal \( x_0 \) (s.t. for no \( x' <_{A_i} x' \) there exists \( x \)). To refer to the rank of an element \( x \) with respect to the preference relation \( < \) we will simply write \( k_M(x) \).

The notion just introduced will be useful in the following. Since \( k_{A_i M} \) and \( <_{A_i} \) are clearly interdefinable (by the previous definition and the properties of \( <_{A_i} \), it easily follows that in all enriched models \( M, x <_{A_i} y \) iff \( k_{A_i M} (x) < k_{A_i M} (y) \), and \( x < y \) iff \( k_M (x) < k_M (y) \). We will shift from one to other whenever this simplifies the exposition.

**Nonmonotonicity and relation with rational closure**

We here define a minimal models mechanism starting from the enriched models of the previous section. With respect to the minimal canonical models used in (Giordano et al. 2015) we define minimal models by separately minimizing all the preference relations with respect to all aspects (steps (i) and (ii) in the definition below), before minimizing \( < \) (steps (iii) and (iv) in the definition below). By the constraints linking \( < \) to the preference relations \( <_{A_i}, \cdots <_{A_n} \), this leads to preferring (with respect to the global \( < \)) the individuals that are minimal with respect to all \( <_{A_i} \) for all aspects \( A_i \), or to aspects of most specific categories than of more general ones. It turns out that this leads to a stronger semantics than what is obtained by directly minimizing \( < \).

**Definition 10** (Minimal Enriched Models) Given two \( \mathcal{ALC}^{\mathcal{R}_E} \) enriched models \( M = (\Delta, <_{A_1}, \ldots, <_{A_n}, <_{I}) \) and \( M' = (\Delta', <'_{A_1}, \ldots, <'_{A_n}, <'_{I}) \) we say that \( M' \) is preferred to \( M \) with respect to the single aspects (and write \( M' <_{\text{Enriched, Aspects}} M \)) if \( \Delta = \Delta' \), \( I = I' \), and:

- (i) for all \( x \in \Delta \), for all \( A_i \), \( k_{A_i M} (x) \leq k_{A_i M} (x) \);
- (ii) for some \( y \in \Delta \), for some \( A_i \), \( k_{A_i M} (y) < k_{A_i M} (y) \);

We let the set \( \text{Min}_{\text{Aspects}} \) to be the set \( \{ M : \) there is no \( M' \) that \( M' <_{\text{Enriched, Aspects}} M \) \}. Given \( M \) and \( M' \in \text{Min}_{\text{Aspects}} \), we say that \( M' \) is overall preferred to \( M \) (and write \( M' <_{\text{Enriched}} M \)) if \( \Delta = \Delta' \), \( I = I' \), and:

- (iii) for all \( x \in \Delta \), \( k_{M'} (x) \leq k_{M} (x) \);
- (iv) for some \( y \in \Delta \), \( k_{M'} (y) < k_{M} (y) \);

We call \( M \) a minimal model of \( K \) if \( M \in \text{Min}_{\text{Aspects}} \). If \( M \in \text{Min}_{\text{Aspects}} \) then \( M \) is a minimal model of \( K \) and there is no \( M' \in \text{Min}_{\text{Aspects}} \) that \( M' <_{\text{Enriched}} M \).

K minimally entails a query \( F \) if \( F \) holds in all minimal \( \mathcal{ALC}^{\mathcal{R}_E} \) models of \( K \). We write \( K \models \mathcal{ALC}^{\mathcal{R}_E} F \). We have developed the semantics above in order to overcome a weakness of rational closure, namely its all-or-nothing character. In order to show that the semantics hits the point, we show here that the semantics is stronger than the one corresponding to rational closure. Furthermore, Example 3 below shows that indeed we have strengthened rational closure by making it possible to separately reason on the different properties. Since the semantic characterization of rational closure is given in terms of rational canonical models, here we restrict our attention to enriched rational models which are canonical.

**Definition 11** (Minimal canonical enriched models of K) An \( \mathcal{ALC}^{\mathcal{R}_E} \) enriched model \( M \) is a minimal canonical enriched model of \( K \) if it satisfies \( K \), is minimal (with respect to Definition 10) and is canonical: for all the sets of concepts \( \{ C_1, C_2, \ldots, C_n \} \) s.t. \( K \models \mathcal{ALC}^{\mathcal{R}_E} C_1 \cap C_2 \cap \cdots \cap C_n \), there exists (at least) a domain element \( x \) such that \( x \in \{ C_1 \cap C_2 \cap \cdots \cap C_n \}^M \).

We call \( \mathcal{ALC}^{\mathcal{R}_E} + \text{min – canonical} \) the semantics obtained by restricting attention to minimal canonical enriched models. In the following we will write:

\[ K \models \mathcal{ALC}^{\mathcal{R}_E} + \text{min – canonical} \] \( C \subseteq D \) to mean that \( C \subseteq D \) holds in all minimal canonical enriched models of \( K \). The following example shows that this semantics allows us to correctly deal with the wanted inferences of the Introduction, as (**). The fact that the semantics \( \mathcal{ALC}^{\mathcal{R}_E} + \text{min – canonical} \) is a genuine strengthening of the semantics corresponding to rational closure is formally shown in Theorem 3 below.

**Example 3** Consider any minimal canonical model \( M^* \) of the same \( K \) used in Example 1. It can be easily verified that \( M^* \) is a domain element \( y \) which is a penguin that does not fly and has a nice feather \( (y \in \text{Penguin}, y \in \text{Bird}^3, y \in \text{HasNiceFeather}^3) \).

First, it can be verified that \( y \in \text{min}_{\text{A}}(\text{Penguin}) \) (by Definition 7, and since by minimality of \( <_{\text{Fly}} \) and...
The following theorem is the important technical result of the paper:

**Theorem 3** The minimal models semantics $\text{ALC}^\text{R}_T E + \min$ – canonical is stronger than the semantics for rational closure. Let $(K = TBox, ABox)$, If $C \subseteq D \in TBox$ then $K \models_{\text{ALC}^\text{R}_T E + \min - \text{canonical}} T(Penguin) \sqsubseteq \text{HasNiceFeather}$, i.e., $K \models_{\text{ALC}^\text{R}_T E + \min - \text{canonical}} T(\text{Penguin}) \sqsubseteq \text{HasNiceFeather}$, which was the wanted inference (**) of the Introduction.

The theorem follows by contraposition.

**Conclusions and Related Works**

A lot of work has been done in order to extend the basic formalism of Description Logics (DLs) with nonmonotonic reasoning features (Straccia 1993; Baader and Hollunder 1995; Donini, Nardi, and Rosati 2002; Eiter et al. 2004; Giordano et al. 2007; 2013a; Ke and Sattler 2008; Britz, Heidema, and Meyer 2008; Bonatti, Lutz, and Wolter 2009; Casini and Straccia 2010; Motik and Rosati 2010; Krissnadhi, Sengupta, and Hitzler 2011; Knorr, Hitzler, and Maier 2012; Casini et al. 2013). The purpose of these extensions is to allow reasoning about prototypical properties of individuals or classes of individuals.

The interest of rational closure for DLs is that it provides a significant and reasonable skeptical nonmonotonic inference mechanism, while keeping the same complexity as the underlying logic. The first notion of rational closure for DLs was defined by Casini and Straccia (Casini and Straccia 2010). Their rational closure construction for $\text{ALC}$ directly uses entailment in $\text{ALC}$ over a materialization of the KB. A variant of this notion of rational closure has been studied in (Casini et al. 2013), and a semantic characterization for it has been proposed. In (Giordano et al. 2013b; 2015) a notion of rational closure for the logic $\text{ALC}$ has been proposed, building on the notion of rational closure proposed by Lehmann and Magidor (Lehmann and Magidor 1992), together with a minimal model semantics characterization.

It is well known that rational closure has some weaknesses that accompany its well-known qualities, both in the context of propositional logic and in the context of Description Logics. Among the weaknesses is the fact that one cannot separately reason property by property, so that, if a subclass of $C$ is exceptional for a given aspect, it is exceptional “tout court” and does not inherit any of the typical properties of $C$. Among the strengths of rational closure there is its computational lightness, which is crucial in Description Logics. To overcome the limitations of rational closure, in (Casini and Straccia 2011; 2013) an approach is introduced based on the combination of rational closure and Defeasible Inheritance Networks, while in (Casini and Straccia 2012) a lexicographic closure is proposed, and in (Casini et al. 2014) relevant closure, a syntactic stronger version of rational closure. To address the mentioned weakness of rational closure, in this paper we have proposed a finer grained semantics of the semantics for rational closure proposed in (Giordano et al. 2015), where models are equipped with several preference relations. In such a semantics it is possible to relativize the notion of typicality, whence to reason about typical properties independently from each other. We are currently working at the formulation of a syntactic characterization of the semantics which will be a strengthening of rational closure. As the semantics we have proposed provides a strengthening of rational closure, a natural question arises whether this semantics is equivalent to the lexicographic closure proposed in (Lehmann 1995). In particular, lexicographic closure construction for the description logic $\text{ALC}$ has been defined in (Casini and
Concerning our Example 3 above, our minimal model semantics gives the same results as lexicographic closure, since $T(Penguin) \sqsubseteq \text{HasNiceFeather}$ can be derived from the lexicographic closure of the TBox and $T(Penguin) \sqsubseteq \text{HasNiceFeather}$ holds in all the minimal canonical enriched models of TBox. However, a general relation needs to be established.

An approach related to our approach is given in (Gil 2014), where it is proposed an extension of $\text{ALC} + T$ with several typicality operators, each corresponding to a preference relation. This approach is related to ours although different: the language in (Gil 2014) allows for several typicality operators whereas we only have a single typicality operator. The focus of (Gil 2014) is indeed different from ours, as it does not deal with rational closure, whereas this is the main contribution of our paper.

**Acknowledgement:** This research is partially supported by INDAM-GNCS Project 2016 “Rationamento Defeasibile nelle Logiche Descrittive”.

**References**


Distributing Knowledge into Simple Bases

Adrian Haret and Jean-Guy Mailly and Stefan Woltran
Institute of Information Systems
TU Wien, Austria
{haret,jmailly,woltran}@dbai.tuwien.ac.at

Abstract
Understanding the behavior of belief change operators for fragments of classical logic has received increasing interest over the last years. Results in this direction are mainly concerned with adapting representation theorems. However, fragment-driven belief change also leads to novel research questions. In this paper we propose the concept of belief distribution, which can be understood as the reverse task of merging. More specifically, we are interested in the following question: given an arbitrary knowledge base $K$ and some merging operator $\Delta$, can we find a profile $E$ and a constraint $\mu$, both from a given fragment of classical logic, such that $\Delta_\mu(E)$ yields a result equivalent to $K$? In other words, we are interested in seeing if $K$ can be distributed into knowledge bases of simpler structure, such that the task of merging allows for a reconstruction of the original knowledge. Our initial results show that merging based on Hamming distance relies heavily on the fragment of choice.

Introduction
Belief change and belief merging have been topics of interest in Artificial Intelligence for three decades (Alchourrón, Gärdenfors, and Makinson 1985; Katsuno and Mendelzon 1991; Konieczny and Pino Pérez 2002). However, the restriction of such operators to specific fragments of propositional logic has received increasing attention only in the last years (Delgrande et al. 2013; Creignou et al. 2014a; 2014b; Zhuang and Pagnucco 2012; Zhuang, Pagnucco, and Zhang 2013; Zhuang and Pagnucco 2014; Delgrande and Peppas 2015; Haret, Rümmele, and Woltran 2015). Mostly, the question tackled in these works is “How should rationality postulates and change operators be adapted to ensure that the result of belief change belongs to a given fragment?”. Surprisingly, the question concerning the extent to which the result of a belief change operation can deviate from the fragment under consideration has been neglected so far. In order to tackle this question, we focus here on a certain form of reverse merging. The question is, given an arbitrary knowledge base $K$ and some IC-merging (i.e. merging with integrity constraint, see (Konieczny and Pino Pérez 2002)), operator $\Delta$ can we find a profile $E$, i.e. a tuple of knowledge bases, and a constraint $\mu$, both from a given fragment of classical logic, such that $\Delta_\mu(E)$ yields a result equivalent to $K$? In other words, we are interested in seeing if $K$ can be distributed into knowledge bases of simpler structure, such that the task of merging allows for a reconstruction of the original knowledge. We call this operation knowledge distribution.

Studying the concept of knowledge distribution can be motivated from different points of view. First, consider a scenario where the storage devices have limited expressibility, for instance, databases or logic programs. Our analysis will show which merging operators are required to reconstruct arbitrary knowledge stored in such a set of limited devices. Second, distribution can also be understood as a tool to hide information; only users who know the used merging operator (which thus acts as an encryption key) are able to faithfully retrieve the distributed knowledge. Given the high complexity of belief change (even for revision in “simple” fragments like Horn and 2CNF (Eiter and Gottlob 1992; Liberatore and Schaerf 2001; Creignou, Pichler, and Woltran 2013)), brute-force attack to guess the merging operator is unthinkable. Finally, from the theoretical perspective our results shed light on the power of different merging operators when applied to profiles from certain fragments. In particular, our results show that merging 1CNF formulas via the Hamming-distance based operator $\Delta^{H,\Sigma}$ does not need additional care, since the result is guaranteed to stay in the fragment.

Related Work. Previous work on merging in fragments of propositional logic proposed an adaptation of existing belief merging operators to ensure that the result of merging belongs to a given fragment (Creignou et al. 2014b), or modified the rationality postulates in order to function in the Horn fragment (Haret, Rümmele, and Woltran 2015). Our approach is different, since we do not require that the result of merging stays in a given fragment. On the contrary, we want to decompose arbitrary bases into a fragment profile. Recent work by Liberatore has also addressed a form of meta-reasoning over belief change operators. In (Liberatore 2015a), the input is a profile of knowledge bases with
the expected result of merging $R$, and the aim is to determine the reliability of the bases (for instance, represented by weights) which allow the obtaining of $R$. In another paper, Liberatore (2015b) identifies, given a sequence of belief revisions and their results, the initial pre-order which characterizes the revision operator. Finally, even if our approach may seem related to Knowledge Compilation (KC) (Darwiche and Marquis 2002; Fargier and Marquis 2014; Marquis 2015), both methods are in fact conceptually different. KC aims at modifying a knowledge base $K$ into a knowledge base $K'$ such that the most important queries for a given application (consistency checking, clause entailment, model counting, ...) are simpler to solve with $K'$. Here, we are interested in the extent to which it is possible to equivalently represent an arbitrary knowledge base by simpler fragments when using merging as a recovery operation.

**Main Contributions.** We formally introduce the concept of knowledge distributability, as well as a restricted version of it where the profile is limited to a single knowledge base (simplifiability). We show that for drastic distance arbitrary knowledge can be distributed into bases restricted to mostly any kind of fragment, while simplifiability is limited to trivial cases. On the other hand, for Hamming-distance based merging the picture is more opaque. We show that for $1CNF$, distributability w.r.t. $\Delta_{H,S}$ is limited to trivial cases, while slightly more can be done with $\Delta_{H,GMin}$ and $\Delta_{H,GMax}$. For $2CNF$ we show that arbitrary knowledge can be distributed and even be simplified. Finally, we discuss the $Horn$ fragment for which the results for $\Delta_{H,S}$, $\Delta_{H,GMin}$ and $\Delta_{H,GMax}$ are situated in between the two former fragments.

**Background**

**Fragments of Propositional Logic.** We consider $\mathcal{L}$ as the language of propositional logic over some fixed alphabet $\mathcal{U}$ of propositional atoms. We use standard connectives $\lor$, $\land$, $\neg$, and constants $\top$, $\bot$. A clause is a disjunction of literals. A clause is called $Horn$ if at most one of its literals is positive. An interpretation is a set of atoms (those set to true). The set of all interpretations is $2^\mathcal{U}$. Models of a formula $\varphi$ are denoted by $Mod(\varphi)$. A knowledge base (KB) is a finite set of formulas and we identify models of a KB $K$ via $Mod(K) = \bigcap_{\varphi \in K} Mod(\varphi)$. A profile is a finite non-empty tuple of KBs. Two formulae $\varphi_1$, $\varphi_2$ (resp. KBs $K_1$, $K_2$) are equivalent, denoted $\varphi_1 \equiv \varphi_2$ (resp. $K_1 \equiv K_2$), when they have the same set of models.

We use a rather general and abstract notion of fragments that (1) for all $\psi \in \mathcal{L}$, $Mod(\psi) = Cl(\mathcal{N}(\psi))$ and (2) for all $\mathcal{M} \subseteq 2^\mathcal{U}$ there is a $\psi \in \mathcal{L}$ with $Mod(\psi) = Cl(\mathcal{M})$. We often denote the closure-operator $Cl$ associated to a fragment $\mathcal{L}$ as $Cl_{\mathcal{L}}$.

**Definition 3.** For a fragment $\mathcal{L'}$, we call a finite set $K \subseteq \mathcal{L'}$ an $\mathcal{L'}$-knowledge base. An $\mathcal{L'}$-profile is a profile over $\mathcal{L'}$-knowledge bases. A KB $K' \subseteq \mathcal{L}$ is called $\mathcal{L'}$-expressible if there exists an $\mathcal{L'}$-KB $K$, such that $K' \equiv K$.

Many well known fragments of propositional logic are indeed captured by our notion. For the Horn-fragment $\mathcal{L}_{Horn}$, i.e. the set of all conjunctions of Horn clauses over $\mathcal{U}$, take the operator $Cl_{\mathcal{L}_{Horn}}$ defined as the fixed point of the function

$$Cl_{\mathcal{L}_{Horn}}(M) = \{\omega_1 \cap \omega_2 \mid \omega_1, \omega_2 \in M\}.$$

The fragment $\mathcal{L}_{2CNF}$ which is restricted to formulas over clauses of length at most 2 is linked to the operator $Cl_{\mathcal{L}_{2CNF}}$ defined as the fixed point of the function $Cl_{\mathcal{L}_{2CNF}}$ given by

$$Cl_{\mathcal{L}_{2CNF}}(M) = \{maj_3(\omega_1, \omega_2, \omega_3) \mid \omega_1, \omega_2, \omega_3 \in M\}.$$

Here, we use the ternary majority function $maj_3(\omega_1, \omega_2, \omega_3)$ which yields an interpretation containing those atoms which are true in at least two out of $\omega_1, \omega_2, \omega_3$. Finally, we are also interested in the $L_{1CNF}$ fragment which is just composed of conjunctions of literals; its associated operator $Cl_{\mathcal{L}_{1CNF}}$ is defined as the fixed point of the function

$$Cl_{\mathcal{L}_{1CNF}}(M) = \{\omega_1 \cap \omega_2 \cup \omega_3 \mid \omega_1, \omega_2 \in M\} \cup \{\omega_3 \mid \omega_1 \subseteq \omega_3 \subseteq \omega_2; \omega_1, \omega_2 \in M\}.$$

Note that full classical logic is given via the identity closure operator $Cl_{\mathcal{L}}(M) = M$.

**Merging Operators.** We focus on $IC$-merging, where a profile is mapped into a KB, such that the result satisfies some integrity constraint. Postulates for IC-merging have been stated in (Konieczny and Pino Pérez 2002). We restate them here for completeness.

**Definition 4.** A distance between interpretations is a mapping $d$ from two interpretations to a non-negative real number, such that for all $\omega_1, \omega_2, \omega_3 \in \mathcal{U}$, (1) $d(\omega_1, \omega_2) = 0$ iff $\omega_1 = \omega_2$; (2) $d(\omega_1, \omega_2) = d(\omega_2, \omega_1)$; and (3) $d(\omega_1, \omega_2) + d(\omega_2, \omega_3) \geq d(\omega_1, \omega_3)$. We will use two specific distances:

1. **drastic distance** $D(\omega_1, \omega_2) = 1$ if $\omega_1 = \omega_2, 0$ otherwise;
2. **Hamming distance** $H(\omega_1, \omega_2) = \|\omega_1 \setminus \omega_2 \cup (\omega_2 \setminus \omega_1)\|.$

We overload the previous notations to define the distance between an interpretation $\omega$ and a KB $K$: if $d$ is a distance between interpretations, then

$$d(\omega, K) = \min_{\omega' \in Mod(K)} d(\omega, \omega').$$

Next, an aggregation function must be used to evaluate the distance between an interpretation and a profile.

**Definition 5.** An aggregation function $\otimes$ associates a non-negative number to every finite tuple of non-negative numbers, such that:
1. If $x \leq y$, then $\otimes(x_1, \ldots, x, \ldots, x_n) \leq \otimes(x_1, \ldots, y, \ldots, x_n)$;
2. $\otimes(x_1, \ldots, x_n) = 0$ iff $x_1 = \cdots = x_n = 0$;
3. For every non-negative number $x$, $\otimes(x) = x$.

As aggregation functions, we will consider the sum $\Sigma$, and $GMax$ and $GMin$, defined as follows. Given a profile $(K_1, \ldots, K_n)$, let $V_\omega = (d_1^\omega, \ldots, d_n^\omega)$ be the vector of distances s.t. $d_i^\omega = d(\omega, K_i)$. $GMax(d_1^\omega, \ldots, d_n^\omega)$ (resp. $GMin(d_1^\omega, \ldots, d_n^\omega)$) is defined by ordering $V_\omega$ in decreasing (resp. increasing) order. Given two interpretations $\omega_1, \omega_2$, $GMax(d_1^\omega, \ldots, d_n^\omega) \leq GMax(d_1^{\omega_1}, \ldots, d_n^{\omega_1})$ (resp. $GMin(d_1^\omega, \ldots, d_n^\omega) \leq GMin(d_1^{\omega_1}, \ldots, d_n^{\omega_1})$) is defined by comparing them w.r.t. the lexicographic ordering.

Finally, let $d$ be a distance, $\omega$ an interpretation and $E = (K_1, \ldots, K_n)$ a profile. Then,
$$d^\otimes(\omega, E) = \otimes(d(\omega, K_1), \ldots, d(\omega, K_n)).$$

If there is no ambiguity about the aggregation function $\otimes$, we write $d(\omega, E)$ instead of $d^\otimes(\omega, E)$.

**Definition 6.** For any distance $d$ between interpretations, and any aggregation function $\otimes$, the merging operator $\Delta^d,\otimes$ is a mapping from a profile $E$ and a formula $\mu$ to a KB, such that
$$Mod(\Delta^d,\otimes_{\mu}(E)) = \min(\{Mod(\mu), \leq_{E}\otimes\}),$$
with $\omega_1 \leq_{E} \omega_2$ iff $d^\otimes(\omega_1, E) \leq d^\otimes(\omega_2, E)$.

When we consider a profile containing a single knowledge base $K$, all aggregation functions are equivalent; we write $\Delta^d_1(K)$ instead of $\Delta^d,\otimes_{\mu}(K)$ for readability. For drastic distance, $GMax$, $GMin$, and $\Sigma$ are equivalent for arbitrary profiles. Thus, whenever we show results for $\Delta^d,\Sigma$, these carry over to $\Delta^d,GMin$ and $\Delta^d,GMax$.

**Main Concepts and General Results**

We now give the central definition for a knowledge base being distributable into a profile from a certain fragment with respect to a given merging operator.

**Definition 7.** Let $\Delta$ be a merging operator, $K \subseteq \mathcal{L}$ be an arbitrary KB, and $\mathcal{L}'$ be a fragment. $K$ is called $\mathcal{L}'$-distributable w.r.t. $\Delta$ if there exists an $\mathcal{L}'$-profile $E$ and a formula $\mu$ in $\mathcal{L}'$, such that $\Delta^d_{\mu}(E) \equiv K$.

**Example 1.** Let $\mathcal{U} = \{a, b\}$ and consider $K = \{a \lor b\}$ which we want to check for $\mathcal{L}$-distributability w.r.t. operator $\Delta^H,\Sigma$. We have $Mod(K) = \{\{\}, \{a\}, \{b\}, \{a, b\}\}$, thus $K$ is not $\mathcal{L}$-expressible (note that $CL_{\mathcal{L}}(Mod(K)) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\} \neq Mod(K)$), otherwise $K$ would be distributable in a simple way (see Proposition 1 below).

Take the $\mathcal{L}$-profile $E = (K_1, K_2)$ with $K_1 = \{a \land b\}$, $K_2 = \{a \lor b\}$, together with the empty constraint $\mu = a \lor \neg a$. We have $Mod(K_1) = \{\{a, b\}\}$, $Mod(K_2) = \{\{a\}, \{b\}, \emptyset\}$. In the following matrix, each line corresponds to the distance between a model of $\mu$ and a KB from the profile $E$ (columns $K_1$ and $K_2$), or between a model of $\mu$ and the profile using the sum-aggregation over the distances to the single KBs (column $\Sigma$).

<table>
<thead>
<tr>
<th></th>
<th>$K_1$</th>
<th>$K_2$</th>
<th>$\Sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${a, b}$</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>${a}$</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>${b}$</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$\emptyset$</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

We observe that $Mod(\Delta^{H,\Sigma}_\mu(E)) = \{\{a\}, \{b\}, \{a, b\}\}$, thus $\Delta^{H,\Sigma}_\mu(E) \equiv K$ as desired. It is easily checked that also other aggregations work: $\Delta^{H,GMin}_\mu(E) \equiv \Delta^{H,GMax}_\mu(E) \equiv K$.

Next, we recall that IC-merging of a single KB yields revision. Thus, the concept we introduce next is also of interest, as it represents a certain form of reverse revision.

**Definition 8.** Let $\Delta$ be a merging operator, $K \subseteq \mathcal{L}$ an arbitrary KB, and $\mathcal{L}'$ a fragment. $K$ is called $\mathcal{L}'$-simplifiable w.r.t. $\Delta$ if there exists an $\mathcal{L}'$-KB $K'$ and $\mu \in \mathcal{L}'$, such that $\Delta^d_{\mu}(K') \equiv K$.

As we will see later, the KB $K$ from Example 1 cannot be $\mathcal{L}$-simplifiable. In other words, we need here at least two KBs to “express” $K$. However, it is rather straightforward that any $\mathcal{L}'$-expressible KB can be $\mathcal{L}'$-simplifiable.

**Proposition 1.** For every fragment $\mathcal{L}'$ and every KB $K$, it holds that $K$ is $\mathcal{L}'$-simplifiable (and thus also $\mathcal{L}'$-distributable) w.r.t. $\Delta$, whenever $K$ is $\mathcal{L}'$-expressible.

**Proof.** Let $K'$ be an $\mathcal{L}'$-KB equivalent to $K$, and let $\mu = \{\phi_{v \in K'}\varphi\}$. Thus, $\mu \in \mathcal{L}'$ by definition of fragments and it is easily verified that $\Delta^d_{\mu}(K') \equiv K$.

Next, we show that in order to determine whether a KB $K$ is $\mathcal{L}'$-distributable, it is sufficient to consider constraints $\mu$ such that $Mod(\mu) = CL_{\mathcal{L}'}(Mod(K))$.

**Proposition 2.** Let $K \subseteq \mathcal{L}$ be a KB, $\mathcal{L}'$ be a fragment, $E$ an $\mathcal{L}'$-profile and $\mu \in \mathcal{L}'$. Then $\Delta^d_{\mu}(E) \equiv K$ implies $\Delta^d_{\mu'}(E) \equiv K$ for any $\mu'$ such that $Mod(\mu') = CL_{\mathcal{L}'}(Mod(K))$.

**Proof.** Let $\Delta = \Delta^{d,\otimes}$. By Definition 6, $Mod(\Delta^d_{\mu}(E)) = \min(\{Mod(\mu), \leq_{E}\otimes\}, \{\emptyset, \{a\}, \{b\}, \{a, b\}\})$, hence $Mod(K) \subseteq Mod(\mu)$. Moreover, $\mu$ is $\mathcal{L}'$-closed, so $CL_{\mathcal{L}'}(Mod(K)) = Mod(\mu') \subseteq Mod(\mu)$. We get $Mod(K) \subseteq Mod(\mu') \subseteq Mod(\mu)$, thus $Mod(K) = \min(\{Mod(\mu'), \leq_{E}\otimes\}, \{\emptyset, \{a\}, \{b\}, \{a, b\}\})$, i.e. $\Delta^d_{\mu'}(E) \equiv K$.

Next, we give two positive results for distributing knowledge in any fragment. The key idea is to use KBs in the profile which have exactly one model (our notion of fragment guarantees existence of such KBs). The first result is independent of the distance notion but requires $GMin$ as the aggregation function. The second result is for drastic distance and thus works for any of the aggregation functions we consider.
Theorem 3. Let $d$ be a distance and $\mathcal{L}'$ be a fragment. Then for every KB $K$, such that for all distinct $\omega_1, \omega_2 \in \text{Mod}(K)$, $d(\omega_1, \omega_2) = \epsilon$ for some $\epsilon > 0$, it holds that $K$ is $\mathcal{L}'$-distributable w.r.t. $\Delta^G_{\text{GMin}}$.

Proof. Build the $\mathcal{L}'$-profile $E$ such that for each $\omega \in \text{Mod}(K)$, there is a KB with $\omega$ as its only model. Thus all models of $K$ get a $\text{GMin}$-vector $(0, e, e, e, \ldots)$. All interpretations from $\text{Cl}_{\mathcal{L}'}(\text{Mod}(K)) \setminus \text{Mod}(K)$ get a vector $(f, g, \ldots)$ with $f > 0$. Hence, we have $\min\text{Mod}(\mu)_{E} \leq d_{G_{\text{Min}}}(\omega, \mu) = \text{Mod}(K)$ using $\mu \in \mathcal{L}'$ with $\text{Mod}(\mu) = \text{Cl}_{\mathcal{L}'}(\text{Mod}(K))$. \qed

Theorem 4. For every fragment $\mathcal{L}'$ and every knowledge base $K$, it holds that $K$ is $\mathcal{L}'$-distributable w.r.t. $\Delta^D$, for $+ \in \{\Sigma, \text{GMin}, \text{GMax}\}$.

Proof. Given a fragment $\mathcal{L}'$, we take $E = \{K_0 \mid \omega \in \text{Mod}(K)\}$ where $K_0 \in \mathcal{L}'$ is a knowledge base with single model $\omega$ (such $K_0 \in \mathcal{L}'$ exists due to our definition of fragments), and let $\mu$ be such that $\text{Mod}(\mu) = \text{Cl}_{\mathcal{L}'}(\text{Mod}(K))$; hence also $\mu \in \mathcal{L}'$. Let $\omega' \in \text{Mod}(\mu)$ and $n = |\text{Mod}(K)|$, we observe that $\Sigma K_{\omega_i \in E} \omega'(K_0) = n - 1$ when $\omega' \in \text{Mod}(K)$, and $n$ otherwise. Thus, $\Delta^D_{\Sigma}(E) = K$. The same result holds for $\Delta^D_{\Sigma \text{GMin}}$ and $\Delta^D_{\Sigma \text{GMax}}$. \qed

Concerning simplifiability w.r.t. drastic distance based operators, Proposition 1 cannot be improved.

Theorem 5. For every fragment $\mathcal{L}'$ and every KB $K$, $K$ is $\mathcal{L}'$-simplifiable w.r.t. $\Delta^D$ iff $K$ is $\mathcal{L}'$-expressive.

Proof. The $(\Rightarrow)$-direction is by Proposition 1. For the other direction, suppose $K$ is not $\mathcal{L}'$-expressive. We show that for any $\mathcal{L}'$-KB $K'$, $\Delta^D(K') \not\equiv K$ with $\mu = \text{Cl}_{\mathcal{L}'}(K)$. By Proposition 2 the result then follows. Now suppose there exists an $\mathcal{L}'$-KB $K'$ such that $\Delta^D(K') \equiv K$. First observe that since $K$ is not $\mathcal{L}'$-expressible, $\text{Mod}(\mu) \supset \text{Mod}(K)$. Since we are working with drastic distance, in order to promote models of $K$, we also need them in $K'$, hence $\text{Mod}(K') \supset \text{Mod}(K)$ and since $K'$ is from $\mathcal{L}'$ we have $\text{Mod}(K') \supset \text{Cl}_{\mathcal{L}'}(K) = \text{Mod}(\mu)$. Thus there exists $\omega \in \text{Cl}_{\mathcal{L}'}(\text{Mod}(K)) \setminus \text{Mod}(K)$ having distance 0 to $K'$, and thus $\omega \in \Delta^D(K')$. Since $\omega \not\in \text{Mod}(K)$, this yields a contradiction to $\Delta^D(K') \equiv K$. \qed

Hamming Distance and Specific Fragments

We first consider the simplest fragment under consideration, namely conjunction of literals. As it turns out, (non-trivial) distributability for this fragment w.r.t. $\Delta^H$ is not achievable. We then see that more general fragments allow for non-trivial distributions. In particular, we show that every KB is distributable (and even simplifiable) in the 2CNF case, and we finally give a few observations for $\mathcal{L}_{\text{Horn}}$.

The 1CNF Fragment

The following technical result is important to prove the main result in this section.

Lemma 6. For any $\mathcal{L}_{1CNF}$-profile $E = (K_1, \ldots, K_n)$ and interpretations $\omega_1, \omega_2$, it holds that:

$$H(\omega_1, E) + H(\omega_2, E) = H(\omega_1 \cap \omega_2, E) + H(\omega_1 \cup \omega_2, E).$$

Proof. It suffices to show that for each $K_i$ in profile $E$, $H(\omega_1, K_i) + H(\omega_2, K_i) = H(\omega_1 \cap \omega_2, K_i) + H(\omega_1 \cup \omega_2, K_i)$. Indeed, summing up these equalities over all $K_i \in E$, we get

$$\Sigma K_{i \in E} H(\omega_1, K_i) + \Sigma K_{i \in E} H(\omega_2, K_i) = \Sigma K_{i \in E} H(\omega_1 \cap \omega_2, K_i) + \Sigma K_{i \in E} H(\omega_1 \cup \omega_2, K_i).$$

Since $H(\omega, E) = \Sigma K_{i \in E} H(\omega, K_i)$, for any interpretation $\omega$, our conclusion then follows immediately.

Thus, take $\omega_1', \omega_2'$ to be two interpretations that are closest to $\omega_1$ and $\omega_2$, respectively, among the models of $\text{Mod}(K_i)$. In other words, $H(\omega_1, \omega_1') = \min_{\omega \in \text{Mod}(K_i)} H(\omega_1, \omega)$ and $H(\omega_2, \omega_2') = \min_{\omega \in \text{Mod}(K_i)} H(\omega_2, \omega)$.

By induction on the number of propositional atoms in $\mathcal{L}$, we can show that $\omega_1 \cap \omega_2$ and $\omega_1 \cup \omega_2$ are closest in $\text{Mod}(K_i)$ to $\omega_1$ and $\omega_2$, respectively. Thus, we have that $H(\omega_1, K_i) = H(\omega_1, \omega_1') = H(\omega_2, \omega_2') = H(\omega_1 \cap \omega_2, \omega_1') = H(\omega_1 \cup \omega_2, \omega_2')$, and our problem reduces to showing that $H(\omega_1, \omega_1') + H(\omega_2, \omega_2') = H(\omega_1 \cap \omega_2, \omega_1') + H(\omega_1 \cup \omega_2, \omega_2')$. By using induction on the number of propositional atoms in $\mathcal{L}$ again, we can show that this equality holds. The argument runs as follows: in the base case, when the alphabet consists of just one propositional atom, the equality is shown to be true by checking all the cases. For the inductive step we assume the claim holds for an alphabet of size $n$ and show that it also holds for an alphabet of size $n+1$. More concretely, we analyze the way in which the Hamming distances between interpretations change when we add a propositional atom to the alphabet. An analysis of all the possible cases shows that the equality holds. \qed

Next we observe certain patterns of interpretations that indicate whether a KB is $\mathcal{L}_{1CNF}$-expressive or not.

Definition 9. If $K$ is a knowledge base, then a pair of interpretations $\omega_1$ and $\omega_2$ are called critical with respect to $K$ if $\omega_1 \not\subseteq \omega_2$ and $\omega_2 \not\subseteq \omega_1$, and one of the following cases holds:

1. $\omega_1, \omega_2 \in \text{Mod}(K)$ and $\omega_1 \cap \omega_2, \omega_1 \cup \omega_2 \not\in \text{Mod}(K)$,
2. $\omega_1 \in \text{Mod}(K)$ and $\omega_1 \cap \omega_2 \not\in \text{Mod}(K)$,
3. $\omega_2 \in \text{Mod}(K)$ and $\omega_1 \cap \omega_2 \not\in \text{Mod}(K)$,
4. $\omega_1 \cap \omega_2 \in \text{Mod}(K)$ and $\omega_1, \omega_2 \not\in \text{Mod}(K)$, or $\omega_1, \omega_2 \not\in \text{Mod}(K)$
5. $\omega_1 \cap \omega_2 \not\in \text{Mod}(K)$ and $\omega_1, \omega_2 \not\in \text{Mod}(K)$.

Lemma 7. If a KB $K$ is not $\mathcal{L}_{1CNF}$-expressible, then there exist $\omega_1, \omega_2 \in \text{Cl}_{\mathcal{L}_{1CNF}}(K)$ being critical with respect to $K$.

Proof. The fact that $K$ is not $\mathcal{L}_{1CNF}$-expressible implies that either: (i) $K$ is not closed under intersection or union, or (ii) there are $w_1, w_2, w_3 \in \text{Cl}_{\mathcal{L}_{1CNF}}(K)$ such that $w_1 \subseteq w_3 \subseteq w_2$ and $w_1, w_2 \in \text{Mod}(K)$, $w_3 \not\in \text{Mod}(K)$. Case (i) implies that there exist $w_1, w_2 \in \text{Mod}(K)$ such that one of Cases 1-3 from Definition 9 holds. If we are in Case (ii), then consider the interpretation $w_4 = (w_2 \cup w_3) \cup w_1$. Clearly, $w_1 \subseteq w_4 \subseteq w_2$, hence $w_4 \in \text{Cl}_{\mathcal{L}_{1CNF}}(K)$. Also,
$w_3 \cap w_4 = w_1$ and $w_3 \cup w_4 = w_2$. There are two sub-cases to consider here. If $w_4 \notin \text{Mod}(K)$, then we are in Case 4 of Definition 9. If $w_4 \in \text{Mod}(K)$, then we are in Case 5 of Definition 9.

**Example 2.** Let us consider the KB $K$ such that $\text{Mod}(K) = \emptyset, \{a\}, \{b\}, \{c\}, \{a,b,c\}$. $K$ is not 1CNF-expressible; indeed, $\text{Cl}_{1\text{CNF}}(\text{Mod}(K)) = \text{Mod}(K) \cup \{\{a,b\}\}$.

Here, we identify several sets of critical interpretations w.r.t. $K$. First, $S_1 = \{\{a,c\}, \{a,b\}, \{a\}, \{a,b,c\}\}$ corresponds to the situation described in Case 5 of Definition 9, with $\omega_1 = \{a,c\}$ and $\omega_2 = \{a,b\}$.

The set $S_2 = \{\{b,c\}, \{a,b\}, \{a,b,c\}\}$ also corresponds to Case 5, with $\omega_1 = \{b,c\}$ and $\omega_2 = \{a,b\}$.

We can also consider the set of interpretations $S_3 = \emptyset, \{\{a\}\}$, which corresponds to Case 2 of Definition 9, with $\omega_1 = \emptyset$ and $\omega_2 = \{a\}$. The models of $K$ and the sets of critical interpretations are represented in Figure 1.

![Figure 1: Models of $K$ are in the shaded areas; critical interpretations are in the dashed areas.](image)

We can now state the central result of this section.

**Theorem 8.** A KB $K$ is $L_{1\text{CNF}}$-distributable with respect to $\Delta^H$, if and only if $K$ is $L_{1\text{CNF}}$-expressible.

**Proof.** If part. By Proposition 1.

Only if part. Let $K$ be a KB that is not $L_{1\text{CNF}}$-expressible. We will show that it is not $L_{1\text{CNF}}$-distributable w.r.t. $\Delta^H$. Suppose, on the contrary, that $K$ is $L_{1\text{CNF}}$-distributable. Then there exists an $L_{1\text{CNF}}$ profile $E = (K_1, \ldots, K_n)$ such that $\Delta^H(E) \equiv K$, where $\text{Mod}(\mu) = \text{Cl}_{L_{1\text{CNF}}}(\text{Mod}(K))$ (cf. Proposition 2).

By Lemma 7, there exist interpretations $\omega_1, \omega_2 \in \text{Mod}(\mu)$ that are critical with respect to $K$. By Lemma 6, we have

$$H(\omega_1, E) + H(\omega_2, E) = H(\omega_1 \cap \omega_2, E) + H(\omega_1 \cup \omega_2, E).$$

Let us now do a case analysis depending on the type of critical pair we are dealing with. If we are in Case 1 of Definition 9, then it needs to be the case that $H(\omega_1, E) = H(\omega_2, E) = m$, $H(\omega_1 \cap \omega_2, E) = m + k_1$, and $H(\omega_1 \cup \omega_2, E) = m + k_2$, for some integers $m \geq 0$ and $k_1, k_2 > 0$. Plugging these numbers into Equality (1), we get that $2m = 2m + k_1 + k_2$ and $k_1 + k_2 = 0$. Since $k_1, k_2 > 0$, we have arrived at a contradiction. If we are in Case 2, then it needs to be the case that $H(\omega_1 \cap \omega_2, E) = H(\omega_1 \cup \omega_2, E) = m$.

In other words, for any $L_{1\text{CNF}}$-profile and $\mu \in 1\text{CNF}$, $\Delta^H,\mu$ is guaranteed to be $L_{1\text{CNF}}$-expressible as well. We have already shown in Theorem 3, this is not necessarily the case if we replace $\Sigma$ by $G\text{Min}$. The following example shows how to obtain a similar behavior for $G\text{Max}$; we then generalize this idea below.

**Example 3.** Let $U = \{a, b\}$ and $K = \{a \lor b, \neg a \land \neg b\}$. We have $\text{Mod}(K) = \{\{a\}, \{b\}\}$. $K$ is not $L_{1\text{CNF}}$-expressible, since $\text{Cl}_{L_{1\text{CNF}}}(\text{Mod}(K)) = 2^U$. Let $K_S$ be the $L_{1\text{CNF}}$-KB with a single model $S$ for any $S \subseteq U$ and let us have a look at the following distance matrix for $\mu = \text{Mod}(\mu) = \text{Cl}_{L_{1\text{CNF}}}(\text{Mod}(K))$, $E = (K(a), K(b))$, and $E' = (K_S, K(a,b))$:

<table>
<thead>
<tr>
<th>$K_S$</th>
<th>$K(a)$</th>
<th>$K(b)$</th>
<th>$K(a,b)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\emptyset$</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>${a}$</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>${b}$</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>${a, b}$</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Recall that the lexicographic order of the involved vectors is $(0,2) < (1,1) < (2,0)$. We thus get that $\Delta^H,\mu,\text{GMax}(E) \equiv K$ (see also Theorem 3), and on the other hand, $\Delta^H,\mu,\text{GMax}(E') \equiv K$.

**Theorem 9.** Any KB $K$ such that $\text{Mod}(K) = \{\omega, \omega'\}$ is $L_{1\text{CNF}}$-distributable with respect to $\Delta^H,\mu,\text{GMax}$.

**Proof.** If $K$ is $L_{1\text{CNF}}$-expressible, then the conclusion follows from Proposition 1. If $K$ is not $L_{1\text{CNF}}$-expressible, then consider the set $\text{Cl}_{L_{1\text{CNF}}}(\text{Mod}(K)) \backslash \text{Mod}(K) = \{\omega_1, \ldots, \omega_n\}$. We define the profile $E = (K_1, \ldots, K_n)$, where $\text{Mod}(K_i) = \{\mu_i(\omega)\}$, for $i \in \{1, \ldots, n\}$. We show that $\Delta^H,\mu,\text{GMax}(E) \equiv K$, where $\text{Mod}(\mu) = \text{Cl}_{L_{1\text{CNF}}}(\text{Mod}(K))$.

First, we have that $H(\omega_i, U | \omega_i) = |U|$, which implies that $H_{\text{GMax}}(\omega_i, E) = \text{GMax}(U, \ldots, U)$, for any $i \in \{1, \ldots, n\}$. Furthermore, since $H(\omega, U | \omega_i) < |U|$ and $H(\omega', U | \omega_i) < |U|$, for any $i \in \{1, \ldots, n\}$, it follows that $\omega < H_{\text{GMax}}^\omega$ and $\omega' < H_{\text{GMax}}^{\omega'}$. Next, we show that $H_{\text{GMax}}(\omega, E) = H_{\text{GMax}}(\omega', E)$.

Consider the vectors $V = (H(\omega_1, \omega), \ldots, H(\omega_n, \omega))$ and $V' = (H(\omega'_1, \omega'), \ldots, H(\omega'_n, \omega'))$. Our claim is that $\text{GMax}(V) = \text{GMax}(V')$. To see why, notice that the elements in $\text{Cl}_{L_{1\text{CNF}}}(\text{Mod}(K))$ form a complete subset lattice with $\omega$ and $\omega'$ as the top and bottom elements, respectively. Let us write $H(\omega, \omega') = m$. This lattice has $2^m$ elements, and the maximum distance of two elements in it is $m$. Thus, the vector $V$ is the vector of distances between $\omega$ and every other element in this lattice, except itself and $\omega'$. A
similar consideration holds for $V'$. Hence $V$ and $V'$ are vectors of length $2^{m-2}$ whose elements are $m-1$, $m-2$, ..., 1. We can actually count how many times each number appears in $V$ and $V'$. The number of interpretations in the lattice that are at distance of 1 from $\omega$ (and $\omega'$) is $\binom{m}{n}$: thus, $m-1$ appears $\binom{m}{n}$ times in $V$ (and $V'$). The number of interpretations that are at distance 2 from $\omega$ (and $\omega'$) is $\binom{m}{n}$, thus $m-2$ appears $\binom{m}{n}$ times in $V$ and $V'$. We iterate this argument for every distance, up to 1. It is then easy to see that, based on these considerations, $V$ and $V'$ are equal when sorted in descending order. Our conclusion follows from this.

The 2CNF Fragment

We show that every knowledge base $K$ can be distributed in the fragment $L_{2CNF}$. Even a single $L_{2CNF}$ knowledge base is enough to represent $K$. Before giving the general result, we sketch the idea via an example.

Example 4. Let $K$ be a KB with $Mod(K) = \{\{a, b\}, \{b, c, e\}, \{a, c, d\}\}$. We observe that $K$ is not $L_{2CNF}$-expressible since $CL_{2CNF}(Mod(K)) = Mod(K) \cup \{a, b, c\}$. However, we can give an $L_{2CNF}$-KB $K'$ using three new atoms $x, y, z$ to penalize the undesired interpretation $\{a, b, c\}$ such that $\Delta^H(K') \equiv K$, with $\mu \in L_{2CNF}$ of the form $Mod(\mu) = CL_{2CNF}(Mod(K))$. To this end, assume $K'$ with $Mod(K') = \{\omega_1, \omega_2, \omega_3, \omega_4\}$ of the form

$$
\omega_1 = \{a, b, x, y\}, \\
\omega_2 = \{b, c, e, x, z\}, \\
\omega_3 = \{a, c, d, y, z\}, \\
\omega_4 = \{a, b, c, x, y, z\}.
$$

One can verify that $CL_{2CNF}(Mod(K')) = Mod(K')$. Thus, $K'$ can be picked from $L_{2CNF}$. We use $\mu$ such that $Mod(\mu) = CL_{2CNF}(Mod(K))$ and get distances

$$
\begin{align*}
\omega_1 &\quad \omega_2 &\quad \omega_3 &\quad \omega_4 \\
\{a, b\} &\quad 2 &\quad 5 &\quad 4 &\quad 2 \\
\{b, c, e\} &\quad 4 &\quad 2 &\quad 6 &\quad 4 &\quad 2 \\
\{a, c, d\} &\quad 4 &\quad 6 &\quad 2 &\quad 4 &\quad 2 \\
\{a, b, c\} &\quad 3 &\quad 4 &\quad 4 &\quad 3 &\quad 3
\end{align*}
$$

Here, each line gives the distance between a model of $\mu$ and a model of $K'$ (\$\omega_i\$ columns), or between a model of $\mu$ and $K'$ (\$\omega_i\$ min column). The key observation is that pairs from $x, y, z$ as used in $\omega_1, \omega_2, \omega_3$ give minimal distances 2 while the remaining interpretation $\omega_4$, which corresponds to the closure of $K$, contains all three new atoms (since $maj_{\{\{x, y\}, \{x, z\}, \{y, z\}\}} = \{x, y, z\}$).

Theorem 10. Any KB $K$ is $L_{2CNF}$-simplifiable w.r.t. $\Delta^H$.

Proof. We have to show that for any KB $K$, there exists an $L_{2CNF}$-KB $K'$ and a formula $\mu \in L_{2CNF}$ such that $\Delta^H(K') \equiv K$. If $K$ is $L_{2CNF}$-expressible, the result is due to Proposition 1. So suppose that $K$ is not $L_{2CNF}$-expressible and let $Mod(K) = \{\omega_1, \ldots, \omega_n\}$. Consider a set of new atoms $A = \{a_1, \ldots, a_n\}$, and for each $\omega_i \in Mod(K)$, let $\omega' = \omega_i \cup A \setminus \{a_i\}$. We define the $L_{2CNF}$-KB $K'$ and $\mu \in L_{2CNF}$ such that

$$
\begin{align*}
Mod(K') &= CL_{2CNF}(\{\omega' \mid \omega_i \in Mod(K)\}) \\
Mod(\mu) &= CL_{2CNF}(Mod(K)).
\end{align*}
$$

Let $\Omega' = \{\omega'_i \mid \omega_i \in Mod(K)\}$. We first show that for each $\omega_i \in Mod(K') \setminus \Omega'$, $A \subseteq \omega_i$. Indeed, for any triple $\omega_i, \omega_j, \omega_k \in Mod(K)$, such that $\omega_{ijkl} = \max_{\{\omega_i, \omega_j, \omega_k\}} \not\in Mod(K)$, we observe that $maj_{\{\omega_i, \omega_j, \omega_k\}}(\omega_{ijkl}) = \omega_{ijkl} \cup \max_{\{\omega_i, \omega_j, \omega_k\}}(A \setminus \{a_i\}) \cup A \setminus \{a_k\}$. Thus, for each $\omega_i \in CL_{2CNF}(\Omega') \setminus \Omega'$, $A \subseteq \omega_i$. Recall that $Mod(K') = CL_{2CNF}(\Omega')$. It follows quite easily that each further interpretation $\omega \in CL_{2CNF}(\Omega') \setminus CL_{2CNF}(\Omega') \cup \Omega'$, also satisfies $A \subseteq \omega$.

This shows that each model of $K'$ contains at least $n-1$ atoms from $A$. Thus, for every model $\omega_i \in K$, $H(\omega, \omega') = H(\omega, \omega_i) = n-1$. It remains to show that for each $\omega \in Mod(K') \setminus Mod(K)$, $H(\omega, K') \geq n$. First, let $\omega \in \Omega'$. Since $\omega \not\in Mod(K)$, $\omega'$ and $\omega$ are incomparable and $\omega'$ contains $n-1$ elements from $A$, we have $H(\omega, \omega') \geq n$. As shown above all other interpretations $\omega'' \in Mod(K') \setminus \Omega'$ contain all $n$ atoms from $A$, thus $H(\omega, \omega'') \geq n$, too.

As an immediate consequence, we obtain that any KB $K$ is $L_{2CNF}$-distributively $\Delta^H$-strong for any aggregation function $\psi$. Note that this result is in strong contrast to the $L_{1CNF}$ fragment, where only $L_{1CNF}$-expressible KBs are $L_{1CNF}$-distributively $\Delta^H$-strong.

The Horn-Fragment

We now turn our attention to the $L_{Horn}$ fragment. Recall Example 1 where we have shown how to distribute some non $L_{Horn}$-expressible KB using a profile over two $L_{Horn}$-KBs. Our first result shows that in this example case we cannot reduce to profiles of a single KB, i.e. that there are KBs which are $L_{Horn}$-distributively but not $L_{Horn}$-simplifiable.

Proposition 11. Given a KB $K$ with $Mod(K) = \{\omega_1, \omega_2\}$, where $\omega_1 = \omega_1 \cup \omega_2$, $H(\omega_1, \omega_2) = 2$ and $\omega_1, \omega_2$ are incomparable. Then $K$ is not $L_{Horn}$-simplifiable w.r.t. $\Delta^H$.

Proof. The situation described in the Proposition corresponds to $K = \{\omega \cup \{a\}, \omega \cup \{b\}, \omega \cup \{a, b\}\}$ with $\omega$ some interpretation which does not contain $a$ or $b$. We need $Mod(\mu) = \{\omega, \omega \cup \{a\}, \omega \cup \{b\}, \omega \cup \{a, b\}\}$, as required by Proposition 2. We want to identify a $L_{Horn}$-KB $K'$ such that $\Delta^H(K') \equiv K$. This means that $\omega$ is the single model of $\mu$ which is not minimal w.r.t. the Hamming distance. Let $\omega'$ be the model in $K'$ closest to $\omega_1 = \omega \cup \{a\}$ and $\omega_2$ the one closest to $\omega_2 = \omega \cup \{b\}$. We need $a \in \omega_1 \setminus b \in \omega_2$, otherwise $H(\omega_1, \omega') < H(\omega_1, \omega_2)$ or $H(\omega_2, \omega_2') < H(\omega_2, \omega_2)$; further we need $b \notin \omega_1'$ and $a \notin \omega_2'$; otherwise $H(\omega_2, \omega_2') < H(\omega_2, \omega_2') = H(\omega_1', \omega_2')$. Hence $\omega_1'$ and $\omega_2'$ are incomparable also $\omega_1 \cap \omega_2 \subseteq Mod(K')$, since $K'$ is a Horn KB. But then $H(\omega_1', \omega_2') \leq H(\omega_1', \omega_1')$.

Our next result shows that $\Delta^H$ nonetheless increases the range of $L_{Horn}$-simplifiable KBs compared to $\Delta^D$ (recall Theorem 5).

Proposition 12. Any knowledge base $K$ with $Mod(K) = \{\omega_1, \omega_2\}$ is $L_{Horn}$-simplifiable w.r.t. $\Delta^H$.

Proof. If $\omega_1, \omega_2$ are comparable, we can apply Proposition 1. Thus, assume $\omega_1, \omega_2$ are incomparable and let $d_1 = \ldots$
|ω1 \ ω2| and d2 = |ω2 \ ω1|. W.l.o.g. assume d1 ≤ d2. Also note that d1 > 0. We use K' with Mod(K') = \{ω' \ ω1 ∪ ω2\} where ω' adds d1 elements from ω2 \ ω1 to ω1. Thus, ω' ⊆ ω1 ∪ ω2 and we can choose K' from LHorn. Moreover, let µ ∈ LHorn such that Mod(µ) = \{ω1, ω2, ω1 ∩ ω2\}. We have the following distances (note that d(ω2, ω1' = d1 + (d2 - d1)).

Hence, ∆_H(K') ≡ K as desired.

Our final result concerns distributability in the Horn fragment. We show that some KBs with three models can be distributed.

**Proposition 13.** Let K be a KB such that Mod(K) = \{ω1, ω2, ω3\}. If ω1, ω2 and ω3 are not all pairwise incomparable, then K is Horn-distributable w.r.t. ∆_H with ⊕ ∈ \{Σ, GMax, GMin\}.

**Proof.** If K is Horn-expressible, then the result follows from Proposition 1. If K is not Horn-expressible, then we do a case analysis on the number of pairwise incomparable models of K.

**Case 1.** If exactly one pair of models of K are incomparable, then we can assume without loss of generality that it is ω1 and ω2. It follows then that ω3 ∉ ω1 ∩ ω2. Also, there must be distinct atoms a and b such that a ∈ ω1, a /∈ ω2 and b ∈ ω2, b /∈ ω1. We consider a constraint µ ∈ LHorn such that Mod(µ) = \{ω1, ω2, ω1 ∩ ω2\}.

**Case 1.1.** If ω1 ⊆ ω3 and ω2 ⊆ ω1, then we take a globally new atom c and KBs K1 and K2 such that:

- Mod(K1) = \{ω1 ∪ \{c\}, ω2 ∪ \{a\}, ω3, (ω1 ∩ ω2) ∪ \{a, b\}\}
- Mod(K2) = \{ω1, ω2, ω3 ∪ \{c\}, ω1 ∩ ω2\}

It is easy to see that K1 and K2 are Horn-expressible. Considering, now, the profile E = (K1, K2), we obtain the following distances:

<table>
<thead>
<tr>
<th>K1</th>
<th>K2</th>
<th>Σ</th>
<th>GMax</th>
<th>GMin</th>
</tr>
</thead>
<tbody>
<tr>
<td>ω1</td>
<td>1 0 1</td>
<td>(1, 0)</td>
<td>(0, 1)</td>
<td></td>
</tr>
<tr>
<td>ω2</td>
<td>1 0 1</td>
<td>(1, 0)</td>
<td>(0, 1)</td>
<td></td>
</tr>
<tr>
<td>ω3</td>
<td>0 1 1</td>
<td>(1, 0)</td>
<td>(0, 1)</td>
<td></td>
</tr>
<tr>
<td>ω1 ∩ ω2</td>
<td>0 2 0</td>
<td>(2, 0)</td>
<td>(0, 2)</td>
<td></td>
</tr>
</tbody>
</table>

So for each ⊕ ∈ \{Σ, GMax, GMin\} we obtain that \(\Delta_\oplus(E) \equiv K\).

**Case 1.2.** If ω3 ⊆ ω1 and ω2 ⊆ ω1, then ω3 ⊆ ω1 ∩ ω2. Moreover, since ω1 /∈ ω1 ∩ ω2, it actually holds that ω3 ⊆ ω1 ∩ ω2. Thus there exists an atom c such that c ∈ (ω1 ∩ ω2) and c /∈ ω1. We now take KBs K1, K2, K3 and K4 such that:

- Mod(K1) = \{ω1 ∪ \{b\}, ω2 ∪ \{a\}, ω3, (ω1 ∩ ω2) \{c\}, (ω1 ∩ ω2) \{a, b\}\}
- Mod(K2) = \{ω1, ω2 ∪ \{a\}, ω3, (ω1 ∩ ω2) \{a\}\}
- Mod(K3) = \{ω1 ∪ \{b\}, ω2, ω3, (ω1 ∩ ω2) \{b\}\}
- Mod(K4) = \{ω1, ω2, ω3 ∪ \{c\}, ω1 ∩ ω2\}

It is easy to see that K1, K2, K3, and K4 are Horn-expressible. Considering, now, the profile E = (K1, K2, K3, K4), we obtain the following distances:

<table>
<thead>
<tr>
<th>K1</th>
<th>K2</th>
<th>K3</th>
<th>K4</th>
<th>Σ</th>
<th>GMax</th>
<th>GMin</th>
</tr>
</thead>
<tbody>
<tr>
<td>ω1</td>
<td>1 0 1</td>
<td>(1, 0)</td>
<td>(0, 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ω2</td>
<td>0 1 1</td>
<td>(1, 0)</td>
<td>(0, 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ω3</td>
<td>1 0 1</td>
<td>(1, 0)</td>
<td>(0, 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ω1 ∩ ω2</td>
<td>1 1 2</td>
<td>(1, 1)</td>
<td>(1, 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ω3</td>
<td>1 0 1</td>
<td>(1, 0)</td>
<td>(0, 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ω1 ∩ ω2</td>
<td>1 1 2</td>
<td>(1, 1)</td>
<td>(1, 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

So for each ⊕ ∈ \{Σ, GMax, GMin\} we obtain that \(\Delta_\oplus(E) \equiv K\).

**Case 2.** If exactly two pairs of models of K are incomparable, then we can assume without loss of generality that it is ω1, ω2 and ω3. We consider a constraint µ ∈ LHorn such that Mod(µ) = \{ω1, ω2, ω3, ω1 ∩ ω2, ω2 ∩ ω3\}. Then there must be distinct atoms a and b such that a ∈ ω1, a /∈ ω2 and b ∈ ω2, b /∈ ω1. Further, there must be distinct atoms c and d such that c ∈ ω2, c /∈ ω3 and d ∈ ω3, d /∈ ω2.

**Case 2.1.** If ω1 ⊆ ω3, then we get that c /∈ ω1 and a ∈ ω3. We take KBs K1 and K2 such that:

- Mod(K1) = \{ω1 ∪ \{c\}, ω2 ∪ \{a\}, ω3, (ω1 ∩ ω2) ∪ \{a\}, (ω2 ∩ ω3) ∪ \{c\}\}
- Mod(K2) = \{ω1, ω2 ∪ \{a\}, (ω1, ω1 ∩ ω2) ∪ \{a\}, (ω2 ∩ ω3) ∪ \{a\}\}

It is easy to see that K1 and K2 are Horn-expressible. Considering, now, the profile E = (K1, K2, K3, K4, K5) and keeping in mind that c /∈ ω1 and a ∈ ω3, we obtain the following distances:

<table>
<thead>
<tr>
<th>K1</th>
<th>K2</th>
<th>K3</th>
<th>K4</th>
<th>Σ</th>
<th>GMax</th>
<th>GMin</th>
</tr>
</thead>
<tbody>
<tr>
<td>ω1</td>
<td>1 0 1</td>
<td>(1, 0)</td>
<td>(0, 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ω2</td>
<td>0 1 1</td>
<td>(1, 0)</td>
<td>(0, 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ω3</td>
<td>1 0 1</td>
<td>(1, 0)</td>
<td>(0, 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ω1 ∩ ω2</td>
<td>1 1 2</td>
<td>(1, 1)</td>
<td>(1, 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ω2 ∩ ω3</td>
<td>1 1 2</td>
<td>(1, 1)</td>
<td>(1, 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

So for each ⊕ ∈ \{Σ, GMax, GMin\} we obtain that \(\Delta_\oplus(E) \equiv K\).

**Case 2.2.** If ω3 ⊆ ω1, then we get that b /∈ ω3 and d ∈ ω1. We take KBs K1, K2 and such that:

- Mod(K1) = \{ω1 ∪ \{b\}, ω2 ∪ \{b\}, (ω1 ∩ ω2) ∪ \{b\}, (ω2 ∩ ω3) ∪ \{b\}\}
- Mod(K2) = \{ω1, ω2 ∪ \{d\}, ω3, (ω1 ∩ ω2) ∪ \{d\}, (ω2 ∩ ω3) ∪ \{d\}\}

It is easy to see that K1 and K2 are Horn-expressible. Considering, now, the profile E = (K1, K2, K3, K4) and keeping in mind that b /∈ ω3 and d ∈ ω1, we obtain the following distances:

<table>
<thead>
<tr>
<th>K1</th>
<th>K2</th>
<th>K3</th>
<th>K4</th>
<th>Σ</th>
<th>GMax</th>
<th>GMin</th>
</tr>
</thead>
<tbody>
<tr>
<td>ω1</td>
<td>1 0 1</td>
<td>(1, 0)</td>
<td>(0, 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ω2</td>
<td>0 1 1</td>
<td>(1, 0)</td>
<td>(0, 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ω3</td>
<td>1 0 1</td>
<td>(1, 0)</td>
<td>(0, 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ω1 ∩ ω2</td>
<td>1 1 2</td>
<td>(1, 1)</td>
<td>(1, 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ω2 ∩ ω3</td>
<td>1 1 2</td>
<td>(1, 1)</td>
<td>(1, 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
So for each $\oplus \in \{\Sigma, G\text{Max}, G\text{Min}\}$ we obtain that $\Delta_{\mu_\oplus}(E) \equiv K$. The cases when $\omega_2 \subseteq \omega_3$ or $\omega_3 \subseteq \omega_2$ are symmetric. This concludes our case analysis, as any other remaining case results in either all of the interpretations $\omega_1$, $\omega_2$ and $\omega_3$ being pairwise incomparable, or in $K$ being Horn-expressible.

The remaining case (i.e., $\text{Mod}(K) = \{\omega_1, \omega_2, \omega_3\}$ with $\omega_1$, $\omega_2$, $\omega_3$ pairwise incompatbile), as well as the more general case when $K$ has an arbitrary number of models is subject to ongoing work.

### Conclusion

In this paper we have proposed the notion of distributability and we have studied the properties of several merging operators with respect to different fragments of propositional logic. Our results are summarized in Table 1. The symbol $\times$ means that only “trivial” knowledge bases (belonging to the considered fragment) can be distributed with the corresponding operator. Alternately, $\checkmark$ means that any knowledge base can be distributed. Symbol $-$ means we know that some non-trivial knowledge bases can be distributed, and finally $\circ$ means that some, but not all, non-trivial bases can be simplified. Interestingly, the picture emerging from Table 1 is that merging operators behave quite differently depending on the distance and aggregation function employed, in a way that does not lend itself to simple categorization. For instance, our results on simplifiability imply that using Dalal revision to $\mathcal{L}_{1\text{CNF}}$ KBs never takes us outside the $1\text{CNF}$ fragment; applying the same revision operator to $\mathcal{L}_{2\text{CNF}}$ KBs can produce any KB in $\mathcal{L}$; and applying it to $\mathcal{L}_{\text{Horn}}$ KBs can produce some, though not all possible KBs.

Several questions are still open for future work. We plan to study the exact characterization of what can (and cannot) be distributed, in order to replace the symbols $-$ and $\circ$ in the previous table. Other merging operators can also be incorporated to our study. Some of our results on distributability require the addition of new atoms to the interpretations. We want to determine whether similar results can be obtained without modifying the set of propositional variables, in particular for the $2\text{CNF}$ fragment. We are also interested in the number of knowledge bases needed to distribute knowledge: given an integer $n$, a knowledge base $K$ and a merging operator $\Delta$, is it possible to distribute $K$ w.r.t. $\Delta$ such that the resulting profile contains at most $n$ knowledge bases? This paper was a first step to understand the limits of distributability; the actual construction of the profile and complexity of this process are important questions that will be tackled in future research. Finally, we also consider applying the concept of distributability to non-classical formalisms, in particular in connection with merging operators proposed for logic programs (Delgrande et al. 2013).

### Acknowledgments

This work was supported by the Austrian Science Fund (FWF) under grant P25521.

### References


Relations between assumption-based approaches in nonmonotonic logic and formal argumentation*

Jesse Heyninck and Christian Straßer
Institute of Philosophy II, Ruhr Universität Bochum
Universitätstraße 150
44 800 Bochum, Germany

Abstract

In this paper we make a contribution to the unification of formal models of defeasible reasoning. We present several translations between formal argumentation frameworks and nonmonotonic logics for reasoning with plausible assumptions. More specifically, we translate adaptive logics into assumption-based argumentation and ASPIC+ into assumption-based argumentation and a fragment of assumption-based argumentation into adaptive logics. Adaptive logics are closely related to Makinson’s default assumptions and to a significant class of systems within the tradition of preferential semantics in the vein of KLM and Shoham. Thus, our results also provide close links between formal argumentation and the latter approaches.

1 Introduction

There is a plethora of logical approaches to the modelling of defeasible reasoning known as nonmonotonic logics (in short, NMLs). These approaches often use different methods, representational formats or key ideas, making it sometimes difficult to compare them, e.g. with respect to the consequence relations they give rise to. Such comparisons are important to systematise the field of NMLs and to gain insights into which forms of defeasible reasoning are expressible in which formal frameworks. An important tool for such comparisons are translations between systems of NML. If one system (or a fragment thereof) is translatable into another system we immediately know that the latter system is at least as expressive as the former. Moreover, this may lead to forms of cross-fertilisation, since meta-theoretic properties become transferable between the translated systems.

In this contribution we will investigate several such translations. Given the richness of the domain of NMLs, we approach the topic from a specific angle. Our focus will be on structured argumentation, on the one hand, and NMLs that model defeasible inferences in terms of strict inference rules and defeasible assumptions, on the other hand. As a side product, the translation will also cover a significant subclass of NMLs in the KLM paradigm based on preferential semantics [18, 12].

At least since Dung introduced abstract argumentation [8], formal argumentation has been an important sub-domain of NML. While in abstract argumentation arguments are not phrased in a formal language and the underlying inferences are not explicated, several systems of structured or instantiated formal argumentation have been developed which overcome this limitation (cf. [5] for a partial overview). In this paper we will focus on two of the most prominent accounts: assumption-based argumentation (in short, ABA) [7, 10, 20] and ASPIC+ [16, 14].

One of the key differences between several formal approaches to defeasible reasoning concerns the question of how to model defeasible inferences. Let $A_1, \ldots, A_n \leadsto B$ denote the defeasible inference from $A_1, \ldots, A_n$ to $B$. The question is whether such an inference should be phrased in terms of a strict inference rule or a defeasible one. A strict inference rule allows for no exceptions: if its premises $A_1, \ldots, A_n$ are true, the consequent $B$ is true as well. In contrast, defeasible rules allow for exceptions, that is, under specific circumstances it may hold that all premises $A_1, \ldots, A_n$ of the rule hold while the consequent $B$ doesn’t. Clearly, in the approach with strict rules defeasibility has to enter in a different way. One way is by means of explicitly stated defeasible assumptions $A_S_1, \ldots, A_S_m$, i.e., specific premises which are assumed to hold by default and which can serve as antecedents of strict rules. An inference is retracted in case there is a demonstration that one of the defeasible assumptions $A_S_1, \ldots, A_S_m$ doesn’t hold.

ABA follows the approach based on strict rules and defeasible assumptions. In ASPIC+ both approaches can be represented. Not surprisingly, ABA has been shown to be translatable to ASPIC+ [16]. In this paper we will show the other (perhaps more surprising) direction, namely that ASPIC+ (without priorities) can be translated into ABA and thus that both frameworks are equi-expressive.

There are several nonmonotonic systems that model defeasible inference by means of strict rules. Among them are adaptive logics (in short, ALs) [4], Makinson’s default assumptions and forms of circumscription. Makinson’s default assumptions –and in view of the translation in [22] also ALs– are generalisations of approaches based on maximal...
consistent subsets [17]. In view of [1] we know that there are close connections between approaches based on maximal consistent subsets and structured argumentation. In this paper the ties will be strengthened. We show that ALs can be translated into ABA and ASPIC+ and present a translation in the opposite direction for a subclass of ABA and ASPIC+.

We will proceed as follows: in Sections 2–5 we introduce the basic systems (ALs, preferential semantics, default assumptions, ABA, and ASPIC+). In Sections 6–8 we provide translations as indicated in Figure 1.

2 Adaptive Logics

ALs are a general framework for the formal explication of defeasible reasoning. It has been applied to a multitude of defeasible reasoning forms (mainly related to questions from philosophical logic), such as nonmonotonic forms of reasoning with inconsistent information, causal discovery, inductive generalisations, abductive hypothesis generation, normative reasoning, etc. (see [19, p.86] for an overview).

The driving idea behind ALs is to apply defeasible inference rules under explicit abnormality assumptions. More specifically, given a compact Tarksi logic L (the core or lower limit logic) in a formal language and with the derivability relation ⊢L, a set of abnormalities Ω ⊆ L is fixed. Now, whenever the core logic gives rise to ⊢L A ∨ ab where ab ∈ Ω, A can be derived in the adaptive logic (based on L and Ω) on the (defeasible) assumption that ab is false.1

In ALs this basic idea of modeling defeasible inferences is implemented in Hilbert-style proofs. We will first explain the proof theory of ALs.2 Then we give alternative characterizations of the adaptive consequence relations that are central to prove the adequacy of our translations in subsequent sections.

In ALs, usual Hilbert-style proofs are adjusted in two major ways. First, to keep track of normality assumptions, proof lines in adaptive proofs are equipped with an additional column in which the abnormalities are listed that are assumed to be false. Second, different retraction mechanisms for lines with abnormality assumptions that turn out mistaken are implemented in terms of so-called adaptive strategies. We will give some examples below.

To further explain how adaptive proofs work, it is useful to turn to a concrete example. As an illustration, we take a look at inconsistency-ALs. These are based on paraconsistent core logics such as LP or CLuN(s).3 These core logics typically do not validate disjunctive syllogism A, ¬A ∨ B ⊢ B since in case A is involved in a contradiction, B would not follow (then ¬A would suffice for the disjunction ¬A ∨ B to be true). Nevertheless, inconsistency-ALs allow for the defeasible application of disjunctive syllogism under the normality assumption that there is no contradiction in A. Hence, in inconsistency ALs the abnormalities in Ω typically have the form of a contradiction A ∧ ¬A. E.g., in paraconsistent core logics it usually holds that A, ¬A ∨ B ⊢ B ∨ (A ∧ ¬A) and thus one can defeasibly derive B under the assumption that there is no contradiction in A. Clearly, sometimes such assumptions turn out to be mistaken in view of the given premises. Obviously, this is the case if A ∧ ¬A is derivable from the given premises. A more interesting case is given, if A ∧ ¬A is not directly derivable but it is derivable as a member of a minimal disjunction of abnormalities. We illustrate this in the following example.

Example 1. Suppose our core logic is a standard paraconsistent logic such as LP or CLuN(s). Let Γ = {¬p, ¬q, p ∨ q, p ∨ r, q ∨ s}.

1
2
3

Each proof line has 4 elements: a line number, a formula, a justification and a set of abnormalities (which are assumed to be false). All inferences of the core logic L can be applied (indicated by L-Inf in lines 6, 10, 11 and 12). In lines 8 and 9 defeasible inferences are made as explained above. E.g., since at line 6 r ∨ (p ∧ ¬p) is derived, at line 8 the abnormality p ∧ ¬p is considered false and thus put in the abnormality column. The rule employed for this is called RC (rule conditional): from (Γ; A ∧ ab; Δ) derive (Γ'; A; l; RC; Δ ∪ {ab}). When further inferences are made calling upon lines with non-empty sets of abnormalities, these abnormalities are carried over (see lines 10 and 11 where the abnormalities of lines 8 and 9 are carried over).

The retraction of lines in adaptive proofs is always determined in view of the minimal disjunctions of abnormalities derived at a given stage of a proof (on the empty set of abnormalities). At line 12 such a minimal disjunction of ab-

---

1. The disjunction ∨ is supposed to be classical. In fact, in the standard format of ALs which we consider here, the core logic is supposed to be supraclassical. Whenever non-classical logics are used as core logics, classical negation ¬ and classical disjunction ∨ are superimposed.

2. Due to spatial restrictions we will focus on the main ideas but explain some aspects of the proof theory (such as adaptive strategies) merely in a semi-formal way. For a more thorough explanation the interested reader is referred to [4, 19].

3. CLuN(s) is positive classical logic enriched by the law of the excluded middle. For an axiomatization and a semantics see [3].
normalities is derived. Clearly, the abnormalities assumed to be false at lines 8–11 are involved in the given disjunction. There are different retraction mechanisms for ALs: so-called adaptive strategies. According to the reliability strategy, any line with an abnormality in the assumption that is part of a minimal disjunction of abnormalities is to be retracted. Retraction is implemented by marking lines that are retracted. In this case:

1. \( r \in \{ p \land \lnot p \} \)
2. \( s \in \{ q \land \lnot q \} \)
3. \( r \lor s \in \{ p \land \lnot p \} \)
4. \( r \lor s \in \{ q \land \lnot q \} \)

There are other, less cautious, strategies. For instance, according to the minimal abnormality strategy, \( r \lor s \) will not be retracted. The reason is as follows. If we interpret our premises strictly as normal as possible, then in view of line 12 it will be the case that either \( p \land \lnot p \) holds (and \( q \land \lnot q \) doesn’t), or \( q \land \lnot q \) holds (and \( p \land \lnot p \) doesn’t). In each case, one of the assumptions of line 10 or 11 is warranted. Due to space limitations, we omit the technical details. Yet another one of the assumptions of line 10 or 11 is warranted. Due to

These retraction mechanisms provided by adaptive strategies make AL proofs dynamic; sometimes a line may get marked, later unmarked, and yet later marked again. In order to define a consequence relation we need a stable notion of derivability. It works as follows: a formula at a line \( l \) of a proof is finally derived at a stage of the proof if \( l \) is not marked and every extension of the proof in which it gets marked can be further extended such that it is unmarked again. The consequence relation of ALs is the defined as follows:

\[ (\Delta, A) \text{ is derivable} \iff \Delta, A \vdash_L \text{ is provable in AL} \]

For our translations alternative characterisations of the consequence relations defined in terms of final derivability in Definition 1 will be very useful. These characterisations are essentially informed by the set of minimal disjunctions of abnormalities derivable from a given premise set by the core logic L.

### Definition 2
Where \( \Gamma \subseteq L \): \( \Sigma_L(\Gamma) \) is the set of all non-empty \( \Delta \subseteq \Omega \) such that \( \Gamma \vdash_L \Delta \lor \Delta' \) and for all non-empty \( \Delta' \subseteq \Delta, \quad \Gamma \vdash_L \Delta' \lor \Delta' \).

A choice set over \( \Sigma_L(\Gamma) \) is a set \( \Theta \) for which \( \Delta \cap \Theta \neq \emptyset \) for all \( \Delta \in \Sigma_L(\Gamma) \).

### Definition 3
Where \( \Gamma \subseteq L \): \( \Phi_L(\Gamma) \) is the set of \( \Theta \) minimal choice sets over \( \Sigma_L(\Gamma) \).

The following facts will be useful in what follows:

**Fact 1** ([19]). 1. For all choice sets \( \Theta \) over \( \Sigma_L(\Gamma) \) there is a \( \Theta' \in \Phi_L(\Gamma) \) such that \( \Theta' \subseteq \Theta \).

2. \( \phi \in \Phi_L(\Gamma) \iff \phi \) is a choice set of \( \Sigma_L(\Gamma) \) and for all \( A \in \phi \) there is a \( \Delta_A \in \Sigma_L(\Gamma) \) for which \( \{ A \} = \Delta_A \cap \phi \).

We now give representation theorems for all three adaptive strategies, a given core logic L and a given set of abnormalities \( \Omega \).

