Arg-LPNMR 2016 Proceedings

Sarah Alice Gaggl, Juan Carlos Nieves, Hannes Strass (Eds.)

November 2016

Preface

This volume contains the papers presented at Arg-LPNMR 2016: First International Workshop on Argumentation in Logic Programming and Nonmonotonic Reasoning held on July 8-10, 2016 in New York City, NY.

Research on argumentation and an Artificial Intelligence (AI) began in full force in the early eighties. The initial efforts showed how argumentation results in a very natural way of conceptualizing commonsense reasoning, appropriately reflecting its defeasible nature. In the mid-nineties, Dung (1995) has shown that argumentation provides a useful perspective for relating different non-monotonic formalisms. Currently, argumentation has been applied in different subfields of AI like Multi-Agent Systems, Semantic Web, knowledge representation and reasoning, etc.

Works in the knowledge representation and reasoning community have shown that argumentation inferences in terms of the so called argumentation semantics have strong roots in logic-based theories and non-monotonic reasoning. In this sense, the relationship between logic programming and argumentation has attracted increased attention in the last years. Studies range from translating one into the other and back, using argumentation to explain logic programming models, and using logic programming systems to implement argumentation-based languages (ASPARTIX, DIAMOND). Influences go both ways and we believe that both fields can benefit from learning from each other. Moreover, argumentation allows to relate several non-monotonic formalisms such as belief revision, reasoning about actions and probabilistic reasoning.

More recently, argumentation has been revealed as a powerful conceptual tool for exploring the theoretical foundations of reasoning and interaction in autonomous systems and Multi-Agent Systems. Different dialogue models have been proposed based on the roots of argumentation. Indeed considering argumentation roots, the so called Agreement Technologies have been suggested in order to deal with the new requirement of interaction between autonomous systems and Multi-Agent Systems.

The workshop centered around four current research strands in abstract argumentation, namely analyzing argumentation semantics, studying dynamics in argumentation, and implementations of systems for abstract argumentation.

The committee decided to accept 2 papers. The program also includes 3 invited talks.

November 2016   Sarah Alice Gaggl
                                                                                                Juan Carlos Nieves
                                                                                                    Hannes Strass

Program Committee

Gerhard Brewka	Leipzig University
Sylvie Doutre	University of Toulouse 1 - IRIT
Dov Gabbay	King's College, London
Sarah Alice Gaggl	Technische Universität Dresden
Gabriele Kern-Isberner	Technische Universitaet Dortmund
Juan Carlos Nieves	Umeå University
Mauricio Osorio	UDLAP
Phan Phan Minh Dung	AIT
Chiaki Sakama	Wakayama University
Jan Sefranek	Comenius University, Bratislava
Guillermo Simari	Dep of Computer Science and Engineering, Universidad Nacional del Sur in Bahia Blanca
Hannes Strass	Leipzig University
Paolo Torroni	University of Bologna
Toshiko Wakaki	Shibaura Institute of Technology
Stefan Woltran	TU Wien

Conference Program

SAT-Based Approaches to Reasoning about Argumentation Frameworks[*]

Matti Järvisalo

Helsinki Institute for Information Technology HIIT, Department of Computer Science,
University of Helsinki, Finland

Introduction

Argumentation is a central topic in modern Artificial Intelligence (AI) research [Bench-Capon and Dunne, 2007] with connections to a range of scientific disciplines, from computational complexity theory and automated reasoning to philosophy and social sciences, with applications to domains such as legal reasoning, multi-agent systems, and decision support. Argumentation frameworks (AFs) [Dung, 1995] have become the graph-based formal model of choice for many approaches to argumentation in AI, with semantics defining sets of jointly acceptable arguments, i.e., extensions.

System implementations for reasoning over AFs have recently received much attention. Many central AF reasoning tasks can be represented in a natural way as Boolean combinations of logical constraints via developing propositional (SAT) encodings. This is true for both what we here refer to as *static* (or *non-dynamic*) problems, as well as problems related to *AF dynamics*, dealing with adjusting (or revising) a given AF to support new knowledge represented as extensions the AF should support. Interestingly, the study of AF dynamics gives rise to optimization problems, inviting the employment of Boolean optimization solvers, such as maximum satisfiability (MaxSAT) solvers, the optimization counterpart of SAT—relying again heavily on SAT solvers.

The state-of-the-art SAT solver technology readily available today offers the core NP decision engines employed in many of the current state-of-the-art argumentation reasoning systems focusing on static reasoning problems [Thimm *et al.*, 2016]. The use of SAT solvers is not restricted to problems in NP. Rather, SAT solvers allow for solving hard decision problems presumably well beyond NP via harnessing instantiations of the general SAT-based *counterexample-guided abstraction refinement* (CEGAR) approach [Clarke *et al.*, 2004; 2003]. In short, SAT-based CEGAR is based on iterative and incremental applications of SAT solvers, iteratively solving a sequence of abstractions and ruling out non-solutions through counterexample-based refinements to the abstraction towards finding one or more solutions to the actual problem instance at hand. As complexity-theoretically very challenging problems are abundant in AF reasoning—various types of decision and optimization problems under different AF semantics exhibiting completeness for different levels of the polynomial hierarchy—developing CEGAR-type SAT-based procedures for AF reasoning tasks is an intuitive choice.

The development of SAT-based procedures for AF reasoning tasks poses interesting research challenges of both theoretical and more applied nature.

Complexity-theoretic analysis. Understanding the complexity of AF reasoning tasks with respect to different parameterizations (AF semantics, reasoning modes, and other problem-specific parameters) is essential for understanding whether a specific reasoning task allows for direct SAT encodings (in NP) and on the other hand is not "too trivial" for SAT solvers (NP-complete, or at least not solvable in close to linear time). For reasoning tasks complete for higher levels of the polynomial hierarchy (Σ_i^p / Π_i^p complete for some $i > 1$), the level i on which a specific task is situated gives guidelines on the requirements for SAT-based CEGAR suitable for the task, connecting theory to practice.

NP encodings and CEGAR. Development of SAT-based approaches is thus guided by complexity analysis for choosing the "right" approach to the AF reasoning task at hand. For problems in NP, a challenge is to develop reasonably compact direct SAT encodings (for decision problems) or MaxSAT encodings (or other constraint optimization formulations, for optimization problems) for the problem. Compactness here refers to ensuring scalability to larger AFs (with the understanding that, at times, SAT solvers can readily solve instances with millions or even tens of millions of variables and clauses [Järvisalo *et al.*, 2012]). For CEGAR, a suitable NP-abstraction is needed, as well as refinement strategies which effectively rule out non-solutions from consideration.

Implementation-level details. From encodings and procedures to implementation, the choice of the SAT and MaxSAT solvers can have a noticeable impact on scalability and efficiency, in connection to the interplay between the underlying structure of a specific AF reasoning problem, the SAT/MaxSAT encoding, and the search techniques and heuristics applying within the solvers. The incremental APIs offered by some of the central SAT solvers also play a key role in implementing CEGAR-style iterative approaches. The use of MaxSAT solvers in CEGAR has been less studied, and poses more challenges, e.g. in that few MaxSAT solvers offer APIs, and still only few are available in open source.

SAT-Based Systems: A Personal View

Cegartix *SAT-based CEGAR procedures for acceptance problems under various semantics*. A successful approach to static AF reasoning is provided by our Cegartix system [Dvořák *et al.*, 2014]. Implementing a SAT-based CE-GAR approach to second-level complete skeptical and credulous acceptance problems, the system ranked at the top on second-level problems in the ICCMA 2015 competition [Thimm *et al.*, 2016].

Pakota *MaxSAT-based procedures for enforcement*. Addressing the so-called extension enforcement problem [Baumann, 2012; Bisquert *et al.*, 2013; Coste-Marquis *et al.*, 2015] in abstract argumentation and its generalizations, in [Wallner *et al.*, 2016] we provide a nearly complete computational complexity map of fixed-argument extension enforcement under various major AF semantics, with results ranging from polynomial-time algorithms to completeness for the second-level of the polynomial hierarchy. Complementing the complexity results, we give algorithms for NP-hard extension enforcement via constrained optimization. Going beyond NP, we propose novel MaxSAT-based CEGAR for the second-level complete problems, as well as an open-source system implementation of the approach. As a continuation, we have generalized the approach to the so-called *status enforcement problem* [Niskanen *et al.*, 2016a], bringing together concepts from both static credulous/skeptical acceptance and AF dynamics, most closely, extension enforcement, resulting in Pakota, a MaxSAT-based enforcement system first of its kind in terms of generality [Niskanen *et al.*, 2016b].

AF synthesis *MaxSAT approaches to synthesizing AFs from examples*. A fundamental knowledge representational aspect related to AFs is *realizability* [Dunne *et al.*, 2015], i.e., the question of whether a specific AF semantics allows for *exactly* representing a given set of extensions as an AF [Dunne *et al.*, 2015; Baumann *et al.*, 2014; Dyrkolbotn, 2014; Pührer, 2015; Linsbichler *et al.*, 2016; 2015]. A set E of extensions is realizable (under a specific semantics) if and only if there is an AF the extensions of which are *exactly* those in E; this requires that we have *complete* knowledge of the extensions of interest, and, in order to actually construct a corresponding AF of interest, relies on the assumption that the set of extensions are *not conflicting* in terms of allowing them to be exactly represented by an AF. Recently in [Niskanen *et al.*, 2016c], we generalized the concept of realizability to accommodate incomplete and noisy information on extensions, proposing what we call the *AF synthesis problem*. Establishing NP-complete and tractable cases of AF synthesis, we have developed a first MaxSAT-based approach to optimal AF synthesis—again going from complexity-theoretic analysis to on actual implemented system for AF synthesis.

References

[Baumann *et al.*, 2014] R. Baumann, W. Dvořák, T. Linsbichler, H. Strass, and S. Woltran. Compact argumentation frameworks. In *ECAI*, volume 263 of *FAIA*, pages 69–74. IOS Press, 2014.

[Baumann, 2012] R. Baumann. What does it take to enforce an argument? Minimal change in abstract argumentation. In *ECAI*, volume 242 of *FAIA*, pages 127–132, 2012.

[Bench-Capon and Dunne, 2007] T.J.M. Bench-Capon and P.E. Dunne. Argumentation in artificial intelligence. *Artif. Intell.*, 171(10-15):619–641, 2007.

[Bisquert *et al.*, 2013] P. Bisquert, C. Cayrol, F. Dupin de Saint-Cyr, and M. Lagasquie-Schiex. Enforcement in argumentation is a kind of update. In *SUM*, volume 8078 of *LNCS*, pages 30–43. Springer, 2013.

[Clarke *et al.*, 2003] E.M. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith. Counterexample-guided abstraction refinement for symbolic model checking. *J. ACM*, 50(5):752–794, 2003.

[Clarke *et al.*, 2004] E. M. Clarke, A. Gupta, and O. Strichman. SAT-based counterexample-guided abstraction refinement. *IEEE T-CAD*, 23(7):1113–1123, 2004.

[Coste-Marquis *et al.*, 2015] S. Coste-Marquis, S. Konieczny, J. Mailly, and P. Marquis. Extension enforcement in abstract argumentation as an optimization problem. In *IJCAI*, pages 2876–2882. AAAI Press, 2015.

[Dung, 1995] P.M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995.

[Dunne *et al.*, 2015] P.E. Dunne, W. Dvořák, T. Linsbichler, and S. Woltran. Characteristics of multiple viewpoints in abstract argumentation. *Artif. Intell.*, 228:153–178, 2015.

[Dvořák *et al.*, 2014] W. Dvořák, M. Järvisalo, J.P. Wallner, and S. Woltran. Complexity-sensitive decision procedures for abstract argumentation. *Artif. Intell.*, 206:53–78, 2014.

[Dyrkolbotn, 2014] S.K. Dyrkolbotn. How to argue for anything: Enforcing arbitrary sets of labellings using AFs. In *KR*, pages 626–629. AAAI Press, 2014.

[Järvisalo *et al.*, 2012] M. Järvisalo, D. Le Berre, O. Roussel, and L. Simon. The international SAT solver competitions. *AI Mag.*, 33(1):89–92, 2012.

[Linsbichler *et al.*, 2015] T. Linsbichler, C. Spanring, and S. Woltran. The hidden power of abstract argumentation semantics. In *TAFA*, volume 9524 of *LNCS*, pages 146–162. Springer, 2015.

[Linsbichler *et al.*, 2016] T. Linsbichler, J. Pührer, and H. Strass. Characterizing realizability in abstract argumentation. In *NMR*, pages 85–94, 2016.

[Niskanen *et al.*, 2016a] A. Niskanen, J.P. Wallner, and M. Järvisalo. Optimal status enforcement in abstract argumentation. In *IJCAI*, pages 1216–1222. AAAI Press, 2016.

[Niskanen *et al.*, 2016b] A. Niskanen, J.P. Wallner, and M. Järvisalo. Pakota: A system for enforcement in abstract argumentation. In *JELIA*, LNCS. Springer, 2016.

[Niskanen *et al.*, 2016c] A. Niskanen, J.P. Wallner, and M. Järvisalo. Synthesizing argumentation frameworks from examples. In *ECAI*, volume 285 of *FAIA*, pages 551–559. IOS Press, 2016.