**Theorem 1** ([4]). \( \Gamma \vdash^{L\text{ma}}_\Theta A \) iff for all \( \Theta \in \Phi_L(\Gamma) \) there is a \( \Delta \subseteq \Omega \setminus \Theta \) such that \( \Gamma \vdash_L \Delta \lor \Delta' \).

**Theorem 2** ([4]). \( \Gamma \vdash^{L\text{ns}}_\Theta A \) iff there is a \( \Delta \subseteq \Omega \setminus \Theta \) such that \( \Gamma \vdash_L \Delta \lor \Delta' \).

**Theorem 3** ([4]). \( \Gamma \vdash^{L\text{str}}_\Theta A \) iff there is a \( \Theta \in \Phi_L(\Gamma) \) and a \( \Delta \subseteq \Omega \setminus \Theta \) such that \( \Gamma \vdash_L \Delta \lor \Delta' \).

### 3 Preferential Semantics and Default Assumptions

The semantics for ALs are a special but rich subclass of the well known preferential semantics as defined in [12] and [18]. As in the previous section we assume a core logic L in a formal language L and a set of abnormalities \( \Omega \subseteq L \).

We also assume that the core logic L comes with an adequate model-theoretic semantics and an associated semantic consequence relation \( \models_L \).

A model \( M \subseteq L \Gamma(\Omega) \) is minimally abnormal iff there is no \( M' \in M \Gamma(\Omega) \) for which \( M' \subseteq M \).

**Definition 4.** \( \Gamma \models^{L\text{ma}}_\Theta A \) iff \( A \models \) for every minimally abnormal model of \( \Gamma \).

**Theorem 4.** Where \( \Gamma \cup \{ A \} \subseteq L \) and \( str \in \{ \text{ma, r, ns} \} \), \( \Gamma \models^{L\text{str}}_\Theta A \) iff \( \Gamma \models^{L\text{str}}_\Theta A \).

In [22], the connection between ALs and Makinson’s Default Assumption Consequence Relations (in short, DACRs) [13, chapter 2] was established. In [13, chapter 2], it is also shown that many other non-monotonic consequence relations, such as Reiter’s Closed World Assumption, Poole’s Background Constraints, etc. can be expressed as DACRs. DACRs give formal substance to the idea that, in many situations, non-monotonic reasoning makes use of a set \( \Delta \) of defeasible background assumptions in combination with the strict and explicit premises in \( \Gamma \). These background assumptions are used to the extent that they are consistent with \( \Gamma \). Accordingly, DACRs make use of the notion of maximal consistent subset:

**Definition 5.** Where \( \Gamma \cup \Delta \subseteq L \) and \( \Theta \subseteq \Delta \) is a maximal \( \Gamma \)-consistent subset of \( \Delta \) iff:

- \( \Gamma \cup \Theta \models A \) for some \( A \in L \) and
- \( \Gamma \cup \Theta' \models A \) for all \( A \in L \) and for every \( \Theta \subseteq \Theta' \subseteq \Delta \).
MCS(Γ, ∆) is the set of all maximal Γ-consistent subsets of ∆.

**Definition 6.** Where Γ ∪ ∆ ∪ {A} ⊆ L, Γ ⊢^{A}_{L} ∆ A iff for every ∆′ ∈ MCS(Γ, ∆), Γ ∪ ∆′ ⊢^{L}_{A} A.

The connection between adaptive logic and DACR’s is the following:

**Theorem 5.** [22, p.10] Where Γ ∪ ∆ ∪ {A} ⊆ L, Γ ⊢^{A}_{L} ∆ A iff for every ∆′ ∈ MCS(Γ, ∆), Γ ∪ ∆′ ⊢^{L}_{A} A.

### 4 Assumption-Based Argumentation

ABA, thoroughly described in [7], is a formal model that allows one to use a set of plausible assumptions “to extend a given theory” [7, p.70] unless and until there are good arguments for not using such an assumption.

Inferences are implemented in ABA by means of a deductive system consisting of a language and rules formulated over this language:

**Definition 7 (Deductive System).** A deductive system is a pair (Ł, R) such that

- Ł is a formal language (consisting of countably many sentences).
- R is a set of inference rules of the form A_{1},...,A_{n} → A and A → A, where A, A_{1},...,A_{n} ∈ Ł.

**Definition 8.** An R-deduction from a theory Γ is a sequence B_{1},...,B_{m}, where m > 0 such that for all i = 1,...,m: B_{i} ∈ Γ or there exists a A_{i},...,A_{n} → B_{i} ∈ R such that A_{1},...,A_{n} ∈ Ł.

**Definition 9.** Where Γ ∪ {A} ⊆ Ł, Γ ⊢^{A}_{Ł} A holds if there is an R-deduction from Γ whose last element is A.

We now introduce defeasible assumptions and a contrariness operator to express argumentative attacks. Given a rule system, an assumption-based framework is defined as follows:

**Definition 10 (Assumption-based framework).** An assumption-based framework is a tuple \( \text{ABF} = ( Ł, R, \Gamma, \text{Ab}) \) where:

- (Ł, R) is a deductive system
- Γ ⊆ Ł
- ∅ ≠ Ab ⊆ Ł is the set of candidate assumptions.
- \( \rightarrow: Ab \rightarrow Ł \) is a contrariness operator.

In most structured accounts of argumentation attacks are defined between arguments which are deductions in a given deductive or defeasible system (e.g., in ASPIC+). In ASPIC+, defeasible Logic Programming [11] or sequents Γ ⊢^{L}_{Ł} A where Ł is an underlying core logic ([2, 6]). In contrast, ABA operates at a higher level of abstraction, since attacks are defined directly on the level of sets of assumptions instead of on the level of R-deductions. ABA can thus be viewed as operating on the level of equivalence classes consisting of arguments generated using the same assumptions.

**Definition 11 (Attacks).** Given an assumption-based framework \( \text{ABF} = ( Ł, R, \Gamma, \text{Ab}) \):

- a set of assumptions ∆ ⊆ Ab attacks an assumption A ∈ Ab iff Γ ∪ ∆ \not{\vdash}^{R}_{Ł} A.
- a set of assumptions ∆ ⊆ Ab attacks a set of assumptions ∆′ ⊆ Ab iff Γ ∪ ∆ \not{\vdash}^{R}_{Ł} A′ for some A ∈ ∆′.

Consequences of a given assumption-based framework are determined with the use of argumentation semantics. On the basis of argumentative attacks, semantics determine sets of assumptions that are acceptable given different criteria of acceptability, such as the requirement that a given set of assumption should not attack itself, or it should be able to defend itself against attacks by other sets of assumptions. Argumentation semantics have been phrased for abstract frameworks in [8] and have been generalised to the level of ABA in e.g. [7].

**Definition 12 (Argumentation semantics).** Where ∆ ⊆ Ab:

- ∆ is closed iff \( \Delta = \{ A \in Ab \mid \Gamma \cup \Delta \nvdash^{R}_{Ł} A \} \).
- ∆ is conflict-free iff for every A ∈ Ab, ∆ ∪ \( \nvdash^{R}_{Ł} A \) or ∆ ∪ \( \nvdash^{R}_{Ł} \nvdash^{Ł}_{Ł} A \).
- A closed set ∆ is naive iff it is maximally (w.r.t. set inclusion) conflict-free.
- A closed set of assumptions ∆ ⊆ Ab is admissible iff it is conflict-free and for each closed set of assumptions ∆′ ⊆ Ab, if ∆′ attacks ∆, then ∆ attacks ∆′.
- A set ∆ is preferred iff it is maximally (w.r.t. set inclusion) admissible.
- A is stable iff it is closed, conflict-free and attacks every a ∈ Ab \Δ.

We write niv(ABF), prf(ABF) resp. stb(ABF) for the set of naive, preferred resp. stable sets of assumptions in ABF.

**Example 2.** Let Ab = \{ q, \neg p \lor \neg q \}, Γ = \{ p \}, let the rule system R characterize classical logic and \( \overline{\text{A}} = \neg A \) (where \( \neg \) is classical negation). Then there are two preferred sets: \{ \neg p \lor \neg q \}, \{ q \}. To see this note that e.g. Γ ∪ \{ \neg p \lor \neg q \} \nvdash^{R}_{Ł} \neg q and Γ ∪ \{ q \} \nvdash^{R}_{Ł} \neg (\neg p \lor \neg q).

We are now in a position to define various consequence relations for ABA:

**Definition 13.** Given an assumption-based framework \( \text{ABF} = ( Ł, R, \Gamma, \text{Ab}) \) and \( \text{sem} \in \{ \text{niv}, \text{prf}, \text{stb} \} \):

- \( \text{ABF} \vdash^{\text{sem}}_{\text{Ł}} A \) if Γ ∪ ∆ \nvdash^{R}_{Ł} A for some ∆ ∈ \text{sem}(ABF).
- \( \text{ABF} \vdash^{\text{sem}}_{\text{Ł}} A \) if Γ ∪ ∆ \nvdash^{R}_{Ł} A for every ∆ ∈ \text{sem}(ABF).
- \( \text{ABF} \vdash^{\text{sem}}_{\text{Ł}} A \) if Γ ∪ ∆ ∩ Δ ∈ \text{sem} \nvdash^{R}_{Ł} A.

### 5 ASPIC+

In ASPIC+, as in ABA, inferences made on the basis of a strict knowledge base can be extended with additional inferences based on plausible assumptions. However, whereas in ABA attacks and extensions where defined directly on the level of these assumptions, in ASPIC+, arguments are specific deductions. More precisely, arguments are constructed
from a knowledge base using an argumentation system. An argumentation system is a generalisation of a deductive system (Def. 7) that allows for a distinction between strict (i.e. deductive or safe) and defeasible rules.\footnote{In the ASPIC$^+$ framework of [16], there is also the possibility to add a preference ordering over the premises and/or defeasible rules. Similar generalisations exist for AIs and approaches based on maximal consistent subsets and their generalisations such as Makinson’s default assumptions. We will present investigations into translations for systems with priorities at a future occasion. In our presentation, we also disregard a special type of premise called “issue” in the context of ASPIC$^+$. Issues are premises that are never acceptable in the sense that they always require further backup by additional arguments.}

**Definition 14** (Defeasible Theory). Given a formal language $\mathcal{L}$, a defeasible theory $\mathcal{R} = (\mathcal{L}, \mathcal{S}, \mathcal{D})$ consists of:
- a set of strict rules $S$ of the form $A_1, \ldots, A_n \rightarrow B$
- a set of defeasible rules $D$ of the form $A_1, \ldots, A_n \rightarrow B$.

We also assume there is a naming function $N : \mathcal{S} \cup \mathcal{D} \rightarrow \mathcal{L}$ s.t. every rule $r \in \mathcal{S} \cup \mathcal{D}$ gets assigned a unique name. $A_1, \ldots, A_n$ are called the antecedents and $B$ is called the consequent of $A_1, \ldots, A_n \rightarrow B$ resp. $A_1, \ldots, A_n \Rightarrow B$.

**Definition 15** (Argumentation System). Given a defeasible theory $\mathcal{R}$, an argumentation system is a tuple $\mathcal{AS} = (\mathcal{R}, \neg \neg)$ where $\neg \neg$ is a contrariness function from $\mathcal{L}$ to $2^\mathcal{L}$.

Arguments are built using defeasible and/or strict rules to derive conclusions from a knowledge base. A knowledge base consists of strict and plausible premises. $\mathcal{K}_n$ is the set of all (necessary) axioms, i.e. premises that are considered to be outside the reach of argumentative attacks. $\mathcal{K}_n$ has an analogous function to the defeasible assumptions in ABA: they are deemed plausible in that they are assumed to be true unless and until a counterargument is encountered.

**Definition 16** (Knowledge Base). A Knowledge Base is a set $\mathcal{K}$, where $\mathcal{K} = \mathcal{K}_n \cup \mathcal{K}_a$ and $\mathcal{K}_n \cap \mathcal{K}_a = \emptyset$.

**Definition 17** (Arguments). Let $\mathcal{L} = (\mathcal{R}, \neg \neg)$ be an argumentation system and $\mathcal{K} = \mathcal{K}_n \cup \mathcal{K}_a$ a knowledge base. An argument $a$ is one of the following:
- a premise argument $A$ if $A \in \mathcal{K}$
- a strict rule-argument $(a_1, \ldots, a_n \rightarrow B)$ if $a_1, \ldots, a_n$ (with $n \geq 0$) are arguments such that there exists a strict rule $\text{conc}(a_1), \ldots, \text{conc}(a_n) \rightarrow B \in \mathcal{S}$
- a defeasible rule-argument $(a_1, \ldots, a_n \Rightarrow B)$ if $a_1, \ldots, a_n$ (with $n \geq 0$) are arguments such that there exists a defeasible rule $\text{conc}(a_1), \ldots, \text{conc}(a_n) \Rightarrow B$

We will use $\text{Arg}(\mathcal{AS}, \mathcal{K})$ to denote the set of all arguments that can be built from a knowledge base $\mathcal{K}$ using an argumentation system $\mathcal{AS}$.

**Example 3**. Let $\mathcal{S} = \{\neg q \rightarrow \neg p\}$, $\mathcal{D} = \{\neg p \Rightarrow s\}$, $\mathcal{K}_n = \{\neg s\}$, and $\mathcal{K}_a = \{\neg q, \neg p, q\}$. We have, e.g., the following arguments:
- $a_1 = (\neg q)$
- $a_2 = (\neg p)$
- $a_3 = (\neg s)$
- $a_4 = (\neg s)$
- $a_5 = (\neg a_3 \Rightarrow s)$
- $a_6 = (\neg q)$
- $a_7 = (\neg \neg s)$

**Definition 18**. Where $a$ is an argument $a = (B)$, $a = \langle a_1, \ldots, a_n \rightarrow B \rangle$ or $a = \langle a_1, \ldots, a_n \Rightarrow B \rangle$, we define:
- $\text{conc}(a) = B$
- $\text{sub}(a) = \text{sub}(a_1) \cup \ldots \cup \text{sub}(a_n) \cup \{a\}$
- where $a$ is a premise argument: $\text{prem}(a) = \{A\}$
- where $a$ is not a premise argument: $\text{prem}(a) = \{\{a\} | a \in \text{sub}(a)\}$

The distinction between strict and defeasible rule-arguments allows us to define a variety of attack forms:

**Definition 19** (Attacks). Where $a, b \in \text{Arg}(\mathcal{AS}, \mathcal{K})$, $a$ attacks $b$ (in signs, $a \Rightarrow b$) iff
- $\text{conc}(a) \in \overline{B}$ for some $B \in \text{prem}(b) \cap \mathcal{K}_a$ (Undermining).
- $\text{conc}(a) \in \overline{B}$ for some $b' \in \text{sub}(b)$ such that $\text{conc}(b') = B$ and $b'$ is of the form $\langle b'_1, \ldots, b'_n \Rightarrow B' \rangle$ (Rebut).
- $\text{conc}(a) = \overline{B}$ for some $b' \in \text{sub}(b)$ such that $b'$ is a defeasible argument (Undercut).

**Example 4** (Ex. 1, contd). Where $\overline{A} = \{B | B \equiv \neg A\}$ for every $A \in \mathcal{L}$, we have: $a_1 \Rightarrow a_0$, $a_0 \Rightarrow a_1$, $a_0 \Rightarrow a_3$, $a_0 \Rightarrow a_2$, $a_2 \Rightarrow a_4$, $a_7 \Rightarrow a_5$.

**Definition 20** (Structured Argumentation Framework). A structured argumentation framework $\mathcal{AT} = (\text{Arg}(\mathcal{AS}, \mathcal{K}), \Rightarrow)$ is a pair where $\text{Arg}(\mathcal{AS}, \mathcal{K})$ is the set of arguments built from $\mathcal{K}$ using the argumentation system $\mathcal{AS}$ and $\Rightarrow$ is an attack relation over $\text{Arg}(\mathcal{AS}, \mathcal{K})$.

Given a structured argumentation framework, we can again make use of Dung’s argumentation semantics to define different notions of acceptable sets of arguments.

**Definition 21** (Argumentation Semantics). Given a structured argumentation framework $\mathcal{AT} = (\text{Arg}(\mathcal{AS}, \mathcal{K}), \Rightarrow)$, where $\mathcal{B} \subseteq \text{Arg}(\mathcal{AS}, \mathcal{K})$,
- $\mathcal{B}$ is conflict-free iff there is no $a, b \in \mathcal{B}$ such that $a \Rightarrow b$
- $\mathcal{B}$ is naive iff it is maximally conflict-free.
- $\mathcal{B}$ defends a $\in \mathcal{A}$ iff for every $c \in \mathcal{A}$ for which $c \Rightarrow a$, there is a $b \in \mathcal{B}$ such that $b \Rightarrow c$.
- $\mathcal{B}$ is admissible iff it is conflict-free and it defends every argument $a \in \mathcal{B}$.
- $\mathcal{B}$ is preferred iff it is maximally (w.r.t. set inclusion) admissible.
- $\mathcal{B}$ is stable iff it is conflict-free and for every $a \in \text{Arg}(\mathcal{AS}, \mathcal{K}) \setminus \mathcal{B}$, $\mathcal{B} \Rightarrow a$.

We write $\text{niv}(\mathcal{AT})$, $\text{prf}(\mathcal{AT})$ resp. $\text{stb}(\mathcal{AT})$ for the set of naive, preferred resp. stable sets of arguments in $\mathcal{AT}$.

**Definition 22**. Where $\mathcal{AT} = (\text{Arg}(\mathcal{AS}, \mathcal{K}), \Rightarrow)$ is a structured argumentation framework and $\text{sem} \in \{\text{niv}, \text{prf}, \text{stb}\}$,
- $\mathcal{AT} \vdash_{\text{sem}} A$ iff there is an $a \in \mathcal{B}$ with $\text{conc}(a) = A$ for some $\mathcal{B} \in \text{sem}(\mathcal{AT})$.
- $\mathcal{AT} \vdash_{\text{sem}} A$ iff for every $B \in \text{sem}(\mathcal{AT})$ there is an $a \in B$ with $\text{conc}(a) = A$.
- $\mathcal{AT} \vdash_{\text{sem}} A$ iff there is an $a \in B$ with $\text{conc}(a) = A$ for every $B \in \text{sem}(\mathcal{AT})$. 
6 Translating Adaptive Logic to Assumption-Based Argumentation

The idea of the translation from ALs to ABA is the following. We translate the lower limit logic L of the given AL into a deductive system, plausible assumptions are negations of abnormalities, and the contrariness operator is classical negation. Recall that the lower limit logic is a superclassical Tarski logic. Hence, there are classical negation ¬ and classical disjunction ∨ in the underlying language of L. In the remainder of this section we will use ¬ and ∨ denoting these classical connectives.

We now go through the technical details of our translation.

Definition 23. Let AL be an AL with the lower limit logic L, in a formal language L and the consequence relation ⊢L, the set of abnormalities Ω ⊆ L and a strategy str (reliability, minimal abnormality, or normal selections). Let L be characterised by the rules in R and the axiom schemes in A. We define the assumption-based framework ABF(L(Γ)) for the premise set Γ ⊆ L as the tuple ABF(L(Γ)) = ((L, R(L)), Ω, A, ⊢L, ¬), where:

• R(L) contains all instances of rules in R and a rule → A for all abnormal A of axiom schemes in A,

• Ω = {¬A | A ∈ Ω}

• ¬ : A → L, where ¬¬A = A

Below we show the following representational theorem:

Theorem 6. Where Γ ∪ {A} ⊆ L and sem ∈ {niv, prf, stb},

1. ABF(L(Γ)) ⊢L sem A iff Γ ⊢L A

2. ABF(L(Γ)) ⊢L ma A iff Γ ⊢L ma A

3. ABF(L(Γ)) ⊢L sem A iff Γ ⊢L ma A.

To avoid clutter we introduce some notational convention:

Notation 1. Where Δ ⊆ Ω, Δ⁻ = {¬A | A ∈ Δ} and Δ⁻ = Δ.

The following fact follows immediately in view of the compactness and the transitivity of L.

Fact 2. Where Γ ∪ {A} ⊆ L, Γ ⊢L(R(L)) A iff Γ ⊢L A.

In view of this fact, we will indiscriminately use ⊢L(R(L)) and ⊢L. Note that in view of the superclassicality of L we have:

Fact 3. Γ ⊢L Δ⁻ | A iff Γ ⊢L ∨ Δ⁻ | A.

We now established that every instantiation of an AL is indeed an assumption-based framework. We prove that the three consequence relations of ALs correspond to intuitive ways of calculating consequences in ABA. The crucial result to prove this is the fact that every preferred extension in some assumption-based framework ABF(L(Γ)) is exactly the set of negations of abnormalities excluding some choice set over the derivable abnormalities. This is shown in the following lemmas.

Lemma 1. Where φ ∈ Φ(L(Γ)), ABF(L(Γ)) \ φ⁻ is stable in ABF(L(Γ)).

Proof. We first show that Δ⁻ = ABF(L(Γ \ φ⁻) is conflict-free. Assume for a contradiction that it is not and hence that there is a B ∈ Ω for which Γ ∪ Δ⁻ | T, ¬B. Hence, by the compactness of L and Fact 3, Γ ⊢L Θ for some finite Θ ⊆ Δ. Let Θ be c-minimal with this property. Hence, Θ ∈ Σ(L(Γ)). However, then φ ⊢L Θ = φ ∧ Θ, a contradiction.

We now show that Δ⁻ is stable. For this, let ¬B ∈ ABF(L(Γ). Hence, B ∈ φ. With Fact 1, there is a Θ ∈ Σ(L(Γ)) such that {B} = φ ⊢L Θ. Since Γ ⊢L Θ, by Fact 3 also Γ ⊢L (Θ⁻ \ {¬B}) | T B. By the monotonicity of L, Γ ⊢L Δ⁻ | T B which means that Δ attacks B.

Since Δ⁻ is conflict-free and attacks every A ∈ ABF(L(Γ \ Δ⁻), it is easy to see that Δ⁻ is closed and stable.

Example 5 (Ex. 1 contd). Take ABF(L(Γ \ Δ⁻) then there is a φ ∈ Φ(L(Γ) for which Δ ⊆ Ω \ φ.

Proof. Suppose Δ ⊆ Ω \ φ for all φ ∈ Φ(L(Γ) and Δ ⊆ Ω. By Fact 1, Ω \ Δ is not a choice set of Σ(L(Γ). Thus, there is a Θ ∈ Σ(L(Γ) for which Θ ⊆ Δ. Since Γ ⊢L Θ, also Γ ∪ (Θ \ {A}) ⊢L A for any A ∈ Θ. Thus, Γ ⊢L Δ is not L-consistent since Γ ∪ Δ⁻ | T, ¬A by monotonicity. By Fact 2, Γ ⊢L Δ⁻ | T, ¬A and thus, Δ is not conflict-free in ABF(L(Γ).

With Lemmas 1 and 2 we immediately get:

Lemma 3. Where Γ ⊆ L, ABF(L(Γ \ Δ⁻) | φ ∈ Φ(L(Γ)) = stb(ABF(L(Γ))) = prf(ABF(L(Γ))) = niv(ABF(L(Γ)).

We are now in a position to prove our main result in this section:

Proof of Theorem 6. In view of Lemma 3 it is enough to show the theorem for sem = stb.

Ad 3. ABF(L(Γ)) ⊢L stb A iff Γ ∪ ∣ Δ⁻ | Δ ∈ stb(ABF(L(Γ))) A. By Lemma 1, this is the case iff Γ ∪ ∣ Ω \ φ⁻ | φ ∈ Φ(L(Γ)) A. Since Φ(L(Γ)) = Σ(L(Γ)) (which is easy to see and left to the reader), this is equivalent to Γ ∪ (Ω \ Σ(L(Γ))) ⊢L A. By compactness, monotonicity and Fact 3, this is equivalent to Γ ⊢L A ∨ Δ⁻ for some finite Δ ⊆ Ω \ Σ(L(Γ)). By Theorem 2 this is equivalent to Γ ⊢L A.

Ad 1. and 2. Analogous.

Translating Adaptive Logic to ASPIC+

In [16] we have a translation from ABA to ASPIC+. Although this translation requires several assumptions that ABF(L(Γ)) does not satisfy, it turns out that it is easy to prove that ABF(L(Γ)) can easily be translated to an assumption-based framework that does satisfy these assumptions.

The underlying idea is basically the same as that for translating AL into ABA: the plausible knowledge base consists of the negated abnormalities, the strict premises of the ASPIC+ framework are the premise set Γ and the strict rules
of the ASPIC+ framework are the inference rules of the monotonic core logic. Due to spatial restrictions, we are not able to present the full technical details of this translation and the adequacy results here.

7 Translating ASPIC+ to Assumption-Based Argumentation

In this section we translate ASPIC+ to ABA. Since in ABA we have no defeasible rules and less attack types than in ASPIC+ the possibility of this translation is less expected than the translation in the other direction (as provided in [16]). In this section we thus offer an answer to the open question stated in [14] whether such a translation can be given. Our translation works as follows:

**Definition 24.** Where \( AS = (R, −) \) is an argumentation system in the formal language \( \mathcal{L} \) with a naming function \( N \) for the rules in \( R \) and \( K = K_n \cup K_a \) is a knowledge base, we translate \( AS \) into an assumption-based framework \( ABF(AS) = (\mathcal{L}', R', K_n, AB, \neg) \) as follows\(^9\):

- \( \mathcal{L}' \subseteq \mathcal{L} \) is such that \( \mathcal{L}' \setminus \mathcal{L} \) contains for each \( r \) in \( R \) a unique name \( n(r) \) and its contrary \( \overline{n(r)} \);\(^10\)
- \( R' \) contains each strict rule from \( R \) and for each defeasible rule \( r : A_1, \ldots, A_n \Rightarrow A \) it contains\(^11\):
  - the rule \( n(r), A_1, \ldots, A_n \Rightarrow A \)
  - the rule \( \overline{A} \Rightarrow \overline{n(r)} \)
- \( AB = K_a \cup \{n(r) \mid r \text{ is a defeasible rule in } R\} \)

> Below we will show that the translation is adequate in view of the following corollary:

**Corollary 1.** Where \( AT = (\text{Arg}(AS, K), \Rightarrow) \) is a structured argumentation framework and \( \text{sem} \in \{ \text{stb, prf} \} \),

1. \( ABF(AS) \vdash^\text{un} \text{sem} \Rightarrow \text{sem} \) iff \( AT \vdash^\text{un} \Rightarrow \text{sem} \).
2. \( ABF(AS) \vdash^\text{stb} \Rightarrow \text{sem} \) iff \( AT \vdash^\text{stb} \Rightarrow \text{sem} \).
3. \( ABF(AS) \vdash^\text{prf} \Rightarrow \text{sem} \) iff \( AT \vdash^\text{prf} \Rightarrow \text{sem} \).

In the following we suppose a given argumentation system \( AS \) and its translation \( ABF(AS) \) as in Definition 24.

**Definition 25.** Where \( \Delta \subseteq Ab \), \( \text{Arg}_\Delta \subseteq \text{Arg}(AS, K) \) is the set of all arguments \( a \) that use only defeasible assumptions in \( \Delta \), any strict rules, and only defeasible rules \( r \) for which \( n(r) \in \Delta \).

Where \( A \subseteq \text{Arg}(AS, K) \) is a set of arguments, \( Ab_A \subseteq Ab \) is the set of assumptions consisting of \( \{1\} \) defeasible assumptions \( A \in K_a \) for which \( \text{prem}(a) = A \) and \( \text{conc}(a) = A \) for

some \( a \in A \) and (2) of \( n(r) \) where \( r \) is a defeasible rule used in some argument in \( A \).

Where \( A \) is a set of arguments in \( \text{Arg}(AS, K) \), \( A^* \) denotes the set \( \text{Arg}_{AB_A} \).

We sometimes write \( Ab_a \) instead of \( Ab_{(a)} \).

**Fact 4.** Where \( A \subseteq \text{Arg}(AS, K) \) is a set of assumptions, \( A \subseteq A^* \).

**Lemma 4.** Where \( A \neq n(r) \) for any \( r \) in \( R \) and \( \Delta \subseteq Ab \), if \( K_n \cup \Delta \vdash \overline{A} \) then

1. If \( A \in \mathcal{L} \), there is an \( a \in \text{Arg}_\Delta \) such that \( \text{conc}(a) = A \).
2. else (if \( A = \overline{n(r)} \)), there is an \( a \in \text{Arg}_\Delta \) for which \( \text{conc}(a) = \overline{B} \) where \( B \) is the consequent of \( r \).

**Proof.** This can be shown by an induction on the length of a deduction from \( K_n \cup \Delta \) to \( A \). Base step: this is trivial since \( A \in K \). Inductive step. We have three possibilities:

1. \( A \) is the result of applying a strict rule \( r \) in \( R \) to \( A_1, \ldots, A_n \), or
2. \( A \) is the result of applying the translation of a defeasible rule \( r = A_1, \ldots, A_n \Rightarrow A \in R \) to \( A_1, \ldots, A_n \) and the rule name \( n(r) \), or
3. \( A = \overline{n(r)} \) is the result of applying a rule \( \overline{B} \Rightarrow \overline{n(r)} \) where \( B \) is the consequent of the defeasible rule \( r \) in \( R \).

**Ad 2.** By the induction hypothesis there are arguments \( a_i \) (\( 1 \leq i \leq n \)) s.t. \( a_i \in \text{Arg}_\Delta \) and \( \text{conc}(a_i) = A_i \). (Note here that \( A_i \notin \mathcal{L}' \setminus \mathcal{L} \).) Clearly, \( a = \langle a_1, \ldots, a_n \Rightarrow A \rangle \) \( A \in \text{Arg}_\Delta \) since \( n(r) \in \Delta \). Ad 1. Analogous. Ad 3. By the induction hypothesis and since \( \overline{B} \in \mathcal{L} \), there is an argument \( a \in \text{Arg}_\Delta \) with \( \text{conc}(a) = \overline{B} \).

The other direction of Lemma 4.1 follows immediately in view of Definition 25:

**Fact 5.** Where \( A \subseteq \text{Arg}(AS, K) \), if \( a \in A \) with \( \text{conc}(a) = A \) then \( K_n \cup Ab_A \vdash R_A \).

**Lemma 5.** Where \( A \subseteq \text{Arg}(AS, K) \), if \( A \) is admissible then \( A^* \) is admissible.

**Proof.** Suppose there are \( a \) and \( b \in A^* \) s.t. \( a \) attacks \( b \). For each attack form it is easy to see that then there is a \( b' \in A \) s.t. \( a \) attacks \( b' \). Take, for instance, rebuttal. Then \( \text{conc}(a) = \overline{b'} \) where \( B' = \overline{n(b')} \) for some \( b' \in \text{sub}(b) \). Hence, there is a defeasible rule \( r \) which is applied in \( b' \) to produce \( B' \). By the definition of \( \text{Arg}_{Ab_A} \) there is an argument \( b'' \in A \) s.t. \( r \) is applied to produce \( \text{conc}(b'') = B' \). For the other attack types (undercuts and undermines) this is shown in an analogous way. Now, since \( A \) is admissible, there is a \( c \in A \) s.t. \( c \) attacks \( a \). Since by Fact 4, \( c \in A^* \), also \( A^* \) is defended. To show that \( A^* \) is conflict-free, assume for a contradiction that \( a \in A^* \). Since \( a \) attacks \( b'' \in A \), \( A \) attacks \( a \) (due to the admissibility of \( A \)). However, in view of the fact that \( A \) and \( A^* \) make use of the same defeasible assumptions and defeasible rules and \( A \) attacks \( a \) in one of the two, this leads to a selfattack in some argument \( a' \in A \). E.g., suppose \( A \) undermines \( a \) in some \( B \in \text{prem}(a) \). Then \( B \in Ab_A \). Hence there is an argument \( a' \in A \) with \( B \in \text{prem}(a') \) and \( A \) attacks \( a' \). Since \( A \) is conflict-free, this is a contradiction.

\[^9\]For simplicity, we will assume that the contrariness function of the ASPIC+ framework assigns a unique contrary to every \( A \in \mathcal{L} \). If this assumption is not satisfied, one has to add \( A_1 \Rightarrow \overline{A}, \ldots, A_n \Rightarrow \overline{A} \) for every \( A_i \in A \), where \( \overline{A} \in \mathcal{L}' \setminus \mathcal{L} \) is the contrary of \( A \) in \( ABF \), as suggested by [20, p.109].

\[^{10}\]Formally: \( \mathcal{L}' \subseteq \mathcal{L} = \{n(r) \mid r \in R \} \cup \{\overline{n(r)} \mid r \in R \} \) (where \( \{n(r) \mid r \in R \} \cup \{\overline{n(r)} \mid r \in R \} = \emptyset \)). This warrants that, unlike the names \( N(r) \) used in \( AS \), the new names \( n(r) \) are not antecedents and consequences of rules in \( R \). We use the new names to simulate defeasible rules in \( ABF \).

\[^{11}\]We suppose that the rules in \( R \) are instances as opposed to schemes. The translation can easily be adjusted to schemes.
Lemma 6. Where $A = A^* \subseteq \text{Arg}(\text{AS}, K)$, $Ab_A$ is closed.

Proof. Suppose $A = A^*$ and $\text{K}_n \cup Ab_A \vdash R A$ for some $A \in Ab$. We have two possibilities: (1) $A = n(r)$ for some $r \in R$ or (2) $A \in \text{K}_n$. Ad 1. Since there are no rules with consequent $n(r)$, $n(r) \notin Ab_A$, Ad 2. By Lemma 4, there is an $a \in A^* = A$ with $\text{conc}(a) = A$. Hence, by the definition of $Ab_A$, $A \in Ab_A$. □

Lemma 7. Where $A = A^* \subseteq \text{Arg}(\text{AS}, K)$, if $A$ is admissible then $Ab_A$ is admissible.

Proof. Suppose $A = A^*$. By Lemma 6, $Ab_A$ is closed. Suppose $Ab_A$ is not conflict-free. Hence, $\text{K}_n \cup Ab_A \vdash R A$ for some $A \in Ab_A$. We use Lemma 4 according to which we have two cases. Case 1: there is an $a \in A^*$ s.t. $\text{conc}(a) = A$. Since $A = A^*$, $a \in A$ and $A$ is not conflict-free. Case 2: $A = n(r)$ and there is an $a \in A$ for which $\text{conc}(a) = B$ where $B$ is the consequent of $r$. Since $n(r) \notin Ab_A$, there is an argument $a' \in A$ which uses rule $r$ to produce $\text{conc}(a') = B$ and is thus rebut-attacked by $a$. Again, $A$ is not conflict-free. Thus, we have shown (by contraposition) that if $A$ is conflict-free then $Ab_A$ is conflict-free.

Suppose $A$ is admissible, $\Delta$ is closed and attacks $Ab_A$. Hence, $\text{K}_n \cup \Delta \vdash R A$ for some $A \in Ab_A$. By Lemma 4 we have two cases. Case 1: there is an $a \in \text{Arg}_A$ s.t. $\text{conc}(a) = A$. Hence, $A \notin n(r)$ for any $r \in R$. Clearly, $a$ attacks $A$. Since $A$ is admissible, there is a $b \in A$ s.t. $b$ attacks $a$. Then $Ab_b \subseteq Ab_A$ and $\text{K}_n \cup Ab_b \vdash \text{conc}(b)$. Thus, $Ab_b$ attacks $Ab_a$ and hence $Ab_A$ attacks $\Delta$.

Case 2: $A = n(r)$ and there is an $a \in \text{Arg}_A$ s.t. $\text{conc}(a) = B$ where $B$ is the consequent of $r$. In this case there is an $a' \in A$ which uses rule $r$ and hence $\text{conc}(a') = B$. Since $A$ is admissible, there is a $c \in A$ which attacks $a$. But then $\Delta_c \subseteq Ab_A$ attacks $\Delta_a$ and hence $Ab_A$ attacks $\Delta$. □

Lemma 8. If $\Delta \subseteq Ab$ is admissible, then $\text{Arg}_A$ is admissible.

Proof. Similar to the previous proof. □

Theorem 7. 1. If $\Delta$ is preferred (resp. stable) then $\text{Arg}_A$ is preferred (resp. stable).

2. If $A$ is preferred (resp. stable) then $\Delta$ is preferred (resp. stable) for some $\Delta \supseteq Ab_A$ for which $\text{Arg}_A = A$.

Proof. Ad.1 Suppose $\Delta$ is preferred. Then, by Lemma 8, $\text{Arg}_A$ is admissible. Suppose there is an $A^* \supseteq Ab_A$ that is admissible, then by Lemma 7, also $Ab_A$ is admissible. Since $\Delta \subseteq Ab_A$ this is a contradiction.

Ad.2 Suppose $A$ is preferred. By Lemma 5 and since trivially $A \subseteq A^*$, $A = A^*$. By Lemma 7, $Ab_A$ is admissible. Now suppose that there is a $\Delta \supseteq Ab_A$ that is admissible. Then by Lemma 8, $\text{Arg}_A$ is admissible. Clearly $A \subseteq \text{Arg}_A$. By the maximality of $A$, $A = \text{Arg}_A$.

Due to space limitations we omit the proof for stable extensions. □

Corollary 1 follows directly with Theorem 7, Lemma 4 and Fact 5.

8 Translating Assumption-based Argumentation to Adaptive Logic

In this section we will translate a fragment of assumption-based argumentation to adaptive logic.

In the following we write $\text{ABA}_R^A(\Gamma)$ for the assumption-based framework $(\mathcal{L}, R), \Gamma, A B^{
eg}$.

For our translation we will use some connectives from Kleene’s well-known 3-valued logic $K_3$ (see Table 1) and superimpose them on a logic that is characterised by the rules in $R$. This works as follows.

We define the 3-valued logic $L^3_R$ semantically in the following way: we superimpose on the language $\mathcal{L}$ the operators $\sim$ and $\lor$ (which are supposed to not occur in the alphabet of $\mathcal{L}$) resulting in the set of well-formed formulas $L^3_R$. The operators are characterised by the truth tables in Table 1.12

<table>
<thead>
<tr>
<th>$A$</th>
<th>$\overline{A}$</th>
<th>$A$</th>
<th>$\sim A$</th>
<th>$\lor$</th>
<th>$\sim$</th>
<th>$\lor$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$u$</td>
<td>$u$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1: Truth-tables for $\sim, \sim$, and $\lor$.

Definition 26. $v : \mathcal{L} \rightarrow \{0, 1, u\}$ is a function which respects the truth-table for $\neg$ (i.e., $v(\overline{A}) = 1$ iff $v(A) = 0$, $v(\overline{\overline{A}}) = 0$ iff $v(A) = 1$, and $v(A) = u$ iff $v(A) = u$). The valuation function $v_M : L^3_R \rightarrow \{0, 1, u\}$ is defined inductively as follows:

1. where $A \in \mathcal{L}$, $v_M(A) = v(A)$.

2. $v_M(\sim A) = 0$ iff $v_M(A) = 1$, and $v_M(\sim A) = 1$ else.

3. $v_M(A \lor B) = \max(v_M(A), v_M(B))$ where $0 < u < 1$.

We write $M \models A$ iff $v_M(A) = 1$ (so 1 is the only designated value). We write $M \models L^3_R$ for the resulting consequence relation.

We now use $L^3_R$ as a lower limit logic for an adaptive logic with the set of abnormalities:

Notation 2. $\Omega^A_{Ab} = \{A \mid A \in Ab\}$.

We translate the rules of $R$ as follows: $A_1, \ldots, A_n \rightarrow B$ is translated to $\sim A_1 \lor \ldots \lor \sim A_n \lor B$.

Notation 3. Where $R$ is a set of rules, we write $R^\sim$ for the set of translated rules.

Our two main representational results in this section are (to be proven below):

Theorem 8. Where $\Gamma \cup \{A\} \subseteq \mathcal{L}$, and $\text{sem} = \text{niv}$.

1. $\text{ABA}_{R^\sim}^{\text{Ab}}(\Gamma) \vdash_{\text{sem}} A$ iff $\Gamma \cup R^\sim \models_{\text{rs}} \Omega^A_{\text{Ab}} L^3_R A$.

2. $\text{ABA}_{R^\sim}^{\text{Ab}}(\Gamma) \vdash_{\text{sem}} A$ iff $\Gamma \cup R^\sim \models_{\text{rs}} \Omega^A_{\text{Ab}} L^3_R A$.

3. $\text{ABA}_{R^\sim}^{\text{Ab}}(\Gamma) \vdash_{\text{sem}} A$ iff $\Gamma \cup R^\sim \models_{\text{rs}} \Omega^A_{\text{Ab}} L^3_R A$.

12In the terminology of [21], our negation $\sim$ corresponds to Bochvar’s “external negation” and $\sim$ corresponds to Kleene’s negation in his $K_3$. Our disjunction $\lor$ is Kleene’s strong disjunction. The requirement of supraclassicality for $L^3_R$ to serve as a core logic for an AL is satisfied in view of the $(\lor, \sim, \sim)$-fragment of $L^3_R$. 


We can strengthen our result if we suppose that the rule
system based on $R$ satisfies the following requirement:
where $\Gamma \cup \{A\} \subseteq \mathcal{L}$. 

**EX** Where $\Delta \subseteq Ab$ is naive in $ABA^{Ab}_{R}(\Gamma)$ and $A \in Ab \setminus \Delta$, 
$\Gamma \cup \Delta \vdash R \overline{A}$.

This criterion ensures that every naive set is stable.

**Theorem 9.** Where $\Gamma \cup \{A\} \subseteq \mathcal{L}$; if $ABA^{Ab}_{R}(\Gamma)$ satisfies 
(EX), items 1–3 in Theorem 8 hold for $sem \in \{niv, pfr, stb\}$.

We are now going to prove the two theorems above. The
following notation will be convenient to avoid clutter:

**Notation 4.** $\Delta^* = \{\sim A \mid A \in \Delta\}$.

The following facts will be useful below:

**Fact 6.** Where $\Gamma \cup \Delta \cup \{A\} \subseteq \mathcal{L}^{L_{R}}_{R}$, (i) $\overline{A} \not\vdash_{L_{R}} \sim A$, (ii) 
$\Gamma \not\vdash_{L_{R}} \Delta \lor A$ iff $\Gamma \cup \Delta \not\vdash_{L_{R}} A$.

$L_{R}$ is obviously a compact Tarski logic.

We say that $\Gamma \subseteq \mathcal{L}$ is $R$-consistent iff there is no $A$ such that 
$\Gamma \vdash_{R} A, \overline{A}$.

**Lemma 9.** Where $\Gamma \cup \{A\} \subseteq \mathcal{L}$,

1. $\Gamma \vdash_{R} A$ implies $\Gamma \vdash_{R} \sim_{L_{R}} A$

2. if $\Gamma$ is $R$-consistent, $\Gamma \vdash_{R} \sim_{L_{R}} A$ implies $\Gamma \vdash_{R} A$.

**Proof.** 1. Simple induction on the number of proof steps.

We show the induction step. Let $M$ be a model of $\Gamma \cup \sim_{L_{R}} A$.

Suppose $\Delta$ follows by means of the application of a rule $A_{1}, \ldots, A_{n} \rightarrow B$. By the induction hypothesis, $M \models A_{1}, \ldots, A_{n}$. Also, $M \models \sim A_{1} \lor \cdots \lor \sim A_{n} \lor B$. Hence, with the truth-tables for $\sim$ and $\lor$, $M \models B$.

2. Suppose $\Gamma \not\vdash_{R} A$. We now construct a counter-model $M$ of $\Gamma \cup \sim_{L_{R}} A$ for $A$ as follows. Let

$$
\nu : B \mapsto \begin{cases} 1 & \Gamma \vdash_{R} B \\ 0 & \Gamma \not\vdash_{R} B \\ u & \text{else} \end{cases}
$$

Note that $\nu(A) \in \{0, u\}$ and hence $M \not\models A$. We have to show that $M$ is a model of $\Gamma \cup \sim_{L_{R}} A$. Since $\Gamma$ is $R$-consistent, the definition warrants that the truth-table for $\sim$ is respected by $\nu$. We thus only have to check whether $\overline{M}$ verifies all formulas in $\Gamma \cup \sim_{L_{R}} A$. As for $\nu$ this holds trivially since every $B \in \Gamma$ is such that $\Gamma \vdash_{R} B$ and thus $\nu(B) = 1$. Let now $A_{1}, \ldots, A_{n} \rightarrow B \in \Gamma$. We have to check whether $M \vdash_{L_{R}} \{v^{M}(A_{i}) \sim A_{i} \mid 1 \leq i \leq n\} \lor B$. Assume the opposite. Thus $v^{M}(A_{i}) = 1$ and $v^{M}(B) \in \{0, u\}$. But then $\Gamma \not\vdash_{R} A_{i}$ and $\Gamma \vdash_{R} B$. Hence, $v^{M}(B) = 1$, a contradiction.

We say that $\Gamma$ is $L_{R}$-consistent, if there is a $A \in \mathcal{L}^{L_{R}}_{R}$ for which $\Gamma \not\vdash_{L_{R}} A$.

**Lemma 10.** Where $\Gamma \subseteq \mathcal{L}$, if $\Gamma$ is $R$-consistent then $\Gamma \cup \sim_{L_{R}}$ is $L_{R}$-consistent.

**Proof.** Suppose $\Gamma$ is $R$-consistent. Then $\Gamma \not\vdash_{R} A, \overline{A}$ for any $A \in \mathcal{L}$. By Lemma 9, also $\Gamma \cup \sim_{L_{R}} \not\vdash_{L_{R}} A, \overline{A}$ for any $A \in \mathcal{L}$.

**Lemma 11.** Where $\Gamma \subseteq \mathcal{L}$; $\Delta^* \subseteq \Omega_{Ab} \setminus \phi$ for some $\phi \in \Phi_{Ab}(\Gamma \cup \overline{\Gamma})$, if $\Delta$ is conflict-free in $ABA^{Ab}_{R}(\Gamma)$.

**Proof.** Suppose $\Delta^* \not\subseteq \Omega_{Ab} \setminus \phi$ for all $\phi \in \Phi_{Ab}(\Gamma \cup \overline{\Gamma})$ and $\Delta \subseteq Ab$. By Fact 1, $\Omega_{Ab} \setminus \Delta^*$ is not a choice set of $\Sigma_{L_{R}}(\Gamma \cup \overline{\Gamma})$ which means that there is a $\Theta \in \Sigma_{L_{R}}(\Gamma \cup \overline{\Gamma})$ such that $\Theta \subseteq \Delta$. Since $\Gamma \cup \overline{\Gamma} \not\vdash_{L_{R}} \Theta \lor \overline{\Theta}$, by Fact 6 also $\Gamma \cup \Theta \setminus \{A\} \cup \overline{\Gamma} \not\vdash_{L_{R}} \sim A$ for any $A \in \Theta$. Hence, 
$\Gamma \cup \Delta \cup \overline{\Gamma}$ is not $L_{R}$-consistent since $\Gamma \cup \Delta \cup \overline{\Gamma} \not\vdash_{L_{R}} A$. Thus by Lemma 10, $\Gamma \cup \Delta$ is not $R$-consistent and thus $\Delta$ is not conflict-free.

**Lemma 12.** Where $\Gamma \subseteq \mathcal{L}$, $\Delta^* = \Omega_{Ab} \setminus \phi$ for some $\phi \in \Phi_{Ab}(\Gamma \cup \overline{\Gamma})$, $\Delta$ is naive in $ABA^{Ab}_{R}(\Gamma)$.

**Proof.** Suppose $\Delta^* = \Omega_{Ab} \setminus \phi$ for some $\phi \in \Phi_{Ab}(\Gamma \cup \overline{\Gamma})$.

We first prove that $\Delta$ is conflict-free. Suppose for a contradiction, there is a $B \in Ab$ such that $\Gamma \cup \Delta \vdash_{R} B, \overline{B}$. By Lemma 9, $\Gamma \cup \Delta \cup \overline{\Gamma} \not\vdash_{L_{R}} B, \overline{B}$. Hence, $\Gamma \cup \Delta \cup \overline{\Gamma}$ is $L_{R}$-inconsistent and by Fact 6 and compactness, $\Gamma \cup \overline{\Gamma} \not\vdash_{L_{R}} \Theta$ for some finite $\Theta \subseteq \Delta$. Let $\Theta$ be $\prec$-minimal with this property, so that $\Theta \in \Sigma_{L_{R}}(\Gamma \cup \overline{\Gamma})$. Since $\phi \neq \Theta = 0$ this is a contradiction to the fact that $\phi \in \Phi_{Ab}(\Gamma \cup \overline{\Gamma})$.

We now show that $\Delta$ is closed. Assume for a contradiction that $\Gamma \cup \Delta \vdash_{R} A$ for some $A \in Ab \setminus \Delta$. By Lemma 9, $\Gamma \cup \Delta \cup \overline{\Gamma} \not\vdash_{L_{R}} A$. Note that $\sim A \in \mathcal{L}$. By Fact 1, there is a $\Theta \in \Sigma_{L_{R}}(\Gamma \cup \overline{\Gamma})$ for which $\sim A = \phi \cap \Theta$. Since $\Gamma \cup \overline{\Gamma} \not\vdash_{L_{R}} \Theta$, by Fact 6, $\Gamma \cup \overline{\Gamma} \not\vdash_{L_{R}} \sim A$. By the monotonicity of $L_{R}$, $\Gamma \cup \overline{\Gamma} \not\vdash_{L_{R}} A$. Thus, $\Gamma \cup \Delta \cup \overline{\Gamma}$ is not $L_{R}$-consistent which implies by Lemma 10 that $\Gamma \cup \Delta$ is not $R$-consistent. This contradicts the fact that $\Delta$ is conflict-free.

The following theorem follows immediately in view of Lemma 11 and Lemma 12:

**Theorem 10.** Where $\Gamma \subseteq \mathcal{L}$, $\Delta$ is a naive extension of $ABA^{Ab}_{R}(\Gamma)$ iff $\Delta^* = \Omega_{Ab} \setminus \phi$ for some $\phi \in \Phi_{Ab}(\Gamma \cup \overline{\Gamma})$.

If we suppose requirement (EX), we can also prove Theorem 9.

**Lemma 13.** Where $\Gamma \subseteq \mathcal{L}$, $\Delta$ is naive in $ABA^{Ab}_{R}(\Gamma)$.

**Proof.** Suppose $\Delta^* = \Omega_{Ab} \setminus \phi$ for some $\phi \in \Phi_{Ab}(\Gamma \cup \overline{\Gamma})$.

In view of Lemma 12 we only need to show that $\Delta$ attacks all $B \in Ab \setminus \Delta$. Thus let $B \in Ab \setminus \Delta$. By (EX) $\Gamma \cup \Delta \vdash_{R} \overline{B}$. Thus, $\Delta$ attacks $B$.

The following Corollary follows immediately in view of Theorem 10 and Lemma 13:

**Corollary 2.** Where $\Gamma$ satisfies (EX), each naive set is stable in $ABA^{Ab}_{R}(\Gamma)$.

In [7], the following was defined resp. proven:

**Definition 27.** An assumption-based framework is normal $iff$ every naive set of assumptions is stable.

**Theorem 11.** For any normal assumption-based framework, for any set of assumptions $\Delta \subseteq Ab$, $\Delta$ is naive iff $\Delta$ is stable.

---

13Or equivalently and analogous to the $R$-consistency: if there is no $A \in \mathcal{L}$ s.t. $\Gamma \not\vdash_{L_{R}} A, \sim A$. 

Corollary 3. If an assumption-based framework satisfies (EX), \( \Gamma \subseteq L, \Delta \) is a preferred, stable and naive extension of \( ABA_{\text{Ab}}^3(\Gamma) \) iff \( \Delta^\sim = \Omega_{\text{Ab}}^3 \setminus \phi \) for some \( \phi \in \Phi_{L_{\text{Ab}}}^3 (\Gamma \cup R^\sim) \).

We are now in a position to prove our two main theorems in this section.

Proof of Theorems 8 and 9. [Theorem 9.1, \( \Leftarrow \): Suppose that \( \Gamma \cup R^\sim \models_{L_{\text{Ab}}} L_{\text{Ab}} A \). By Theorem 3, there is a \( \Delta^\sim \subseteq \Omega_{\text{Ab}}^3 \setminus \phi \) for some \( \phi \in \Phi_{L_{\text{Ab}}}^3 (\Gamma \cup R^\sim) \) s.t. \( \Gamma \cup R^\sim \models_{L_{\text{Ab}}} L_{\text{Ab}} A \lor \bigvee \Delta^\sim \). By the monotonicity of \( L_{\text{Ab}} \) and Fact 6, \( \Theta \cup \Gamma \cup R^\sim \models_{L_{\text{Ab}}} L_{\text{Ab}} A \) where \( \Theta^\sim = \Omega_{\text{Ab}}^3 \setminus \phi \). By Lemma 13, \( \Theta \) is stable. Thus, \( \Gamma \cup \Theta \) is \( R \)-consistent. By Lemma 9, \( ABA_{\text{Ab}}^3(\Gamma) \vdash_{\text{sen}} A \).

The other direction and the other cases are shown analogously. \( \square \)

9 Conclusion

In this paper we provided translations between several prominent systems in nonmonotonic logic (see Fig. 1 for an overview). In this conclusion we discuss some benefits.

In view of the translation of ALs into ABA we know that ALs can be understood as forms of formal argumentation. In view of the fact that ALs are equi-expressive with the syntactically characterised preferential semantics in Sec. 3 and Makinson’s default assumptions, the same can be said about the latter two frameworks. Since a broad variety of defeasible reasoning forms in a wide range of application contexts have been explicated within the ALs family (see Sec. 2), all these reasoning forms are now available in the domain of formal argumentation. This may lead to further refinements. For instance, once embedded in ASPIC\(^+\) we gain rich resources to express preferences and priorities.

In view of the other direction, from a subclass of ABA to ALs, we know that this class can be understood in terms of the model-theoretic tools provided by KLM-style preferential semantics or, alternatively, as consistency management in terms of maximal consistent subsets as provided by default assumptions. This also means that meta-theoretic insights from, for instance, ALs become available for this subclass of ABA. For example, the computational complexity of ALs is well-understood [23, 15]. Moreover, properties of the AL consequence relations apply to this class of ABA. For instance, we know that adaptive consequence relations are cumulative (in the notation of Section 2, where AL is an adaptive logic, \( \Gamma, \Delta, \{ A \} \subseteq L, \) and \( \Gamma \vdash_{\text{AL}} B \) for all \( B \in \Delta, \Gamma \vdash_{\text{AL}} A \) iff \( \bigvee \Delta \cup \bigvee \Delta^\sim \)). For a study of meta-theoretic properties of ALs see [4, 19]. Finally, besides the available dialogue-based methods to model argumentative reasoning processes (e.g. [9]), now the dynamic proof theory of adaptive logics can also be used for this purpose.

Finally, we complete the circle between ABA and ASPIC\(^+\) (without priorities/preferences) by providing a translation from the latter to the former, whereas the other direction has been presented in [16]. As a side-product this provides a way to phrase the defeasible rules of ASPIC\(^+\) as strict rules. This shows that the strict fragment of ASPIC\(^+\) (without strict rules and thus without rebuttals and undercutts) is equi-expressive with full ASPIC\(^+\). Such insights are conceptually interesting and may simplify future meta-theoretic investigations into ASPIC\(^+\).

In future work we intend to generalise our investigations to approaches with priorities and preferences as provided in ASPIC\(^+\) and some generalisations of ALs. An interesting question will be, for instance, whether full ASPIC\(^+\) is translatable into lexicographic ALs [19, ch.5] or whether the latter can be translated to ABA or ASPIC\(^+\).

References


Ordinal Conditional Functions for Nearly Counterfactual Revision

Aaron Hunter
BCIT
Burnaby, BC, Canada
aaron_hunter@bcit.ca

Abstract

We are interested in belief revision involving conditional statements where the antecedent is almost certainly false. To represent such problems, we use Ordinal Conditional Functions that may take infinite values. We model belief change in this context through simple arithmetical operations that allow us to capture the intuition that certain antecedents cannot be validated by any number of observations. We frame our approach as a form of finite belief improvement, and we propose a model of conditional belief revision in which only the “right” hypothetical levels of implausibility are revised.

Introduction

The theory of belief change is concerned with the way agents incorporate new information. Typically, the focus is on new information that is given as a propositional formula. In this paper, we are concerned with situations where an agent needs to revise by a conditional where the antecedent is almost certainly false. More precisely, we consider antecedents that will not be believed given any finite amount of “regular” supporting evidence. We represent the degree of belief in such formulas using Ordinal Conditional Functions that may take infinite values, and we provide an approach to conditional revision based on basic ordinal arithmetic.

This paper makes several contributions to existing work on belief change. First, we demonstrate that a simple algebra of belief change in the finite case extends naturally to the infinite case, giving a form of belief improvement. In the process, we demonstrate that there are natural examples in commonsense reasoning where multiple levels of infinite implausibility are actually useful. In particular, we introduce a natural approach to revision by conditional statements with little in the way of new formal machinery.

Motivating Example

Consider the following claims:

1. heavy: Your dog is overweight.
2. fly: Your dog can fly.
3. hollow\|fly: If your dog can fly, then it has hollow bones.

The first two claims are simple declarative statements. But note that there is a clear difference in the amount of evidence needed to convince the agent to believe each claim. For (1), it presumably takes some finite number of reports from a trusted source. For (2), it seems unlikely that any finite number of reports would be convincing. This statement is almost certainly false, though it is possible to imagine a situation that would convince an agent to believe it.

The third statement is a conditional with a highly unlikely antecedent. Nevertheless, the perceived “impossibility” of (2) does not mean that (3) is free of content. Revision by (3) should change an agent’s beliefs in a counterfactual sense; they may need to change their beliefs about hollow bones in some hypothetical scenario. Moreover, if ever the notion of flying dogs becomes believable, then this report will take on significance at the level of factual beliefs. In this paper, we refer to claims such as (3) as nearly counterfactual. We will provide a formal characterization of such claims, as well as a suitable approach to revision.

Preliminaries

Belief Revision

Belief revision is the belief change that occurs when new information is presented to an agent with some prior, possibly contradictory, set of beliefs. We assume an underlying propositional signature \( P \). An interpretation over \( P \) is called a state, while a logically closed set of formulas over \( P \) is called a belief set. A belief revision operator is a function that combines the initial belief set and a formula to produce a new belief set.

Formal approaches to belief revision typically require an agent to have some form of ordering or ranking that gives the relative plausibility of possible states. For example, in the well-known AGM approach, total pre-orders over states are used to represent the perceived likelihood of each state (AGM85; KM92). Unfortunately, this approach does not handle the problem of iterated belief revision. Related work has addressed iterated revision by explicitly specifying how the ordering changes, rather than just the belief set (DP97; BM06; JT07).
Ordinal Conditional Functions

An ordinal conditional function (OCF) is a function that maps each state to an ordinal (Spohn 1988; Williams 1994). In this approach, strength of belief is captured by ordinal precedence. Hence, if \( r \) is an OCF and \( r(s) < r(t) \), then \( s \) is a more plausible state than \( t \). There is an obvious advantage to this approach in that a ranking function is clearly more expressive than a total pre-order.

While the original definition allows the range of an OCF to be the class of all ordinals, in existing work it is common to restrict the range to the natural numbers, possibly with an additional symbol \( \infty \) representing impossibility. In this paper, we will actually use a slightly larger range; so we need to briefly review ordinal arithmetic.\(^2\) For our purposes, it is sufficient to note that ordinals are actually sets defined by an “order type.” The finite ordinals are the natural numbers. The order type of the natural number \( n \) is unique, because it is the only ordinal that has exactly \( n - 1 \) preceding ordinals. The first infinite ordinal is \( \omega \), the set of all natural numbers. Every countably infinite subset of the natural numbers is order-isomorphic to \( \omega \).

It is easy to construct a countably infinite set that is not order isomorphic to \( \omega \); just add another symbol \( \infty \) at the end that is larger than every natural number. The ordinal that defines the order type of this set is written \( \omega + 1 \). Similarly, there exists a distinct ordinal \( \omega + n \) for any natural number \( n \). And if we add a complete copy of the natural numbers, then we have the ordinal \( \omega + \omega \) which is normally written as \( \omega \cdot 2 \). We can proceed in this manner indefinitely to define a countably infinite sequence of ordinals. By taking powers, we can get even more order types; we will not delve further into this topic.

Ordinal addition can be understood in terms of the informal discussion above. Given ordinals \( \alpha \) and \( \beta \), the ordinal \( \alpha + \beta \) has the order type obtained by taking a set with order type \( \alpha \) and then appending a set with order type \( \beta \) where all the elements of \( \beta \) follow the elements of \( \alpha \) in the underlying ordering. For finite ordinals, this coincides with the usual notion of addition. For infinite valued ordinals it does not. Note for example that \( 1 + \omega = \omega \); adding a number that precedes \( 0 \) does not change the order type, because the resulting structure is isomorphic to the natural numbers. On the other hand \( \omega + 1 \neq \omega \). So ordinal addition is not commutative. It is also worth noting that ordinal subtraction is, in general, not well defined. In particular, it is not possible to define subtraction by \( \omega \).

Belief Change as Ordinal Arithmetic

Although our goal is to address revision by conditionals, we first introduce a simple approach to belief change based on the addition of ordinals. This will allow us to precisely define the notion of a nearly counterfactual statement, which is important for the class of conditionals that we wish to consider.

Restricted Domains

The following definition allows us to define conditional functions over any set of ordinals.

**Definition 1** Let \( S \) be a non-empty set of states and let \( \Gamma \) be a collection of ordinals. A \( \Gamma \)-CF (\( \Gamma \) conditional function) over \( S \) is a function \( r : S \to \Gamma \) such that \( r(s) = 0 \) for some state \( s \).

Note that the definition of \( \Gamma \)-CFs does not actually specify that \( \Gamma \) is a set, because we do not wish to specify the underlying set theory in detail.

Several special cases are immediate:

- Spohn’s ordinal conditional functions are \( \Omega \)-CFs, where \( \Omega \) is the collection of all ordinals.
- The class of \( \omega \)-CFs coincides with the finite valued ranking functions common in the literature.
- The class of \( (\omega + 1) \)-CFs is the set of ranking functions that can take finite values, as well as the single “impossible” plausibility value \( \infty \). This is essentially equivalent to the possibilistic logic framework of (DP04), that uses the “necessity measure” of 0.

In this paper, we are primarily interested in the class of \( \omega^2 \)-CFs. Note that \( \omega^2 \) can be specified as follows:

\[
\omega^2 = \bigcup \{ \omega \cdot k + c \mid k, c \in \omega \}.
\]

Hence, every element of \( \omega^2 \) can be written as \( \omega \cdot k + c \) for some \( k \) and \( c \). We think of these conditional functions as having countably many infinite levels of implausibility. A picture of \( \omega^2 \) is shown in Figure 1.

If \( r \) is a \( \Gamma \)-CF, we write

\[
Bel(r) = \{ x \mid r(x) = 0 \}.
\]

The degree of strength of a conditional function \( r \) is the least \( n \) such that \( n = r(v) \) for some \( v \notin Bel(r) \). Hence, the degree of strength is a measure of how difficult it would be for an agent to abandon the currently believed set of states.

Finite Arithmetic on Conditional Functions

In the finite case, belief change can be captured through addition on ranking functions. Some variant of the following definition has appeared previously in published work by several authors; it is restated here and translated to our terminology.

---

\(^2\)It is beyond the scope of this paper to give a complete treatment of infinite ordinals, and ordinal arithmetic. In the discussion here, we skip over fundamental set theory, and the fact that order-types are defined in terms of set-containment. We refer the reader to (Dev93) for an excellent introduction.
Definition 2 Let $r_1$ and $r_2$ be $\omega$-CFs over $S$, and let $m$ be the minimum value of $r_1 + r_2$. Then $r_1 \bar{+} r_2$ is the function on $S$ defined as follows:

$$r_1 \bar{+} r_2(x) = r_1(x) + r_2(x) - m.$$  

It is easy to check that this operation is associative, commutative, and that every element is invertible in the sense that, for each $r$ there is an $r'$ such that $r + r' = 0$. Therefore, in terms of algebra, we say that the class of $\omega$-CFs is an abelian group under $\bar{+}$.

Note that Spohn’s conditionalization can be seen as a special case of this algebra on ranking functions. Let $r_1$ be a finite plausibility function representing the initial beliefs of an agent. Let $\phi$ be a formula, let $d$ be a positive integer, and let $r_2$ be the ranking function defined as follows:

$$r_2(s) = \begin{cases} 0 & \text{if } s \models \phi \\ d & \text{otherwise} \end{cases}$$

Then $r_1 \bar{+} r_2$ is equivalent to Spohn’s conditionalization of $r_1$ by $\phi$ with strength $d$. Similarly, if $r_2$ takes only two values and the degree of strength of $r_2$ is strictly larger than the degree of strength of $r_1$, then $r_1 \bar{+} r_2$ is AGM revision.

This approach does not extend to larger classes of ordinals.

Proposition 1 Let $\beta$ be an ordinal such that $\omega \in \beta$. Then $\bar{+}$ is not well-defined over the class of $\beta$-CFs.

The problem is that subtraction is not defined for all pairs of (infinite) ordinals.

Example Consider the motivating example. We can define the following $(\omega + 1)$-CFs:

$$r_1(s) = \begin{cases} 0 & \text{if } s \models \{\text{fly}\} \\ \omega & \text{otherwise} \end{cases}$$

$$r_2(s) = \begin{cases} \omega & \text{if } s \models \{\text{fly}\} \\ 0 & \text{otherwise} \end{cases}$$

Normalized addition of $r_1$ and $r_2$ requires us to calculate $\omega - \omega$. But this subtraction is not defined, so the calculation can not be completed.

This problem could be avoided by removing the normalization, but the result would no longer be an OCF. If we want to work with ranking functions that are closed under some form of addition, then we must either modify the definition, or we must relax the constraint that the pre-image of 0 is non-empty. We opt for the former.

Finite Zeroing

We define an algebra over $\omega^2$-CFs based on finite zeroing. The following relation will be useful in proving results. In the definition, and in some future results, it is useful to consider functions over ordinals that do not necessarily take the value 0 for any argument. We use the general term $\Gamma$ ranking to refer to an arbitrary function from $S$ to $\Gamma$.³

³Konieczny refers to this kind of OCF as a free OCF (Kon09)

Definition 3 For $\Gamma$ rankings $r_1$ and $r_2$, we write $r_1 \sim r_2$ just in case the following condition holds for every pair of states $s,t$

$$r_1(s) < r_1(t) \iff r_2(s) < r_2(t).$$

Clearly, $\sim$ is an equivalence relation.

The intuition behind finite zeroing is that each conditional function can be categorized by its minimum value, in a manner that is useful for revision. Given any $\omega^2$ ranking $r$, let $\min(r)$ denote the minimum value $r(s)$. Note that a minimum is guaranteed by the fact that the ordinals are well-ordered.

Definition 4 Let $r$ be an $\omega^2$ ranking with $\min(r) = \omega \cdot k + c$. Then $k$ is the degree of strength of $r$, and $c$ is the finite shift, written $\deg(r)$ and $\fin(r)$ respectively.

We can use the degree and the finite shift to define the following operation.

Definition 5 Let $r$ be an $\omega^2$ ranking with $\deg(r) = k$ and $\fin(r) = c$. Define $\bar{\ast}$ as follows. Let $s$ be a state with $r(s) = \omega \cdot m + p$.

1. If $m > k$, then $\bar{\ast}(s) = \omega \cdot (m - k) + c$
2. If $m = k$, then $\bar{\ast}(s) = (p - c)$

We call $\bar{\ast}$ the finite zeroing of $r$. Intuitively, elements at the “lowest level” are normalized to zero and elements at higher levels are shifted down by the degree of $r$. The following result is easy to prove.

Proposition 2 If $r$ is an $\omega^2$ ranking, then $\bar{\ast}$ is a $\omega^2$-CF and $r \sim \bar{\ast} r$.

Hence, the finite zeroing of any ranking is an equivalent $\omega^2$-CF. We can now extend the definition of $\ast$ to $\omega^2$-CFs.

Definition 6 Let $r_1, r_2$ be $\omega^2$-CFs. Then

$$r_1 \ast r_2 = r_1 + r_2$$

Using this definition, $\ast$ is consistent with $\bar{\ast}$ for $\omega$-CFs. Hence, $\ast$ can capture standard belief revision operators (e.g., AGM, DP) by restricting to finite values and setting the degree of strength of each function appropriately. This is the natural extension of revision, therefore, to the case that allows infinite plausibility values.

Example The motivating example over $\{\text{heavy}, \text{fly}\}$ can be captured by the following function:

$$r(s) = \begin{cases} \omega & \text{if } s \models \text{fly} \\ 10 & \text{if } s \models \text{heavy} \land \neg \text{fly} \\ 0 & \text{otherwise} \end{cases}$$

We let $\ast^n$ to denote a finite iteration of the $\ast$ operator. Suppose that, for each $V \in \{\text{heavy}, \text{fly}\}$, $r_V$ is an OCF such that $r_V(s) = 2$ if and only if $s \not\models V$. The following are immediate:

- $r \ast^n r_{\text{heavy}}(s) = 0$ if $n \geq 5$.
- $r \ast^n r_{\text{fly}}(s) \neq 0$ for any $n$. 
Hence, it takes 5 reports to convince the owner that their dog is overweight. No finite number of reports will convince them that the dog can fly.

In the $\omega^2$ case, the algebra obtained is not identical to the finite case.

**Proposition 3** The class of $\omega^2$-CFs is a non-abelian group under $\ast$. (i.e. it is closed, associative, and every element has an inverse, but it is not commutative).

The fact that $\ast$ is not commutative has interesting consequences, as illustrated in the following example.

**Example** Assume again that the vocabulary contains the predicates $\{\text{heavy}, \text{fly}\}$. Define

$$r_1(s) = \begin{cases} 
\omega & \text{if } s \models \text{fly} \\
0 & \text{otherwise}
\end{cases}$$

$$r_2(s) = \begin{cases} 
0 & \text{if } s \models \neg \text{heavy} \land \text{fly} \\
1 & \text{if } s \models \text{heavy} \land \text{fly} \\
2 & \text{otherwise}.
\end{cases}$$

Hence, $r_1$ says that an agent believes dogs can not fly; moreover, the agent essentially believes that a flying dog is an impossibility. On the other hand, $r_2$ says that an agent believes that light dogs can fly - although the strength of belief in this claim is only finite. Moreover, $r_2$ gives an ordering over less plausible states as well. Note that both $r_1$ and $r_2$ can be either an initial belief state or an observation. The following calculations are immediate.

$$r_1 \ast r_2(s) = \begin{cases} 
\omega & \text{if } s \models \neg \text{heavy} \land \text{fly} \\
\omega + 1 & \text{if } s \models \text{heavy} \land \text{fly} \\
0 & \text{otherwise}.
\end{cases}$$

$$r_2 \ast r_1(s) = \begin{cases} 
\omega & \text{if } s \models \{\text{fly}\} \\
0 & \text{otherwise}.
\end{cases}$$

What is the significance of this example? It shows that conditional beliefs from an observation can be maintained at higher plausibility levels. In both cases, the underlying agent will not believe dogs can fly following revision. But the first revision allows the ordering of states to be refined somewhat at the conditional level. The second revision, on the other hand, washes away the finite level distinctions in the original belief set. This is similar to AGM revision in the sense that recent information seems to carry some particular weight. However, the infinite jumps in plausibility outweigh the preference for recency.

**Nearly Counterfactual Reasoning**

**Motivation**

In this section, we demonstrate how infinite-valued ordinal conditional functions can be useful for reasoning about conditional statements.

**Example** We return to the flying-dog example. Suppose that we initially believe $\neg \text{fly}$ and $\neg \text{hollow}$; in other words, we believe that dogs do not fly and that dogs do not have hollow bones. Now suppose we are told that flying dogs have hollow bones. Informally, we want to revise by the conditional statement $(\text{hollow} \mid \text{fly})$.

Note that $(\text{hollow} \mid \text{fly})$ actually does not give any new information about dogs. This revision should not change the relative ordering of any worlds with a finite strength of belief. However, it does result in a change of belief. If one is later convinced of the existence of flying dogs, then the fact about hollow bones should be incorporated.

We refer to the reasoning in the preceding example as *nearly-counterfactual* revision. It is essentially a form of counterfactual reasoning, in which hypothetical worlds are considered in isolation. At the same time, however, we keep a form of conditional memory at higher ordinal levels. This is not only useful for perspective altering revelations, but we argue it can also be useful for analogical reasoning.

One important feature that is typically taken as a requirement for conditional reasoning is the Ramsey Test. In the context of revision by conditional statements, Kern-Isberner formulates the Ramsey Test as follows: when revising by a conditional, one would like to ensure that revision by $(\psi \mid \phi)$ followed by a revision by $\phi$ should guarantee belief in $\psi$ (Ker99). We suggest that this formulation needs to be refined in order to be used in the case where infinite ranks are possible.

In the case of the flying dog, one is quite likely to accept the conditional $(\text{hollow} \mid \text{fly})$ based on a single report with finite strength. However, a single report of $\text{fly}$ with finite strength will not be believed. If the antecedent of the conditional is “very hard” to believe, then we should not expect the Ramsey Test to hold without some additional condition on the strength of the subsequent report. The problem, in a sense, is that the notion of believing a conditional is quite different than the notion of believing a fact. In order to believe $(\text{hollow} \mid \text{fly})$, we simply need to keep some kind of record of this fact for the unlikely case where we discover that flying dogs happen to exist. On the other hand, in order to believe $\text{fly}$, we really need to make a significant change in our current world view.

**Levels of Implausibility**

Approaches to counterfactual reasoning are typically inspired to some degree by Lewis, who indicates that the truth of a counterfactual sentence is determined by its truth in alternative worlds (Lew73). We can represent this idea with $\omega^2$-CFs. At each limit ordinal $\omega \cdot k$, we essentially have an entirely new plausibility ordering. As $k$ increases, each such ordering represents an increasingly implausible world. However, a sufficiently strong observation can force our beliefs to jump to any of these unlikely worlds. As such, these are not truly counterfactual worlds, because we admit the possibility that they may eventually be believed.

The important property that we can capture with $\omega^2$-CFs is the following: there are some formulas that may be true, yet we can not be convinced to believe them based on any finite number of pieces of “weak evidence.” This allows us
to give the following formal definition of the term nearly counterfactual.

Definition 7 Let $r$ be an OCF. A formula $\phi$ is nearly counterfactual with respect to $r$ just in case there is no $\omega$-CF $r'$ such that $\text{Bel}(r \cdot r') \models \phi$.

The following is an immediate consequence of this definition.

Proposition 4 If $\phi$ is nearly counterfactual with respect to $r$, then there is no finite sequence $r_1, \ldots, r_n$ of $\omega$-CFs such that $\text{Bel}(r \cdot r_1 \cdots r_n) \models \phi$.

We introduce some useful notation.

Definition 8 Let $\phi$ be a formula. An OCF $r$ is a $\phi$-strengthening iff $\text{Bel}(r) = \{ s \mid s \models \phi \}$.

So, a $\phi$-strengthening is just a ranking function where the minimal states are exactly the models of $\phi$. For any formula $\phi$, let $(\phi, n)$ be the $\phi$-strengthening of $\phi$ where models of $\phi$ have plausibility 0 and every other state has plausibility $n$.

Definition 9 Let $r$ be an $\omega^2$-CF. For any limit ordinal $\omega \cdot k$, let $r_k$ be the following partial function:

$$r_k(s) = \begin{cases} r(s), & \text{if } s = \omega \cdot k + c \text{ for some } c \\ \text{undefined otherwise} \end{cases}$$

Hence, $r_k$ is just the restriction of $r$ to those states with plausibility values at level $k$. We say that $\phi$ is believed at level $k$ if $\{ s \mid s \in \text{min}(r_k) \} \models \phi$. Let $\text{poss}(\phi)$ denote the set of natural numbers $k$ such that $s \models \phi$ for some $s$ in the domain of $r_k$.

We can now introduce a form of strengthening with nearly counterfactual conditionals. In the definition, given an $\omega^2$-CF $r$, we let $\deg(s)$ denote the value $k$ such that $r(s) = \omega \cdot k + c$.

Definition 10 Let $r$ be an $\omega^2$-CF and let $\psi, \phi$ be formulas where $\phi$ is nearly counterfactual with respect to $r$. Let $n \in \omega$.

$$r \cdot (n, \psi|\phi)(s) = \begin{cases} r(s), & \text{if } \deg(s) \notin \text{poss}(\phi) \\ r \cdot (\psi, n)(s), & \text{otherwise} \end{cases}$$

We call this function the $n$-strengthening of $\psi$ conditioned on $\phi$. This function finds all levels of $r$ where $\phi$ is possible, and then strengthens $\psi$ at only those levels.

Example Let $r$ again be the plausibility function

$$r(s) = \begin{cases} \omega & \text{if } s \models \text{fly} \\ 10 & \text{if } s \models \text{heavy} \land \neg \text{fly} \\ 0 & \text{otherwise} \end{cases}$$

It is easy to verify that $\text{fly}$ is nearly counterfactual with respect to $r$. Now suppose that we extend the vocabulary to include the predicate symbol $\text{hollow}$. Define a new function $r'$ as follows:

$$r'(s) = \begin{cases} r(s), & \text{if } s \models \neg \text{hollow} \\ r(s) + 1, & \text{if } s \models \text{hollow} \end{cases}$$

This just says that we initially believe our dog does not have hollow bones; however, it is not particularly implausible. It follows that:

- $r'(s) = \omega$ if $s \models \text{fly} \land \neg \text{hollow}$.
- $r'(s) = \omega + 1$ if $s \models \text{fly} \land \text{hollow}$.

From these results, it follows that:

- $r' \cdot (2, \text{hollow}|\text{fly})(s) = r'(s)$, if $s \models \neg \text{fly}$.
- $r' \cdot (2, \text{hollow}|\text{fly})(s) = \omega$, if $s \models \text{fly} \land \text{hollow}$.
- $r' \cdot (2, \text{hollow}|\text{fly})(s) = \omega + 1$, if $s \models \text{fly} \land \neg \text{hollow}$.