[Pührer, 2015] J. Pührer. Realizability of three-valued semantics for abstract dialectical frameworks. In *IJCAI*, pages 3171–3177. AAAI Press, 2015.

[Thimm *et al.*, 2016] Matthias Thimm, Serena Villata, Federico Cerutti, Nir Oren, Hannes Strass, and Mauro Vallati. Summary report of the first international competition on computational models of argumentation. *AI Mag.*, 37(1):102–104, 2016.

[Wallner *et al.*, 2016] J.P. Wallner, A. Niskanen, and M. Järvisalo. Complexity results and algorithms for extension enforcement in abstract argumentation. In *AAAI*, pages 1088–1094. AAAI Press, 2016.

Acceptability semantics for argumentation frameworks

Leila Amgoud
IRIT – CNRS
118 route de Narbonne
F-31062 Toulouse Cedex 9, France
amgoud@irit.fr

Abstract

An argument is a reason or justification of a claim. It has an intrinsic strength and may be attacked by other arguments. Hence, the evaluation of its overall strength is mandatory. Such an evaluation is done by acceptability semantics.

In this talk, we provide the foundations of a semantics, i.e., key concepts and principles on which an evaluation is based. Each concept (principle) is described by an axiom. We then present two families of semantics: extension semantics and ranking semantics. We analyze them against the axioms shedding thus light on the assumptions and choices they made. The analysis allows also a clear comparison between semantics of the same family, and between extension semantics and ranking ones.

1 Introduction

An *argument* gives reason to support a claim that is questionable, or open to doubt. It is made of three components: *premises* representing the reason, a *conclusion* which is the supported claim, and a *link* showing how the premises lead to the conclusion. The link is hence the logical "glue" that binds premises and conclusions together.

An argument has an *intrinsic strength* which may come from different sources: the certainty degree of its reason [Amgoud and Cayrol, 2002], the importance of the value it promotes if any [Bench-Capon, 2003], the reliability of its source [Parsons *et al.*, 2011], ... Whatever its intrinsic strength (strong or weak), an argument may be *attacked* by other arguments. An attack amounts to undermining one of the components of an argument, and has thus a negative impact on its target. An evaluation of the *overall strength* (or *overall acceptability*) of an argument becomes mandatory, namely for judging whether or not its conclusion is reliable.

The evaluation of arguments has received great interest from the computational argumentation community. Indeed, two families of acceptability semantics were defined for this purpose: *extension* semantics and *ranking* semantics.

Inspired from logic programming, extension semantics were initially introduced by Dung [1995]. Starting with a set of arguments and attacks between them, they return a set of extensions, each of which is a set of arguments that are acceptable together. Some semantics allow multiple extensions while others allow only a single extension. Using a membership criterion, a qualitative acceptability degree is assigned to each argument. Examples of such semantics are the classical semantics of Dung (complete, stable, preferred, ...) and their different refinements (e.g. [Baroni *et al.*, 2005; Caminada, 2006; Dung *et al.*, 2007]).

Unlike extension semantics, ranking semantics do not compute extensions. They use scoring functions which assign a numerical acceptability degree to each argument. The degree of an argument is computed in an iterative way on the basis of the degrees of its direct attackers. Examples of such semantics are h-Categorizer [Besnard and Hunter, 2001], its generalized version [Cayrol and Lagasquie-Schiex, 2005; Pu *et al.*, 2014], game-theoretic semantics [Matt and Toni, 2008], Bbs, Dbs [Amgoud and Ben-Naim, 2013], parametrized semantics [Amgoud *et al.*, 2016], and those proposed in [Gabbay, 2012; Gabbay and Rodrigues, 2015; Leite and Martins, 2011; da Costa Pereira *et al.*, 2011].

In this talk, we compare the two families of semantics (extension semantics and ranking semantics). For that purpose, we start by providing a unified definition of semantics. Then, we present the axiomatic foundations of a semantics as developed in [Amgoud and Ben-Naim, 2016]. We recall the set of axioms that was proposed. Each axiom represents a property that a semantics would satisfy or a principle on which it should be based. Furthermore, some axioms are mandatory while others are optional and represent strategic choices. We then analyze existing extension/ranking semantics against the axioms. The analysis shows the assumptions underlying each semantics, and compares the various semantics. We show that ranking semantics take into account both the number of attackers and their strengths while extension semantics neglect the number of attackers. Unlike ranking semantics, in extension semantics the effect of an attack may be lethal.

References

[Amgoud and Ben-Naim, 2013] Leila Amgoud and Jonathan Ben-Naim. Ranking-based semantics for

argumentation frameworks. In *7th International Conference on Scalable Uncertainty Management, (SUM'13)*, pages 134–147, 2013.

[Amgoud and Ben-Naim, 2016] Leila Amgoud and Jonathan Ben-Naim. Axiomatic foundations of acceptability semantics. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning, KR'16*, pages 2–11, 2016.

[Amgoud and Cayrol, 2002] Leila Amgoud and Claudette Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34(1-3):197–215, 2002.

[Amgoud *et al.*, 2016] Leila Amgoud, Jonathan Ben-Naim, Dragan Doder, and Srdjan Vesic. Ranking arguments with compensation-based semantics. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning, KR'16*, pages 12–21, 2016.

[Baroni *et al.*, 2005] P. Baroni, M. Giacomin, and G. Guida. Scc-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence Journal*, 168:162–210, 2005.

[Bench-Capon, 2003] Trevor Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.

[Besnard and Hunter, 2001] Philippe Besnard and Anthony Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence Journal*, 128(1-2):203–235, 2001.

[Caminada, 2006] M. Caminada. Semi-stable semantics. In *Proceedings of the 1st International Conference on Computational Models of Argument, (COMMA'06), IOS Press*, pages 121–130, 2006.

[Cayrol and Lagasquie-Schiex, 2005] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Graduality in argumentation. *Journal of Artificial Intelligence Research*, 23:245–297, 2005.

[da Costa Pereira *et al.*, 2011] Celia da Costa Pereira, Andreas Tettamanzi, and Serena Villata. Changing one's mind: Erase or rewind? In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence, IJCAI'11*, pages 164–171, 2011.

[Dung *et al.*, 2007] P.M. Dung, P. Mancarella, and F. Toni. Computing ideal skeptical argumentation. *Artificial Intelligence Journal*, 171:642–674, 2007.

[Dung, 1995] Phan Minh Dung. On the Acceptability of Arguments and its Fundamental Role in Non-Monotonic Reasoning, Logic Programming and n-Person Games. *Artificial Intelligence Journal*, 77:321–357, 1995.

[Gabbay and Rodrigues, 2015] Dov M. Gabbay and Odinaldo Rodrigues. Equilibrium states in numerical argumentation networks. *Logica Universalis*, 9(4):411–473, 2015.

[Gabbay, 2012] Dov M. Gabbay. Equational approach to argumentation networks. *Argument & Computation*, 3(2-3):87–142, 2012.

[Leite and Martins, 2011] J. Leite and J. Martins. Social abstract argumentation. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 2287–2292, 2011.

[Matt and Toni, 2008] P-A Matt and F. Toni. A game-theoretic measure of argument strength for abstract argumentation. In *Proceedings of 11th European Conference on Logics in Artificial Intelligence, JELIA'08*, volume 5293 of *Lecture Notes in Computer Science*, pages 285–297. Springer, 2008.

[Parsons *et al.*, 2011] Simon Parsons, Yuqing Tang, Elizabeth Sklar, Peter McBurney, and Kai Cai. Argumentation-based reasoning in agents with varying degrees of trust. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems, AAMAS'11*, pages 879–886, 2011.

[Pu *et al.*, 2014] F. Pu, J. Luo, Y. Zhang, and G. Luo. Argument ranking with categoriser function. In *Knowledge Science, Engineering and Management - 7th International Conference, KSEM 2014, Proceedings*, pages 290–301, 2014.

Aggregating Opinions in Abstract Argumentation

Richard Booth
Cardiff University, UK

1 Introduction and Background

The problem of *judgment aggregation* (JA), i.e., the problem of aggregating the opinions of a group of agents over a set of logically inter-connected propositions, has received quite a lot of attention recently from researchers in multi-agent systems, philosophy and economics. In this talk we report recent and ongoing work on a version of this problem in the setting of abstract argumentation.

We assume a fixed set $Ag = \{1, \ldots, n\}$ of agents, who are evaluating arguments of a given argumentation framework (AF) $\mathcal{A}$. Each evaluation takes the form of an $\mathcal{A}$-*labelling*, i.e., an assignment of one of the labels in (denoting *accepted*), out (*rejected*) or undec (*undecided*) to each argument of $\mathcal{A}$. Not all possible $\mathcal{A}$-labellings are *feasible*. The feasible labellings are the ones that conform to the particular argumentation *semantics* under consideration. This could be, for example, the *complete*, *stable* or *preferred* semantics, etc. We assume each agent submits a feasible $\mathcal{A}$-labelling. In this talk we will just use complete semantics, because that has been the focus of our work until now.

Given an AF $\mathcal{A} = \langle Args, \rightharpoonup \rangle$, a complete $\mathcal{A}$-labelling is a function $L : Args \rightarrow \{\texttt{in}, \texttt{out}, \texttt{undec}\}$ that satisfies, for all $a, b \in Args$:

- $L(a) = \texttt{in}$ iff $L(b) = \texttt{out}$ for all $b \in Args$ s.t. $b \rightharpoonup a$.

- $L(a) = \texttt{out}$ iff $L(b) = \texttt{in}$ for some $b \in Args$ s.t. $b \rightharpoonup a$.

We will sometimes also talk about *admissible* $\mathcal{A}$-labellings, i.e, labellings that are required to satisfy only the left-to-right directions of the above two conditions.

An $\mathcal{A}$-profile $\mathbf{L} = (L_1, \ldots, L_n)$ is a sequence of complete $\mathcal{A}$-labellings, one for each agent. Our aim is to define, in a principled manner, a concrete *aggregation method* for the agents.

Definition 1 *A (resolute) aggregation method is a function F that assigns, to every AF $\mathcal{A}$ and every $\mathcal{A}$-profile $\mathbf{L}$, an $\mathcal{A}$-labelling $F_{\mathcal{A}}(\mathbf{L})$.*

$F_{\mathcal{A}}(\mathbf{L})$ represents the $\mathcal{A}$-labelling of the group, given each agent i submits L_i.

2 Postulates for Aggregation Methods

What properties do we want our aggregation methods to satisfy? A most basic requirement is that we want the output to be feasible[1]:

Collective Completeness $F_{\mathcal{A}}(\mathbf{L})$ is a complete $\mathcal{A}$-labelling.

Compatibility enforces a level of consistency of the output with each of the agents' individual labellings. Here, for any label x, $\neg$x denotes in if x = out, out if x = in, and undec otherwise.

Compatibility For all $i \in Ag$ and $a \in Args_{\mathcal{A}}$ we have $[F_{\mathcal{A}}(\mathbf{L})](a) = \neg L_i(a)$ implies $[F_{\mathcal{A}}(\mathbf{L})](a) = \texttt{undec}$.

A weakening of *Compatibility* is the following.

in/out-Plurality If x is the in/out-loser in $(L_i(a))_{i \in Ag}$ then $[F_{\mathcal{A}}(\mathbf{L})](a) \neq$ x

Here, the in/out-loser (resp. winner) of a given tuple of labels is that label among $\{\texttt{in}, \texttt{out}\}$ that appears the fewer (resp. more) number of times.

The next property says that the collective labelling of a given argument a is independent of whichever other arguments might be present or absent in the given AF. Here $\mathbf{L}[A]$ denotes the restriction of a given profile $\mathbf{L}$ to just the arguments in A for any $A \subseteq Args_{\mathcal{A}}$.

AF-Independence If $\mathbf{L}_1$ and $\mathbf{L}_2$ are profiles over $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively and $a \in Args_{\mathcal{A}_1} \cap Args_{\mathcal{A}_2}$ then $\mathbf{L}_1[a] = \mathbf{L}_2[a]$ implies $[F_{\mathcal{A}_1}(\mathbf{L}_1)](a) = [F_{\mathcal{A}_2}(\mathbf{L}_2)](a)$.

AF-Independence turns out to be incompatible with *Collective Completeness* in the presence of a couple more mild requirements [Booth *et al.*, 2014]. A weaker version, that is arguably more appropriate anyway in an argumentation setting, is:

Directionality Suppose $\mathcal{A} \subseteq_{\mathsf{f}} \mathcal{A}'$ and suppose $Args_{\mathcal{A}}$ is unattacked in $\mathcal{A}'$. Then for any $\mathcal{A}'$-profile $\mathbf{L}$ and $a \in Args_{\mathcal{A}}$ we have $[F_{\mathcal{A}'}(\mathbf{L})](a) = [F_{\mathcal{A}}(\mathbf{L}[Args_{\mathcal{A}}])](a)$.

3 Families of Aggregation Methods

3.1 Interval Methods

In *interval aggregation methods* [Booth *et al.*, 2014], the collective label for $a \in Args_{\mathcal{A}}$ is taken to be the in/out-winner x in $\mathbf{L}[a]$, *provided that* the victory of x over $\neg$x is "sufficiently decisive". Otherwise we just take undec. Formally, let $Int_n = \{(k, l) \mid k < l, \ k, l \in \{0, 1, \ldots, n\}\}$. Let

---

[1]These and other postulates are discussed in [Booth *et al.*, 2014].