So, roughly speaking, after strengthening by $(\text{hollow}|\text{fly})$, we now believe that hollow bones are more plausible in all hypothetical situations where we believe flying dogs are possible.

Note that plausibility of a state is only changed at levels where $\phi$ is considered possible. Since the definition is only applied to nearly counterfactual conditions, this means that only hypothetical states are affected by the strengthening.

It remains to move from conditional strengthening to conditional revision. Recall that, for any $\omega^2$-CF with $\min(r) = \omega \cdot k + c$, we write $\text{fin}(r) = c$.

Definition 11 Let $r$ be an $\omega^2$-CF and let $\psi, \phi$ be formulas where $\phi$ is nearly counterfactual with respect to $r$.

$$r \cdot (\psi|\phi)(s) = \begin{cases} r(s), & \text{if } \deg(s) \notin \text{poss}(\phi) \\ r \cdot (\psi, \text{fin}(r_k))(s), & \text{if } r(s) = \omega \cdot k + c \end{cases}$$

Hence, for revision, we strengthen belief in $\psi$ by the least value that will ensure $\psi$ is believed at level $k$.

Under this definition, we satisfy a modified form of the Ramsey Test.

Proposition 5 Let $r$ be an $\omega^2$-CF and let $s$ be a state with $r(s) = \omega \cdot k + c$. If $r'$ is an $\omega^2$-CF with degree of strength larger than $k$ and $\text{Bel}(r') \models \phi$, then

$$\text{Bel}(r \cdot (\psi|\phi)) \models r' \models \psi.$$
Improvement operators (KP08) are belief change operators that address this issue by introducing a new set of postulates. The most important postulate states that an improvement operator \( \circ \) must have the property that:

**Case I** There exists \( n \in \mathbb{N} \) such that \( B(\Psi \circ^n \phi) \vdash \phi \).

Here \( \Psi \) is an epistemic state, and \( B(\cdot) \) maps an epistemic state to the minimal elements of the underlying ordering. Hence, an improvement operator has the property that an agent will be convinced to believe \( \phi \) after a finite number of improvements. The remaining postulates for a weak improvement operator are essentially the DP postulates applied to the operation \( \circ^n \) obtained from (I). We refer the reader to (KP08) for the complete list of postulates.

We define an analog of (II) as follows. If \( r_\phi \) denotes a \( \phi \)-strengthening, we can express the condition as follows.

**Case II** There exists \( n \in \mathbb{N} \) such that \( Bel(r \circ^n \bar{r} \circ n \cdot \phi) \vdash \phi \).

The truth of this property depends on the degrees of strength of the functions.

**Proposition 6** If \( r \) is an \( \omega \)-CF and \( r_\phi \) is a \( \phi \)-strengthening with finite strength, then \( V' \) holds.

For an epistemic state \( \Psi \) defined by \( \sim \), let \( r_\phi \) be the canonical representation of \( \Psi \). Define \( c_n \) such that \( \Psi \circ \phi \) is obtained by taking the ordering induced by \( r_\phi \circ r(\phi, n) \).

**Proposition 7** For any \( n \in \mathbb{N} \), the operator \( c_n \) is a weak improvement operator.

We call \( c_n \) a finite improvement operator, because the degree of strength is finite. This result is essentially a corollary of Proposition 6, and it suggests that our \( \circ \) operation based on normalized addition is actually the natural extension of improvement to the setting of \( \omega \)-CFs.

The advantage of infinite plausibility values is that they give us greater flexibility in modelling improvement.

**Proposition 8** If \( r \) is an \( \omega^2 \)-CF and \( r_\phi \) is a \( \phi \)-strengthening with finite strength, then \( V' \) does not hold.

This result essentially states that (II) is not a sound property for \( \circ \) if we allow infinite plausibility values. This distinction can be seen in our running example. There is no finite number of improvements that will force the agent to believe that dogs can fly.

It is actually difficult to express the analog of Proposition 7 in the context of \( \omega^2 \)-CFs, because a total pre-order over states can not capture the “infinite jumps” in plausibility encoded by \( \omega^2 \) ordinal ranks. But it is possible to define a correspondence between sequences of orderings and ordinals in \( \omega^2 \).

**Definition 12** Let \( r \) be an \( \omega^2 \)-CF where \( \text{max}(r) = \omega \cdot d + b \) for some \( d, b \). For \( i \leq d \), let \( r_i \) denote the function defined as follows:

1. **domain** \( (r_i) = \{ s \mid r(s) = \omega \cdot i + c \} \).
2. If \( r(s) = \omega \cdot i + c \), then \( r_i(s) = c \).

The following propositions are immediate.

---

**Proposition 9** Each \( r \), is a \( \omega \) ranking, and there exists an \( \omega \)-CF such that \( r_i \sim r \).**

**Proposition 10** For any \( \omega^2 \)-CF \( r \) over a vocabulary \( P \) with \( \text{deg}(r) = d \), there is an extended vocabulary \( P_1 \) and a sequence \( r_0, \ldots, r_d \) of \( \omega \)-CFs such that, for each \( i \leq d \), the \( r_i \) is equivalent (i.e. \( \sim \)) to the restriction of \( r \) to ordinals of degree \( i \).

This result is proved by just extending the vocabulary appropriately with propositional variables that make each infinite jump in the ordinal value definable. By breaking \( r \) into a set of \( \omega \)-CFs, it follows that (I') holds at level \( d \) when \( r_\phi \) has degree of strength \( \omega \cdot d \). Therefore, belief change by normalized addition on \( \omega^2 \)-CFs can really just be seen as a finite collection of improvements as each level. The important point, however, is that no finite sequence of improvements at level \( d \) will ever impact the actual beliefs at lower levels.

**Conditional Belief Revision**

Conditional belief revision was previously addressed by Kern-Isberner, who proposes a set of rationality postulates for conditional revision (Ker99). A concrete approach to conditional revision is also proposed, through the following \( \omega \)-CF:

\[
\begin{align*}
\phi \cdot (\psi(\phi))(s) = & \begin{cases} 
\{ r(s) - r(\psi(\phi)), if s \models \phi \land \psi \\
\{ r(s) + \alpha + 1, if s \models \phi \land \neg \psi \\
\{ r(s), if s \models \neg \phi
\end{cases}
\end{align*}
\]

where \( \alpha = -1 \) if \( r(\phi, \psi) < r(\phi) \), and \( \alpha = 0 \) otherwise. This operation satisfies all of the postulates for conditional revision, as well as the Ramsey Test. We remark, however, that this approach is not well-defined if we allow infinite plausibility values because of the ordinal subtraction on the right hand side. We suggest that this is not just a formal artefact of the theory; conditionals that are “almost certainly” false actually must be treated slightly differently.

In our approach, we essentially require the evidence for \( \phi \) to be substantially stronger than the evidence for the conditional. We suggest that our beliefs following conditional revision should be changed in sort of an infinitesimally small way. While our beliefs about the actual world do not change, our beliefs about some (nearly) impossible world do, in fact, change.

Note that it is actually possible to reconcile our approach with Kern-Isberner’s approach, by using the conditional revision above on each level \( r_k \) of the initial OCF \( r \). At present, we are using a simple strengthening on each level, which actually flattens the plausibility structure after ordinal addition. A combined approach could respect the infinite jumps in plausibility, while satisfying the postulates for conditional revision at each level. We leave an investigation of this combined approach for future work.

**Discussion**

**Conclusion**

In this paper, we have explored the use of infinite ordinals for reasoning about belief change and conditional reasoning. We have shown that allowing plausibility values to range
over $\omega^2$ results in a belief algebra that is only slightly more complicated, and we gain an expressive advantage. In particular, we can represent situations where stubbornly held beliefs are resistant to evidence to the contrary. We have demonstrated that this results in a slightly more expressive class of improvement operators where evidence increases relative belief, but no finite number of improvements will actually lead to a change in the belief state. Finally, we addressed so-called “nearly counterfactual” revision, where we incorporate information that is conditional on a highly unlikely statement.

Future Work

This paper is a preliminary exploration into different applications and formal properties of infinite valued ordinal conditional functions. It remains to move beyond $\omega^2$-CFs, to completely characterize the relationship with improvement operators, and to consider further practical applications.

In the present framework, we have discussed nearly counterfactual reasoning as a tool for keeping a sort of “memory” about unlikely situations, in order to incorporate this information later if necessary. But there is also a natural kind of reasoning that would allow us to use conditionals to reason by analogy about the actual state of the world. Consider the following well-known ambiguity from (Lew73), and originally attributed to Quine:

1. If Caesar was president, he would use nuclear weapons.
2. If Caesar was president, he would use catapults.

As a conditional, we could write both as $(W|C)$, where $W$ stands for a weapon that would be used and $C$ is the condition “Caesar is president.” But (1) suggests that we condition by imagining Caesar alive in the current world. So this is a conditional statement interpreted in the current state of the world. On the other hand, (2) suggests that we consider what would happen in some past world where Caesar exists.

Now suppose that we believe a certain politician is actually very similar to Caesar. If we believe that Caesar would use nuclear weapons, then we may conclude that this “real” politician would also use nuclear weapons. Formally, we could proceed as follows: if some hypothetical world is isomorphic to the current state of the world when we restrict the vocabulary (to not include Caesar), then we can use inferences about the hypothetical world to draw conclusions about the actual world. This is a form of ampliative reasoning that we intend to explore through $\omega^2$-CFs in future work.

References


Characterizing Realizability in Abstract Argumentation

Thomas Linsbichler
TU Wien
Austria

Jörg Pührer and Hannes Strass
Leipzig University
Germany

Abstract
Realizability for knowledge representation formalisms studies the following question: Given a semantics and a set of interpretations, is there a knowledge base whose semantics coincides exactly with the given interpretation set? We introduce a general framework for analyzing realizability in abstract dialectical frameworks (ADFs) and various of its subclasses. In particular, the framework applies to Dung argumentation frameworks, SETAFs by Nielsen and Parsons, and bipolar ADFs. We present a uniform characterization method for the admissible, complete, preferred and model/stable semantics. We employ this method to devise an algorithm that decides realizability for the mentioned formalisms and semantics; moreover the algorithm allows for constructing a desired knowledge base whenever one exists. The algorithm is built in a modular way and thus easily extensible to new formalisms and semantics. We have also implemented our approach in answer set programming, and used the implementation to obtain several novel results on the relative expressiveness of the abovementioned formalisms.

1 Introduction
The abstract argumentation frameworks (AFs) introduced by Dung (1995) have garnered increasing attention in the recent past. In his seminal paper, Dung showed how an abstract notion of argument (seen as an atomic entity) and the notion of individual attacks between arguments together could reconstruct several established KR formalisms in argumentative terms. Despite the generality of those and many more results in the field that was sparked by that paper, researchers also noticed that the restriction to individual attacks is often overly limiting, and devised extensions and generalizations of Dung’s frameworks: directions included generalizing individual attacks to collective attacks (Nielsen and Parsons, 2006), leading to so-called SETAFs; others started offering a support relation between arguments (Cayrol and Lagasquie-Schiex, 2005), preferences among arguments (Amgoud and Cayrol, 2002; Modgil, 2009), or attacks on attacks into arbitrary depth (Baroni et al., 2011). This is only the tip of an iceberg, for a more comprehensive overview we refer to the work of Brewka, Polberg, and Woltran (2014).

One of the most recent and most comprehensive generalizations of AFs has been presented by Brewka and Woltran (2010) (and later continued by Brewka et al., 2013) in the form of abstract dialectical frameworks (ADFs). These ADFs offer any type of link between arguments: individual attacks (as in AFs), collective attacks (as in SETAFs), and individual and collective support, to name only a few. This generality is achieved through so-called acceptance conditions associated to each statement. Roughly, the meaning of relationships between arguments is not fixed in ADFs, but is specified by the user for each argument in the form of Boolean functions (acceptance functions) on the argument’s parents. However, this generality comes with a price: Strass and Wallner (2015) found that the complexity of the associated reasoning problems of ADFs is in general higher than in AFs (one level up in the polynomial hierarchy). Fortunately, the subclass of bipolar ADFs (defined by Brewka and Woltran, 2010) is as complex as AFs (for all considered semantics) while still offering a wide range of modeling capacities (Strass and Wallner, 2015). However, there has only been little concerted effort so far to exactly analyze and compare the expressiveness of the abovementioned languages.

This paper is about exactly analyzing means of expressiveness in natural language and showing examples that some formalisms seem to be able to express but others do not, we tackle the problem in a formal way. We use a precise mathematical definition of expressiveness: a set of interpretations is realizably by a formalism under a semantics if and only if there exists a knowledge base of the formalism whose semantics is exactly the given set of interpretations. Studying realizability in AFs has been started by Dunne et al. (2013, 2015), who analyzed realizability for extension-based semantics, that is, interpretations represented by sets where arguments are either accepted (in the extension set) or not accepted (not in the extension set). While their initial work disregarded arguments that are never accepted, there have been continuations where the existence of such “invisible” arguments is ruled out (Baumann et al., 2014; Linsbichler, Spanring, and Woltran, 2015). Dyrkolbotn (2014) began to analyze realizability for labeling-based semantics of AFs, that is, three-valued semantics where arguments can be accepted (mapped to true), rejected (mapped to false) or neither (mapped to unknown). Strass (2015) started to ana-

*This research has been supported by DFG (project BR 1817/7-1) and FWF (projects I1102 and P25518).
lyze the relative expressiveness of two-valued semantics for ADFs (relative with respect to related formalisms). Most recently, Pührer (2015) presented precise characterizations of realizability for ADFs under several three-valued semantics, namely admissible, grounded, complete, and preferred. The term “precise characterizations” means that he gave necessary and sufficient conditions for an interpretation set to be ADF-realizable under a semantics.

The present paper continues this line of work by lifting it to a much more general setting. We combine the works of Dunne et al. (2015), Pührer (2015), and Strass (2015) into a unifying framework, and at the same time extend them to formalisms and semantics not considered in the respective papers: we treat several formalisms, namely AFs, SETAFs, and (B)ADFs, while the previous works all used different approaches and techniques. This is possible because all of these formalisms can be seen as subclasses of ADFs that are obtained by suitably restricting the acceptance conditions.

Another important feature of our framework is that we uniformly use three-valued interpretations as the underlying model theory. In particular, this means that arguments cannot be “invisible” any more since the underlying vocabulary of arguments is always implicit in each interpretation. Technically, we always assume a fixed underlying vocabulary and consider our results parametric in that vocabulary. In contrast, for example, Dykorkolb (2014) presents a construction for realizability that introduces new arguments into the realizing knowledge base; we do not allow that. While sometimes the introduction of new arguments can make sense, for example if new information becomes available about a domain or a debate, it is not sensible in general, as these new arguments would be purely technical with an unclear dialectical meaning. Moreover, it would lead to a different notion of realizability, where most of the realizability problems would be significantly easier, if not trivial.

The paper proceeds as follows. We begin with recalling and introducing the basis and basics of our work—the formalisms we analyze and the methodology with which we analyze them. Next we introduce our general framework for realizability; the major novelty is our consistent use of so-called characterization functions, firstly introduced by Pührer (2015), which we adapt to further semantics. The main workhorse of our approach will be a parametric propagate-and-guess algorithm for deciding whether a given interpretation set is realizable in a formalism under a semantics. We then analyze the relative expressiveness of the considered formalisms, presenting several new results that we obtained using an implementation of our framework. We conclude with a discussion.

2 Preliminaries

We make use of standard mathematical concepts like functions and partially ordered sets. For a function $f : X \rightarrow Y$ we denote the update of $f$ with a pair $(x,y) \in X \times Y$ by $f \mid_{xy} : X \rightarrow Y$ with $z \mapsto y$ if $z \equiv x$, and $z \mapsto f(z)$ otherwise. For a function $f : X \rightarrow Y$ and $y \in Y$, its preimage is $f^{-1}(y) = \{x \in X \mid f(x) = y\}$. A partially ordered set is a pair $(S, \sqsubseteq)$ with $\sqsubseteq$ a partial order on $S$. A partially ordered set $(S, \sqsubseteq)$ is a complete lattice if and only if every $S' \subseteq S$ has both a greatest lower bound (glb) $\bigwedge S' \in S$ and a least upper bound (lub) $\bigvee S' \in S$. A partially ordered set $(S, \sqsubseteq)$ is a complete meet-semilattice iff every non-empty subset $S' \subseteq S$ has a greatest lower bound $\bigwedge S' \in S$ (the meet) and every ascending chain $C \subseteq S$ has a least upper bound $\bigvee C \in S$.

Three-Valued Interpretations

Let $A$ be a fixed finite set of statements. An interpretation $v : A \rightarrow \{t, f, u\}$ that assigns one of the truth values true ($t$), false ($f$) or unknown ($u$) to each statement. An interpretation is two- valued if $v(A) \subseteq \{t, f\}$, that is, the truth value $u$ is not assigned. Two-valued interpretations $v$ can be extended to assign truth values $v(\varphi) \in \{t, f\}$ to propositional formulas $\varphi$ as usual.

The three truth values are partially ordered according to their information content: we have $u < t$ and $u < f$ and no other pair in $<_i$, which intuitively means that the classical truth values contain more information than the truth value unknown. As usual, we denote by $\leq_i$ the partial order associated to the strict partial order $<_i$. The pair $(\{t, f, u\}, \leq_i)$ forms a complete meet-semilattice with the information meet operation $\sqcap_i$. This meet can intuitively be interpreted as consensus and assigns $t \sqcap_i t = t, t \sqcap_i f = f$, and returns $u$ otherwise.

The information ordering $\leq_i$ extends in a straightforward way to interpretations $v_1, v_2$ over $A$ in that $v_1 \leq_i v_2$ iff $v_1(a) \leq_i v_2(a)$ for all $a \in A$. We say for two interpretations $v_1, v_2$ that $v_2$ extends $v_1$ iff $v_1 \leq_i v_2$. The set $V$ of all interpretations over $A$ forms a complete meet-semilattice with respect to the information ordering $\leq_i$. The consensus meet operation $\sqcap_i$ of this semilattice is given by $(v_1 \sqcap_i v_2)(a) = v_1(a) \sqcap_i v_2(a)$ for all $a \in A$. The least element of $(V, \leq_i)$ is the valuation $v_u : A \rightarrow \{u\}$ mapping all statements to unknown—the least informative interpretation. By $V_\exists$ we denote the set of two-valued interpretations; they are the $\leq_i$-maximal elements of the meet-semilattice $(V, \leq_i)$. We denote by $[v]_2$ the set of all two-valued interpretations that extend $v$. The elements of $[v]_2$ form an $\leq_i$-antichain with greatest lower bound $v = \bigvee [v]_2$.

Abstract Argumentation Formalisms

An abstract dialectical framework (ADF) is a tuple $D = (A, L, C)$ where $A$ is a set of statements (representing positions one can take or not take in a debate), $L \subseteq A \times A$ is a set of links (representing dependencies between the positions), $C = \{C_a\}_{a \in A}$ is a collection of functions $C_a : 2^{\text{par}(a)} \rightarrow \{t, f\}$, one for each statement $a \in A$. The function $C_a$ is the acceptance condition of $a$ and expresses whether $a$ can be accepted, given the acceptance status of its parents $\text{par}(a) = \{b \in S \mid (b, a) \in L\}$. We usually represent each $C_a$ by a propositional formula $\varphi_a$ over $\text{par}(a)$. To specify an acceptance condition, then, we take $C_a(M \cap \text{par}(a)) = t$ to hold iff $M$ is a model for $\varphi_a$.

Brewka and Woltran (2010) introduced a useful subclass of ADFs: an ADF $D = (A, L, C)$ is bipolar iff all links in $L$ are supporting or attacking (or both). A link $(b, a) \in L$ is supporting in $D$ iff for all $M \subseteq \text{par}(a)$, we have that $C_a(M) = t$ implies $C_b(M \cup \{b\}) = t$. Symmetrically, a link $(b, a) \in L$ is attacking in $D$ iff for all $M \subseteq \text{par}(a)$, we
have that $C_a(M \cup \{b\}) = t$ implies $C_a(M) = t$. If a link $(b,a)$ is both supporting and attacking then $b$ has no actual influence on $a$. (But the link does not violate bipolarity.) We write BADFs as $D = (A, L^+ \cup L^−, C)$ and mean that $L^+$ contains all supporting links and $L^−$ all attacking links.

The semantics of ADFs can be defined using an operator $\Gamma_D$ over three-valued interpretations (Brewka and Woltran, 2010; Brewka et al., 2013). For an ADF $D$ and a three-valued interpretation $v$, the interpretation $\Gamma_D(v)$ is given by

$$a \mapsto \bigwedge \{ w(\varphi_a) \mid w \in [v]_2 \}$$

That is, for each statement $a$, the operator returns the consensus truth value for its acceptance formula $\varphi_a$, where the consensus takes into account all possible two-valued interpretations $w$ that extend the input valuation $v$. If this $v$ is two-valued, we get $[v]_2 = \{ v \}$ and thus $\Gamma_D(v)(a) = v(\varphi_a)$.

The standard semantics of ADFs are now defined as follows. For ADF $D$, an interpretation $v : A \to \{ t, f, u \}$ is

- **admissible** iff $v \leq_t \Gamma_D(v)$;
- **complete** iff $\Gamma_D(v) = v$;
- **preferred** iff it is $\leq_t$-maximal admissible;
- a two-valued model iff it is two-valued and $\Gamma_D(v) = v$.

We denote the sets of interpretations that are admissible, complete, preferred, and two-valued models by $\text{adm}(D)$, $\text{com}(D)$, $\text{prf}(D)$ and $\text{mod}(D)$, respectively. These definitions are proper generalizations of Dunng’s notions for AFs: For an AF $(A, R)$, where $R \subseteq A \times A$ is the attack relation, the **ADF associated to $(A, R)$** is $D_{(A,R)} = (A, R, C)$ with $C = \{ \varphi_a \}_{a \in A}$ and $\varphi_a = \bigwedge_{b : (a,b) \in R} \neg b$ for all $a \in A$. AFs inherit their semantics from the definitions for ADFs (Brewka et al., 2013, Theorems 2 and 4). In particular, an interpretation is stable for an AF $(A, R)$ if and only if it is a two-valued model of $D_{(A,R)}$.

A SETAF is a pair $S = (A, X)$ where $X \subseteq (2^A \setminus \{ \emptyset \}) \times A$ is the (set) attack relation. We define three-valued counterparts of the semantics introduced by Nielsen and Parsons (2006), following the same conventions as in three-valued semantics of AFs (Caminada and Gabbay, 2009) and argumentation formalisms in general. Given a statement $a \in A$ and an interpretation $v$ we say that $a$ is acceptable wrt. $v$ if $\forall (B,a) \in X \, \exists a' \in B : v(a') = f$ and $a$ is unacceptable wrt. $v$ if $\exists (B,a) \in X \forall a' \in B : v(a') = t$. For an interpretation $v : A \to \{ t, f, u \}$ it holds that

- $v \in \text{adm}(S)$ iff for all $a \in A$, $a$ is acceptable wrt. $v$ iff $v(a) = t$ and $a$ is unacceptable wrt. $v$ iff $v(a) = f$;
- $v \in \text{com}(S)$ iff for all $a \in A$, $a$ is acceptable wrt. $v$ iff $v(a) = t$ and $a$ is unacceptable wrt. $v$ iff $v(a) = f$;
- $v \in \text{prf}(S)$ iff $v$ is $\leq_t$-maximal admissible; and
- $v \in \text{mod}(S)$ iff $v \in \text{adm}(F)$ and $\exists a \in A : v(a) = u$.

For a SETAF $S = (A, X)$ the corresponding ADF $D_S$ has acceptance formula $\varphi_a = \bigwedge_{(B,a) \in X} v_{-a} = t$ for each statement $a \in A$. (Polberg, 2016)

**Proposition 1.** For any SETAF $S = (A, X)$ it holds that $\sigma(S) = \sigma(D_S)$, where $\sigma \in \{ \text{adm}, \text{com}, \text{prf}, \text{mod} \}$.

**Proof.** Given interpretation $v$ and statement $a$, it holds that $\Gamma_{D_S}(v)(a) = t$ iff $\forall w \in [v]_2 : w(a) = t$ iff $\forall (B,a) \in X \, \exists a' \in B : v(a') = f$ iff $a$ is acceptable wrt. $v$ and $\Gamma_{D_S}(v)(a) = f$ iff $\forall w \in [v]_2 : w(a) = f$ iff $\exists (B,a) \in X \forall a' \in B : v(a') = t$ iff $a$ is unacceptable wrt. $v$. Hence $\sigma(S) = \sigma(D_S)$ for $\sigma \in \{ \text{adm}, \text{com}, \text{prf}, \text{mod} \}$.

**Realizability**

A set $V \subseteq \mathcal{V}$ of interpretations is **realizable** in a formalism $\mathcal{F}$ under a semantics $\sigma$ if and only if there exists a knowledge base $\mathcal{K} \in \mathcal{F}$ having exactly $\sigma(\mathcal{K}) = V$. Pührer (2015) characterized realizability for ADFs under various three-valued semantics. We will reuse the central notions for capturing the complete semantics in this work.

**Definition 1 (Pührer 2015).** Let $V$ be a set of interpretations. A function $f : V_2 \to V_2$ is a **com-characterization of** $V$ iff: for each $v \in V$ we have $v \in V$ iff for each $a \in A$:

- $v(a) \neq u$ implies $f(v_2)(a) = v(a)$ for all $v_2 \in [v]_2$ and
- $v(a) = u$ implies $f(v_2')(a) = t$ and $f(v_2'')(a) = f$ for some $v_2', v_2'' \in [v]_2$.

From a function of this kind we can build a corresponding ADF by the following construction. For a function $f : V_2 \to V_2$, we define $D_f$ as the ADF where the accept formula for each statement $a$ is given by

$$\varphi'_a = \bigvee_{w \in \text{V}_2} \phi_w \quad \text{with} \quad \phi_w = \bigwedge_{w(a') = t} a' \land \bigwedge_{w(a') = f} \neg a'.$$

Observe that we have $v(\varphi_w) = t$ iff $v = w$ by definition. Intuitively, the acceptance condition $\varphi'_a$ is constructed such that $v$ is a model of $\varphi'_a$ if and only if we find $f(v)(a) = t$.

** Proposition 2 (Pührer 2015).** Let $V \subseteq \mathcal{V}$ be a set of interpretations. (1) For each ADF $D$ with $\text{com}(D) = V$, there is a com-characterization $f_D$ for $V$; (2) for each com-characterization $f : V_2 \to V_2$ for $V$ we have $D(f_D) = V$.

The result shows that $V$ can be realized under complete semantics if and only if there is a com-characterization for $V$.

### 3 A General Framework for Realizability

The main underlying idea of our framework is that all abstract argumentation formalisms introduced in the previous section can be viewed as subclasses of abstract dialectical frameworks. This is clear for ADFs themselves and for BADFs by definition; for AFs and SETAFs it is fairly easy to see. However, knowing that these formalisms can be recast as ADFs is not everything. To employ this knowledge for realizability, we must be able to precisely characterize the corresponding subclasses in terms of restricting the ADFs’ acceptance functions. Alas, this is also possible and paves the way for the framework we present in this section. Most importantly, we will make use of the fact that different formalisms and different semantics can be characterized modularly, that is, independently of each other.

Towards a uniform account of realizability for ADFs under different semantics, we start with a new characterization of realizability for ADFs under admissible semantics that is based on a notion similar in spirit to com-characterizations.
Definition 2. Let \( V \) be a set of interpretations. A function \( f : V^2 \to V^2 \) is an adm-characterization of \( V \) iff: for each \( v \in V \) we have \( v \in V \) iff for every \( a \in A \):
\[
\bullet \ v(a) \neq u \ \text{implies} \ f(v)(a) = (v(a)) \ \text{for all} \ v_2 \in [v]_2. \norsymbol
\]
Note that the only difference to Definition 1 is dropping the second condition related to statements with truth value \( u \).

Proposition 3. Let \( V \subseteq V \) be a set of interpretations. (1) For each ADF such that \( \text{adm}(D) = V \), there is an adm-characterization \( f_D \) for \( V \); (2) for each adm-characterization \( f : V_2 \to V_2 \) for \( V \) we have \( \text{adm}(D_f) = V \).

Proof. (1) We define the function \( f_D : V_2 \to V_2 \) as \( f_D(v_2)(a) = v_2(\varphi_a) \) for every \( v_2 \in V_2 \) and \( a \in A \) where \( \varphi_a \) is the acceptance formula of \( a \) in \( D \). We will show that \( f_D \) is an adm-characterization for \( V = \text{adm}(D) \). Let \( v \) be an interpretation. Consider the case \( v \in \text{adm}(D) \) and \( v(a) \neq u \) for some \( a \in A \) and some \( v_2 \in [v]_2 \). From \( v \subseteq \Gamma_D(v) \) we get \( v(\varphi_a) = v(a) \). By definition of \( f_D \) it follows that \( f_D(v_2)(a) = v(a) \). Now assume \( v \notin \text{adm}(D) \) and consequently \( v \notin \Gamma_D(v) \). There must be some \( a \in A \) such that \( v(a) \neq u \) and \( v(a) \notin \Gamma_D(v) \). Hence, there is some \( v_2 \in [v]_2 \) with \( v_2(\varphi_a) \neq v(a) \) and \( f_D(v_2)(a) \neq v(a) \) by definition of \( f_D \). Thus, \( f_D \) is an adm-characterization.

(2) Observe that for every two-valued interpretation \( v_2 \) and every \( a \in A \) we have \( f(v_2)(a) = v_2(\varphi_a) \). Let \( v \in \text{adm}(D_f) \) be an interpretation and \( a \in A \) a statement such that \( v(a) \neq u \). To \( v \in [v]_2 \) we have \( v(a) = v_2(\varphi_a) \). Therefore, by our observation it must also hold that \( f(v)(a) = v(a) \). Thus, by Definition 2, \( v \in V \).

(2) Consider an interpretation \( v \) such that \( v \notin \text{adm}(D_f) \). We show that \( v \notin V \). From \( v \notin \text{adm}(D_f) \) we get \( v \notin \Gamma_D(v) \). There must be some \( a \in A \) such that \( v(a) \neq u \) and \( v(a) \notin \Gamma_D(v) \). Hence, there is some \( v_2 \in [v]_2 \) with \( v_2(\varphi_a) \neq v(a) \) and consequently \( f(v_2)(a) \neq v(a) \). Thus, by Definition 2 we have \( v \notin V \).

When listing sets of interpretations in examples, for the sake of readability we represent three-valued interpretations by sequences of truth values, tacitly assuming that the underlying vocabulary is given and has an associated total ordering. For example, for the vocabulary \( A = \{a, b, c\} \) we represent the interpretation \( \{\text{a} \to \text{t}, \text{b} \to \text{t}, \text{c} \to \text{u}\} \) by the sequence \( \text{tfu} \).

Example 1. Consider the sets \( V_1 = \{\text{uuu}, \text{ttf}, \text{ftf}\} \) and \( V_2 = \{\text{ttf}, \text{ftu}\} \) of interpretations over \( A = \{a, b, c\} \). The mapping \( f = \{\text{ttf} \to \text{ttt}, \text{ttf} \to \text{tft}, \text{ttf} \to \text{ttt}, \text{ttf} \to \text{ttf}, \text{ttf} \to \text{ttt}, \text{ftf} \to \text{ttf}, \text{ff} \to \text{ttf}, \text{ff} \to \text{ttf}, \text{ff} \to \text{ttf}\} \) is an adm-characterization for \( V_1 \). Indeed, the realizing ADF has \( V_1 \) as its admissible interpretations. Indeed, the realizing ADF has \( V_1 \) as its admissible interpretations. Indeed, the realizing ADF has \( V_1 \) as its admissible interpretations. Indeed, the realizing ADF has \( V_1 \) as its admissible interpretations. Indeed, the realizing ADF has \( V_1 \) as its admissible interpretations.

\[
\begin{align*}
\varphi_a^1 & \equiv (a \land b \land \neg c) \lor (a \land \neg b \land a) \\
\varphi_b^1 & \equiv (a \land c) \lor (a \land c) \lor \neg (a \land \neg b \land a) \\
\varphi_c^1 & \equiv (a \land b) \lor (a \land b \land \neg c) \lor (\neg a \land \neg b \land a)
\end{align*}
\]
For \( V_2 \) no adm-characterization exists because \( \text{ttf} \notin V_2 \) but the implication of Definition 2 trivially holds for \( a, b, \) and \( c \).

We have seen that the construction \( D_f \) for realizing under complete semantics can also be used for realizing a set \( V \) of interpretations under admissible semantics. The only difference is that we here require \( f \) to be an adm-characterization instead of a com-characterization for \( V \). Note that admissible semantics can be characterized by properties that are easier to check than existence of an adm-characterization (see the work of Pührer, 2015). However, using the same type of characterizations for different semantics allows for a unified approach for checking realizability and constructing a realizing ADF in case one exists.

For realizing under the model semantics, we can likewise present an adjusted version of com-characterizations.

Definition 3. Let \( V \subseteq V \) be a set of interpretations. A function \( f : V_2 \to V_2 \) is a mod-characterization of \( V \) if and only if: (1) \( f \) is defined on \( V \) (that is, \( V \subseteq V_2 \)) and (2) for each \( v \in V_2 \), we have \( v \in V \) iff \( f(v) = v \).

As we can show, there is a one-to-one correspondence between mod-characterizations and ADF realizations.

Proposition 4. Let \( V \subseteq V \) be a set of interpretations. (1) For each ADF \( D_f \) such that \( \text{mod}(D) = V \), there is a mod-characterization \( f_D \) for \( V \); (2) vice versa, for each mod-characterization \( f : V_2 \to V_2 \) for \( V \) we find \( \text{mod}(D_f) = V \).

Proof. (1) Let \( D \) be an ADF with \( \text{mod}(D) = V \). It immediately follows that \( V \subseteq V_2 \). To define \( f_D \) we can use the construction in the proof of Proposition 3. It follows directly that for any \( v \in V_2 \), we find \( f_D(v) = v \) iff \( v \in V \). Thus \( f_D \) is a mod-characterization for \( V \).

(2) Let \( V \subseteq V_2 \) and \( f : V_2 \to V_2 \) be a mod-characterization of \( V \). For any \( v \in V_2 \) we have:
\[
\begin{align*}
v \in V \iff & v = f(v) \\
\iff & \forall a \in A : (v(a) = f(v)(a)) \\
\iff & \forall a \in A : (v(a) = t \iff f(v)(a) = t) \\
\iff & \forall a \in A : (v(a) = t \iff \exists w \in V_2 : f(w)(a) = t \\
\iff & \forall a \in A : (v(a) = t \iff \exists w \in V_2 : f(w)(a) = t \\
\iff & \forall a \in A : (v(a) = t \iff \exists w \in V_2 : f(w)(a) = t \\
\iff & \forall a \in A : (v(a) = t \iff \exists w \in V_2 : f(w)(a) = t)
\end{align*}
\]
A related result was given by Strass (2015, Proposition 10). The characterization we presented here fits into the general framework of this paper and is directly usable for our realizability algorithm. Wrapping up, the next result summarizes how ADF realizability can be captured by different types of characterizations for the semantics we considered so far.
Theorem 5. Let \( V \subseteq \mathcal{V} \) be a set of interpretations and consider \( \sigma \in \{ \text{adm}, \text{com}, \text{mod} \} \). There is an ADF \( D \) such that
\[
\sigma(D) = V \text{ if and only if there is a } \sigma\text{-characterization for } V.
\]
The preferred semantics of an ADF \( D \) is closely related to its admissible semantics as, by definition, the preferred interpretations of \( D \) are its \( \leq_{\sigma} \)-maximal admissible interpretations. As a consequence we can also describe preferred realizability in terms of \( \sigma\)-characterizations. We use the lattice-theoretic standard notation \( \max_{\leq_{\sigma}} V \) to select the \( \leq_{\sigma} \)-maximal elements of a given set \( V \) of interpretations.

Corollary 6. Let \( V \subseteq \mathcal{V} \) be a set of interpretations. There is an ADF \( D \) with \( \text{prf}(D) = V \) if and only if there is an \( \sigma\)-characterization for some \( V' \subseteq \mathcal{V} \) with \( V \subseteq V' \) and \( \max_{\leq_{\sigma}} V' = V \).

Finally, we give a result on the complexity of deciding realizability for the mentioned formalisms and semantics.

Proposition 7. Let \( \mathcal{F} \in \{ \text{AF SETAF BADF ADF} \} \) be a formalism and \( \sigma \in \{ \text{adm com mod} \} \) be a semantics. The decision problem “\( \text{\sigma} \)-given a vocabulary \( A \) and a set \( V \subseteq \mathcal{V} \) of interpretations over \( A \), is there a \( \sigma \)-ADF \( D \) such that \( \sigma(D) = V \)” can be decided in nondeterministic time that is polynomial in the size of \( V \).

Proof. For all considered \( \mathcal{F} \) and \( \sigma \), computing all \( \sigma \)-interpretations of a given witness \( k_b \in \mathcal{F} \) can be done in time that is linear in the size of \( V \). Comparing the result to \( V \) can also be done in linear time. \( \square \)

3.1 Deciding Realizability: Algorithm 1

Our main algorithm for deciding realizability is a propagate-and-guess algorithm in the spirit of the DPLL algorithm for deciding propositional satisfiability (Gomes et al., 2008). It is generic with respect to (1) the formalism \( \mathcal{F} \) and (2) the semantics \( \sigma \) for which should be realized. To this end, the propagation part of the algorithm is kept exchangeable and will vary depending on formalism and semantics. Roughly, in the propagation step the algorithm uses the desired set \( V \) of interpretations to derive certain necessary properties of the realizing knowledge base (line 2). This is the essential part of the algorithm: the derivation rules (propagators) used there are based on characterizations of realizability with respect to formalism and semantics. Once propagation of properties has reached a fixed point (line 7), the algorithm checks whether the derived information is sufficient to construct a knowledge base. If so, the knowledge base can be constructed and returned (line 9). Otherwise (no more information can be obtained through propagation and there is not enough information to construct a knowledge base yet), the algorithm guesses another assignment for the characterization (line 11) and calls itself recursively.

The main data structure that Algorithm 1 operates on is a set of triples \((v,a,x)\) consisting of a two-valued interpretation \( v \in \mathcal{V}_2 \), an atom \( a \in A \) and a truth value \( x \in \{ t, f \} \). This data structure is intended to represent the \( \sigma \)-characterizations introduced in Definitions 1 to 3. There, a \( \sigma \)-characterization is a function \( f : \mathcal{V}_2 \rightarrow \mathcal{V}_2 \) from two-valued interpretations to two-valued interpretations. However, as the algorithm builds the \( \sigma \)-characterization step by step and there might not even be a \( \sigma \)-characterization in the end (because \( V \) is not realizable), we use a set \( F \) of triples \((v,a,x)\) to be able to represent both partial and incoherent states of affairs. The \( \sigma \)-characterization candidate induced by \( F \) is partial if we have that for some \( v \) and \( a \), neither \((v,a,t)\) nor \((v,a,f)\) are in \( F \); likewise, the candidate is incoherent if for some \( v \) and \( a \), both \((v,a,t)\) and \((v,a,f)\) are in \( F \). If \( F \) is neither partial nor incoherent, it gives rise to a unique \( \sigma \)-characterization that can be used to construct the knowledge base realizing the desired set of interpretations. The correspondence to the characterization function is then such that \( f(v)(a) = x \) if \( (v,a,x) \in F \).

In our presentation of the algorithm we focused on its main features, therefore the guessing step (line 11) is completely “blind”. It is possible to use common CSP techniques, such as shaving (removing guessing possibilities that directly lead to inconsistency). Finally, we remark that the algorithm can be extended to enumerate all possible realizations of a given interpretation set – by keeping all choice points in the guessing step and thus exhaustively exploring the whole search space.

In the case where the constructed relation \( F \) becomes functional at some point, the algorithm returns a realizing knowledge base \( k_b^{\sigma}(F) \). For ADFs, this just means that we denote by \( f \) the \( \sigma \)-characterization represented by \( F \) and set \( k_b^{\text{ADF}}(F) = D_f \). For the remaining formalisms we will introduce the respective constructions in later subsections.

The parameter is parametric in two dimensions, namely with respect to the formalism \( \mathcal{F} \) and with respect to the semantics \( \sigma \). These two aspects come into the algorithm via
so-called propagators. A propagator is a formalism-specific or semantics-specific set of derivation rules. Given a set $V$ of desired interpretations and a partial $σ$-characterization $F$, a propagator $p$ derives new triples $(v, a, x)$ that must necessarily be part of any total $σ$-characterization $F$ for such that $f$ extends $F$. In the following, we present semantics propagators for admissible, complete and two-valued model (in SET)AFs terms semantics, and formalism propagators for BADFs, AFs, and SETAFs.

### 3.2 Semantics Propagators

These propagators (cf. Figure 1) are directly derived from the properties of $σ$-characterizations presented in Definions 1 to 3. While the definitions provide exact conditions to check whether a given function is a $σ$-characterization, the propagators allow us to derive definite values of partial characterizations that are necessary to fulfill the conditions for being a $σ$-characterization.

For admissible semantics, the condition for a function $f$ to be an admissible characterization of a desired set of interpretations $v$ (cf. Definition 2) can be split into a condition for desired interpretations $v ∈ V$ and two conditions for undesired interpretations $v /∈ V$. Propagator $p^{adm}_{adm}$ derives new triples by considering interpretations $v ∈ V$. Here, for all two-valued interpretations $v_2$ that extend $v$, the value $f(v_2)$ has to be in accordance with $v$ on $v_2$’s Boolean part, that is, the algorithm adds $(v_2, a, v(a))$ whenever $v(a) /∈ u$. On the other hand, $p^{adm}_{adm}$ derives new triples for $v /∈ V$ in order to ensure that there is a two-valued interpretation $v_2$ extending $v$ where $f(v_2)$ differs from $v$ on a Boolean value of $v$. Note that while $p^{adm}_{adm}$ immediately allows us to derive information about $F$ for each desired interpretation $v ∈ V$, propagator $p^{adm}_{adm}$ is much weaker in the sense that it only derives a triple if there is no other way to meet the conditions for an undesired interpretation. Special treatment is required for the interpretation $v_u$ that maps all statements to $u$ and is admissible for every ADF. This is not captured by $p^{adm}_{adm}$ and $p^{adm}_{adm}$ as these deal only with interpretations that have Boolean mappings. Thus, propagator $p^{adm}_{adm}$ serves to check whether $v_u ∈ V$. If this is not the case, the propagator immediately makes the relation $F$ incoherent and the algorithm correctly answers “no”.

For complete semantics and interpretations $v ∈ V$, propagator $p^{com}_{com}$ derives triples just like in the admissible case. Propagator $p^{com}_{com}$ deals with statements $a ∈ A$ having $v(a) = u$ for which there have to be at least two $v_2, v_2 /∈ [v]_2$ having $f(v_2)(a) = t$ and $f(v_2 ′)(a) = f$. Hence $p^{com}_{com}$ derives triple $(v_2, a, ¬x)$ if for all other $v_2 /∈ [v]_2$ we find a triple $(v_2, a, x)$. For interpretations $v /∈ V$ it must hold that there is some $a ∈ A$ such that (i) $v(a) /∈ u$ and $f(v)(a) /∈ v(a)$ for some $v ∈ [v]_2$ or (ii) $v(a) = u$ but for all $v ∈ [v]_2$, $f(v)$ assigns the same Boolean truth value to $a$. Now if neither (i) nor (ii) can be fulfilled by any statement $b ∈ A \backslash \{a\}$ due to the current contents of $F$, propagators $p^{com}_{com}$ and $p^{com}_{com}$ derive triple $(v_2, a, ¬a)$ for $v(a) /∈ u$ if needed for $a$ to fulfill (i) and $(v_2, a, ¬x)$ for $v(a) = u$ if needed for $a$ to fulfill (ii), respectively.

#### Example 2
Consider the set $V_3 = \{u, u, u, u, u, u, f, f, f\}$. First, we consider a run of realize(ADF, $adm, V_3, \emptyset$). In the first iteration, propagator $p^{adm}_{adm}$ ensures that $F_{A}$ in line 2 contains $(f, f, a, t)$, $(f, f, a, f)$, $(f, f, f, c, f)$, and $(f, f, f, c, f)$. Based on the latter three tuples and $uf /∈ V_3$, propagator $p^{adm}_{adm}$ derives $(f, f, a, t)$ in the second iteration which together with $(f, f, f, c)$ causes the algorithm to return “no”. Consequently, $V_3$ is not $adm$-realizable. A run of realize(ADF, $com, V_3, \emptyset$) on the other hand returns $com-characterization f$ for $V_3$ that maps $ttf$ to $ttf$, $ttf$ and $ttf$ to $ttf$ and all other $v_2 ∈ V_2$ to $ttf$. Hence, $ADF_{D_1}$, given by the acceptance conditions

\[
p^\downarrow_{com} = a ∧ b ∧ ¬c, \quad p^\downarrow_{com} = ¬a ∧ b ∧ c,
\]

has $V_3$ as its complete semantics.

Finally, for two-valued model semantics, propagator $p^{mod}_{mod}$ derives new triples by looking at interpretations $v ∈ V$. For those, we must find $f(v) = e$ in each $mod$-characterization $f$ by definition. Thus the algorithm adds $(v, a, v(a))$ for each $a ∈ A$ to the partial characterization $F$. Propagator $p^{mod}_{mod}$ looks at interpretations $v /∈ V_2$ and concludes it must hold that $f(v) ≠ v$. Thus there must be a statement $a ∈ A$ with $v(a) = f(v)(a)$, which is exactly what this propagator

\[
p^{mod}_{adm}(V, F) = \{(v_2, a, ¬x) | v ∈ V, v_2 ∈ [v]_2, v(a) ≠ u, x ∈ \{t, f\}, v_2 ′ /∈ [v]_2, v_2 ′ ≠ v_2, v_2 ′ = f(v_2)(a)\}.
\]

\[
p^{mod}(V, F) = \{(v, a, v(a)) | v ∈ V, v_2 ∈ [v]_2, v(a) = u\}.
\]
Algorithm 2 realizePrf(F, V)

Input: • a formalism F
• a set V of interpretations v : A → {t, f, u}

Output: Return some kb ∈ F if one exists or "no" otherwise.

1: if ∃σ ≤ V̸= V then
2: return "no"
3: end if
4: set V< := \{v ∈ V | ∃v' ∈ V : v <, v'\}
5: set X := ∅
6: repeat
7: choose V' ⊆ V< with v' /∈ X
8: set X := X ∪ \{V'\}
9: set Vadm := V ∪ V'
10: if realize(F, adm, Vadm, \emptyset) /≠ "no" then
11: return realize(F, adm, Vadm, \emptyset)
12: end if
13: until ∀V' ⊆ V< : V' /∈ X
14: return "no"

derives whenever it is clear that there is only one statement candidate left. This, in turn, is the case whenever all b ∈ A with the opposite truth value ¬v(a) and all c ∈ A with c ≠ a cannot coherently become the necessary witness any more. The propagator padm checks whether V ⊆ Vf, that is, the desired set of interpretations consists entirely of two-valued interpretations. In that case this propagator makes the relation F incoherent, following a similar strategy as padm.

Preferred Semantics Realizing a given set of interpretations V under preferred semantics requires special treatment. We do not have a σ-characterization function for σ = prf at hand to directly check realizability of V but have to find some V' ⊆ \{v ∈ V | ∃v' ∈ V : v <, v'\} such that V ∪ V' is realizable under admissible semantics (cf. Corollary 6). Algorithm 2 implements this idea by guessing such a V' (line 7) and then using Algorithm 1 to try to realize V ∪ V' under admissible semantics (line 11). If realize returns a knowledge base kb realizing V ∪ V' under adm we can directly use kb as solution of realizePrf since it holds that prf(kb) = V, given that V is an ≤,-antichain (line 2).

3.3 Formalism Propagators

When constructing an ADF realizing a given set V of interpretations under a semantics σ, the function kbADF(F) makes use of the σ-characterization given by F in the following way: v is a model of the acceptance condition ϕσ if and only if we find (v, a, t) ∈ F. Now as bipolar ADFs, SETAFs and AFs are all subclasses of ADFs by restricting the acceptance conditions of statements, these restrictions also carry over to the σ-characterizations. The propagators defined below use structural knowledge on the form of acceptance conditions of the respective formalisms to reduce the search space or to induce incoherence of F whenever V is not realizable.

Bipolar ADFs For bipolar ADFs, we use the fact that each of their links must have at least one polarity, that is, must be supporting or attacking. Therefore, if a link is not supporting, it must be attacking, and vice versa. For canonical realizability, we obtain the polarities of links, i.e. the sets L+ and L−, as defined in Figure 2.

AFs To explain the AF propagators, we first need some more definitions. On the two classical truth values, we define the truth ordering f ≤ t, whence the operations \( \land \) and \( \lor \) with f \( \lor \) t = t and f \( \land \) t = f result. These operations can be lifted pointwise to two-valued interpretations as usual, that is, (v1 \( \land \) v2)(a) = v1(a) \( \land \) v2(a) and (v1 \( \lor \) v2)(a) = v1(a) \( \lor \) v2(a). Again, the reflexive version of ≤, is denoted by ≤,t. The pair (V., ≤,t) of two-valued interpretations ordered by the truth ordering forms a complete lattice with g b ∩L and lubL. This complete lattice has the least element vL : A → \{f\} the interpretation mapping all statements to false, and the greatest element vR : A → \{t\} mapping all statements to true, respectively.

Acceptance conditions of AF-based ADFs have the form of conjunctions of negative literals. In the complete lattice (V., ≤,t), the model sets of AF acceptance conditions correspond to the lattice-theoretic concept of an ideal, a subset of Vf that is downward-closed with respect to ≤,t and upward-closed with respect to ≤t. The propagator directly implements these closure properties: application of pADF ensures that when a σ-characterization F that is neither incoherent nor partial is found in line 8 of Algorithm 1, then there is, for each a ∈ A, an interpretation va such that (va, a, t) ∈ F and v <, a, t for each (v, a, t) ∈ F. Hence va is crucial for the acceptance condition, or in AF terms the attacks, of a and we can define kbADF(F) = (A, \{(b, a) | a, b ∈ A, vL(b) = t\}).

SETAFs The propagator for SETAFs, pSETAF, is a weaker version of that of AFs, since we cannot presume upward-closure with respect to ≤t. In SETAF-based ADFs the acceptance formula is in conjunctive normal form containing only negative literals. By a transformation preserving logical equivalence we obtain an acceptance condition in disjunctive normal form, again with only negative literals; in other words, a disjunction of AF acceptance formulas. Thus, the model set of a SETAF acceptance condition is not necessarily an ideal, but a union of ideals. For the canonical realization we can make use of the fact that, for each a ∈ A, the set V+ = \{v ∈ V | (v, a, t) ∈ F\} is downward-closed with respect to ≤,t, hence the set of models of \( V_{\in \text{max}<} V^+ \land L_v(b) = b \) is exactly V+. The clauses of its corresponding CNF-formula exactly coincide with the sets of arguments attacking a in kbADF(F).

3.4 Correctness

For a lack of space, we could not include a formal proof of soundness and completeness of Algorithm 1, but rather present arguments for termination and correctness.

Termination With each recursive call, the set F can never decrease in size, as the only changes to F are adding the results of propagation in line 3 and adding the guesses in line 11. Also within the until-loop, the set F can never decrease in size; furthermore there is only an overall finite number of triples that can be added to F. Thus at some point
we must have $F_A = \emptyset$ and leave the until-loop. Since $F$ always increases in size, at some point it must either become functional or incoherent, whence the algorithm terminates.  

**Soundness** If the algorithm returns a realizing knowledge base $kb^F_{\sigma}(F)$, then according to the condition in line 8 the relation $F$ induced a total function $f : V_2 \rightarrow V_2$. In particular, because the until-loop must have been run through at least once, there was at least one propagation step (line 2). Since the propagators are defined such that they enforce everything that must hold in a $\sigma$-characterization, we conclude that the induced function $f$ indeed is a $\sigma$-characterization for $V$. By construction, we consequently find that $\sigma(kb^F_{\sigma}(F)) = V$.

**Completeness** If the algorithm answers "no", then the execution reached line 5. Thus, for the constructed set $F$, there must have been an interpretation $v \in V_2$ and a statement $a \in A$ such that $\{(v, a, t), (v, a, f)\} \not\subseteq F$, that is, $F$ is incoherent. Since $F$ is initially empty, the only way it could get incoherent is in the propagation step in line 2. (The guessing step cannot create incoherence, since exactly one truth value is guessed for $v$ and $a$.) However, the propagators are defined such that they infer only assignments (triples) that are necessary for the given $F$. Consequently, the given interpretation set $V$ is such that either there is no realization within the ADF fragment corresponding to formalism $F$ (that is, the formalism propagator derived the incoherence) or there is no $\sigma$-characterization for $V$ with respect to general ADFs (that is, the semantics propagator derived the incoherence). In any case, $V$ is not $\sigma$-realizable for $F$.

4. Implementation

As Algorithm 1 is based on propagation, guessing, and checking it is perfectly suited for an implementation using answer set programming (ASP) (Niemelä, 1999; Marek and Truszczynski, 1999) as this allows for exploiting conflict learning strategies and heuristics of modern ASP solvers. Thus, we developed ASP encodings in the Gringo language (Gebser et al., 2012) for our approach. Similar as the algorithm, our declarative encodings are modular, consisting of a main part responsible for constructing set $F$ and separate encodings for the individual propagators. If one wants, e.g., to compute an AF realization under admissible semantics for a set $V$ of interpretations, an input program encoding $V$ is joined with the main encoding, the propagator encoding for admissible semantics as well as the propagator encoding for AFs. Every answer set of such a program encodes a respective characterization function. Our ASP encoding for preferred semantics is based on the admissible encoding and guesses further interpretations following the essential idea of Algorithm 2. For constructing a knowledge base with the desired semantics, we also provide two ASP encodings that transform the output to an ADF in the syntax of the DIAMOND tool (Eilmuthalser and Strass, 2014), respectively an AF in ASPARTIX syntax (Egly, Gaggl, and Woltran, 2010; Gaggl et al., 2015). Both argumentation tools are based on ASP themselves. The encodings for all the semantics and formalisms we covered in the paper can be downloaded from http://www.dbai.tuwien.ac.at/research/project/adf/unreal/. A selection of them is depicted in Figure 3 on the next page.

5. Expressiveness Results

In this section we briefly present some results that we have obtained using our implementation. We first introduce some necessary notation to describe the relative expressiveness of knowledge representation formalisms (Gogic et al., 1995; Strass, 2015). For formalisms $F_1$ and $F_2$ with semantics $\sigma_1$ and $\sigma_2$, we say that $F_2$ under $\sigma_2$ is at least as expressive as $F_1$ under $\sigma_1$ and write $F_1 <_{\sigma_1} F_2$ if and only if $\Sigma_{F_1}^{\sigma_1} \subseteq \Sigma_{F_2}^{\sigma_2}$, where $\Sigma_{F}^{\sigma} = \{\sigma(kb) | kb \in F\}$ is the signature of $F$ under $\sigma$. As usual, we define $F_1 <_{\sigma_2} F_2$ iff $F_1 <_{\sigma_1} F_2$ and $F_2 <_{\sigma_2} F_1$.

We now start by considering the signatures of AFs, SETAfs and (B)ADF$s for the unary vocabulary $\{a\}$:

$$\Sigma_{adm}^{AF} = \Sigma_{adm}^{SETAF} = \{\{u\}, \{u, t\}\}$$

$$\Sigma_{com}^{AF} = \Sigma_{com}^{SETAF} = \{\{u\}, \{t\}\}$$

$$\Sigma_{prf}^{AF} = \Sigma_{prf}^{SETAF} = \{\{u\}, \{t\}\}$$

$$\Sigma_{mod}^{AF} = \Sigma_{mod}^{SETAF} = \emptyset, \{t\}\}$$

$$\Sigma_{adm}^{ADF} = \Sigma_{adm}^{BADF} = \Sigma_{adm}^{AF} \cup \{\{u, f\}, \{u, t, f\}\}$$

$$\Sigma_{com}^{ADF} = \Sigma_{com}^{BADF} = \Sigma_{com}^{AF} \cup \{\{f\}, \{u, t, f\}\}$$

$$\Sigma_{prf}^{ADF} = \Sigma_{prf}^{BADF} = \Sigma_{prf}^{AF} \cup \{\{f\}, \{t\}\}$$

$$\Sigma_{mod}^{ADF} = \Sigma_{mod}^{BADF} = \Sigma_{mod}^{AF} \cup \{\{f\}, \{t\}\}$$

The following result shows that the expressiveness of the formalisms under consideration is in line with the amount of restrictions they impose on acceptance formulas.

**Theorem 8.** For any $\sigma \in \{adm, com, prf, mod\}$:

1. $AF^\sigma <_{\sigma} SETAF^\sigma$.
2. $SETAF^\sigma <_{\sigma} BADF^\sigma$.
3. $BADF^\sigma <_{\sigma} AF^\sigma$.

**Proof.** (1) $AF^\sigma <_{\sigma} SETAF^\sigma$ is clear (by modeling individual attacks via atom lists). For $SETAF^\sigma \not<_{\sigma} AF^\sigma$, the witnessing model sets over vocabulary $A = \{a, b, c\}$ are $\{uuu, ttf, tft, ftt\} \in \Sigma_{AF}^{\sigma}$ and $\{ttf, tft, ftt\} \in \Sigma_{AF}^{\sigma}$.
Disregarding the possibility of realizing the empty set of interpretations under the two-valued model semantics, we have two interpretations:

\[ \text{int}((A, I)) := \text{s}(A), \text{cterm}(A, A), \text{int}(I), \text{smaller}(A, I) \]

\[ \text{hasU}(I, A) := \text{int}(I), \text{not} \text{member}(\text{t}(A), I), \text{smaller}(A, (n, I)) := \text{s}(A), \text{cterm}(T, H), A < T, \text{int}(I). \]

\[ \text{member}(T, (X, I), (X, B)) := \text{int}((X, I)), \text{member}(T, (T, I), I) := \text{int}((T, I)). \]

\[ \text{member}(T, I, B) := \text{int}(I), \text{smaller}(A, nil) := \text{s}(A). \]

\[ \text{smaller}(A, \text{nil}) := \text{s}(A). \]

\[ \text{int}(\text{nil}). \]

\[ \text{ch}(A, J, t) := \text{sup}(B, A), \text{ch}(A, I, t), \text{diffFT}(I, J, B). \]

\[ \text{ch}(A, J, f) := \text{att}(B, A), \text{ch}(A, I, f), \text{diffFT}(I, J, B). \]

\[ \text{ch}(C, I, f) := \text{s}(C), \text{member}(f(C), I), C != A. \]

\[ \text{ch}(B, I, f) := \text{s}(B), \text{member}(f(B), I), \text{ch}(C, I, f) := \text{s}(C), \text{member}(f(C), I), C != A. \]

\[ \text{ch}(B, I, t) := \text{s}(B), \text{member}(f(B), I), \text{ch}(C, I, f) := \text{s}(C), \text{member}(f(C), I), C != A. \]

\[ \text{ch}(B, I, t) := \text{s}(B), \text{member}(f(B), I), \text{ch}(C, I, f) := \text{s}(C), \text{member}(f(C), I), C != A. \]

\[ \text{ch}(B, I, t) := \text{s}(B), \text{member}(f(B), I), \text{ch}(C, I, f) := \text{s}(C), \text{member}(f(C), I), C != A. \]

% guess a characterization function

\[ 1 \{ \text{ch}(A, I, t); \text{ch}(A, I, t), I := \text{s}(A), \text{int}(I) \}. \]

\[ \frac{\sum_{\text{SETA}} \setminus \sum_{\text{ADB}}}{\sum_{\text{ADB}}}, \text{with } \sigma \in \{ \text{adm, com} \} \text{ and } \tau \in \{ \text{prf, mod} \}. \]

By each pair of arguments of \( A \) being \( t \) in at least one model, a realizing AF cannot feature any attack, immediately giving rise to the model \( \tau \).

\[ (\sigma) \text{ is clear that SETAF}^{\sigma} \triangleleft \triangleleft \text{BADF}^{\sigma} \text{ holds (all parents are always attacking). For BADF}^{\tau} \triangleleft \triangleleft \text{SETA}^{\tau} \text{ the respective counterexamples can be read off the signatures above: for } \sigma \in \{ \text{adm, com} \} \text{ we find } \{ u, t, f \} \in \sum_{\text{ADB}} \setminus \sum_{\text{SETA}} \text{ and for } \tau \in \{ \text{prf, mod} \} \text{ we find } \{ t, f \} \in \sum_{\text{ADB}} \setminus \sum_{\text{SETA}}. \]

\[ (3) \text{ For } \sigma = \text{mod} \text{ the result is known (Strass, 2015, Theorem 14); for the remaining semantics the model sets witnessing ADF}^{\sigma} \triangleleft \triangleleft \text{BADF}^{\sigma} \text{ over vocabulary } A = \{ a \} \text{ are } \{ uu, tu, tt, tf, fu \} \in \sum_{\text{AD}} \setminus \sum_{\text{ADB}} \text{ and } \{ uu, tu, tt, tf, fu \} \in \sum_{\text{ADB}} \setminus \sum_{\text{AD}} \text{ and } \{ tt, tf, fu \} \in \sum_{\text{AD}} \setminus \sum_{\text{ADB}} \text{ and } \{ tt, tf, fu \} \in \sum_{\text{ADB}} \setminus \sum_{\text{AD}}. \]

\[ \text{A witnessing ADF is given by } \varphi_a = a = a \leftrightarrow b. \]

Theorem 8 is concerned with the relative expressiveness of the formalisms under consideration, given a certain semantics. Considering different semantics we find that for all formalisms the signatures become incomparable:

**Proposition 9.** \( \text{F}^{\sigma_1} \triangleleft \triangleleft \text{F}^{\sigma_2} \) and \( \text{F}^{\sigma_2} \triangleleft \triangleleft \text{F}^{\sigma_1} \) for all formalisms \( F_1, F_2 \in \{ \text{AF, SETAF, BADF, ADF} \} \) and all semantics \( \sigma_1, \sigma_2 \in \{ \text{adm, com, prf, mod} \} \) with \( \sigma_1 \neq \sigma_2 \).

**Proof.** First, the result for \( \text{adm and com} \) follows by \( \{ u, t \} \in \sum_{\text{ADB}}, \) but \( \{ u, t \} \notin \sum_{\text{AD}} \text{ and } \{ t \} \in \sum_{\text{AD}} \), such that \( \{ t \} \notin \sum_{\text{ADB}}. \) Moreover, taking into account that the set of preferred interpretations (resp. two-valued models) always forms a \( \leq \)-antichain while the set of admissible (resp. complete) interpretations never does, the result follows for \( \sigma_1 \in \{ \text{adm, com} \} \) and \( \sigma_2 \in \{ \text{prf, mod} \} \). Finally, since a \( kb \in F \) may not have any two-valued models and a preferred interpretation is not necessarily two-valued, the result for \( \text{prf and mod} \) follows. □

Disregarding the possibility of realizing the empty set of interpretations under the two-valued model semantics, we...
obtain the following relation for ADFs.

**Proposition 10.** $(Σ_\text{mod}^\text{ADF} \setminus \{\emptyset\}) ⊆ Σ_\text{prf}^\text{ADF}$.  

**Proof.** Consider some $V ∈ Σ_\text{mod}^\text{ADF}$ with $V ≠ \emptyset$. Clearly $V ⊆ V_2$ and by Proposition 4 there is a mod-characterization $f : V_2 → V_2$ for $V$, that is, $f(v) = v$ iff $v ∈ V$. Define $f' : V_2 → V_2$ for $V$, that is, $f'(v) = f(v) = v$ for all $v ∈ V$ and $f'(a) = \neg v(a)$ for all $v ∈ V \setminus V$ and $a ∈ A$. Now it holds that $f'$ is an adm-characterization of $V' = \{v ∈ V \mid ∀v_2 ∈ [v_2] : v_2 ∈ V\} \cup \{v_\emptyset\}$. Since $\max ≤ V$, $V' = V$ we get that the ADF $D$ with acceptance formula $ϕ^a_f$ for each $a ∈ A$ has $prf(D) = V$ whence $V ∈ Σ_\text{prf}^\text{ADF}$. □

In contrast, this relation does not hold for AFs, which was shown for extension-based semantics by Linschicher, Spanring, and Woltran (2015) (Theorem 5) and immediately follows for the three-valued case.

### 6 Discussion

We presented a framework for realizability in which AFs, SETAFs, BADFs and general ADFs can be treated in a uniform way. The centerpiece of our approach is an algorithm for deciding realizability of a given interpretation-set in a formalism under a semantics. The algorithm makes use of so-called propagators, by which it can be adapted to the different formalisms and semantics. We also presented an implementation of our framework in answer set programing and several novel expressiveness results that we obtained using our implementation. In related work, Polberg (2016) studies a wide range of abstract argumentation formalisms, in particular their relationship with ADFs. This can be the basis for including further formalisms into our realizability framework: all that remains to do is figuring out suitable ADF fragments and developing propagators for them, just like we did exemplarily for Nielsen and Parsons’s SETAFs. For further future work, we could also streamline existing propagators such that they do not only derive absolutely necessary assignments, but also logically weaker conclusions, such as disjunctions of (non-)assignments.

### References


Using Enthymemes to Fill the Gap between Logical Argumentation and Revision of Abstract Argumentation Frameworks

Jean-Guy Mailly
Institute of Information Systems
TU Wien, Austria
jmailly@dbai.tuwien.ac.at

Abstract

In this paper, we present a preliminary work on an approach to fill the gap between logic-based argumentation and the numerous approaches to tackle the dynamics of abstract argumentation frameworks. Our idea is that, even when arguments and attacks are defined by means of a logical belief base, there may be some uncertainty about how accurate is the content of an argument, and so the presence (or absence) of attacks concerning it. We use enthymemes to illustrate this notion of uncertainty about arguments and attacks. Indeed, as argued in the literature, real arguments are often enthymemes instead of completely specified deductive arguments. This means that some parts of the pair (support, claim) may be missing because they are supposed to belong to some “common knowledge”, and then should be deduced by the agent which receives the enthymeme. But the perception that agents have of the common knowledge may be wrong, and then a first agent may state an enthymeme that her opponent is not able to decode in an accurate way. It is likely that the decoding of the enthymeme by the agent leads to mistaken attacks between this new argument and the existing ones. In this case, the agent can receive some information about attacks or arguments acceptance statuses which disagree with her argumentation framework. We exemplify a way to incorporate this new piece of information by means of existing works on the dynamics of abstract argumentation frameworks.

Introduction

Argumentation frameworks (AFs) are a convenient way to represent conflicting information and to deduce which subset of the information can be inferred. For instance, they can be used to model dialogs between several agents (Angoued and Hameurlain 2006) or to analyze on-line discussion between social network users (Leite and Martins 2011). Argumentation can also be useful in a mono-agent setting, for instance to infer non-trivial conclusions from an inconsistent knowledge base (Besnard and Hunter 2001).

The domain called dynamics of argumentation has become a hot topic in recent years, with numerous publications about it. The first ones consider really classical debate scenarios as the source of the dynamic process (Boella, Kaci, and van der Torre 2009a; 2009b; Cayrol, de Saint-Cyr, and Lagasque-Schiex 2010; Baumann and Brewka 2010; Bisquert et al. 2011; 2013; Baumann 2012; Booth et al. 2013). These approaches are perfectly well-suited for classical exchange of arguments between agents. Then, some approaches have proposed to consider new scenarios, closer to what happens with belief change in logical settings (Alchourrón, Gärdenfors, and Makinson 1985; Katsuno and Mendelzon 1991; 1992): these approaches propose to question the existing relation between arguments, and to modify this relation if it is required (Doutré, Herzig, and Perrussel 2014; Nouioua and Würbel 2014; Coste-Marquis et al. 2014a; 2014b; 2015; Baumann and Brewka 2015; Diller et al. 2015).

These works directly deal with the structure of the abstract AFs. An interesting question is “What does AF revision mean when we consider logic-based AFs?”. Indeed, it is not obvious that attacks between arguments can be changed, since they stem from the logical inference relation; for instance, if arguments a and b attack each other because their claims are the negation of each other (rebuttal attack), then it is not accurate to consider that the attack between a and b could be removed. But this is only the case when we consider completely specified deductive arguments (Besnard and Hunter 2001). As argued in the literature (Hunter 2007), the arguments which are used in real situations are often enthymemes, which are partially specified arguments: some parts of the support or some parts of the claim are not described, because it is supposed that they belong to some “common knowledge”. There may be different reasons for an agent not to share some part of her knowledge, such as some cost on the communication process. Then, the agent who receives an enthymeme must decide how to complete the content of the enthymeme to be able to use it. But if the missing formulae to complete the enthymeme are not part of the agent’s beliefs (or at least, are not considered by the agent as the most accurate way to complete the enthymeme), then she will use a badly completed enthymeme in her argumentation framework. We see with this situation that, even with an underlying logical belief base, the nature of arguments and attacks is...
not absolute; it depends on the agent’s beliefs and on her way to complete enthymemes.

So we propose to consider the use of enthymemes in the argumentation process to explain the questionability of some attacks. We illustrate the possibility that a logic-based argumentation framework contains mistaken attacks. Then we show that the existing work on the dynamics of AFs can be used on such enthymeme-based AFs, as soon as a distinction between classical deductive arguments and enthymemes is done in the abstract AF, and that this distinction is used in the revision process.

The paper is organized as follows. The first section presents the background notions required to the understanding of the paper. In particular, we describe briefly belief revision, abstract argumentation and revision of AFs, and logic-based argumentation. Then in the second section, we focus on enthymemes in logic-based AFs; we explain how using enthymemes can be a source of mistaken attacks in the resulting AF. The following section illustrates the revision process on logic-based AFs which contain enthymemes. After the description of a basic approach in which each attack concerning an enthymeme is questionable, we propose a refinement of this approach based on the notion of fixed part of an enthymeme. Finally, the last section concludes the paper and sketches some interesting future work.

**Background**

**Belief Revision**

Belief revision is well-known when an agent’s beliefs are represented in a logical setting. The intuitive idea is “How can an agent incorporate a new piece of information into her beliefs?”, which is not a trivial question when the agent’s previous beliefs and the new piece of information are conflicting. One of the most influential works on this topic is the AGM framework (Alchourrón, Gärdenfors, and Makinson 1985), which gives rationality postulates for belief change operators, when the beliefs are represented as deductively closed sets of formulae. Here we are interested in the adaptation of AGM revision to finite propositional logic by Katsuno and Mendelzon (1991). They explain that revising a formula \( \varphi \) by a formula \( \alpha \) is equivalent to selecting some models of \( \alpha \) which are minimal w.r.t. some plausibility relation. This relation has to satisfy some properties.

**Definition 1** (Katsuno and Mendelzon 1991). A faithful assignment is a mapping from a formula \( \varphi \) to a total pre-order between interpretations \( \preceq_\varphi \) such that:

1. If \( I \models \varphi \) and \( I' \models \varphi \), then \( I \simeq_\varphi I' \);
2. If \( I \models \varphi \) and \( I' \not\models \varphi \), then \( I <_\varphi I' \);
3. If \( \varphi \equiv \varphi' \), then \( \varphi \equiv \varphi ' \).

Then, a KM revision operator \( \sigma \) is a mapping from two formulae \( \varphi, \alpha \) to a new formula such that

\[
\text{mod} (\varphi \circ \alpha) = \min (\text{mod} (\alpha), \varphi).
\]

For instance, the Dalal revision operator can be defined through the pre-order built on the Hamming distance.

**Definition 2** (Hamming 1950; Dalal 1988). The **Hamming distance** between two propositional interpretations \( I, I' \) is the number of assignments which differ between \( I \) and \( I' \), formally: \( d_H(I, I') = |I \Delta I'| \). The total pre-order \( \preceq^H \) is defined by

\[
I \preceq^H I' \iff \min_{J \in \text{mod}(\varphi)} (d_H(I, J)) \leq \min_{J \in \text{mod}(\varphi)} (d_H(I', J))
\]

The Dalal revision operator \( \sigma_D \) is a mapping from two formulae \( \varphi, \alpha \) to a new formula such that

\[
\text{mod} (\varphi \circ_D \alpha) = \min (\text{mod} (\alpha), \preceq^H)
\]

Let us illustrate the behavior of the Dalal revision operator.

**Example 1.** Consider \( V = \{a, b, c, d\} \) and \( \varphi = [(a \land b) \lor (\neg a \land c) \lor \neg (b \lor (a \land c))] \land \neg d \). The models of \( \varphi \) are \( \{a, c\}, \{a, b, c\}, \{a, b, c, d\} \). We revise \( \varphi \) by \( \alpha = a \land \neg b \land c \). The models of \( \alpha \) are \( \{a, c\}, \{a, c, d\} \).

Table 1 gives the Hamming distance between models of \( \varphi \) and models of \( \alpha \).

<table>
<thead>
<tr>
<th>Models of ( \varphi ) ( \varphi )</th>
<th>Models of ( \alpha ) ( \alpha )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \emptyset )</td>
<td>2</td>
</tr>
<tr>
<td>( {a} )</td>
<td>1</td>
</tr>
<tr>
<td>( {c} )</td>
<td>1</td>
</tr>
<tr>
<td>( {a, b} )</td>
<td>2</td>
</tr>
<tr>
<td>( {b, c} )</td>
<td>2</td>
</tr>
<tr>
<td>( {a, b, c} )</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 1: Hamming distance between models of \( \varphi \) and \( \alpha \)

Since the minimal Hamming distance between \( \{a, c\} \) and a model of \( \varphi \) is 1 (\( d_H(\{a, c\}, \{a\}) \) for instance), while the distance between \( \{a, c, d\} \) and any model of \( \varphi \) is at least 2, then \( \{a, c\} <^H \{a, c, d\} \) and so \( \text{mod} (\varphi \circ_D \alpha) = \{\{a, c\}\} \).

**Abstract Argumentation and AF Revision**

An abstract AF is a directed graph which represents the arguments and the attacks between them. The usual problem to solve with such an abstract AF is “How to determine which arguments are accepted?”. This question is tackled in the seminal paper by Dung (1995).

**Definition 3** (Dung 1995). An argumentation framework (AF) is a pair \( F = (A, R) \) where \( A \) is a set of abstract entities called arguments, and \( R \subseteq A \times A \) is the attack relation which represents the conflicts between arguments.

Given a semantics \( \sigma \), the \( \sigma \)-extensions of \( F \), denoted \( \sigma(F) \), are subsets of \( A \) which can be accepted. An argument is then **skeptically accepted** by \( F \) w.r.t. \( \sigma \) iff it belongs to each \( \sigma \)-extension of \( F \).

In this paper, we illustrate our approach on the **stable semantics**: \( S \subseteq A \) is a stable extension of \( F \) (denoted by \( S \in \text{st}(F) \)) iff

- \( \exists x, y \in S \text{ s.t. } (x, y) \in R \);
- \( \forall y \in A \setminus S, \exists x \in S \text{ s.t. } (x, y) \in R \).
Example 2. Given the set of arguments $A = \{x, y, z, t, u\}$, the AF $F_1 = \langle A, R \rangle$ with $R = \{(x, y), (x, t), (y, x), (y, z), (z, u), (t, u)\}$ is given in Figure 1.

![Figure 1: The AF $F_1$](image)

Its stable extensions are $st(F_1) = \{\{x, z\}, \{y, t\}\}$.

As explained in the introduction, the question of change in AFs has been tackled by several approaches. Here we use the translation-based revision from (Coste-Marquis et al. 2014b). The idea of this method is to translate the AF and the semantics into a propositional formula, to use a KM revision operator to perform the expected change, and then to decode the models of the revised formula to obtain a set of revised AFs. The propositional encoding is a generalization of a result from Besnard and Doutre (2004). They have defined a formula $\Xi$, built on propositional variables corresponding to the arguments, such that the set of models of $\Xi$ exactly correspond to the set of stable extensions of an AF. Coste-Marquis et al. (2014b) generalize this encoding with the addition of two other kinds of variables $V = \{att_{x,y} \mid x, y \in A\} \cup \{acc_x \mid x \in A\}$. $att_{x,y}$ means that there is an attack from the argument $x$ to the argument $y$, and $acc_x$ means that the argument $x$ is skeptically accepted.

Definition 4 (Coste-Marquis et al. 2014b). Given an AF $F = \langle A = \{x_1, \ldots, x_n\}, R \rangle$, the stable encoding of $F$ is

$$f_{st}(F) = (\bigwedge_{(x,y) \in R} att_{x,y}) \land (\bigwedge_{(x,y) \notin R} \neg att_{x,y}) \land th_{st}(A)$$

where

$$th_{st}(A) = \bigwedge_{x \in A} [acc_x \Leftrightarrow \forall x_1, \ldots, \forall x_n, (\bigwedge_{y \notin A} (att_{x,y} \Rightarrow \neg z) \Rightarrow x)]$$

In general, $f_{st}(F)$ can be defined for any semantics $\sigma$ as soon as the formula $\Xi$ exists; for semantics with a complexity higher than NP, we can consider for instance QBF encodings to define $\Xi$.

Then, the revision operator is defined as follow:

Definition 5 (Coste-Marquis et al. 2014b). Given $\circ$ a KM revision operator and $\varphi$ a propositional formula built from the set of variables $V$, the translation-based revision operator $\star_\circ$ is defined as

$$F \star_\circ \varphi = dec(f_{\sigma}(F) \circ (\varphi \land th_{\sigma}(A)))$$

with $dec$ a mapping from a formula $\psi$ to a set of AFs $F$ such that each AF $F' \in F$ corresponds to one of the models $\omega$ of $\psi$: $(x, y)$ appears in $F'$ iff $att_{x,y}$ is true in $\omega$.

This general definition allows to change any attack and argument status as long as it is compatible with $\sigma$. If additional constraints should be satisfied,\(^1\) the use of a constrained version is possible:

Definition 6 (Coste-Marquis et al. 2014b). Given $\circ$ a KM revision operator and $\varphi$, $\mu$ two propositional formulae built from the set of variables $V$, the constrained translation-based revision operator $\star_{\circ,\mu}$ is defined as

$$F \star_{\circ,\mu} \varphi = dec(f_{\sigma}(F) \circ (\varphi \land th_{\sigma}(A) \land \mu))$$

To conclude, let us mention a particular revision operator proposed by (Coste-Marquis et al. 2014b): we call $\star_{att}$ (resp. $\star_{acc}$) the translation-based (resp. constrained translation-based) revision operator which gives priority to the minimal change of the attack relation. This operator is similar to the Dalal-based revision revision operator $\star_{\circ,\mu}$, but it uses a weighted version of the Hamming distance such that changing the value of a single $att_{x,y}$ variable is more expensive than changing the value of each $acc_x$ variable.

Example 3. We consider the AF $F_1$ given in Figure 1. We suppose the existence of an integrity constraint $att_{t,u} \land att_{z,u}$, which means that the attacks from $t$ and $z$ to $u$ must not be removed. The result of the revision $F_1 \star_{\circ,\mu} acc_u$ is the AF $F_2$ described in Figure 2. Now the extensions are

![Figure 2: $F_2 = F \star_{\circ,\mu} acc_u$](image)

$s(t(F_2)) = \{\{x, u\}, \{y, u\}\}$, so $u$ is skeptically accepted.

We focus on this kind of revision operators because (Coste-Marquis et al. 2014b) already proposes a way to incorporate a constraint on the attack relation, which is required by our approach. Other revision or update operators could be used instead, but we should adapt their definition to take into account the constraint.

Logic-based Arguments: Deductive Arguments

The question of the exact nature of arguments and attacks is tackled by several approaches which can be gathered under the name structural argumentation. Here we focus on one of the most prominent ones: deductive argumentation (Besnard and Hunter 2001).

Definition 7 (Besnard and Hunter 2001). A deductive argument built from a belief base $\Delta$ is a pair $(\Phi, \alpha)$, where $\Phi$ is called the support and $\alpha$ the claim, such that:

1. $\Phi \subseteq \Delta$,
2. $\Phi \not\vdash \bot$,
3. $\Phi \vdash \alpha$,
4. $\Phi$ is minimal with respect to $\subseteq$ among the sets of formulae which satisfy items 1, 2, and 3.

\(^1\)Such as external constraint depending on the particular application, or some rules of the world.
There is an intuitive explanation to this definition. First the agent is supposed to use her beliefs to justify her claim, which explains the first condition. The second and third conditions guarantee that the claim is actually supported by the beliefs of the agent, but not by conflicting beliefs (for instance, the sentence “It is raining and it is not raining, so I am the Queen of England.” is not an argument at all).

Finally, the last condition ensures that there is no useless piece of information in the support: “It is raining, when it is raining I should use an umbrella, and I love chocolate. So I will use my umbrella.” is not accurate either.

The conflicts between deductive arguments may have different natures. The most general sort of conflict is defined as follow:

Definition 8 (Besnard and Hunter 2001). A defeater for an argument \( (\Phi, \alpha) \) is an argument \( (\Phi', \alpha') \) such that \( \alpha' \vdash \neg(\varphi_1 \land \cdots \land \varphi_n) \), for some \( \{\varphi_1, \ldots, \varphi_n\} \subseteq \Phi \).

It is possible to use deductive arguments to build an argument tree with the arguments and counterarguments which attack and defend a given claim. We can also build a full argumentation framework from the set of deductive arguments generated from a belief base.

Definition 9 (Besnard and Hunter 2014). Given \( A \) the set of deductive arguments generated from the belief base \( \Delta \), the exhaustive graph associated with \( \Delta \) is the AF \( F = (A, R) \) with \( R = \{(x, y) \in A \times A \mid x \text{ is a defeater for } y\} \).

Here we focus on the defeater relation, which is the most general one, but exhaustive graphs can be generated with another attack relation which guarantees additional properties for the defeaters (undercut, rebuttal, and so on). Moreover, these graphs may be infinite in general; Besnard and Hunter propose an approach to circumvent this problem. See (Besnard and Hunter 2014) for more details.

Enthymemes and their Role in Mistaken Attacks

Intuitive Explanation

Before formalizing our approach, we want to explain intuitively, with natural language arguments, why agents can disagree on the attack relation, and more generally why attacks could be questionable. Let us consider the following arguments:

(c) The US army is preparing a secret plan to retreat from Afghanistan (source: Wikileaks).

(b) Our informed sources say that the Wikileaks documents are fake (source: NY Times).

(a) The media cannot be trusted on military issues (source: N. Chomsky).

Now we consider three agents \( A_1, A_2, A_3 \); each of them may have some personal beliefs which are not shared with the other agents.

- \( A_1 \) thinks that Chomsky is the most credible source, and considers that Wikileaks is a media more reliable than NY Times. So her AF is the one given in Fig. 3a.
- \( A_2 \) thinks that Chomsky is a more credible source than NY Times, and NY Times is a more credible source than Wikileaks. She also believes that Wikileaks cannot be seen as a media. So her AF is the one given in Fig. 3b.
- Finally, \( A_3 \) thinks that NY Times is the most credible source, and that Chomsky is not reliable on this topic. So her AF is the one given in Fig. 3c.

These personal AFs may depend on many different parameters (additional information which is not available to each agent, preferences, context, previous experience of each agent, and so on).

Of course, under the assumption that the agents share all their knowledge and beliefs, the personal beliefs of the agents can be represented as additional arguments and we obtain a single AF representing the whole information about a topic. But we think that this assumption is too strong for at least three reasons. First, there may be technical issues with this information sharing; for instance, there may be some cost on communication between agents, or the global amount of information in the network may be too important to be stored in a centralized way. Then, for strategic reasons, agents may choose not to share their knowledge and beliefs. Also, if argument are mined from natural language (for instance, for an analysis of social networks debates), there are likely some implicit pieces of information used in the argumentation process. This explains why some attacks may be questionable.

For instance, if the agents \( A_1 \) and \( A_3 \) consider that agent \( A_2 \) is trustworthy, then they could have to change the attack relation in their own AFs if they receive from agent \( A_2 \) the information “c should be accepted”. On the opposite, if agents vote to determine the arguments statuses, there will be a majority of agents (\( A_1 \) and \( A_3 \)) voting against c (meaning that c is rejected in their AFs), so agent \( A_2 \) should modify the attack relation to incorporate this piece of information in her AF.

Enthymemes with partial support. Now let us formalize this notion of “arguments with partial knowledge”, and their role in the existence of mistaken attacks. Hunter (2007) defines what he calls approximate arguments, which are pairs \( (\Phi, \alpha) \) which do not satisfy the four conditions of deductive
arguments. He classifies them depending on which properties they satisfy, and then focuses on enthymemes. An enthymeme is a pair $\langle \Phi, \alpha \rangle$ such that $\Phi \not\vdash \alpha$, but there is a set $\Psi \subseteq \Delta$ such that $(\Phi \cup \Psi, \alpha)$ is a deductive argument. Intuitively, $\Psi$ represents some “common knowledge” that the agent supposes to be known by her opponents. Then it is not useful for the agent to state the full deductive argument to be able to exchange information and to reach her goal (persuading her opponent, helping to take a decision, negotiating, and so on).

**Example 4.** To illustrate this concept, we borrow a simple example of real life use of enthymemes from (Hunter 2007). Let us consider John and his wife Yoko, who is going outside without an umbrella. If John tells her “You should take your umbrella, because the weather report predicts rain”, there is no formal reason to consider that $(\Phi, \alpha)$ (with $\Phi = \{ \text{rain predicted} \}$ and $\alpha = \text{take umbrella}$) is an argument. It is in fact an enthymeme, because John supposes that $\Psi = \{ \text{rain predicted} \Rightarrow \text{take umbrella} \}$ is an argument. So we can formally define the class of enthymemes (with a partial claim) $\Phi$ as follows:

1. $\langle \{ \alpha, \alpha \Rightarrow \beta \}, \beta \rangle$ with $\beta = \text{“John has bought a copy of the newspaper The Times”;}$
2. $\langle \{ \alpha, \alpha \Rightarrow \gamma \}, \gamma \rangle$ with $\gamma = \text{“John has bought the company which publishes the newspaper The Times.”}$

Similarly to what we have described for enthymemes with a partial claim, in the attack relation can appear. For instance, she may consider that $\alpha$ attacks an argument $b$ because some part of $b$’s support is conflicting with the completed claim (either $\beta$ or $\gamma$ in our example). If she later receives a piece of information which is not compatible with this attack, then she may have to consider a removal of this attack (for instance, because the chosen claim is not accurate).

So we can formally define the class of enthymemes (with partial claims and partial support) as follows:

**Definition 11** (Black and Hunter 2012). Given $d = \langle \Phi, \alpha \rangle$ a deductive argument, an approximate argument $\langle \Phi', \alpha' \rangle$ is an enthymeme for $d$ iff $\Phi' \subseteq \Phi$ and $\alpha \vdash \alpha'$.

Stated otherwise, the pair $\langle \Phi, \alpha \rangle$ is an enthymeme for $(\Phi \cup \Psi, \alpha \land \beta)$, with $\Phi$ the partial support and $\alpha$ the partial claim. In the rest of this paper, we call such a pair $\langle \Phi, \alpha \rangle$ a non-completed enthymeme and $(\Phi \cup \Psi, \alpha \land \beta)$ a completed enthymeme. A completed enthymeme may not satisfy the conditions stated in Definition 7, since the set of formulæ $\Phi$ comes from another agent’s belief base. Moreover, contrary to a fully specified argument stemming from the agent’s beliefs, a completed enthymeme can be questioned.

\[\text{Example 4. To illustrate this concept, we borrow a simple example of real life use of enthymemes from (Hunter 2007). Let us consider John and his wife Yoko, who is going outside without an umbrella. If John tells her “You should take your umbrella, because the weather report predicts rain”, there is no formal reason to consider that $(\Phi, \alpha)$ (with $\Phi = \{ \text{rain predicted} \}$ and $\alpha = \text{take umbrella}$) is an argument. It is in fact an enthymeme, because John supposes that $\Psi = \{ \text{rain predicted} \Rightarrow \text{take umbrella} \}$ is part of the knowledge he shares with Yoko.}

One of the questions tackled in (Hunter 2007) is “How does the agent knows that $\Psi$ is actually part of the common knowledge?” Hunter supposes that each agent has a way to evaluate the plausibility that a given formula will be part of the knowledge shared between her and another agent.

**Definition 10** (Hunter 2007). For each agent $A_i$ whose beliefs are expressed in the propositional language $\mathcal{L}$,

- $\Delta_i \subseteq \mathcal{L}$ denotes her own personal base,
- and for each other agent $A_j$, $\mu_{i,j}$ is a mapping from the language $\mathcal{L}$ to $[0,1]$, such that $\mu_{i,j}(\alpha)$ represents the certainty that $\alpha$ is common to both agents $A_i$ and $A_j$.

**On enthymemes and mistaken attacks** This mapping $\mu_{i,j}$ is used by the agent to build her arguments and decide whether they should be fully specified deductive arguments, or whether enthymemes can be used. The idea is simply to keep only the formulæ $\varphi$ in the support such that the associated value $\mu_{i,j}(\varphi)$ is less than a given threshold $r$; the other ones can be omitted because they are supposed to be known by agent $A_j$.

This process may lead to some problems in the exchange of arguments. There are at least two sources of mistakes.

1. The mapping $\mu_{i,j}$ describes the perception that agent $A_i$ has of her common knowledge with $A_j$. If this perception is wrong, then there could be some exchange of enthymemes that the agent $A_j$ cannot decode accurately.

2. Even with a good evaluation of the common knowledge by $\mu_{i,j}$, the choice of a bad threshold could also lead to enthymemes that the other agent cannot decode.

In both these situations, agent $A_j$ receives some “argument” $a = \langle \Phi, \alpha \rangle$ which is not fully specified, and then she has to complete the support with some $\Psi$ from her own belief base, which could of course lead to the addition of some attacks from an existing argument $b$ to this new argument $a$, for instance if the claim of $b$ is the formula $\neg \psi$, for some $\psi \in \Psi'$. Even if for low-level treatments, it can be represented as $a' = \langle \Phi \cup \Psi', \alpha \rangle$ (for instance to determine if possibly new incoming arguments attack it), at a higher level it is still the argument $a$ which is used. Indeed, this $a'$ is not an argument that $A_j$ has built by herself, since some of the premises are not part of her belief base. Then, agent $A_j$ can receive a new piece of information about the argument $a$ which is incompatible with the attack from $b$ to $a$ (the simplest example being “$a$ and $b$ should be accepted together”). So she has to build a new internal state $a''$ from a subset $\Psi''$ from her belief base; for the same reason as previously, at a higher level it is still the argument $a$ originally built from agent $A_i$’s beliefs.
Dynamics of AFs and Enthymemes

Building a Dung’s AF from Enthymemes

For several reasons, the use of an abstract AF by the agent is interesting, even when she uses an underlying belief base. For instance, the development of efficient approaches to solve abstract argumentation problems permits to obtain the conclusion of the agent’s AF with respect to several semantics and inference policies (see for instance the competition of argumentation solvers (Thimm and Villata 2015)). But to avoid the loss of information about the nature of arguments and attacks, we propose to refine the definition of the AF.

Definition 12. Given D and E which denote respectively the agent’s deductive arguments and enthymemes, the agent’s enthymeme-based AF is \( F(D, E) = \langle A, R \rangle \) with

- \( A = D \cup E \);
- \( R = R_D \cup R_E \);
- \( R_D \subseteq D \times D \) the set of certain attacks (between deductive arguments);
- \( R_E \subseteq (A \times A) \setminus (D \times D) \) the set of questionable attacks (concerning at least one enthymeme).

Computing the extensions of such an AF is identical to the process for classical Dung’s AFs; differentiating both kinds of attacks is useful only for the dynamics scenarios such as revision.

In this setting, each attack can be added or removed as soon as it concerns at least one enthymeme. We will refine this later.

Example 5. Let \( F_3 \) be the enthymeme-based AF presented in Fig. 4. Arguments with rounded corners are the enthymemes while the other ones are deductive arguments.

Similarly, the dashed arrows represent the questionable attacks, while the other ones are the certain attacks. In this example, we suppose that the agent has received the enthymemes in the following way:

- \( e_1 = \{\{\alpha\}, \gamma\} \), which has been completed by \( \Psi_1 = \{\alpha \Rightarrow \beta, \beta \Rightarrow \gamma\} \);
- \( e_2 = \{\{\eta\}, \top\} \), which has been completed by \( \Psi_2 = \{\eta \Rightarrow \neg \epsilon\} \) in the support and \( \neg \epsilon \) in the claim.

With this AF, the accepted arguments are \( \{e_2, d_1\} \).

Applying Dynamics of Abstract AFs to Enthymeme-based AFs

The existence of mistaken attacks in an enthymeme-based AF can be tackled through some approaches of the dynamics of abstract argumentation (Bisquert et al. 2013; Doutre, Herzig, and Perrussel 2014; Coste-Marquis et al. 2014a; 2014b; 2015). In the case when some arguments and the relations between them are certain (in particular, when they are fully specified arguments instead of enthymemes), integrity constraints can simply be added to these revision/update/enforcement operators to ensure that forbidden attacks will not be added, and mandatory attacks will not be removed. Since it is already defined by (Coste-Marquis et al. 2014b), we will exemplify the dynamics of argumentation with their constrained revision approach, presented previously.

We can encode an integrity constraint to fix the attacks and non-attacks concerning the deductive arguments into the setting from (Coste-Marquis et al. 2014b).

Definition 13. Given \( F(D, E) \) an enthymeme-based AF, the integrity constraint on deductive arguments is

\[
\mu_D = \left( \bigwedge_{(x,y) \in R_D} \text{att}_{x,y} \right) \land \left( \bigwedge_{(x,y) \in (D \times D) \setminus R_D} \neg \text{att}_{x,y} \right)
\]

Now, if the agent receives some piece of information about the arguments statuses or the attack relation, then she can use the AF revision operator \( *_{\mu_D} \) as defined previously in the case when this new piece of information disagrees with the current AF. This revision operator guarantees that the relations between deductive arguments will not be modified during the revision process, which is desirable since they are directly stemming from the logical inference relation.

Example 5 Continued. We continue the previous example. The agent receives the piece of information “\( e_1 \) should be accepted”, which corresponds to the formula acc\( _{e_1} \). The integrity constraint is \( \text{att}_{d_1, d_1} \land \neg \text{att}_{d_2, d_2} \), which ensures that the attacks between the deductive arguments \( d_1 \) and \( d_2 \) will not be modified. The possible results are given in Fig. 5.

![Figure 5: Possible Results of the Revision](image-url)

The exact change operator which should be used depends on the properties expected for the process, for instance it is well-known that performing an update (Bisquert et al. 2013; Doutre, Herzig, and Perrussel 2014) is accurate when the change is explained by an evolution of the world, while performing a revision (Coste-Marquis et al. 2014a; 2014b) is accurate when the evolution only concerns the agent’s beliefs about the world; thus these operations do not
satisfy the same properties. Similarly, among the different approaches in the state of the art about the dynamics of AFs, each of them do not have the same expressivity. For instance, the revision approach described in (Coste-Marquis et al. 2014a) permits to revise by a formula concerning the extensions, while the translation-based approach illustrated here permits to revise by a formula concerning skeptical acceptance of arguments and attacks at the same time. So, the choice of a change operator completely depends on the application and the agent’s needs and preferences. In the following, we continue to consider revision to be consistent with the previous example, but update and extension enforcement (Baumann and Brewka 2010) could be considered as well.

**From Revised AFs to new Completed Enthymemes**

After obtaining the result of the revision process, the agent should now decode this result which of the revised AFs is the most plausible real AF corresponding to her beliefs, and which enthymemes should be internally modified (and how they should be internally modified) to ensure that the abstract AF and the logic-based AF coincide.

**Definition 14.** Let $F$ be the set of AFs obtained from the revision process. For each $F' \in F$, $F'$ is called an acceptable AF if for each attack which differs between the original AF $F$ and $F'$, the agent’s belief base contains some formulae which allow to complete the enthymemes s.t. this new completion is consistent with the attacks in $F'$.

**Example 5 Continued.** Continuing the previous example, let us suppose that the agent’s belief base contains the formulae $\Psi' = \{\alpha \Rightarrow \theta, \theta \Rightarrow \gamma\}$. Then the enthymeme $e_1$ can be completed into $\{(\alpha, \alpha \Rightarrow \theta, \theta \Rightarrow \gamma), \gamma\}$, which leads to the acceptable AF $F_4$ given in Fig. 6a. Similarly, if the agent’s belief base contains the formulae $\Psi'' = \{\eta \Rightarrow \iota\}$, then the agent can consider the acceptable AF $F_5$ given in Fig. 6b, since $e_2$ can be completed into $\{(\eta, \eta \Rightarrow \iota), \iota\}$.

When the set of acceptable AFs is not a singleton, we can consider two different solutions:

- the agent can keep the whole set as the result.
- the agent can use external information (preferences between AFs, preferences between formulae in the enthymemes, and so on) to select a single acceptable AF as the result.

None of them is in general more desirable than the other one, the choice depends on the situation (specific application, user’s preferences, computational issues, ...).

**Refining Questionable Attacks**

In the previous parts, we suppose that each attack concerning an enthymeme is questionable. But we can be more precise in the definition of the enthymeme-based AF. Indeed, even when we consider an enthymeme $e$, some of the attacks concerning it may be certain. We know that some parts of $e$, that we have called previously the partial support and the partial claim, are fixed. If the reason of an attack between $e$ and a deductive argument is a logical conflict involving one of these fixed parts of $e$, then this attack can be considered as certain. Similarly, when we consider another enthymeme $e'$, if there is an attack between $e$ and $e'$ which is stemming from the fixed part of $e$ and the fixed part of $e'$, then this attack cannot be removed either.

Let us first formalize this notion of fixed part.

**Definition 15.** If $a = \langle \Phi, \alpha \rangle$ is a deductive argument or a non-completed enthymeme, then the fixed part of $a$ is $fix(a) = \Phi \cup \{\alpha\}$.

If $a = \langle \Phi \cup \Psi, \alpha \land \beta \rangle$ is a completed enthymeme, then $fix(a) = \Phi \cup \{\alpha\}$.

So if we consider a fully specified deductive argument or a non-completed enthymeme, the fixed part is the set of all the formulae involved in it. But when we consider an enthymeme completed with the agent’s beliefs, then the fixed part is the set of formulae which appear in the enthymeme that the agent has originally received, but do not appear in the completed version of it.

**Example 5 Continued.** Let us consider again the arguments $d_1$, $d_2$, $e_1$ and $e_2$. The fixed parts of the deductive arguments

\[
e_1 = \{\langle \alpha, \alpha \Rightarrow \theta, \theta \Rightarrow \gamma, \gamma\}\}
\]

\[
d_1 = \{\langle \delta, \delta \Rightarrow \beta, \beta \Rightarrow \gamma, \gamma\}\}
\]

\[
d_2 = \{\langle \epsilon, \epsilon \Rightarrow \delta, \delta\}\}
\]

\[
e_2 = \{\langle \eta, \eta \Rightarrow \iota, \iota\}\}
\]

![Figure 6: Different Acceptable AFs](image-url)

(a) Instantiation of $F_4$ by a New Completion of $e_1$

(b) Instantiation of $F_5$ by a New Completion of $e_2$
are trivially the union of their support and their claim. The result is more interesting for the enthymemes:

- \( \text{fix}(e_1) = \{ \alpha, \gamma \} \);
- \( \text{fix}(e_2) = \{ \eta, \top \} \).

Now let us define the involved part of an argument in an attack.

**Definition 16.** Let \( a = (\Phi, \alpha) \) and \( b = (\Phi', \alpha') \) be two arguments (deductive arguments, completed enthymemes or non-completed enthymemes). If there is an attack between \( a \) and \( b \), then the involved part of \( a \) in the conflict between \( a \) and \( b \), denoted by \( \text{inv}_b(a) \), is the set \( \Psi \subseteq \Phi \cup \{ \alpha \} \) such that \( (\bigwedge_{\psi \in \Psi} \psi) \land (\bigwedge_{\psi' \in \Phi \cup \{ \alpha' \}} \neg \psi') \uparrow \bot \) and \( \Psi \) is minimal w.r.t. \( \subseteq \). Otherwise, \( \text{inv}_b(a) = \text{inv}_a(b) = \emptyset \).

So, if we have a rebuttal conflict between \( a \) and \( b \) (meaning that the claims are the contradiction of each other) then \( \text{inv}_b(a) = \{ \alpha \} \) and \( \text{inv}_a(b) = \{ \alpha' \} \). If the conflict is an undercut from \( a \) to \( b \) (meaning that the claim \( \alpha \) of \( a \) is conflicting with some part \( \alpha' \) of the support of \( b \)), then \( \text{inv}_a(a) = \{ \alpha \} \) and \( \text{inv}_b(b) = \{ \alpha' \} \).

**Example 5 Continued.** Now we can see which parts of the arguments \( d_1, d_2, e_1 \) and \( e_2 \) are involved in the conflicts. The certain attack \( (d_2, d_1) \) comes from the contradiction between \( \delta \) and \( \neg \delta \), so \( \text{inv}_d(d_2) = \{ \neg \delta \} \) and \( \text{inv}_d(d_1) = \{ \delta \} \). Concerning the questionable attacks, we have:

- \( \text{inv}_{e_1}(d_1) = \{ \beta \land \neg \gamma \} \) and \( \text{inv}_{e_1}(e_1) = \{ \beta \Rightarrow \gamma \} \);
- \( \text{inv}_{e_2}(d_2) = \{ \varepsilon \} \) and \( \text{inv}_{e_2}(e_2) = \{ \neg \varepsilon \} \).

Now we can refine the definition of an enthymeme-based AF.

**Definition 17.** Given \( D \) and \( E \) which denote respectively the agent’s deductive arguments and enthymemes, the agent’s refined enthymeme-based AF is \( F(D, E) = (\mathcal{A}, \mathcal{R}) \) with

- \( \mathcal{A} = D \cup E \);
- \( \mathcal{R} = R_C \cup R_Q \);
- \( R_C = \{ (x, y) \in A \times A \mid \text{inv}_y(x) \subseteq \text{fix}(x) \} \) and \( \text{fix}(y) \subseteq \text{fix}(y) \); the set of certain attacks;
- \( R_Q \subseteq (A \times A \times A) \times R_C \); the set of questionable attacks.

We use \( R_D \) as a notation for \( R_C \cap (D \times D) \), which is the set of attacks between deductive arguments.

Of course, if an argument is a fully specified deductive argument, then the part of it which is involved in conflicts is a fixed part. So to refine the AF, we need to check if it is the case with the enthymemes.

**Example 5 Continued.** Studying the relations between involved parts and fixed parts for the enthymemes \( e_1 \) and \( e_2 \), we obtain the following:

- \( \text{inv}_{e_1}(e_1) = \{ \beta \Rightarrow \gamma \} \not\subseteq \text{fix}(e_1) = \{ \alpha, \gamma \} \);
- \( \text{inv}_{e_2}(e_2) = \{ \gamma \} \not\subseteq \text{fix}(e_2) = \{ \eta, \top \} \).

So none of the attacks \( (e_2, d_2) \) and \( (d_1, e_1) \) is certain.

But we can exhibit more interesting cases, for which the use of a refined enthymeme-based AF leads to another result than the basic enthymeme-based AF.

**Example 6.** Let \( d_3 = \{ \{ \nu, \nu \Rightarrow \neg \lambda \}, \neg \lambda \} \) be a deductive argument, and \( e_3 = \{ \{ \kappa \}, \lambda \} \) an enthymeme, which can be completed for instance by the additional support \( \Phi' = \{ \kappa \Rightarrow \lambda \} \).

It is easy to see here that \( \text{inv}_{d_3}(e_3) = \{ \lambda \} \not\subseteq \text{fix}(e_3) = \{ \kappa, \lambda \} \), so the conflict between \( d_3 \) and \( e_3 \) is not questionable, and the AF corresponding to these arguments is \( F_6 \) given in Fig. 7.

\[
\begin{align*}
e_3 &= \{ \{ \kappa \}, \kappa \Rightarrow \lambda \}, \lambda \} \\
d_3 &= \{ \{ \nu, \nu \Rightarrow \neg \lambda \}, \neg \lambda \}
\end{align*}
\]

**Figure 7: The Refined Enthymeme-based AF \( F_6 \)**

When we consider these refined enthymeme-based AFs in the revision process, the integrity constraint must be adapted to take into account each certain attack, and not only the ones between deductive arguments:

**Definition 18.** Given \( F(D, E) \) a refined enthymeme-based AF; the integrity constraint on certain attacks is

\[
\mu_C = \left( \bigwedge_{(x, y) \in R_C} \text{att}_{x, y} \right) \land \left( \bigwedge_{(x, y) \in (D \times D) \setminus R_D} \neg \text{att}_{x, y} \right)
\]

This new constraint ensures that a certain attack will not be removed during the revision process, and that attacks between deductive arguments will not be added if they do not belong to the original AF.

**Back to Chomsky Example.**

To conclude, let us formalize the intuitive “Chomsky example”, showing the different completions of enthymemes which lead to the different agents AFs. We use the following propositional variables: \( \text{retreat} \) means that the US army will retreat; \( \text{wkf} \) means that the Wikileaks information about \( \text{retreat} \) is true; \( \text{mnt} \) means that the Wikileaks documents are \( \text{false} \); \( \text{mnt} \) means that \( \text{media} \) cannot be trusted on military issues. As they are stated, the arguments \( a, b \) and \( c \) which are shared by the agents are these ones:

\[
a = \{ \{ \text{mnt} \}, \top \}, \ b = \{ \{ \text{wkf} \}, \top \}, \ c = \{ \{ \text{wkfr} \}, \top \}
\]

All of them are enthymemes. For all the agents, the completion of \( c \) is \( \{ \text{wkfr}, \text{wkfr} \Rightarrow \text{retreat}, \text{retreat} \} \). But they disagree on the completion of the other enthymemes. Agent \( A_1 \) considers that \( a = \{ \{ \text{mnt}, \text{mnt} \Rightarrow \neg \text{wkf}, \text{mnt} \Rightarrow \neg \text{wkfr} \}, \neg \text{wkf} \land \neg \text{wkfr} \} \) and \( b = \{ \{ \text{wkf}, \top \} \). Agent \( A_2 \) completes the enthymemes as follows: \( a = \{ \{ \text{mnt}, \text{mnt} \Rightarrow \neg \text{wkf}, \text{mnt} \Rightarrow \neg \text{wkfr} \}, \neg \text{wkf} \land \neg \text{wkfr} \} \) and \( b = \{ \{ \text{wkf}, \text{wkf} \Rightarrow \neg \text{wkfr} \}, \neg \text{wkfr} \}. \)

Finally, agent \( A_3 \) uses these completions of enthymemes: \( a = \{ \{ \text{mnt} \}, \top \} \) and \( b = \{ \{ \text{wkfr}, \text{wkfr} \Rightarrow \neg \text{wkfr} \}, \neg \text{wkfr} \} \).

These completions of enthymemes lead to the AFs described in Figure 3, with all arguments which are enthymemes, and all attacks which are questionable. So, in case of a revision, the revision operator is used with the
integrity constraint $\top$, which is equivalent to a revision without a constraint.

We mentioned in the introduction two scenarios which require to use dynamics of argumentation techniques. First, we suppose that agent $A_2$ is considered to be trustworthy by other agents. Then, when she says that $c$ should be accepted (which is represented by the formula $\text{acc}_c$), the other agents have to revise their AF with this new piece of information.

The result of the revision for $A_1$, with a corresponding completion of enthymemes which are modified because of the revision, is given in Figure 8.

$$a = \langle \{\text{mnt}, \text{mnt} \Rightarrow \neg \text{wkf}, \neg \text{wkf} \} \rangle$$

Figure 8: Revision for Agent $A_1$

Similarly, for $A_3$, Figure 9 describes the possible revised AFs, with the modified enthymemes corresponding to it.

$$a = \langle \{\text{mnt}, \text{mnt} \Rightarrow \neg \text{wkf}, \neg \text{wkf} \} \rangle$$

Figure 9: Possible Revisions for Agent $A_3$

Finally, the other scenario was a vote on the acceptance status of $c$. Since the majority of agents rejects $c$, $A_2$ has to revise her AF by $\neg \text{acc}_c$ to find an agreement with the majority. Possible results are described in Figure 10.

$$a = \langle \{\text{mnt}, \text{mnt} \Rightarrow \neg \text{wkf}, \neg \text{wkf} \} \rangle$$

Figure 10: Possible Revisions for Agent $A_2$

### Conclusion

In this paper, we argue that, in realistic situations, agents do not share all their knowledge and beliefs. There are different possible reasons, among them, technical and strategical reasons seem to be the most intuitive explanations. Also, implicit information is frequently used in natural language argumentation (on social networks for instance). When this situation occurs, there is some uncertainty in the resulting argumentation frameworks built by the agents. It is likely that agents’ opinion about arguments’ meaning and relations between arguments will differ; there may be some misunderstanding in the communication process which leads to mistakes in the generation of arguments and attacks. Here, this is formalized with the use of enthymemes, instead of deductive arguments, to represent the uncertain nature of some arguments. For this reason, the reception of a new piece of information (supposed to be reliable) can force the agents to question the current attack relation to obtain a result which is compatible with the new piece of information. We have described formally how the use of enthymemes in the argumentation process can lead to the existence of these mistaken attacks, and how to define an argumentation framework which makes the distinction between the certain attacks and the questionable attacks. Then, we have seen that the existing works on the dynamics of abstract AFs can be used to perform a change on an enthymeme-based AF when it is required to incorporate a new piece of information. Here, we exemplify it with the translation-based revision from (Coste-Marquis et al. 2014b), but it can be adapted to any revision, update or enforcement approach as soon as it is possible to consider an integrity constraint on the attack relation.

This paper only presents some preliminary work on this question. Many future works can be envisioned. First, we want to model the uncertainty by other means than enthymemes. For instance, using weights could lead to the definition of original change operators which define the notion of minimal change w.r.t. these weights; it would then be more expensive to change a single attack which has a high weight than to change several attacks with low weights. Determining what can be the origin of these weights is particularly interesting. Combined with the use of enthymemes, we think that giving a low weight to an attack which is easy to modify (because there are many possible completions of enthymemes in the belief base) is an interesting way to tackle the problem. For the EAFs defined in this paper, as well as the weighted approach mentioned above, several questions are opened. We have made the simplifying hypothesis that the agents will have some possible completion of arguments at their disposal, which is not always the case in real world situations. Similarly, the revision operators may lead to an empty-set of results (because of the integrity constraint), or on the opposite, to a non-singleton set of results. All these cases must be investigated to ensure the possibility of some practical applications. The complexity of revising such framework, compared to the original revision approach in the abstract setting, will also be studied to be able to identify which approaches can be used for real applications. We also plan to use some of the existing pieces of software for the dynamics of AFs (in particular the one described in (Coste-Marquis et al. 2015; Wallner, Niskanen, and Järvisalo 2015) for extension enforcement), to study the scalability of the approaches on practical examples. But it requires first an important work to build argumentation graphs from logical knowledge-bases, since the existing works focus only on argumentation trees (Efthathiou and Hunter 2008; 2011; Besnard et al. 2010).

### Acknowledgments

This work was as supported by the Austrian Science Fund (FWF) under grants P25521 and I1102.
References
Iterated Ontology Revision by Reinterpretation

Özgür L. Özcep
Institute of Information Systems (IFIS)
University of Lübeck, Germany
oezcep@ifis.uni-luebeck.de

Abstract

Iterated applications of belief change operators are essential for different scenarios such as that of ontology evolution where new information is not presented at once but only in piecemeal fashion within a sequence. I discuss iterated applications of so-called reinterpretation operators that trace conflicts between ontologies back to the ambiguous of symbols and that provide conflict resolution strategies with bridging axioms. The discussion centers on adaptations of the classical iteration postulates according to Darwiche and Pearl. The main result of the paper is that reinterpretation operators fulfill the postulates for sequences containing only atomic triggers. For complex triggers, a fulfillment is not guaranteed and indeed there are different reasons for the different postulates why they should not be fulfilled in the particular scenario of ontology revision with well-developed ontologies.

1 Introduction

Iterated applications of belief change operators are essential for different scenarios such as that of ontology evolution where new information is not presented at once but only in piecemeal fashion within a sequence. Ontology evolution is a form of ontology change (Flouris et al. 2008) where an ontology modification is triggered by changes in the domain or in the conceptualization. The response to the change is the application of (a set of) predefined operators.

In this paper I consider the special scenario where an ontology (called the receiver’s ontology) has to be changed along the arrival of a sequence of triggering bits of ontology fragments coming from another ontology (sender’s ontology). In the terminology of Flouris et al. (Flouris et al. 2008) the one-step change would be termed ontology merge as the purpose is to get a better understanding of the domain from merging two ontologies over the same domain. As in our setting the merge is directed I call the kind of change operation iterated ontology revision.

One instance of iterated ontology revision is given by iterated reinterpretation operators (Eschenbach and Özcep 2010). In these operators, conflicts between the trigger and the receiver’s ontology is explained by ambiguous use of terms. Consider an example of two online library systems with ontologies: the sender may use Article to denote publications in journals whereas the receiver may use Article to denote publications in journals or proceeding volumes. As the ontologies are assumed to be over the same domain, the receiver guesses relations on relations between her and the sender’s uses and stipulates them as bridging axioms, e.g., stating that all articles in the sender’s sense are articles in the receiver’s sense.

Now, a challenging aspect is to define adequateness criteria that iterated ontology revision operators should fulfill. I consider the classical iteration postulates of Darwiche and Pearl (Darwiche and Pearl 1994) as possible candidates and state whether they are fulfilled by the reinterpretation operators. Moreover I discuss, for each of them, whether it should be fulfilled at all. The main result of the paper is that reinterpretation operators fulfill the postulates sequences with atomic triggers. For sequences of complex triggers, a fulfillment is not guaranteed and indeed there are different reasons—corresponding to different postulates—why they should not be fulfilled in the particular scenario of ontology revision with well-developed ontologies.

The rest of the paper is structured as follows. After some logical preliminaries and general terminology (Sect. 2) the necessary definitions for reinterpretation operators are recapitulated (Sect. 3). Before the sections on related work and the conclusion, the adapted postulates for iterated revision, results on their fulfillment by reinterpretation operators, and a discussion of the results are given in Sect. 4.

2 Terminology and Logical Preliminaries

The reinterpretation framework described in the following works for any FOL theory but we consider here finite knowledge bases formulated in description logics.

A non-logical DL vocabulary consists of concept symbols (= atomic concepts) \(N_C\), role symbols \(N_R\), and individual constants \(N_I\). Using these, more complex concept descriptions can be built up in a recursive fashion. The set of possible concept constructors depends on the specific DL. We consider in particular the basic constructors \(\Pi, \cup, \neg, \exists, C(V)\) the set of all possible concept descriptions that can be built from the symbols in \(V\) in the given description logic.

The semantics is the usual Tarskian semantics based on interpretations \(\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})\) with a domain \(\Delta^\mathcal{I}\) and denotation function \(\cdot^\mathcal{I}\) which gives for every \(c \in N_I\) an ele-
ment $C^I \subseteq \Delta^I$ for every atomic concept $A \in N_C$ a set $A^I \subseteq \Delta^I$ and for every role symbol $R \in N_R$ a binary relation $R^I \subseteq \Delta^I \times \Delta^I$. The denotation function is extended recursively to all concept descriptions in the usual manner. For the ones we use here, we have $(C \cap D)^I = C^I \cap D^I$, $(C U D)^I = C^I U D^I$, $(\neg C)^I = \Delta^I \setminus C^I$, $(\exists R.C)^I = \{ x \in \Delta^I | \exists y \in \Delta^I . (x, y) \in R^I \}$. For concept descriptions one can build axioms which can be evaluated as true (in satisfied by) or false in (satisfied by) an interpretation. We consider TBox (terminological Box) of the form

- $C \subseteq D$ (concept subsumption) for $C, D \in \mathcal{C}(\mathcal{V})$ with semantics $I \models C \subseteq D$ iff $C^I \subseteq D^I$;
- $R_1 \subseteq R_2$ (role subsumption) for $R_1, R_2 \in N_R$ with semantics $I \models R_1 \subseteq R_2$ iff $(R_1)^I \subseteq (R_2)^I$.

Moreover, we consider ABox axioms (assertional axioms) of the form $C(a)$ and $R(a, b)$ for $C \in \mathcal{C}(\mathcal{V})$, $a, b \in N_I$ and $R \in N_R$. The semantics is $I \models C(a)$ iff $a^I \in C^I$ and $I \models R(a, b)$ iff $(a, b)^I \in R^I$. We call ABox axioms of the form $C(a)$ and $\neg A(a)$ with $A \in N_C$ concept assertions or concept-based literals. A stands for $A$ or $\neg A$. Additionally, equalities $a \equiv b, a, b \in N_I$ may be allowed.

Consistency (= satisfiability) of a set of axioms $X$ means that there is an interpretation $I$ making all axioms in $X$ true, for short $I \models X$. Entailment is defined as usual by $O_1 \models O_2$ iff for all $I$, if $I \models O_1$, then $I \models O_2$. A consequence operator $Cn$ gives the set of all axioms following from a set: $Cn(X) = \{ ax | X \models ax \}$. If necessary, one can specify the vocabulary of the axioms: $Cn^\mathcal{V}(X) = \{ ax \in X | \mathcal{V} \models ax \}$. By $\mathcal{V}(O)$ we denote all non-logical symbols occurring in the set of axioms $O$. $C(O) = C(\mathcal{V}(O))$.

The ontology notion of this paper slightly extends the one known from the semantic web and DL community—the extension relying on the distinction between an internal vocabulary $\mathcal{V}$ and a public vocabulary $\mathcal{V}'$.

**Definition 1** An ontology $O$ is a triple $O = (\mathcal{O}, \mathcal{V}, \mathcal{V}')$ consisting of a set of axioms $\mathcal{O}$ over a logic with non-logical symbols $\mathcal{V}$ (the public vocabulary) and $\mathcal{V}'$ (the internal vocabulary).

In the following I will abuse terminology by calling also the set of axioms $O$ ontology.

Let $O_1 \cap O_2$ be the dual remainder sets modulo $O_2$ (Delgrande 2008). This is the set of inclusion maximal subsets $X$ of $O_1$ that are consistent with $O_2$, i.e., $X \subseteq O_1 \cap O_2$ iff $X \subseteq O_1$, $X \cup O_2$ is consistent and for all $Y \subseteq O_1$ with $X \subseteq Y$ the set $Y \cup O_2$ is not consistent.

We are going to deal with substitutions as means to realize name space dissociations. The set of ambiguity compliant resolution substitutions, denoted $AR(\mathcal{V}, \mathcal{V}')$, consists of substitutions of symbols in $\mathcal{V}$ by symbols in $\mathcal{V} \cup \mathcal{V}'$. Here, we assume $\mathcal{V} \cap \mathcal{V}' = \emptyset$ where $\mathcal{V}'$ is the set of symbols used for internalization. The substitutions in $AR(\mathcal{V}, \mathcal{V}')$ get as input a non-logical symbol in $\mathcal{V}$ (a constant, an atomic concept or role in DL speak) and map it either to itself or to a new non-logical symbol (of the same type) in $\mathcal{V}'$. The set of symbols $s \in \mathcal{V}$ for which $s(\sigma) \neq s$ is called the support of $\sigma$ and is denoted $sp(\sigma)$. In the following I use postfix notation for substitutions, i.e., $X\sigma = \sigma(X)$. Moreover, I use the following shorthands $sp_1(\sigma) = sp(\sigma) \cap N_I$ and $sp_{C,R}(\sigma) = sp(\sigma) \cap (N_C \cup N_R)$. A substitution with support $S$ is also denoted by $\sigma_S$. For substitutions $\sigma_1, \sigma_2 \in AR(\mathcal{V}, \mathcal{V}')$ we define an ordering by: $\sigma_1 \leq \sigma_2$ iff $sp(\sigma_1) \subseteq sp(\sigma_2)$. $AR(\mathcal{V}, \mathcal{V}')$ can be partitioned into equivalence classes of substitutions that have the same support. We assume that for every equivalence class a representative substitution $\Phi(S) \in AR(\mathcal{V}, \mathcal{V}')$ with support $S$ is fixed. $\Phi$ is called a disambiguation schema.

### 3 Reinterpretation Operators

This section recapitulates the definitions of ontology revision operators called reinterpretation operators (Eschenbach and Özcep 2010; Özcep 2008). The envisioned scenario is that of two agents holding well-developed ontologies, one called receiver’s ontology, the other called sender’s ontology. The ontologies are over the same domain and the receiver gets bits of information from the sender’s ontology that she wants to integrate into her ontology in order to get a better, more fine-grained model of the domain. A challenging aspect is to preserve the consistency of the ontology. The kind of inconsistency that is considered here is that of inter-ontological ambiguity: the sender and the receiver may use the same symbol with different meanings (compare for example the different uses of Article in the example below).

So, the conflict resolution strategy that is exploited by the reinterpretation operators is based on disambiguating symbols. The sender or the receiver has to reinterpret an ambiguous symbol. In the more interesting non-monotonic setting, that I consider in this paper, it is always the receiver who reinterprets the ambiguous symbol—by storing the old symbol in a new name space and relating her use of the symbols to the sender’s use by bridging axioms. This is in line with classical (prioritized) belief revision where one has full trust in the trigger information. In (Eschenbach and Özcep 2010) these reinterpretation operators are called type-2 operators, contrasting them with type-1 operators in which it is the sender’s terminology that is interpreted.

The weak reinterpretation operators $\otimes$ and strong reinterpretation operators $\boxtimes$ are binary operators with a finite set of ontology axioms ontology as left and right argument. The following example (Özcep 2012) demonstrates the main ideas for the weak reinterpretation operators.

**Example 1** Consider the sets of ontology axioms $O_1, O_2$ of the receiver and sender, resp.:

\[
O_1 = \{ \text{Article}(pr_1), \text{Article}(pr_2), \neg \text{Article}(bo_1) \}, \\
O_2 = \{ \neg \text{Article}(pr_1) \}
\]

Applying the weak reinterpretation operator $\otimes$ gives the following set of axioms:

\[
O_1 \otimes O_2 = \{ \text{Article}'(pr_1), \text{Article}'(pr_2), \neg \text{Article}'(bo_1), \neg \text{Article}(pr_1), \text{Article} \subseteq \text{Article}' \}
\]

For the purpose of the example I assume that only concept and role symbols but not constant symbols may be used ambiguously. So, the above conflict between the sender’s and
receiver’s ontology can only be caused by different uses of the atomic concept Article. The receiver (holder of \(O_1\)) gives priority to the sender’s use of Article over her use of Article, and hence she adds \(\neg\text{Article}(pr_1)\) into the result \(O_1 \otimes O_2\). The receiver’s use of Article is internalized, i.e., all occurrences of Article in \(O_1\) are substituted by a new symbol Article'. This step of internalization will also be called the step of dissociation or disambiguation. Additionally, the receiver adds hypotheses on the semantical relatedness (bridging axioms) of her and the sender’s use of Article, here Article \(\sqsubset\) Article' which states that Article is a subconcept of Article'.

Technically, the disambiguation is realized by uniform substitutions from \(\text{AR}(V, V')\) (see section on logical preliminaries). For the disambiguation, one has to deal with the a potential multiplicity of conflicts. The minimal conflict symbols sets defined below describe the smallest sets of symbols which have to be disambiguated in order to resolve conflicts.

**Definition 2** For ontologies \(O_1, O_2\) over \(V\) the set of minimal conflicting symbols sets, \(\text{MCS}(O_1, O_2)\), is defined as follows:

\[
\text{MCS}(O_1, O_2) = \{ S \subseteq V | \text{There is a } \sigma_S \in \text{AR}(V, V'), \text{ s.t. } O_1 \sigma_S \cup O_2 \text{ is consistent, and for all } \sigma_s \in \text{AR}(V, V') \text{ with } \sigma_s \prec \sigma_S \text{ O}_1 \sigma_S \cup O_2 \text{ is not consistent.} \}
\]

Following the strategy of AGM partial meet revision (Alchourrón, Gärdenfors, and Makinson 1985), we assume that a selection function \(\gamma_1\) selects candidates from \(\text{MCS}(O_1, O_2)\) to be used for the resolution:

\[
\gamma_1(\text{MCS}(O_1, O_2)) \subseteq \text{MCS}(O_1, O_2).
\]

So the symbol set defined by \(S^\# = \bigcup \gamma_1(\text{MCS}(O_1, O_2))\) is the set of symbols which will be internalized.

To regain as much as possible from the receiver’s ontology in the ontology revision result, the disambiguated symbols of \(S^\#\) are related by bridging axioms. Depending on what kind of bridging axioms are chosen, different revision operators result. In this paper we consider two classes of bridging axioms, the simple bridging axioms and the strong bridging axioms (Özcep 2008). Let \(\sigma = \sigma_S \in \text{AR}(V, V')\) be a substitution with support \(S \subseteq V\). Let \(P\) be a concept or role symbol in \(S\), \(\sigma(P) = P'\).

**Definition 3** Let \(\sigma = \sigma_S \in \text{AR}(V, V')\) for \(S \subseteq V \cap (N_C \cup N_R)\). The set of simple bridging axioms w.r.t. \(\sigma\) is

\[
\mathcal{B}(\sigma) = \{ P \subseteq P', P' \subseteq P | P \in S \}.
\]

The set of strong bridging axioms w.r.t. \(\sigma\) is defined as:

\[
\mathcal{B}(\sigma, O) = \{ s \subseteq C \in \mathcal{C}(O), s \in \text{sp}_{\mathcal{C}(O)}(s), O \models C \subseteq s \} \cup \{ s \subseteq C \in \mathcal{C}(O), s \in \text{sp}_{\mathcal{C}(O)}(s), O \models s \subseteq C \} \cup \{ s = s' \in \text{sp}_{\mathcal{C}(O)}(s) \}
\]

In case of conflict, not all bridging axioms of \(\mathcal{B}(S^\#)\) (resp. \(\mathcal{B}(S^\#)\)) can be added to the integration result (compare Ex. 1). Hence, one searches for subsets that are compatible with the union of the internalized ontology and sender ontology, \(O_1 \sigma \cup O_2\). That means, possible candidate sets of bridging axioms can be described by dual remainder sets (see section on logical preliminaries) as \(\mathcal{B}(\sigma)\),\(\mathcal{B}(\sigma)^\top\) \((O_1 \sigma \cup O_2)\).

Again, as there is no preference for one candidate over the other we assume that a second selection function \(\gamma_2\) is given with \(\gamma_2(\mathcal{B}(\sigma) \cup \mathcal{B}(\sigma)^\top) \subseteq (\mathcal{B}(\sigma) \cup \mathcal{B}(\sigma)^\top)\). The intersections of the selected bridging axioms is the set of bridging axioms added to the integration result. (Compare this with the partial meet revision functions of AGM (Alchourrón, Gärdenfors, and Makinson 1985)).

**Definition 4** Let \(V, V'\) be disjoint vocabularies and \(\Phi\) a disambiguation scheme. Moreover let \(\gamma_1, \gamma_2\) be selection functions and for short let \(\mathcal{T} = (\gamma_1, \gamma_2)\). For any ontology \(O_1\) and \(O_2\) over \(V\) let \(S^\# = \bigcup \gamma_1(\text{MCS}(O_1, O_2))\) and \(\sigma = \Phi(S^\#)\). Then the weak reinterpretation operator \(\otimes\mathcal{T}\) and the strong reinterpretation operator \(\otimes\mathcal{T}\) are defined as follows:

\[
\begin{align*}
O_1 \otimes \mathcal{T} O_2 &= O_1 \sigma \cup O_2 \cup \bigcap \gamma_2(\mathcal{B}(\sigma)(O_1 \sigma \cup O_2)) \\
O_1 \otimes \mathcal{T} O_2 &= O_1 \sigma \cup O_2 \cup \bigcap \gamma_2(\mathcal{B}(\sigma, O_1) \cup (O_1 \sigma \cup O_2))
\end{align*}
\]

The definition for weak reinterpretation operators is the same as in (Özcep 2008), the definition of the strong operators is an extension.

In the following I will simplify the discussion by simplifying the first step of internalization: In the internalization step now all symbols of the receiver are internalized. There is only a selection function for bridging axioms. Due to the fact that the maximal candidates of bridging axioms are used in the reinterpretation operators unnecessary internalizations do not occur. I therefore can write in the following, e.g., \(\otimes\mathcal{T}\) instead of \(\otimes\mathcal{T}\).

A particularly interesting case of ontology change appears in the context of ABox update (Ahmeti, Calvanese, and Polleres 2014; Güttner, Hurtado, and Vaisman 2011), where the trigger informations are assertional axioms (ABox) axioms. I consider the special case that only atomic bits from the sender ontology occur as trigger, namely, the trigger \(O_2\) is of the form \(O_2 = \{A(a)\}\) or of the form \(O_2 = \{\neg A(a)\}\). That is, the trigger is a concept assertion with an atomic symbol \((A \in N_C)\) or the negated atomic symbols \(\neg A\). For this special case particular strong reinterpretation operators can be defined (Eschenbach and Özcep 2010). The first class assumes that within the underlying DL a most specific concept w.r.t. an ontology exists. \(C^\#\) is most specific concept for \(b\) in the ontology \(O\) if \(O \models C(b)\) and for all \(C\) s.t. \(O \models C(b)\) also \(O \models C \subseteq C^\#\). The most specific concept is unique modulo concept equivalence (w.r.t. \(O\), hence it is denoted by \(\text{msc}_{\mathcal{O}}(b)\).

**Definition 5** Let \(\mathcal{O} = \langle O, V, V' \rangle\) be an ontology, \(\Phi\) a disambiguation scheme, \(A \in V \cap N_C\) and \(b \in V \cap N_R\). Assume \(\sigma = \{A/A'\}\) is the substitution fixed by \(\Phi\) and assume that \(\text{msc}_{\mathcal{O}}(b)\) exists. The msc-based strong reinterpretation operators for concept-based literals \(\otimes\mathcal{O}\) are defined as follows:

\[
\begin{align*}
O \otimes \mathcal{O} A(b) &= \sigma(O) \cup \{A(b), A' \subseteq A, A \subseteq A' \cup \text{msc}_{\mathcal{O}}(b)\}
\end{align*}
\]
If $O \cup \{\neg A(b)\} \not\models \bot$, let $O \sqcap \neg A(b) = O \cup \{\neg A(b)\}$. Else:

$O \sqcap \neg A(b) = \sigma(O) \cup \{\neg A(b), A \sqsubseteq A', A' \sqsubseteq A \cup \text{msc}_{O_1}(b)\}$

The following examples illustrates the second case.

**Example 2** Assume that the receiver's ontology from the beginning is extended with two additional facts on the “problematic” entity $pr_1$:

$O^+_1 = O_1 \cup \{\text{publishedIn}(pr_1, \text{proc}1), \text{Proceed}(\text{proc}1)\}$

The most specific concept of $pr_1$ w.r.t. $O^+_1$ is

$\text{msc}_{O^+_1}(pr_1) = \text{Article} \sqcap \exists \text{publishedIn. Proceed}$

Hence the result of strong reinterpretation w.r.t. triggering concept assertions $O^+_1 \sqcap \neg \text{Article}(pr_1)$ adds the following additional bridging axiom

$\text{Article'} \sqsubseteq \text{Article} \sqcap (\text{Article'} \sqsubseteq \exists \text{publishedIn. Proceed})$

This says that the wider use of Article by the receiver adds (only) those publications in proceedings into the extension.

The selection-based strong operators for triggering literals provide more bridging axioms between the internalized and non-internalized symbols.

**Definition 6** Let $O = (O, V, V')$ be an ontology, $\Phi$ a disambiguation scheme, $A \in V \cap N_C$ and $b \in V \cap N_i$. Assume $\sigma = [A/A']$ is the substitution fixed by $\Phi$ and that $\text{msc}_{O}(b)$ exists. Moreover, let $sel$ be an arbitrary selection function, defined as $sel(X) \subseteq X$. The selection-based strong reinterpretation operators for concept-based literals $O_{\alpha}^{sel}$ are defined as follows (using auxiliary definitions for the specific bridging axioms):

\[
oa(O, A(b), K') = \{A \sqsubseteq A' \sqcup C \mid C \in C(V \sqcup V'), \quad O \models C(b) \text{ and } A \not\models V(C)\}
\]

\[
oa(O, \neg A(b), A') = \{A \sqsubseteq A' \sqcup C \mid C \in C(V \sqcup V'), \quad O \models C(b) \text{ and } A \not\models V(C)\}
\]

\[
O_{\alpha}^{seg} = \begin{cases} O \cup \{\alpha\} \text{ if } O \cup \{\alpha\} \not\models \bot \\ O \cup \{\alpha\} \cup \text{sel}(oa(O, A, A')) \text{ else} \end{cases}
\]

Though the complexity of the trigger is low the induced concept lattice for the reinterpretation with $O_{\alpha}^{seg}$ is not trivial as illustrated by Fig. 1. Nonetheless, the figure does not suggest that the computation of the revision outcome is more complex than for other revision operators for DL ontologies: It just illustrates the subsumption connections of the concepts within the resulting ontology: the calculation of the lattice is not part of constructing the revision result.

Without proof I state here some observations on the conservativity of reinterpretation operators. Proofs can be found in (Eschenbach and Ozczep 2010).

**Proposition 1** Let $O = (O, V, V')$ be an ontology, $\Phi$ a disambiguation scheme, $A \in V \cap N_C$ and $a, c \in V \cap N_i$. Assume $\sigma = [A/A']$ is the substitution fixed by $\Phi$ and that $\text{msc}_{O}(a)$ exists. Let $\alpha = A(a) \in C(a)$ or $\alpha = \neg A(a)$ and $\epsilon = \neg A(c)$. Let $\beta$ be an assertion with $V(\beta) \subseteq (V \cup V(O)) \setminus \{A\}$. Let $sel$ be a selection function for bridging axioms and $sel$ a corresponding function selecting corresponding concepts: $sel(oa(O, A(a), A')) = \{C \mid A \sqsubseteq A' \sqcup C \in sel(oa(O, A(a), A'))\}$.

If $O \models \neg \alpha$, then

1. $O \alpha \models \beta$ iff $O \models \beta$
2. $O \alpha \models \epsilon$ iff $O \cup \{a \neq c\} \models \epsilon$
3. $O \alpha \not\models \neg \epsilon$
4. $O \alpha \models \neg \epsilon$ and $O \models \neg \text{msc}_{O}(a)(\epsilon)$
5. $O \alpha^{sel} \models \neg \epsilon$ iff $O \alpha \models \neg \epsilon$ and $O \models \neg \sqcap \text{sel}(oa2(O, A, A'))(A' \sqcup A'(c))$

**4 Postulates for Iterated Reinterpretation**

Many forms of ontology change (Flouris et al. 2008), in particular ontology evolution (Kharlamov, Zheleznyakov, and Calvanese 2013), require the iterated application of a change operator under new sets of informations. In iterated belief revision, this problem is approached systematically by defining, both, postulates and operators for iterated applications of revision operators. The first systematic study of iterated belief revision goes back to the work of Darwiche and Pearl (Darwiche and Pearl 1994) who stressed the fact that the AGM postulates (Alchourrón, Gärdenfors, and Makinson 1985) are silent w.r.t. the iterated application of operators. Indeed, the only postulates that can be said to touch some form of iteration are those dealing with the revision of conjunctions of triggers (supplementary postulates 7 and 8).

I state those postulates in a form adapted to ontologies.

**(RAGM 7)** $Cn^V(O \circ (O_1 \cup O_2)) \subseteq Cn^V((O \circ O_1) \cup O_2)$

Postulate (RAGM 7) says that all sentences over $V$ following from $O \circ (O_1 \cup O_2)$ are contained in the revision by $O_1$ followed by an expansion with $O_2$.

**(RAGM 8)** If $(O \circ O_1) \cup O_2 \not\models \bot$, then:

$Cn^V((O \circ O_1) \cup O_2) \subseteq Cn^V(O \circ (O_1 \cup O_2))$

Postulate (RAGM 8) says that all sentences over $V$ following from the result of revising with $O_1$ and expanding with $O_2$ also follow from revising $O$ with the union of $O_1$ and $O_2$. A precondition is that the revision result by $O_1$ is compatible with $O_2$.

In general, the reinterpretation operators do not fulfill these postulates. This can be shown with examples similar those provided by Delgrande and Schaub (Delgrande and Schaub 2003, p. 13).

But if one chooses particular selection functions for the reinterpretation operators on triggering ontologies, then one can show that the postulates (RAGM 7) and (RAGM 8) are fulfilled. This result is similar to an AGM theorem (Alchourrón, Gärdenfors, and Makinson 1985) which says that a partial meet revision operator on belief sets fulfills all AGM postulates (in particular the supplementary ones) iff it can be defined as a transitive relational partial meet revision operator.
If $O \models \neg A(b)$, we assume that the set of concepts chosen by $\text{sel}$ is representable as a concept description $C$. Here, $A_1 \cap A_2$ is abbreviated as $A_1A_2$, $\neg A$ by $\overline{A}$ and $A_1 \cup A_2$ by $A_1|A_2$.

**Definition 7** A selection function $\gamma$ for bridging axioms is called a maximum based selection function for bridging axioms if the following holds:

1. $|\gamma(X)| = 1$ for all $\emptyset \neq X \subseteq B(\sigma_2)$.
2. If $BA_1$ and $BA_2$ are non-empty sets of bridging axioms from $B(\sigma_2)$, i.e., $BA_1, BA_2 \in \text{Pw}(B(\sigma_2)) \setminus \{\emptyset\}$ s.t. for all $X_2 \in BA_2$ there is a $X_1 \in BA_1$ with $X_2 \subseteq X_1$, and if additionally also $\gamma(BA_1) \subseteq BA_2$ holds, then $\gamma(BA_2) = \gamma(BA_1)$.

Now one can show

**Proposition 2** Let $O = (O, V, V')$, $O_1 = (O_1, V, \emptyset)$ and $O_2 = (O_2, V, \emptyset)$ be ontologies and $\gamma$ be a maximum based selection function for bridging axioms. Then:

If $(O \otimes \gamma O_2) \cup O_2$ is consistent, then $O \otimes \gamma (O_1 \cup O_2) = (O \otimes \gamma O_1) \cup O_2$.

This proposition shows that weak reinterpretation operators with maximum based selection function fulfill (RAGM 8). Moreover, one sees immediately that they fulfill the postulate (RAGM 7) because if $(O \otimes \gamma O_1) \cup O_2$ is inconsistent, then trivially:

$$C_{\emptyset}^V(O \otimes \gamma (O_1 \cup O_2)) \subseteq C_{\emptyset}^V((O \otimes \gamma O_1) \cup O_2)$$

The supplementary postulates do not give constraints for the interesting case where for both triggers genuine revisions have to be applied. This motivated Darwiche and Pearl (Darwiche and Pearl 1994) to define four iteration postulates which, in an adaptation for the ontology revision scenario, are investigated in the following. The results below show that the reinterpretation operators in general do not fulfill the postulates.

The postulates are, along the original ideas of Darwiche and Pearl (Darwiche and Pearl 1994), described for finite sets of sentences (here: ontology axioms), and not for epistemic states as in their follow-up paper (Darwiche and Pearl 1997). Using the terminology of Freund and Lehmann (Freund and Lehmann 2002), the type of iterated revision I consider in this paper is static: There is no (used) encoding of the revision history in an epistemic state nor do I consider the dynamic change of revision operators from step to step. As we consider the justification of the iteration postulates not as a whole but one by one this approach does not stand in contradiction to the insights made in the follow-up paper by Darwiche and Pearl (Darwiche and Pearl 1997).

In the following postulates, $O = (O, V, V')$ is the initial ontology, $O_1 = (O_1, V, \emptyset)$ the first triggering ontology and $O_2 = (O_2, V, \emptyset)$ the second triggering ontology. Note that the trigger ontologies do not contain internal symbols—which fits the idea that only the public parts of the sender ontologies are communicated.

**RDP 1** If $O_2 \models O_1$, then $(O \circ O_1) \circ O_2 \equiv O \circ O_2$.

In natural language: If the axioms of the second trigger ontology are stronger than the ones of the first trigger ontology, then the two-step outcome (relativized to the public vocabulary) is already covered by the revision with the second trigger ontology.

**RDP 2** If $O_1 \cup O_2$ is not consistent, then $(O \circ O_1) \circ O_2 \equiv O \circ O_2$.

In natural language: If the axioms of the first and second trigger ontology are incompatible, then the two-step outcome
In natural language: If the revision by the second trigger ontology entails the first trigger ontology, then the entailment still holds for the revision with the first ontology followed by the second trigger ontology.

(RDP 4) If $O_1 \cup (O \circ O_2)$ is consistent, then so is $O_1 \cup (O \circ O_1) \circ O_2$.

In natural language: If the revision by the second trigger ontology is compatible with the first trigger ontology, then the compatibility still holds for the revision with the first followed by the second trigger ontology.

As the following Proposition 3 shows, the fulfillment of all adapted iteration postulates cannot be guaranteed if the trigger is an ontology. (This is the same as for the operators of Delgrande and Schaub (Delgrande and Schaub 2003).) If the trigger is of atom natural situation is different due to the fact that there is only one symbol to be reinterpreted.

For triggering literals only (RDP 2) is not fulfilled.

Proposition 3 states results for all reinterpretation operators mentioned in this paper: Regarding the weak operators a distinction is made between triggering literals and ontologies. Table 1 summarizes the results. The rows contain the operators: The first three having concept-based literals as triggers, the last two ontologies. The columns except for the last one refer to the iteration postulates. The last column gives a reference to the corresponding result in Proposition 3. Regarding the counterexamples I draw the following distinction—also reflected in the table: The weak counterexamples are those that construct ontologies for any selection function. The strong counterexamples are those that construct ontologies for any selection function.

In the counter examples that were used to prove the negative results all reinterpretation operators reinterpret only atomic concepts and roles but not constants. As long as a conflict resolution by reinterpreting only concepts and roles is possible, the reinterpretation operators can be modeled by a suitable definition of a selection function $\gamma^{CR}$. $\gamma^{CR}$ selects only sets of bridge axioms that contain all identities for all constants which, in the end, means that the constants are not reinterpreted. In this case I call $\gamma^{CR}$ a selection function that prefers the reinterpretation of role and concept symbols.

Proposition 3 Regarding the fulfillment of the adapted iteration postulates of Darwiche and Pearl (Darwiche and Pearl 1997) the following results hold.

1. Reinterpretation operators for concept-based triggers $(\ominus, \ominus^\text{sel})$ fulfill (RDP 1), (RDP 3) and (RDP 4).

There are ontologies $O$, $O_1$, $O_2$ such that $\ominus$, $\ominus^\text{sel}$ and $\ominus^\text{sel}$ for all selection functions do not fulfill (RDP 2). There are ontologies $O$, $O_1$, $O_2$ and a selection function $\text{sel}$ such that $\ominus^\text{sel}$ does not fulfill (RDP 3) and does not fulfill (RDP 4).

2. For weak reinterpretation operators over triggering ontologies $\ominus^\gamma$ the following holds:

For all postulates (RDP $x$), $1 \leq x \leq 3$, there are ontologies $O$, $O_1$, $O_2$ such that for all selection functions $\gamma^{CR}$ that prefer the reinterpretation of role and concept symbols $\ominus^\gamma$ does not fulfill (RDP $x$).

There are ontologies $O$, $O_1$, $O_2$ and a selection function $\gamma^{CR}$ that prefer the reinterpretation of role and concept symbols that $\ominus^\gamma$ does not fulfill (RDP 4).

3. For strong reinterpretation operators over triggering ontologies $\ominus^\gamma$ the following holds:

For all postulates (RDP 1), (RDP 3), (RDP 4) there are ontologies $O$, $O_1$, $O_2$ and selection functions $\gamma$ such that $\ominus^\gamma$ does not fulfill any of them.

There are ontologies $O$, $O_1$, $O_2$ such that for all selection functions $\gamma^{CR}$ that prefer the reinterpretation of concepts and role symbols $\ominus^\gamma$ does not fulfill (RDP 2).

I discuss the outcomes of the proposition for the four postulates one by one starting with (RDP 2) which (in its original form given by Darwiche and Pearl) evoked most of the criticism. Regarding this postulate I follow the argument of Delgrande and Schaub (Delgrande and Schaub 2003) according to which (RDP 2) may make sense only for non-complex triggers. For complex triggers, say in our case: complex ontologies $O_1$ and $O_2$, does not work. Assume $O_1$ is made out of two sub-ontologies $O_{11}$ and $O_{12}$ s.t. only $O_{12}$ is not compatible with $O_2$. All those assertions that follow from $O \circ O_1$ on the basis of $O_{11}$ should be conserved after the revision with $O_2$. But according to (RDP 2) amnesic revision would be allowed if $O_2$ would not entail $O_{11}$: All sentences inferred with $O_{11}$ would be eliminated in favor of the new ontology $O_2$.

Regarding the first iteration postulate, the following simple example by Delgrande and Schaub (Delgrande and Schaub 2003) demonstrates its questionable status. Actually, for the proof of Proposition 3.2 I use adapted variants of this example. Consider the ontologies $O = \{A(a)\}$, $O_1 = \{(A \sqcup B)(a)\}$ und $O_2 = \{A(a)\} \models O_1$. Let $\gamma$ be a selection function that prefers the reinterpretation of concept and role symbols. The second ontology $O_2$ is stronger than the first ontology $O_1$. Revision with $O_2$ leads to an ontology in which $B(a)$ does not hold: $O \ominus O_2 \not\models B(a)$. The revision with the first ontology leads to an ontology in which $B(a)$ holds: $O \ominus O_1 \models \sim \sim A(a), (A \sqcup B)(a) \models B(a)$. The revision by the first and then by the second ontology gives an ontology that still entails $B(a)$:

$$(O \ominus O_1) \ominus O_2 = \{\neg A(a), (A \sqcup B)(a), A'(a'), A \subseteq A', a \equiv a', B \subseteq B', B' \subseteq B \} \models B(a)$$

All preconditions in the antecedent of (RDP 1) are fulfilled but not the succeeds: $(O \ominus O_1) \ominus O_2 \not\subseteq (O \ominus O_2)$. There is no plausible revision operator for this particular ontology setting that would fulfill (RDP 1). Such an operator would have to fulfill $O \circ O_1 \models B(a)$ as $O$ and $O_1$ are compatible. The revision with $O_2$ should not eliminate $B(a)$ as $B(a)$ is not relevant for the conflict: $(O \circ O_1) \circ O_2 \not\models B(a)$. Clearly, one could define syntax-sensitive revision operators on belief bases s.t. $(O \circ O_1) \circ O_2 \not\models B(a)$ so that the fulfillment of (RDP 1) could be achieved also for this ontol-
Table 1: Results of Proposition 3

A + entry means that the postulate is fulfilled for all ontologies. A – entry means that there is an ontology such that the postulate is not fulfilled (only used for triggering literals). An entry of type – ( ∀ sel) resp. – ( ∀ γ CR) means that there are ontologies s.t. for all selection functions sel resp. γ CR the postulate is not fulfilled. An entry of type – (∃γ CR) means that there is a selection function γ CR such that the postulate is not fulfilled.

<table>
<thead>
<tr>
<th>Operator</th>
<th>(RDP 1)</th>
<th>(RDP 2)</th>
<th>(RDP 3)</th>
<th>(RDP 4)</th>
<th>Proposition 3, x</th>
</tr>
</thead>
<tbody>
<tr>
<td>⊗ sel</td>
<td>+</td>
<td>–</td>
<td>+</td>
<td>+</td>
<td>1</td>
</tr>
<tr>
<td>□</td>
<td>+</td>
<td>–</td>
<td>–</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>⊗ γ</td>
<td>– ( ∀ γ CR)</td>
<td>– ( ∀ γ CR)</td>
<td>– ( ∀ γ CR)</td>
<td>– ( ∀ γ CR)</td>
<td>2</td>
</tr>
<tr>
<td>⊡</td>
<td>+</td>
<td>–</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
</tbody>
</table>

An operator ⊗ fulfills the iterated preservation postulate for the left argument (Preservation) has to guarantee that there is a substitution σ s.t. the initial ontology O is preserved in internalized form Oσ in the result of iterated revision with a sequence SEQ.

(Preservation) There is a substitution σ s.t.: Oσ ⊆ O · SEQ

An operator ⊗ that fulfills (Reconstruction) has to guarantee the existence of a substitution ρ such that the initial ontology O and the set set(SEQ) of all triggering ontologies in the sequence SEQ are contained in a renamed variant of the revision result (O · SEQ)ρ.
(Reconstruction) There is a substitution $\rho$ s.t.:

$$O \cup \text{set}(\text{SEQ}) \subseteq (O \circ \text{SEQ})\rho$$

All reinterpretation operators of this paper fulfill both postulates. I state this proposition the proof of which is a slight adaptation of the proof given in (Eschenbach and Özçep 2010) for triggering concept-based literals.

**Proposition 4** Let $O = (O, \mathcal{V}, \mathcal{V}')$ be an ontology and $\text{SEQ}$ be a finite sequence of sets of ontology axioms over $\mathcal{V}$ and $\circ$ a reinterpretation operator for triggering ontologies. Then there exists $\sigma$ and $\rho$, such that:

1. $O \subseteq O \circ \text{SEQ}$
2. $O \cup \text{set}(\text{SEQ}) \subseteq (O \circ \text{SEQ})\rho$
3. For all symbols $C \in \mathcal{V}(O) \cup \mathcal{V}$ one has: $C = C\rho$.

## 5 Related Work

The reinterpretation operators are constructed in a similar fashion as those by Delgrande and Schaub (Delgrande and Schaub 2003) but differ in that they are defined not only for propositional logic but also for DLs (and FOLs). Moreover, I consider different stronger forms of bridging axioms than the implications of (Delgrande and Schaub 2003).

Bridging axioms are special mappings that are used in the reinterpretation operator as auxiliary means to implement ontology revision. One may also consider mappings by themselves as the objects of revision (Qi, Ji, and Haase 2009; Meilicke and Stuckenschmidt 2009). A particularly interesting case of mapping revision comes into play with mappings used in the ontology based data access paradigm (Calvanese et al. 2009). These mappings are meant to lift data from relational DBs to the ontology level thereby mapping between close world of data and the open world of ontologies. In this setting different forms of inconsistencies induced by the mappings can be defined (such as local vs. global inconsistency) and based on this mapping evolution ensuring (one form of consistency) be investigated (Lembo et al. 2015).

In this paper I used reinterpretation operators as change operators on ontologies described in DLs. There are different other approaches that use the ideas of belief revision for different forms of ontology change such as ontology evolution over DL-Lite ontologies (Kharlamov, Zheleznyakov, and Calvanese 2013) or ontology debugging (Ribeiro and Wassermann 2009). As the consequence operator over DLs do not fulfill all preconditions assumed by AGM (Alchourrón, Gärdenfors, and Makinson 1985) one cannot directly transfer AGM constructions and ideas one-to-one to the DL setting as noted, e.g., by Flouris and colleagues (Flouris, Plexousakis, and Antoniou 2005) and dealt in more depth for non-classical logics by Ribeiro (Ribeiro 2012). For the definition of the reinterpretation operators the constraint is not essential. Nonetheless, they lead to constraints in providing appropriate counter examples: namely ontologies expressible in the DL at hand.

## 6 Conclusion

The paper discussed iterative applications of reinterpretation operators meant to handle conflicts due to ambiguous use of symbols in related and well-developed ontologies. Reinterpretation operators may also be used for solving consistencies not due to ambiguity but due to false information—and indeed, the related revision operators in (Delgrande and Schaub 2003) do not talk about ambiguity. But reinterpretation operators cannot be used to solve inconsistencies that clearly cannot be explained by ambiguity: namely consistencies due to different constraints of the sender and the receiver regarding the number of possible objects in the domain (this was discussed under the term reinterpretation compatibility in (Özçep 2008)).

The reinterpretation for triggering literals were shown to fulfill (adapted versions of) classical iteration postulates of Darwiche and Pearl (Darwiche and Pearl 1994) whereas the reinterpretation operators for ontologies were shown in general not to fulfill them. Some of the postulates were criticized for general reasons. Nonetheless, still one may consider other forms of reinterpretation operators that incorporate the reinterpretation history in order to define dynamic operators: For example one might weight symbols according to the number of times they were reinterpreted and then use a comparison of the weights in the next iteration step in order to decided which symbols to reinterpret next.

In addition to the general criticisms I discussed the adequateness of the other postulates in view of the special ontology change scenario. Here one cannot guarantee the fulfillment by reinterpretation operators that implement a uniform reinterpretation. But uniformity is necessary in order to guarantee the fulfillment of postulates that express the preservation and reconstructibility of the ontologies in the revision result.

### Appendix: Proofs

**Proof of Proposition 2**

For the proof we need the following lemma

**Lemma 1** For all $X_2 \subseteq B(\sigma\gamma)^{\top}(O_{\sigma\gamma} \cup O_1 \cup O_2)$ there is a $X_1 \in B(\sigma\gamma)^{\top}(O_{\sigma\gamma} \cup O_1)$ such that $X_2 \subseteq X_1$.

**Proof.** If $O_{\sigma\gamma} \cup O_1 \cup O_2 \cup X_2$ is consistent, so is $O_1 \cup O_2 \cup X_2$. If $X_2$ is not maximal, then there is a superset $X_1 \in B(\sigma\gamma)^{\top}(O_{\sigma\gamma} \cup O_1)$.

Define the following sets

$$X_1 = \gamma(\mathcal{B}(\sigma\gamma)^{\top}(O_{\sigma\gamma} \cup O_1))$$
$$X_2 = \gamma(\mathcal{B}(\sigma\gamma)^{\top}(O_{\sigma\gamma} \cup O_1))$$

With this notation we have $(O \otimes^\gamma O_1) \cup O_2 = O_{\sigma\gamma} \cup O_1 \cup O_2 \cup X_1$ and $(O \otimes^\gamma O_1) \cup O_2 = O_{\sigma\gamma} \cup O_1 \cup O_2 \cup X_2$. Assume that $(O \otimes^\gamma_1 O_1) \cup O_2$ is consistent. Then there is a $X' \in B(\sigma\gamma)^{\top}(O_{\sigma\gamma} \cup O_2)$ with $X_1 \subseteq X'$. Because of Lemma 1 there is a $X'' \in B(\sigma\gamma)^{\top}(O_{\sigma\gamma} \cup O_1)$ with $X' \subseteq X''$. Hence $X_1 \subseteq X''$. But as $X_1$ is inclusion maximal, one gets $X_1 = X'' = X'$ and hence $X_1 \in B(\sigma\gamma)^{\top}(O_{\sigma\gamma} \cup O_1 \cup O_2)$. Due to Lemma 1 the first precondition for maximum based selection functions was shown to hold. Now we showed also that the second condition holds, hence $X_1 \cup X_2$. In particular: $O \otimes^\gamma_1 (O_1 \cup O_2) = (O \otimes^\gamma_1 O_1) \cup O_2$. 
Proof of Proposition 3

All results hold trivially if \( O \) is not consistent. So for the following assume that \( O \) is consistent.

**Proof of 1.** In the following let \( \circ \in \{\Box, \Diamond, \circ_{\text{sel}}\} \).

For proof (RDP 1): As \( O_1 \) and \( O_2 \) are trigger literals, \( O_2 \upharpoonright O_1 \) holds if \( O_1 = O_2 \). Hence \( (O \circ O_1) \circ O_2 = (O \circ O_2) \circ O_2 = O \circ O_2 \), as the reinterpretation operators fulfill the success postulate.