$Y \subseteq Int_n$ be such that $(0, n) \in Y$. Then we define aggregation method F^Y by setting, for each $\mathcal{A}$, $\mathcal{A}$-labelling profile $\mathbf{L}$ and $a \in Args_{\mathcal{A}}$:

$$[F_{\mathcal{A}}^Y(\mathbf{L})](a) = \begin{cases} \texttt{x} & \text{if } \texttt{x} \in \{\texttt{in}, \texttt{out}\} \text{ and} \\ & \qquad (|V_{a:\neg\texttt{x}}^{\mathbf{L}}|, |V_{a:\texttt{x}}^{\mathbf{L}}|) \in Y \\ \texttt{undec} & \text{otherwise} \end{cases}$$

Here, $V_{a:\texttt{x}}^{\mathbf{L}}$ denotes the set of agents who labelled argument a with x. If Y satisfies $[(k, l) \in Y$ implies $(s, t) \in Y$ whenever $s \leqslant k$ and $l \leqslant t]$ then F^Y is a *widening interval method*. If Y satisfies $[(k, l) \in Y$ implies $k = 0]$ then F^Y is *zero-based*. As special cases of both of these classes we have the *sceptical initial* aggregation method, for which $Y = \{(0, n)\}$, and the *credulous initial* aggregation method, for which $Y = \{(0, l) \mid l > 0\}$ [Caminada and Pigozzi, 2011].

Interval methods have been axiomatised in [Booth *et al.*, 2014]. Their characteristic postulates include *AF-Independence* and in/out-*Plurality*. Also the widening and zero-based interval methods have been characterised. An interval method is zero-based iff it satisfies *Compatibility*.

3.2 DAUC Interval Methods

Interval methods don't satisfy *Collective Completeness*. To remedy this we use the *down-admissible* and *up-complete* procedures to *repair* $F^Y(\mathbf{L})$ and make it complete. The down-admissible labelling $\downarrow L$ of an arbitrary labelling L is obtained by relabelling every illegally in- or out-labelled argument with undec. The resulting labelling will be admissible but not necessarily complete. The up-complete labelling $\uparrow L$ of a given admissible labelling can be obtained by relabelling every illegally undec-labelled argument with in or out as appropriate. (See [Caminada and Pigozzi, 2011] for details.)

The *DAUC version* of an interval method F^Y is obtained by taking as output, for any given AF $\mathcal{A}$ and $\mathcal{A}$-profile $\mathbf{L}$, $\Uparrow F_{\mathcal{A}}^Y(\mathbf{L})$. *Collective Completeness* is guaranteed by construction. DAUC interval methods don't satisfy *AF-Independence* but they satisfy instead *Directionality*. Surprisingly, in/out-*Plurality* holds only if F^Y is zero-based [Booth *et al.*, 2014]. In [Caminada and Booth, 2016] we outline *discussion-based procedures* for determining whether an argument should be labelled in by $\Uparrow F_{\mathcal{A}}^Y(\mathbf{L})$ for the special cases in which F^Y is the sceptical initial and credulous initial methods.

3.3 Partial Resolution-based Methods

Another way to use a family of intervals Y to define an aggregation method is the *partial resolution* approach [Baroni *et al.*, 2011]. Let's say $a \in Args_{\mathcal{A}}$ is Y-*supported in* $\mathbf{L}$ if $(|V_{a:\texttt{out}}^{\mathbf{L}}|, |V_{a:\texttt{in}}^{\mathbf{L}}|) \in Y$. Then modify $\mathcal{A}$ by resolving any mutual attack in favour of Y-supported arguments, i.e., if $a \rightleftharpoons b$, remove one of these attacks if it goes from an un-Y-supported argument to a Y-supported one. Then $G_{\mathcal{A}}^Y(\mathbf{L})$ is defined by taking the *grounded* labelling of this modified AF. The resulting method G^Y satisfies *Collective Completeness*. Furthermore if F^Y is widening then F^Y and G^Y give the same results when restricting to the class of *symmetric* AFs (i.e., AFs for which every attack is mutual).

3.4 Distance-based Approaches

Another approach to aggregation is to base it on some notion of *distance* $d(L_1, L_2)$ between labellings. For instance, given such a distance, we could take the result of aggregation to be that complete labelling that is *closest* to the group. Of course it may well turn out that we end up with several closest labellings, which means the output will be a *set* of labellings rather than a single one. We also need to define an appropriate distance measure. One possibility here is the *issue-based distance* [Booth *et al.*, 2012]. Roughly, the arguments in an AF can be partitioned into a set of *issues*, i.e., arguments whose labels are either always the same or always opposite (in vs out) in every complete labelling. The distance $d(L_1, L_2)$ is then determined by taking one representative a from each issue and summing $diff(L_1(a), L_2(a))$ over all these representatives, where $diff$ is some measure of difference between *labels* (e.g., $diff(\texttt{in}, \texttt{out}) = 2$, $diff(\texttt{in}, \texttt{undec}) = diff(\texttt{out}, \texttt{undec}) = 1$, etc.). By summing just over these representatives rather than over all arguments we avoid *double counting*, i.e., we disregard some differences in the labelling that might already be necessary consequences of some other differences. An initial study of the resulting aggregation methods can be found in [Podlaszewski, 2015].

4 Further and Ongoing Work

We are in the process of expanding the results not just to AFs but also to *abstract dialectical frameworks (ADFs)*. Indeed DAUC interval methods have been extended to ADFs in [Booth, 2015]. Secondly, it remains to explore the precise relation with JA: does our setting subsume the JA one, or the other way around? Finally we'd like to fully characterise the families of aggregation methods described here.

References

[Baroni *et al.*, 2011] Pietro Baroni, Paul E Dunne, and Massimiliano Giacomin. On the resolution-based family of abstract argumentation semantics and its grounded instance. *Artificial Intelligence*, 175(3):791–813, 2011.

[Booth *et al.*, 2012] Richard Booth, Martin Caminada, Mikołaj Podlaszewski, and Iyad Rahwan. Quantifying disagreement in argument-based reasoning. In *Proc. AAMAS 2012*, 2012.

[Booth *et al.*, 2014] Richard Booth, Edmond Awad, and Iyad Rahwan. Interval methods for judgment aggregation in argumentation. In *Proc. KR 2014*, 2014.

[Booth, 2015] Richard Booth. Judgment aggregation in abstract dialectical frameworks. In *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation*, pages 296–308. Springer, 2015.

[Caminada and Booth, 2016] Martin Caminada and Richard Booth. A dialectical approach for argument-based judgment aggregation. In *Proc. COMMA 2016*, 2016.

[Caminada and Pigozzi, 2011] Martin Caminada and Gabriella Pigozzi. On judgment aggregation in abstract argumentation. *Autonomous Agents and Multi-Agent Systems*, 22(1):64–102, 2011.

[Podlaszewski, 2015] Mikołaj Podlaszewski. *Poles Apart: Navigating the Space of Opinions in Argumentation*. PhD thesis, University of Luxembourg, Luxembourg, 2015.

Collaborative Planning and Decision Support for Practical Reasoning in Decentralised Supply Chains

Naeem Khalid Janjua
School of Business
University of New South Wales, Canberra, Australia.
n.janjua@unsw.edu.au

Abstract

Current advanced systems for collaborative planning among supply chain trading partners are scalable in a deterministic environment, but their scalability does not translate for effective decision making when planning under severe uncertainties that arise because of imperfect knowledge and potentially contradictory preferences. This article addresses this gap and aims to explore the logical foundations and implementation of efficient and effective Collaborative Planning and Decision Support System (CP-DSS) for practical reasoning in decentralized supply chains. Using CP-DSS the planning information, which is incomplete and contradictory, can be shared, reasoned, integrated and visualized in a form which can be readily understood by all trading partners who can then challenge both the decision and the thinking that underpins the decision. Practical reasoning [1] underlines the need for informed and experienced judgment during planning and decision support and helps in building Systems of Engagement(SoE) [2] i.e., capturing and using peer-to-peer interaction information along with transactional data for decision support, which not only provides answers to the questions related to patterns of "*what is happening*" but also "*why it is happening and what is the rationale behind it*".

1 Introduction

Supply chain (SC) activities move the entire economy of the world and are one of the major contributors to nation's GDP [3]. The increase in outsourcing and the rise of digital technologies has led to the widespread adoption of e-business models and businesses are involved in collaboration and mergers with others on a global scale, competing as a SC rather than as individuals [4]. Due to the decentralized nature of SCs, the current SC trading partners are still far from structured collaborative planning and decision support in presence of information asymmetry. This means one SC trading partner might possess some important information (potentially incomplete and/or contradictory) that is neither available in the public domain nor verifiable by a third party. The inability to share and use such information in collaborative planning and decision support may result in suboptimal decisions.

Over the past decade, SC trading partners focused on building a *Systems of Record* (SoR) i.e. structured product data, orders and demand forecasts [5]. The decision support derived from the SoR can answer questions related to the pattern "*What is happening*" but provides no information to answer questions related to the pattern "*Why it is happening and what is the rationale behind it*". As a result, complex SC networks have a tendency to become vulnerable to uncertainties and operational risks [6]. Therefore, collaborative planning and decision support faces the challenge of aligning the activities of SC trading partners in a decentralized network which contributes to the value creation of the product or service for the customer in the presence of information asymmetry to overcome operational risks. This article aims to explore the logical foundations and implementation of efficient and effective CP-DSS for practical reasoning in decentralized supply chains. Although a large body of knowledge exists on collaborative planning and decision support, this approach of practical reasoning is novel due to the following two features:

1. Incomplete and contradictory planning information is considered. Developing and agreeing on plans that involve multiple players is not a trivial task, especially when there is a lack of information pertaining to the potential risk. In real life, when the underlying available information is incomplete and/or contradictory, people are reluctant to make decisions. As a result, they indulge themselves into the process of argumentation. Argumentation involves building arguments in favour and against a certain issue. Arguments are defeasible in nature i.e. additional information may invalidate what has been previously accepted as an argument. Propositions accepted on the basis of given arguments don't grow monotonically with the available information. Using this approach, not only can we reason and deliberate over the means by which choices are compared, we can critically analyse the information upon which this comparison takes place.

2. Planning is interleaved with reasoning about preferences. Preferences are often context-dependent and conflicting preferences contribute towards the growing complexity of collaborative planning and the decision-

making processes. In order to create balanced proposals, conflicting preferences between SC trading partners need to be considered as early as possible during the planning phase for further deliberation to find a consensus over a course of action. This process is known as meta-argumentation. During meta-argumentation, reasoning and supporting data are provided by each member to support their viewpoint, and they agree or disagree by providing a justification for their opinion. The re-configuration of positions as a result of meta-argumentation between members strengthens each other's cumulative contribution towards practical planning.

2 State-of-the-art.

The collaboration problem in decentralized SCs consists of two interleaving functions, namely planning and decision support. Planning involves generating plans whose success is warranted by some evidence coming from either one or different SC trading partners and decision support to evaluate the acceptability of these plans by comparing the evidence supporting them against possible objections. A planner needs to reason about their actions. Therefore, during collaborative planning, decision makers need decision support models for choosing, organising, and revisiting their actions and plans [7]. There is a plethora of work on planning and decision support in supply chains which can be divided into the following three categories:

2.1 Mathematical and analytical based approaches

Several supply chain collaboration practices, such as the Vendor Managed Inventory, Just In Time, Efficient Consumer Response, Continuous Replenishment and Accurate Response, Collaborative Planning, Forecasting and Replenishment (CPFR) that have been suggested in the literature focus on better planning through tight processes, integration and sharing the forecasting information among the SC trading partners [8]. However, in these applications, either no or very little decision-making aid is provided with respect to the negotiation process. Several researchers introduced a process model concerned with the decision-making and negotiation aspect of collaborative planning between trading partners while respecting their local decision-making authority [9; 10]. The research approaches discussed above are drawn from applied mathematics, such as optimization, statistics and decision theory and act as a black box for decision makers. They take in information and generate results but provide no visibility as to the underlying reasoning and justification behind the results. As a result, there is limited adoption of such approaches in SCs [11].

2.2 Logic-based approaches for automated planning and decision making

Multi-agent planning systems are advanced planning systems (APS) applied to independent or loosely-coupled problems to enhance the benefit of distributed planning between autonomous software agents [12]. During coordination, software agents who are engaged in collaboration build a model of other agents' mental states and update their own beliefs and goals as the dialogue progresses[13]. To handle uncertainty, argumentation-driven frameworks that allow different software agents to share their knowledge and resolve conflicts between them to reach the common goal have been proposed [12]. However, in most APS systems, planning provides the solution, and on execution it merely traverses the identified path. Such systems work well in the Closed-World Assumption where all the possible effects of each action are known in advance. However, planning becomes very challenging if the environment is dynamically changing (the Open-World Assumption) and is not pre-engineered to conform to software agent's needs. Furthermore, planners want to use tools for better visibility of the planning process but want to control the decision-making part of the planning phase.

2.3 Logic based approaches supporting planning and decision making

The introduction of Semantic Web technology tools for collaboration has addressed some of the issues of collaboration among SC trading partners, such as information has meaning attached to it that makes it understandable across organisational boundaries and facilitates data sharing and integration [14]. Attempts have been made to represent incomplete and contradictory information in information systems such as Dr-Prolog, Dr-Device, and Situated Courteous logic [15]. These implementations only represent and handle individual conflicting preferences by defining priorities based on a single criterion between them before engaging in collaboration. Therefore, these attempts do not provide a solution for collaborative planning that is subject to inconsistencies that derive from multiple data/information sources and multiple users. Furthermore, these techniques have not yet been applied to collaborative planning and decision support domains. The Collaborative Planning and Acting Model [16] is the first attempt to support planners in managing and planning information and facilitates the planning process with automated reasoning. However, the model lacks the means to represent incomplete and contradictory information and logical relations that define constraints and the axioms of the domain being modeled.

This research is first of its kind to study collaborative planning and decision support for practical reasoning in decentralized SCs. It aims to better align the activities of SC trading partners to overcome the operational risks as shown in Figure 1. The challenge is two-fold: _firstly_, how to take into account incomplete and contradictory planning information, reason and integrate it for proactive risk prediction and better planning; _secondly_, the current logic-based approaches supporting planning and decision making only handle individual preferences in the form of priorities. Additionally, the use of these priorities is usually embedded in the reasoning mechanism and competing rules are compared individually during the reasoning process. Therefore, the derivation notion is bound to one single comparison criterion. In such a scenario, the explanation of the results is based on a single criterion only and fails to take into account the multiple factors important for decision-making.
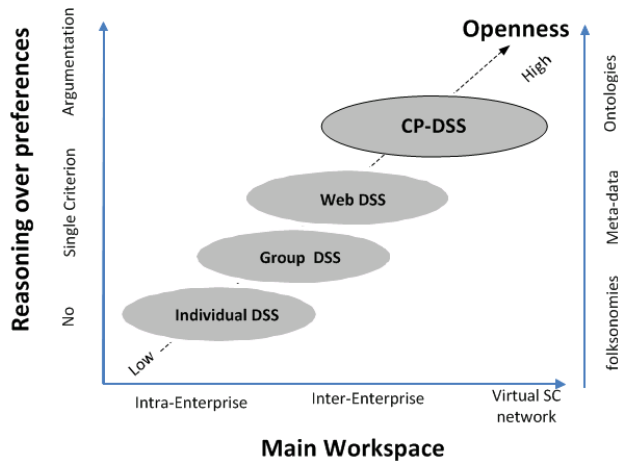
Figure 1: Evolution of Decision Support Systems

Figure 2: Collaboration in decentralized supply chain

3 Motivation

Despite the increasing use of advanced planning tools and technologies, the current SC trading partners are far from structured collaborative planning and decision support for practical reasoning. Of the various factors, data integration has been considered as one of the core problems in business information systems [17]. In the last decade, the focus of SC trading partners was on *Systems of Records* (SoR) i.e. structured data about sales, customer and product information, inventory forecasts and so, and it was used for planning and decision making. As a result, centralized enterprise information systems such as data warehousing systems, exclusively dealt with record-oriented data that was carefully mapped using schema-centric mediation approaches by knowledge experts to support planning decisions. In such information systems, the business intelligence derived for planning can often provide decision support by answering the questions related to patterns of "*what is happening*". To answer questions related to patterns of "*why is it happening and what is the rationale behind it*", it is necessary to conjointly mine the SoR with the information generated as a result of the *Systems of Engagement* (SoE) with business partners or customers. The SoE are more decentralized, incorporate digital technologies for peer-to-peer interactions, and enable SC members in a network to collaborate and engage across a range of pivotal transactional processes on a global scale as shown in Figure 2. Therefore, SoE information complements SoR data with better insight, reason and interpretation. For example, by using SoR, a supplier can predict SC disruptions that may be caused by unexpected demand patterns from other trading partners through predictive analytics, however, they would not able to obtain information on the nature of complaints or requests made by the trading partner during the negotiation process which are not captured and stored as SoE in the repositories' holding emails or transcribed phone call information records during the planning and decision-making process. Therefore, this problem of knowledge sharing and practical reasoning involves creating SoE along with SoR and uses them conjointly t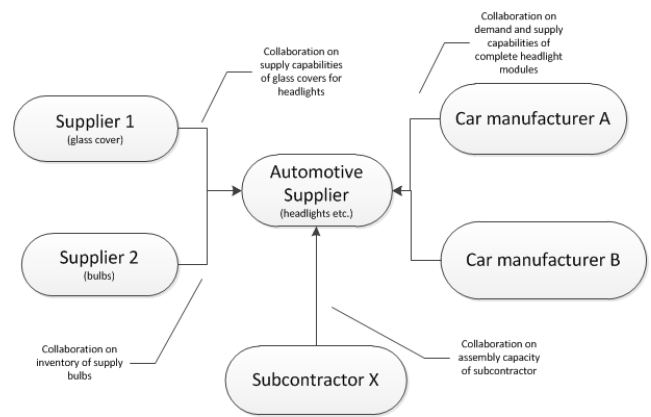o find patterns, knowledge and relationships for better decision support during collaborative planning to overcome the issue of information asymmetry. Practical reasoning is reasoning about what is to be done and it includes some actions or the adoption of an intension to initiate a sequence of actions later [1]. It underlines the need for informed and experienced judgment in many situations for decision support applications in SCs.

4 Proposed conceptual framework

In this section, the solution for Collaborative planning and Decision Support in Decentralized Supply Chains is proposed to aligned the activities of trading partners in a supply chain. The proposed framework uses Defeasible logic programming (DeLP) as knowledge representation and reasoning language [18]. DeLP which is general-purpose defeasible argumentation formalism based on logic programming, is used to model inconsistent and potentially contradictory knowledge. A defeasible logic program has the form $\psi = (\Pi, \Delta)$, where Π and Δ stand for strict knowledge (non-defeasible) and defeasible knowledge (tentative), respectively. DeLP uses the argumentation formalism for reasoning over contradictory information by identifying conflicting information in the knowledge base and applying the dialectical process to decide which information prevails during the argumentative reasoning process. DeLP only uses goal-driven reasoning with the objective of serving only the users queries. It does not provide a solution for data-driven reasoning to infer new knowledge from existing information [19]. In my previous work [20], DeLP has been extended in order to make it suitable for information representation and reasoning in a Semantic Web application. The extensions made are as follows :

- Defined syntax and semantics for DeLP to represent business (planning) rules for hybrid reasoning.

- Proposed an argumentative production system that uses DeLP as the information representation language, and performs hybrid reasoning over incomplete and/or contradictory information.

In this research, I use extended DeLP for Semantic Web application to represent the planning tasks in the CP-DSS. Figure 2 depicts the proposed framework for CP-DSS and the
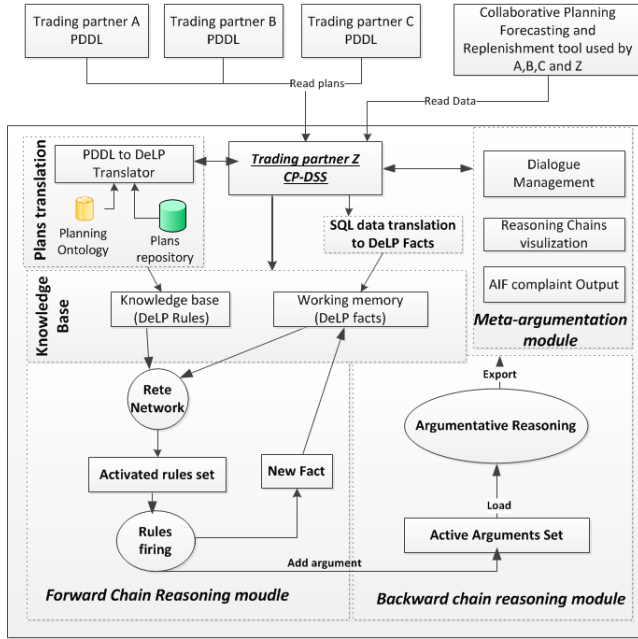
Figure 3: Conceptual framework for CP-DSS

next sub-sections outline the functionality of framework in detail.

4.1 Translating PDDL to DeLP

The PDDL (Planning Domain Description Langauge) is a standard language to describe real world planning domains [21]. Various tools such as GIPO [22] and itSIMPLE [23] provide graphical user interface to planning experts to model the real-world problems and export the plans in PDDL format so that they can be consumed by decision support tools and automated planners.

A PDDL planning problem is described in two sections: domain definition and problem specification. The domain describes all the elements which characterize the domain for planning, i.e., object types, predicates, actions (by specifying their inputs, outputs, preconditions and effects), etc. The problem essentially describes initial and goal states, by specifying the set of predicates assumed to be true in the initial state and the set of predicates to be satisfied in the goal state.

Figure 4 depicts the translation mapping from PDDL to DeLP constructs using PDDL-DeLP translator and algorithm 1 shows the entire process of the translation to DeLP constructs from PDDL in a higher level algorithm language. The first step is to download the trading partners PDDL files. During this process, the translator reads the PDDL files and saves certain information about the planning tasks such as file URL, owner/creator of planning tasks, download date etc., and saves the information in a database for their profiling. Once the download is complete, the next step is translation of PDDL files to DeLP rules and facts and save them in the knowledge base.

In the knowledge base, a planning task takes the following form: *[rule identifier][rule body] [type of rule] [head]*.

Figure 4: Translation mapping from PDDL to DeLP constructs

The rule body represents the precondition and rule head represent the effects. There are two types of rules supported b the systems i.e., strict rule represented by solid arrow and defeasible rule represented by dotted arrow. During translation, a planning action (DeLP rule) that is in conflict with another planning action (DeLP rule) in the knowledge base is represented as defeasible rule while rest of the DeLP rules are represented as strict rules. Once the translation is complete, the CP-DSS also reads the relevant SQL data from CPFR and translate them into DeLP facts, stores them in the knowledge base.

Data: PDDL files {a,b,c}
Result: DeLP Knowledge base
Array trans []= {a, b, c}
int i=0;
foreach *trans.length* **do**
    pddlfile[i].profiling(trans[i]);
    foreach *action in pddlfile[i]* **do**
        Array actions[] = readActions($action$);
        foreach *action in actions[i]* **do**
            Array predicates=create(predicate);
            Array types=createtypes;
            Array constants=create(constants);
            Array delpRules=create(predicates,types,
                constants, action[i])
        end
        i++;
    end
end
i =0;
CreateKnowledgeBase(delpRules);

Algorithm 1: Translation of PDDL file to DeLP knowledge base

Considering supplier chain scenario discussed in figure 2, the trading partner z downloads the trading partners PDDL files, translated and resulted into ψ as follows:

$$\begin{cases} [a.d1]shopper(X), product(Y), not\ advancePyament(X,Y) \\ \dashrightarrow \sim giveDiscount(X) \\ [b.d2]shopper(X), purhcase(X,Y), \\ bulkOrder(X,Y) \dashrightarrow giveDiscount(X) \\ [a.d3]eShop(Z), packaging(Y,Z) \dashrightarrow gstFree(Y) \\ [c.d5]eShop(Z), not\ packaging(Y,Z) \dashrightarrow \sim gstFree(Y) \\ [z.s2]gstFree(Y), giveDiscount(X) \\ \rightarrow ordinaryDiscount(X) \\ [z.s1]not\ gstFree(Y), giveDiscount(X) \\ \rightarrow normalDiscount(X) \\ [z.d7]shopper(X), normalDiscount(X) \\ \dashrightarrow platinumDiscount(X) \\ [c.d8]shopper(X), normalDiscount(X), \\ plansSlowToPay(X) \dashrightarrow \sim platinumDiscount(X) \end{cases}$$

[a.d1] is rule identifier where 'a' represents the trading partner and 'd1' represents the rule identifier.

4.2 Hybrid reasoning for arguments construction and conflicts handling

Once the translation from PDDL to DeLP is complete then the next step is arguments construction. This step is further divided into the following two sub-steps: Firstly, compilation of DeLP rules as a Rete network; Secondly, data driven reasoning for arguments construction.

In the proposed framework, the general Rete network [24] has been extended to represent incomplete and/or contradictory information as Rete nodes in the network. The extensions made to one-input nodes are as follows:

- AssertCondition: The one-input nodes have been extend to represent contradictory information by introduction strong negation i.e.$\sim$, as an attribute in the AssertCondition class.

- NegativeConditionNAF: A new type of one-input node was introduced to indicate incomplete information represented by the symbol 'not'.

To explain the compilation of DeLP rules in a Rete network, consider the rule base ψ outlined in section 4.1 and its subset of rules compilation represented in fig. 6 in the form of a Rete network. The predicates that make up the body of the planning rules such as bulkOrder(X,Y), shopper(X) etc are represented as one input node and the claim of the DeLP rules such as a.d1, b.d2 and a.d3 are depicted as terminal nodes. The nodes in between the one-input node and the terminal nodes are represented as two-input nodes.