Counterexample for (RDP 2): Let \( O = \{A(b)\} \). \( O_2 = \{A(a)\} \) and \( O_2 = \{\neg A(a)\} \). Then \( O_1 \cup O_2 \not= \bot \). On the other hand \( (O \circ O_1) \circ O_2 = \{A(b), \neg A(a)\} \upharpoonright A(b) \); one could infer with Proposition 1.4 and 1.5 that \( O \circ A(c) \upsilon \neg A(c) \) which would mean that \( O_1 \cup O_2 \) is not consistent—contradicting the consistency of \( O \circ O_1 \) and the fact that \( O \circ O_2 \parallel \). \( O \).

**Proof of 2.** Counterexample for (RDP 1): Consider

\[
O = \{\neg A(a)\}, \\
O_1 = \{(A \cup B)(a)\}, \\
O_2 = \{A(a)\}
\]

Let be \( \gamma \) an arbitrary selection function that prefers the reinterpretation of role and concept symbols. Then we get on the one hand \( O \circ \gamma O_2 = \{\neg A'(a'), A(a), A' \subseteq A, a \equiv a'\} \).

And so \( O \circ \gamma O_2 \not= \neg B(a) \). On the other hand \( O \circ \gamma O_1 = \{\neg A(a), (A \cup B)(a)\} \not= B(a) \). And last

\[
(O \circ \gamma O_1) \circ \gamma O_2 = \{\neg A'(a'), (A' \cup B')(a'), A(a), A' \subseteq A, a \equiv a', B' \subseteq B', B' \subseteq B\}
\]

But then \( (O \circ \gamma O_1) \parallel B(a) \).

Counterexample for (RDP 2): See counterexample for triggering literals.

Counterexample for (RDP 3):

\[
O = \{A(a), \exists R_1.A \subseteq \neg B, \exists R_2.A \subseteq A, \\
R_2(b, e) \}, \\
O_1 = \{\neg A(b)\}, \\
O_2 = \{\neg A(a), B(a), A(c), \exists R_3.A \subseteq A, R_3(c, b)\}
\]

For all selection functions \( \gamma \) that prefer the reinterpretation of concept and role symbols one has:

\[
O \circ \gamma O_1 = O \cup O_1 \\
O \circ \gamma O_2 = O \circ O_2 \cup B(\sigma_v) \setminus \{A' \subseteq A\} \\
\upharpoonright \neg A(b) \\
(O \circ \gamma O_1) \circ \gamma O_2 = (O \circ O_1) \sigma_v \cup O_2 \cup B(\sigma_v) \setminus \{A' \subseteq A, A' \subseteq A'\} \\
\not= \neg A(b)
\]

Note that the \( O, O_1, O_2 \) have simple structures and do not presuppose complex DL constructors.

Counterexample for (RDP 4): Consider:

\[
O = \{B(a), B(b) \vee C(b)\}, \\
O_1 = \{\neg A(a), \neg B(b)\}, \\
O_2 = \{\neg B(a) \cup A(a), \neg B(b), \neg C(b)\}
\]

Choose \( \gamma \) such that the following results hold:

\[
O \circ \gamma O_2 = \{B'(a'), B'(b') \vee C'(b'), \neg B(a) \vee A(a), \neg B(b), \neg C(b), \neg A(a), \neg A(c), \neg A(b'), \neg C(b')\}
\]

Choose \( \gamma \) such that the following results hold:

\[
O \circ \gamma O_2 = \{B'(a'), B'(b') \vee C'(b'), \neg B(a) \vee A(a), \neg B(b), \neg C(b), \neg A(a), \neg A(c), \neg A(b'), \neg C(b')\}
\]

\[
\not= \neg \bigwedge O_1
\]
\[ O \otimes 2 O_1 = \{ B(a), B(b) \lor C(b), \neg A(a), \neg B(b) \} \models C(b) \]

\[ (O \otimes 2 O_1) \otimes 2 O_2 = \{ B'(a'), B'(b') \lor C'(b'), \neg A'(a'), \neg B'(b') \}, \neg B \cup A(a), \neg B, B' \subseteq B', B' \subseteq B, C \subseteq C', A' \subseteq A \} \models \neg \bigwedge \neg O_1 \]

Note that we use here boolean ABoxes. As in the previous counterexample a purely DL counter example with standard ABoxes should be constructible.

**Proof of 3.** Counter examples for (RDP 1), (RDP 3) und (RDP 4): Consider the same set of ontology axioms as in the corresponding counter examples for the weak reinterpretation \( \otimes \). The selection function \( \gamma \) can be chosen such that the same results follow as in the case of \( \otimes \).

Counter example for (RDP 2): See counter example for triggering literal.

**References**


Law Tests for Semantically-Safe Rule Interchange

Adrian Paschke and Tara Athan
Freie Universitaet Berlin

Abstract
In this paper, we extend a highly successful trend in software engineering (SE), namely test-driven development, to define self-validating rule bases to safeguard rule interchange in distributed environments such as the Web. The concept of a law test is introduced as a way of compactly representing a collection of test matching a predefined pattern, or “law”. Sets of law tests, associated with multiple laws, can be used to provide evidence of the semantically correct execution of an interchanged rule-based Logic Program (LP) in a target execution environment, such as an inference service or agent, by running, in the target inference engine, the law tests attached to and interchanged with the LP. Even in cases when the service/agent does not provide explicit information about its supported semantics, e.g. by its interface description, some failed law tests can be used to prove that the intended semantic properties for the interchange rule program are violated by the implemented semantics, while successful law tests can provide evidence that the intended semantics is supported.

1 Introduction
A strong demand for rule-based functionalities comes from the distributed systems and Web community, in particular in the area of Semantic Web (SW). Here ever larger and more complex rule bases are increasingly managed and maintained in a distributed environment and interchanged over domain and system boundaries between platform-specific inference engines using more-or-less standardized rule markup interchange formats such as RuleML. The correct execution of the interchanged rule-based LPs depends on both the intended semantics of the interchanged LP and the actual (implemented) semantics of the inference engine used at the target SW service/agent.

The contribution of this paper is a test-driven approach supporting semantically-safe rule interchange and verification & validation (V&V) of inference engines provided e.g. as open SW inference services or rule-based agents on the Web. Tests, which are typically interchanged together with an LP, can support decisions as to whether that LP has the intended behavior if it is executed in the target environment (Paschke 2006; Paschke et al. 2006; Dietrich and Paschke 2005). Our new contribution are “law tests”, which probe the supported laws of the logic of a knowledge representation (KR) language of interest and their semantic properties (LP semantics and non-monotonic semantics).

We first present some background in LP semantics in section 2. In section 3 we introduce our conceptual solution using law tests for probing the laws of logic of an inference engine. In section 4 we further refine this idea of law tests for deciding if an interchange rule program can be correctly interpreted with the semantics supported by the target inference service/agent. We demonstrate and evaluate the concept by a proof-of-concept implementation in section 5. In section 6 we discuss related work and we conclude the paper in section 7 with future work.

2 Background
It is beyond the scope of this paper to review or compare all the different LP- and NMR (non-monotonic reasoning)-semantics. For an overview we refer to (Apt and Bol 1994; Minker 1993; Dix 1995a). We give some general definitions and an initial taxonomy of semantics and LP classes.

Definitions
For the scope of this paper, we will assume all intended and implemented semantics are based on the 3-valued Herbrand semantics of the connectives \( \land, \lor, \neg \) as well as “weak” (Lukasiewicz) implication \( \vdash \) and equivalence \( \iff \).

- A semantics \( SEM \) of a class of programs \( \mathcal{L} \) (the LP class of \( SEM \)) is a mapping that assigns to each member \( P \) of \( \mathcal{L} \) a value \( SEM_P \) in the powerset of 3-valued Herbrand models on \( \mathcal{L}_P \), the restriction of \( \mathcal{L} \) to the nonlogical constants that appear in \( P \).
- If \( U \) is a set of atoms, then \( SEM_P(U) \) may be written as \( SEM_P(U) \).
- A sceptical entailment relation (\( SEM_{\text{sept}} \) or \( \models_{\text{P}} \)) is defined from sets of atoms to sets of literals, such that \( U \models_{\text{P}} V \) if every \( v \in V \) is a Lukasiewicz consequence of \( M \models_{\text{P}} V \) for every \( M \in SEM_P(U) \). If \( U \) is the empty set, it may be omitted. The entailment relation is extended to single atoms and literals by considering them as singleton sets.
- A semantics \( SEM' \) extends a semantics \( SEM \) (denoted by \( SEM' \geq SEM \)) iff for all \( P \) in \( \mathcal{L}_P \), the LP class
Figure 1: Classes of LPs. Lines between classes represent containment, with the larger (more expressive) class higher in the diagram.

Figure 2: Semantics for LPs and their Extension Relations.

of $SEM$, $SEM'_{scept,p} \supseteq SEM_{scept,p}$. Note that if $SEM'$ extends $SEM$, then the LP class of $SEM' \supseteq \mathcal{L}_p$.

A semantics $SEM'$ conservatively extends a semantics $SEM$ (denoted by $SEM' \succeq SEM$) iff $SEM'$ extends $SEM$ and $SEM'_{scept,p}$ equals $SEM_{scept,p}$ for all $P \in \mathcal{L}_p$, the LP class of $SEM$.

Łukasiewicz tautologies, e.g. $a = p \leftarrow p$, are denoted $|= a$.

Figure 1 shows different classes of LPs partially-ordered by expressiveness, i.e. a more expressive class $Q$ includes a less expressive class $Q$. For example, the class of extended disjunctive LPs is more expressive than the class of normal LPs; it extends the normal LP class with explicit negation and disjunctions.

Figure 2 illustrates the partial orderings $\supseteq$ and $\succeq$, of semantics on LP classes up to general disjunctive LPs, with semantics having larger LP classes higher in the diagram. Lines between the semantics represent conservative extensions ($\succeq$), while non-conservative extensions are indicated with $\supseteq$.

An LP $P$ can be executed by an inference engine if and only if all of the following holds:

- $P$ is in the LP class of the implemented semantics $SEM'$ of the inference engine.
- the literals entailed by $P$ under the implemented semantics of the inference engine that are in the LP class of the intended semantics $SEM$ are entailed under the intended semantics.
- the literals entailed by $P$ under the intended semantics $SEM$ are entailed under the implemented semantics of the inference engine.

If $SEM' \supseteq_c SEM$, then all conditions above hold; thus it is a sufficient (but not necessary) condition for $P$ to be executable by an inference engine that the implemented semantics be a conservative extension of the intended semantics$^2$. This information about the semantics may be given by explicit annotations, e.g. referencing predefined semantic profiles (see e.g. RuleML’s semantic profile mechanism (Paschke 2014)), or it may need to be determined.

### 3 Concept - Law Tests

In the context of rule interchange with open, distributed inference engines, which might be provided as open inference services, an important question is, whether the inference engine correctly implements a semantics. Tests, possibly interchangechanged together with the LP, can be used to support verification and validation of the execution of the interchanged LP in the target environment and therefore establish trust in this service.

Following software engineering standards, e.g. (ISO/IEC/IEEE 29119-1 2013), we make use of the following terminology about testing in general:

- a test case is a performative, typically a query, together with the expected result.
- A test suite, a set of assertions (which may be logic programs or ground facts) together with a set of tests.
- A test is a test case or a test suite.
- A test program is a logic program asserted within a test suite.

At the lowest level of evaluation, the performative of each test case is executed on the appropriate assertions (if any). It succeeds if the actual results agree with the expected results, and fails otherwise, i.e., if there is no termination or the actual results do not agree with the expected results. The typical test protocol for a test suite is that it succeeds if and only if all the tests it contains succeed, but other test protocols are allowed, as we will see below.

For example, consider a test suite asserting a simple test program with the following rules:

#### Example 3.1. Simple LP

```
a <- b
b <- a
```

$^2$However note, that the complexity of $SEM'$ in most cases is higher than in $SEM$ if $SEM'$ is an extension of $SEM$, and hence for performance reasons an inference engine supporting only the intended semantics $SEM$ is typically optimal for programs of the LP class for which $SEM$ is defined.
and containing one test case \( T = \{ c? \Rightarrow true \} \), i.e., the test query performative \( c? \) which as expected result should yield true. In the case that the inference engine supports stable model semantics (STABLE) this test case, and the test suite, succeeds.

However, if the inference engine supports well-founded semantics (WFS) the test query leads to the actual result "unknown" and accordingly the test case, and the test suite, fails.

While the test suite described above directly relates to the V&V of a particular interchanged LP with a particular semantics, it is a cumbersome task to develop such tests. Further, these tests are limited in scope - in a context where one of a limited number of semantics is known to be correctly implemented, one or a few such tests may be sufficient to discriminate between the possible implemented semantics. However, a systematic approach to developing tests based on the properties, or laws, of the intended semantics is needed to support target engine selection in a more general context.

We define a law test to be a test whose failure indicates that a particular law does not hold in the implemented semantics, and whose success provides evidence that the law holds in the implemented semantics. In general, a law test suite is a test suite that is a law test, and a law test case is a law test that is a test case. In this paper, we are concerned only with a particular form of law test suite where the tests it contains are organized into two suites: a pre-test suite and a post-test suite with one test case. For the law test to succeed, either the pre-test suite must fail or the post-test suite must succeed.

Law tests can be used to assess the correctness of the semantics of an inference engine or to automatically determine the semantics of an inference engine, e.g., in cases when the semantics of the receiving inference service/agent is not explicitly stated. Individual law tests are used to probe the laws of an implemented semantics, such as the supported rules of inference, or its structural properties. By applying a number of law tests to a particular inference service/agent, the implemented semantics provided by the engine can be partially characterized by the set of successful and failed law tests.

For example, SLDNF resolution typically suffers from loops, whereas e.g. SLG resolution implements some form of loop checking and loop prevention. A simple law test for (partially) verifying the "law" of loop avoidance might be, e.g., the following test program:

**Example 3.2. Loop LP**

```
query p(X) :- r(f(X)), q(X), r(f(X)).
```

and the test case \( \{ a? \Rightarrow true \} \), which succeeds for an inference engine supporting reflexivity and loop prevention, e.g. WFS, whereas for an inference engine supporting COMP and SLDNF resolution it does not terminate, demonstrating the lack.

Another problem of SLDNF resolution is that it cannot handle free variables in negative subgoals due to the procedural negation-as-finite-failure, e.g.:

**Example 3.3. Free Variables in Negative Subgoals LP**

```
q(b)
q(f(a))
```

The test case \( \{ p(a)? \Rightarrow true \} \) succeeds when the implemented semantics is SLDNF. However, a test case \( \{ p(X)? \Rightarrow true : \{ X=a \} \} \) with the expected variable binding \( X = a \) fails, because the negation-as-failure tree is entered with a free variable, which fails due to the fact that no variable binding for \( X \) is computed, even though \( r(f(X)) \) holds and \( q(a) \) does not. This problem is also known as the floundering problem, so the above test may be considered a law test for the "no-floundering law".

For more unsolvable problems that can be used for testing SLDNF semantics see e.g. (Shepherdson 1991). Other well-known problems and paradoxes from literature, e.g. Yale Shooting Problem, which are solved by a particular semantics or resolution, resp., can also be used as tests.

Tests of this sort can be bundled into larger suites of law tests for assessing the validity of certain inference engine implementations (resolution algorithm), semantics (e.g. by testing the entailments) and logical calculi. These suites of law tests can be either interchanged directly together with a rule program or provided in a public testing repository accessible via a service interface, e.g., provided by standards body such as RuleML.

### 4 Solution - Semantically Safe Rule Interchange with Law Tests

To systematically test if an interchanged rule program (LP) can be correctly interpreted with the supported semantics of an inference service/agent we have proposed above the use of suites of law tests for the intended semantic properties of the interchanged program. In this section, we consider how this approach can be applied to some known laws that effectively discriminate between existing LP semantics. Kraus et al. (Kraus, Lehmann, and Magidor 1990) and Dix (Dix 1995b; 1995c) proposed several weak and structural (strong) properties for arbitrary (non-monotonic) semantics. In the following we will briefly review these properties and show how they can be adapted to law tests.

The following structural properties for a sceptical entailment relation \( \models \) are adapted from similar properties for a classical logic language \( L \) as stated in (Kraus, Lehmann, and Magidor 1990) and Dix (Dix 1995b), where here the conventional proof format is used for meta-level implications, the subscript is dropped on the sceptical entailment relation when not needed, \( a, b \) are atoms, \( f, g \) are literals, and \( C, K \) are sets of literals, \( P, Q \) are programs:

- **Right Weakening:** \( \frac{\models p \lor f, C \models f}{C \models g} \)
- **Reflexivity:** \( C \models C \)
- **And:** \( \frac{C \models J, K}{C \models J \land K} \)
- **Or:** \( a \models P, J, b \models P, J \)
- **Left Logical Equivalence:** \( \frac{\models p \lor f, f \models J}{g \models J} \)
Cautious Monotony: $\vdash_{\neg P} J, \vdash_{\neg P} K \vdash_{\neg P} J K$

Cut: $\vdash_{\neg P} J, \vdash_{\neg P} K \vdash_{\neg P} K$

Cumulativity (Cut and Cautious Monotony) If $\vdash_{\neg P} J$ then $\vdash_{\neg P} J K$ if and only if $\vdash_{\neg P} K$

Rational Monotony: $\vdash_{\neg P} J \vdash_{\neg P} J$

Disjunctive Rationality: $\vdash_{\neg P} J \vdash_{\neg P} J$

Negation Rationality: $\vdash_{\neg P} J \vdash_{\neg P} J$

In addition to these structural/strong properties the following weak properties describing general conditions a reasonable (skeptical) semantics should satisfy are derived from (Dix 1995c) (programs consist of a set of rules of the form $H \leftarrow B$ where $H$ and $B$ are sets of formulas and a set of facts that are formulas):

Elimination of Tautologies: If a tautological rule ($\models H \leftarrow B$) is eliminated from a program $P$, then the resulting program $P'$ is semantically equivalent. I.e., $SEM_{\text{scept}, P} = SEM_{\text{scept}, P'}$ where $P' = P \setminus \{H \leftarrow B\}$ and $H \subseteq B$.

Generalized Principle of Partial Evaluation (GPPE): If a rule $H \leftarrow B$, where $B$ contains a subgoal $b$, is replaced in a program $P'$ by the $n$ rules $H \cup (H' - b) \leftarrow ((B - b) \cup B')$, where $H' \leftarrow B'$ are all rules for which $b \in H'$, then the $SEM_{\text{scept}, P} = SEM_{\text{scept}, P'}$.

Positive/Negative Reduction: If a rule $H \leftarrow B$ is replaced in a program $P'$ by $H \leftarrow B' \setminus \{c\}$ ($c$ is a formula) where $c$ appears in no rule head, or a rule $H \leftarrow B$ is deleted from $P$ where there is a fact $a$ in $P$ such that $\neg a \subseteq B$, then $SEM_{\text{scept}, P} = SEM_{\text{scept}, P'}$.

Elimination of Non-Minimal Rules/Subsumption: If a rule is deleted from a program $P$ when there is another rule that subsumes it, then the resulting program $P'$ is semantically equivalent. I.e., $SEM_{\text{scept}, P} = SEM_{\text{scept}, P'}$ if there are distinct rules $H \leftarrow B$ and $H' \leftarrow B'\subseteq P$ such that $B \subseteq B'$ and $H \subseteq H'$ and $P' = P \setminus \{H \leftarrow B\}$.

Closure: The reduction of a program by its entailments has no new entailments. I.e., $SEM_{\text{scept}, P} = SEM_{\text{scept}, P'}$ where the reduction $P^{M}$ of a program by a consistent set of literals $M$ is defined in (Dix 1995c) as a transformation from $LP$ into $LP_{M}$.

Independence: The truth value of a literal $c$ with respect to a semantics $SEM_{P}$ does not change when $P$ is extended by a program $P'$ in a disjoint language. I.e., $\vdash_{P} c$ iff $\vdash_{P_{0} \cup P_{1}} c$ provided that the languages of $P$ and $P'$ are disjoint and $c$ belongs to the language of $P$.

Relevance: The truth value of a literal $c$ with respect to a semantics $SEM_{P}$, only depends on the subprogram formed from the relevant rules of $P$ (relevant$(P)$) with respect to $c$: $\vdash_{P} c$ iff $\vdash_{\text{relevant}(P,c)} c$.

Table 1 shows for some common semantics the properties that they satisfy.

In the first category of properties for a reasonable semantics we mainly focus on Rational Monotony and Cumulativity, since in the settings of the redefined sceptical consequence relation the properties Right Weakening, reflexivity, Left Logical Equivalence are trivial and always satisfied for the intended sceptical semantics which we consider in this paper. Cumulativity can be tested using tests for Cut and Cautious Monotony, since it is equivalent to their combination. Further, satisfaction of Rational Monotony implies Cautious Monotony, Disjunctive Rationality and Negation Rationality: $\text{Rational} \Rightarrow \text{Disjunctive Rationality} \Rightarrow \text{Negation Rationality}$.

The other properties such as Or are useful in the context of disjunctive LPs, e.g. to distinguish between an exclusive and an inclusive interpretation of $\lor$. Cut is a natural condition for non-monotonic formalisms. To test Cut resp. Cautious Monotony we only need to add those atoms that have to be added in order to satisfy Cut resp. Cautious Monotony such that $SEM_{P} \leq SEM_{P_{0}}(M)$ where $M_{0}$ is as sequence of atoms $a$ added to $P$ with $M_{0} \subseteq \emptyset$. Rational Monotony is in any sceptical semantics a stronger form of Cautious Monotony because $a \models \neg b \Rightarrow \neg a \models \neg b$. Rational Monotony can be inductively proved $SEM_{P_{0}}(M_{0}) \leq SEM_{P_{1}}(M_{0})$ with $M_{0} \subseteq \emptyset$ and a maximum $M_{1}$.

If we can find an extension of atoms to an initial program such that this extension violates the property under test, we have a counter example, and the law test which queries for this atom will have an expected result of false. We reuse this counter example as a law test case to assess whether an arbitrary inference engine, i.e., the semantics implemented by this inference engine, solves this counter example. If the law test is successful the inference engine may satisfy this property. To demonstrate this approach we will now give some examples for testing the properties as defined by (Dix 1995b; 1995c).

Given LPs $P_{0}, P_{1}, \ldots$ and $P'$ as well as multiple test cases, $T_{0}, T_{1}, \ldots$ and $T'$, the law test suite $\{(P_{0}, T_{0}), (P_{1}, T_{1}), \ldots\}$, called the "pre-test suite" and $(P', T')$, called the "post-test", together form a conditional law test. The conditional law test fails if every test case $T_{i}$ in the pre-test suite is passed when applied to the test program $P'$ and the post-test case $T'$ fails when applied to its test program $P'$—otherwise, the conditional law test succeeds.

An engine might fail the pre-test suite due to some $P_{i}$ being outside of its input language, or when the expected results are not obtained because preconditions of the semantic property are not satisfied. In either of these cases, it is not possible to draw a conclusion about the property, whether the post-test fails or not - this is not evidence that the law holds. However, each such success of a law test when the pre-test suite succeeds provides some evidence that the law holds for the implemented semantics, and the accumulation of the evidence from multiple law tests may be used to form a decision about the applicability of the implemented engine.

Example 4.1. STABLE does not satisfy Cautious Monotony

P0: $a \leftarrow \neg b$

P': $a \leftarrow \neg b$

---

3Such laws are still useful in V&V of implemented semantics.
Table 1: Table (General Properties of Semantics)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>COMP</td>
<td>Normal</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COMP₃</td>
<td>Normal</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>WFS</td>
<td>Normal</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>STABLE</td>
<td>Normal</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>WGCWA</td>
<td>Pos. Disj.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CGWA</td>
<td>Strat. Disj.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PERFECT</td>
<td>Strat. Disj.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

\[ T_0 = T' : a \Rightarrow true, b \Rightarrow true, c \Rightarrow true \]

\[ \neg P: d \Leftarrow \neg c \]

\[ T_0 = T' : a \Rightarrow true, c \Rightarrow true \]

\[ O – SEMP \text{ derives } \{\neg p, q\} \text{ from } P \text{ since } \neg a \text{ is not derivable. Hence, } T_0 \text{ succeeds. But } P' := P \cup \{c\} \text{ where } STABLE P \text{ has another stable model } \{\neg a, b, c\}, \text{ i.e. } a \text{ can no longer be derivable (i.e. } T' \text{ fails) and cautious monotony is not satisfied.} \]

**Example 4.2. O-SEM is not rational**

\[ P: a \Leftarrow \neg a \]

\[ P': a \Leftarrow \neg a \]

\[ p \Leftarrow a \]

\[ q \Leftarrow \neg p \]

\[ \neg a \]

\[ T_0 = T' : \neg p \Rightarrow true, q \Rightarrow true \]

\[ O – SEMP \text{ derives } \{\neg p, q\} \text{ from } P \text{ since } \neg a \text{ is not derivable. Hence, } T_0 \text{ succeeds. But } P' := P \cup \{c\} \text{ derives } \{a, p\} \text{ and therefore } \neg p \text{ and } q \text{ are not derivable (} T' \text{ now fails), as Rational Monotony would require.} \]

**Example 4.3. EWFS does not satisfy CUT**

\[ P: d \Leftarrow \neg c \]

\[ P': d \Leftarrow \neg c \]

\[ a \Leftarrow \neg a \]

\[ a \Leftarrow \neg a \]

\[ b \Leftarrow \neg a, a \]

\[ b \Leftarrow \neg a, a \]

\[ c \Leftarrow \neg b \]

\[ c \Leftarrow \neg b \]

\[ \neg b \]

\[ \neg b \]

\[ T_0 = T' : a \Rightarrow true, b \Rightarrow true, \neg c \Rightarrow false, \neg d \Rightarrow false \]

**Example 4.4. STABLE does not satisfy Relevance**

\[ P_0: a \Leftarrow \neg b \]

\[ P': a \Leftarrow \neg b \]

\[ c \Leftarrow \neg c \]

\[ T_0 = T' : a \Rightarrow true \]

The unique stable model of \( P_0 \) is \( \{a\} \). If the rule \( c \Leftarrow \neg c \) is added, \( a \) is no longer derivable because no stable model exists. **Relevance** is violated, because the truth value of \( a \) depends on atoms that are totally unrelated to \( a \).

**Example 4.5. GWFS does not satisfy GPPE**

\[ P_0: d \Leftarrow \neg b \]

\[ P': d \Leftarrow \neg b \]

\[ a \Leftarrow \neg b \]

\[ a \Leftarrow \neg b \]

\[ b \Leftarrow c \]

\[ b \Leftarrow d, a \]

\[ c \Leftarrow d, a \]

\[ c \Leftarrow d, a \]

\[ T_0 = T' : \neg c \Rightarrow true, \neg b \Rightarrow true, d \Rightarrow true, a \Rightarrow true \]

The two-valued Herbrand models of \( P_0 \) are \( \{a, d\} \) and \( \{b\} \). Since \( c \) is in neither minimal model, then \( \neg c \) is derived by negation-as-failure (naf) in GWFS. In the next iteration, the rule with \( c \) as subgoal is removed, so \( \neg b \) is derived by negation-as-failure. Finally, \( \{a, d\} \) are derived based on \( \neg b \). Thus \( P_0 \) entails \( \{a, \neg b, \neg c, d\} \). However, for \( P' \), \( \neg b \) does not follow by naf, since \( c \) does not occur in the body of the rule having \( b \) in the head; thus, \( P' \) entails only \( \{\neg c\} \). Hence although \( P' \) partially evaluates \( P_0 \), they are semantically not equivalent, which violates the principle of **GPPE**.

Such law tests for checking for violations of either strong or weak semantic properties provides us with a tool for supporting the decision of what semantics should be used for a particular application. For example, an application might require small rule bases, e.g. because the rules are interchanged frequently. Obviously, here the principles of Elimination of Tautologies and Elimination of Non-Minimal Rules are important, in order to keep the rule base as small as possible without changing its semantics.

Moreover, by taking both kinds of properties together a semantics might be characterized by these, i.e. via applying above. These properties are defined with respect to a semantic equivalence relation \( SEM_P = SEM_{P'} \) for a particular kind of program transformations from \( P \) to \( P' \). In a similar way as with the first kind of strong properties we can provide sets of ‘pre-tests’ and ‘post-tests’, which together constitute a conditional law test. The post-tests are derived from the initial programs via applying the defined transformations of the respective property/law of concern. Hence, if the pre-test succeeds and the post-test fails it proves that the implemented semantics does not satisfy this property.
the complete law test suite consisting of the initial pre-test programs and the post-tests, we can gather evidence as to which properties are satisfied by an arbitrary semantics and which are not. The initial pre-tests not only check the conditions of the semantic properties, they also help to determine whether this particular law test can be applied at all, e.g. it might already give us a clue about the LP class supported by the inference engine of concern. For example, a test program including disjunctions or explicit negation might not be supported by the inference engine.

Given that the pre-tests succeed, the failure of the post-test provides a counter-example of the particular property of concern. The derived set of satisfied and unsatisfied properties for a particular unknown semantics can then be compared to known results for such properties of different semantics. If the set of satisfied and unsatisfied properties derived by applying the meta test programs in the target inference engine, matches the satisfied and unsatisfied properties of a semantics for a particular LP class, this provides evidence that the inference engine supports/implements this semantics.

The semantic principles described in this section are also very important in the context of applying refactoring to LPs. In general, a refactoring to a rule base should optimize the rule code without changing the semantics of the program. Removing tautologies or non-minimal rules or applying positive/negative reductions are typically applied in rule base refinements using refactoring (Dietrich and Paschke 2005) and the semantic equivalence relation between the original and the refined program defined for these principles is therefore an important prerequisite to safely apply a refactoring of this kind.

5 Proof-of-Concept: Integration of Law Tests into Testing Frameworks and Rule Markup Languages

For our proof-of-concept implementation we use an extended ISO-Prolog-related scripting syntax (called Prova for PPROlog + JaVA). In Prova, variables start with a uppercase letter, e.g. X, Y, Z, a constant/individual with a lowercase letter, e.g. a, b, c and a query is written as a function

\[ \text{a() :- not(b())}. \]

\[ \text{solve(test ./examples/pre-tcl.test)}. \]

\[ \text{a() :- not(b())}. \]

\[ \text{solve(test ./examples/post-tcl.test)}. \]

A law test script consists of a unique test ID denoted by testcase(<ID>), optional input assertions such as input facts and test rules, an optional pre-test defining the test queries and variable bindings testSuccess(<Test Name>,<Optional Message for JUnit>), a post-test rule testFailure(<Test Name>,<Message>) and a runTest rule which is used by the meta program which implements the test axioms. The conditional law tests are interpreted by using the test rules to derive the success or failure of each test case.

Example 5.1. Rulebase with Test Case

\[ \text{a() :- not(b())}. \]

\[ \text{solve(test ./examples/pre-tcl.test)}. \]

\[ \text{c() :- not(c())}. \]

\[ \text{solve(test ./examples/post-tcl.test)}. \]

Test Case: 
\[ \text{tcl := \{a() :- \text{true} \}} \]

\[ \text{testcase(tcl.test), \% id} \]

\[ \text{testSuccess("tcl","succeeded"):-} \]

\[ \text{testcase(tcl.test),a().} \]

\[ \text{testFailure("tcl","Relevance law violated"):-} \]

\[ \text{not(testSuccess("tcl",Message)).} \]

\[ \text{define the active tests - used by meta-program} \]

\[ \text{runTest(*/examples/pre-tcl.test):-} \]

\[ \text{testSuccess("Test 1",Message).} \]

\[ \text{runTest(*/examples/post-tcl.test):-} \]

\[ \text{testFailure("Test 1",Message).} \]

The example shows the pre- and post test programs for testing the Relevance law. The test tcl1 first runs the pre-test deriving a(), and then tests with the post-test can no longer be derived, i.e. Relevance is violated. The second argument in the testSuccess/testFailure heads specifies a success resp. failure message which is used for reporting, e.g. to create a JUnit test report. The runTest rule with the test case ID as argument is evaluated by the meta-program implementing the test functionality. A test case might define several tests (testSuccess/testFailure).

Test cases can be bundled into test suites which are also represented as LPs consisting of a test suite ID denoted by test_suite(< name >). and a list of test cases referenced by their URI test_case(< UR1 >). For automated testing we have implemented support for JUnit which runs the tests, creates a final test report which gives the test coverage and the failure and success messages of the executed tests.

A test coverage measurement has been integrated into the test framework. (Paschke 2006, Paschke et al. 2006). The implemented coverage meta-program implements different functions to compute e.g. the minimalized substitution of two terms, the instances of clauses under the instance ordering, the lgg of two clauses, the subsumption of clauses, the generalized subsumption of two clause sets, the relative generalization and the coverage. The results are used to create an automated test coverage report.

To support platform-independent rule interchange we have integrated test case and updates into the Rule Markup Language (RuleML 1.02) 4. It provides a rich syntax for derivation and reaction rules and supports different LP classes such as datalog, hornlog (with naf), extended (with neg) disjunctive, FOL.

A markup serialization syntax for test performatives/test suites/test cases and knowledge updates has been implemented as an extension to Reaction RuleML 1.02 with the following constructs:

\[ \text{Test := \{vvi | Entails | TestSuite | TestCase\}.} \]

\[ \text{vvi := \{Entails | TestSuite | TestCase\}.} \]

\[ \text{TestSuite := \{testbase | Assert | Consult\}.} \]

\[ \text{TestCase := \{do | Assert | Retract | Update |} \]

\[ \text{Query | Answer | Test | Send | Receive |} \]

\[ \text{Action | Consult\}.} \]

---

4http://ruleml.org/1.02
The vvi (verification, validation, integrity) tests of a TestSuite are a set of entailments Entails, nested TestSuites or single TestCases. A TestSuite consists of one or more test bases testbase with assertions Assert or consulted imports Consult and one or more vvi tests. A single TestCase consist of a double performative, such as a Query and the predefined expectedResult such as an Answer. The Test performative executes the tests it contains.

The optional @style attribute can be used to assign meta-information about the intended testing protocol of a test performative, or the semantics (referenced semantic profile Profile) of a test suite, or a test case, e.g. to select a particular semantics from a target inference engine that implements the rules and explore its execution trace.

6 Related Work

V&V of KB systems and in particular rule-based systems such as LPs with Prolog interpreters have received much attention from the mid ’80s to the early ’90s, see e.g. (Antoniou et al. 1998). Criteria for verification and validation range from e.g. structural checks for relevance, redundancy and reachability to semantics tests for completeness and consistency. For a survey see (Preece 2001). Different approaches and methodologies to V&V of rule-based systems have been proposed in the literature such as model checking, code inspection, operational debugging (Byrd 1980) via instrumenting the rule base and exploring the execution trace, tabular methods (van Melle, Shortliffe, and Buchanan 1984), which pairwise compare the rules of the rule base to detect relationships among premises and conclusions, methods based on formal graph theory (Ramaswamy, Sarkar, and sho Chen 1997) or Petri Nets (He et al. 1999) which translate the rules into graphs or Petri nets, methods based on declarative debugging (Shapiro 1983) which build an abstract model of the LP and navigate through it or methods based on algebraic interpretation (Laita et al. 1999) which transform a KB into an algebraic structure, e.g. a boolean algebra which is then used to verify the KB.

Simple operational debugging approaches which instrument the rules and explore its execution trace place a huge cognitive load on the user, who needs to analyze each step of the conclusion process and needs to understand the structure of the rule system under test. On the other hand, typical heavy-weight V&V methodologies are often not suitable for rule-based systems, because they induce high costs of change and do not facilitate evolutionary modelling of rule bases. Model-checking techniques and methods based e.g. on algebraic-, graph- or Petri-net-based interpretations are computationally very costly, inapplicable for expressive rule languages and presuppose a deep understanding of both domains, i.e. of the testing language / models and of the rule language and the rule inferences.

Tests are particular suitable when rule bases grow larger and more complex and are maintained, possibly distributed and interchanged, by different users. Due to their inherent simplicity, tests better support different roles which are involved during the engineering process and give an expressive but nevertheless easy to use testing language since tests are written in the same language as the tested rule program. Research has also been directed at the test driven refinement of rule bases (Dietrich and Paschke 2005), on the automatic generation of test cases, and test coverage measurement computations for test suites. (Denne 1991; Luo et al. 1992; Paschke 2006)

7 Conclusion and Future Work

In the context of rule interchange and selection of target inference engines we have proposed a testing methodology exploiting law tests to analyze the implementation specifics of an inference engine with respect to general properties of the semantics. The approach assesses the suitability of a target environment for the interchanged LP even if no further meta-information about the execution environment is available. This helps to safeguard rule interchange.

The concepts introduced in the paper have been implemented in an extended Prolog KR (called Prova) by integrating the law-test-driven development into the common testing framework JUnit with support for automated Ant task execution. This adds tool-support for law-test-driven development of logic programs.

Finally, we have introduced a concrete RuleML-based syntax for serialization of law tests enabling standardized platform-independent rule interchange.

We plan to extend our approach in several ways,

• automated law test instance generation from compact law statements,
• applying automated refactoring,
• larger suites of law test to test logic programs and inference engines,
• in order to provide them as open services on the web, a deeper and more fine-grained vocabulary to annotate rule engines and logic programs/law tests.

Our vision is to reach a similar market maturity of test-driven development in the agile development of rule programs and rule-based knowledge engineering, as is available in Extreme Programming for imperative software development.

8 Acknowledgments

This work has been partially supported by the Innofil Transfer project “Corporate Smart Content” funded by
the German Federal Ministry of Education and Research (BMBF) and the BMBF Innovation Initiative for the New German Länder - Entrepreneurial Regions.

References


On Stochastic Belief Revision and Update and their Combination

Gavin Rens
Centre for Artificial Intelligence Research,
University of KwaZulu-Natal, School of Mathematics, Statistics and Computer Science and
CSIR Meraka, South Africa
Email: gavinren@gmail.com

Abstract

I propose a framework for an agent to change its probabilistic beliefs when a new piece of propositional information \( \alpha \) is observed. Traditionally, belief change occurs by either a revision process or by an update process, depending on whether the agent is informed with \( \alpha \) in a static world or, respectively, whether \( \alpha \) is a ‘signal’ from the environment due to an event occurring. Boutilier suggested a unified model of qualitative belief change, which “combines aspects of revision and update, providing a more realistic characterization of belief change.” In this paper, I propose a unified model of quantitative belief change, where an agent’s beliefs are represented as a probability distribution over possible worlds. As does Boutilier, I take a dynamical systems perspective. The proposed approach is evaluated against several rationality postulated, and some properties of the approach are worked out.

Information acquired can be due to evolution of the world or revelation about the world. That is, one may notice via some ‘signal’ generated by the changing environment that the environment has changed, or, one may be informed by an independent agent in a static environment that some ‘fact’ holds.

In the present work, I deal with belief change of agents who handle uncertainty by maintaining a probability distribution over possible situations. The agents in this framework also have models for nondeterministic events, and noisy observations. Noisy observation models can model imperfect sensory equipment for receiving environmental signals, but they can also model untrustworthy informants in a static world.

In this paper, I provide the work of Boutilier (1998) as background, because it has several connections with and was the seed for the present work. However, I do not intend simply to give a probabilistic version of his Generalized Update Semantics. Whereas Boutilier (1998) presents a model for unifying qualitative belief revision and update, I build on his work to present a unified model of belief revision and update in a stochastic (probabilistic) setting. I also take a dynamical systems perspective, like him. Due to my quantitative approach, an agent can maintain a probability distribution over the worlds it believes possible, using an expectation semantics of change. This is in contrast to Boutilier’s “generalized update” approach, which takes a most-plausible event and most-plausible world approach. Finally, my proposal requires a trade-off factor to mix the changes in probability distribution over possible worlds brought about due to the probabilistic belief revision process and, respectively, the probabilistic belief update process. Boutilier’s model has revision and update more tightly coupled. For this reason, his approach is better called “unified” while mine is called “hybrid”.

The belief change community does not study probabilistic belief update; it is studied almost exclusively in frameworks employing Bayesian conditioning – for modeling events and actions in dynamical domains (e.g., DBNs, MDPs, POMDPs) (Koller and Friedman, 2009; Poole and Mackworth, 2010, e.g.). The part of my approach responsible for updating stays within the Bayesian framework, but combines the essential elements of belief update with unobservable events and belief update as partially observable Markov decision process (POMDP) state estimation.

On the other hand, there is plenty of literature on probabilistic belief revision (Voorbraak, 1999; Grove and Halpern, 1998; Kern-Isberner, 2008; Yue and Liu, 2008, e.g.). The subject is both deep and broad. There is no one accepted approach to argue which is the best and not the focus of this paper. I shall choose one reasonable method for probabilistic belief revision suitable to the task at hand.

In the first section, Boutilier’s ‘generalized update’ is reviewed. Then, in the next section, I introduce stochastic update and stochastic revision, culminating in the ‘hybrid stochastic belief change’ (HSBC) approach. The final section presents an example inspired by Boutilier’s article (1998) and analyses the results.

Some proofs of propositions are omitted to save space; they are available on request.

Boutilier’s Generalized Update

I use Boutilier’s notation and descriptions, except that I am more comfortable with \( \alpha \) and \( \beta \) to represent sentences, instead of \( A \) and \( B \). It is assumed that an agent has a deducively closed belief set \( K \), a set of sentences drawn from some logical language reflecting the agent’s beliefs about the current state of the world. For ease of presentation, I assume

Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
a logically finite, classical propositional language, denoted \( L \) (\( L_{\text{CPL}} \) in Boutilier (1998)), and consequence operation \( Cn \). The belief set \( K \) will often be generated by some finite knowledge base \( KB \) (i.e., \( K = Cn(KB) \)). The identically true and false propositions are denoted \( \top \) and \( \bot \), respectively. Given a set of possible worlds \( W \) (or valuations over \( L \)) and \( \alpha \in L \), the set of \( \alpha \)-worlds, that is, the elements of \( W \) satisfying \( \alpha \), is denoted by \( \{ \alpha \} \). The worlds satisfying all sentences in a set \( K \) is denoted \( \{ K \} \).

**Update**

Given a belief set \( K \), an agent will often observe a change in the world \( \alpha \), requiring the agent to change \( K \). This is the update of \( K \) by \( \alpha \), denoted \( K^+ \).

\[ \{ KB \} \quad \text{represents the set of possibilities we are prepared to accept as the actual state of affairs. Since observation } \alpha \text{ is the result of some change in the actual world, we ought to consider, for each possibility } w \in \{ KB \}, \text{ the most plausible way (or ways) in which } w \text{ might have changed in order to make } \alpha \text{ true. That is, we want to consider the most plausible evolution of world } w \text{ into a world satisfying the observation } \alpha. \]

To capture this intuition, Katsuno and Mendelzon (1991) propose a family of preorders \( \{ \leq_w \} \ (w \in W) \), where each \( \leq_w \) is a reflexive, transitive relation over \( W \). We interpret each such relation as follows: if \( u \leq_w v \) then \( u \) is at least as plausible a change relative to \( w \) as it is \( v \); that is, situation \( u \) would more readily evolve into \( u \) than it would into \( v \).

Finally, a faithfulness condition is imposed: for every world \( w \) and the preorder \( \leq_w \) has \( w \) as a minimal element; that is, \( w \leq_v v \) for all \( v \neq w \). Naturally, the most plausible candidate changes in \( w \) that result in \( \alpha \) are those worlds \( v \) satisfying \( \alpha \) that are minimal in the relation \( \leq_w \). The set of such minimal \( \alpha \)-worlds for each relation \( \leq_w \), and each \( w \in \{ KB \} \), intuitively capture the situations we ought to accept as possible when updating \( KB \) with \( \alpha \).” (Boutilier, 1998, p. 9).

In other words,

\[ \{ KB \circ \alpha \} = \bigcup_{w \in \{ KB \}} \{ \text{Min}(\alpha, \leq_w) \}, \]

where \( \text{Min}(\alpha, \leq_w) \) specifies the minimal \( \alpha \)-worlds with respect to the preorder \( \leq_w \). Then \( K^+ = Cn(KB \circ \alpha) \), where \( K \) is the belief set associated with \( KB \).

**Revision**

Given a belief set \( K \), an agent will often obtain information \( \alpha \) in a static world, which must be incorporated into \( K \). This is the revision of \( K \) by \( \alpha \), denoted \( K^\alpha \).

The AGM theory of belief revision (Alchourrón, Gärdenfors, and Makinson, 1985) provides a set of guidelines, in the form of the postulates, governing the process. “Unfortunately, while the postulates constrain possible revisions, they do not dictate the precise beliefs that should be retracted when \( \alpha \) is observed. An alternative model of revision, based on the notion of epistemic entrenchment (Gärdenfors, 1988), has a more constructive nature.” (Boutilier, 1998, p. 6).

“Semantically, an entrenchment relation (hence a revision function) can be modeled using an ordering on possible worlds reflecting their relative plausibility (Grove, 1988; Boutilier, 1994). However, rather than use a qualitative ranking relation, we adopt the presentation of (Spohn, 1988; Goldszmidt and Pearl, 1992) and rank all possible worlds using a \( \kappa \)-ranking. Such a ranking \( \kappa : W \rightarrow N \) assigns to each world a natural number reflecting its plausibility or degree of believability. If \( \kappa(w) < \kappa(v) \) then \( w \) is more plausible than \( v \) or more consistent with the agent’s beliefs. We insist that \( \kappa^{-1}(0) \neq \emptyset \), so that maximally plausible worlds are assigned rank 0. These maximally plausible worlds are exactly those consistent with the agent’s beliefs; that is, the epistemically possible worlds according to \( K \) are those deemed most plausible in \( \kappa \) (see Spohn (1988) for further details).

We sometimes assume \( \kappa \) is a partial function, and loosely write \( \kappa(w) = \infty \) to mean \( \kappa(w) \) is not defined (i.e., \( w \) is not in the domain of \( \kappa \), or \( w \) is impossible)” (Boutilier, 1998, p. 6).

A \( \kappa \)-ranking captures the entrenchment of the agent’s beliefs in its belief set \( K \). This entrenchment determines how \( K \) will be revised when the agent receives new information \( \alpha \) makes an observation \( \alpha \) induces a belief set as follows.

\[ K = \{ \alpha \in L \mid \kappa^{-1}(0) \subseteq \{ \alpha \} \}. \]

Due to the ranking or entrenchment of knowledge provided by \( \kappa, \) \( \kappa \) is considered an epistemic state.

“In other words, the set of most plausible worlds (those such that \( \kappa(w) = 0 \)) determine the agent’s beliefs. The ranking \( \kappa \) also induces a revision function: to revise by \( \alpha \) an agent adopts the most plausible \( \alpha \)-worlds as epistemically possible,” (Boutilier, 1998, p. 6).

Let \( W_i = \{ w \in W \mid \kappa(w) = i \} \). And let \( \text{Min}(\alpha, \kappa) \) be the set \( W_i \) with the least \( i \) such that for all \( w^i \in W_i, w^i \models \alpha \). Then

\[ K^\alpha = \{ \beta \in L \mid \text{Min}(\alpha, \kappa) \subseteq \{ \beta \} \}. \]

In words, the belief set revised by \( \alpha \) contains all those sentences entailed by the set of worlds with the same rank, where that rank is the least such that they are all \( \alpha \)-worlds.

**Generalized Update**

As explained in the introduction, my intention with this paper is not to give a probabilistic version of the Generalized Update approach (Boutilier, 1998). For completeness, however, I sketch the approach here covering the approach in detail would take up unnecessary space without lending much insight into my Hybrid Stochastic Belief Change approach.

Boutilier motivates the need for a generalized update method which includes revision, by claiming that KM update (Katsuno and Mendelzon, 1991) is insufficient. He provides the following example adopted from Moore (1990). Suppose you want to test whether the contents of a beaker are chemically acid or base. If it is acid, a piece of litmus paper will turn red, if base, the paper will turn blue. Suppose that the test has not yet been performed, but you believe that the contents in the beaker are acidic. When the litmus paper is dipped into and pulled out of the beaker, the paper turns blue, indicating a basic compound. “Unfortunately, the KM theory does not allow this to take place. [...] One is forced to accept that, if the contents were acidic (in which case it should turn red), some extraordinary change occurred (the
test failed, the contents of the beaker were switched, etc.). [...]. Of course, the right thing to do is simply admit that the beaker did not, in fact, contain an acid—the agent should revise its beliefs about the contents of the beaker.” (Boutilier, 1998, p. 13).

Boutilier adopts an event-based approach where a set of events $E$ is assumed. These events are allowed to be non-deterministic, and each possible outcome of an event is ranked according to its plausibility via a $\kappa$-ranking. “As in the original event-based semantics, we will assume world has an event ordering associated with it that describes the plausibility of various event occurrences at that world,” (Boutilier, 1998, p. 14).

A generalized update model is then defined as $(W, \kappa, E, \mu)$, where $W$ is a set of worlds (the agent’s epistemic state), $\kappa$ is a ranking over $W$, $E$ is a set of events (mappings over $W$), and $\mu$ is an event ordering (a set of mappings over $E$).

As with KM update, updates usually occur in response to some observation, with the assumption that something occurred to cause this observation. After observing $\alpha$, an agent should adjust its beliefs by considering that only the most plausible transitions leading to $\alpha$ actually occurred. The set of possible $\alpha$-transitions are those transitions leading to $\alpha$-worlds. The most plausible $\alpha$-transitions are those possible $\alpha$-transitions with the minimal $\kappa$-ranking. Given that $\alpha$ has actually been observed, an agent should assume that one of these transitions describes the actual course of events. The worlds judged to be epistemically possible are those that result from the most plausible of these transitions.

Boutilier (1998) has a proposition that states that generalized belief update as described above is equivalent to “first determining the predicted updated ranking $\kappa^\alpha$ followed by standard (AGM) revision by $\alpha$ with respect to $\kappa^\alpha$,” (Boutilier, 1998, p. 16). $\kappa^\alpha$ is determined by taking the worlds in the current possible worlds $||K||$ (induced from $\kappa$) and shifting them to all possible worlds given all possible transitions given all possible events (the actual event is unknown), taking into account the relevant plausibility rankings.

**Stochastic Belief Change**

I now consider agents who deal with uncertainty by maintaining a probability distribution over possible situations (worlds) they could be in. Let a belief state $b$ be defined as the set $\{(w, p) \mid w \in W, p \in [0, 1]\}$, where $\sum_{(w, p) \in B} p = 1$. The probability of being in $w$ is denoted $b(w)$. That is, $b$ is a probability distribution over all the worlds in $W$. In the hybrid stochastic belief change (HSBC) framework, an agent maintains a belief state, which changes as new information is received or observed.

An agent is assumed to have a model of how the world works.

**Definition 1.** The stochastic belief change model $M$ has the form $(W, \varepsilon, T, E, O, os)$, where

- $W$ is a set of possible worlds,
- $\varepsilon$ is a set of events,
- $T : (W \times \varepsilon \times W) \to [0, 1]$ is a transition function such that for every $e \in \varepsilon$ and $w \in W$, $\sum_{w' \in W} T(w, e, w') = 1$ ($T(w, e, w')$ models the probability of a transition to world $w'$, given the occurrence of event $e$ in world $w$),
- $E$ is the event likelihood function $(E(e, w) = P(e \mid w)$, the probability of the occurrence of event $e$ in $w$),
- $O : (L \times W) \to [0, 1]$ is an observation function such that for every world $w$, $\sum_{o \in O} O(\alpha, w) = 1$ ($O(\alpha, w)$ models the probability of observing $\alpha$ in $w$), where $\Omega \subset L$ is the set of possible observations, up to equivalence, and where if $\alpha = \beta$, then $O(\alpha, w) = O(\beta, w)$, for all worlds $w$.
- $os : (\Omega \times W) \to [0, 1]$ ($os(\alpha, w)$ is the agent’s ontic strength for $\alpha$ perceived in $w$).

**Definition 2.** $b(\alpha) := \sum_{w \in W \mid w = \alpha} b(w)$.

Let $b_0 := b \circ \alpha$ so that we can write $b_0(w)$, where $\circ$ is any update or revision operator.

Often, in the exposition of this paper, a world will be referred to by its truth vector. For instance, if the vocabulary is $\{q, r\}$ and $w_3 = \neg q \land r$, then $w_3$ may be referred to as 01.

For parsimony, let $b = \langle p_1, \ldots, p_n \rangle$ be the probabilities that belief state $b$ assigns to $w_1, \ldots, w_n$, where $\langle w_1, w_2, w_3, w_4 \rangle = (11, 10, 01, 00)$, and $\langle w_1, w_2, \ldots, w_8 \rangle = (111, 110, \ldots, 000)$.

**Update**

Transitions associated with the observation of $\alpha$ from a world $w$ in the current belief state $b_{cur}$ to a world $w'$ could be caused by different events. According to Boutilier (1998), update can be defined as

$$b_{new} := \left\{ (w', p') \mid w' \in W, p' = \sum_{w \in W \mid e \in \varepsilon} T(w, e, w')E(e, w)b_{cur}(w) \right\}.$$

Because the actual event is unobservable/hidden, $p'$ is the expected probability of reaching $w'$, given the event probabilities.

In partially observable Markov decision process (POMDP) theory (Aström, 1965; Monahan, 1982; Lovejoy, 1991), events are actions chosen by the agent (and thus observable) and observations are hidden. Then, given current belief state $b_{cur}$, selected action $a$ and observation $o$, the state estimation function is defined by

$$b_{new}^{pomdp} := \left\{ (w', p') \mid w' \in W, p' = \frac{O(o, a, w') \sum_{w'' \in W} T(w, a, w'')b_{cur}(w)}{P(o \mid a, b_{cur})} \right\},$$

where $\Omega$ is a set of observation objects and $O : (\Omega \times A \times W) \to [0, 1]$ is an observation function, such that for every $a$ and $w'$, $\sum_{o \in O} O(o, a, w') = 1$. $O(o, a, w')$ models the probability of perceiving $o$ in arrival world $w'$, given the execution of some action $a \in A$. Note that $P(o \mid a, b_{cur})$ is a normalizing constant.

$\equiv$ denotes logical equivalence.
But what is the probabilistic update, given new information/evidence $\alpha$? I suggest that $\alpha$ is the (overt) ‘signal’ generated by the (covert) event. An important question is, When is $\alpha$ received – in the current/departure world ($w_\text{c}$) or in the new/arrival world ($w_n$)? Although it is not clear to me, in POMDP theory, observations are always assumed to be received in the arrival world – I shall assume the same.

In the present framework, actions are not selected by the agent, but by nature. In other words, actions are considered to be events occurring in the environment, uncontrollable by the agent. Further, at the present stage of research, I shall assume that the agent has a less detailed observation model, that is, an agent only knows $O(\alpha, w_n)$, the probability of perceiving $\alpha$ in arrival world $w_n$ (defined in Def. 1). Hence, I propose to weight $b_{\text{new}}^\epsilon(w')$ by $O(\alpha, w_n)$ when receiving new information $\alpha$ and one knows that one’s belief state should be updated (due to an evolving world). Then we can define

**Definition 3.**

$$b \circ \alpha := \{(w', p') \mid w' \in W, p' = \frac{1}{\gamma} O(\alpha, w') \sum_{w \in W} \sum_{e \in \epsilon} T(w, e, w')E(e, w)b(w)\},$$

where $\gamma$ is a normalizing factor.

As far as I know, no-one has proposed rationality postulates for probabilistic update. The reason is likely due to probabilistic update being defined in terms of standard probability theory. The axioms of probability theory have been argued to be rational for several decades (although it is not without its detractors).

The following basic postulates for my probabilistic belief update are proposed. (Unless stated otherwise, it is assumed that $\alpha$ is logically satisfiable, i.e., $\vdash \neg \alpha$ is false.)

- (P$^3$) $b^\alpha_\circ$ is a belief state iff not $\vdash \neg \alpha$
- (P$^2$) $b^\alpha_\circ(\alpha) = 1$
- (P$^3$) If $\alpha \equiv \beta$, then $b^\alpha_\circ = b^\beta_\circ$

**Proposition 1.** If $b^\alpha_\circ(\alpha) > 0$, it is not necessary that $b^\alpha_\circ(\alpha) = 1$.

**Proof.** Let the vocabulary be $\{q, r\}$. Let $b = (0.4, 0.0, 0.1, 0.5)$. Let there be only one event $e$. Let the transition function be specified as $T(11, e, 11) = 0.5$, $T(10, e, 01) = 1$, $T(01, e, 00) = 1$, $T(00, e, 11) = 1$. Let $E(e, w) = 1$ for all $w \in W$. Let the evidence be $q$. Let $O(q, 11) = 0.2$, $O(q, 10) = 0$, $O(q, 01) = 0$, $O(q, 00) = 0.3$. Then applying operation $\circ$ to $b$ produces $b^q_\circ = (0.82, 0.0, 0.18)$. Hence, $b^q_\circ(\alpha) = 0.82 \neq 1$. $\square$

Although the following proposition is mostly negative, the reader will soon see that constraining the stochastic belief change model to be ‘rational’, the negative postulates become positive.

**Proposition 2.** Postulate (P$^3$) holds, while (P$^1$) and (P$^2$) do not hold.

**Definition 4.** We say event $e$ is event-rational when for all $w \in W$: there exists a $w'$ such that $T(w, e, w') > 0$ iff $E(e, w) > 0$.

**Definition 5.** We say $\alpha$ is an e-signal when for all $w' \in W$: there exists a $w'$ such that $T(w, e, w') > 0$ iff $O(\alpha, w') > 0$.

**Definition 6.** We say a model $M$ is observation-rational iff for all $\alpha$, whenever $\vdash \neg \alpha$, $O(\alpha, w) = 0$ for all $w \in W$.

The proposition below says that if one is rational w.r.t. observations and w.r.t. some event, and $\alpha$ is a signal produced by that event, then updating on $\alpha$ is defined.

**Proposition 3.** If $M$ is observation-rational, there exists an event $e \in \epsilon$ which is event-rational and $\alpha$ is an e-signal, then $b^\alpha_\circ$ is a belief state iff not $\vdash \neg \alpha$ (i.e., then (P$^3$) holds).

(P$^3$) does not hold under the antecedents of Proposition 3. Another definition is required as qualification:

**Definition 7.** We say evidence $\alpha$ is trustworthy iff for all $w \in W$, if $w \not\models \alpha$, then $O(\alpha, w) = 0$.

The proposition below says that if $\alpha$ is trustworthy, one is rational w.r.t. some event, and $\alpha$ is a signal produced by that event, then one should accept $\alpha$ in the updated belief state.

**Proposition 4.** If $\alpha$ is trustworthy, there exists an event $e \in \epsilon$ which is event-rational and $\alpha$ is an e-signal, then $b^\alpha_\circ(\alpha) = 1$ (i.e., then (P$^3$) holds).

**Proof.** Not $\vdash \neg \alpha$ is assumed by default. Recall that $b^\alpha_\circ(\alpha) = \sum_{w \in W} b^\alpha_\circ(w)$. Referring to the (\Rightarrow) part of the proof of Proposition 3, $b^\alpha_\circ(\alpha)$ is a belief state and thus $\sum_{w \in W} b^\alpha_\circ(w) = 1$. Hence, for $b^\alpha_\circ(\alpha)$ to be less than 1, there must exist a $w' \in W$ s.t. $w' \not\models \alpha$ and $b^\alpha_\circ(w') > 0$. But then $O(\alpha, w') > 0$. Therefore, for (P$^3$) not to hold, an agent needs to believe that $O(\alpha, w') > 0$ for some world $w'$ where $w' \not\models \alpha$. But then $\alpha$ cannot be trustworthy (i.e., then (P$^3$) holds).

**Definition 8.** (Gärdenfors, 1988). A probabilistic belief change operation $\circ$ is said to be preservative iff for all belief states $P$ and for all propositions $\alpha$ and $\beta$, if $P(\alpha) > 0$ and $P(\beta) = 1$, then $P(\circ) = 1$.

**Proposition 5.** Operation $\circ$ is not preservative.

**Definition 9.** We say evidence $\alpha$ is $\beta$-trustworthy if for all $w \in W$, if $w \not\models \beta$, then $O(\alpha, w) = 0$.

**Proposition 6.** If $b^\alpha(\beta)$ is a belief state, $b(\beta) = 1$ and $\alpha$ is $\beta$-trustworthy, then $b^\alpha(\beta) = 1$.

**Proof.** $\sum_{w \in W} b^\alpha_\circ(w) = 1$. Hence, for $b^\alpha_\circ(\beta)$ to be less than 1, there must exist a $w^\times \in W$ s.t. $w^\times \not\models \beta$ and $b^\alpha_\circ(w^\times) > 0$. And because $b(\beta) = 1$, $b(w^\times) = 0$. So some probability mass must have been shifted from some $\beta$-world to the non-$\beta$-world $w^\times$.

By definition, $b^\alpha_\circ(w^\times) = \frac{1}{\gamma} O(\alpha, w^\times)$

$\sum_{w \in W} \sum_{e \in \epsilon} T(w, e, w^\times)E(e, w)b(w)$. So for $b^\alpha_\circ(w^\times)$ to be $> 0$, $O(\alpha, w^\times)$ must be $> 0$.

However, because $\alpha$ is $\beta$-trustworthy, $O(\alpha, w^\times) = 0$. Hence, $O(\alpha, w^\times) < 0$ and $b^\alpha_\circ(\beta) < 1$. $\square$

**Proposition 7.** $b^\alpha_\circ \not= (b^\alpha_\circ)^\circ$.
Proof. For instance, consider the example used in the proof of Proposition 1. Let $\alpha$ be $q$ and let $\beta$ be $q \land r$. Note that $\alpha \land \beta$ is then logically equivalent to $q \land r$. Let $O(q \land r, 11) = O(q \land r, 10) = 0.5$ and $O(q \land r, 01) = O(q \land r, 00) = 0$.

We know that $b_q = (0.82, 0, 0, 0.18)$. Then $b_q^{q \land r} = (1, 0, 0, 0)$. On the other hand, $b_q^{q \land r} = (0.875, 0, 0, 0.125)$.

Revison

Using Bayes’ Rule$^2$, $P(w \mid \alpha)$ can be determined:

$$P(w \mid \alpha) = \frac{O(\alpha, w) b(w)}{\sum_{w' \in W} O(\alpha, w') b(w')}.$$  

Note that if $O(\alpha, w) = 0$, then $P(w \mid \alpha) = 0$.

It is not yet universally agreed what revision means in a probabilistic setting. In classical belief change, it is understood that if the new information $\alpha$ is consistent with the agent’s current beliefs $KB$, then revision is equivalent to belief expansion (denoted $+$), where expansion is the logical consequences of $KB \cup \{\alpha\}$. It is mostly agreed upon that Bayesian conditioning corresponds to classical belief expansion. This is evidenced by Bayesian conditioning (BC) being defined only when $b(\alpha) \neq 0$. In other words, one could define revision to be

$$b_{BC} \alpha := \{ (w, p) \mid w \in W, p = P(w \mid \alpha) \},$$

as long as $P(\alpha) \neq 0$.  

To accommodate cases where $b(\alpha) \neq 0$, that is, where $\alpha$ contradicts the agent’s current beliefs and its beliefs need to be revised in the stronger sense, we shall make use of a method known as $\alpha$-worlds. It has also been discussed in the work of, for instance, Gärdenfors (1988); Dubois and Prade (1993); Chhogyal et al. (2014); Rens and Meyer (2015). The following version of imaging must not be regarded as a fundamental part of the larger belief change framework presented here; it should be regarded as a placeholder or suggestion for the ‘revision-module’ of the framework. Informally, Lewis’s original solution for accommodating contradicting evidence $\alpha$ is to move the probability of each world to its closest, $\alpha$-world. Lewis made the strong assumption that every world has a unique closest $\alpha$-world. More general versions of imaging allow worlds to have several, equally proximate, closest worlds.

Gärdenfors (1988) calls one of his generalizations of Lewis’s imaging general imaging. Our method is also a generalization. We thus refer to his as Gärdenfors’s general imaging and to our method as generalized imaging to distinguish them. It should be noted that these imaging methods are general revision methods and can be used in place of Bayesian conditioning for expansion. “Thus imaging is a more general method of describing belief changes than conditionalization.” (Gärdenfors, 1988, p. 112).

Let $Min(\alpha, w, d)$ be the set of $\alpha$-worlds closest to $w$ measured with $d$. Formally,

$$Min(\alpha, w, d) := \{ w' \in ||\alpha|| \mid \forall w'' \in ||\alpha||, d(w', w) \leq d(w'', w) \},$$

where $d(\cdot)$ is some acceptable measure of distance between worlds (e.g., Hamming or Dalal distance). It must also obey the faithfulness condition that for every world $w$, $d(w, w') < d(v, w)$ for all $v \neq w$.

**Example 1.** Let the vocabulary be $\{q, r, s\}$. Let $\alpha$ be $q \land r$ or $q \land \neg r$ and time, it would be irrational for an agent to ignore its observation model when revising its beliefs. Another proposed

**Definition 10.**

$$b \ Gl \alpha := \{ (w, p) \mid w \in W, p = 0 \text{ if } w \notin ||\alpha||,$$

$$\text{else } p = \sum_{w' \in W, w' \in Min(\alpha, w', d)} b(w')/|Min(\alpha, w', d)| \}.$$  

**Example 2.** Continuing on Example 1: Let $b = (0, 0.1, 0, 0.2, 0, 0.3, 0, 0.4)$. Let $\alpha = (q \land r) \lor (q \land \neg r)$ and $s$ is abbreviated as $\alpha$. Then $b_{Gl}(111) = \sum_{w' \in W} b(w')/|Min(\alpha, w', d)| = b(111)/|Min(\alpha, 111, d)| + b(011)/|Min(\alpha, 011, d)| = 0/1 + 0/1 = 0.$

$$b_{Gl}(110) = \sum_{w' \in W} b(w')/|Min(\alpha, w', d)| = b(110)/|Min(\alpha, 110, d)| + b(100)/|Min(\alpha, 100, d)| + b(010)/|Min(\alpha, 010, d)| + b(000)/|Min(\alpha, 000, d)| = 0/1 + 0/2 + 0/3 + 0/4 = 0.7.$$

$$b_{Gl}(101) = \sum_{w' \in W} b(w')/|Min(\alpha, w', d)| = b(101)/|Min(\alpha, 101, d)| + b(100)/|Min(\alpha, 100, d)| + b(001)/|Min(\alpha, 001, d)| + b(000)/|Min(\alpha, 000, d)| = 0/1 + 0/2 + 0/1 + 0/4/2 = 0.3.$$

And $b_{Gl}(00) = b_{Gl}(011) = b_{Gl}(010) = b_{Gl}(001) = b_{Gl}(000) = 0$.

Notice how the probability mass of non-$\alpha$-worlds is shifted to their closest $\alpha$-worlds. If a non-$\alpha$-world $w^\times$ with probability $p$ has $n$ closest $\alpha$-worlds (equally distant), then each of these closest $\alpha$-worlds gets $p/n$ mass from $w^\times$.

Recall that in the proposed framework, agents have access to an observation model (formalized via an observation function $O(\cdot, \cdot)$). Given enough computational power and time, it would be irrational for an agent to ignore its observation model when revising its beliefs. Another proposed
definition for a stochastic belief revision operation based on imaging (denoted OGI) is thus

**Definition 11.**

\[
b_{OGI}(\alpha) = \left\{ (w, p) \mid w \in W, p = \frac{O(\alpha, w) b^{OGI}_{\alpha}(w)}{\sum_{w' \in W} O(\alpha, w') b^{OGI}_{\alpha}(w')} \right\}
\]

where the denominator is a normalizing factor.

\[
b_{OGI}(\alpha) \text{ is not defined as } \\
\left\{ (w, p) \mid w \in W, p = 0 \text{ if } w \not\in \{\alpha\} \right\}
\]

else \( p = \sum_{w' \in W} O(\alpha, w') b(w') / |\text{Min}(\alpha, w', d)| \),

because \( \alpha \) is assumed perceived in the new world \( w \), not the old world \( w' \).

Note that if \( P(\cdot \mid \alpha) \) were used instead of \( O(\alpha, \cdot) \), then OGI would be undefined whenever \( b(\alpha) = 0 \). But this is exactly the problem we want to avoid by using imaging. Another justification to rather use \( O(\alpha, w) \) is that its value is positively correlated with \( P(w \mid \alpha) \): If \( O(\alpha, w) = 0 \), then \( P(w \mid \alpha) = 0 \). If \( P(w \mid \alpha) = 1 \), then \( O(\alpha, w) \) is maximal in \( b \) in the following sense: for all \( w' \in W \), if \( w' \neq w \), then either \( b(w') = 0 \) or \( O(\alpha, w') = 0 \), whereas \( b(w) > 0 \) and \( O(\alpha, w) > 0 \).

Note that the denominator may be zero, making OGI undefined in that case. I shall deal with this issue a little bit later.

**Example 3.** Recall from Example 1 that \( b_{GI}^{\beta} = (0, 0, 0, 0, 0, 0, 0, 0, 0) \) and \( \alpha \) is \((q \land r) \lor (q \land \neg r \land s)\). Let \( O(\alpha, w) = 0.3 \), say; for all \( w \in W \). Then \( b_{GI}^{\beta} = b_{GI}^{\alpha} \). Obviously, if the observation model carries no information with respect to \( \alpha \), then it has no influence on the agent’s revised beliefs.

Now let \( O(\alpha, w) = 0.3 \) if \( w \models r \), else \( O(\alpha, w) = 0.2 \). Then \( b_{GI}^{\beta} = (0.3 \times 0/0.23, 0.2 \times 0.7/0.23, 0.2 \times 0.3 \times 0.3 / 0.23, 0.2 \times 0/0.23, 0.3 \times 0/0.23, 0.2 \times 0/0.23) = (0.61, 0.39, 0, 0, 0, 0, 0, 0). \) If the agent has an observation model telling it that \( \alpha \) is more likely to be perceived in \( r \)-worlds than in \( \neg r \)-worlds, then when it receives \( \alpha \), the agent should be biased to believing that it is actually in an \( r \)-world. However, the agent was certain that it was in a \( \neg r \)-world when its belief state was \( b \). GI thus pushes the agent to favour the \( \alpha \)-worlds being \( r \)-worlds. Hence, in this example there is tension between being in a \( \neg r \)-world (due to previous beliefs) and being in an \( r \)-world (due to the observation model).

**Definition 12.**

\[
b_{BCI} \alpha := \left\{ \begin{array}{ll} b_{BC} \alpha & \text{if } b(\alpha) > 0 \\
\text{b}_{OGI} \alpha & \text{if } b(\alpha) = 0 \end{array} \right.
\]

I denote the expansion of belief state \( b \) on \( \alpha \) as \( b_{\alpha}^{*} \) (resp., probability function \( P \) on \( \alpha \) as \( P_{\alpha}^{*} \)) and delay its definition till later. \( P_{\perp} \) is conventionally defined to be the absurd probability function which is defined to be \( P_{\perp}(\delta) = 1 \) for all \( \delta \in L \).

Gärdenfors (1988) proposed six rationality postulates for probabilistic belief revision. (Unless stated otherwise, it is assumed that \( \alpha \) is logically satisfiable, i.e., \( \vdash \neg \alpha \) is false.)

1. \( P^{*}_{\alpha} \) is a probability function
2. \( P^{*}_{\alpha}(\alpha) = 1 \)
3. If \( \alpha \models \beta \), then \( P^{*}_{\alpha} = P^{*}_{\beta} \)
4. \( P^{*}_{\alpha} \neq P_{\perp} \) iff \( \vdash \neg \alpha \)
5. If \( P(\alpha) > 0 \), then \( P^{*}_{\alpha} = P_{\alpha}^{+} \)
6. If \( P^{*}_{\alpha}(\beta) > 0 \), then \( P^{*}_{\alpha \wedge \beta} = (P^{*}_{\alpha})^{+}_{\beta} \).

Instead of saying that the result of an operation is \( P_{\perp} \), I simply say that the result is undefined. And by noting that the result of an operation is not a belief state if it is undefined, one can merge postulates 1 and 4. The stochastic belief revision postulates in my notation are thus

\( P^{*}_{\alpha}(\beta) > 0 \) if \( \vdash \neg \alpha \)

\( P^{*}_{\alpha}(\alpha) = 1 \)

\( P^{*}_{\alpha}(\alpha \wedge \beta) = (P^{*}_{\alpha})^{+}_{\beta} \).

I now test OGI and BCI against each of the five postulates.

Recall that if the denominator in the definition of OGI is zero, it is undefined. To guarantee that OGI is defined,

\[
\sum_{w' \in W} O(\alpha, w') b^{OGI}_{\alpha}(w')
\]

must be non-zero, that is, there must be at least one \( w' \in W \) for which \( O(\alpha, w') b^{OGI}_{\alpha}(w') > 0 \). We know that when \( w' \not\models |\alpha|, O(\alpha, w') b^{OGI}_{\alpha}(w') = b^{OGI}_{\alpha}(w') = 0 \).

**Definition 13.** We say \( \alpha \) is weakly observable if there exists \( a \in W \) such that \( w \models \alpha \) and \( O(\alpha, w) > 0 \). We say \( \alpha \) is strongly observable if for all \( w \in W \) for which \( w \models \alpha \), \( O(\alpha, w) > 0 \).

**Proposition 8.** When \( * \) is OGI, postulate \((P^{*})\), in general, does not hold, but does hold if evidence \( \alpha \) is strongly observable.

**Proof:** Firstly, observe that

\[
b(w') = \sum_{w' \in W} O(\alpha, w', d) b(w') / |\text{Min}(\alpha, w', d)|.
\]

Therefore,

\[
1 = \sum_{w' \in W} b(w')
\]

\[
= \sum_{w' \in W} \sum_{w \in \text{Min}(\alpha, w', d)} b(w') / |\text{Min}(\alpha, w', d)|
\]

\[
= \sum_{w' \in W} \sum_{w \in \text{Min}(\alpha, w', d)} b(w') / |\text{Min}(\alpha, w', d)|
\]

\[
= \sum_{w \in W} \sum_{w' \in W, w \in \text{Min}(\alpha, w', d)} b(w') / |\text{Min}(\alpha, w', d)|
\]

\[
= \sum_{w \in W} b(w') / |\text{Min}(\alpha, w', d)|
\]

Let \( b = (0, 0.1, 0.2, 0, 0.3, 0.4) \) and \( \alpha \) be \((q \land r) \lor (q \land \neg r \land s)\). Let \( O(\alpha, 111) = 0.9 \) and \( O(\alpha, w) = 0 \) for all \( w \in W \), \( w \neq 111 \). (Notice that \( \alpha \) is weakly observable.) From Example 2, we know that \( b^{OGI}_{\alpha}(111) = 0 \), implying that \( b^{OGI}_{\alpha}(111) = 0 \), and one can deduce that \( b^{OGI}_{\alpha}(w) = 0 \) for all \( w \in W \), due to the specification of the observation model.
Now, let \( \alpha \) be strongly observable: let \( O(\alpha, 111) = O(\alpha, 110) = O(\alpha, 101) = 0 \). Then \( \beta_{OGI} = 0 \). Also, \( O(\alpha, w) > 0 \) for all \( w \in W \) for which \( w \models \alpha \). By definition of \( \alpha \), the probability mass of all non-\( \alpha \)-worlds is shifted to their closest \( \alpha \)-worlds; the total mass (of the \( \alpha \)-worlds) thus remains 1. Hence, \( b_{OGI}(\alpha) = 1 \) and there exists a \( w' \models \alpha \) s.t. \( b_{OGI}(w') > 0 \). Now, by definition of strong observability, \( O(\alpha, w') > 0 \). Therefore, \( O(\alpha, w')b_{OGI}(w') > 0 \). And due to the normalizing effect of the denominator in the definition of \( \alpha \), \( b_{OGI} \) is a belief state.

**Proposition 9.** When \( * \) is OGL, postulate \( (P^*2) \), in general, does not hold and does hold when \( \alpha \) is strongly observable.

**Proof.** This result follows directly from an understanding of the proof of Proposition 8.

**Proposition 10.** When \( * \) is OGL, postulate \( (P^*3) \) holds.

**Proposition 11.** Let \( * \) be OGI. If \( + \) is OGL, postulate \( (P^+4) \) holds, otherwise it does not.

Assuming \( (P^+4) \) holds, I consider whether \( (P^+5) \) holds only for two combinations of instantiations of \( * \) and \( + \).

**Proposition 12.** When \( * \) is OGI and \( + \) is OGI, postulate \( (P^+5) \) does not hold.

**Proof.** An instance is provided where \( b_{OGI}^*(\beta) > 0 \) and \( b_{OGI(G)}^*(\beta) \neq (b_{OGI}^*)^\beta \).

Continuing with Example 3, where \( b = (0, 0, 1, 0, 0, 2, 0, 0, 3, 0, 0, 4, 0, 0) \), \( \alpha \) is \((q \land r) \lor (q \land \neg r \land s) \) and \( b_{OGI}^* = b_{GI}^* = (0, 0, 7, 0, 3, 0, 0, 0, 0, 0) \). Let \( \beta \) be \((q \land r) \lor (q \land \neg r \land s) \) and \( b_{OGI(G)}^* = b_{GI(G)}^* = (0, 1, 0, 0, 0, 0, 0, 0, 0) \). But \( b_{OGI(G)}^*(\beta) = 0 \). The latter is due to conditioning on \( \alpha \), \( \beta \), and \( \alpha \) being non-\( \alpha \)-worlds.

**Proposition 13.** When \( * \) is BCI, postulate \( (P^*1) \) holds.

**Proof.** It is known that Bayesian conditioning results in a belief state when the conditional is non-contradictory.

**Proposition 14.** When \( * \) is BCI, postulate \( (P^*2) \) holds.

**Proof.** By definition of BCI, all non-\( \alpha \)-worlds get zero probability and the probabilities of the remaining \( \alpha \)-worlds are magnified to sum to 1.

**Proposition 15.** When \( * \) is BCI, postulate \( (P^*3) \) holds.

**Proposition 16.** Let \( * \) be BCI. If \( + \) is BC or BCI, postulate \( (P^+4) \) holds, otherwise it does not.

For the proof of the next proposition, a lemma is required.

**Lemma 1.** Let \( b(\alpha) > 0 \). If \( b_{BC}^*(\beta) > 0 \), then \( b(\alpha \land \beta) > 0 \).

**Proof.** Assume \( b_{BC}^*(\beta) > 0 \). Then there exists a \( w^\beta \in W \) s.t. \( w^\beta \models \beta \) and \( b_{BC}^*(w^\beta) > 0 \). By definition, \( b_{BC}^*(w) = b(\alpha, w) \rightarrow b_{BC}^*(w^\beta) > 0 \). Hence, \( b(\alpha \land \beta) > 0 \) and \( w^\beta \models \alpha \). But if \( w^\beta \models \alpha \), then \( w^\beta \models \alpha \land \beta \), and due to \( b(\alpha) > 0 \), \( b(\alpha \land \beta) > 0 \).

**Proposition 17.** When \( * \) is BCI and \( + \) is BC, postulate \( (P^+5) \), in general, does not hold, but does hold when \( b(\alpha) > 0 \).

Recall that a probabilistic belief change operation \( * \) is preservative iff for all belief states \( \beta \) and for all propositions \( \alpha \) and \( \beta \), if \( b(\alpha) > 0 \) and \( b(\beta) = 1 \), then \( b(\alpha \land \beta) = 1 \).

**Proposition 18.** Operation OGI is not preservative, while BCI is preservative.

**Proof.** OGI: Let the vocabulary be \( \{ q, r \} \) and \( b = (0, 0, 5, 0, 0) \). Let \( \alpha \) be \( q \land \beta \) and \( \beta \) be \( q \land \neg r \). Then \( b(\alpha) > 0 \), \( b(\beta) = 1 \) and \( b_{GI}^*(\beta) = (0, 0, 5, 0, 0, 0) \). Let \( O(\alpha, w) = 1 \) for all \( w \in W \). Then \( b_{GI}^*(\alpha) = (0, 0, 5, 0, 0, 0) \) and \( b_{GI(G)}^*(\beta) = 0.5 \). BCI: \( b_{BC}^*(\beta) = b_{BC}^*(\beta) \). By assuming that \( b(\alpha) > 0 \) and \( b(\beta) = 1 \), one is implicitly assuming that if \( w \models \alpha \) s.t. \( b(\alpha) > 0 \), then \( w \models \beta \). This in turn implies that whenever \( b_{BC}^*(w) > 0 \), that \( w \models \beta \). The latter is due to conditioning on \( \alpha \) and \( \beta \) are subsets of \( W \) where \( b_{BC}^*(w) > 0 \) for all \( w \in W \); if \( b_{BC}^*(w) > 0 \), then \( w \models \beta \), it must then be the case that \( b_{BC}^*(\beta) = 1 \).

**Ontic and Epistemic Strength**

Suppose there is a range of degrees for information being ontic (the effect of a physical action or occurrence) or epistemic (purely informative). I shall assume that the higher the information’s degree of being ontic, the lower the epistemic status of that information. An agent has a certain sense of the degree to which a piece of received information is due to a physical action or event in the world. This sense may come about due to a combination of sensor readings and reasoning. If the agent performs an action and a change in the local environment matches the expected effect of the action, it can be quite certain that the effect is ontic information. If the agent receives the information from another agent (e.g., radio, through reading, a person speaking directly to the agent), then it should be clear to the agent that the information is epistemic and thus has a low degree of being ontic. If the agent’s sensors show activity, but the agent knows that it did not precisely perform an action with an effect matching its sensor readings, and if the readings do not reveal an epistemic source for the information, then the agent will have to infer from the present world conditions and the information received, or access learnt knowledge matching the present world conditions and the information received, the degree to which the information should be regarded as ontic. For instance, a person might stop talking just after you ask him/her to be quiet. Under particular conditions the person may stop talking due to your request and in other conditions he/she may have stopped talking anyway. Depending on the present world conditions, you might assign a higher (but not definitely certainty) or lower (but not definitely zero) degree of likelihood that the information (i.e., that the person stopped talking) is ontic. Or suppose you have been wearing dark glasses for one hour. You put them on due to the sky being clear and (too) bright. When you take your glasses off, it is not as bright as you thought it would be. So, has the ambient

\[ d \] Here, \( d \) is Hamming distance.
brightness decreased due to changes in the weather, or does it only seem darker when you remove your glasses, due to some unknown physiological process? In this case, it would be convenient to consider the brightness/darkness information as being equally likely ontic and epistemic.

Recall from Definition 1 that \( \text{os}(\alpha, w) \) indicates an agent’s sense for the ontic strength of \( \alpha \) received in \( w \). We say that \( \text{os}(\alpha, w) \) \( = 1 \) when \( \alpha \) is certainly ontic in \( w \). When \( \alpha \) is certainly epistemic in \( w \), then \( \text{os}(\alpha, w) = 0 \). In fact, let the epistemic strength of \( \alpha \) in \( w \) be defined as \( \text{es}(\alpha, w) := 1 - \text{os}(\alpha, w) \).

**Combining Update and Revision**

I propose a way of trading off the probabilistic update and probabilistic revision defined earlier, using the notion of ontic strength.

The hybrid stochastic change of belief state \( b \) due to new information \( \alpha \) with ontic strength (denoted \( b \prec \alpha \)) is defined as

**Definition 14.**

\[
b \prec \alpha := \left\{(w, p) \mid w \in W, p = \frac{1}{\gamma} \left(1 - \text{os}(\alpha, w)b^*_{\alpha}(w) + \text{os}(\alpha, w)b^*_{b}(w)\right)\right\},
\]

where \( \gamma \) is a normalizing factor so that \( \sum_{w \in W} b^*_{\alpha}(w) = 1 \).

Due to our assumption that \( \alpha \) is observed in the arrival world, not the departure world, \( \text{os}(\cdot) \) is applied to the arrival world.

Considering the rationality postulates presented so far for belief update and revision, one can naturally suggest the following postulates for their combination:

- (\( P^<1 \)) \( b^*_{\alpha} \) is a belief state iff not \( \vdash \neg \alpha \)
- (\( P^{>2} \)) \( b^*_{\alpha}(\alpha) = 1 \)
- (\( P^{>3} \)) If \( \alpha \equiv \beta \), then \( b^*_{\alpha} = b^*_{\beta} \)

**Proposition 19.** Postulate (\( P^{>1} \)) does not hold.

**Proposition 20.** Postulate (\( P^{>2} \)) does not hold.

Proof. (\( P^{>2} \)) does not hold because (\( P^{>2} \)) does not hold.

**Proposition 21.** Postulate (\( P^{>3} \)) holds.

Proof. (\( P^{>3} \)) is holds because (\( P^{>3} \)) and (\( P^{>3} \)) hold.

**Theorem 1.** If the agent model \( M \) is observation-rational, \( \alpha \) is trustworthy and strongly observable, there exists an event \( e \in \mathcal{E} \) which is event-rational and \( \alpha \) is an e-signal, then (i) \( b^*_{\alpha} \) is a belief state iff not \( \vdash \neg \alpha \) (i.e., then (\( P^{>1} \)) is true) and (ii) \( b^*_{\alpha}(\alpha) = 1 \) (i.e., then (\( P^{>2} \)) is true).

Proof. Note that by Propositions 3 and 4, (\( P^{>1} \)) and (\( P^{>2} \)) hold. And recall that (\( P^{>1} \)) and (\( P^{>2} \)) are true when \( \alpha \) is strongly observable (see Props. 8, 9, 13 and 14).

(i) (\( P^{>1} \)) Given the antecedents of this proposition, we know by Proposition 4 that \( b^*_{\alpha} \) is defined iff not \( \vdash \neg \alpha \). And by (\( P^{>4} \)), \( b^*_{\alpha} \) is defined iff not \( \vdash \neg \alpha \).

(\( \Rightarrow \)) Assume \( b^*_{\alpha} \) is defined. So there exists a \( w \in W \) s.t. \( b^*_{\alpha}(w) > 0 \), that is, \( \frac{1}{\gamma} \left(1 - \text{os}(\alpha, w)b^*_{\alpha}(w) + \text{os}(\alpha, w)b^*_{\alpha}(w)\right) > 0 \). Thus, either \( b^*_{\alpha}(w) > 0 \) (while \( 1 - \text{os}(\alpha, w) > 0 \) or \( b^*_{\alpha}(w) > 0 \) (while \( \text{os}(\alpha, w) > 0 \) or both), which implies that \( b^*_{\alpha} \) resp. \( b^*_{\alpha} \) is defined. Therefore, not \( \vdash \neg \alpha \).

(\( \Leftarrow \)) Assume not \( \vdash \neg \alpha \). Then \( b^*_{\alpha} \) and \( b^*_{\alpha} \) are defined. This implied that there exists a \( w \in W \) s.t. either \( b^*_{\alpha}(w) > 0 \) or \( b^*_{\alpha}(w) > 0 \) (or both). Hence, \( b^*_{\alpha}(w) > 0 \) and due to normalization in the definition of \( \prec \), \( b^*_{\alpha} \) is defined.

(ii) (\( P^{>2} \)) \( b^*_{\alpha}(\alpha) = \sum_{w \in W} b^*_{\alpha}(w) = \frac{1}{\gamma} \left(1 - \text{os}(\alpha, w)b^*_{\alpha}(w) + \text{os}(\alpha, w)b^*_{\alpha}(w)\right) \), where \( \gamma = \sum_{w \in W} \left[1 - \text{os}(\alpha, w)b^*_{\alpha}(w) + \text{os}(\alpha, w)b^*_{\alpha}(w)\right] \). By (\( P^{>2} \)) and (\( P^{>2} \)), if \( w \not\equiv \alpha \), then \( b^*_{\alpha}(w) = 0 \) and \( b^*_{\alpha}(w) = 0 \). Hence, \( \gamma = \sum_{w \in W} \left[1 - \text{os}(\alpha, w)b^*_{\alpha}(w) + \text{os}(\alpha, w)b^*_{\alpha}(w)\right] \right) = 1 \).

Although one cannot expect \( \nabla \) to be preservative, due to probabilistic update not being preservative (Prop. 5), one can expect \( \nabla \) to have preservative-like behaviour under particular conditions: Recall that \( \alpha \) is defined to be \( \beta \)-trustworthy if for all \( w \in W \), if \( w \not\equiv \beta \), then \( \text{os}(\alpha, w) = 0 \).

**Proposition 22.** If \( b^*_{\alpha}(\beta) \) is a belief state, \( b^*_{\alpha}(\beta) \) is a belief state, \( b(\beta) = 1 \), \( \alpha \) is \( \beta \)-trustworthy and \( \nabla \) is BCI, then \( b^*_{\alpha}(\beta) = 1 \).

Proof. By Proposition 6, \( b^*_{\alpha}(\beta) = 1 \), when \( \alpha \) is \( \beta \)-trustworthy. By Proposition 18, \( \nabla \) is preservative when defined as BCI. Then, for all \( w \in W \), if \( w \not\equiv \beta \), then \( b^*_{\alpha}(\beta) = b^*_{\alpha}(\beta) = 0 \). Hence, for all \( w \in W \), if \( w \not\equiv \beta \), \( b^*_{\alpha}(w) = 0 \). Therefore, because \( b^*_{\alpha}(\beta) \) is a belief state,

- \( b^*_{\alpha}(\beta) = 1 - b^*_{\alpha}(\beta) \)
- \( 1 - \sum_{w \in W, w \not\equiv \beta} b^*_{\alpha}(w) \)
- \( 1 - 0 \)
- \( 1 \).

**Examples and Analysis**

HSBC is now analyzed via examples. The example domain is adapted from one of the domains in the article of Boutilier (1998) – here though, worlds are associated with probabilities, not plausibility ranks. There are eight possible worlds, depending on whether a book \( B \) is inside the house (if it is not in the house, then it is assumed to be on the patio, adjacent to the lawn), whether the book is dry and whether the lawn-grass \( G \) is dry. There are three events: \text{rain} – it rains, \text{sprink} – the sprinkler is on, and \text{null} – neither of these, the
null event. In Boutilier’s example, events are deterministic; however, events in this paper are modeled to be stochastic, to better illustrate the behaviour of the framework.

To simplify calculations and to aid the reader in understanding the results, in the following examples, the agent will associate equal epistemic/ontic strength to a particular piece of information for all worlds (for example case).

I shall compute the agent’s new belief state for each of os(α, w) ∈ {0, 0.25, 0.5, 0.75, 1} (for all w ∈ W), for the two cases where α is ¬Ω(G) and where α is ¬Ω(G) ∧ Ω(B).

Boutilier models the agent’s current (initial) epistemic state with the most plausible situation (rank 0) being (¬Inside(B), Ω(B), Ω(G)) and the next plausible situation (rank 1) being (Inside(B), Ω(B), Ω(G)). I translate this as the agent having a belief state where b(¬Inside(B), Ω(B), Ω(G)) = 0.67 and b(Inside(B), Ω(B), Ω(G)) = 0.33. Observe that in these examples, revision as OGl is equivalent to revision as BCI, because b(¬Inside(B), ¬Ω(B), ¬Ω(G)) = b(¬Ω(G) ∧ ¬Ω(B)) = 0.

The HSBC model M = ( W, ε, T, E, O, os) is now specified.

Let w1, . . . , w8 refer to worlds

w1: (Inside(B), Ω(B), Ω(G))
w2: (Inside(B), Ω(B), ¬Ω(G))
w3: (¬Inside(B), Ω(B), Ω(G))
w4: (¬Inside(B), Ω(B), ¬Ω(G))
w5: (¬Inside(B), ¬Ω(B), Ω(G))
w6: (¬Inside(B), ¬Ω(B), ¬Ω(G))
w7: (¬Ω(B), ¬Ω(B), Ω(G))
w8: (¬Ω(B), ¬Ω(B), ¬Ω(G))

The events are ε = {rain, sprnk, null}.

The following probabilities are debatable; they should not be taken too seriously but serve to illustrate the framework.

T(w1, null, w1) = 0.75 T(w5, null, w5) = 1
T(w1, null, w2) = 0.1 T(w5, null, w6) = 0
T(w1, null, w3) = 0.1 T(w5, null, w7) = 0
T(w1, null, w4) = 0.05 T(w5, null, w8) = 0

T(w1, rain, w1) = 0 T(w5, rain, w5) = 0
T(w1, rain, w2) = 0.75 T(w5, rain, w6) = 0.05
T(w1, rain, w3) = 0 T(w5, rain, w7) = 0.05
T(w1, rain, w4) = 0.25 T(w5, rain, w8) = 0.9

T(w1, sprnk, w1) = 0 T(w5, sprnk, w5) = 0
T(w1, sprnk, w2) = 0.8 T(w5, sprnk, w6) = 0.8
T(w1, sprnk, w3) = 0.05 T(w5, sprnk, w7) = 0.05
T(w1, sprnk, w4) = 0.2 T(w5, sprnk, w8) = 0.15

E(null, w1) = 0.06 E(null, w5) = 0.15
E(rain, w1) = 0.31 E(rain, w5) = 0.7
E(sprnk, w1) = 0.63 E(sprnk, w5) = 0.15

Recall that the current belief state is b = (0.33, 0.0, 0.67, 0.0, 0.0). The following is a list of resulting belief states b′ = b ∪ ¬Ω(G) for the specified ontic strengths.

<table>
<thead>
<tr>
<th>os(·)</th>
<th>b ∪ ¬Ω(G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>(0.00, 0.33, 0.00, 0.00, 0.00, 0.67, 0.00, 0.00)</td>
</tr>
<tr>
<td>0.25</td>
<td>(0.00, 0.32, 0.00, 0.00, 0.00, 0.53, 0.00, 0.13)</td>
</tr>
<tr>
<td>0.50</td>
<td>(0.00, 0.31, 0.00, 0.04, 0.00, 0.40, 0.00, 0.25)</td>
</tr>
<tr>
<td>0.75</td>
<td>(0.00, 0.30, 0.00, 0.06, 0.00, 0.26, 0.00, 0.38)</td>
</tr>
<tr>
<td>1.00</td>
<td>(0.00, 0.28, 0.00, 0.08, 0.01, 0.12, 0.00, 0.51)</td>
</tr>
</tbody>
</table>

Several behaviours can be noted: When the observation is completely epistemic, the probabilities of the two believed worlds are each shifted to their closest ¬Ω(G)-worlds. The more the agent considers the information to be ontic, the more its beliefs are spread out due to the nondeterminism of the events. Whether the observation is considered ontic or epistemic, the agent has a relatively strong belief (between 28% and 33%) that the book is inside and dry. However, in cases where the book is outside, there is a considerable shift in probability from the book being dry (w5) to it being wet (w8), as the agent moves towards an ontic mindset. One could perhaps argue that in an ontic mindset, the agent has access to event/transition information so as to reason about the causes of the book getting wet: it believes that there is a moderate to high likelihood that the book will get wet if it is on the patio, due to the sprinkler coming on or it starting to rain (explaining the wet-grass evidence).

The following is a list of resulting belief states b′ = b ∪ ¬Ω(G) ∧ Ω(B) for the specified epistemic strengths.

<table>
<thead>
<tr>
<th>os(·)</th>
<th>b ∪ ¬Ω(G) ∧ Ω(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>(0.00, 0.29, 0.00, 0.00, 0.00, 0.71, 0.00, 0.00)</td>
</tr>
<tr>
<td>0.25</td>
<td>(0.00, 0.33, 0.00, 0.00, 0.03, 0.59, 0.00, 0.04)</td>
</tr>
<tr>
<td>0.50</td>
<td>(0.01, 0.37, 0.00, 0.00, 0.07, 0.47, 0.01, 0.07)</td>
</tr>
<tr>
<td>0.75</td>
<td>(0.01, 0.41, 0.00, 0.01, 0.10, 0.35, 0.01, 0.11)</td>
</tr>
<tr>
<td>1.00</td>
<td>(0.02, 0.45, 0.00, 0.01, 0.14, 0.23, 0.01, 0.15)</td>
</tr>
</tbody>
</table>

When the agent considers the observation completely epistemically, its beliefs change very similarly to when it was only told that the grass is wet; the agent already believed that the book was dry. However, the extra information has a significant impact on how the agent’s beliefs change when the observation is considered ontically. The agent now believes much less that the book is outside and wet and the grass is wet, and with 78% (as opposed to 40% with the first observation) that the book is dry and the grass is wet (independent of where the book is located). The reason is that when the received information includes a dry book, transitions are focused on going to dry-book worlds.

---

5I shall assume that the null event may include some unknown events (with unknown effects).
Conclusion

In this paper I suggested a method to arrive at a new (probabilistic) belief state when the agent has mixed feelings about whether to revise or update its beliefs, given a new piece of information. Much attention was given to the design and analysis of the separate update and revision operations. The postulates and finally Theorem 1 add weight to my argument that the hybrid stochastic belief change (HSBC) operation is rational when the agent has a rational frame of mind.

Looking at the examples above, the way in which probabilities shift among the possible worlds, given the different ontic/epistemic strengths, seems justifiable. However, more analysis is required here, especially when considering more complicated specification patterns of the ontic/epistemic strengths.

Determining $as(\alpha, w)$ for every foreseen $\alpha$ in every possible world $w$ will be challenging for a designer. Some deep questions are: Should the designer/agent provide the strengths (via stored values or programmed reasoning), or do these strengths come to the agent attached to the new information? What is the reasoning process we go through to determine whether information is epistemic or ontic, if at all? In general, how does an agent know when information is epistemic (requiring revision) or ontic (requiring update)?

References


Revising Incompletely Specified Convex Probabilistic Belief Bases

Gavin Rens
CAIR∗,
University of KwaZulu-Natal,
School of Mathematics, Statistics and Comp. Sci.
CSIR Meraka, South Africa
Email: gavinrens@gmail.com

Thomas Meyer
CAIR,
University of Cape Town,
Dept. of Comp. Sci.
CSIR Meraka, South Africa
Email: tmeyer@cs.uct.ac.za

Giovanni Casini
University of Luxembourg,
Comp. Sci. and Communication Research Unit
Luxembourg
Email: giovanni.casini@uni.lu

Abstract

We propose a method for an agent to revise its incomplete probabilistic beliefs when a new piece of propositional information is observed. In this work, an agent’s beliefs are represented by a set of probabilistic formulae – a belief base. The method involves determining a representative set of ‘boundary’ probability distributions consistent with the current belief base, revising each of these probability distributions and then translating the revised information into a new belief base. We use a version of Lewis Imaging as the revision operation. The correctness of the approach is proved. The expressivity of the belief bases under consideration are rather restricted, but has some applications. We also discuss methods of belief base revision employing the notion of optimum entropy, and point out some of the benefits and difficulties in those methods. Both the boundary distribution method and the optimum entropy method are reasonable, yet yield different results.

Suppose an agent represents its probabilistic knowledge with a set of statements; every statement says something about the probability of some features the agent is aware of. Ideally, the agent would want to have enough information to, at least, identify one probability distribution over all the situations (worlds) it deems possible. However, if the agent could not gather sufficient data or if it was not told or given sufficient information, it would not be able to pinpoint exactly one probability distribution. An agent with this sort of ignorance, can be thought of as having beliefs compatible with a set of distributions. Now, this agent might need to revise its beliefs when new (non-probabilistic) information is received, even though the agent’s beliefs do not characterize a particular probability distribution over its current possible worlds.

Several researchers argue that using a single probability distribution requires the agent to make unrealistically precise uncertainty distinctions (Grove and Halpern, 1998; Voorbraak, 1999; Yue and Liu, 2008). One widely-used approach to dealing with this has been to consider sets of probability measures as a way of modeling uncertainty, (Grove and Halpern, 1998). However, simply applying standard probabilistic conditioning to each of the measures/distributions in the set individually and then combining the resulting sets is also not recommended. The framework presented in this paper proposes two ways to go from one ‘probabilistically incomplete’ belief base to another when new information is acquired.

Both belief revision methods presented, essentially follow this process: From the original belief base, determine a relatively small set of belief states / probability distributions ‘compatible’ with the belief base which is, in a sense, representative of the belief base. (We shall use the terms belief state, probability distribution, probability function and distribution interchangeably). Then revise every belief state in this representative set. Finally, induce a new, revised belief base from the revised representative set.

We shall present two approaches to determine the representative set of belief states from the current belief base: (i) The approach we focus on involves finding belief states which, in a sense, are at the boundaries of the constraints implied by the belief base. These ‘boundary belief states’ can be thought of as drawing the outline of the convex space of beliefs. This outline is then revised to form a new outline shape, which can be translated into a new belief base. (ii) As a possible alternative approach, the representative set is a single belief state which can be imagined to be at the center of the outline of the first approach. This ‘central’ belief state is found by determining the one in the space of beliefs which is least biased or most entropic in terms of information theory (Jaynes, 1978; Cover and Thomas, 1991).

For approach (i) – where the canonical set is the set of boundary belief states – we shall prove that the revised canonical set characterizes the set of all belief states which would have resulted from revising all (including interior) belief states compatible with the original belief base.

The relevant background theory and notations are now introduced.

We shall work with classical propositional logic. Let \( P \) be the finite set of atomic propositional variables (atoms, for short). Formally, a \( \text{world} \) is a unique assignment of truth values to all the atoms in \( P \). There are thus \( 2^n \) conceivable worlds. An agent may consider some non-empty subset \( W \).
of the conceivable worlds called the possible worlds. Often, in the exposition of this paper, a world will be referred to by its truth vector. For instance, if the vocabulary is placed in order \( \{q,r\} \) and \( w_3 \models \neg q \land r \), then \( w_3 \) may be referred to as \( 01^2 \). Let \( L \) be all propositional formulae which can be formed from \( P \) and the logical connectives \( \land \) and \( \neg \), with \( \top \) abbreviating tautology and \( \bot \) abbreviating contradiction.

Let \( \beta \) be a sentence in \( L \). \([\beta]\) denotes the set of \( \beta \)-worlds, that is, the elements of \( W \) satisfying \( \beta \). The worlds satisfying all sentences in a set of sentences \( K \) are denoted by \([K]\).

We define the probabilistic language \( L_{prob} = \{ (\alpha) \bowtie x | \alpha \in L, \bowtie \in \{ \leq, =, \geq \}, x \in [0,1] \}\}. Sentences with strict inequalities \((<,>)\) are excluded from the language for now. Such sentences are more challenging to deal with and their inclusion is left for future work. We propose a belief base (BB) to be a consistent (logically satisfiable) subset of \( L_{prob} \).

A BB specifies an agent’s knowledge.

The basic semantic element of an agent’s beliefs is a probability distribution or a belief state

\[
b = \{(w_1, p_1), (w_2, p_2), \ldots, (w_n, p_n)\},
\]

where \( p_i \) is the probability that \( w_i \) is the actual world in which the agent is. \( \sum_{(w,p) \in B} p = 1 \). We may also use \( c \) to refer to a belief state. For parsimony, let \( b = \langle p_1, \ldots, p_n \rangle \) be the probabilities that belief state \( b \) assigns to \( w_1, \ldots, w_n \) where \( (w_1, w_2, w_3, w_4) = (11, 10, 01, 00) \), and \( (w_1, w_2, \ldots, w_9) = (111, 110, \ldots, 000) \). Let \( \Pi \) be the set of all belief states over \( W \).

\( b(\alpha) \) abbreviates \( \sum_{w \in W, \alpha \bowtie w} b(w) \). \( b \) satisfies formula \( (\alpha) \bowtie x \) (denoted \( b \models (\alpha) \bowtie x \)) iff \( b(\alpha) \bowtie x \). If \( B \) is a set of formulae, then \( b \) satisfies \( B \) (denoted \( b \models B \)) iff \( \forall \gamma \in B, b \models \gamma \). If \( B \) and \( B' \) are sets of formulae, then \( B \) entails \( B' \) (denoted \( B \models B' \)) iff for all \( b \in \Pi \), \( b \models B' \) whenever \( b \models B \). If \( B \models \gamma \) then we simply write \( B \models \gamma \). \( B \) is logically equivalent to \( B' \) (denoted \( B \equiv B' \)) iff \( B \models B' \) and \( B' \models B \).

Instead of an agent’s beliefs being represented by a single belief state, a BB \( B \) represents a set of belief-states: Let \( \Pi^B := \{ b \in \Pi | b \models B \} \). A BB \( B \) is satisfiable (consistent) iff \( \Pi^B \neq \emptyset \).

The technique of Lewis imaging for the revision of belief states, requires a notion of distance between worlds to be defined. We use a pseudo-distance measure between worlds, as defined by Lehmann, Magidor, and Schlecha (2001) and adopted by Chhogyal et al. (2014).

We add a “faithfulness” condition, which we feel is lacking from the definition of Lehman, Magidor, and Schlecha (2001); without this condition, a pseudo-distance measure would allow all worlds to have zero distance between them. Boutilier (1998) mentions this condition, and we use his terminology: “faithfulness”.

**Definition 1.** A pseudo-distance function \( d : W \times W \to \mathbb{Z} \) satisfies the following four conditions: for all worlds \( w, w', w'' \in W \).

1. \( d(w, w') \geq 0 \) (Non-negativity)
2. \( d(w, w) = 0 \) (Identity)
3. \( d(w, w') = d(w', w) \) (Symmetry)
4. \( d(w, w') + d(w', w'') \geq d(w, w'') \) (Triangular Inequality)
5. if \( w \neq w' \), then \( d(w, w') > 0 \) (Faithfulness)

Presently, the foundation theory, or paradigm, for studying belief change operations is commonly known as AGM theory (Alchourrón, Gärdenfors, and Makinson, 1985; Gärdenfors, 1988). Typically, belief change (in a static world) can be categorized as expansion, revision or contraction, and is performed on a belief set, the set of sentences \( K \) closed under logical consequence. Expansion (denoted +) is the logical consequences of \( K \cup \{ \alpha \} \), where \( \alpha \) is new information and \( K \) is the current belief set. Contraction of \( \alpha \) is the removal of some sentences until \( \alpha \) cannot be inferred from \( K \). It is the reduction of beliefs. Revision is when \( \alpha \) is (possibly) inconsistent with \( K \) and \( \alpha \) is (minimally) modified so that the new \( K \) remains consistent and entails \( \alpha \). In this view, when the new information is consistent with the original beliefs, expansion and revision are equivalent.

The next section presents a generalized imaging method for revising probabilistic belief states. Then we describe the application of generalized imaging in our main contribution; revising boundary belief states instead of all belief states. The subsequent section explain another approaches of revising our belief bases, which prepares us for discussions in the rest of the paper. The latter method finds a single representative belief state through maximum entropy inference. Both the boundary belief state method and the maximum entropy method are reasonable, yet yield different results – a seeming paradox is thus uncovered. Then future possible directions of research are discussed. We end with a section on the related work and the concluding section.

**Generalized Imaging**

It is not yet universally agreed what revision means in a probabilistic setting. One school of thought says that probabilistic expansion is equivalent to Bayesian conditioning. This is evidenced by Bayesian conditioning (BC) being defined only when \( b(\alpha) = 0 \), thus making BC expansion equivalent to BC revision. In other words, one could define expansion (restricted revision) to be

\[
b \text{ BC } \alpha = \{ (w, p) | w \in W, p = b(w | \alpha), b(\alpha) = 0 \}\}
\]

To accommodate cases where \( b(\alpha) = 0 \), that is, where \( \alpha \) contradicts the agent’s current beliefs and its beliefs need to be revised in the stronger sense, we shall make use of imaging. Imaging was introduced by Lewis (1976) as a means of revising a probability function. It has also been discussed in the work of, for instance, Gärdenfors (1988); Dubois and Prade (1993); Chhogyal et al. (2014); Rens and Meyer (2015). Informally, Lewis’s original solution for accommodating contradicting evidence \( \alpha \) is to move the probability of each world to its closest, \( \alpha \)-world. Lewis made the strong assumption that every world has a unique closest \( \alpha \)-world. More general versions of imaging allows worlds to have several, equally proximate, closest worlds.
Gärdenfors (1988) calls one of his generalizations of Lewis’s imaging general imaging. Our method is also a generalization. We thus refer to his as Gärdenfors’s general imaging and to our method as generalized imaging to distinguish them. It should be noted that all three these imaging methods are general revision methods and can be used in place of Bayesian conditioning for expansion. “Thus imaging is a more general method of describing belief changes than conditionalization.” (Gärdenfors, 1988, p. 112).

Let $\text{Min}(\alpha, w, d)$ be the set of $\alpha$-worlds closest to $w$ with respect to pseudo-distance $d$. Formally,

$$\text{Min}(\alpha, w, d) := \{w' \in [\alpha] \mid \forall w'' \in [\alpha], d(w', w) \leq d(w'', w)\},$$

where $d(-)$ is some pseudo-distance measure between worlds (e.g., Hamming or Dalal distance).

**Example 1.** Let the vocabulary be $\{q, r, s\}$. Let $\alpha$ be $(q \land \neg r) \lor (q \land \neg r \land s)$. Suppose $d$ is Hamming distance. Then

$$\text{Min}((q \land r) \lor (q \land \neg r \land s), 111, d) = \{111\}$$

$$\text{Min}((q \land r) \lor (q \land \neg r \land s), 110, d) = \{110\}$$

$$\text{Min}((q \land r) \lor (q \land \neg r \land s), 101, d) = \{101\}$$

$$\text{Min}((q \land r) \lor (q \land \neg r \land s), 100, d) = \{101, 110\}$$

$$\text{Min}((q \land r) \lor (q \land \neg r \land s), 011, d) = \{111\}$$

$$\text{Min}((q \land r) \lor (q \land \neg r \land s), 010, d) = \{110\}$$

$$\text{Min}((q \land r) \lor (q \land \neg r \land s), 001, d) = \{101\}$$

$$\text{Min}((q \land r) \lor (q \land \neg r \land s), 000, d) = \{101, 110\}$$

**Definition 2 (Gl).** Then generalized imaging (denoted Gl) is defined as

$$b_{\text{Gl}}(w) := \{w, p\mid w \in W, p = 0 \text{ if } w \not\in [\alpha]\},$$

else $p = \sum_{w' \in \text{Min}(\alpha, w', d)} b(w')/|\text{Min}(\alpha, w', d)|$.

In words, $b_{\text{Gl}}$ is the new belief state produced by taking the generalized imaging of $b$ with respect to $\alpha$. Notice how the probability mass of non-$\alpha$-worlds is shifted to their closest $\alpha$-worlds. If a non-$\alpha$-world $w'$ with probability $p$ has $n$ closest $\alpha$-worlds (equally distant), then each of these closest $\alpha$-worlds gets $p/n$ mass from $w'$.

We define $b_{\text{Gl}}(w) := b \circ \alpha$ so that we can write $b_{\text{Gl}}(w)$, where $\circ$ is a revision operator.

**Example 2.** Continuing on Example 1: Let $b = \{0, 0, 1, 0, 0, 2, 0, 0, 3, 0, 4\}$.

$$(q \land r) \lor (q \land \neg r \land s)$$

is abbreviated as $\alpha$.

$$b_{\text{Gl}}(111) = \sum_{w' \in W, 111 \in \text{Min}(\alpha, w', d)} b(w')/|\text{Min}(\alpha, 111, d)| = b(111)/|\text{Min}(\alpha, 111, d)| + b(011)/|\text{Min}(\alpha, 011, d)| = 0/1 + 0/1 = 0.$$

$$b_{\text{Gl}}(110) = \sum_{w' \in W, 110 \in \text{Min}(\alpha, w', d)} b(w')/|\text{Min}(\alpha, 110, d)| = b(110)/|\text{Min}(\alpha, 110, d)| + b(100)/|\text{Min}(\alpha, 100, d)| + b(010)/|\text{Min}(\alpha, 010, d)| + b(000)/|\text{Min}(\alpha, 000, d)| = 0.1/1 + 0.2/2 + 0.3/1 + 0.4/2 = 0.7.$$

**Revision via Gl and boundary belief states**

Perhaps the most obvious way to revise a given belief base (BB) $B$ is to revise every individual belief state in $\Pi B$ and then induce a new BB from the set of revised belief states. Formally, given observation $\alpha$, first determine a new belief state $b'$ for every $b \in \Pi B$ via the defined revision operation:

$$\Pi B^\alpha = \{b'^\alpha = b_{\text{Gl}}(w) \mid b, w \in \Pi B\}.$$ 

If there is more than only a single belief state in $\Pi B$, then $\Pi B$ contains an infinite number of belief states. Then how can one compute $\Pi B^\alpha$? And how would one subsequently determine $B^\alpha$ from $\Pi B^\alpha$?

In the rest of this section we shall present a finite method of determining $\Pi B^\alpha$. What makes this method possible is the insight that $\Pi B$ can be represented by a finite set of ‘boundary’ belief states – those belief states which, in a sense, represent the limits or the convex hull of $\Pi B$. We shall prove that the set of revised boundary belief states defines $\Pi B^\alpha$.

Inducing $B^\alpha$ from $\Pi B^\alpha$ is then relatively easy, as will be seen.

Let $W_{\text{perm}}$ be every permutation on the ordering of worlds in $W$. For instance, if $W = \{w_1, w_2, w_3, w_4\}$, then $W_{\text{perm}} = \{w_1, w_2, w_3, w_4\}, \{w_1, w_2, w_4, w_3\}, \{w_1, w_3, w_2, w_4\}, \ldots, \{w_4, w_3, w_2, w_1\}$.

Given an ordering $W^\# \in W_{\text{perm}}$, let $W^\#(i)$ be the $i$-th element of $W^\#$; for instance, $\{w_4, w_3, w_2, w_1\}(2) = w_3$. Suppose we are given a BB $B$. We now define a function which, given a permutation of worlds, returns a belief state where worlds earlier in the ordering are assigned maximal probabilities according to the boundary values enforced by $B$.

**Definition 3.** MaxASAP($B, W^\#$) is the $b \in \Pi B$ such that for $i = 1, \ldots, |W|$, $\forall b' \in \Pi B$, if $b' \neq b$, then $\sum_{j=1}^{i} b(W^\#(j)) \geq \sum_{k=1}^{i} b(W^\#(k))$.

**Example 3.** Suppose the vocabulary is $\{q, r\}$ and $B_1 = \{(q) \geq 0.6\}$. Then, for instance, $\text{MaxASAP}(B_1, (01, 00, 11, 10)) = \{(01, 0.4), (00, 0), (11, 0.6), (10, 0)\} = \{(11, 0.6), (10, 0), (01, 0.4), (00, 0)\}.$

**Definition 4.** We define the boundary belief states of BB $B$ as the set

$$\Pi B_{\text{bd}} := \{b \in \Pi B \mid W^\# \in W_{\text{perm}}, b = \text{MaxASAP}(B, W^\#)\}.$$

Note that $|\Pi B_{\text{bd}}| \leq |W_{\text{perm}}|$.
Example 4. Suppose the vocabulary is \{q,r\} and \(B_1 = \{(q \geq 0.6)\). Then
\[
\Pi^{B_1}_{bnd} = \{(11,1,0), (10,0,0), (01,0,0), (00,0,0)\),
\{(11,0,0), (10,1,0), (01,0,0), (00,0,0)\),
\{(11,0,0), (10,0,0), (01,0,4), (00,0,0)\),
\{(11,0,0), (10,0,0), (04,0,0), (00,0,0)\),
\{(11,0,0), (10,0,0), (00,0,0), (00,0,4)\),
\{(11,0,0), (10,0,0), (00,0,0), (00,0,0)\).
\]

Observe that there must be a \(W^# \in W_{\text{perm}}\) such that
\(W^# = \langle \omega_1, \ldots, \omega_{n_1}, \ldots, \omega_{m_{n_m}} \rangle.\) Then by the

Example 5. Suppose the vocabulary is \{q,r\} and \(B_1 = \{(q \geq 0.6)\). Let \(\alpha \) be \((q \land r) \lor (\neg q \land r)\). Then
\[
\Pi^{B_1}_{bnd} = \{(11,0,0), (10,0,0), (01,0,0), (00,0,0)\),
\{(11,0,0), (10,0,0), (01,0,0), (00,0,0)\),
\{(11,0,0), (10,0,0), (01,0,0), (00,0,0)\),
\{(11,0,0), (10,0,0), (01,0,0), (00,0,0)\),
\{(11,0,0), (10,0,0), (01,0,0), (00,0,0)\).
\]

(Two revision operations produce \{(11,0), (10,0.5), (01,0.5), (00,0)\).)

To induce the new BB \(B^\alpha_{bnd}\) from \((\Pi^{B_1}_{bnd})_{\alpha}\), the following procedure is executed. For every possible world, the procedure adds a sentence enforcing the upper (resp., lower) probability limit of the world, with respect to all the revised boundary belief states. Trivial limits are excepted.

\[
\text{For every } w \in W, (\phi_w) \leq \bar{y} \in B^\alpha, \text{ where } \bar{y} = \max_{b \in (\Pi^{B_1}_{bnd})_{\alpha}} b(w), \text{ except when } \bar{y} = 1, \text{ and } (\phi_w) \geq \underline{y} \in B^\alpha, \text{ where } \underline{y} = \min_{b \in (\Pi^{B_1}_{bnd})_{\alpha}} b(w), \text{ except when } \underline{y} = 0.
\]

The intention is that the procedure specifies \(B^\alpha\) to represent the upper and lower probability envelopes of the set of revised boundary belief states – \(B^\alpha\) thus defines the entire revised belief state space (cf. Theorem 1).

Example 6. Continuing Example 5, using the translation procedure just above, we see that \(B^\alpha_{bnd} = \{(\phi_{11}) \leq 0, (\phi_{10}) \geq 0.3, (\phi_{01}) \leq 0.7, (\phi_{00}) \geq 0\). Note that if we let \(B^\gamma = \{(q \land r) \lor (\neg q \land r)\) = 1, \((q \land r) \geq 0.3\), then \(B^\gamma = \Pi^{B_1}_{bnd}\). □

Example 7. Suppose the vocabulary is \{q,r\} and \(B_2 = \{(q \land r) > 0.1\). Let \(\alpha = \neg q\). Then
\[
\Pi^{B_2}_{bnd} = \{(11,0,9), (10,0), (01,0), (00,0.1)\),
\{(11,0), (10,0), (01,0), (00,0.1)\),
\{(11,0), (10,0), (01,0), (00,0.1)\),
\{(11,0), (10,0), (01,0), (00,0)\),
\{(11,0), (10,0), (01,0), (00,0)\).
\]

\[
(\Pi^{B_2}_{bnd})_{\alpha} = \{(11,0), (10,0), (01,0), (00,0.1)\),
\{(11,0), (10,0), (01,0), (00,0.1)\),
\{(11,0), (10,0), (01,0), (00,0)\).
\]

Note that if we let \(B^\gamma = \{(q \land r) \lor (\neg q \land r)\) = 1, \((q \land r) \geq 0.3\), then \(B^\gamma = \Pi^{B_1}_{bnd}\). □

Let \(W^{\text{Min}(\alpha, d)}\) be a partition of \(W\) such that \{\(w_0, \ldots, w_{n_d}\)\} is a block in \(W^{\text{Min}(\alpha, d)}\) iff \(\text{Min}(\alpha, w_i, d) = \cdots = \text{Min}(\alpha, w_{n_d}, d)\). Denote an element of block \{\(w_1, \ldots, w_{n_d}\)\} as \(w\), and the block of which \(w\) is an element as \(\{w\}\). Let \(i = \text{Min}(\alpha, w, d)\), in other words, the superscript in \(w\) indicates the size of \(\text{Min}(\alpha, w, d)\). Let \(m := \max_{w \in W} \text{Min}(\alpha, w, d)\).

Observation 1. Let \(\delta_1, \delta_2, \ldots, \delta_m\) be positive integers such that \(i < j\) if \(\delta_i < \delta_j\). Let \(\nu_1, \nu_2, \ldots, \nu_m\) be values in \([0,1]\) such that \(\sum_{k=1}^{m} \nu_k = 1\). Associate with every \(\nu\), a maximum value it is allowed to take: \(\text{most}(\nu)\). For every \(\nu\), we define the assignment value
\[
av(\nu) := \begin{cases} 
\text{most}(\nu) & \text{if } \sum_{k=1}^{i} \nu_k \leq 1 \\
1 - \sum_{k=1}^{i-1} \nu_k & \text{otherwise}
\end{cases}
\]

Determine first \(av(\nu_1)\), then \(av(\nu_2)\) and so on. Then
\[
av(\nu_1) + \cdots + \av(\nu_{m}) = \nu_1 + \cdots + \nu_{m}
\]
whenever \(\nu 
(\equiv av(\nu))\) for some \(i\). □

For instance, let \(\delta_1 = 1, \delta_2 = 2, \delta_3 = 3, \delta_4 = 4\). Let \(\text{most}(\nu_1) = 0.5, \text{most}(\nu_2) = 0.3, \text{most}(\nu_3) = 0.2, \text{most}(\nu_4) = 0.3\). Then \(av(\nu_1) = 0.5, av(\nu_2) = 0.3, av(\nu_3) = 0.2, av(\nu_4) = 0\) and
\[
0.5 + 0.3 + \frac{2}{3} + 0.01 = 0.716.
\]

But
\[
0.49 + 0.3 + \frac{2}{3} + 0.01 = 0.709.
\]

And
\[
0.5 + 0.29 + \frac{2}{3} + 0.01 = 0.714.
\]

Lemma 1 essentially says that the belief state in \(\Pi^B\) which causes a revised belief state to have a maximal value at world \(w\) (w.r.t. all belief states in \(\Pi^B\)), will be in \(\Pi^B_{bnd}\).

Lemma 1. For all \(w \in W\), arg max_{\alpha \in \Pi^B} \sum_{w' \in W} b_X(w')/\text{Min}(\alpha, w', d) = \text{in } \Pi^B_{bnd}.
\]

Proof. Note that
\[
\sum_{w' \in W} b(w')/\text{Min}(\alpha, w', d)
\]
can be written in the form
\[
\sum_{w' \in \{\omega_{n_1}, \ldots, \omega_{m_{n_m}}\}} \sum_{w' \in W} b(w')
\]
Observe that there must be a \(W^# \in W_{\text{perm}}\) such that \(W^# = \langle \omega_1, \ldots, \omega_{n_1}, \ldots, \omega_{m_{n_m}} \rangle.\) Then by the
definition of the set of boundary belief states (Def. 4), \( \text{MaxASAP}(B, W^\#) \) will assign maximal probability mass to \([w^1] = \{w_1^1, \ldots, w_N^1\}\), then to \([w^2] = \{w_1^2, \ldots, w_N^2\}\) and so on.

That is, by Observation 1, for some \( b_x \in \Pi_{\text{bound}}^B \), \( b_x(w) = \max_{b_x \in \Pi^B} \sum_{w' \in \text{Min}(\alpha, w', d)} b_X(w') / [\text{Min}(\alpha, w', d)] \)

for all \( w \in W \). Therefore, \( \arg \max_{b_x \in \Pi^B} \sum_{w' \in \text{Min}(\alpha, w', d)} b_X(w') / [\text{Min}(\alpha, w', d)] \) is in \( \Pi_{\text{bound}}^B \).

Definition 5 (Shannon Entropy).
\[
H(b) := - \sum_{w \in W} b(w) \ln b(w),
\]
where \( b \) is a belief state.

Lemma 2 states that for every world, the upper/lower probability of the world with respect to \( \Pi_{\text{bound}}^B \) is equal to the upper/lower probability of the world with respect to \( \Pi^B \). The proof requires Observation 1 and Lemma 1.

Lemma 2. For all \( w \in W \), \( \overline{y}^w = \overline{\Xi}^w \) and \( y^w = \Xi^w \).

Proof. Note that if \( w \not\in \{\alpha\} \), then \( \overline{y}^w = \overline{\Xi}^w = 0 \) and \( y^w = \Xi^w = 0 \).

We now consider the cases where \( w \in \{\alpha\} \).

\[
\overline{y}^w = \overline{\Xi}^w \iff \max_{b \in \Pi^B} b(w) = \max_{b \in \Pi^B} b(w)
\]

iff \( \max_{b \in \Pi^B} \sum_{w' \in W} b_x(w') / [\text{Min}(\alpha, w', d)] \)

\[
= \max_{b \in \Pi^B} \sum_{w' \in W} b_X(w') / [\text{Min}(\alpha, w', d)]
\]

if \( \overline{b}_x(w) = \overline{b}_X(w) \), where

\[
\overline{b}_x(w) := \max_{b \in \Pi^B} \sum_{w' \in W} b_x(w') / [\text{Min}(\alpha, w', d)]
\]

and

\[
b_X(w) := \max_{b \in \Pi^B} \sum_{w' \in W} b_X(w') / [\text{Min}(\alpha, w', d)].
\]

Note that
\[
\sum_{w' \in W} b(w') / [\text{Min}(\alpha, w', d)]
\]

can be written in the form
\[
\sum_{w \in \text{Min}(\alpha, w', d)} b(w) + \cdots + \sum_{w \in \text{Min}(\alpha, w', d)} b(w)
\]

Then by Observation 1, \( \overline{b}_X(w) \) is in \( \Pi_{\text{bound}}^B \). Also by Lemma 1, the belief state in \( \Pi_{\text{bound}}^B \) identified by \( \overline{b}_X(w) \) must be the one which maximizes
\[
\sum_{w' \in W} b_x(w') / [\text{Min}(\alpha, w', d)],
\]
where \( b_x \in \Pi_{\text{bound}}^B \). That is, \( \overline{b}_x = \overline{b}_X \).

With a symmetrical argument, it can be shown that \( y^w = \overline{y}^w \).

In intuitive language, the following theorem says that the BB determined through the method of revising boundary belief states captures exactly the same beliefs and ignorance as the belief states in \( \Pi^B \) which have been revised. This correspondence relies on the fact that the upper and lower probability envelopes of \( \Pi^B \) can be induce from \( \Pi_{\text{bound}}^B \), which is what Lemma 2 states.

Theorem 1. Let \( \Pi^B \) be the BB induced from \( \Pi_{\text{bound}}^B \). Then \( \Pi^B = \{b^\alpha \} \).

Proof. We show that \( \forall b' \in \Pi^B \iff b' \in \Pi_{\text{bound}}^B \).

(\( \Rightarrow \)) \( b' \in \Pi_{\text{bound}}^B \implies \forall w \in W, \overline{y}^w \leq b'(w) \leq \overline{y}^w \) (by definition of \( B_{\text{bound}}^\alpha \)). Lemma 2 states that for all \( w \in W \), \( \overline{y}^w = \overline{\Xi}^w \) and \( y^w = \Xi^w \). Hence, \( \forall w \in W, \overline{y}^w \leq b'(w) \leq \Xi^w \).

Therefore, \( \overline{b}'(w) \in \Pi^B \).

(\( \Leftarrow \)) \( b'(w) \in \Pi^B \implies \forall w \in W, \overline{y}^w \leq b'(w) \leq \Xi^w \).

Hence, by Lemma 2, \( \forall w \in W, \overline{y}^w \leq b'(w) \leq \Xi^w \).
Therefore, by definition of \( B_{\text{bound}}^\alpha \), \( b' \in \Pi^B \).

Revising via a Representative Belief State

Another approach to the revision of a belief base (BB) is to determine a representative of \( \Pi^B \) (call it \( b_{\text{rep}} \)), change the representative belief state via the defined revision operation and then induce a new BB from the revised representative belief state. Selecting a representative probability function from a family of such functions is not new (Goldszmidt, Morris, and Pearl, 1990; Paris, 1994, e.g.). More formally, given observation \( \alpha \), first determine \( b_{\text{rep}} \in \Pi^B \), then compute its revision \( b_{\text{rep}}' \), and finally induce \( \Pi^B \) from \( b_{\text{rep}}' \).

We shall represent \( \Pi^B \) (and thus \( B \)) by the single ‘least biased’ belief state, that is, the belief state in \( \Pi^B \) with highest entropy:

Definition 5 (Shannon Entropy).
\[
H(b) := - \sum_{w \in W} b(w) \ln b(w),
\]
where \( b \) is a belief state.
Definition 6 (Maximum Entropy). Traditionally, given some set of distributions \( \Pi \), the most entropic distribution in \( \Pi \) is defined as

\[
h^H := \arg \max_{b \in \Pi} H(b). \]

Suppose \( B_2 = \{ (\neg q \land r) = 0.1 \} \). Then the belief state \( b \in \Pi^{B_2} \) satisfying the constraints posed by \( B_2 \) for which \( H(b) \) is maximized is \( b_{\text{rep}} = h^H = (0.3, 0.3, 0.3, 0.1) \).

The above distribution can be found directly by applying the principle of maximum entropy: the true belief state is estimated to be the one consistent with known constraints, but is otherwise as unbiased as possible, or “Given no other knowledge, assume that everything is as random as possible.” That is, the probabilities are distributed as uniformly as possible consistent with the available information.” (Poole and Mackworth, 2010). Obiously world 00 must be assigned probability 0.1. And the remaining 0.9 probability mass should be uniformly spread across the other three worlds.

Applying GI to \( b_{\text{rep}} \) on evidence \( \neg q \) results in \( b_{\text{rep}}^q = (0, 0, 0.6, 0.4) \).

Example 8. Suppose the vocabulary is \{\( q, r \)\}, \( B_1 = \{ (q \geq 0.6) \} \) and \( \alpha = (q \land \neg r) \lor (\neg q \land r) \). Then \( b_{\text{rep}} = \arg \max_{b \in \Pi^{B_1}} H(b) = (0.3, 0.3, 0.2, 0.2) \). Applying GI to \( b_{\text{rep}} \) on \( \alpha \) results in \( b_{\text{rep}}^\alpha = (0, 0.61, 0.39, 0) \). \( b_{\text{rep}}^\alpha \) can be translated into \( B_{\text{rep}}^\alpha \) as \( \{ (q \land r) = 0.61, (\neg q \land r) = 0.39 \} \).

Still using \( \alpha = (q \land \neg r) \lor (\neg q \land r) \), notice that \( \Pi^{B_{\text{rep}}} \neq \Pi^{B_{\text{rep}}} \). But how different are \( B_{\text{rep}}^\alpha \) from the representative of \( B_{\text{rep}}^{\neg \alpha} \)? The least biased belief state satisfying \( B_{\text{rep}}^{\neg \alpha} \) is \( \{ (0, 0.5, 0.5, 0) \} \). That is, How different are \( (0, 0.61, 0.39, 0) \) and \( (0, 0.5, 0.5, 0) \)?

In the case of \( B_2, \) we could compare \( B_{\text{bound}}^{\neg q} = \{ (\phi_1) \leq 0, \ (\phi_0) \leq 0, \ (\phi_0) \geq 0.9, \ (\phi_{10}) \geq 0.1 \} \) with \( b_{\text{rep}}^q = (0, 0, 0.6, 0.4) \). Or if we take the least biased belief state satisfying \( B_{\text{bound}}^{\neg q} \), we can compare \( (0, 0, 0.5, 0.5) \) with \( (0, 0, 0.6, 0.4) \).

It has been extensively argued (Jaynes, 1978; Shore and Johnson, 1980; Paris and Vencovsk, 1997) that maximum entropy is a reasonable inference mechanism, if not the most reasonable one (w.r.t. probability constraints). And in the sense that the boundary belief states method requires no compression / information loss, it also seems like a very reasonable inference mechanism for revising BBs as defined here. Resolving this misalignment in the results of the two methods is an obvious task for future research.

Future Directions

Some important aspects still missing from our framework are the representation of conditional probabilistic information such as is done in the work of Kern-Isberner and the association of information with its level of entrenched. On the latter point, when one talks about probabilities or likelihoods, if one were to take a frequentist perspective, information observed more (less) often should become more (less) entrenched. Or, without considering observation frequencies, an agent could be designed to have, say, one or two sets of deeply entrenched background knowledge (e.g., domain constraints) which does not change or is more immune to change than ‘regular’ knowledge.

Given that we have found that the belief base resulting from revising via the boundary-belief-states approach differs from the belief base resulting from revising via the representative-belief-state approach, the question arises, ‘When is it appropriate to use a representative belief state defined as the most entropic belief state of a given set \( \Pi^B \)? This is an important question, especially due to the popularity of employing the Maximum Entropy principle in cases of underspecified probabilistic knowledge (Jaynes, 1978; Goldszmidt, Morris, and Pearl, 1990; Hunter, 1991; Voorbraak, 1999; Kern-Isberner, 2001; Kern-Isberner and Rdder, 2004) and the principle’s well-behavedness (Shore and Johnson, 1980; Paris, 1994; Kern-Isberner, 1998).

Katsuno and Mendelzon (1991) modified the eight AGM belief revision postulates (Alchourrón, Gärdenfors, and Makinson, 1985) to the following six (written in the notation of this paper), where * is some revision operator. 3

1. \( B^α \bowtie (\alpha) = 1 \).
2. If \( B \cup \{ (\alpha) = 1 \} \) is satisfiable, then \( B^α = B \cup \{ (\alpha) = 1 \} \).
3. If \( (\alpha) = 1 \) is satisfiable, then \( B^α = B^\beta \).
4. \( B^α \cup \{ (\beta) = 1 \} = B^α \land B^\beta \).
5. If \( B^α \cup \{ (\beta) = 1 \} \) is satisfiable, then \( B^α \land B^\beta \bowtie (\beta) = 1 \).

Testing the various revision operations against these postulates is left for a sequel paper.

An extended version of maximum entropy is minimum cross-entropy (MCE) (Kullback, 1968; Csiszár, 1975):

Definition 7 (Minimum Cross-Entropy). The ‘directed divergence’ of distribution \( c \) from distribution \( b \) is defined as

\[
R(c, b) := \sum_{w \in W} c(w) \ln \frac{c(w)}{b(w)}. \]

\( R(c, b) \) is undefined when \( b(w) = 0 \) while \( c(w) > 0 \); when \( c(w) = 0, R(c, b) = 0 \), because \( \lim_{x \to 0} \ln(x) = 0 \). Given new evidence \( \phi \in L^\text{prob}, \) the distribution \( c \) satisfying \( \phi \) diverging least from current belief state \( b \) is

\[
\arg \min_{c \in \Pi^{\text{rep}} \bowtie \phi} R(c, b). \]

Definition 8 (MCI). Then MCE inference (denoted (MCI)) is defined as

\[
b \text{MCI} \alpha := \arg \min_{b' \in \Pi, b'(\alpha) = 1} R(b', b). \]

In the following example, we interpret revision as MCE inference.

\footnote{In these postulates, it is sometimes necessary to write an observation \( \alpha \) as a BB, i.e., as \( \{ (\alpha) = 1 \} \) – in the present framework, observations are regarded as certain.}
Example 9. Suppose the vocabulary is \{q, r\} and \(B_1 = \{ (q \geq 0.6) \}. \) Let \( \alpha \) be \((q \land \neg r) \lor (\neg q \land r). \) Then
\[
\Pi_{bnd}^{B_1} = \{ \{ (11, 1.0), (10, 0.0), (01, 0.0), (00, 0.0) \}, \\
\{ (11, 0.0), (10, 1.0), (01, 0.0), (00, 0.0) \}, \\
\{ (11, 0.6), (10, 0.0), (01, 0.4), (00, 0.0) \}, \\
\{ (11, 0.6), (10, 0.0), (01, 0.0), (00, 0.4) \}, \\
\{ (11, 0.0), (10, 0.6), (01, 0.4), (00, 0.0) \}, \\
\{ (11, 0.0), (10, 0.6), (01, 0.0), (00, 0.4) \} \}.
\]
\[
(\Pi_{bnd}^{B_1})_{MCI} = \{ \{ (11, 0), (10, 0), (01, 1), (00, 0) \}, \\
\{ (11, 0), (10, 1), (01, 0), (00, 0) \}, \\
\{ (11, 0), (10, 0.6), (01, 0.4), (00, 0) \} \} \text{ and}
\]
\( B_{bnd}^{0} = \{ (\phi_{11}) \leq 0, (\phi_{00}) \leq 0 \}. \)

Note that if we let \( B' = \{ (q \land \neg r) \lor (\neg q \land r) \} = 1, \) then \( \Pi^{B'} = \Pi^{B_{bnd}}. \)

\[\square\]

Recall from Example 6 that \( B' \) included \((q \land \neg r) \geq 0.3. \) Hence, in this particular case, combining the boundary belief states approach with MCI results in a less informative revised belief base than when GI is used. The reason for the loss of information might be due to \( R(\cdot, \{ (11, 1.0), (10, 0.0), (01, 0.0), (00, 0.0) \}) \) and \( R(\cdot, \{ (11, 0.6), (10, 0.0), (01, 0.0), (00, 0.4) \}) \) being undefined: Recall that \( R(c, b) \) is undefined when \( b(w) = 0 \) while \( c(w) > 0. \) But then there is no belief state \( c \) for which \( c \models \alpha \) and \( R(\cdot) \) is defined (with these two belief states as arguments). Hence, there are no revised counterparts of these two belief states in \( (\Pi_{bnd}^{B_1})_{MCI}. \) We would like to analyze MCI more within this framework. In particular, in the future, we would like to determine whether a statement like Theorem 1 holds for MCI too.

In MCE inference, \( b-\)consistency of evidence \( \phi \) is defined as: There exists a belief state \( c \) such that \( c \models \phi \) and \( c \) is \( \text{totally continuous} \) with respect to \( b \) (i.e., \( b(w) = 0 \) implies \( c(w) = 0). \) MCE is unde\n
The computational complexity of the process to revise a belief base is at least exponential. This work focused on theoretical issues. If the framework presented here is ever used in practice, computations will have to be optimized.

The following example illustrates how one might deal with strict inequalities.

Example 10. Suppose the vocabulary is \{q, r\} and \( B_3 = \{ (q > 0.6) \}. \) Let \( \alpha \) be \((q \land \neg r) \lor (\neg q \land r). \) Let \( \epsilon \) be a real number which tends to 0. Then \( \Pi_{bnd}^{B_3} = \)
\[
\{ \{ (11, 1.0), (10, 0.0), (01, 0.0), (00, 0.0) \}, \\
\{ (11, 0.0), (10, 1.0), (01, 0.0), (00, 0.0) \}, \\
\{ (11, 0.6 + \epsilon), (10, 0.0), (01, 0.4 - \epsilon), (00, 0.0) \}, \\
\{ (11, 0.6 + \epsilon), (10, 0.0), (01, 0.0), (00, 0.4 - \epsilon) \}, \\
\{ (11, 0.0), (10, 0.6 + \epsilon), (01, 0.4 - \epsilon), (00, 0.0) \}, \\
\{ (11, 0.0), (10, 0.6 + \epsilon), (01, 0.0), (00, 0.4 - \epsilon) \}. 
\]
\[
(\Pi_{bnd}^{B_3})_{MCI} = \{ \{ (11, 0.0), (10, 0.5), (01, 0.5), (00, 0.0) \}, \\
\{ (11, 0.0), (10, 1.0), (01, 0.0), (00, 0.0) \}, \\
\{ (11, 0.0), (10, 0.3 + \epsilon), (01, 0.7 - \epsilon), (00, 0.0) \}, \\
\{ (11, 0.0), (10, 0.6 + \epsilon), (01, 0.4 - \epsilon), (00, 0.0) \}, \\
\{ (11, 0.0), (10, 0.8 + \epsilon), (01, 0.2 - \epsilon), (00, 0.0) \} \text{ and}
\]
\( B_{bnd}^{0} = \{ (\phi_{11}) \leq 0, (\phi_{10}) \geq 0.3 + \epsilon, (\phi_{01}) \leq 0.7 - \epsilon, (\phi_{00}) \leq 0.0 \}. \)

Note that if we let \( B' = \{ (q \land \neg r) \lor (\neg q \land r) \} = 1, \) then \( \Pi^{B'} = \Pi^{B_{bnd}}. \)

\[\square\]

It has been suggested by one of the reviewers that GI could be an affine map (i.e., geometry), thus allowing the proof of Theorem 1 to refer to existing results in the study of affine maps to significantly simplify the proof. The authors are not familiar with affine maps and thus leave investigation of the suggestion to other researchers.

Related Work

Voorbraak (1999) proposed the partial probability theory (PTT), which allows probability assignments to be partially determined, and where there is a distinction between probabilistic information based on \( (i)\) hard background evidence and \( (ii)\) some assumptions. He does not explicitly define the “constraint language”, however, from his examples and discussions, one can infer that he has something like the language \( L_{PTT} \) in mind: it contains all formulæ which can be formed with sentences in our \( L_{prob} \) in combination with connectives \( \neg, \land \) and \( \lor. \) A “belief state” in PTT is defined as the quadruple \( (\Omega, B, A, C) \), where \( \Omega \) is a sample space, \( B \subset L_{PTT} \) is a sets of probability constraints, \( A \subset L_{PTT} \) is a sets of assumptions and \( C \subset W \) represents specific information concerning the case at hand” (an observation or evidence).4 Our epistemic state can be expressed as a restricted PTT “belief state” by letting \( \Omega = W, B = B, A = \emptyset \) and

---

4Voorbraak (1999)’s “belief state” would rather be called and epistemic state or knowledge structure in our language.
\( C = \{ w \in W \mid w \models \alpha \} \), where \( B \) is a belief base and \( \alpha \) is an observation in our notation.

Voorbraak (1999) mentions that he will only consider conditioning where the evidence does not contradict the current beliefs. He defines the set of belief states corresponding to the conditionalized PPT “belief state” as \( \{ b' \in C \mid b \in B, b'(C) > 0 \} \). In our notation, this corresponds to \( \{ b \in B \mid b(C) > 0 \} \), where \( \alpha \) corresponds to \( b' \).

Voorbraak (1999) proposes constraining \( \alpha \) as an alternative to conditioning: Let \( \phi \in L^{prob} \) be a probability constraint. In our notation, constraining \( \Pi_B \) on \( \phi \) produces \( \Pi_B|_{\phi} \).

Note that expanding a belief set reduces the number of models (worlds) and expanding a PPT “belief state” with extra constraints also reduces the number of models (belief states / probability functions).

In the context of belief sets, it is possible to obtain any belief state from the ignorant belief base by a series of expansions. In PPT, constraining, but not conditioning, has the analogous property. This is one of the main reasons we prefer to constrain and not conditioning to be the probabilistic version of expansion. (Voorbraak, 1999, p. 4)

But Voorbraak does not address the issue that \( C \) and \( \phi \) are different kinds of observations, so constraining, as defined here, cannot be an alternative to conditioning. \( C \) cannot be used directly for constraining and \( \phi \) cannot be used directly for conditioning.

W.l.o.g., we can assume \( C \) is represented by \( \alpha \). If we take \( b \models \alpha \) to be an expansion operation whenever \( b(\alpha) > 0 \), then one might ask, Is it possible to obtain any belief base \( B' \) from the ignorant belief base \( B = \emptyset \) by a series of expansions, using our approach? The answer is, No. For instance, there is no observation or series of observations which can change \( B = \emptyset \) into \( B' = \{ (q) \mid q \geq 0.6 \} \). But if we were to allow sentences (constraints) in \( L^{prob} \) to be observations, then we could obtain any \( B' \) from the ignorant \( B \).

Grove and Halpern (1998) investigate what “update” (incorporation of an observation with current beliefs, such that the observation does not contradict the belief(s) means in a framework where beliefs are represented by a set of belief states. They state that the main purpose of their paper is to illustrate how different the set-of-distributions framework can be, “technically”, from the standard single-distribution framework. They propose six postulates characterizing what properties an update function should have. They say that some of the postulates are obvious, some arguable and one probably too strong. Out of seven (families of) update functions only the one based on conditioning \( (Upd_{cond}() \) and the one based on constraining \( (Upd_{constrain}() \) satisfy all six postulates, where \( Upd_{cond}(\Pi_B, \alpha) := \{ b \in B \mid b(C) > 0 \} \) and where they interpret Voorbraak’s (1999) constraining as \( Upd_{constrain}(\Pi_B, \alpha) := \{ b \in B \mid b(\alpha) = 1 \} \). Grove and Halpern (1998) do not investigate the case when an observation must be incorporated while it is (possibly) inconsistent with the old beliefs. (i.e., revision).

Kern-Isberner (2001) develops a new perspective of probabilistic belief change. Based on the ideas of Alchourrón, Gärdenfors, and Makinson (1985) and Katsuno and Mendelzon (1991) (KM), the operations of revision and update, respectively, are investigated within a probabilistic framework. She employs as basic knowledge structure a belief base \( (B, \mathcal{R}) \), where \( B \) is a probability distribution (belief state) of background knowledge and \( \mathcal{R} \) is a set of probabilistic conditionals of the form \( A \rightarrow B(x) \) meaning ‘The probability of \( B \), given \( A \), is \( x \). A universal inference operation – based on the techniques of optimum entropy – is introduced as an “adequate and powerful method to realize probabilistic belief change”.

By having a belief state available in the belief base, minimum cross-entropy can be used. The intention is then that an agent with belief base \( (b, \mathcal{T}) \) should always reason w.r.t. belief state \( b^T := \arg \min_{\in \Pi(b, \mathcal{T})} T(c, b) \). Kern-Isberner (2001) then defines the probabilistic belief revision of \( (b, \mathcal{R}) \) by evidence \( S \) as \( (b, \mathcal{R} \cup S) \). And the probabilistic belief update of \( (b, \mathcal{R}) \) by evidence \( S \) is defined as \( (b^S, S) \). She distinguishes between revision as a knowledge adding process, and updating as a change-recording process. Kern-Isberner (2001) sets up comparisons of maximum-cross-entropy belief change with AGM revision and KM update. Cases where, for update, new information \( \mathcal{R} \) is inconsistent with the prior distribution \( b \), or, for revision, is inconsistent with \( b \) or the context \( \mathcal{R} \), are not dealt with (Kern-Isberner, 2001, p. 399, 400).

Having a belief state available for modification when new evidence is to be adopted is quite convenient. As Voorbraak (1999) argues, however, an agent’s ignorance can hardly be represented in an epistemic state where a single belief state must always be chosen.

The reader may also refer to a later paper (Kern-Isberner, 2008) in which many of the results of the work just reviewed are generalized to belief bases of the form \((\Psi, \mathcal{R})\), where \( \Psi \) denotes a general epistemic state. In that paper, she considers two instantiations of \( \Psi \), namely as a probability distribution and as an ordinal conditional function (first introduced by Spohn (1988)).

Yue and Liu (2008) propose a probabilistic revision operation for imprecise probabilistic beliefs in the framework of Probabilistic Logic Programming (PLP). New evidence may be a probabilistic (conditional) formula and needs not be consistent with the original beliefs. Revision via imaging (e.g., GI) also overcomes this consistency issue.

Essentially, their probabilistic epistemic states \( \Psi \) are induced from a PLP program which is a set of formulae, each formula having the form \((\psi \mid \phi)[l, u]\), meaning that the probability of the conditional \((\psi \mid \phi) \) lies in the interval \([l, u]\).

The operator they propose has the characteristic that if an epistemic state \( \Psi \) represents a single probability distribution, revising collapses to Jeffrey’s rule and Bayesian conditioning.

They mention that it is required that the models (distributions) of \( \Psi \) is a convex set. There might thus be an opportunity to employ their revision operation on a representative set of boundary distributions as proposed in this paper.

---

This is a very simplified version of what she presents. Please refer to the paper for details.
Conclusion
In this paper, we propose an approach how to generate a new probabilistic belief base from an old one, given a new piece of non-probabilistic information, where a belief base is a finite set of sentences, each sentence stating the likelihood of a proposition about the world. In this framework, an agent’s belief base represents the set of belief states compatible with the sentences in it. In this sense, the agent is able to represent its knowledge and ignorance about the true state of the world.

We used a version of the so-called imaging approach to implement the revision operation.

Two methods were proposed: revising a finite set of ‘boundary belief states’ and revising a least biased belief state. We focussed on the former and showed that the latter gives different results.

There were two main contribution of this paper. The first was to prove that the set of belief states satisfying $B_{new}$ is exactly those belief states satisfying the original belief base, revised. The second was to uncover an interesting conflict in the results of the two belief base revision methods. It is worth further understanding the reasons behind such a difference, as such an investigation could give more insight about the mechanisms behind the two methods and indicate possible pros and cons of each.

Acknowledgements
The work of Giovanni Casini has been supported by the Fonds National de la Recherche, Luxembourg, and cofunded by the Marie Curie Actions of the European Commission (FP7-COFUND) (AFR/9181001).

References


Reactive Policies with Planning for Action Languages *

Zeynep G. Saribatur and Thomas Eiter
Institut für Informationssysteme, Technische Universität Wien
Favoritenstraße 9-11, A-1040 Vienna, Austria
{zeynep,eiter}@kr.tuwien.ac.at

Abstract

We describe a representation in a high-level transition system for policies that express a reactive behavior for the agent. We consider a target decision component that figures out what to do next and an (online) planning capability to compute the plans needed to reach these targets. Our representation allows one to analyze the flow of executing the given reactive policy, and to determine whether it works as expected. Additionally, the flexibility of the representation opens a range of possibilities for designing behaviors.

Autonomous agents are systems that decide for themselves what to do to satisfy their design objectives. These agents have a knowledge base that describes their capabilities, represents facts about the world and helps them in reasoning about their course of actions. A reactive agent interacts with its environment. It perceives the current state of the world through sensors, consults its memory (if there is any), reasons about actions to take and executes them in the environment. A policy for these agents gives guidelines to follow during their interaction with the environment.

As autonomous systems become more common in our lives, the issue of verifying that they behave as intended becomes more important. During the operation of an agent, one would want to be sure that by following the designed policy, the agent will achieve the desired results. It would be highly costly, time consuming and sometimes even fatal to realize at runtime that the designed policy of the agent does not provide the expected properties.

For example, in search and rescue scenarios, an agent needs to find a missing person in unknown environments. A naive approach would be to directly try to find a plan that achieves the main goal of finding the person. However, this problem easily becomes troublesome, since not knowing the environment causes the planner to consider all possible cases and find a plan that guarantees reaching the goal in all settings. Alternatively, one can describe a reactive policy for the agent that determines its course of actions according to its current knowledge, and guides the agent in the environment towards the main goal. A possible such policy could be “always move to the farthest unvisited point in visible distance, until a person is found”. Following this reactive policy, the agent would traverse the environment by choosing its actions to reach the farthest possible point from the current state, and by reiterating the decision process after reaching a new state. The agent may also remember the locations it has been in and gain information (e.g. obstacle locations) through its sensors on the way. Verifying beforehand whether or not the designed policy of the agent satisfies the desired goal (e.g. can the agent always find the person?), in all possible instances of the environment is nontrivial.

Action languages (Gelfond and Lifschitz 1998) provide a useful framework on defining actions and reasoning about them, by modeling dynamic systems as transition systems. Their declarative property helps in describing the system in an understandable, concise language, and they also address the problems encountered when reasoning about actions. By design, these languages are made to be decidable, which ensures reliable descriptions of dynamic systems. As these languages are closely related with classical logic and answer set programming (ASP) (Lifschitz 2008; 1999), they can be translated into logic programs and queried for computation. The programs produced by such translations can yield sound and complete answers to such queries. There have been various works on action languages (Gelfond and Lifschitz 1998; 1993; Giunchiglia and Lifschitz 1998) and their reasoning systems (Giunchiglia et al. 2004; Gebser, Grote, and Schaub 2010), with underlying mechanisms that rely on SAT and ASP solvers.

The shortage of representations that are capable of modeling reactive policies prevents one from verifying such policies using action languages as above before putting them into use. The necessity of such a verification capability motivates us to address this issue. We thus aim for a general model that allows for verifying the reactive behavior of agents in environments with different types in terms of observability and determinism. In that model, we want to use the representation power of the transition systems described by action languages and combine components that are efficient for describing reactivity.

Towards this aim, we consider in this paper agents with a reactive behavior that decide their course of actions by determining targets to achieve during their interaction with the environment. Such agents come with an (online) planning

---

*This work has been supported by Austrian Science Fund (FWF) project W1255-N23. Copyright © 2016, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
capability that computes plans to reach the targets. This method matches the observe-think-act cycle of Kowalski and Sadri (1999), but involves a planner that considers targets. The flexibility in the two components - target development and external planning - allow for a range of possibilities for designing behaviors. For example, one can use HEX (Eiter et al. 2005) to describe a program that determines a target given the current state of an agent, finds the respective plan and the execution schedule. ACTHEX programs (Fink et al. 2013), in particular, provide the tools to define such reactive behaviors as it allows for iterative evaluation of the logic programs and the ability to observe the outcomes of executing the actions in the environment. Specifically, we make the following contributions:

(1) We introduce a novel framework for describing the semantics of a policy that follows a reactive behavior, by integrating components of target establishment and online planning. The purpose of this work is not synthesis, but to lay foundations for verification of behaviors of (human-designed) reactive policies. The outsourced planning might also lend itself for modular, hierarchic planning, where macro actions (expressed as targets) are turned into a plan of micro actions. Furthermore, outsourced planning may also be exploited to abstract from correct sub-behaviors (e.g. going always to the farthest point).

(2) We relate this to action languages and discuss possibilities for policy formulation. In particular, we consider the action language C (Giunchiglia and Lifschitz 1998) to illustrate an application.

The remainder of this paper is organized as follows. After some preliminaries, we present a running example and then the general framework for modeling policies with planning. After that, we consider the relation to action languages, and as a particular application we consider (a fragment of) the action language C. We briefly discuss some related work and conclude with some issues for ongoing and future work.

**Preliminaries**

**Definition 1.** A transition system $T$ is defined as $T = \langle S, S_0, A, \Phi \rangle$ where

- $S$ is the set of states.
- $S_0 \subseteq S$ is the set of possible initial states.
- $A$ is the set of possible actions.
- $\Phi : S \times A \rightarrow 2^S$ is the transition function, returns the set of possible successor states after applying a possible action in the current state.

For any states $s, s' \in S$, we say that there is a trajectory between $s$ and $s'$, denoted by $s \rightarrow^\sigma s'$ for some action sequence $\sigma = a_1, \ldots, a_n$ where $n \geq 0$, if there exist $s_0, \ldots, s_n \in S$ such that $s = s_0, s' = s_n$ and $s_{i+1} \in \Phi(s_i, a_{i+1})$ for all $0 \leq i < n$.

We will refer to this transition system as the original transition system. The constituents $S$ and $A$ are assumed to be finite in the rest of the paper. Note that, this transition system represents fully observable settings. Large environments cause high number of possibilities for states, which cause the transition systems to be large. Especially, if the environment is nondeterministic, the resulting transition system contains high amount of transitions between states, since one needs to consider all possible outcomes of executing an action.

**Action Languages**

Action languages describe a particular type of transition systems that are based on action signatures. An action signature consists of a set $V$ of value names, a set $F$ of fluent names and a set $A$ of action names. Any fluent has a value in any state of the world.

A transition system of an action signature $(V, F, A)$ is similar to Defn. 1, where $A = A$ and $\Phi$ corresponds to the relation $R \subseteq S \times A \times S$. In addition, we have a value function $V : F \times S \rightarrow V$, where $V(P, s)$ shows the value of $P$ in state $s$. A transition system can be thought as a labeled directed graph, where a state $s$ is represented by a vertex labeled with the function $P \rightarrow V(P, s)$, that gives the value of the fluents. Every triple $(s, a, s') \in R$ is represented by an edge leading from a state $s$ to a state $s'$ and labeled by $a$.

An action $a$ is executable at state $s$, if there is at least one state $s'$ such that $(s, a, s') \in R$ and $a$ is deterministic if there is at most one such state. Concurrent execution of actions can be defined by considering transitions in the form $(s, A, s')$ with a set $A \subseteq A$ of actions, where each action $a \in A$ is executable at $s$.

An action signature $(V, F, A)$ is propositional if its value names are truth values: $V = \{ f, t \}$. In this work, we confine to propositional action signatures.

The transition system allows one to answer queries about the program. For example, one can find a plan to reach a goal state from an initial state, by searching for a path between the vertices that represent these states in the transition system. One can express properties about the paths of the transition system by using an action query language.