Once the compilation of DeLP as a rete network is complete, the next step is to perform data-driven reasoning over underlying information in the knowledge base by passing the DeLP facts in the working memory through the Rete network. Data-driven reasoning is a forward chain reasoning that starts by the introduction of DeLP facts in the Rete network. This results in the activation and firing of the DeLP rules. The derived DeLP facts flow back into the Rete network which, in turn, results in the activation of new DeLP rules. This process continues until no more rule/s are activated. During this process, the activated DeLP rules are saved arguments in an arguments set. It is important to note here is that if the activated DeLP rules' body represents some predicate starting with the symbol 'not', then before its firing, a query is sent to the DeLP server to compute its truthfulness by querying the knowledge base. If the query returns yes, then the DeLP rule is fired, otherwise the activated DeLP rule will be removed from the activated rule set. It is important to note here is that in current research rete network has been extended from the
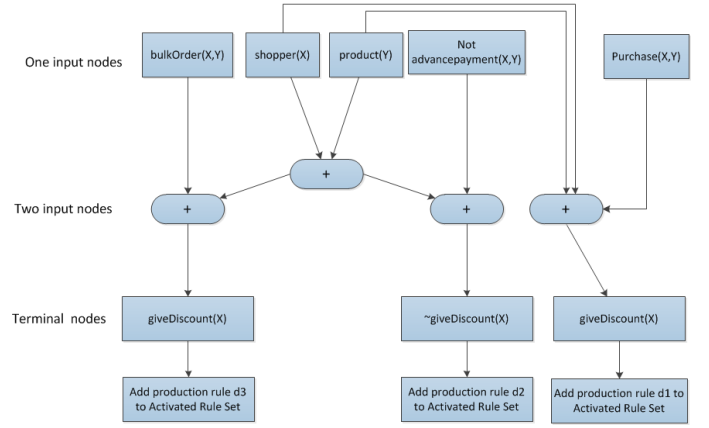

Figure 5: Compilation of production rules in the form of a Rete network

single rule execution strategy to execute all rules that are activated during data-driven reasoning. Taking into consideration the scenario depicted in figure 3, forward chain reasoning is used to digitize the planning tasks and make them alive for the planners as shown in the following illustration:

$$\begin{cases} [a.d1]shopper(david), purchase(david, rawMaterial) \\ \dashrightarrow giveDiscount(david) \\ [b.d2]shopper(david), notadvancePayment(david, rawMaterial) \\ \dashrightarrow \sim giveDiscount(david) \\ [a.d3]shopper(david), purchase(david, rawMaterial), \\ bulkOrder(david, rawMaterial) \dashrightarrow giveDiscount(david). \\ [c.d5]eShop(BigW), not\ packaging(BigW, rawMaterial) \dashrightarrow \\ \sim gstFree(rawMaterial) \\ [z.s1]notgstFree(rawMaterial), giveDiscount(david) \\ \rightarrow normalDiscount(david) \\ [z.d7]shopper(david), normalDiscount(david) \\ \dashrightarrow platinumDiscount(david) \\ [c.d8]shopper(david), product(rawMaterial), \\ havefeedback(rawMaterial, feedback), \\ reviewRate(feedback, good) \dashrightarrow purchase(david), rawMaterial) \end{cases}$$

It is important to note here is that the DeLP rules are initiated with the domain knowledge defined in the descriptive model and the DeLP facts imported from CPFR. Once the arguments construction is complete, the next step is conflicts identification and their resolution using goal-driven reasoning. Conflicts identification and their resolution is a recursive process consisting of the following two steps:

- Identification of an argument and its counter-argument.

- Compute priority between conflicting arguments by building and marking of dialectical trees as depicts in Algorithm 2.

(Dialectical tree [18]) If an argument $\mathcal{A}$ counter-argues argument $\mathcal{B}$, and no static defeat exists, then we construct a dialectical tree for argument $\mathcal{A}$ to determine whether argument $\mathcal{A}$ defeats argument $\mathcal{B}$ or vice versa.

Let $\mathcal{A}$ be an argument. A dialectical tree for argument $\mathcal{A}$, is $\Sigma(\mathcal{A}, h)$ where h is claim($\mathcal{A}$), is recursively defined as follows:

(1) A single node labeled with an argument ($\mathcal{A}$, h) with no counter-argument is by itself a dialectical tree for ($\mathcal{A}$, h). This node is also the root of the tree.

(2) Suppose that $\Sigma(\mathcal{A}, h)$ is an argument with counter-arguments ($\mathcal{A}_1, h_1$), ($\mathcal{A}_2, h_2$),.........,($\mathcal{A}_n, h_n$), we construct the dialectical tree for ($\mathcal{A}$, h), $\Sigma(\mathcal{A}, h)$ by labeling the root node
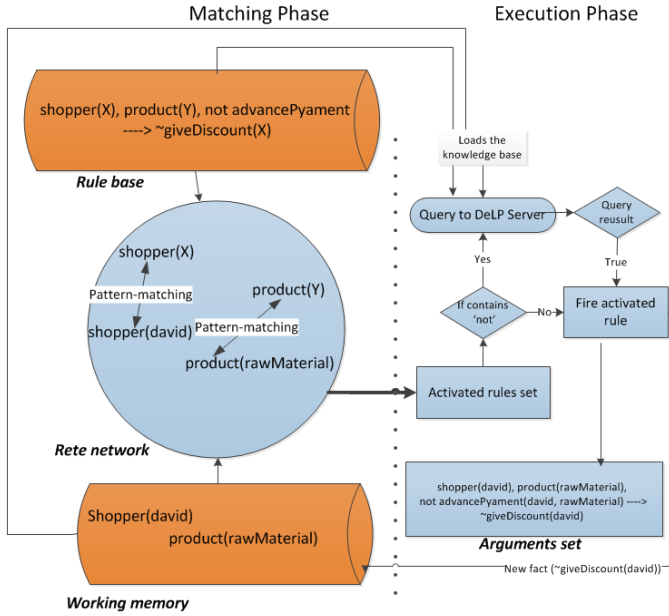
Figure 6: Data-driven reasoning by passing the facts through the Rete network

```
Data: (𝒜, h)
Result: Σ_status(𝒜, h)
Let C ← get all counter-arguments of (𝒜, h);
if C ≠ ∅ then
    while there is no Σ_U(𝒜_i, h_i) ∈ C do
        for every argument in C do
            Let (𝒜_i, h_i)← minimal non-labelled element
              BuildDialecticalTree((𝒜_i, h_i)) getting result
              as Σ(𝒜_i, h_i);
            Put Σ(𝒜_i, h_i) ξ (𝒜, h)
        end
        if there exist some Σ_U(𝒜_i, h_i) then
            Set Σ_D(𝒜, h);
        else
            Set Σ_U(𝒜, h);
        end
    end
else
    Σ(𝒜, h)= (𝒜, h);
    Set Σ(𝒜, h) ← defeated;
end
```

Algorithm 2: Building and marking of Dialectical trees

with (𝒜,h) and by making this node the parent of the root of dialectical trees for (𝒜₁,h₁), (𝒜₂,h₂),........,(𝒜ₙ,hₙ) i.e. Σ(𝒜₁, h₁),Σ(𝒜₂, h₂),........,Σ(𝒜ₙ, hₙ).

(Marking of dialectical tree [18]):

- Leaves of Σ(𝒜, h) are U-nodes.

- Let (ℬ, q) be an inner node of Σ(𝒜, h). Then (B, q) will be a U-node iff every child of (ℬ, q) is a D-node. The node (ℬ, q) will be a D-node if it has at least one U-node as a child.

If dialectical tree is marked as undefeated then this will trigger the process of meta-argumentation which is described in section 4.3. Once the hybrid reasoning is complete, the CP-DSS will display the reasoning results in a graphical format so that decision makers are able to better comprehend the results.

(Reasoning Chain): An argument 𝒜 supported by a chain of sub-arguments produces a reasoning chain $\lambda_{\mathcal{A}}$= (𝒜₁,......,𝒜ₙ) for an argument 𝒜 . The claim of supported argument 𝒜, is called a 'result' of the reasoning chain and the chain of sub-arguments is called a 'support' for the result of the reasoning chain and is define as follows: $\forall r, s \in Args$ { $if(s \xi r) then \lambda_{(r,j)}= \lambda_{(r,j)}\cup s$ where ξ is used to represent sub-argument relationship and $\lambda_{(r,j)}$ is used to represent a reasoning chain with result j. Algorithm 3 outline the process of reasoning chains construction. To share the reasoning chain with other Semantic Web applications, the reasoning chains will be annotated with an argument interchange format (AIF)[25] and is shared with trading partners in RDF\XML format.

4.3 Interleaving planning and reasoning over conflicting preference using meta-argumentation

Meta-argumentation is a deliberation dialogue that involves participants who share responsibility and collaborate on deciding what action or course of actions should be undertaken in a given situation [26]. In such dialogues, participants don't have fixed positions at the start of the dialogue and the goal and need for action can originate from any of the various participants involved. During the course of action, however, participants may be involved in a persuasion dialogue which may motivate them to model a persuasion dialogue as embedded in a deliberation dialogue. In particular, the following tasks will be performed in this sub-aim:

A plethora of work exists on building dialogue-based systems for software agents. This research focus on extending the work done by [27] using argumentation schemes [28]. During the process of argumentation, relationships between the arguments are linked with each other in a certain pattern to support the ultimate conclusion. Such linking patterns are called 'argumentation schemes' and allow reasoning to be performed using a set of premises and a conclusion. These argumentation schemes have emerged from informal logic [29]. The schemes help to categorize the way that arguments are built. They bridge the gap between logic-based application and human reasoning by capturing stereotypical patterns of human reasoning. An example is an argument from an expert opinion scheme. Formally, an argumentation scheme is composed of a set of premises A_i, a conclusion denoted as S, and a set of critical questions CQ_i is aimed at defeating the derivation of the consequent. In this research, arguments are built using argumentation schemes during meta-argumentation. The objective is two-fold, firstly; to enable planners to put forward their arguments that may be incomplete statements and offer them ways of advancing well-formed arguments as well as to reuse arguments that often

appear in discussions; secondly, with the help of algorithms, to compute the acceptability of arguments at any stage of the discussion.

Data: $(\mathcal{A}, h)$
Result: $\lambda_{(\mathcal{A},h)}$
Let S $\leftarrow$ get all sub-arguments of $(\mathcal{A}, h)$;
if $S \neq \emptyset$ **then**
    foreach $(\mathcal{A}_i, h_i) \in S$ **do**
        if *noCounterArgument*$(\mathcal{A}_i, h_i)$ *or* $\Sigma_U(\mathcal{A}_i, h_i)$ **then**
            BuildReasoningChain$((\mathcal{A}_i, h_i))$;
            Put $\lambda(\mathcal{A}_i, h_i)\,\xi\,(\mathcal{A}, h)$;
        end
    end
else
    $\lambda_{(\mathcal{A},h)} = (\mathcal{A}, h)$;
end

Algorithm 3: Construction of a reasoning chain

The deliberation dialogue system is defined by:

1. Topic Language: DeLP as a logical language.

2. Argumentation Logic: as defined in [15]. The only difference is that in our previous work it was assumed that the system has collated all the relevant information and reasoning engine reasoning over it. In this system, human planners are collaborating and conflict resolution process is a dialogue -driven activity. We reuse the definition of argument, sub-argument, attack, static defeat and dynamic defeat.

3. Communication Language to define set of Locutions S and two binary relation Ra and Rs of attacking and surrendering reply on S. Dialogue moves and termination as defined in [27].

To answer the questions of a decision maker which may help him to understand the reasoning process (that is, to obtain an explanation on the conclusion achieved), CP-DSS provides a querying mechanism to query the knowledge base. A query 'q' consists of a predicate, and can be executed on the argument set 'Args' with the help of function executeQuery(q) to check the support for the predicate in the argument set and returns the dialectical tree. Taking into considering a supply chain scenario depicts in figure 3, the figure 7 depicts the two conflicting reasoning chains produced by hybrid reasoning engine that will call for meta-argumentation in order to proceed for an action.

4.4 Prototype development and future work

The development of to address the requirements of different collaborative planning for pratical reasoning in supply chains is carried out with help Microsoft Visual Studio 2010[1], NRuler [2] whis is a fast production system library based on the RETE algorithm, written in C sharp. This library is extended for the development of the hybrid reasoning engine. QuickGraph [3] that provides generic directed/undirected graph data

---

[1]http://www.microsoft.com/visualstudio/en-us
[2]http://nruler.codeplex.com/
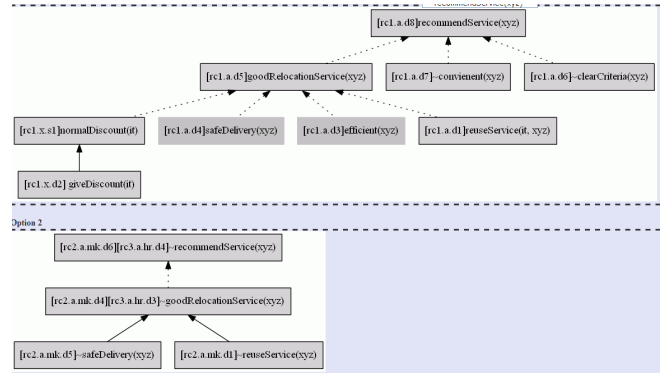[3]http://quickgraph.codeplex.com/

Figure 7: CP-DSS depicting graphical representation of conflicting plans to start meta-argumentation

structures and algorithms for .NET. It also supports Graphviz [4] to render the graphs. It is used to generate the graphical representation of the reasoning results produced by theCP-DSS and DeLP Server that is is an implementation of defeasible logic programming (DeLP). It is used as a back-end server for the development of the hybrid reasoning engine. MySQL [5] open source relational database for storing PDDL profiling information.