**Running Example: Search Scenarios**

Consider a memoryless agent that can sense horizontally and vertically, in an unknown $n \times n$ grid cell environment with obstacles, where a missing person needs to be found. Suppose we are given the action description of the agent with a policy of “always going to the farthest reachable point in visible distance (until a person is found)”. Following this reactive policy, the agent chooses its course of actions to reach the farthest reachable point, referred as target, from its current location with respect to its current knowledge about the environment. After executing the plan and reaching a state that satisfies the target, the decision process is reiterated and a new target, hence a new course of actions, is determined.

Given such a policy, one would want to check whether or not the agent can always find the person, in all instances of the environment. Note that we assume that the obstacles are placed in a way that the person is always reachable.

Figure 1 shows some possible instances for $n = 3$, where the square in a cell represents the agent and the dot represents the missing person. The course of actions determined by the policy in all the instances is to move to $(3,1)$, which is the
farthest reachable point, i.e. target. It can be seen that (a) is an instance where the person can be found with the given policy, while in (b) the agent goes in a loop and can’t find the person, since after reaching (3,1) it will decide to move to (1,1) again. In (c), after reaching (3,1) following its policy, the agent has two possible directions to choose, since there are two farthest points. It can either move to (3,3), which would result in seeing the person, or it can move back to (1,3), which would mean that there is a possibility for the agent to go in a loop.

Notice that our aim is different from finding a policy (i.e. global plan) that satisfies certain properties (i.e. goals) in an unknown environment. On the contrary, we assume that we are given a representation of a system with a certain policy, and we want to check what it is capable (or incapable) of.

Modeling Policies in Transition Systems

A reactive policy is described to reach some main goal, by guiding the agent through its interaction with the environment. This guidance can be done by determining the course of actions to bring about targets from the current situation, via externally computed plans. A transition system that models such policies should represent the flow of executing the policy, which is the agent’s actual trajectory in the environment following the policy. This would allow for verifying whether execution of a policy results in reaching the desired main goal, i.e. the policy works.

We define such a transition system by clustering the states into groups depending on a profile. A profile is determined by evaluating a set of formulas over a state that informally yield attribute (respectively feature) values; states with the same attribute values are clustered into one. The choice of formulas for determining profiles depends on the given policy or the environment one is considering. Then, the transitions between these clusters are defined according to the policy. The newly defined transitions are able to show the evaluation of the policy by a higher level action from one state to the next state. This next state satisfies the target determined by a target component, and the higher level action corresponds to the execution of an externally computed plan.

Having such a classification on states and defining higher level transitions between the states can help in reducing the state space or the number of transitions when compared to the original transition system. Furthermore, it aids in abstraction and allows one to emulate a modular hierarchic approach, in which a higher level (macro) action, expressed by a target, is realized in terms of a sequence of (micro) actions that is compiled by the external planner, which may use different ways (planning on the fly, resorting to scripts etc.)

State profiles according to the policy

We now describe a classification of states, which helps to omit parts of the state that are irrelevant with respect to the environment or the policy. This classification is done by determining profiles, and clustering the states accordingly.

Example 1. Remember the possible instances from Figure 1 in the running example. Due to partial observability, the agent is unable to distinguish the states that it is in, and the unobservable parts are irrelevant to the policy. Now assume that there are fluents that hold the information of the agent’s location, the locations of the obstacles and the reachable points. One can determine a profile of the form “the agent is at (1,1), sees an obstacle at (1,3), and is able to reach to points (1,2), (2,1), (3,1)” by not considering the remaining part of the environment that the agent cannot observe. The states that have this profile can be clustered in one group as in Figure 2, where the cells with question marks demonstrate that they are not observable by the agent.

For partially observable environments, the notion of indistinguishable states can be used in the classification of states. The states that provide the same observations for the agent are considered as having the same profile. However, in fully observable environments, observability won’t help in reducing the state space. One needs to find other notions to determine profiles.

We consider a classification function, \( h : S \rightarrow \Omega_h \), where \( \Omega_h \) is the set of possible state clusters. This is a general notion applicable to fully and partially observable cases.

Definition 2. An equalized state relative to the classification function \( h \) is a state \( \hat{s} \in \Omega_h \). The term equalized comes from the fact that the states in the same classification are considered as the same, i.e. equal.

To talk about a state \( s \) that is clustered into an equalized state \( \hat{s} \), we use the notation \( s \in \hat{s} \), where we identify \( \hat{s} \) with its pre-image (i.e. the set of states that are mapped to \( \hat{s} \) according to \( h \)).

Different from the work by Son and Baral (2001) where they consider a “combined-state” which consists of the real state of the world and the states that the agent thinks it may be in, we consider a version where we combine the real states into one state if they provide the same classification (or observation, in case of partial observability) for the agent. The equalization of states allows for omitting the details that are irrelevant to the behavior of the agent.

Transition systems according to the policy

We now define the notion of a transition system that is able to represent the evaluation of the policy on the state clusters.
**Definition 3.** An equalized (higher level) transition system $T_h$, with respect to the classification function $h$, is defined as $T_h = \langle S, S_0, G_B, B, \Phi_B \rangle$, where

- $\hat{S}$ is the finite set of equalized states;
- $\hat{S}_0 \subseteq \hat{S}$ is the finite set of initial equalized states, where $\hat{s} \in \hat{S}_0$ if there is some $s_i \in \hat{s}$ such that $s_i \in S_0$ holds;
- $G_B$ is the finite set of possible targets relative to the behavior, where a target can be satisfied by more than one equalized state;
- $B : \hat{S} \rightarrow 2^{G_B}$, is the target function that returns the possible targets to achieve from the current equalized state, according to the policy;
- $\Phi_B : \hat{S} \rightarrow 2^{\hat{S}}$ is the transition function according to the policy, referred to as the policy execution function, returns the possible resulting equalized states after applying the policy in the current equalized state.

The target function gets the equalized state as input and produces the possible targets to achieve. These targets may be expressed as formulas over the states (in particular, of states that are represented by fluents or state variables), or in some other representation. A target can be considered as a subgoal condition to hold at the follow-up state, depending on the current equalized state. The aim would be to intend to reach a state that satisfies the conditions of the target, without paying attention to the steps taken in between. That’s where the policy execution function comes into the picture.

The formal description of the policy execution function is as follows:

$$\Phi_B(\hat{s}) = \{\hat{s}' \mid \hat{s}' \in \text{Res}(\hat{s}, \sigma), \sigma \in \text{Reach}(\hat{s}, g_B), g_B \in B(\hat{s})\},$$

where $\text{Reach}$ is an outsourced function that returns a plan $\sigma = (a_1, \ldots, a_n), n \geq 0$ needed to reach a state that meets the conditions $g_B$ from the current equalized state $\hat{s}$:

$$\text{Reach}(\hat{s}, g_B) \subseteq \{\sigma \mid \forall \hat{s}' \in \text{Res}(\hat{s}, \sigma) : \hat{s}' \models g_B\}$$

where $\parallel g_B \leftrightarrow \forall \hat{s} \in \hat{s} : \hat{s} \models g_B$, and $\text{Res}$ gives the resulting states of executing a sequence of actions at a state $\hat{s}$:

$$\text{Res}(\hat{s}, \{a_1, \ldots, a_n\}) = \{\hat{s}'' \mid \exists \hat{s}' \in \text{Res}(\hat{s}', \{a_2, \ldots, a_n\}) \land \hat{\Phi}(\hat{s}, a_1) \neq \emptyset\}$$

and

$$\text{Res}(\hat{s}, \{\}) = \{\hat{s}\}$$

where the state $\hat{s}_{err}$ is an artifact state that does not satisfy any of the targets, and

$$\hat{\Phi}(\hat{s}, a) = \{\hat{s}' \mid \exists \hat{s}'' \in \hat{s}' \exists \hat{s} \in \hat{s} : \hat{s}' \in \Phi(\hat{s}, a)\}.$$ 

Figure 3 demonstrates a transition in the equalized transition system. The equalized states may contain more than one state that has the same profile. Depending on the current state, $\hat{s}$, the policy chooses the next target, $g_B$, that should be satisfied. There may be more than one equalized state satisfying the same target. The policy execution function $\Phi_B(\hat{s})$ finds a transition into one of these equalized states, $\hat{s}'$, that is reachable from the current equalized state. The transition $\Phi_B$ is considered as a big jump between states, where the actions taken and the states passed in between are omitted.

Notice that we assume that the outsourced $\text{Reach}$ function is able to return conformant plans that guarantee to reach a state that satisfies the determined targets. In particular, $\sigma$ may also contain only one action. For practical reasons, we consider $\text{Reach}$ to be able to return a subset of all conformant plans. The maximal possible $\text{Reach}$, where we have equality, is denoted with $\text{Reach}_0$.

Consider the case of uncertainty, where the agent requires to do some action, e.g. `checkDoor`, in order to get further information about its state. One can define the target function to return as target a dummy fluent to ensure that the action is made, e.g. `doorIsChecked`, and given this target, the $\text{Reach}$ function can return the desired action as the plan. The non-determinism or partial observability of the environment is modeled through the set of possible successor states returned by $\text{Res}$.

The generic definition of the equalized transition system allows for the possibility of representing well-known concepts like purely reactive systems or universal planning (Cimatti, Riveri, and Traverso 1998a). To represent reactive systems, one can describe a policy of “pick some action”. This way one can model reactive systems that do not do reasoning, but immediately react to the environment with an action. As for the exactly opposite case, which is finding a plan that guarantees reaching the goal, one can choose the target as the main goal. Then, the $\text{Reach}$ would have the difficult task of finding a universal plan or a conformant plan that reaches the main goal. If however, one is aware of such a plan, then it is possible to mimic the plan by modifying the targets $g_B$ and the target function $B$ in a way that at each point in time the next action in the plan is returned by $\text{Reach}$, and the corresponding transition is made. For that, one needs to record information in the states and keep track of the targets.

As the function $\text{Reach}$ is outsourced, we rely on an implementation that returns conformant plans for achieving transitions in the equalized transition systems. This naturally raises the issue of whether a given such implementation is suitable, and leads to the question of soundness (only correct
plans are output) and completeness (some plan will be output, if one exists). We next assess how expensive it is to test this, under some assumptions about the representation and computational properties of (equalized) transition systems, which will then also be used for assessing the cost of policy checking.

Assumptions We have certain assumptions on the representation of the system. We assume that given a state \( s \in S \) which is implicitly given using a binary encoding, the cost of evaluating the classification \( h(s) \), the (original) transition \( \Phi(s, a) \) for some action \( a \), and recognizing the initial state, say with \( \Phi_{\text{init}}(s) \), is polynomial. The cost could also be in NP, if projective (i.e. existentially quantified) variables are allowed. Furthermore, we assume that the size of the representation of a “target” in \( G_B \) is polynomial in size of the state, so that given a string, one can check in polynomial time if it is a correct target description \( g_B \). This test can also be relaxed to be in NP by allowing projective variables.

Given these assumptions, we have the following two results. These results show the cost of checking whether an implementation of \( \text{Reach} \) that we have at hand is sound (delivers correct plans) and in case does not skip plans (is complete); we assume here that testing whether \( \sigma \in \text{Reach}(\hat{s}, g_B) \) is feasible in \( \Pi^P_2 \) (this is the cost of verifying conformant plans, and we may assume that \( \text{Reach} \) is no worse than a naive guess and check algorithm).

Theorem 1 (soundness of \( \text{Reach} \)). Let \( \mathcal{T}_\sigma = (\hat{S}, \hat{S}_0, G_B, \mathcal{E}, \Phi_B) \) be a transition system with respect to a classification function \( h \). The problem of checking whether every transition found by the policy execution function \( \Phi_B \) induced by a given implementation \( \text{Reach} \) is correct is in \( \Pi^P_2 \).

Proof (Sketch). According to the definition of the policy execution function, every transition from a state \( s \) to some state \( s' \) corresponds to some plan \( \sigma \) returned by \( \text{Reach}(\hat{s}, g_B) \). Therefore, first one needs to check whether each plan \( \sigma = (a_1, a_2, \ldots, a_n) \) returned by \( \text{Reach} \) given some \( s \) and \( g_B \) is correct. For that we need to check two conditions on the corresponding trajectories of the plan:

(i) for all partial trajectories \( \hat{s}_0, \hat{s}_1, \ldots, \hat{s}_{i-1} \) it holds that for the upcoming action \( a_i \) from the plan \( \sigma \), \( \Phi(\hat{s}_{i-1}, a_i) \neq \emptyset \) (i.e. the action is applicable)
(ii) for all trajectories \( \hat{s}_0, \hat{s}_1, \ldots, \hat{s}_n \), \( \hat{s}_n \models g_B \).

Checking whether these conditions hold is in \( \Pi^P_2 \).

Thus, to decide whether for some state \( \hat{s} \) and target \( g_B \) the function \( \Phi_B(\hat{s}, g_B) \) does not work correctly, we can guess \( \hat{s} \) (resp. \( s \in \hat{s} \)), \( g_B \) and a plan \( \sigma \) and verify that \( \sigma \in \text{Reach}(\hat{s}, g_B) \) and that \( \sigma \) is not correct. As the verification is doable with an oracle for \( \Sigma^P_3 \) in polynomial time, a counterexample for correctness can be found in \( \Sigma^P_3 \); thus the problem is in \( \Pi^P_2 \).

The complexity is lower, if output checking of \( \text{Reach} \) has lower complexity (in particular, it drops to \( \Pi^P_2 \) if output checking is in co-NP).

The result for soundness of \( \text{Reach} \) is complemented with another result for completeness with respect to short (polynomial size) conformant plans that are returned by \( \text{Reach} \).

Theorem 2 (completeness of \( \text{Reach} \)). Let \( \mathcal{T}_\sigma = (\hat{S}, \hat{S}_0, G_B, \mathcal{E}, \Phi_B) \) be a transition system with respect to a classification function \( h \). Deciding whether for a given implementation \( \text{Reach} \), \( \Phi_B \) fulfills \( \hat{s}' \in \Phi_B(\hat{s}) \) whenever a short conformant plan from \( \hat{s} \) to \( \hat{s}' \) exists in \( \mathcal{T}_\sigma \), is in \( \Pi^P_3 \).

Proof (Sketch). For a counterexample, we can guess some \( \hat{s} \) and \( \hat{s}' \) (resp. \( s \in \hat{s} \), \( s' \in \hat{s}' \)) and some short plan \( \sigma \) and verify that (i) \( \sigma \) is a valid conformant plan in \( \mathcal{T}_\sigma \) to reach \( \hat{s}' \) from \( \hat{s} \), and (ii) that a target \( g_B \) exists such that \( \text{Reach}(\hat{s}, g_B) \) produces some output. We can verify (i) using a \( \Pi^P_2 \) oracle to check that \( \sigma \) is a conformant plan, and we can verify (ii) using a \( \Pi^P_3 \) oracle (for all guesses of targets \( g_B \) and short plans \( \sigma' \), either \( g_B \) is not a target for \( s \) or \( \sigma' \) is not produced by \( \text{Reach}(\hat{s}, g_B) \)). This establishes membership in \( \Pi^P_3 \).

As in the case of soundness, the complexity drops if checking the output of \( \text{Reach} \) is lower (in particular, to \( \Pi^P_2 \) if the output checking is in co-NP).

We also restrict the plans \( \sigma \) that are returned by \( \text{Reach}(\hat{s}, g_B) \) to have polynomial size. This constraint would not allow for exponentially long conformant plans (even if they exist). Thus, the agent is forced under this restriction to develop targets that it can reach in polynomially many steps, and then to go on from these targets. Informally, this does not limit the capability of the agent in general. The “long” conformant plans can be split into short plans with a modified policy and by encoding specific targets into the states.

We denote the main goal that the reactive policy is aiming for by \( g_{\max} \). Our aim is to have the capability to check whether following the policy always results in reaching some state that satisfies the main goal. That is, for each run, i.e. sequence \( s_0, s_1, \ldots \) such that \( s_0, s_1, \ldots \) is in the equalized transition system and \( s_i = \Phi_B(s_i) \) for all \( i \geq 0 \), there is some \( j \geq 0 \) such that \( s_j \models g_{\max} \). (The behavior could be easily modified to stop or to loop in any state \( s \) that satisfies the goal.) This way we can say whether the policy works or not. Under the assumptions from above, we obtain the following

Theorem 3. The problem of determining that the policy works is in \( \text{PSPACE} \).

Proof (Sketch). One needs to look at all runs \( \hat{s}_0, \hat{s}_1, \ldots \) from every initial state \( \hat{s}_0 \) in the equalized transition system and check whether each such run has some state \( \hat{s}_j \) that satisfies the main goal \( g_{\max} \). Given that states have a representation in terms of fluent or state variables, there are at most exponentially many different states. Thus to find a counterexample, a run of at most exponential length in which \( g_{\max} \) is not satisfied is sufficient. Such a run can be nondeterministically built in polynomial space; as \( \text{NPSPACE} = \text{PSPACE} \), the result follows.

Note that in this formulation, we have tacitly assumed that the main goal can be established in the original system, thus at least some trajectory from some initial state to a state fulfilling the goal exists (this can be checked in \( \text{PSPACE} \) as...
well). In a more refined version, we could define the working of a policy relative to the fact that some abstract plan would exist that makes $g_\infty$ true; naturally, this may impact the complexity of the policy checking.

Above, we have been considering arbitrary states, targets and transitions in the equalized transition system. In fact, for the particular behavior, only the states that can be encountered in runs really matter; these are the reachable states defined as follows.

**Definition 4.** A state $\hat{s}$ is **reachable** from an initial state in the equalized transition system if and only if $s \in R_i$, for some $i \in \mathbb{N}$ where $R_i$ is defined as follows:

\[
R_0 = \hat{S}_0 \\
R_{i+1} = \bigcup_{s \in R_i} \Phi_B(\hat{s}) \\
R_\infty = \bigcup_{i \geq 0} R_i.
\]

Under the assumptions that apply to the previous results, we can state the following:

**Theorem 4.** The problem of determining whether a state in an equalized transition system is reachable is in PSPACE.

The notions of soundness and completeness of an outsourced planning function $\text{Reach}$ could be restricted to reachable states; however, this, would not change the cost of testing these properties in general (assuming that $\hat{s}$ is decidable with sufficiently low complexity).

**Constraining equalization**

The definition of $\hat{\Phi}$ allows for certain transitions between equalized states that don’t have corresponding concrete transitions in the original transition system. However, the aim of defining such an equalized transition system is not to introduce new features, but to keep the structure of the original transition system and discard the unnecessary parts with respect to the policy. Therefore, one needs to give further restrictions on the transitions of the equalized transition system, in order to obtain the main objective.

Let us consider the following condition

\[
\hat{s}' \in \hat{\Phi}(\hat{s}, a) \Leftrightarrow \forall s' \in \hat{s}', \exists s \in \hat{s}: s' \in \Phi(s, a) \tag{1}
\]

This condition ensures that a transition between two states $\hat{s}_1, \hat{s}_2$ in the equalized transition system represents that any state in $\hat{s}_2$ has a transition from some state in $\hat{s}_1$. An equalization is called **proper** if condition (1) is satisfied.

**Theorem 5.** Let $T_B = (\hat{S}, \hat{S}_0, \hat{G}_B, B, \hat{\Phi}_B)$ be a transition system with respect to a classification function $h$. Let $\hat{\Phi}$ be the transition function that the policy execution function $\Phi_B$ is based on. The problem of checking whether $\hat{\Phi}$ is proper is in $\Pi^p_2$.

**Proof (sketch).** As a counterexample, one needs to guess $\hat{s}, a, \hat{s}' \in \hat{\Phi}(\hat{s}, a)$ and $s' \in \hat{s}'$ such that no $s \in \hat{s}$ has $s' \in \Phi(s, a)$.

The results in Theorems 1-5 are all complemented by lower bounds for realistic realizations of the parameters (notably, for typical action languages such as fragments of $\mathcal{C}$).

The following proposition is based on the assumption that the transition function $\Phi$ satisfies condition (1).

**Proposition 1 (soundness).** Let $T_B = (\hat{S}, \hat{S}_0, \hat{G}_B, B, \hat{\Phi}_B)$ be a transition system with respect to a classification function $h$. Let $\hat{s}_1, \hat{s}_2 \in \hat{S}$ be equalized states that are reachable from some initial states, and $\hat{s}_2 \in \Phi_B(\hat{s}_1)$. Then for any concrete state $s_2 \in \hat{s}_2$ there is a concrete state $s_1 \in \hat{s}_1$ such that $s_1 \xrightarrow{\sigma} s_2$ for some action sequence $\sigma$, in the original transition system.

**Proof.** For equalized states $\hat{s}_1, \hat{s}_2$, having $\hat{s}_2 \in \Phi_B(\hat{s}_1)$ means that $\hat{s}_2$ satisfies a goal condition that is determined at $\hat{s}_1$, and is reachable via executing some plan $\sigma$. With the assumption that (1) holds, we can apply backwards tracking from any state $s_2 \in \hat{s}_2$ following the transitions $\hat{\Phi}$ corresponding to the actions in the plan $\sigma$ backwards. In the end, we can find a concrete state $s_1 \in \hat{s}_1$, from which one can reach the state $s_2 \in \hat{s}_2$ by applying the plan $\sigma$ in the original transition system.

Thus, we can conclude the following corollary, with the requirement of only having initial states clustered into the equalized initial states (i.e. no “non-initial” state is mapped to an initial equalized state). Technically, it should hold that $\forall s \in S_0 : h^{-1}(h(s)) \subseteq S_0$.

**Corollary 1.** If there is a trajectory in the equalized transition system with initial state clustering from an equalized initial state $s_0$ to $g_\infty$, then it is possible to find a trajectory in the original transition system from some concrete initial state $s_0 \in S_0$ to $g_\infty$.

We want to be able to study the reactive policy through the equalized transition system. In case the policy does not work as expected, there should be trajectories that shows the reason of the failure. Knowing that any such trajectory found in the equalized transition system exists in the original transition system is enough to conclude that the policy indeed does not work.

Current assumptions can not avoid the case where a plan $\sigma$ returned by $\text{Reach}$ on the equalized transition system does not have a corresponding trajectory in the original transition system. Therefore, we consider an additional condition as

\[
\hat{s}' \in \hat{\Phi}(\hat{s}, a) \Leftrightarrow \forall s \in \hat{s}, \exists s' \in \hat{s}' : s' \in \Phi(s, a) \tag{2}
\]

that strengthens the properness condition (1). Under this condition, every plan returned by $\text{Reach}$ can be successfully executed in the original transition system and will establish the target $g_B$. However, still we may lose trajectories of the original system as by clustering states they might not turn into conformant plans. Then one would need to modify the description of determining targets, i.e. the set of targets $G_B$ and the function $B$.

**Example 2.** Remember the environment and the policy described in the running example, and consider the scenario shown in Figure 4(a). It shows a part of the equalized transition system constructed according to the policy. The states that are not distinguishable due to the partial observability are clustered into the same state.
The policy is applied according to current observations, and the successor states show the possible resulting states. The aim of the policy is to have the agent move to the farthest reachable point, which for \( s_1 \) is \((3,1)\). As expected, there can be several states that satisfy the target \( g_B = \text{robotAt}(3,1) \). The successor states of \( \Phi_B(s_1) \) is determined by \( \text{Res}(s_1, \sigma) \) computing the possible resulting states after executing the plan \( \sigma \) returned by \( \text{Reach}(s_1, g_B) \). Considering that the agent will gain knowledge about the environment while moving, there are several possibilities for the resulting state.

Notice that this notion of a transition system can help in reducing the number of states, due to the fact that it is able to disregard states with information on fluents that do not have any effect on the system’s behavior. For example, Figure 4(b) shows a case where the unknown parts behind the obstacles are not relevant to the agent’s behavior, i.e. the person can be found nonetheless.

### Relation with Action Languages

In this section, we describe how our definition of a higher-level transition system that models the behavior can fit into the action languages. Given a program defined by an action language and its respective (original) transition system, we now describe how to model this program following a reactive policy and how to construct the corresponding equalized transition system according to the policy.

#### Classifying the state space

The approach to classify the (original) state space relies on defining a function that classifies the states. There are at least two kinds of such classification; one can classify the states depending on whether they give the same values for certain fluents and omit the knowledge of the values of the remaining fluents, or one can introduce a new set of fluents and classify the states depending on whether they give the same values for the new fluents:

- **Type 1**: Extend the set of truth values by \( V' = V \cup \{u\} \), where \( u \) denotes the value to be unknown. Extend the value function by \( V' : F \times S \rightarrow V' \). Then, consider a new set of groups of states, \( \hat{S} = \{\hat{s}_1, \ldots, \hat{s}_n\} \), where a group state \( \hat{s}_i \) contains all the states \( s \in S \) that give the same values for all \( p \in F \), i.e. \( \hat{S} = \{\hat{s} \mid \forall d,e \in S, d,e \in \hat{s} \iff \forall p \in F : V'(p,d) = V'(p,e)\} \). The value function for the new group of states is \( \hat{V} : F \times \hat{S} \rightarrow V' \).

- **Type 2**: Consider a new set of (auxiliary) fluent names \( F_a \), where each fluent \( p \in F_a \) is related with some fluents of \( F \). The relation can be shown with a mapping \( \Delta : 2^F \times V \rightarrow F_a \times V \). Then, consider a new set of groups of states, \( \hat{S} = \{\hat{s}_1, \ldots, \hat{s}_n\} \), where a group state \( \hat{s}_i \) contains all the states \( s \in S \) that give the same values for all \( p \in F_a \), i.e. ~\( \hat{S} = \{\hat{s} \mid \forall d,e \in S, d,e \in \hat{s} \iff \forall p \in F_a : V(p,d) = V(p,e)\} \). The value function for the new group of states is \( \hat{V} : F_a \times \hat{S} \rightarrow V \).

We can consider the states in the same classification to have the same *profile*, and the classification function \( h \) as a membership function that assigns the states into groups.

#### Remarks:

1. In Type 1, introducing the value unknown for the fluents allows for describing sensing actions and knowing the true value of a fluent at a later state. Also, one needs to give constraints for a fluent to have the unknown value. E.g. it can’t be the case that a fluent related to a grid cell is unknown while the robot is able to observe it.

2. In Type 2, one needs to modify the action descriptions according to the newly defined fluents and define abstract actions. However, in Type 1, the modification of the action definitions is not necessary, assuming that the actions are defined in a way that the fluents that are used when determining an action always have known values.

Once a set of equalized states is constructed according to the classification function, one needs to define the reactive policy to determine the transitions. Next, we describe how a policy can be defined from an abstract point of view, through a target language which figures out the targets and helps in determining the course of actions, and show how the transitions are constructed.

#### Defining a target language

Let \( \mathcal{F} \) denote the set of fluents that the equalized transition system is built upon. Let \( \mathcal{F}(\mathcal{F}) \) denote the set of formulas in an abstract language that can be constructed over \( \mathcal{F} \).

We consider a declarative way of finding targets. Let \( \mathcal{F}_B(\mathcal{F}) \subseteq \mathcal{F}(\mathcal{F}) \) be the set of formulas that describe target determination. Let \( \mathcal{F}_{\mathcal{B}_n}(\mathcal{F}) \subseteq \mathcal{F}(\mathcal{F}) \) denote the set of possible targets that can be determined via the evaluation of the formulas \( \mathcal{F}_B(\mathcal{F}) \) over the related fluents in the equalized states.

Notice that separation of the target determining formulas \( \mathcal{F}_B(\mathcal{F}) \) and the targets \( \mathcal{F}_{\mathcal{B}_n}(\mathcal{F}) \) is to allow for outsourced planners that are able to understand simple target formulas. These planners do not need to know about the target language in order to find plans. However, if one is able to use planners that are powerful enough, then the target language can be given as input to the planner, so that the planner determines the target and finds the corresponding plan.
To define a relation between $F_B(\hat{F})$ and $F_G(\hat{F})$, we introduce some placeholder fluents. Let $F_B(\hat{F}) = \{f_1, \ldots, f_n\}$ be the set of target formulas. Consider a new set of fluents $\hat{F}_B = \{p_{f_1}, \ldots, p_{f_n}\}$ where each of the formulas in $F_B$ is represented by some fluent. The value of a fluent depends on whether its respective formula is satisfied or not, i.e. for a state $s$, $s \models f \iff V(p_{f_j}) = t$. Now consider a mapping $M : 2^{F_B(\hat{F})} \rightarrow 2^{F_G(\hat{F})}$ where

\[ M(\{p_{f_1}, p_{f_2}, \ldots, p_{f_m}\}) = \{g_1, \ldots, g_r\}, m \leq n \text{ and } r \geq 1 \]

means that if there is a state $s$ such that $s \models f_i, 1 \leq i \leq m$ and $s \not\models f$ for the remaining formulas $f \in F_B(\hat{F}) \setminus \{f_1, \ldots, f_m\}$, then in the successor state $s'$ of $s$, $s' \models g_i$ for some $1 \leq i \leq r$, should hold. We consider the output of $M$ to be a set of targets in order to represent the possibility of nondeterminism in choosing a target.

**Transition between states**

The transition for the equalized transition system can be denoted with $R \subseteq S \times \hat{S}$, where $R$ corresponds to the policy execution function $\Phi_B$ that uses (a) the target language to determine targets, (b) an outsourced planner (corresponding to the function Reach) to find conformant plans and (c) the computation of executing the plans (corresponding to the function Res). The outsourced planner finds a sequence of actions $\sigma \in 2^A$ from an equalized state $s$ to one of its determined targets $g$. Then the successor equalized states are computed by executing the plan from $s$. Transition $R$ shows the resulting states after applying the policy.

**Example 3.** Let us consider a simple blocksworld example where a policy (of two phases) is defined as follows:

- if at phase 1 and not all the blocks are on the table, move one free block on a stack with highest number of blocks to the table.
- if all the blocks are on the table, move to phase 2.
- if at phase 2 and not all the blocks are on top of each other, move one of the free blocks on the top of the stack with more than one block (if exists any, otherwise move the block on top of some block).

Since the policy does not take blocks’ labels into consideration, a classification can be of the following form for $n$ number of blocks: We introduce an $n$-tuple $(b_1, \ldots, b_n)$ to denote equalized states such that for $i \leq n$, $b_i$ would represent the number of stacks that have $i$ blocks. For example, for 4 blocks, a state $(1, 0, 1, 0)$ where $b_1 = 1, b_2 = 0, b_3 = 1, b_4 = 0$ would represent all the states in the original transition system with the profile “contains a stack of 1 block and a stack of 3 blocks”. Notice that in the original transition system for 4 labeled blocks, there are 24 possible states that have the profile and if the blocks need to be in order, then there are 4 possible states.

Figure 5 demonstrates the corresponding equalized transition system for the case of 4 blocks. The equalized transition system for this example is in the following form:

- $\hat{S}$ is the set of equalized states according to the abstraction as described above.

![Figure 5: Eq. transition system of blocksworld ($n = 4$)](attachment:image.png)

**Application on Action Language $C$**

In this section, we describe how one can construct an equalized transition system for a reactive system that is represented using the action language $C$ (Giunchiglia and Lifschitz 1998). First, we give some background information about the language $C$, then move on to the application of our definitions.

**Syntax** A formula is a propositional combination of fluents. Given a propositional action signature $\langle \{f, t\}, F, E \rangle$, whose set $E$ of elementary action names is disjoint from $F$, an action description is a set of expressions of the following forms:

- **static laws:**
  $\text{caused } F \text{ if } G$. \hfill (3)

  where $F$ and $G$ are formulas that do not contain elementary actions;

- **dynamic laws:**
  $\text{caused } F \text{ if } G \text{ after } U$. \hfill (4)

  where $F$ and $G$ are as above, and $U$ is a formula.

**Semantics** The transition system $\langle S, V, R \rangle$ described by an action description $D$ is defined as follows:

(i) $S$ is the set of all interpretations $s$ of $F$ such that, for every static law (3) $s$ satisfies $F$ if $s$ satisfies $G$,

(ii) $V(P, s) = s(P)$, i.e. identify $s$ with $V(P, s)$,

(iii) $R$ is the set of all triples $(s, A, s')$, $A \subseteq E$, such that $s'$ is the only interpretation of $F$ which satisfies the heads of all

- static laws (3) in $D$ for which $s'$ satisfies $G$, and

- dynamic laws (4) in $D$ for which $s'$ satisfies $G$. \hfill (5)
• dynamic laws (4) in \( D \) for which \( s' \) satisfies \( G \) and \( s \cup A \) satisfies \( U \).

We focus on a fragment of the language \( C \) where the heads of the static and dynamic laws only consist of literals. This restriction on the laws reduces the cost of evaluating the transitions \( (s, A, s') \in R \) to polynomial time. Thus, we match the conditions on complexity from above. Furthermore, by well-known results on the complexity of action language \( C \) (Turner 2002; Eiter et al. 2004) all the results in Theorems 1-5 can be turned into completeness results already for this fragment.

**Defining a policy**

Let \( \bar{F} \) be the set of fluents that are relevant to the policy. The target language is defined explicitly via static laws using the fluents in \( \bar{F} \), denoted \( \mathcal{F}_G(\bar{F}) \), where a target is determined by the evaluation of these formulas in a state.

**Example 4.** An example of a target language for the running example uses causal laws from \( C \):

\[
\begin{align*}
\text{caused } & \text{target}(X_1, Y_1) \text{ if } \\
& \text{robotAt}(X, Y) \land \text{farthest}(X, Y, X_1, Y_1) \land \text{not personDetected}.
\end{align*}
\]

\[
\begin{align*}
\text{caused } & \text{personDetected} \text{ if } \text{personDetected}(X, Y).
\end{align*}
\]

\[
\begin{align*}
\text{caused } & \text{targetPerson}(X, Y) \text{ if } \text{personDetected}(X, Y).
\end{align*}
\]

\[
\begin{align*}
\text{caused } & \text{personFound} \text{ if } \text{personDetected}(X, Y) \land \text{robotAt}(X, Y).
\end{align*}
\]

where \( \mathcal{F}_{G_n}(\bar{F}) \) consists of all atoms \( \text{target}(X, Y) \) and \( \text{targetPerson}(X, Y) \) for \( 1 \leq X \leq n, 1 \leq Y \leq n \).

The target of a state according to the policy is computed through joint evaluation of these causal laws over the state with the known fluents about the agent’s location and the reachable points. Then, the outsourced planner may take as input the agent’s current location and the target location, to find a plan to reach the target.

**Equalized transition system**

The equalized transition system \((\hat{S}, \hat{V}, \hat{R})\) that describes the policy is defined as follows:

(i) \( \hat{S} \) is the set of all interpretations of \( \bar{F} \) such that, for every static law (3) \( \hat{s} \) satisfies \( F \) if \( \hat{s} \) satisfies \( G \).

(ii) \( \hat{V}(P, \hat{s}) = \hat{s}(P) \), where \( P \in \bar{F} \).

(iii) \( \hat{R} \subseteq \hat{S} \times \hat{S} \) is the set of all \( (\hat{s}, \hat{s}') \) such that

(a) for every \( s' \in \hat{s}' \) there is a trajectory from some \( s \in \hat{s} \) of the form \( s, A_1, s_1, \ldots, A_n, s' \) in the original transition system;

(b) for static laws \( f_1, f_2, \ldots, f_m \in \mathcal{F}_G(\bar{F}) \) for which \( \hat{s} \) satisfies the body, it holds that \( \hat{s}' \models g \) for some \( g \in \mathcal{M}(p_{f_1}, \ldots, p_{f_m}) \).

Notice that in the definition of the transition relation \( \hat{R} \) in (iii) there is no description of (a) how a trajectory is computed or (b) how a target is determined. This gives flexibility on the implementation of these components.

Other languages can be similarly used to describe the equalized transition system, as long as they are powerful enough to express the concepts from the previous section.

**Related Work**

There are works being conducted on the verification of GOLOG programs (Levesque et al. 1997), a family of high-level action programming languages defined on top of action theories expressed in the situation calculus. The method of verifying properties of non-terminal processes are sound, but do not have the guarantee of termination due to the verification problem being undecidable (De Giacomo, Ternovskaia, and Reiter 1997; Claßen and Lakemeyer 2008). By resorting to action formalisms based on description logic, decidability can be achieved (Baader and Zarrieß 2013).

Verifying temporal properties of dynamic systems in the context of data management is studied by (Calvanese et al. 2013) in the presence of description logic knowledge bases. However, target establishment and planning components are not considered in these works, and they do not address real life environment settings.

The logical framework for agent theory developed by Rao and Georgeff (1991) is based on beliefs, desires and intentions, in which agents are viewed as being rational and acting in accordance with their beliefs and goals. There are many different agent programming languages and platforms based on the BDI approach. Some works carried out on verifying properties of agents represented in these languages, such as (Bordini et al. 2006; Dennis et al. 2012). These approaches consider very complex architectures that even contain a plan library where plans are matched with the intentions or the agent’s state and manipulate the intentions.

**Synthesizing and Verifying Plans**

There have been various works on synthesizing plans via symbolic model checking techniques by Cimatti et al. (1998b; 1998a), Bertoli et al. (2006). These approaches are able to solve difficult planning problems like strong planning and strong cyclic planning.

Son and Baral (2001) extend the action description language by allowing sensing actions and allow to query conditional plans. These conditional plans are general plans that consist of sensing actions and conditional statements.

These works address a different problem then ours. When nondeterminism and partial observability are taken into account, finding a plan that satisfies the desired results in the environment is highly demanding. We consider a much less ambitious approach where given a behavior, we aim to check whether or not this behavior gives the desired results in the environment. However, our framework is capable emulating the plans found by these works.

**Execution Monitoring**

There are logic-based monitoring frameworks that monitor the plan execution and recover the plans in case of failure. The approaches that are studied are replanning (De Giacomo, Reiter, and Soutchanski 1998), backtracking to the point of failure and continuing from there (Soutchanski 2003), or diagnosing the failure and recovering from the failure situation (Fichtner, Großmann, and Thielerscher 2003; Eiter et al. 2007).

These works consider the execution of a given plan, while we consider a given reactive policy that determines targets and use (online) planning to reach these targets.
Conclusion and Future Work

In this paper, we described a high-level representation that models reactive behaviors, and integrates target development and online planning capabilities. Flexibility in these components does not bound one to only use action languages, but allows for the use of other formalizations as well. For future work, one could imagine targets to depend on further parameters or to incorporate learning from experience in the framework. It is also possible to use other plans, e.g. short conditional plans, in the planner component.

The long-term goal of this work is to check and verify properties of the reactive policies for action languages. In order to solve these problems practically, it is necessary to use techniques from model checking, such as abstraction, compositional reasoning and parameterization. Also, the use of temporal logic formulas is needed to express complex goals such as properties of the policies. Our main target is to work with action languages, and to incorporate their syntax and semantics with such model checking techniques. The general structure of our framework allows one to focus on action languages, and to investigate how to merge these techniques.

References


Introduction

“There is a point where in the mystery of existence contradictions meet; where movement is not all movement and stillness is not all stillness... where the idea and the form, the within and the without, are united; where infinite becomes finite, yet not losing its infinity.”

Rabindranath Tagore

In 1970 Edgar (Ted) F. Codd introduced a relational data model and two query languages: relational calculus and relational algebra. Relational calculus is what we usually call FO logic. The key contribution of Codd was to associate with a declarative specification language (FO logic), a procedural counterpart which is the relational algebra, that later was implemented by smart engineers and became a multi-billion dollar industry of relational database management systems (RDBMS).

A significant change has happened in the past decade. While at the low level everything boils down to SQL queries, interactions between “larger pieces” became increasingly important. Such “larger pieces” are business enterprises, web services, software, solvers in the world of declarative problem solving, etc. The emergence of large-scale interactions poses significant challenges to KR and database researchers. Among those challenges, the most urgent is scalability of proposed approaches to data management. The importance of scalability is justified by the Moore’s Law (1965), according to which the number of transistors in a dense integrated circuit doubles approximately every two years. Moore’s prediction proved accurate for several decades: capabilities of hardware grow exponentially.

We expect that, as hardware capabilities grow, people will be able to combine, as easily as relational tables in SQL queries, entities of a completely different magnitude:

• knowledge bases, e.g. representing the work of various enterprises, into complex business processes,  
• specifications of combinatorial optimization and search problems for solving new, more complex, problems.

To be more concrete, let us think that each one of these heterogeneous “larger pieces” is represented declaratively, e.g., by Logic Programming rules, ILP equations, SMT theories, FO knowledge bases, etc.

While database queries, expressed using Codd’s relational calculus, can be viewed as relations definable with respect to a structure (a database), declarative problem specifications can be understood as axiomatizations of classes of structures. The two notions (in italic) are defined in two consecutive chapters in the classic textbook on mathematical logic (Enderton 1972).

The main idea of our Algebra of Modular Systems is to lift Codd’s algebra from operations on relational tables to operations on classes of structures and to add recursion.¹

The operations of the algebra (to be defined formally) can be seen in this grammar for Modular Systems:

\[
E ::= \bot \mid M \mid Z \mid E \times E \mid E + E \mid E|\pi\delta E|\sigma\Theta E|\mu Z_j E,
\]

and they roughly correspond to conjunction, disjunction, negation, existential quantifier (\(\pi\delta\)) of FO logic. Selection (\(\sigma\Theta\)) restricts the module to structures satisfying the formula \(\Theta\). In addition, least fixed point is used. We now consider two examples, a simple and a more realistic.

Example 1. Let \(M_{HC}(V, X, Y)\) and \(M_{2Col}(V, X, Z, T)\) be atomic modules “computing” a Hamiltonian Circuit, and a 2-colouring. They can do it in different ways. For example, \(M_{HC}\) can use ASP, and \(M_{2Col}\) an imperative program or a

¹Here, we require a model-theoretic semantics, although various generalizations are possible.
human child with two pencils. Here, \( V \) is a relational variable of arity 1, \( X, Y \) are relational variables of arity 2, and the first module decides if \( Y \) forms a Hamiltonian Circuit (represented as a set of edges) in the graph given by vertex set \( V \) and edge set \( X \). Variable \( X \) of the second module has arity 2, and variables \( Z, T \) are unary; the module decides if unary relations \( Z, T \) specify a proper 2-colouring of the graph with edge set \( X \). The following algebraic expression determines a combination of 2-Colouring and Hamiltonian Circuit, that is whether or not there is a 2-colourable Hamiltonian Circuit:\(^2\)

\[
M_{2Col-HC}(V, X, Z, T) := 
\pi_{V,X,Z,T}((M_{HC}(V, X, Y) \times M_{2Col}(V, Y, Z, T))).
\]

Projection “keeps” \( V, X, Z, T \) and hides the interpretation of \( Y \) in \( M_{HC} \), since it is the same as \( Y \)’s in \( M_{2Col} \).

**Example 2.** This modular system can be used by a company that provides logistics services (arguments of atomic modules are omitted).

\[
M_{LSP} := \sigma_{B=B'}(M_K \times M_{TSP}).
\]

It decides how to pack goods and deliver them. It solves two NP-complete tasks interactively, – Multiple Knapsack \( M_K \) and Travelling Salesman Problem (module \( M_{TSP} \)). The system takes orders from customers (items to deliver, their profits, weights), and the capacity of trucks, decides how to pack items in trucks, and for each truck, solves a TSP problem. The feedback \( B' \) about solvability of TSP is sent back to \( M_K \). The two sub-problems, \( M_K \) and \( M_{TSP} \), are solved by different sub-divisions of the company (with their own business secrets) that cooperate towards the common goal. A solution to the compound module, \( M_{LSP} \), to be acceptable, must satisfy both sub-systems.

Most practical examples use simple combinations of modules, where only conjunctions, disjunctions, perhaps with projections and selections, are used. The university timetabling example (Järvisalo et al. 2009) and Examples 1, 2 are of this kind. Examples where a (declaratively specified) module calls itself recursively are much harder to come by. We will however show two natural applications. In the first one, combinatorial search problems on graphs are solved using their tree decompositions. In that application, the algebra allows us to write compact specifications of Dynamic Programming algorithms on tree decompositions. In the second application, our algebra is used to describe a recursive algorithm for solving quantified boolean formulas (QBFs). Note that here we are talking about recursion over a module, not over a predicate symbol, which happens frequently inside, say, ASP modules. Recursion over predicate symbols is not really needed in the algebra itself since free second-order variables are implicitly \( \exists \text{SO}-\)quantified, which gives at least as much expressive power as such a recursion can provide. Feedbacks from module to module, however, (as in Example 2) are very useful.

**Dynamic View** Problem solving often involves finding solutions for given inputs. Most combinatorial problems are of that form. The Logistics Service Provider in Example 2 has, e.g., customer requests as an input, and routes and packing solutions as outputs. One can have e.g. edges of a graph on the input to formula (1), and colours on the output. To capture this meaning, we introduce information flow to the algebra. Thus, we can now reason about actions or changes performed by the modules. Interpreting our algebra over a transition system gives rise to a modal temporal logic. There are several examples of connections between classical (“still” or “static”) and modal (“dynamic”) logics, and the notion of bisimulation plays an important role there. These are the so-called model-theoretic characterization theorems. For example, van Benthem shown that modal logic is a bisimulation-invariant fragment of first-order logic. Janin and Walukiewicz proved that the modal \( \mu \)-calculus \( L_{\mu} \) is the bisimulation-invariant fragment of monadic second-order logic MSO. (Abu Zaid, Grädel, and Jaax 2014) studied a related notion of bisimulation safety introduced by van Benthem. They introduced a new logic called Bisimulation Safe Fixed Point Logic (BSFP) that is more expressive than MSO. The BSFP logic is a very elegant formalism that uses both unary and binary fixed points. We will discuss it in more detail and use this logic in the context of modular systems with information flow. We investigate connections with other modal temporal logics. From the connection to BSFL, it follow that Propositional Dynamic Logic (PDL) is a fragment of our algebra with information flow.

We were quite surprised to notice that essentially the same constructions as in BSFP, but in axiomatic setting, appeared some 18 years earlier in the context of the situation calculus and GOLOG (Levesque et al. 1997; De Giacomo, Ternovskaja, and Reiter 1997). We discuss connections to the situation calculus in detail.

The duality of the “static” and “dynamic” views on modular systems opens new possibilities for developing algorithms to answer questions about the “static” algebra. For example, finding solutions to multi-language constraint problems can be done by solving Model Checking task for the modal temporal logic. This implies that both SAT-based and symbolic model checking techniques can be used.

**Preliminaries** A *vocabulary* (denoted, e.g. \( \tau, \sigma, \epsilon, \nu \)) is a finite sequence of non-logical (predicate and function) symbols, each with an associated arity. A \( \tau \)-structure, e.g. \( \mathcal{A} = (A; S^A_1, \ldots, S^A_n; f^A_1, \ldots, f^A_m; c^A_1, \ldots, c^A_m) \) is a domain \( A \) together with interpretations of predicate symbols, function and constants (0-ary functions) in \( \tau \). To simplify presentation, we view functions as a particular kind of relations and consider relational structures only. We use notations \( \text{vocab}(A), \text{vocab}(\phi), \text{vocab}(M) \) to denote vocabulary of structure \( A \), formula \( \phi \) and module \( M \), respectively, and we use \( B|_\sigma \) to mean structure \( B \) restricted to vocabulary \( \sigma \).

Symbol := means “denotes” or “is by definition”.

Least Fixed Point logic FO(LFP) is broadly used in CS and is described in several books. For a background on that logic we refer to, e.g. (Grädel et al. 2007).

Just as the basic modal logic is a fragment of first-order logic, the modal \( \mu \)-calculus \( L_{\mu} \) (that includes the well-known temporal logics CTL, CTL* and LTL) is a frag-
ment of FO(LFP). Moreover, the modal mu-calculus \( L_\mu \) is a bisimulation-invariant fragment of monadic second-order logic MSO. More details on fixed point logics and MSO can be found in (Dawar and Gurevich 2002; Libkin 2004; Grädel et al. 2007).

“Still” Algebra

We call this version of the algebra “still” to distinguish it from the version with information propagation below.\(^5\)

**Syntax** Let \( \tau \) be a fixed vocabulary. W.l.o.g. we assume that \( \tau \) is relational (it does not contain function symbols). Let \( \tau_M = \{ M_1, M_2, \ldots \} \) be a fixed vocabulary of atomic module symbols, \( \tau \cap \tau_M = \emptyset \). Atomic module symbols are of the form \( M_i(X_{i_1}, \ldots, X_{i_n}) \) (also written \( M_i(X) \)), where each \( X_j \) is a relational variable. Each \( X_j \) has an associated arity \( a_j \). The set \( \{ X_{i_1}, \ldots, X_{i_n} \} \) is called the variable vocabulary of \( M_i \) and is denoted \( \text{vocab}(M_i) \). We also allow relational constants from \( \tau \) to appear in the interpretation of symbols, provided they have a fixed arity. In this case, \( \text{vocab}(M_i) \) denotes the combined (variable and constant) vocabularies of \( M_i \). Let \( \tau_1, \tau_2, \ldots \) be a collection of module variables. Algebraic expressions for modules are built by the grammar:

\[
E ::= \bot[M_i]E \times E \lor E \lor E \land \pi E \lor [\sigma E]E \lor \mu Z_j.E. \quad (2)
\]

Modules in \( \tau_M \) are atomic. Modules that are not atomic are called compound. The operations (except \( \mu Z_j.E \)) are like in Codd’s relational algebra, but are of a higher order,\(^4\) and are defined on classes of structures rather than on relational tables. The three set-theoretic operations are union (\( \lor \)), intersection (\( \land \)), complementation (\( \pi \)). Projection (\( \pi E \)) is a family of unary operations, one for each \( \delta \). Each relational symbol (constant or variable) in \( \delta \) must appear in \( E \). The operation restricts each structure \( A \) of \( M \) to \( A|_\delta \) leaving the interpretation of other symbols open. Thus, it increases the number of models. The condition \( \Theta \) in selection \( \sigma E \) is an expression of the form \( L_1 \equiv L_2 \), where \( L_i \) is a relational variable or a relational constant from \( \tau \), or ‘\( R' \), where \( R \) is a relation (set of tuples of domain elements).\(^5\) Thus, we bring semantic elements into syntax in the latter case. Selection reduces the number of models.\(^6\)

**Semantics** To interpret algebraic expressions, we use valuations \( (v, V) \). Intuitively, \( v \) maps relational variables in \( \text{vocab}(M_i) \) to symbols from a relational vocabulary \( \tau \) so that the arities of the relational variables in \( \text{vocab}(M_i) \) match those of the corresponding symbols in \( \tau \). Function \( V \) is parameterised by \( v \) and provides a domain (which does not have to be finite), and interpretations of atomic modules \( M_i \), as explained below. Let \( C \) be the set of all \( \tau \)-structures over the domain fixed by \( V \). Valuation \( V \) maps each atomic module symbol \( M_i \) to a subset \( V[v, M_i] \) of \( C \) so that for any two \( \tau \)-structures \( A_1, A_2 \) which coincide on \( \text{vocab}(M_i) \), we have \( A_1 \in V[v, M_i] \) iff \( A_2 \in V[v, M_i] \). All of the above applies to module variables \( Z_j \) as well. In practice, \( V \) can, for example, associate one module symbol with stable models of an ASP program, another module symbol with models of an ILP encoding, yet another one with a set of databases used by a particular enterprise, etc.

**Remark 1.** Note that \( v \) resembles a “call by reference” in programming. Valuations \( V \) (parameterized with \( v \)) can be viewed as “oracles” or decision procedures associated with modules, and can be of arbitrary computational complexity.

The extensions \( [E]^{v,\nu}_V \) of algebraic expressions \( E \) are subsets of \( C \) defined as follows.

\[
\begin{align*}
[\bot]^{v,\nu}_V & := \emptyset, \\
[M_i]^{v,\nu}_V & := V(v, M_i) \text{ for some } v, \\
[\pi_j E]^{v,\nu}_V & := V(v, Z_j) \text{ for some } v, \\
[-E]^{v,\nu}_V & := C \setminus [E]^{v,\nu}_V, \\
[\sigma E]^{v,\nu}_V & := \{ A | \exists A' (A' \in [E]^{v,\nu}_V \text{ and } A|_\delta = A'|_\delta) \}, \\
[\mu Z_j E]^{v,\nu}_V & := \{ A | [E]^{v,\nu}_V \} \quad \text{and } L_1^\mu = L_2^\mu, \\
\end{align*}
\]

Here, \( | Z := E \) means a valuation that is exactly like \( V \) except \( Z \) is interpreted as \( E \).

Given a well-formed algebraic expression \( E \) defined by (2), we say that structure \( A \) satisfies \( E \) under valuation \( (v, V) \), notation \( A \models (v, v) E \) if \( A \in [E]^{v,\nu}_V \).

Some useful operations on modules are: Extending the vocabulary of \( E \) to a bigger vocabulary: \( \pi_j (E \times T) \). Renaming \( P \) to \( Q \) in \( E: \pi_{\text{vocab}(E)} \cup (P) \sigma P = Q (E \times T) \). Difference: \( E_1 - E_2 = E_1 \times (-E_2) \). Universal module: \( \top = \bot \). Also, as we would expect, \( E_1 \times E_2 = -(E_1 \times E_2) \).

**Remark 2.** Note that while individual modules are already capable of solving optimization tasks (the optimum value can be given as an output in one of the arguments), the least fixed point construct can generate the least value over a collection of modules combined in an algebraic expression.

**Proposition 1** (Logic Counterpart). The algebraic operations are equivalently representable in logic, where ‘\( +^* \)’ corresponds to disjunction, ‘\( -^* \)’ to negation, ‘\( \nu \)’ to second-order existential quantification over \( \tau \), ‘\( \sigma \)’ to conjunction with \( \Theta \), \( \mu Z.E \) to the least fixed point construct.

Thus, our formalism is FO(LFP) over modules that are of an arbitrary expressive power.

**Example 3.** Expression (1) for \( M_{2\text{Col}} \) is represented in logic as

\[
\exists Y [M_{1\text{HC}}(V, X, Y) \land M_{2\text{Col}}(V, Y, Z, T)].
\]

**Dynamic Programming on Tree Decompositions**

We use our algebra to specify high-level recursive control in solving combinatorially hard problems using dynamic programming and tree decomposition, along the lines of (Abseher et al. 2014; Charwat and Woltran 2015).
Definition 1. (Robertson and Seymour 1984) A tree decomposition of an undirected graph \( G = (V; E) \) is a pair \((T, X)\) where \( T = (V_T, E_T) \) is a tree and \( X: V_T \rightarrow 2^V \) assigns to every node \( V_T \) of the tree a set of vertices \( V \) from the original graph. The sets of vertices \( X = \{X_t\}_{t \in V_T} \) have to satisfy the following conditions: (1) \( \bigcup_{x \in V_T} X_t = V \). (2) \( \{x, y\} \in E \Rightarrow \exists t \in V_T: \{x, y\} \subseteq X_t \). (3) \( x \in X_v \land \ x \in X_w \land \therefore \exists \rho^m \in \text{path}(v, w) \Rightarrow x \in \rho^m \). \( X_t \) is also called the bag for the vertex \( t \in V_T \). The width \( w \) of the decomposition is \( \max_{x \in V_T} |X_t| - 1 \). The width \( k \) of a graph is the minimum width over all its tree decompositions.

It follows that every vertex of the graph is contained in some bag of the tree decomposition, adjacent vertices appear together in some bag, and nodes that contain the same vertex are connected. We denote an edge between vertices \( x, y \) by \( \{x, y\} \). For a decomposition node \( t \), we denote by \( E_t = \{\{x, y\} \in E \mid x, y \in X_t\} \) the edges of \( G \) induced by the vertices \( X_t \). We focus on a special type of tree decomposition.

Definition 2. A tree decomposition \( T = (V_T, E_T) \) is called normalized if every \( t \in V_T \) is of one of the following types: (1) Leaf node: it has no children nodes. (2) Introduction node: it has exactly one child node \( t' \) with \( X_{t'} \subset X_t \land |X_{t'}| = |X_t| - 1 \). (3) Removal node: it has exactly one parent node \( t' \) with \( X_{t'} \subset X_t \land |X_{t'}| = |X_t| + 1 \). (4) Join node: it has exactly two child nodes \( t' \) with \( X_{t'} \land X_t \). The problem is represented as \( M_{TD}(V, E, X_t, U_t, E_{U_t}, X_{U_t}, L_t) \) and 3-Colouring \( M_{Col}(X_t, E_t, R, B, G) \).

Example 4. Modules used: tree decomposition \( M_{TD}(V, E, X_t, U_t, E_{U_t}, X_{U_t}, L_t) \) and 3-Colouring \( M_{Col}(X_t, E_t, R, B, G) \). Intuitively, \((V, E)\) is the original graph, \((X_t, E_t)\) is the current bag with its edges, \(X_{U_t}, X_{U_t}'\) are its children, \(U_t\) is true on the vertex that has been introduced or removed, and \(E_{U_t}\) on the edges that lead to such vertices, and \(L_t\) specifies the label in \(\{l, j, r, i\}\) of a particular node in tree decomposition. In \(M_{Col}(X_t, E_t)\) the graph which we colour with \(R, B, G\). The problem is represented as \(M_{TD}(V, E, X_t, U_t, E_{U_t}, X_{U_t}, X_{U_t}')/\mu Z.\Psi(Z, M_{TD}, M_{Col})\), where \(Z\) is a module variable of the form \(Z(V, E, X_t, E_t, R, B, G)\) over which recursive iteration is performed. \(\mu Z.\Psi(Z, M_{TD}, M_{Col})\) specifies dynamic programming algorithm by recursion over tree decomposition, with \(\Psi(Z, M_{TD}, M_{Col}) := \phi_l \lor \phi_j \lor \phi_r \lor \phi_i\).

Base case, leaf:

\[
\phi_l := \sigma_{(x,t)}(x) \land X_{U_t} = 1 \land X_{U_t}' = 1 \land (t, l) \in E_{U_t}
\]

We use \(\perp\) for predicate constant “false”. The selection above requires that the label of the leaf \(l\) there are no child nodes (bags) \((X_t, l) \equiv \perp\land X_{U_t}' \equiv \perp\), and there is no introduced/removed vertex \(v\) in the bag \(X_t (U_t = \perp)\). Module \(M_{Col}\) performs 3-Colouring of the bag \(X_t\) with edges \(E_t\), where the tree decomposition is \(M_{TD}\).

Introduced vertex case:

\[
\phi_l := \sigma_{(x,t)}(x) \land E_{U_t}(X_{U_t} = 1) \land \neg Z(V, E, X_t, E_t, R, B, G) \land M_{Col}(X_t, E_t, R, B, G) \land M_{TD}(V, E, X_t, U_t, E_{U_t}, X_{U_t}, X_{U_t}', L_t)
\]

Removed vertex case:

\[
\phi_r := \exists R \exists B \exists G[\sigma_{(r,x,t)}(x) \land \neg E_{U_t}(X_{U_t} = 1) \land \neg Z(V, E, X_t, E_t, R, B, G) \land M_{TD}(V, E, X_t, U_t, E_{U_t}, X_{U_t}, X_{U_t}', L_t)]
\]

Join vertex case:

\[
\phi_j := \sigma_{(j,x,t)}(x) \land Z(V, E, X_t, E_t, R, B, G) \land Z(V, E, X_t, E_t, R, B, G) \land M_{TD}(V, E, X_t, U_t, E_{U_t}, X_{U_t}, X_{U_t}', L_t)
\]

Dynamic programming algorithms for different problems on tree decompositions of graphs (e.g. from Charwat and Woltran 2015) can be formulated in our algebra in a similar manner.

Algebra with Information Flow

Modules that have inputs and outputs are very common. Many software programs and hardware devices are of that form. In the Logistics Service Provider (Example 2), e.g. users’ requests could be on the input, and truck routes and packing solutions on the output. In this section, we add information propagation to the algebra, so that modules become binary higher-order input-output relations. This version of the algebra may be called “dynamic”. In algebraic expressions and corresponding logic formulas, we underline designated input symbols, i.e., those in \(\sigma_M\). Output symbols are free (are not quantified). For example:

\[
\exists Y[M_{HC}(V, X, Y) \land M_{Col}(V, Y, Z, T)]
\]

The quantified symbol \(Y\) is not visible from the outside. The output vocabulary of this compound modular system is \(\varepsilon = Z, T\) (for the two colours), the input vocabulary is \(\sigma = V, X\) (for the vertices and edges). In fact any direction of information propagation can be specified, e.g. from colours to graphs in 2-Colouring.

Fixing an input and an output vocabulary in some modules allows us to talk about the model expansion (MX) task (Mitchell and Ternovska 2005). In this task, a given structure, which might have an empty vocabulary, is expanded with interpretations of new vocabulary symbols to satisfy a specification. Complexity-wise, MX lies in-between model checking (full structure is given) and satisfiability (no structure is given). The task generalizes to the formalism of Modular Systems.

Model Expansion (MX) Task

Given: \(B|_\alpha\) and algebraic expression \(\alpha\) with input symbols \(\sigma\). Find: \(B\) such that \(B\) satisfies \(\alpha\). Structure \(B\) expands structure \(B|_\alpha\) and is called a solution of modular system \(\alpha\) for a particular input \(B|_\alpha\).

Thus, the algebra with information flow may be called “a logic of hybrid MX tasks”, and it will be interpreted over transition systems.\(^7\)

\(^7\)An interesting version is where only some inputs and outputs are specified, and some modules, even though are known to be binary, do not have input-output assignments. In this case, we obtain a collection of transition systems, one for each possible assignment, and both sceptical (under all input-output assignments) and brave reasoning (under some input-output assignment) can be studied.
Syntax  Fix a relational vocabulary \( \tau \) and a vocabulary of 
atomic module symbols \( \tau_M \) so that \( \tau \cap \tau_M = \emptyset \).
Let \( \tau_P \), where \( \tau_P \subseteq \tau_M \), be a vocabulary of 
atomic module symbols \( \{X_1, \ldots, X_n\} \), where inputs 
are underlined. We call them actions. For one module 
symbol \( M_i \), we can potentially have both a proposition and sev-
eral actions, depending on the choice of the inputs.
We define a calculus of binary relations as follows.

\[
\alpha ::= \bot \left| M_i \right| \tau_M \left| Z_j \right| \alpha \lor \alpha \circ \alpha \sigma \circ \sigma \alpha \sim \alpha \mu Z_j,\alpha \quad (4)
\]

Here, \( M_i \) are propositions, \( \mu M_a \) are actions, \( \sim \) is a unary 
operation which is a special kind of negation, as is used 
in modal temporal logics. Variables \( Z_j \) range over actions.
We require that \( Z_j \) occurs positively (under an even number of 
negations \( \sim \)) in \( \mu Z_j,\alpha \). Requirements on \( \pi_M \alpha \) and \( \sigma M \alpha \) 
are as before. The calculus is essentially BSFP (Abu Zaid, 
Grädel, and Jaax 2014), but with selection and projection 
added, and where we use modules for both unary and binary 
relations, which makes the semantics much more compi-
cated.

Semantics  Define a transition system \( T := (V; (M^T_{\alpha_M}, (M^T_{\alpha})^i)) \) (parameterized 
by valuation \( \langle \alpha, v \rangle \)) that has domain \( V \) which is the set of all 
\( \tau \)-structures over a fixed domain, and it interprets all actions 
\( M_i \) as subsets of \( V \times V \) denoted by \( \langle M^T_{\alpha_M}, (M^T_{\alpha})^i \rangle \), and 
all monadic propositions \( \tau \) by structures (now nodes in the 
transition graph) \( [M^T_{\alpha_M}, (M^T_{\alpha})^i] \subseteq V \) in which they are true.
Module variables \( Z_j \) that occur free in \( \alpha \) are interpreted 
as actions, i.e., as subsets of \( V \times V \). Their interpretations 
are denoted \( \tau \langle Z_j \rangle^T_{\alpha_M} \). We require that for each \( M^T_{\alpha_M} \) 
(respectively, \( \mu M_a \)), symbols in \( \text{vocab}(M^T_{\alpha_M}) \) (respectively, 
\( \text{vocab}(M_a) \), \( \text{vocab}(Z_j) \)) are interpreted by \( (T, V, v) \) in the 
same way for all structures (of states of \( T \)) \( B \) (respectively, 
pairs of structures) on which they are true. As before, 
we require that for any two \( \tau \)-structures \( A_1, A_2 \) which coincide 
on \( \text{vocab}(M_i) \), we have \( A_1 \in [V(v, M_i)] \) iff \( A_2 \in [V(v, M_i)] \).
We define the extension \( \alpha \langle \alpha \rangle^T_{\alpha_M} \) of formula \( \alpha \in \tau \) under 
valuation \( \langle \alpha, v \rangle \) inductively below. One can understand 
this definition as follows. First, \( v \) maps relational variables to 
symbols of the vocabulary \( \tau \), so that we can talk about \( \tau \)- 
structures (that compose concrete modules). Second, \( V \) 
provides an interpretation to each atomic module, which is 
a set of structures as before (i.e., a concrete module).
From now on, we view modules as either actions or propositions, 
depending on whether or not inputs-outputs are specified.
Actions are the transitions in the transition system \( T \), and 
propositions are labels of the states of \( T \). Sequential com-
position \( \circ \) and non-deterministic choice \( + \) act as expected, 
projection adds non-determinism, similarly to \(+\), and selection 
restricts the action to that where the interpretations of 
\( L_1 \) and \( L_2 \) are equal (there are three cases, when both \( L_1 \) 
and \( L_2 \) are inputs, both are outputs, and one is input, one 
is output), with interpretations as we would expect. The 
semantics of \( \mu Z_j,\alpha \) is exactly like that of the least fixed point 
operator in the modal mu-calculus \( \mu \). An interesting operation 
is negation \( \sim \). It acts like negation in modal logic, and 
its meaning will be more clear when we discuss the \( \sim \)-
two-sorted version of (4). The semantics of atomic actions will be 
more clear when we explain the connections to the situa-
tion calculus.

\[
\begin{align*}
\tau \langle \bot \rangle^T_{\alpha_M} &= \emptyset, \\
\tau \langle M_i \rangle^T_{\alpha_M} &= \{ (B, B) \in V^2 \times V^2 \mid B \in \mathcal{V}(v, M_i) \text{ for some } v \}, \\
\tau \langle \alpha_1 \circ \alpha_2 \rangle^T_{\alpha_M} &= \{ (\alpha_1 \circ \alpha_2)^T_{\alpha_M} \subseteq \{ (\alpha_1)^T_{\alpha_M} \cup (\alpha_2)^T_{\alpha_M} \}, \\
\tau \langle \alpha_1 \circ \alpha_2 \rangle^T_{\alpha_M} &= \{ (\alpha_1, B) \in V^2 \times V^2 \mid \\
\quad \mathcal{X}((A, C) \subseteq [\alpha_1]^T_{\alpha_M} \text{ and } (C, B) \subseteq [\alpha_2]^T_{\alpha_M}). \}
\end{align*}
\]

\[
\begin{align*}
\tau \langle \mu Z_j,\alpha \rangle^T_{\alpha_M} &= \{ R \subseteq V^2 \times V^2 \mid [\alpha] \tau \langle \emptyset \rangle^T_{\alpha_M} \subseteq R \}.
\end{align*}
\]

Example 5. To illustrate transitions using our examples, in 
(3), first \( M_{\text{DC}}(Y, X; Y, Z, T) \) makes transition by produc-
ally several Hamiltonian Circuits. The interpretation of the 
output vocabulary, \( \{ Y \} \) changes, everything else is trans-
ferred by inertia. Then each resulting structure is taken as 
in input to 2-Colouring, \( M_{\text{DCol}}(V, Y, Z, T) \), where edges in the 
cycle, \( Y \), are “fed” to the second argument of \( M_{\text{DCol}} \), 
although this is hidden from the outside observer by the ex-
istential quantifier in (3). The second module produces non-
deterministic transitions, one for each generated colouring.
Following (Abu Zaid, Grädel, and Jaax 2014), we define:

\[
D ::= \bot, \quad \pi_1(\alpha) ::= \sim \alpha.
\]

By these definitions,

\[
\begin{align*}
D \langle \alpha \rangle^T_{\alpha_M} &= \{ (B, B) \in V^2 \times V^2 \}, \\
\pi_1(\alpha) \langle \alpha \rangle^T_{\alpha_M} &= \{ (B, B) \in V^2 \times V^2 \mid \exists B' (B, B') \notin [\alpha]^T_{\alpha_M} \}
\end{align*}
\]

That is, \( D \) is the diagonal and \( \pi_1 \) abbreviates projection onto 
the first argument of the binary relation (onto all the inputs).
The latter operation identifies the states in \( V \) where there is 
an outgoing \( \pi_1\)-transition.

Two-Sorted Syntax  The grammar (4) for the algebra with 
information flow can be equivalently represented in a \( \sim \)-
two-sorted syntax, where expressions for state formulas \( \phi \) 
and processes \( \alpha \) are defined by mutual recursion.

\[
\begin{align*}
\alpha ::= D \mid [\emptyset] \mid M_a \mid Z_j \mid \alpha + \alpha \circ \alpha \sigma \circ \sigma \alpha \sim \alpha \mu Z_j,\alpha \quad (5)
\end{align*}
\]

Thus, we can write \( \langle \alpha \rangle \phi \) (respectively, \( [\alpha] \phi \)) to express that 
after some (respectively, all) executions of modular system 
\( \alpha \), property \( \phi \) holds. Notice that we have binary (for pro-
cesses) and unary (for state formulas) fixed points.

The two representations of the algebra (one-sorted and two-sorted) are equivalent. It follows from a theorem from (Abu Zaid, Grädel, and Jaax 2014) that holds also in our setting, where we have modules for actions and propositions.

**Theorem 1.** For every state formula \( \phi \) in two-sorted syntax (5) there is a formula \( \hat{\phi} \) in the minimal syntax (4) such that \( T, V, v \models \hat{\phi} \) iff \( T, \forall, (v,v) \models \pi_v \phi \), and for every action formula \( \alpha \) there is an equivalent formula \( \hat{\alpha} \) in the minimal syntax.

It also follows from (Abu Zaid, Grädel, and Jaax 2014) that the well-known Propositional Dynamic Logic (PDL) is a fragment of the logic introduced above.

**Proposition 2. The Propositional Dynamic Logic (PDL)**

\[
\alpha ::= \alpha_0 | \alpha + \alpha | \alpha \circ \alpha | \phi ? | \alpha^* \\
\phi ::= \mathit{M}_{\mathcal{T}} L \phi \therefore \neg \phi \therefore (\alpha) \phi
\]

is a fragment of (5).

**Proof.** It is sufficient to express \( \alpha^* \) since the other operations of PDL are a subset of (5). We have \( \alpha^* ::= \mu Z. (D + Z \circ \alpha) \).

☐

**Model Expansion and Model Checking Tasks for Modular Systems**

A very naive method to solve model expansion for a modular system \( \alpha \) would be to guess a structure expanding the input, and to check if it satisfies the algebraic expression. However, one can also develop an algorithm that identifies the set of all states \( S \subseteq V \) in the transition system where an algebraic expression holds. Such an algorithm can be developed for a fragment of the calculus that corresponds to the mu-calculus \( L_\mu \). The states in \( S \) will contain all expansions, for all instances. Then one can check whether a particular instance structure is in that set. A basic way to obtain \( S \) is by labelling the states of \( T \) by sub-expressions of \( \alpha \) that hold in those states, going bottom-up on the structure of \( \alpha \). A better way is to use Binary Decision Diagrams (BDDs) and perform this labelling symbolically, as is standard in symbolic model checking.

**Specification of Solving Quantified Boolean Formulas (QBFs) as an Algebraic Expression**

In this section, we consider quantified boolean formulas, such as, for example, \( \forall x \exists y (x \cdot y + x \cdot z) \). We will demonstrate an algebraic expression with a fixed point that encodes an algorithm for evaluation of such QBFs. We also connect such an evaluation with symbolic model checking of a mu-calculus formula.

**Example 6.** We assume that the QBF formulas that are used in the input are well-formed, all negations are pushed inwards to appear in front of boolean atoms only. The “smallest” formulas appearing in the parse tree are propositional formulas that occur just after inner-most quantifier. We also assume that all variables are bound by either \( \exists \) or \( \forall \).

**Modules used:** parse tree of a QBF formula: \( M_{\mathcal{PT}}(S, X_0, X_1, X_{t'}, X_{t''}, L_1, Q_1) \) and \( \mathsf{SAT} \) \( M_{\mathcal{SAT}}(X_t, T_{in}, T_{out}, F_{in}, F_{out}) \). Intuitively, unary relation \( S \) specifies a set of symbols, those used in QBFs and a special symbol \( _s \), e.g. \( x, y, z, \ldots, (,) , \exists , \forall , _s \).

Recall that QBF variables are elements of the domain. In each structure, all variables appearing in the QBF are partitioned into the interpretations of \( T_{out}, F_{out} \), which stand for \textit{true} and \textit{false}, respectively. These two relations describe all possible satisfying assignments of each sub-formula.

\[
M_{\mathcal{SAT}}(X_t, T_{in}, F_{in}, T_{out}, F_{out}) \text{ generates all possible truth assignments of the sub-formula encoded by } X_t \text{ (the so-called ‘generate’ part of the overall algebraic expression encoding solving QBFs), and each application of the selection operator in the recursive computation limits those assignments (the so-called ‘constraint’ or ‘test’ part).}
\]

The interpretations of \( T_{in}, F_{in} \) each contain exactly one element in each structure, and are constructed as follows. For all variables, for all structures in the module \( M_{\mathcal{SAT}} \), if the interpretation of \( T_{out} \) contains that variable, then it appears in the interpretation of \( T_{in} \) in that structure, and the interpretation of \( F_{in} \) is empty. Symmetrically, for all variables, for all structures in the module \( M_{\mathcal{SAT}} \), if the interpretation of \( F_{out} \) contains that variable, then it appears in the interpretation of \( F_{in} \) in that structure, and the interpretation of \( T_{in} \) is empty.

The interpretations of \( T_{in}, F_{in} \) and those of \( T_{out}, F_{out} \) in each structure of \( M_{\mathcal{SAT}} \) are constructed so that the truth value of a particular QBF variable is not true and false in the same structure, as it should be in a truth assignment.

The problem is represented (in the one-sorted syntax) as

\[
\exists x_0 \exists x_{t'} \exists x_{t''} \exists Q_1 \exists T_{in} \exists F_{in} \exists T_{out} \exists F_{out} [ \\

\mathsf{MPT}(S, X_0, X_1, X_{t'}, X_{t''}, L_1, Q_1) \land \mu Z. \Psi(Z, M_{\mathcal{PT}}, M_{\mathcal{SAT}}),
\]

where \( Z \) is a module variable of the form \( Z(X_t, Q_1, T_{in}, F_{in}, T_{out}, F_{out}) \) over which recursive iteration is performed. Existentially quantified variables are not visible “from the outside”. Thus, we have \( S \) and \( X_0 \) on the input, and \( T_{out}, F_{out} \) on the output.

Expression \( \mu Z. \Psi(Z, M_{\mathcal{PT}}, M_{\mathcal{SAT}}) \) specifies an algorithm for evaluating QBFs by recursion over their parse trees, with \( \Psi(Z, M_{\mathcal{PT}}, M_{\mathcal{SAT}}) ::= \phi_1 \lor \exists \forall \lor \forall \phi_1 \lor \exists \phi_1 \lor \forall \phi_1 \).

**Base case, leaf:**

\[
\phi_1 ::= \mathsf{M}_{\mathcal{PT}}(S, X_0, X_1, X_{t'}, X_{t''}, L_1, Q_1)
\]

\[
M_{\mathcal{SAT}}(X_t, T_{in}, F_{in}, T_{out}, F_{out}) \land \mu Z. \Psi(Z, M_{\mathcal{PT}}, M_{\mathcal{SAT}}).
\]

We use \( \perp \) for predicate constant “false”. The selection above requires that the label of the leaf is \( _s \), there are no quantified variables \( (Q_1 \equiv ('_s') \text{ there are no child subformulas } (X_{t'} \equiv \perp \lor X_{t''} \equiv \perp)) \).

Module \( M_{\mathcal{SAT}} \) solves propositional satisfiability of the subformulas \( X_t \), where the parse tree is given by \( M_{\mathcal{PT}} \).

Case \( \forall \):

\[
\phi_1 ::= \mathsf{M}_{\mathcal{PT}}(S, X_0, X_1, X_{t'}, X_{t''}, L_1, Q_1) \land \mathsf{SAT}(X_{t'}, T_{in}, T_{out}, F_{in}, F_{out})
\]

\[
\land \mathsf{M}_{\mathcal{SAT}}(X_t, T_{in}, F_{in}, T_{out}, F_{out}) \land \mathsf{M}_{\mathcal{PT}}(S, X_0, X_1, X_{t'}, X_{t''}, L_1, Q_1) \land \mathsf{SAT}(X_{t'}, T_{in}, T_{out}, F_{in}, F_{out}) \land \mathsf{M}_{\mathcal{SAT}}(X_t, T_{in}, F_{in}, T_{out}, F_{out}).
\]

Case \( \exists \) is very similar, except disjunction is used instead of conjunction:

\[
\phi_1 ::= \mathsf{M}_{\mathcal{PT}}(S, X_0, X_1, X_{t'}, X_{t''}, L_1, Q_1) \land \mathsf{SAT}(X_{t'}, T_{in}, T_{out}, F_{in}, F_{out})
\]

\[
\lor \mathsf{M}_{\mathcal{SAT}}(X_t, T_{in}, F_{in}, T_{out}, F_{out}) \land \mathsf{M}_{\mathcal{PT}}(S, X_0, X_1, X_{t'}, X_{t''}, L_1, Q_1) \land \mathsf{SAT}(X_{t'}, T_{in}, T_{out}, F_{in}, F_{out}).
\]
Conjunction case (×):
\[
\phi_\times := \sigma_{(L_{\mu}, Z_{\times}(\gamma, x))} [\mu T ((\sum X_0, X_1, X_1', X_1', L_1, Q_1) \\
\wedge \exists Z (X_1', Q_1, T_0, F_{in}, T_{out}, F_{out}) \\
\wedge \exists Z (X_1', Q_1, T_0, F_{in}, T_{out}, F_{out})].
\]

Disjunction case (+):
\[
\phi_+: := \sigma_{(L_{\mu}, Z_{+}(\gamma, x))} [\mu T ((\sum X_0, X_1, X_1', X_1', L_1, Q_1) \\
\wedge \exists Z (X_1', Q_1, T_0, F_{in}, T_{out}, F_{out}) \\
\wedge \exists Z (X_1', Q_1, T_0, F_{in}, T_{out}, F_{out})].
\]

Temporal Model Checking to Solve QBFs
Since, in the dynamic version, we interpret algebraic expressions over transition systems, and the specification above (7) is a modal mu-calculus \( L_{\mu} \) formula, we can use temporal logic model checking to solve QBFs. Each state in the transition system is a structure over the entire vocabulary. The main difference is that instead of simple propositions that hold in states of the transition system, we have potentially computationally complex modules that have to be checked in those states. For example, in the case of our QBF example, we need to check whether leaf formulas hold in the states, that is we need to use the propositional satisfiability module \( M_{SAT} \).