In future, CP-DSS will be extend with Information sharing and integration for proactive operational risk prediction. In a static environment, SC members may choose to share specific and efficient process linkages and information sharing/exchange mechanisms with selected partners. However, in a dynamic environment SC business partners need to develop more robust and reconfigurable digital linkages that can deal with changes in the business environment [30]. For example, a SC works as a network and when some event such as operational risk occurs, it is not the entire supply chain that needs to deal with this event. Once the semantic-annotated information is shared, it is used for proactive risk identification to reduce uncertainty. Uncertainty in this context is a lack of knowledge regarding the occurrence of an event in a SC to overcome operational risk. Using information for event detection requires the identification of the correlation between events and shared information among SC members. The idea is how to relate concepts to a certain event and predict the occurrence of the event using shared information.

References

[1] R. Audi, "A theory of practical reasoning," *American Philosophical Quarterly*, vol. 19, no. 1, pp. 25–39, 1982.

[2] G. Moore, "Systems of engagement and the future of enterprise IT-a sea change in enterprise IT," *Association for Information and Image Managment, Silver Spring*, 2011.

[3] A. L. Council, "The economic significance of the Australian Logistics Industry," tech. rep., Australian

---

[4]http://www.graphviz.org/
[5]http://www.mysql.com/

Logistics Council, http://austlogistics.com.au/wp-content/uploads/2014/07/Economic-Significance-of-the-Australian-Logistics-Indsutry-FINAL.pdf, 2014.

[4] R. Spekman and E. W. Davis, "The extended enterprise: a decade later," *International Journal of Physical Distribution & Logistics Management*, vol. 46, no. 1, pp. 43–61, 2016.

[5] T. S. Dillon, Y.-P. P. Chen, E. Chang, and M. Mohania, "Conjoint mining of data and content with applications in business, bio-medicine, transport logistics and electrical power systems," in *Artificial Intelligence Applications and Innovations*, 2014.

[6] Y. Daultani, S. Kumar, O. S. Vaidya, and M. K. Tiwari, "A supply chain network equilibrium model for operational and opportunism risk mitigation," *International Journal of Production Research*, vol. 53, no. 18, pp. 5685–5715, 2015.

[7] M. Ghallab, D. Nau, and P. Traverso, "The actor's view of automated planning and acting: A position paper," *AI*, vol. 208, no. 0, pp. 1–17, 2014.

[8] Y. Sheffi, "The value of CPFR," in *Proceedings of the Fourth International Congress on Logistics Research, IMRL, Lisbon, Portugal*, 2002.

[9] Y. Dong, X. Huang, K. K. Sinha, and K. Xu, "Collaborative demand forecasting: Toward the design of an exception-based forecasting mechanism," *Journal of Management Information Systems*, vol. 31, no. 2, pp. 245–284, 2014.

[10] G. Dudek, *Collaborative planning in supply chains: a negotiation-based approach*, vol. 533. Springer Science & Business Media, 2013.

[11] E. J. Umble, R. R. Haft, and M. M. Umble, "Enterprise resource planning: Implementation procedures and critical success factors," *European journal of Operational Research*, vol. 146, no. 2, pp. 241–257, 2003.

[12] A. Torreño, E. Onaindia, and Ó. Sapena, "An approach to multi-agent planning with incomplete information.," in *ECAI*, pp. 762–767, 2012.

[13] A. Gabaldon and P. Langley, "Dialogue understanding in a logic of action and belief," *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[14] N. K. Janjua, *A defeasible logic programming-based framework to support argumentation in Semantic Web applications*. Springer Science & Business Media, 2014.

[15] N. K. Janjua and F. K. Hussain, "Web@ IDSS-argumentation-enabled web-based IDSS for reasoning over incomplete and conflicting information," *Knowledge-Based Systems*, vol. 32, pp. 9–27, 2012.

[16] J. Patel and M. Dorneich, "Improving coalition planning by making plans alive," *Intelligent Systems, IEEE*, vol. 28, pp. 17–25, Jan 2013.

[17] G. C. Stevens and M. Johnson, "Integrating the supply chain 25 years on," *International Journal of Physical Distribution & Logistics Management*, vol. 46, no. 1, pp. 19–42, 2016.

[18] A. J. Garcia and G. R. Simari, "Defeasible logic programming: an argumentative approach," *Theory and Practice of Logic Programming*, vol. 4, no. 1+2, pp. 95–138, 2004.

[19] N. K. Janjua, O. K. Hussain, F. K. Hussain, and E. Chang, "Philosophical and logic-based argumentation-driven reasoning approaches and their realization on the WWW: A survey," *The Computer Journal*, vol. 58, no. 9, pp. 1967–1999, 2015.

[20] N. K. Janjua and F. K. Hussain, "Web@idss @ argumentation-enabled web-based idss for reasoning over incomplete and conflicting information," *Knowledge-Based Systems*, no. 0, pp. –, 2011.

[21] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins, "Pddl-the planning domain definition language," *Tech Report CVC TR-98-003/DCS TR-1165*, 1998.

[22] R. M. Simpson, D. E. Kitchin, and T. McCluskey, "Planning domain definition using gipo," *The Knowledge Engineering Review*, vol. 22, no. 02, pp. 117–134, 2007.

[23] T. S. Vaquero, J. R. Silva, M. Ferreira, F. Tonidandel, and J. C. Beck, "From requirements and analysis to pddl in itsimple3. 0," *Proceedings of the Third International Competition on Knowledge Engineering for Planning and Scheduling, ICAPS 2009*, pp. 54–61, 2009.

[24] C. L. Forgy, "Rete: A fast algorithm for the many pattern/many object pattern match problem," *Artificial Intelligence*, vol. 19, no. 1, pp. 17–37, 1982.

[25] I. Rahwan and C. Reed, "The argument interchange format," in *Argumentation in Artificial Intelligence* (I. Rahwan and G. R. Simari, eds.), ch. 19, pp. 383–402, Springer, 2009.

[26] D. N. Walton and E. C. Krabbe, "Commitment in dialogue: Basic concepts of interpersonal reasoning.," *State University of New York Press*, vol. 35, 1995.

[27] E. M. Kok, J.-J. C. Meyer, H. Prakken, and G. A. Vreeswijk, "A formal argumentation framework for deliberation dialogues," in *Argumentation in Multi-Agent Systems*, pp. 31–48, Springer, 2011.

[28] N. K. Janjua, F. K. Hussain, and O. K. Hussain, "Semantic information and knowledge integration through argumentative reasoning to support intelligent decision making," *Information Systems Frontiers*, vol. 15, no. 2, pp. 167–192, 2013.

[29] D. J. Godden and D. Walton, "Advances in the theory of argumentation schemes and critical questions," *Informal Logic*, vol. 27, pp. 267–292, 2007.

[30] S. Fayezi, A. Zutshi, and A. O'Loughlin, "Understanding and development of supply chain agility and flexibility: A structured literature review," *International Journal of Management Reviews*, pp. 1–30, 2016.

Revision of Abstract Dialectical Frameworks: Preliminary Report*

Thomas Linsbichler and **Stefan Woltran**

Institute of Information Systems, TU Wien, Austria

{linsbich,woltran}@dbai.tuwien.ac.at

Abstract

Abstract Dialectical Frameworks (ADFs) enhance the capability of Dung's argumentation frameworks by modelling relations between arguments in a flexible way, thus constituting a very general formalism for abstract argumentation. Since argumentation is an inherently dynamic process, understanding how change in ADFs can be formalized is important. In this work we study AGM-style revision operators for ADFs by providing various representation results. We focus on the preferred semantics and employ tools recently developed in work on revision of Horn formulas as well as logic programs. Moreover, we present an alternative family of operators based on a variant of the postulates considering preferred interpretations of the original and admissible interpretations of the revising ADF.

1 Introduction

Within the research field of argumentation in artificial intelligence (Bench-Capon and Dunne, 2007), *abstract argumentation frameworks* (AFs) as introduced by Dung (1995) have turned out to be a suitable modelling tool for various argumentation problems. This is partly due to their conceptional simplicity, being just a directed graph where nodes represent abstract arguments and edges represent conflicts between arguments. However, this comes also with limitations in terms of expressibility, which has led to the introduction of several enhancements of Dung's AFs, incorporating support (Cayrol and Lagasquie-Schiex, 2005), preferences (Modgil, 2009), attacks on attacks (Baroni *et al.*, 2011) and other concepts (see (Brewka *et al.*, 2014) for an overview). One of the most recent and powerful generalizations of AFs constitute *abstract dialectical frameworks* (ADFs) (Brewka and Woltran, 2010; Brewka *et al.*, 2013), where the relation between arguments is modelled via *acceptance conditions* for each argument (in the form of Boolean functions), capturing various forms of attack and support. This enhanced modelling capability of ADFs has been used for preferential reasoning (Brewka *et al.*, 2013), judgment aggregation (Booth, 2015), and legal reasoning (Al-Abdulkarim *et al.*, 2016).

Argumentation as such is a highly *dynamic process*. Therefore the evaluation of formalisms modelling argumentation problems is subject to constant changes in the model. As a consequence, there has been a tremendous amount of research on the dynamics of argumentation frameworks and in particular on the *revision* of Dung AFs (see e.g. (Falappa *et al.*, 2011)) in the last years. The prominent AGM approach for belief change (Alchourrón *et al.*, 1985; Katsuno and Mendelzon, 1991) was applied to AFs by Coste-Marquis *et al.* (2014), characterizing minimal change revision operators by so-called *representation results*. Diller *et al.* (2015) used recent insights on the expressiveness of AFs (Dunne *et al.*, 2015) as well as on how to deal with fragments of classical logic in belief revision (Delgrande and Peppas, 2015) to characterize AGM revision operators which return a single AF instead of a set of such.

In this work we study such AGM revision operators for ADFs. We obtain representation theorems characterizing *all* operators satisfying an adapted version of the AGM postulates by rankings on interpretations. The main challenge is the fact that ADFs are not able to express arbitrary sets of interpretations under these semantics (supported models being an exception in this matter). Fortunately, the exact *expressiveness* of ADF-semantics has recently been established by Pührer (2015) and Strass (2015), who gave exact characterizations for realizability under three-valued and two-valued semantics, respectively. We will extend and employ these results which will result in a different characterization for each semantics. Most semantics evaluate ADFs based on *three-valued* interpretations, generalizing labelling-based semantics of AFs (Caminada and Gabbay, 2009). Therefore, to obtain concrete operators, we employ a distance measure for three-valued interpretations to define rankings.

We will focus on preferred and admissible semantics – preferred interpretations are defined as maximal admissible interpretations. For revision under preferred semantics we obtain a representation result by adjusting the conditions on rankings to the limited expressiveness of the semantics and adding an additional postulate inspired by (Delgrande and Peppas, 2015) preventing cycles. The approach is similar to revision of AFs (Diller *et al.*, 2015), but deals with sets of three-valued interpretations instead of two-valued extensions. Moreover, will define a three-valued version of Dalal's well-known revision operator (Dalal, 1988). Admissible seman-

tics, on the other hand, yield only a single operator satisfying the postulates. Since, as we will argue, both approaches have some weaknesses, we propose a novel hybrid approach which bases rankings on preferred interpretations but allows admissible interpretations of the revising ADF to be the result of the revision.

Finally, we informally discuss the representation of operators for the two-valued semantics, namely stable and supported models. Moreover, we argue that complete and grounded semantics cannot be captured by the AGM approach, i.e. there are no operators satisfying the postulates.

2 Background

We assume a fixed finite set of statements A. An *interpretation* is a mapping $v : A \to \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ assigning one of the truth values true ($\mathbf{t}$), false ($\mathbf{f}$) or unknown ($\mathbf{u}$) to each statement. The set of statements to which v assigns a particular truth value $\mathbf{x} \in \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ is denoted by $v^{\mathbf{x}}$. An interpretation is *two-valued* if $v^{\mathbf{u}} = \emptyset$, i.e. the truth value $\mathbf{u}$ is not assigned. Two-valued interpretations v can be extended to assign truth values $v(\varphi) \in \{\mathbf{t}, \mathbf{f}\}$ to propositional formulas φ as usual.

The three truth values are partially ordered according to their information content: we have $\mathbf{u} <_i \mathbf{t}$ and $\mathbf{u} <_i \mathbf{f}$ and no other pair in $<_i$, meaning that the classical truth values contain more information than the truth value unknown. As usual, $\leq_i$ denotes the partial order associated to the strict partial order $<_i$. The pair $(\{\mathbf{t}, \mathbf{f}, \mathbf{u}\}, \leq_i)$ forms a complete meet-semilattice with the information meet operation $\sqcap_i$. This meet can intuitively be interpreted as *consensus* and assigns $\mathbf{t} \sqcap_i \mathbf{t} = \mathbf{t}$, $\mathbf{f} \sqcap_i \mathbf{f} = \mathbf{f}$, and returns $\mathbf{u}$ 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 then say, for two interpretations v_1, v_2, that v_2 *extends* v_1 iff $v_1 \leq_i v_2$. The set $\mathcal{V}$ 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 $a \in A$. By $\mathcal{V}^2$ we denote the set of two-valued interpretations; they are the $\leq_i$-maximal elements of the meet-semilattice $(\mathcal{V}, \leq_i)$. We denote by $[v]_2$ the set of all two-valued interpretations that extend v.

Two interpretations v_1 and v_2 are *compatible* if $v_1^{\mathbf{t}} \cap v_2^{\mathbf{f}} = v_1^{\mathbf{f}} \cap v_2^{\mathbf{t}} = \emptyset$ and *incompatible* otherwise. A set of interpretations $V \subseteq \mathcal{V}$ is compatible if each pair $v_1, v_2 \in V$ is compatible and incompatible otherwise; its *adm-closure*, $cl(V)$, contains exactly those $v \in \mathcal{V}$ such that $\forall a \in (v^{\mathbf{t}} \cup v^{\mathbf{f}}) \forall v_2 \in [v]_2 \exists v' \in V$ s.t. $v' \leq_i v_2 \wedge v'(a) = v(a)$. We will use $cl(v_1, v_2)$ as shorthand for $cl(\{v_1, v_2\})$.

We define the symmetric distance function $\triangle$ between truth values as follows: $\mathbf{t} \triangle \mathbf{f} = 1$, $\mathbf{t} \triangle \mathbf{u} = \mathbf{f} \triangle \mathbf{u} = \frac{1}{2}$, and $\mathbf{x} \triangle \mathbf{x} = 0$ for $\mathbf{x} \in \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$. Lifted to interpretations $v_1, v_2 \in \mathcal{V}$, it is defined as $v_1 \triangle v_2 = \sum_{a \in A} v_1(a) \triangle v_2(a)$.

Finally note that we will represent interpretations by sequences of truth values, assuming a total ordering on the underlying vocabulary. For instance the interpretation $\{a \mapsto \mathbf{u}, b \mapsto \mathbf{t}, c \mapsto \mathbf{f}\}$ will be abbreviated by $\mathbf{utf}$.
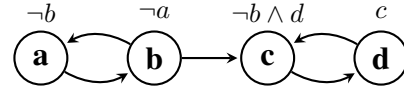

Figure 1: ADF $F = \{\langle a, \neg b \rangle, \langle b, \neg a \rangle, \langle c, \neg b \wedge d \rangle, \langle d, c \rangle\}$.

2.1 Abstract Dialectical Frameworks

An ADF F is a set of tuples $\langle s, \varphi_s \rangle$ where $s \in A$ is a statement and φ_s is a propositional formula over A, the acceptance condition of s. Note that this formalization syntactically differs from the original one (Brewka and Woltran, 2010), where an ADF is represented by a triple (A, L, C) where L is a set of links between statements and C a set of total functions $2^A \mapsto \{\mathbf{t}, \mathbf{f}\}$. It is however easy to see that these two notions are equivalent, as the set of links L is implicitly given by the atoms occurring in the acceptance conditions and the fact that the total functions C can be expressed by propositional formulas. We denote the set of all ADFs by $\mathcal{F}_A$.

The semantics of ADFs can be defined via an operator Γ_F over three-valued interpretations. Given an ADF F and an interpretation v, it is defined as

$$\Gamma_F(v)(a) = \bigsqcap_{w \in [v]_2} w(\varphi_a).$$

Intuitively, the operator returns, for each statements a, the consensus truth value of the evaluation of the acceptance formula φ_a with each two-valued interpretation extending v. The semantics can now be defined as follows:

Definition 1. Given an ADF F, an interpretation v is

- *admissible* for F iff $v \leq_i \Gamma_F(v)$,
- *complete* for F iff $v = \Gamma_F(v)$,
- *preferred* for F iff v is admissible for F and each $v' \in \mathcal{V}$ with $v <_i v'$ is not admissible for F,
- *grounded* for F iff v is complete for F and each $v' \in \mathcal{V}$ with $v' <_i v$ is not complete for F,
- a *(supported) model* of F iff v is two-valued and $v = \Gamma_F(v)$,
- a *stable model* of F iff v is a model of F and $v^t = w^t$, where w is the grounded interpretation of $F^v = \{\langle a, \varphi_a[x/\bot : v(x) = \mathbf{f}] \rangle \mid a \in v^{\mathbf{t}}\}$.

We denote the admissible, complete, preferred, and grounded interpretations for, and supported and stable models of an ADF F by $ad(F)$, $co(F)$, $pr(F)$, $gr(F)$, $mo(F)$, and $st(F)$, respectively. For alternative semantics we refer to (Strass, 2013; Polberg, 2014).

The semantics have been shown to be proper generalizations of AF-semantics (Brewka and Woltran, 2010; Brewka *et al.*, 2013), with both supported and stable models generalizing stable semantics of AFs, differing only in the treatment of support cycles.

Example 1. Consider the ADF F depicted in Figure 1. The admissible interpretations of F are as follows: $ad(F) = \{\mathbf{uuuu}, \mathbf{tfuu}, \mathbf{tfff}, \mathbf{tftt}, \mathbf{ftuu}, \mathbf{ftfu}, \mathbf{ftff}, \mathbf{uuff}\}$. Further

observe that $co(F) = ad(F) \setminus \{\mathbf{ftuu}, \mathbf{ftfu}\}$, $pr(F) = \{\mathbf{tfff}, \mathbf{tftt}, \mathbf{ftff}\}$, $gr(F) = \{\mathbf{uuuu}\}$, $mo(F) = pr(F)$, and $st(F) = pr(F) \setminus \{\mathbf{tftt}\}$.

A set of interpretations V is *realizable* under a semantics σ if there is an ADF F with $\sigma(F) = V$. The following proposition recalls results which are either explicitly stated or immediate consequences of (Pührer, 2015) and (Strass, 2015).

Proposition 1. *A set of interpretations V is realizable under*

- *ad iff $V \neq \emptyset$ and $V = cl(V)$;*
- *pr iff $V \neq \emptyset$ and V is incompatible;*
- *mo iff $V \subseteq \mathcal{V}^2$;*
- *st iff $V \subseteq \mathcal{V}^2$ and $v_1^{\mathbf{t}} \not\subseteq v_2^{\mathbf{t}}, v_2^{\mathbf{t}} \not\subseteq v_1^{\mathbf{t}}$ for all $v_1, v_2 \in V$.*

For $\sigma \in \{ad, pr, mo, st\}$ it holds that $\sigma(F) \cap \sigma(G)$ is realizable under σ for arbitrary ADFs F and G, given that $\sigma(F) \cap \sigma(G) \neq \emptyset$. This does not hold for co and gr in general.

Definition 2. Given a semantics σ, the function $f_\sigma : 2^{\mathcal{V}} \mapsto \mathcal{F}_A$ maps sets of interpretations to ADFs such that $\sigma(f_\sigma(V)) = V$ if V is realizable under σ and $\sigma(f_\sigma(V)) = \mathbf{u}\ldots\mathbf{u}$ otherwise.

Note that canonical constructions for $f_\sigma(V)$ for realizable sets V can be found in (Pührer, 2015) and (Strass, 2015). Although f_σ is not unique in general, it is assumed to be fixed for every σ throughout the paper. In particular, $f_\sigma(V) = \{\langle a, \neg a \rangle \mid a \in A\}$ for V not realizable under σ.

2.2 Belief Revision

The most prominent approach to belief revision was introduced by Alchourrón *et al.* (1985) and reformulated for propositional formulas by Katsuno and Mendelzon (1991). They define an equivalent version of the AGM-postulates for operators $*$ mapping pairs of formulas to a revised formula.

$(R1)$ $\psi * \mu \models \mu$.

$(R2)$ If $\psi \wedge \mu$ is satisfiable, then $\psi * \mu = \psi \wedge \mu$.

$(R3)$ If μ is satisfiable, then $\psi \wedge \mu$ is also satisfiable.

$(R4)$ If $\psi_1 \equiv \psi_2$ and $\mu_1 \equiv \mu_2$, then $\psi_1 * \mu_1 \equiv \psi_2 * \mu_2$.

$(R5)$ $(\psi * \mu) \wedge \phi \models \psi * (\mu \wedge \phi)$.

$(R6)$ If $(\psi * \mu) \wedge \phi$ is satisfiable, then $\psi * (\mu \wedge \phi) \models (\psi * \mu) \wedge \phi$.

While postulates $R1$ to $R4$ are self-explanatory, note that $R5$ and $R6$ ensure that revision is performed with minimal change to the revised formula ψ.

The main result of (Katsuno and Mendelzon, 1991) is that there is a one-to-one correspondence between operators which are rational according to the AGM postulates and functions mapping each formula to a certain binary relation among interpretations. Thus, for constructing an AGM operator, a necessary and sufficient condition is the existence of such a function.

Definition 3. A *preorder* $\preceq$ on $\mathcal{V}$ is a reflexive, transitive binary relation on $\mathcal{V}$. The preorder $\preceq$ is

- *total* if $v_1 \preceq v_2$ or $v_2 \preceq v_1$ for any $v_1, v_2 \in \mathcal{V}$,
- *i-max-total* if $v_1 \preceq v_2$ or $v_2 \preceq v_1$ for any $v_1, v_2 \in \mathcal{V}$ with $v_1 \not\preceq_i v_2$ and $v_2 \not\preceq_i v_1$.

Moreover, for $v_1, v_2 \in \mathcal{V}$, $v_1 \prec v_2$ denotes the strict part of $\preceq$, i.e. $v_1 \preceq v_2$ and $v_2 \not\preceq v_1$. We write $v_1 \approx v_2$ in case $v_1 \preceq v_2$ and $v_2 \preceq v_1$.

Given a preorder, the construction of the corresponding operator is then based on the following selection function:

$$\min(V, \preceq) = \{v_1 \in V \mid \nexists v_2 \in V : v_2 \prec v_1\}.$$

3 Revising ADFs

In this section we apply the AGM approach to the revision of ADFs by studying operators $* : \mathcal{F}_A \times \mathcal{F}_A \mapsto \mathcal{F}_A$. It is inspired by the approach by Diller *et al.* (2015) to revision of AFs. We begin by reformulating the postulates for our setting, parameterized by the used semantics.

$(A1_\sigma)$ $\sigma(F * G) \subseteq \sigma(G)$.

$(A2_\sigma)$ If $\sigma(F) \cap \sigma(G) \neq \emptyset$, then $\sigma(F * G) = \sigma(F) \cap \sigma(G)$.

$(A3_\sigma)$ If $\sigma(G) \neq \emptyset$, then $\sigma(F * G) \neq \emptyset$.

$(A4_\sigma)$ If $\sigma(G) = \sigma(H)$, then $\sigma(F * G) = \sigma(F * H)$.

$(A5_\sigma)$ $\sigma(F * G) \cap \sigma(H) \subseteq \sigma(F * f_\sigma(\sigma(G) \cap \sigma(H)))$.

$(A6_\sigma)$ If $\sigma(F * G) \cap \sigma(H) \neq \emptyset$, then $\sigma(F * f_\sigma(\sigma(G) \cap \sigma(H))) \subseteq \sigma(F * G) \cap \sigma(H)$.

Next we define two types of rankings which will be the counterpart to the postulates in the representation results.

Definition 4. Given a semantics σ and an ADF F, a preorder $\preceq_F$ is a *(i-max-)faithful ranking* for F if it is (i-max-)total and for all (incompatible) interpretations $v_1, v_2 \in \mathcal{V}$ it holds that

(i) if $v_1, v_2 \in \sigma(F)$ then $v_1 \approx_F v_2$, and

(ii) if $v_1 \in \sigma(F)$ and $v_2 \notin \sigma(F)$ then $v_1 \prec_F v_2$.

A function mapping each ADF to a (i-max-)faithful ranking is called *(i-max-)faithful assignment*.

3.1 Revision under Preferred Semantics

In the remainder of this section we will focus on the preferred semantics. To fulfill the postulates, a revision operator will have to result in an ADF having certain preferred interpretations. However, as can be already seen by Proposition 1, preferred semantics (and all the others) underlie certain limits in terms of expressiveness. That is, certain desired outcomes may not be realizable. We first give sufficient conditions for realizability we will make use of in the following.

Proposition 2. *A set of interpretations $V \subseteq \mathcal{V}$ is realizable under pr if one of the following holds:*

1. *$V \subseteq pr(F)$ and $V \neq \emptyset$ for some $F \in \mathcal{F}_A$;*
2. *$V = \{v_1, v_2\}$ and v_1 and v_2 are incompatible; or*
3. *$V = \{v\}$.*

The following example shows that the standard set of postulates is not enough to get a correspondence to preorders on interpretations.

Example 2. Consider an arbitrary ADF F and the binary relation $\preceq$ having $pr(F)$ as least elements, containing the cycle $\mathbf{uft} \prec \mathbf{ttf} \prec \mathbf{fut} \prec \mathbf{tuf} \prec \mathbf{uft}$ and being a linear order otherwise. Note that $\preceq$ is not transitive and therefore

only a pseudo-preorder. However, the revision operator $*$ induced by $\preceq$ can be shown to satisfy all postulates $A1_{pr}-A6_{pr}$. Moreover, every binary relation $\preceq'$ inducing the same operator $*$ must contain this cycle. Consider the pair of interpretations $\mathtt{uft}$ and $\mathtt{ttf}$. They are incompatible, hence realizable (cf. Proposition 1). Therefore the revision of F by $f_{pr}(\{\mathtt{uft}, \mathtt{ttf}\})$ must have $\mathtt{uft}$ as single preferred interpretation, hence $\mathtt{uft} \prec' \mathtt{ttf}$. This holds for every neighboring pair of the cycle, hence $\preceq'$ must contain the same cycle.