If we want to use symbolic model checking, we need to represent subsets of the set of all states, i.e., subsets of the set of all structures as BDDs. This is possible since each structure (a state in the transition system) can be viewed as a set of ground atomic formulas. The transition relation in the QBF example is generated dynamically, through the binary module variable \( Z \), and it follows the subformula relation in the parse tree of the QBF on the input. A BDD for this relation can also be constructed, and the rest is standard.

Connection to the Situation Calculus and GOLOG
The Situation Calculus(see (Reiter 2001)) is a second order language that gives an axiomatic way to describe transition systems. All changes are the result of actions. A possible world history, which is a sequence of actions, is represented by a first order term called a situation. The constant \( S_0 \) is used to denote the initial situation. A binary function symbol \( do(\alpha, s) \) denotes the successor situation to \( s \) resulting from performing the action \( \alpha \). Predicate symbol \( Poss(\alpha, s) \) specifies conditions on actions being executable is a situation. Relations (functions) whose truth values vary from situation to situation are called \( fluent \). They are denoted by predicate (function) symbols taking a situation term as their last argument. Axiomatizations of a dynamic domain include: (1) Action precondition axioms, one for each primitive action. These characterize the relation \( Poss, \) and the preconditions for the performance of an action in a situation. (2) Successor state axioms, one for each fluent. These capture the causal laws of the domain, together with a solution to the frame problem. (3) Unique names axioms for the primitive actions, stating that different names for actions denote different actions. (4) Axioms describing the initial situation. All the axioms together specify a transition system.

Connection to the Situation Calculus
Atomic module actions in 4 are very similar to the actions of the situation calculus. Module \( M_a \) “looks” at the interpretations of the input symbols \( \sigma_{M_a} \) in the structure \( B_1 \) (current “situation”), and expands \( B_1[\sigma_{M_a}] \) to produce interpretation of \( \varepsilon_{M_a} \) “recorded” in the structure \( B_2 \) (one of the “\( M_a \)-successor “situations”). Interpretations of all other symbols, including those in \( M_a \), stay the same, and get transferred from \( B_1 \) to \( B_2 \) by inertia. This is similar to the frame axioms in the situation calculus as described in R. Reiter’s book (Reiter 2001).

Notice that, similarly to the situation calculus, inertia is applied to atomic modules only. Each structure in module \( M_a \) may be seen as composed from its \( \sigma_{M_a} \) part “checked” in the structure \( B_1 \), and its \( \varepsilon_{M_a} \) part “recorded” in the structure \( B_2 \). Notice also that because of inertia, successor structure \( B_2 \) contains information about both \( \sigma_{M_a} \) and \( \varepsilon_{M_a} \) part of a structure in \( M_a \). Thus, the union of all \( M_a \)-successor structures, when limited to \( vocab(M_a) \), is the entire module \( M_a \).

GOLOG (Levesque et al. 1997) is a situation calculus-based language for defining complex actions using user-specified primitive actions. It can be described as:
\[
\delta := a \mid \delta; \delta \mid \delta \mid \delta^* \mid \pi x.\delta \mid \phi^? \mid Z(t) \mid \text{Proc} Z(t)\delta,
\]
where \( a \) is a primitive action, \( \delta; \delta \) is a sequence, \( \delta|\delta \) is choice, \( \delta^* \) non-deterministic iteration, \( \pi x.\delta \) is non-deterministic choice of argument, \( \phi^? \) is test action, \( Z(t) \) is procedure call, and \( \text{Proc} Z(t)\delta \) is procedure description (that includes recursion). Constructs if \( \phi \) then \( \delta_1 \) else \( \delta_2 \) and while \( \phi \) do \( \delta \) are definable through the other operations. The semantics of procedures is given through a least fixed point construct, which is like the binary fixed point construct \( mu Z.\delta \). Since iteration \( \delta^* \) is definable, (8) is a subset of the operations in the first line of (5). Another interesting observation is that unary fixed point properties, as in the second line of (5), were already formulated, in the context of the situation calculus, in (De Giacomo, Ternovskaia, and Reiter 1997).

Related Work
Literature on modularity and combined problem solving is enormous, and we do not attempt to review it here. Instead, we discuss only the most relevant work. A large part of this paper studies Codd’s algebra in the context of Model Expansion task. These tasks are common in AI planning, scheduling, logistics, supply chain management, etc. Java programs, if they are of input-output type, can be viewed as model expansion tasks, regardless of what they do internally. ASP systems, e.g., Clasp (Gebser, Kaufmann, and Schaub 2012) mostly solve model expansion, and so do CP languages such as Essence (Frisch et al. 2008), as shown in (Mitchell and Ternovska 2008). Problems solved in ASP competitions are mostly in model expansion form. CSP in the traditional AI form (respectively, in the homomorphism form) is representable by model expansion where mappings to domain elements (respectively, homomorphism functions) are expansion functions. The notion of a module in (Tisharrofi and Ternovska 2011) is mathematically the same as in the current paper. A module
there is a class of structures. Information propagation there happens through equivalent vocabulary symbols. However, the set of operations in that paper is smaller than in our paper here (there is no recursion, only a particular kind of selection (Feedback) is used). Moreover, connections to Codd’s relational algebra or modal temporal logic are not established there.

The paper that originally inspired our Modular Systems framework is (Järvisalo et al. 2009), but we developed a model-theoretic approach and provided additional operations (in (Järvisalo et al. 2009), projections and sequential compositions are possible, but not the other operations used here (unions, selections, recursion, etc.). Compositions in (Lierler and Truszczynski 2015) are products only.

Unlike, e.g. dlvhex programs (Eiter et al. 2006), recursion in our algebra is over a module variable $Z_i$, not a predicate variable. Thus, the purely syntactic requirement of $Z_i$ to occur positively in $E$ is sufficient to ensure monotonicity.

Multi-Context Systems (MCSs) (Brewka and Eiter 2007) combine knowledge bases in arbitrary languages, under arbitrary semantics (that do not have to be model-theoretic) through rules of logic programming with negation as failure, which is a totally different philosophy of combinations.

Conclusion

By adding information flow, we uncovered complex choreographies in the stillness of the simple declarative language of Modular Systems. There are significant benefits in studying modal fragments of our algebra (i.e., algebra with information flow). Such fragments often possess good model-theoretic and algorithmic properties. The main practical implication of the duality is that traditional “dynamic” techniques such as situation calculus, automata theory, temporal logic model checking etc. can be used to answer questions about the “static” formalism. Model checking for the dynamic system, for an important fragment of the logic, can be used to solve model expansion for the “still” version. Thus, the dynamic view opens up possibilities of new algorithms.

References


AGM-Style Revision of Beliefs and Intentions from a Database Perspective
(Preliminary Version)

Marc van Zee and Dragan Doder
Computer Science and Communications,
University of Luxembourg, University of Luxembourg

Abstract
We introduce a logic for temporal beliefs and intentions based on Shoham’s database perspective. We separate strong beliefs from weak beliefs. Strong beliefs are independent from intentions, while weak beliefs are obtained by adding intentions to strong beliefs and everything that follows from that. We formalize coherence conditions on strong beliefs and intentions. We provide AGM-style postulates for the revision of strong beliefs and intentions. We show in a representation theorem that a revision operator satisfying our postulates can be represented by a preorder on interpretations of the beliefs, together with a selection function for the intentions.

1 Introduction
Recently there has been an increase in articles studying the dynamics of intentions in logic [Ditmarsch et al., 2011; Icard et al., 2010; Lorini and Herzig, 2008; van der Hoek et al., 2007; Lorini et al., 2009; Grant et al., 2010]. Most of those papers take as a starting point the logical frameworks derived from Cohen and Levesque [1991], which in turn formalize Bratman’s [1987] planning theory of intention. In this paper, we take a different starting point, and study the revision of intentions from a database perspective [Shoham, 2009]. The database perspective consists of a planner, a belief database and an intention database. Shoham [2016] describes it as “(...) a generalization of the AGM scheme for belief revision, (...). In the AGM framework, the intelligent database is responsible for storing the planner’s beliefs and ensuring their consistency. In the enriched framework, there are two databases, one for beliefs and one for intentions, which are responsible for maintaining not only their individual consistency but also their mutual consistency.” (p.48) Shoham further developed these ideas with Jacob Banks, one of his PhD students, and behavioral economist Dan Ariely in the intelligent calendar application Timeful, which attracted over $6.8 million in funding and was acquired by Google in 2015¹. Shoham [2016] says himself: “The point of the story is there is a direct link between the original journal paper and the ultimate success of the company.” (p.47) Thus, it seems clear that his philosophical proposal has lead to some success on the practical side. In this paper, we investigate whether his proposal can lead to interesting theoretical insights as well. More specifically, the aim of this paper is to develop a suitable formal theory for the belief and intention database in the database perspective, and to study belief and intention revision for this theory. Following Shoham’s proposal, our methodology is to generalize AGM revision [Alchourron et al., 1985] for temporal beliefs and intentions in terms of a representation theorem.

In the area of intention revision and reconsideration, Grant et al. [2010] combine intention revision with AGM-like postulates. There have also been a number of contributions applying AGM-style revision to action logics [Shapiro et al., 2011; Jin and Thiel, 2004; Scherl, 2005; Scherl and Levesque, 2003; Bonanno, 2007]. However, these proposals only characterize revision using a set of postulates, without proving representation theorems. There are also approaches that focus on the semantical level by postulating revision on a Kripke model [Baral and Zhang, 2005]. Both Icard et al. [2010] and van Zee et al. [2015b] develop a logic based on Shoham’s database perspective and prove representation theorems, but we will analyze the shortcomings of their proposals in the next section.

The three main technical results of this paper are as follows: First we review two recent formalisations based on Shoham’s database perspective, namely the IPS framework [Icard et al., 2010] and the PAL framework [van Zee et al., 2015a], and we discuss the shortcomings of these logics. Secondly, we extend PAL in order to define a coherence condition on strong beliefs and intentions, and we separate strong, or intention-independent, beliefs from weak beliefs. Lastly, we characterize revision of beliefs and intentions through AGM-style postulates and we prove a representation theorem relating the postulates for revision to an ordering among interpretations and a selection function that accommodates new intentions while restoring coherence. The three next sections of the paper are in line with these three results.

¹http://venturebeat.com/2015/05/04/google-acquires-scheduling-app-timeful-and-plans-to-integrate-it-into-google-apps/
2 Preliminaries: The Database Perspective

We review Shoham’s database perspective, and we discuss the main limitations of two recent formalizations. For the first formalization (IPS), the separation between strong and weak beliefs is problematic, and the representation theorem only holds for a specific type of beliefs. For the second formalization (PAL), intention revision is missing.

2.1 Shoham’s Database Perspective

Shoham’s database perspective contains a planner (e.g., a STRIPS-like planner) that is itself engaged in some form of practical reasoning. In the course of planning, it may add actions to be taken at various times in the future to an intention database and add observations to a belief database. The intentions are future-directed intentions of the form \((a,t)\), meaning that action \(a\) will be executed at time \(t\). The beliefs are also time-indexed, and are of the form \(p_t\), meaning that \(p\) is true at time \(t\). Shoham treats the planner as a “black box”: It provides the databases with input but its internal workings are unknown. Shoham proposes informal revision procedures for beliefs and intentions based on the following coherence conditions:

1. If two intended actions immediately follow one another, the earlier cannot have postconditions that are inconsistent with the preconditions of the latter.
2. If you intend to take an action you cannot believe that its preconditions do not hold.
3. If you intend to take an action, you believe that its postconditions hold.

Note that requirement 2 and 3 describe an asymmetry between pre- and postconditions: The postconditions are believed to be true after an intended action, but the preconditions may not. Therefore, we might think of the requirements as one of “optimistic” beliefs. According to Shoham [2009]: “It is a good fit with how planners operate. Adopting an optimistic stance, they feel free to add intended actions so long as they are consistent with current beliefs.” (p.7)

2.2 Icard et al. (IPS)

Icard et al. [2010] develop a “formal semantical model to capture action, belief and intention, based on the “database perspective”” (p.1). They assume a set of atomic sentences \(\text{Prop}=\{p,q,r,\ldots\}\) and deterministic primitive actions \(\text{Act}=\{a,b,c,\ldots\}\). Entries in the belief database are represented by a language generated from:

\[ \varphi := p_t \mid \text{pre}(a)_t \mid \text{post}(a)_t \mid \text{Do}(a)_t \mid \Box \varphi \mid \varphi \land \varphi \mid \neg \varphi \]

with \(p_t \in \text{Prop}, a \in \text{Act}, \text{and } t \in \mathbb{Z}\). \(p_t\) means that \(p\) is true at time \(t\), \(\text{Do}(a)_t\) means that the agent does action \(a\) at time \(t\), and \(\text{pre}(a)_t\) and \(\text{post}(a)_t\) represent respectively the preconditions and postcondition of action \(a\) at time \(t\).

Icard et al. use a semantics of appropriate paths. They define \(P = \mathcal{P}\{\text{Prop} \cup \{\text{pre}(a), \text{post}(a) : a \in \text{Act}\}\}\), and a path

\[ \pi: \mathbb{Z} \rightarrow (P \times \text{Act}) \]

as a mapping from a time point to a set of proposition-like formulas true at that time (denoted \(\pi(t)\) and the next action \(a\) on the path (denoted \(\pi(t+1)\)). They define an equivalence relation \(\pi \sim \pi'\), which means that \(\pi\) and \(\pi'\) represent the same situation up to \(t\). Using this, they propose a notion of appropriateness:

**Definition 1** (Appropriate Set of Paths). A set of paths \(\Pi\) is appropriate iff for all \(p \in \Pi:\)

- If \(\pi(t)_2 = a\), then \(\text{post}(a) \in \pi(t+1)\).
- If \(\text{pre}(a) \in \pi(t)_1\), then there exists \(\pi' \sim \pi\) s.t. \(\pi'(t)_2 = a\).

The truth definition \(\models_{\Pi}\) is defined relative to an appropriate set of paths \(\Pi\), and the modality is defined as follows:

\[ \pi, t \models_{\Pi} \Box \varphi \] iff all \(\pi' \in \Pi\), if \(\pi \sim \pi'\) then \(\pi', t \models \varphi\).

A model for a formula is an appropriate set of paths. They introduce an intention database \(I = \{(a,t),\ldots\}\) as a set of action-time pairs \((a,t)\) and put the following coherence condition on their logic:

\[ \text{Cohere}^*(I) = \emptyset \bigwedge_{(a,t) \in I} \text{pre}(a)_t. \]

This captures the intuition that an agent considers it possible to carry out all intended actions. They state that a set of models is coherent if and only if there exists a model in which \(\text{Cohere}^*(I)\) is true. IPS distinguishes intention-contingent, or weak, beliefs from non-contingent, or strong, beliefs. Contingent beliefs \(B^*\) are obtained from a belief-intention database \((B,I)\) as follows:

\[ B^* = \text{CI}(B \cup \{\text{Do}(a)_t : (a,t) \in I\}). \]

In order to switch from belief bases to an appropriate set of paths, Icard et al. introduce the functions \(\rho\) and \(\beta\): “Given a set of formulas \(B\), we can consider the set of paths on which all formulas of \(B\) hold at time \(0\), denoted \(\rho(B)\). Conversely, given a set of paths \(\Pi\), we let \(\beta(\Pi)\) be defined as the set of formulas valid at \(0\) in all paths in \(\Pi^*\)” (p.3)

The first issue with IPS is that the definition of non-contingent beliefs is problematic for coherence. The following example shows that non-contingent beliefs that are dependent on the actual path can lead to a coherent agent with inconsistent weak beliefs.

**Example 1.** Consider an IPS belief-intention base \((B,I)\) with the belief base \(B = \{p_1, \neg \text{do}(a_2) \land p_1\}\) and intention base \(I = \{(a,2)\}\). While \(B\) is consistent with \(\Box \text{pre}(a_2)_2\), \(B^*\) is inconsistent since \(\neg \Box (\text{do}(a_2) \land p_1) \land \text{do}(a_2)\) derives \(\neg p_1\), but this is inconsistent with the initial belief \(p_1\).

In Section 3 we will define our notion of coherence that together with a separation between strong and weak beliefs resolves this problem.

The second issue with IPS is that the definition of \(\rho\) is circular, and as a result it does not seem to be possible for all formulas of their logic. Consider the following example.

**Example 2.** Suppose \(B = \{\Box p_1 \lor \neg \Box p_1\}\). The set of paths \(\rho(B)\) contains all paths \(\pi\) for which \(\pi \models B\), i.e., \(\pi \models \Box p_1 \lor \neg \Box p_1\). Take such a path \(\pi\) arbitrary, so either \(\pi \models \Box p_1\) or \(\pi \models \neg \Box p_1\). Suppose \(\pi \models \Box p_1\). Then it follows that for all \(\pi' \sim \pi\) we have \(\pi' \models p_1\). On the other hand, if \(\pi \models \neg \Box p_1\), then it follows that for all \(\pi' \sim \pi\) we have \(\pi' \models \neg p_1\). So in both cases we end up with two different sets for \(\rho(B)\). In other words, the set of paths \(\rho(B)\) is not defined.
Essentially, according to IPS, $\rho(B) = \{ \pi \mid \pi \models \Pi B \}$. The set of appropriate paths for a belief base $B$ is thus constructed using $\models \Pi$. However, $\models \Pi$ is already defined relatively to some set of paths $\Pi$. Therefore, the definition of $\rho$ is circular. It seems that the $\rho$ function only works for belief bases containing no modalities (in other words, to construct a single path). We omit details for space constraints, but the construction of the canonical model in the proof of their representation theorem uses the $\rho$ function to switch from a belief base to a set of paths (see Proof Sketch in the Appendix of Icard et al. [2010]). Therefore, the representation theorem does not hold for all formulas of the logic, since it is not possible to apply the function $\rho$ to all beliefs.

Summarizing, we recognize two shortcomings of the IPS framework as a formal basis for the database perspective: The definition of contingent beliefs is problematic, and the representation theorem does not hold for belief bases containing modalities.

### 2.3 van Zee et al. (PAL)

van Zee et al. [2015b] develop Parameterized-time Action Logic (PAL) as an alternative to IPS, claiming that the IPS framework contains an unsound axiom and that their logic is noncompact [van Zee et al., 2015a]. They define $\text{Act} = \{a, b, c, \ldots\}$ as a finite set of deterministic primitive actions, and $\text{Prop} = \{p, q, r, \ldots\} \cup \{\text{pre}(a), \text{post}(a) \mid a \in \text{Act}\}$ as a finite set of propositions. They denote atomic propositions with $\chi$. PAL differs syntactically from IPS in that the $\Box$-modality is indexed by a time-point. Their language $\mathcal{L}_{\text{PAL}}$ is inductively defined by the following BNF grammar:

$$\varphi ::= \chi | \text{do}(a) | \Box \varphi | \varphi \land \varphi | \neg \varphi$$

PAL uses a CTL*-like tree semantics consisting of a tree $T = (S, R, v, \text{act})$ where $S$ is a set of states, $R$ is an accessibility relation that is serial, linearly ordered in the past and connected, and $v : S \rightarrow 2^\text{Prop}$ is a valuation function from states to sets of propositions, and $\text{act} : R \rightarrow \text{Act}$ is a function from accessibility relations to actions, such that actions are deterministic, i.e. if $\text{act}(s, s') = a \rightarrow \text{act}(s, s'')$, then $s'' = s'$. Similarly to Icard et al., PAL uses an equivalence relation $\sim$ on paths. Using this equivalence relations they define a model as a pair $(T, \pi)$ on which the same conditions hold as an IPS model. PAL formulas are evaluated in a model on a path. van Zee et al. axiomatize PAL and show that it is sound and strongly complete, i.e. $T \models \varphi$ iff $T = \models \varphi$. They characterize AGM belief revision in this logic and restrict time up to some $t$. Using these constraints, they are able to represent a belief set $B$ as a propositional formula $\psi$ such that $B = \{\varphi \mid \psi \models \varphi\}$ and they prove the Katsuno and Mendelzon (KM) [1991] and the Darwiche and Pearl (DP) [Darwiche and Pearl, 1997] representation theorems.

van Zee et al. only consider revision of PAL formulas and thus do not consider the problem of intention revision, and therefore also does not distinguish between strong and weak beliefs. However, PAL uses a standard branching time (CTL*-like [Reynolds, 2002]) semantics, and it therefore also does not suffer from the shortcomings that we addressed in the previous subsection. Therefore, we take PAL as a starting point and use the remainder of this paper to extend it so that we can study intention revision.

### 3 The Belief-Intention Database

We assume an intention database $I = \{\{a, t\}, \ldots\}$ consisting of time-indexed actions. We next define a coherence condition on beliefs and intentions in PAL, and we separate strong and weak beliefs.

#### 3.1 The Coherence Condition

We first demonstrate that $\text{Cohere}^*(I)$ of IPS (Section 2.2) is too permissive because it allows models in which intentions are not jointly executable.

**Example 3 (Running Example).** An agent is considering to attend the NMR workshop at time 0 and the IJCAI conference at time 1. Although it would like to attend both events, there is insufficient budget available. Consider a possible partial semantic model of this situation depicted in Figure 1, where the thick path represents the actual path. In the actual path, the agent believes it does nothing at time 0 and attends IJCAI at time 1. It also considers it possible to attend the NMR workshop at time 0 and do nothing at time 1 in an alternative path. However, it does not consider it to be possible to attend both events.

Suppose the agent of the running example has two intentions: $I = \{\langle nmr, 0\rangle, \langle ijc\text{ai}, 1\rangle\}$. Intuitively, the agent’s intentions do not cohere with its beliefs, because it believes it cannot execute them both due to insufficient budget. However, according to $\text{Cohe}^*(I)$ the agent is coherent because the preconditions of all intentions hold on some path (namely the current path). Thus, $\text{Cohe}^*(I)$ does not fulfill Shoham’s coherence condition 1 (Section 2.1). More specifically, the problem is that it is not possible to define the precondition of a set of actions in terms of preconditions of individual actions, because it cannot be ensured that all the intentions are fulfilled on the same path as well. Therefore, in order to formalize a coherence condition in PAL, we extend the language with preconditions of finite action sequences, which ensures that after executing the first action, the precondition for the remaining actions are still true. We modify the language, the definition of a model, the axiomatization, and we show that the new axiomatization is sound and strongly complete. We call the new logic PAL-P (Parameterized-time Action Logic with extended Preconditions).

**Definition 2 (PAL-P Language).** The language $\mathcal{L}$ is obtained from $\mathcal{L}_{\text{PAL}}$ by adding $\{\text{pre}(a, b, \ldots), t \mid \{a, b, \ldots\} \subseteq \text{Act}, t \in \mathbb{N}\}$ to the set of propositions. Moreover, $\text{Past}(t)$ consists of boolean combinations of $p_t, \text{pre}(a, b, \ldots), t, \Box \varphi$ and $\text{do}(a)_{t-1}$ where $t \leq t$ and $\varphi$ is some formula from $\mathcal{L}$.
We also extend the definition of a model accordingly.

**Definition 3 (PAL-P Model).** A model is a pair \((T, \pi)\) with \(T = (S, R, v, \text{act})\) such that for all \(\pi \in T\) the following holds:
1. If \(\text{act}(\pi) = a\), then \(\text{post}(a) \in v(\pi_{a+1})\).
2. If \(\text{pre}(a) \in v(\pi)\), then there is some \(\pi' \in T\) with \(\pi \sim_{\pi} \pi'\) and \(\text{act}(\pi') = a\).
3. If \(\text{pre}(\{a, b\}) \in v(\pi)\), then \(\text{pre}(\{a\}) \in v(\pi)\).
4. If \(\text{pre}(\{a, b, \ldots\}) \in v(\pi)\), then there is some \(\pi' \in T\) with \(\pi \sim_{\pi'} \pi'\), \(\text{act}(\pi') = a\), and \(\text{pre}(b, \ldots)_{\pi+1} \in v(\pi_{a+1})\).

We refer to models of PAL-P with \(M_1, M_2, \ldots\), we refer to sets of models with \(M_1, M_2, \ldots\), and we refer to the set of all models with \(M\).

**Definition 4.** The logic PAL-P consists of the all the axiom schemas and rules of PAL [van Zee et al., 2015b] (Def. 7), and the following two:

\[
\begin{align*}
\text{pre}(\{a, b\}) \rightarrow \text{pre}(\{a\}) & \quad (A11) \\
\text{pre}(\{a, b, \ldots\}) \land \text{do}(a) & \rightarrow \text{pre}(b, \ldots)_{t+1} \quad (A12)
\end{align*}
\]

The relation \(\sim\) is defined in the usual way with the restriction that necessitation can be applied to theorems only.

**Theorem 1 (Completeness Theorem).** The logic PAL-P is sound and strongly complete, i.e. \(T \vdash \phi\) iff \(T \models \phi\).

The proof of the theorem is a direct extension of the proof of van Zee et al. with the new axioms and the new conditions on a model.

Note that it is not directly possible in PAL-P to express preconditions for actions that do not occur directly after each other. In order to do so, we simply make a disjunction over all possible action combinations in the time points in between the actions. Thus, if for instance \(\text{Act} = \{a, b\}\) and \(I = \{(a, 1), (b, 3)\}\), then \(\text{Cohere}(I) = \psi_0 \lor \text{pre}(a, x, b) = \psi_0(\text{pre}(a, b, b)) \lor \text{pre}(a, b, b)\).

**Definition 5 (Coherence).** Given an intention database \(I = \{(b_1, t_1), \ldots, (b_n, t_n)\}\) with \(t_1 < \ldots < t_n\), let

\[
\text{Cohere}(I) = \psi_0 \lor \text{pre}(a_1, a_{t+1}, \ldots, a_n)_{t+1}.
\]

For a given set of models \(M\), we say that \((M, I)\) is coherent iff there exists some \(m \in M\) with \(m \models \text{Cohere}(I)\).

We now show that the new coherence conditions correctly specify that the agent of our running example is not coherent.

---

We provide the full proofs of all theorems and propositions in this paper in a separate technical report: http://www.dropbox.com/s/798g6klihuq3gdw/jica2016proofof.pdf?dl=1

---

3.2 Strong and Weak Beliefs

The idea behind strong beliefs is that they represent the agent’s ideas about what is inevitable, no matter how it would act in the world. Weak beliefs, on the other hand, represent the beliefs of an agent that are the consequence of its planned actions [Van der Hoek and Wooldridge, 2003]. Formally, we define strong beliefs at some time \(t\) as formulas that start either with \(\Box\) or \(\Diamond\). The set of all strong beliefs \(B_t\) in time \(t\) for \(\mathcal{P}_{\text{PAL}}\) is inductively defined by the following BNF grammar:

\[
\phi ::= \Box \psi \mid \psi \land \phi \mid \neg \phi,
\]

where \(\psi \in \mathcal{P}_{\text{PAL}}\) and \(t \in \mathbb{N}\). A strong belief set in \(t\) is \(B_t \subseteq B\).

In the remainder of this paper, we assume \(t = 0\) and we simply write \(B\) and \(\Box\) to abbreviate \(B_0\) and \(\Box_0\).

The weak beliefs \(\text{WB}(B, I)\) are obtained from the strong beliefs by adding beliefs that are contingent on the intentions of the agent. In other words, the agent weakly believes everything that it strongly believed and moreover that all intentions will be realised, and everything that follows from this:

\[
\text{WB}(B, I) = \{\text{do}(a), (a, t) \in I\}.
\]

**Example 5 (Continued).** Some example strong beliefs of the agent of Figure 1 are \(\psi_0(\text{do}(\text{nmr}_0)), \psi_0(\text{do}(\text{jcai}_1)), \text{and} \text{do}(\text{post}(\text{nop}_1))\). If the agent has the intention database \(I = \{(\text{jcai}_1)\}\), then \(\text{post}(\text{jcai}_2)\) is one of its weak beliefs.

4 Belief and Intention Revision

Postulates Following KM, we fix a way of representing a belief set \(B\) consisting of strong beliefs by a propositional formula \(\psi\) such that \(B = \{\phi \mid \psi \vdash \phi\}\). Since intentions and beliefs that have been added by a planner are naturally bounded up to some time point \(t\), we define a bounded revision function and we restrict the syntax and semantics of PAL-P up to a specific time point. As a consequence, it is then possible to obtain the single formula \(\psi\) for a set of strong beliefs \(B\) (Corollary 1). We first define some notation that we use in the rest of this paper.

**Definition 6.** An agent is a pair \((\psi, I)\) consisting of a belief formula \(\psi\) and an intention base \(I\). \(A\) denotes the set of all agents, \(B\) denotes the set of all strong beliefs, \(I\) denotes the set of all intentions, and \(\text{DB}\) denotes the set of all intention databases. We denote \(A, B, I, \text{DB}\) bounded up to \(t\) with respectively \(A^t, B^t, I^t, \text{DB}^t\). However, if the restriction is clear from context, we may omit the superscript notation.

We now define a bounded revision function \(\psi^+\), revising an agent \((\psi, I)\) with a tuple \((\phi, i)\) consisting of a strong belief \(\phi\) and an intention \(i\), denoted \((\psi, I) \psi^+ (\phi, i)\), where \(i\) is the maximal time point occurring in \(\psi, I, \phi, \psi, i\).
Definition 7 (Agent Revision Function). An Agent revision function \( \ast_t : \mathcal{A} \times (\mathcal{B} \times \mathcal{I}) \to \mathcal{A} \) maps an agent, a strong belief formula, and an intention— all bounded up to \( t \)— to an agent bounded up to \( t \) such that:

- \( (\psi, I) \ast_t (\phi, i) = (\psi', I') \)
- \( (\psi_2, I_2) \ast_t (\phi_2, i_2) = (\psi_2', I_2') \)

then following postulates hold:

1. \((P1)\) \( \psi' \) implies \( \phi \).
2. \((P2)\) If \( \psi \land \phi \) is satisfiable, then \( \psi' \equiv \psi \land \phi \).
3. \((P3)\) If \( \phi \) is satisfiable, then \( \psi' \) is also satisfiable.
4. \((P4)\) If \( \psi \equiv \psi_2 \) and \( \phi \equiv \phi_2 \), then \( \psi' \equiv \psi_2' \).
5. \((P5)\) If \( \psi \not\equiv \psi_2 \) and \( \phi \equiv \phi_2 \), then \( \psi' \land \phi' \) implies \( \psi_2' \).
6. \((P6)\) If \( \psi \equiv \psi_2 \), \( \phi \equiv \phi_\land \phi' \), and \( \psi' \land \phi' \) is satisfiable, then \( \psi_2' \) implies \( \psi' \land \phi' \).

Postulates (P1)-(P6) are simply the KM postulates in our setting, which are equivalent to the AGM postulates [Katsuno and Mendelzon, 1991]. They also state that the revision of strong beliefs does not depend on the intentions. Postulates (P7)-(P10) also appear in IPS. Postulate (P7) states that the outcome of a revision should be coherent. Postulate (P8) states that the new intention \( i \) take precedence over all other current intentions; if possible, it should be added, even if all current intentions have to be discarded. Postulate (P9) and (P10) together state that if it is possible to simply add the intention, then this is the only change that is made. Postulate (P11) states that if we revise with the same \( i \) but with a different belief, and we end up with the same belief in both cases, then we also end up with the same intentions. Finally, (P12) states that we do not discard intentions unnecessarily. This last postulate is comparable to the parsimony requirement introduced by Grant et al. [2010].

Representation Theorem We next characterize all revision schemes satisfying (P1)-(P12) in terms of minimal change with respect to an ordering among interpretations and a selection function accommodating new intentions while restoring coherence. We bound models of strong beliefs up to \( t \), which means that all the paths in the model are “cut off” at \( t \). This ensures finitely many non-equivalent formulas for some belief set \( B \). A \( t \)-bounded model \( M^t = (T^t, \pi^t) \) is a model containing a tree \( T \) in which all paths, including \( \pi \), have length \( t \). Strong beliefs are about possibility and necessity, and they are independent of a specific path. Therefore, if a single path in a tree is a model of a strong belief, then all paths in this tree are models of this strong belief. Formally, a set of models of a strong belief \( M_{SB} \) satisfies the following condition:

\[
\text{If } (T, \pi) \in M_{SB}, \text{ then } (T, \pi') \in M_{SB} \text{ for all } \pi' \subseteq T.
\]

A set of \( t \)-bounded models of a strong belief \( M^t_{SB} \) contains only \( t \) restricted models of a strong belief. We write \( M^t_{SB} \) to denote the set of all sets of \( t \)-bounded models of strong beliefs. We now show that we can represent a set of models of strong beliefs by a single formula.

Lemma 1. Let \( \text{Ext}(M^t_{SB}) \) be the set of all possible extensions of a set of bounded model of strong beliefs \( M^t_{SB} \) to models, i.e.

\[
\text{Ext}(M^t_{SB}) = \{ m \in \mathbb{M} \mid m^t \in M^t_{SB} \}.
\]

Given a set of \( t \)-bounded models of strong beliefs \( M^t_{SB} \), there exists a strong belief formula \( \varphi \) such that \( \text{Mod}(\varphi) = \text{Ext}(M^t_{SB}) \).

Corollary 1. Given a \( t \)-bounded strong belief set \( B \), there exists a formula \( \varphi \) such that \( B = \{ \varphi \mid \varphi \vdash \varphi \} \).

Proof Sketch. For a given belief set \( B \), we can show that there exists a set of \( t \)-bounded models of a strong belief \( M^t_{SB} \) s.t. \( \text{Ext}(M^t_{SB}) = \text{Mod}(B) \). If \( \varphi = \text{form}(M^t_{SB}) \), then \( \text{Mod}(\varphi) = \text{Mod}(B) \), and by the completeness theorem, \( B = \text{Cl}(\varphi) \).

Given an intention database \( I \), we define a selection function \( \gamma \) that tries to accommodate a new intention based on strong beliefs. The selection function specifies preferences on which intention an agent would like to keep in the presence of the new beliefs.

Definition 8 (Selection Function). Given an intention database \( I \), a selection function \( \gamma : M_{SB} \times \mathbb{I} \to \mathbb{I} \) maps a set of models of a strong belief and an intention to an updated intention database— all bounded up to \( t \)— such that if \( \gamma(I, \{i\}) = I' \), then:

1. \( (M^t, I') \) is coherent.
2. If \( (M^t, \{i\}) \) is coherent, then \( i \subseteq I' \).
3. If \( (M^t, I \cup \{i\}) \) is coherent, then \( I \cup \{i\} \subseteq I' \).
4. If \( I' \subseteq I \cup \{i\} \).
5. For all \( I' \) with \( I' \subseteq I \cup \{i\} \), \( \gamma(I, \{i\}) \) is not coherent.

The five conditions on the selection function are in direct correspondence with postulates (P7)-(P10), (P12) of the agent revision function \( \ast_t \). Note that postulate (P11) doesn’t have a corresponding condition in the definition above but is represented by the fact that the selection function takes the revised beliefs as input. That is, intention revision occurs after belief revision.

KM define a faithful assignment from a belief formula to a pre-order over models. Since we are also considering intentions, we extend this definition such that it also maps intentions databases to selection functions.

Definition 9 (Faithful assignment). A faithful assignment is a function that assigns to each strong belief formula \( \psi \in \mathbb{B} \) a total pre-order \( \preceq_{\psi} \) over \( \mathbb{M} \) and to each intention database \( I \in \mathbb{I} \) a selection function \( \gamma_I \) and satisfies the following conditions:

1. If \( m_1, m_2 \in \text{Mod}(\psi) \), then \( m_1 \preceq_{\psi} m_2 \) and \( m_2 \preceq_{\psi} m_1 \).
2. If \( m_1 \in \text{Mod}(\psi) \) and \( m_2 \not\in \text{Mod}(\psi) \), then \( m_1 \prec m_2 \).
3. If \( \psi \equiv \phi \), then \( \preceq_{\psi} = \preceq_{\phi} \).
4. If \( T = T^t \), then \( (T, \pi) \preceq_{\psi} (T^t, \pi_2) \) and \( (T^t, \pi_2) \preceq_{\psi} (T, \pi) \).
Conditions 1 to 3 on the faithful assignment are the same as those of KM. Condition 4 ensures that we do not distinguish between models in the total pre-order \( \leq' \), whose trees are the same up to time \( t \). This is essentially what is represented in the revision function by bounding the all input of the revision function \( \ast \), up to \( t \). Moreover, \( \leq' \) does not distinguish between models obtained by selecting different paths from the same tree. This corresponds to the fact that we are using strong belief formulas in the revision, which do not distinguish between different paths in the same tree as well.

**Theorem 2 (Representation Theorem).** An agent revision operator \( \ast \) satisfies postulates (P1)-(P12) iff there exists a faithful assignment that maps each \( \psi \) to a total pre-order \( \leq' \) and each \( I \) to a selection function \( \gamma_I \) such that if \( (\psi, I) \ast (\phi, I) = (\psi', I') \), then:

1. \( \text{Mod}(\psi') = \min(\text{Mod}(\phi), \leq') \)
2. \( I' = \gamma'_I(\text{Mod}(\psi'), I) \)

**Proof Sketch.** We only sketch the proof of \( \Rightarrow \): Suppose that some agent revision operator \( \ast \) satisfies postulates (P1)-(P12). Given models \( m_1 \) and \( m_2 \), let \( (\psi, \emptyset) \ast (\text{form}(m_1') \lor \text{form}(m_2') \land I) = (\psi', \emptyset) \). We define \( \leq' \) by \( m_1 \leq' m_2 \) iff \( m_1 \models \psi \) or \( m_1 \models \psi' \). We also define \( \gamma'_I \) by \( \gamma'_I(M_{SB}^I, I) = I' \), where \( (\text{form}(M_{SB}^I), I) \ast (\top, I) = (\psi, I') \) (note that \( \psi \equiv \text{form}(M_{SB}^I) \)).

Let us prove condition 4 of Definition 9. For \( m_1 = (T, \pi) \) and \( m_2 = (T_2, \pi_2) \), let \( \psi' \) be as above. Since \( \psi, \psi' \in \mathcal{B} \), and \( T_2 = T_2^\ast \), we have \( m_1 \models \psi \) iff \( m_2 \models \psi \) and \( m_1 \models \psi' \) iff \( m_2 \models \psi' \), so \( m_1 \leq' m_2 \) and \( m_2 \leq' m_1 \).

Following KM, one can show that conditions 1 to 3 from Definition 9 hold, and furthermore that \( \text{Mod}(\psi') = \min(\text{Mod}(\phi), \leq') \). We now prove \( I' = \gamma'_I(\text{Mod}(\psi'), I) \). By our definition of \( \gamma'_I \) we have that \( (\psi, I) \ast (\top, I) = (\psi_2, \gamma'_I(\text{Mod}(\psi'), I)) \) (recall that \( \psi \equiv \psi_2 \)). Since \( (\psi, I) \ast (\phi, I) = (\psi', I') \), by (P11) we obtain that \( I' = \gamma'_I(\text{Mod}(\psi'), I) \).

Using postulate (P7)-(P10) and (P12) we can prove that \( \gamma'_I \) is a selection function.

Finally, it turns out to be straightforward to formulate the DP postulates for iterated revision in our framework for the strong beliefs and to prove their representation theorem. Due to space constraints we have omitted the results, but they can be found in a separate technical report.\(^5\)

**5 Related Work**

Grant et al. [2010] develop AGM-style postulates for belief, intention, and goal revision. They provide a detailed analysis and propose different reconsideration strategies, but restrict themselves to a syntactic analysis. Much effort in combining AGM revision with action logics (e.g., the Event Calculus [Mueller, 2010], Temporal Action Logics [Kvarnström, 2005], extensions to the Fluent Calculus [Thielscher, 2001], and extensions to the Situation Calculus (see [Patkos, 2010, Ch.2] for an overview)) concentrates on extending these action theories to incorporate sensing or knowledge-producing actions. Shapiro et al. [2011] extend the Situation Calculus to reason about beliefs rather than knowledge by introducing a modality \( B \) and shows that both the AGM postulates and the DP postulates are satisfied in this framework. A similar approach concerning the Fluent Calculus has been formalized by Jin and Thielscher [2004], and is further developed by Scherl [2005] and Scherl and Levesque [2003] by taking into account the frame problem as well. However, none of these approaches prove representations theorems linking revision to a total pre-order on models. Baral and Zhang [2005] model belief updates on the basis of semantics of modal logic S5 and show that their knowledge update operator satisfies all the KM postulates. Bonanno [2007] combines temporal logic with AGM belief revision by extending a temporal logic with a belief operator and an information operator. Both these approaches do not take action or time into account and do not prove representation theorems. The concept of strong beliefs has been discussed extensively in the literature, for instance in the story of Little Nell [McDermott, 1982] or a paradox found in knowledge-based programs [Fagin et al., 1995] (see (van der Hoek et al. [2003] for a detailed discussion).

**6 Conclusion**

We develop a logical theory for reasoning about temporal beliefs and intentions based on Shoham’s database perspective. We propose postulates for revision of strong beliefs and intentions, and prove a representation theorem relating the postulates to our formal model. Icard et al. prove a comparable representation theorem, but we show in this paper that it does not hold in general. In their proof, they use a canonical model construction, while we proof our representation theorem using standard techniques from belief revision. It remains an open problem whether a canonical model construction is possible when proving the representation theorem of Icard et al. For future work, we aim to extend our formalism with goals, which seems a natural extension in order to allow the agent to, for instance, replace intentions instead of merely discarding them. This paves the road to develop a richer notion of intentions, such as that “intentions normally pose problems for the agents: the agent needs to determine a way of achieving them” [Cohen and Levesque, 1990]. Interestingly, adding goals to the formalism blurs the distinction between planner and databases. If the databases take over part of the planning, then well-known problems such as the frame problem become more stringent: Once a fact is established (for example, as a postcondition of an intention), it persists until it explicitly contradicts postconditions established by future intentions. Existing action logics (e.g., the Event Calculus or the Fluent Calculus) and database approaches (e.g., TMMS [Dean and McDermott, 1987]) have dealt with these problems in detail, so comparing and possibly enriching them with our formalism seems both useful and relevant future work.

\(^5\) http://www.dropbox.com/s/798g6ki0uqt3gds/iiccai2016prooﬁs.pdf?dl=1
References


A New Approach for Revising Logic Programs

Zhiqiang Zhuang1 James Delgrande2 Abhaya Nayak3 Abdul Sattar1
1 Institute for Integrated and Intelligent Systems, Griffith University, Australia
2 School of Integrated and Intelligent Systems, Griffith University, Australia
3 School of Integrated and Intelligent Systems, Griffith University, Australia

Abstract
Belief revision has been studied mainly with respect to background logics that are monotonic in character. In this paper we study belief revision when the underlying logic is non-monotonic instead—an inherently interesting problem that is under explored. In particular, we will focus on the revision of a body of beliefs that is represented as a logic program under the answer set semantics. While the new information is also similarly represented as a logic program. Our approach is driven by the observation that unlike in a monotonic setting where, when necessary, consistency in a revised body of beliefs is maintained by jettisoning some old beliefs, in a non-monotonic setting consistency can be restored by adding new beliefs as well. We will define a syntactic revision function and subsequently provide representation theorem for characterising it.

Introduction
The ability to change one’s beliefs when presented with new information is crucial for any intelligent agent. In the area of belief change, substantial effort has been made towards the understanding and realisation of this process. Traditionally, it is assumed that the agent’s reasoning is governed by a monotonic logic. For this reason, traditional belief change is inapplicable when the agent’s reasoning is non-monotonic. Our goal in this research program is to extend belief base (Hansson 1999) approaches in belief revision to non-monotonic setting. In this paper, we focus on disjunctive logic programs, as a well-studied and well-known approach to non-monotonic reasoning that also has efficient implementations.

Much, if not most, of our day-to-day reasoning involves non-monotonic reasoning. To illustrate issues that may arise, consider the following example. In a university, professors generally teach, unless they have an administrative appointment. Assume we know that John is a professor. Since most faculty do not have an administrative appointment, and there is no evidence that John does, we conclude that he teaches. This reasoning is a classical form of non-monotonic reasoning, namely using the closed world assumption. It can be represented by the following logic program under the answer set semantics.

\[
\text{Teach}(X) \leftarrow \text{Prof}(X), \text{not Admin}(X). \quad (1)
\]

\[
\text{Prof}(John) \leftarrow . \quad (2)
\]

The answer set \{\text{Prof}(John), \text{Teach}(John)\} for this logic program corresponds exactly to the facts we can conclude.

Suppose we receive information that John does not teach, which we can represent by the rule

\[
\leftarrow \text{Teach}(John). \quad (3)
\]

Now our beliefs about John are contradictory; and it is not surprising that the logic program consisting of rules (1) – (3) has no answer set. For us or any intelligent agent in this situation to function properly, we need a mechanism to resolve this inconsistency. This is a typical belief revision problem; however, the classical (AGM) approach can not be applied, as we are reasoning non-monotonically.

It is not hard to suggest possible causes of the inconsistency and to resolve it. It could be that some of our beliefs are wrong; perhaps professors with administrative duties may still need to do teaching or perhaps John is not a professor. Thus we can restore consistency by removing rule (1) or (2). Alternatively and perhaps more interestingly, it could be that assuming that John is not an administrative staff via the absence of evidence is too adventurous; that is he may indeed be an administrative staff member but we don’t know it. Thus we can also restore consistency by adding the missing evidence of John being an administrative staff member by

\[
\text{Admin}(John) \leftarrow . \quad (4)
\]

The second alternative highlights the distinction for belief revision in monotonic and non-monotonic settings. In the monotonic setting, an inconsistent body of knowledge will remain inconsistent no matter how much extra information is supplied. On the other hand, in the non-monotonic setting, inconsistency can be resolved by either removing old information, or adding new information, or both. Therefore, belief revision functions in a non-monotonic setting should allow a mixture of removal and addition of information for inconsistency-resolution. In this paper, we will define one such revision functions for disjunctive logic programs under the answer set semantics.
The revision function is called slp-revision and is a belief base revision which takes syntactic information into account. In revising a logic program \( P \) by \( Q \), an slp-revision function first obtains a logic program \( R \) that is consistent with \( Q \) and differs minimally from \( P \), and then combines \( R \) with \( Q \). For example, if \( P = \{ (1), (2) \} \) and \( Q = \{ (3) \} \), then \( R \) would be \( \{ (1) \} \) (i.e., resolving inconsistency by removing \( (2) \)); \( \{ (2) \} \) (i.e., resolving inconsistency by removing \( (1) \)); or \( \{ (1), (2), (4) \} \) (i.e., resolving inconsistency by adding \( (4) \)).

The next section gives logical preliminaries. The following one develops our approach to slp-revision in which we provide postulates, a semantic construction, and a representation result. This is followed by a comparison to other work, and a brief conclusion.

Preliminary Considerations

In this paper, we consider only fully grounded disjunctive logic programs. That is variables in program rules are replaced by the set of their ground instances. Thus a logic program (or program for short) here is a finite set of rules of the form:

\[
\text{a}_1; \ldots; \text{a}_m \leftarrow \text{b}_1, \ldots, \text{b}_n, \neg \text{c}_1, \ldots, \neg \text{c}_o
\]

where \( m, n, o \geq 0, m + n + o > 0 \), and \( \text{a}_i, \text{b}_j, \text{c}_k \in \text{A} \) for \( \text{A} \) a finite set of propositional atoms. Connective \( \neg \) is called default negation. We denote the set of all logic programs by \( P \). For each rule \( r \), let \( H(r) = \{ \text{a}_1, \ldots, \text{a}_m \}, B^+(r) = \{ \text{b}_1, \ldots, \text{b}_n \}, \) and \( B^-(r) = \{ \neg \text{c}_1, \ldots, \neg \text{c}_o \} \). The letters \( P \) and \( Q \) are used to denote a logic program throughout the paper.

An interpretation is represented by the subset of atoms in \( \text{A} \) that are true in the interpretation. A classical model of a program \( P \) is an interpretation in which all rules of \( P \) are true according to the standard definition of truth in propositional logic, and where default negation is treated as classical negation. The set of classical models of \( P \) is denoted as \( \text{Mod}(P) \). Given an interpretation \( Y \), we write \( Y \models P \) to mean \( Y \) is a classical model of \( P \). The reduce of a program \( P \) with respect to an interpretation \( Y \), denoted \( P^Y \), is the set of rules:

\[
\{ H(r) \leftarrow B^+(r) \mid r \in P, B^-(r) \cap Y = \emptyset \}.
\]

An answer set \( Y \) of \( P \) is a subset-minimal classical model of \( P^Y \). The set of all answer set of \( P \) is denoted as \( \text{AS}(P) \).

An SE interpretation (Turner 2003) is a pair \((X, Y)\) of interpretations such that \( X \subseteq Y \subseteq A \). The set of all SE interpretations (over \( A \)) is denoted \( \text{SE} \). The letters \( M \) and \( N \) are used to denote a set of SE interpretations throughout the paper. An SE interpretation is an SE model of a program \( P \) if \( Y \models P \) and \( X \models P^Y \). The set of all SE models of \( P \) is denoted as \( \text{SE}(P) \). SE models are proposed to capture strong equivalence (Lifschitz et al. 2001) between programs that is \( \text{SE}(P) = \text{SE}(Q) \) iff \( P \) and \( Q \) are strongly equivalent, thus they contain more informations than answer sets.

The following two properties of SE models (Turner 2003) are crucial to this paper:

1. \( Y \in \text{AS}(P) \) iff \( (Y, Y) \in \text{SE}(P) \) and there is no \( (X, Y) \in \text{SE}(P) \) such that \( X \subset Y \).
2. \( (Y, Y) \in \text{SE}(P) \) iff \( Y \in \text{Mod}(P) \).

So \( \text{SE}(P) \neq \emptyset \) iff \( \text{Mod}(P) \neq \emptyset \) but \( \text{SE}(P) \neq \emptyset \) does not imply \( \text{AS}(P) \neq \emptyset \). This gives rise to two notions of consistency.

Definition 1. \( P \) is consistent iff \( \text{AS}(P) \neq \emptyset \) and \( P \) is m-consistent \(^1\) iff \( \text{SE}(P) \neq \emptyset \).

It is clear from the SE model properties that consistency implies m-consistency; m-inconsistency implies inconsistency. In other words, a consistent program is m-consistent but not vice versa.

In subsequent sections, we will need to describe the difference between two logic programs. For this purpose, we use the symmetric difference operator \( \oplus \) which is defined as:

\[
X \oplus Y = (X \setminus Y) \cup (Y \setminus X)
\]

for any sets \( X \) and \( Y \).

SLP-Revision Functions

In this section, we give a syntax-based revision function \( * : P \times P \rightarrow P \) for revising one logic program by another. The function takes a logic program \( P \) called the original logic program and a logic program \( Q \) called the revising logic program, and returns another logic program \( P * Q \) called the revised logic program. Following AGM belief revision, we want to have \( Q \) contained in \( P * Q \) (i.e., \( Q \subseteq P * Q \)), \( P * Q \) is consistent whenever possible, and that as much of \( P \) as consistently possible is contained in \( P * Q \).

Clearly, a key issue in defining \( * \) is to deal with the possible inconsistency between \( Q \) and \( P \). As illustrated in the teaching example, one means of ensuring that \( P * Q \) is consistent is to remove a minimal set of beliefs from \( P \) so that adding \( Q \) to the result is consistent. Of course there may be more than one way to remove beliefs from \( P \). Following this intuition, we obtain all maximal subsets of \( P \) that are consistent with \( Q \), which we call the s-revision compatible programs of \( P \) with respect to \( Q \).

Definition 2. The set of s-revision compatible programs of \( P \) with respect to \( Q \), denoted \( P \downarrow Q \), is such that \( R \in P \downarrow Q \) iff:

1. \( R \subseteq P \).
2. \( R \cup Q \) is consistent, and
3. if \( R \subseteq R' \subseteq P \), then \( R' \cup Q \) is inconsistent.

The notion of s-revision compatible programs is not new, classical revision functions (Alchourrón et al. 1985; Hansson 1993) are based on more or less the same notion. The difference is that this notion alone is sufficient to capture the inconsistency-resolution strategy of classical belief revision, but there is more that one can do in non-monotonic belief revision.

In our non-monotonic setting, we are able to express assumptions (i.e., negation as failure) and to reason with them. Earlier, we assumed John is not an administrator, in the absence of evidence to the contrary. With this, we came to the conclusion that he has to teach. Consequently, if we learn

\(^1\)“m” stands for “monotonic” which indicates that the notion of m-consistency is based on a monotonic characterisation (i.e., SE models) for logic programs.
that John does not teach, as in our example, one way of re-
solving this inconsistency is by adding information so that our assumption does not hold. Following this intuition, we obtain all the minimal supersets of \( P \) that are consistent with \( Q \), which we call the \( s \)-expansion compatible program of \( P \) with respect to \( Q \).

**Definition 3.** The set of \( s \)-expansion compatible programs of \( P \) with respect to \( Q \), denoted \( P \uparrow Q \), is such that \( R \subseteq P \uparrow Q \) iff
1. \( P \subseteq R \),
2. \( R \cup Q \) is consistent, and
3. if \( P \subseteq R' \subseteq R \), then \( R' \cup Q \) is inconsistent.

Since the \( s \)-expansion and \( s \)-removal compatible programs are consistent with \( Q \) and are obtained by removing or adding minimal sets of rules from or to \( P \), the union of \( Q \) with any of these sets is consistent and comprises a least change made to \( P \) in order to achieve consistency. These programs clearly should be candidates for forming the revised logic program \( P \uparrow Q \); however, they do not form the set of all candidates. In particular, we can obtain a program that differs the least from \( P \) and is consistent with \( Q \) by removing some beliefs of \( P \) and at the same time adding some new beliefs to \( P \). Thus we consider all those logic programs that differ the least from \( P \) and are consistent with \( Q \); these are called the \( s \)-compatible programs of \( P \) with respect to \( Q \).

**Definition 4.** The set of \( s \)-compatible programs of \( P \) with respect to \( Q \), denoted \( P \uparrow Q \), is such that \( R \subseteq P \uparrow Q \) iff
1. \( P \cup Q \) is consistent and
2. if \( P \cup R' \subseteq P \cup Q \), then \( R' \cup Q \) is inconsistent.

For example, let \( P = \{a \leftarrow b, \text{not } c, b, e \leftarrow f, \text{not } g, f, \} \) and \( Q = \{a, \text{not } c, e, \} \). Then \( P \cup Q \) is inconsistent since \( a \) and \( e \) can be concluded from \( P \) but they contradict the rules of \( Q \). To resolve the inconsistency via making the least change to \( P \), we could remove \( b \leftarrow P \) (which eliminates the contradiction about \( a \)) and add \( q \leftarrow P \) (which eliminates the contradiction about \( e \)). The program thus obtained (i.e., \( P \uparrow \{b\} \cup \{g\} \)) is a \( s \)-compatible program in \( P \uparrow Q \).

It is obvious, but worth noting that the notion of \( s \)-compatible program subsumes those of \( s \)-removal and \( s \)-expansion compatible programs. In the above example, \( P \uparrow Q \) also contains \( \{b, f\} \) and \( P \cup \{c, g\} \), which are respectively an \( s \)-removal and an \( s \)-expansion compatible program of \( P \) with respect to \( Q \).

**Proposition 1.** \( (P \uparrow Q) \cup (P \downarrow Q) \subseteq P \uparrow Q \).

There are cases in which we cannot resolve inconsistency by only adding new beliefs which means the set of \( s \)-expansion compatible programs is empty. For example, if \( P = \{a\} \) and \( Q = \{\text{not } a\} \), then \( P \cup Q \) is inconsistent and we cannot restore consistency without removing \( a \leftarrow P \). In these cases, the inconsistency is due to contradictory facts that can be concluded without using any reasoning power beyond that of classical logic. Clearly, the inconsistency is of a monotonic nature, that is, in our terminology, \( m \)-inconsistency.

**Proposition 2.** If \( P \cup Q \) is \( m \)-inconsistent, then \( P \uparrow Q = \emptyset \).

So far, we have identified the candidates for forming \( P \star Q \). It remains to pick the “best” one. Such extralogical information is typically modelled by a selection function, which we do next.

**Definition 5.** A function \( \gamma \) is a selection function for \( P \) iff for any program \( Q \), \( \gamma(P \downarrow Q) \) returns a single element of \( P\downarrow Q \) whenever \( P\downarrow Q \) is non-empty; otherwise it returns \( P \).

The revised logic program \( P \star Q \) is then formed by combining \( Q \) with the \( s \)-compatible program picked by the selection function for \( P \). We call the function \( * \) defined in this way a \( slp \)-revision function for \( P \).

**Definition 6.** A function \( * \) is a \( slp \)-revision function for \( P \) iff
\[
P \star Q = \gamma(P \downarrow Q) \cup Q
\]
for any program \( Q \), where \( \gamma \) is a selection function for \( P \).

In classical belief revision, multiple candidates maybe chosen by a selection function, and their intersection is combined with the new belief to form the revision result. There, a selection function that picks out a single element is called a maxichoice function (Alchourrón et al. 1985). In classical logic, maxichoice selection functions leads to undesirable properties for belief set revision but not for belief base revision. In our non-monotonic setting, picking multiple candidates does not make sense, as intersection of \( s \)-compatible programs may not be consistent with the revising program. For example, let \( P = \{a \leftarrow \text{not } b, \text{not } c\} \) and \( Q = \{\text{not } a\} \). We can restore consistency of \( P \) with \( Q \) by, for instance, adding the rule \( b \leftarrow P \) which corresponds to the \( s \)-compatible program \( P \cup \{b\} \) or by adding the rule \( c \leftarrow P \) which corresponds to the \( s \)-compatible program \( P \cup \{c\} \).

However, the intersection of the two \( s \)-compatible programs is inconsistent with \( Q \).

We turn next to properties of \( slp \)-revision functions. Consider the following set of postulates where \( * : \mathcal{P} \times \mathcal{P} \to \mathcal{P} \) is a function.

\[
\text{(s*): } Q \subseteq P \star Q
\]
\[
\text{(s\&): } \text{If } P \text{ is m-consistent, then } P \star Q \text{ is consistent}
\]
\[
\text{(s\&m): } \text{If } P \text{ is m-consistent, then } P \star Q = P \cup Q
\]
\[
\text{(s\&r): } \text{If } P \neq \emptyset \text{ and } R \subseteq P \setminus (P \star Q), \text{ then } (P \star Q) \cup R \text{ is inconsistent}
\]
\[
\text{(s\&e): } \text{If } E \neq \emptyset \text{ and } E \subseteq (P \star Q) \setminus (P \cup Q), \text{ then } (P \star Q) \cup E \text{ is inconsistent}
\]
\[
\text{(s\&mr): } \text{If } R \neq \emptyset, \text{ and } R \subseteq P \setminus (P \star Q), \text{ then } (P \star Q) \cup (P \cup Q) \text{ is inconsistent}
\]
\[
\text{(s\&u): } \text{If } P \uparrow Q = P \downarrow R, \text{ then } P \star (P \star Q) = P \star (P \star R)
\]
\[
\text{(s\&s): } \text{(Success) states that a revision is always successful in incorporating the new beliefs. (s\&c): (Consistency) states that a revision ensures consistency of the revised logic program whenever possible. In the monotonic setting, a revision results in inconsistency only when the new beliefs are themselves inconsistent. This is not the case in the non-monotonic setting. For example, consider the revision of } P = \{a\} \text{ by } Q = \{b \leftarrow \text{not } b\}. \text{ Although } Q \text{ is inconsistent, we have } P \cup \{b\} \text{ as a } s \text{-compatible program of } P
\]

\begin{align*}
173
\end{align*}
with respect to \( Q \). Thus we can have \( P \cup \{b.\} \cup Q \) as the revised logic program, which contains \( Q \) and is consistent. Here, a revision results in inconsistency only when the revising logic program is m-inconsistent. In such a case, \((s+\psi)\) (Failure) states that the revision corresponds to the union of the original and revising logic program.

\((s+rr)\) (Removal Relevance) states that if some rules are removed from the original logic program for the revision, then adding them to the revised logic program results in inconsistency. It captures the intuition that nothing is removed unless its removal contributes to making the revised logic program consistent. \((s+er)\) (Expansion Relevance) states that if some new rules other than those in the revising logic program are added to the original logic program for the revision, then removing them from the revised logic program causes inconsistency. It captures the intuition that nothing is added unless adding it contributes to making the revised logic program consistent. \((s+mr)\) (Mixed Relevance) states that if some rules are removed from the original logic program and some new rules other than those in the revising logic program are added to the original logic program for the revision, then adding back the removed ones and removing the added ones result in inconsistency of the revised logic program. Its intuition is a mixture of the two above. Note that putting \((s+rr)\) and \((s+er)\) together does not guarantee \((s+mr)\), nor the reverse. In summary, these three postulates express the necessity of adding and/or removing certain belief for resolving inconsistency and hence to accomplish a revision. In classical belief revision, inconsistency can only be resolved by removing old beliefs; the necessity of removing particular beliefs is captured by the Relevance postulate (Hansson 1993).\(^2\) The three postulates are the counterparts of Relevance in our non-monotonic setting, and we need all three of them to deal respectively with addition, removal, and a mixture of addition and removal.

Finally, \((s+u)\) (Uniformity) states the condition under which two revising logic programs \( P, R \) trigger the same changes to the original logic program \( P \). That is the rules removed from \( P \) (i.e., \( P \setminus (P \ast Q) \)) and the rules added to \( P \) (i.e., \( (P \ast Q) \setminus (P \cup Q) \)) for accommodating \( Q \) are identical to those for accommodating \( R \). Certainly having \( Q \) and \( R \) be strongly equivalent (i.e., \( SE(Q) = SE(R) \)) is a sufficient condition. However, it is too strong a requirement. Suppose \( P = \{a \leftarrow a\} \), \( Q = \{a\} \), and \( R = \{a \leftarrow b, b\} \). Then the minimal change to \( P \) we have to made to accommodate \( Q \) and \( R \) are the same, that is we remove \( \leftarrow a \). However \( Q \) and \( R \) are not strongly equivalent, even though they incur the same change to \( P \). The essential point of this example is that instead of a global condition like strong equivalence, we need a condition that is local to the original logic program \( P \). Unfortunately, it seems there is no existing notion in the logic programming literature that captures this local condition. Thus we use our newly defined notion of s-compatible programs and come up with the local but more appropriate condition in \((s+u)\).

We can show that these postulates are sufficient to characterise all slp-revision functions.

**Theorem 1.** A function \( \ast \) is a slp-revision function iff it satisfies \((s+)\), \((s+c)\), \((s+f)\), \((s+rr)\), \((s+er)\), \((s+mr)\), and \((s+u)\).

**Comparisons with Existing Approaches**

There has been much work on belief revision for logic programs. The seminal work of Delgrande et al (2013b) generalises Satoh’s (1988) and Dalal’s (1988) revision operators to logic programs. Significantly, they bring SE model into the picture. They do not work with answer sets as a basis for revision, but rather they base their definitions directly on SE models. The work has inspired several other SE model approaches. Schwind and Inoue (2013) provide a constructive characterisation for the revision operators in (Delgrande et al. 2013b). Delgrande et al (2013a) adapt the model-based revision of Katsuno and Mendelzon (1992) to logic programs and provide a representation theorem. Finally, Binnewies et al (2015) provide a variant of partial meet revision and contraction for logic programs.

Firstly, the SE model approaches are essentially belief set revision whereas our slp-revision is a belief base one. Secondly and more importantly, these approaches assume a weaker notion of consistency, that is m-consistency. For this reason, some contradictions will not be dealt with in these approaches. For instance, the contradictory rule \( a \leftarrow \text{not } a \) is m-consistent thus is considered to be an acceptable state of belief. Also in our teaching example, as the program consisting of rules (1) – (3) is m-consistent, no attempt will be made to resolve the contradiction about John’s teaching duty by the SE model approaches. Therefore for application scenarios in which such contradictions can not be tolerant, our slp-revision function is clearly a better choice.

Apart from the SE model approaches, Krümpelmann and Kern-Isberner (2012) provide a revision function for logic programs that originates from Hansson’s semi-revision (Hansson 1997). Since they assume the same notion of consistency as ours, all the above mentioned contradictions will be resolved in their approach.

As we have noted, classical belief revision is defined for monotonic setting, not for non-monotonic ones. Inconsistency can be caused by wrong assumptions in the non-monotonic setting but not in the monotonic setting. Such causes are not considered in (Krümpelmann and Kern-Isberner 2012). Consequently, their approach only support one of the many possible inconsistency-resolution strategies we have developed. Specifically, in (Krümpelmann and Kern-Isberner 2012), inconsistency can be resolved only by removing old beliefs; this strategy is captured by a notion analogous to s-removal compatible programs. The inconsistency-resolution strategies captured by the notion of s-expansion compatible program and s-compatible program in general are not considered.

**Conclusion and Future Work**

Depending on the application scenario, the logic governing an agent’s beliefs could be either monotonic or non-monotonic. Traditional belief revision assumes that an agent
reasons monotonically; therefore, by definition, it is applicable to such situations only. Here we have aimed to provide a belief revision framework for situations in which the agent reasons non-monotonically. To this end, we defined a belief revision function for disjunctive logic programs under the answer set semantics.

Inconsistency-resolution is an essential task for belief revision. However, the strategies used in traditional belief revision functions are limited to situations when the agent reasons monotonically. With a logic program we have the luxury of making assumptions via lack of contrary evidence, and we can deduce certain facts from such assumptions. Thus if a set of beliefs is inconsistent, then one possible cause is that we made the wrong assumption. In such cases, we can resolve the inconsistency by adding some new rules so that the assumption can no longer be made. Such a cause of inconsistency and the associated inconsistency-resolution strategy is beyond the scope of traditional belief revision, but is crucial for non-monotonic belief revision. We argue that this rationale, which is encoded in our belief revision function, captures the fundamental difference between monotonic and non-monotonic belief revision.

This paper then has explored belief base revision in the non-monotonic setting of disjunctive logic programs. Note that the characterising postulates of the base revision are formulated in terms of set-theoretic notions (e.g., subsets, set differences); the only logical notion required is consistency. Moreover the key idea, namely the notion of s-compatible programs, is also based on the same set-theoretic and logical notions. These notions are present in all non-monotonic settings. In future work we propose to extend the base revision to a general approach to belief revision in arbitrary non-monotonic settings.

Appendix: Proof of Results

In this appendix, we give the proof for the main results.

Proof for Proposition 2

Let $P$ and $Q$ are logic programs. Suppose $P \cup Q$ is m-inconsistent. We need to show $P \uparrow Q = \emptyset$.

Since $P \cup Q$ is m-inconsistent, we have $SE(P) \cap SE(Q) = \emptyset$. By the definition of s-expansion compatible program, any element in $P \uparrow Q$ has to be a superset of $P$ and consistent with $Q$. However, for any superset $R$ of $P$, we have $SE(R) \subseteq SE(P)$. Thus $SE(R) \cap SE(Q) = \emptyset$ which implies $R \cup Q$ is m-inconsistent.

Proof for Theorem 1

For one direction, suppose $*$ is a slp-revision function for $P$ and the associated selection function is $\gamma$. We need to show $*$ satisfies $(s\forall s)$, $(s\forall c)$, $(s\forall f)$, $(s\forall r)$, $(s\forall er)$, $(s\forall mr)$, and $(s\forall u)$, $(s\forall s)$, $(s\forall c)$, and $(s\forall f)$ follow immediately from the definition of slp-revision functions and compatible programs.

$(s\forall r)$: Suppose there is a set $R$ such that $R \not= \emptyset$ and $R \subseteq P \setminus (P \setminus Q)$. By the definition of slp-revision, we have $P^* \setminus Q = \gamma(P \downarrow Q) \cup Q$, hence $P^* \setminus \gamma(P \downarrow Q) \cup Q \not= \emptyset$ which implies $\gamma(P \downarrow Q) \not= P$. Then it follows from the definition of selection function that $P^* \downarrow Q \not= \emptyset$ and $\gamma(P \downarrow Q) \not= P \downarrow Q$. Let $\gamma(P \downarrow Q) = X$. Then $(P \setminus Q) \cup R = X \cup Q \cup R$. Since $\emptyset \not= R \subseteq P$, we have $(X \cup R) \cup P \subseteq (X \cup R) \cup P$. By the definition of compatible program, $X \cup R \cup Q$ is consistent that is $(P \cup Q) \cup R$ is inconsistent.

$(s\forall er)$: Suppose there is a set $E$ such that $E \not= \emptyset$ and $E \subseteq (P \setminus Q) \setminus (P \cup Q)$. By the definition of slp-revision, we have $P^* \setminus Q = \gamma(P \downarrow Q) \cup Q$, hence $\gamma(P \downarrow Q) \not= \emptyset$ which implies $\gamma(P \downarrow Q) \not= P$. Then it follows from the definition of selection function that $P^* \downarrow Q \not= \emptyset$ and $\gamma(P \downarrow Q) \not= P \downarrow Q$. Let $\gamma(P \downarrow Q) = X$. Then $(P \setminus Q) \cup E = X \cup E$. Since $E \cap P = \emptyset$ and $E \not= \emptyset \subseteq X$, $(X \setminus E) \cup P \subseteq (X \cup P)$. By the definition of compatible program, $(X \setminus E) \cup Q$ is inconsistent. Then since $E \cap Q = \emptyset$, we have $(X \setminus E) \cup Q = X \setminus (P \cup Q)$. Thus $(P \setminus Q) \cup E$ is inconsistent.

$(s\forall mr)$: Can be proved by combining the proving method for $(s\forall r)$ and $(s\forall er)$.

$(s\forall u)$: Suppose $P \downarrow Q = P \downarrow R$. Then $\gamma(P \downarrow Q) = \gamma(P \downarrow R)$. If $P \downarrow Q = P \downarrow R = \emptyset$, then by the definition of slp-revision $P^* \setminus Q = P \setminus Q$ and $P \setminus R = P \setminus R$. Thus $P \setminus (P \setminus Q) = P \setminus (P \setminus R) = P \setminus R$ and $(P \setminus Q) \cup P = (P \setminus R) \cup P = P \setminus R$. So suppose $P \not= \emptyset$ and let $X = \gamma(P \downarrow Q) = \gamma(P \downarrow R)$. By the definition of slp-revision, we have $P \setminus (P \setminus Q) = P \setminus (X \setminus Q)$. Assume $X \not= \emptyset \subseteq P \setminus Q$. Since $X \cup (P \cap Q) = X \cup P \setminus Q$ is consistent with $Q$ and $(X \cup (P \cap Q)) \subseteq P \setminus Q$, $X \not= \emptyset \setminus Q$ is a contradiction! Thus $P \setminus Q = \emptyset$ or $P \not= Q \subseteq X$. In either case we have by set theory that $P \setminus (P \setminus Q) = P \setminus (X \setminus Q) = P \setminus X$. Thus $P \setminus (P \setminus Q) = P \setminus (P \setminus R)$. Again by the definition of slp-revision, we have $(P \setminus Q) \cup (P \cap Q) = (X \setminus Q) \cup (P \cap Q) = X \setminus P$. Similarly $(P \setminus R) \setminus (P \cap Q) = (X \setminus R) \setminus (P \cap Q) = X \setminus P$. Thus $(P \setminus Q) \setminus (P \cap Q) = (P \setminus R) \setminus (P \cap Q).

For the other direction, suppose $\gamma$ is a function that satisfies $(s\forall s)$, $(s\forall c)$, $(s\forall f)$, $(s\forall r)$, $(s\forall er)$, $(s\forall mr)$, and $(s\forall u)$. We need to show $\gamma$ is a selection function.

Let $\gamma$ be defined as:

$$\gamma(P \downarrow Q) = ((P \setminus Q) \cap P) \cup ((P \setminus Q) \setminus Q)$$

for all $Q$. It suffices to show $\gamma$ is a selection function for $P$ and $P \setminus Q = \gamma(P \downarrow Q) \cup Q$.

Part 1: For $\gamma$ to be a selection function, it must be a function. Suppose $P \downarrow Q = P \downarrow R$. Then $(s\forall u)$ implies $P \setminus (P \setminus Q) = P \setminus (P \setminus R)$ and $(P \setminus Q) \setminus (P \setminus Q) = (P \setminus R) \setminus (P \setminus R)$. Since $P \setminus (P \setminus Q) \cup ((P \setminus Q) \setminus P) = (P \setminus R) \cup ((P \setminus R) \setminus P)$, $P \setminus (P \setminus Q) \setminus P = (P \setminus R) \setminus P$. Thus $(P \setminus Q) \setminus (P \setminus Q) = (P \setminus R) \setminus (P \setminus R)$. By the definition of set theory, we have $(P \setminus Q) \cap P = (P \setminus R) \cap P$ and $(P \setminus Q) \cup (P \setminus Q) = (P \setminus R) \cup (P \setminus R)$. Finally, it follows from the definition of $\gamma$ that $\gamma(P \downarrow Q) = \gamma(P \downarrow R)$.

If $P \downarrow Q = \emptyset$, then we have to show $\gamma(P \downarrow Q) = P$. $P \downarrow Q = \emptyset$ implies $Q$ is m-inconsistent, hence it follows from $(s\forall f)$ that $P \setminus Q = P \setminus Q$. Then by the definition of $\gamma$, $\gamma(P \downarrow Q) = ((P \setminus Q) \cap P) \cup ((P \setminus Q) \setminus Q) = (P \setminus Q) \cap P \cup ((P \setminus Q) \setminus Q) = P$. 

175
If \( P \uparrow Q \neq \emptyset \), then we have to show \( \gamma(P \uparrow Q) \in P \uparrow Q \).

Since \( P \uparrow Q \neq \emptyset \), \( Q \) is m-consistent. Then \((s\ast c)\) implies \( P \ast Q \) is consistent. Since \( \gamma(P \uparrow Q) \cup Q = ((P \uparrow Q) \cap P) \cup ((P \uparrow Q) \cap Q) = P \ast Q \), \( \gamma(P \uparrow Q) \cup Q \) is consistent.

Assume there is \( X \) s.t. \( X \cup Q \) is consistent and \( X \cap P \subseteq \gamma(P \uparrow Q) \cap P \). Then we have three cases:

Case 1, there is \( R \) s.t. \( \emptyset \neq R \subseteq P \) \( \gamma(P \uparrow Q) \cup Q \), and \( X = \gamma(P \uparrow Q) \cup R \). If \( R \cap Q = \emptyset \), then since \( \gamma(P \uparrow Q) \cup Q = P \ast Q, R \cap (P \ast Q) = \emptyset \). Then it follows from \((s\ast r)\) that \( (P \ast Q) \cup R \) is inconsistent. Since \( X \cup Q = (P \ast Q) \cup R \), \( X \cup Q \) is inconsistent, a contradiction! Thus \( R \cap Q \neq \emptyset \) is an impossible case.

Case 2, there is \( E \) s.t. \( E \cap P = \emptyset, \emptyset \neq E \subseteq \gamma(P \uparrow Q), \) and \( X = \gamma(P \uparrow Q) \cup E \). Then \( E \subseteq \gamma(P \uparrow Q) \cup Q = P \ast Q \). If \( E \cap Q = \emptyset \), then \((s\ast e)\) implies \( (P \ast Q) \cup E \) is inconsistent. Since \( X \cup Q = \gamma(P \uparrow Q) \cup E \cup Q = (P \ast Q) \cup E, X \cup Q \) is inconsistent, a contradiction! If \( E \cap Q \neq \emptyset \), then \( E \subseteq (P \ast Q) \cap Q \). Since \( E \cap P = \emptyset \), we have \( E \cap (P \ast Q) \cap Q = \emptyset \). Thus \( E \subseteq (P \ast Q) \cap (P \ast Q) \cup Q \), \( \gamma(P \uparrow Q) \cup Q \subseteq (P \ast Q) \), a contradiction! Thus \( E \cap Q = \emptyset \) is an impossible case.

Case 3, there are \( R \) and \( E \) s.t. \( \emptyset \neq R \subseteq P, R \cap \gamma(P \uparrow Q) = \emptyset, E \cap P = \emptyset, \emptyset \neq E \subseteq \gamma(P \uparrow Q), \) and \( X = \gamma(P \uparrow Q) \cup R \). Then we can show as in Case 1 and 2 that \( R \cap P \ast Q = \emptyset \) and \( E \subseteq P \ast Q \). If \( R \cap Q = \emptyset \) and \( E \cap Q = \emptyset \), then \((s\ast m)\) implies \((P \ast Q) \cup R \) is inconsistent. Thus \( X \cup Q = ((P \uparrow Q) \cup R) \cup E \cup Q = ((P \ast Q) \cup R) \cup E \) is inconsistent, a contradiction! Also we can show as in Case 1 and 2 that that \( R \cap Q = \emptyset \) and \( E \cap Q = \emptyset \) are impossible cases.

Part 2: By set theory, \( \gamma(P \uparrow Q) \cup Q = ((P \ast Q) \cap P) \cup ((P \ast Q) \cap Q) = ((P \ast Q) \cap P) \cup (P \ast Q) = P \ast Q \).

\[ \square \]

References