The following postulate, which is adapted from Delgrande and Peppas (2015), closes this gap. Note that it is redundant in classical AGM revision.

$(Acyc_\sigma)$ If for $1 \leq i < n$, $\sigma(F * G_{i+1}) \cap \sigma(G_i) \neq \emptyset$ and $\sigma(F * G_1) \cap \sigma(G_n) \neq \emptyset$ then $\sigma(F * G_n) \cap \sigma(G_1) \neq \emptyset$.

We are now ready to give our first representation result.

Theorem 1. *Let F be an* ADF *and $\preceq_F$ an i-max-faithful ranking for F. Define an operator $* : \mathcal{F}_A \times \mathcal{F}_A \mapsto \mathcal{F}_A$ by*

$$F * G = f_{pr}(\min(pr(G), \preceq_F)).$$

Then $*$ satisfies postulates $A1_{pr} - A6_{pr}$ and $Acyc_{pr}$.

Proof. By definition of f_{pr} and the fact that any non-empty $V \subseteq pr(G)$ is realizable under pr (cf. Proposition 2.1) it holds that $pr(f_{pr}(\min(pr(G), \preceq_F))) = \min(pr(G), \preceq_F)$, hence $pr(F * G) = \min(pr(G), \preceq_F)$. Thus $A1_{pr}$ and $A4_{pr}$ follow.

For $A2_{pr}$, assume $pr(F) \cap pr(G) \neq \emptyset$. Since $\preceq_F$ is i-max-faithful and all $v_1, v_2 \in pr(G)$ are pairwise incompatible we get that $\min(pr(G), \preceq_F) = pr(F) \cap pr(G)$ and hence $pr(F * G) = pr(F) \cap pr(G)$.

As $\preceq_F$ is transitive (by being a preorder) and A is finite, $\min(pr(G), \preceq_F) \neq \emptyset$, hence $A3_{pr}$ holds.

For $A5_{pr}$ and $A6_{pr}$ we consider the non-trivial case where $pr(F * G) \cap pr(H) \neq \emptyset$. Recalling that $pr(F) \cap pr(G)$ is realizable under pr (cf. Proposition 1), we have to show that $\min(pr(G), \preceq_F) \cap pr(H) = \min(pr(G) \cap pr(H), \preceq_F)$. Towards a contradiction assume there is some $v \in \min(pr(G), \preceq_F) \cap pr(H)$ such that $v \notin \min(pr(G) \cap pr(H), \preceq_F)$. As then $v \in pr(G)$ and $v \in pr(H)$ there must be some $v' \in pr(G) \cap pr(H)$ with $v' \prec_F v$, contradicting $v \in \min(pr(G), \preceq_F)$. On the other hand assume, again to the contrary, that there is some $v \in \min(pr(G) \cap pr(H), \preceq_F)$ such that $v \notin \min(pr(G), \preceq_F) \cap pr(H)$. From $v \in pr(H)$ we get $v \notin \min(pr(G), \preceq_F)$. As by assumption $pr(F * G) \cap prf(H) \neq \emptyset$, let $v' \in \min(pr(G), \preceq_F)$ and $v' \in pr(H)$. Then also $v' \in pr(G) \cap pr(H)$. Since $v, v' \in pr(H)$, v and v' are incompatible, $\preceq_F$ is i-max-total and $v \in \min(pr(G) \cap pr(H), \preceq_F)$ by assumption, we get $v \preceq_F v'$. Thus $v \in \min(pr(G), \preceq_F)$ because $v' \in \min(pr(G), \preceq_F)$, a contradiction.

For $Acyc_{pr}$ consider a sequence of ADFs $G_1, \ldots, G_n$ such that $pr(F * G_{i+1}) \cap pr(G_i) \neq \emptyset$ for $1 \leq i < n$ and $pr(F * G_1) \cap pr(G_n) \neq \emptyset$. Let $1 \leq i < n$. By definition of $*$ we have $pr(f_{pr}(\min(pr(G_{i+1}), \preceq_F))) \cap pr(G_i) \neq \emptyset$. Then, by Proposition 2, $\min(pr(G_{i+1}), \preceq_F) \cap pr(G_i) \neq \emptyset$ follows. Hence there is some $v_i' \in pr(G_i)$ such that $v_i' \preceq_F v_{i+1}$ for all $v_{i+1} \in pr(G_{i+1})$. From transitivity of $\preceq_F$ we infer that there is a $v_1' \in pr(G_1)$ such that $v_1' \preceq_F v_n$ for all $v_n \in pr(G_n)$.

From $pr(F * G_1) \cap pr(G_n) \neq \emptyset$ it follows that there is some $v_1'' \in \min(G_1, \preceq_F)$ (hence also $v_1'' \in pr(G_1)$ and $v_1'' \preceq_F v_1'$) with $v_1'' \in pr(G_n)$. We have $v_1'' \preceq_F v_1' \preceq_F v_n$ (for each $v_n \in pr(G_n)$), hence $v_1'' \in \min(pr(G_n), \preceq_F)$. This together with $v_1'' \in pr(G_0)$ means that $pr(F * G_n) \cap pr(G_1) \neq \emptyset$, which was to show. □

Theorem 2. *Let $* : \mathcal{F}_A \times \mathcal{F}_A \mapsto \mathcal{F}_A$ be a revision operator satisfying postulates $A1_{pr} - A6_{pr}$ and $Acyc_{pr}$. Then there is an assignment mapping each* ADF *F to an i-max-faithful ranking $\preceq$ such that $pr(F * G) = \min(pr(G), \preceq)$ for every* ADF *G.*

Proof. Assume an arbitrary ADF F. We will define $\preceq$ and show that it is an i-max-faithful ranking for F and that $pr(F * G) = \min(pr(G), \preceq)$.

First let $\preceq'$ be the relation on $\mathcal{V}$ such that for each $v \in \mathcal{V}$, $v \approx' v$, and for any incompatible interpretations $v_1, v_2 \in \mathcal{V}$,

$$v_1 \preceq' v_2 \Leftrightarrow v_1 \in pr(F * f_{pr}(\{v_1, v_2\})).$$

The relation $\preceq$ is defined as the transitive closure of $\preceq'$:

$$v \preceq v' \Leftrightarrow \exists w_1, \ldots, w_n : v \preceq' w_1 \preceq' \cdots \preceq' w_n \preceq' v'.$$

First, $\preceq$ is clearly reflexive and transitive, making it a preorder on $\mathcal{V}$. Moreover, for incompatible interpretations $v_1, v_2 \in \mathcal{V}$ we know from Proposition 2.2 that $\{v_1, v_2\}$ is realizable under pr, hence $pr(f_{pr}(\{v_1, v_2\})) = \{v_1, v_2\}$. By $A1_{pr}$ and $A3_{pr}$ we therefore get that either $v_1 \preceq' v_2$ or $v_2 \preceq' v_1$, and, consequently, also $v_1 \preceq v_2$ or $v_2 \preceq v_1$, hence $\preceq$ is i-max-total.

We proceed by showing that $\preceq$ is i-max-faithful. To show (i), let $v_1, v_2 \in pr(F)$ and note that $pr(f_{pr}(\{v_1, v_2\})) = \{v_1, v_2\}$. Hence, by $A2_{pr}$, we get $pr(F * f_{pr}(\{v_1, v_2\})) = \{v_1, v_2\}$. Therefore, by definition of $\preceq$, $v_1 \preceq v_2$ and $v_2 \preceq v_1$, i.e. $v_1 \approx v_2$. For (ii), we begin with the intermediate statement

$$\text{for } v_1 \ldots v_n \in \mathcal{V} : v_1 \preceq' \cdots \preceq' v_n \preceq' v_1 \Rightarrow v_1 \preceq' v_n \quad (1)$$

For $n \leq 2$ the statement is immediate. Assume $n > 2$. By definition of $\preceq'$ we first get that v_i and v_{i+1} for $i \in \{1, \ldots, n-1\}$ as well as v_n and v_1 are incompatible, hence $f_{pr}(\{v_i, v_{i+1}\}) = \{v_i, v_{i+1}\}$ and $f_{pr}(\{v_n, v_1\}) = \{v_n, v_1\}$ by Proposition 2. Moreover, we get $v_i \in pr(F * f_{pr}(\{v_i, v_{i+1}\}))$ for $i \in \{1, \ldots, n-1\}$ and $v_n \in pr(F * f_{pr}(\{v_n, v_1\}))$. It follows that $v_1 \in pr(F * f_{pr}(\{v_1, v_2\})) \cap \{v_n, v_1\}$, $v_i \in pr(F * f_{pr}(\{v_i, v_{i+1}\})) \cap \{v_{i-1}, v_i\}$ for $i \in \{2, \ldots, n-1\}$, and $v_n \in pr(F * f_{pr}(\{v_n, v_1\})) \cap \{v_{n-1}, v_n\}$. Considering $Acyc$ we get $pr(F * f_{pr}(\{v_n, v_1\})) \cap \{v_1, v_2\} \neq \emptyset$, meaning further by $A5_{pr}$ and $A6_{pr}$ that $pr(F * f_{pr}(\{v_n, v_1\})) \cap \{v_1, v_2\} = pr(F * f_{pr}(\{v_n, v_1\} \cap \{v_1, v_2\})) = pr(F * f_{pr}(\{v_1\}))$. By $pr(f_{pr}(\{v_1\})) = \{v_1\}$ (cf. Proposition 2.3), $A1_{pr}$ and $A3_{pr}$, we follow that $v_1 \in pr(F * f_{pr}(\{v_n, v_1\}))$, meaning that $v_1 \preceq' v_n$, concluding the proof for (1). We proceed by showing the statement

$$\text{for } v_1, v_2 \in \mathcal{V} : v_1 \prec' v_2 \Rightarrow v_1 \prec v_2 \quad (2)$$

$v_1 \preceq v_2$ is clear by definition. Assume, towards a contradiction, that $v_2 \preceq v_1$. Then $\exists w_1, \ldots, w_n$ such that $v_1 \preceq'

$w_1 \preceq' \cdots \preceq' w_n \preceq' v_2$. As by assumption $v_1 \preceq' v_2$ we follow by (1) that $v_2 \preceq' v_1$, a contradiction to $v_1 \prec' v_2$, showing (2). Now let v_1 and v_2 be incompatible interpretations such that $v_1 \in pr(F)$ and $v_2 \notin pr(F)$. By $A2_{pr}$ we get $pr(F * f_{pr}(\{v_1, v_2\})) = pr(F) \cap \{v_1, v_2\} = \{v_1\}$, implying $v_1 \preceq' v_2$. Therefore, by (2), also $v_1 \preceq v_2$, showing (ii) and, consequently, that $\preceq$ is i-max-faithful.

Before showing that $*$ is indeed simulated by $\preceq$, we prove

$$\text{for } v_1, v_2 \in \mathcal{V} \text{ s.t. } v_1 \preceq' v_2, G \in \mathcal{F}_A :$$
$$v_1 \in pr(G) \wedge v_2 \in pr(F * G) \Rightarrow v_1 \in pr(F * G) \quad (3)$$

Let $G \in \mathcal{F}_A$ such that $v_1 \in pr(G)$ and $v_2 \in pr(F * G)$. First note that, by $*$ fulfilling $A1_{pr}$, $v_2 \in pr(G)$, meaning that v_1 and v_2 are incompatible and therefore $pr(f_{pr}(\{v_1, v_2\})) = \{v_1, v_2\}$. From $A5_{pr}$ and $A6_{pr}$ we then get that $pr(F * G) \cap \{v_1, v_2\} = pr(F * f_{pr}(pr(G) \cap \{v_1, v_2\})) = pr(F * f_{pr}(\{v_1, v_2\}))$. By the assumption that $v_1 \preceq' v_2$ it holds that $v_1 \in pr(F * f_{pr}(\{v_1, v_2\}))$, hence (3) follows. The last intermediate step is to show that

$$\text{for } G \in \mathcal{F}_A : \min(pr(G), \preceq) = \min(pr(G), \preceq') \quad (4)$$

Consider some $G \in \mathcal{F}_A$. ($\subseteq$) Let $v_1 \in \min(pr(G), \preceq)$ and suppose there exists an $v_2 \in pr(G)$ with $v_2 \prec' v_1$. This means, by (2), that also $v_2 \preceq v_1$, a contradiction. Hence $v_2 \nprec' v_1$ for all $v_2 \in pr(G)$, i.e. $v_1 \in \min(pr(G), \preceq')$. ($\supseteq$) Let $v_1 \in \min(pr(G), \preceq')$ and $v_2 \in pr(G)$. We show that $v_1 \preceq' v_2$, since then $v_1 \preceq v_2$ and, consequently, $v_1 \in \min(pr(G), \preceq)$ follows by definition of $\preceq$. If $v_1 = v_2$ we have $v_1 \preceq' v_2$ by definition of $\preceq'$. If $v_1 \neq v_2$ observe that, by $v_1, v_2 \in pr(G)$, v_1 and v_2 are incompatible, hence at least one of $v_1 \preceq' v_2$ and $v_2 \preceq' v_1$ must hold. By $v_1 \in \min(pr(G), \preceq')$ it cannot hold that $v_2 \prec' v_1$, hence $v_1 \preceq' v_2$.

We are now ready to show that, for any ADF G, $pr(F*G) = \min(pr(G), \preceq)$. Considering (4) we just have to show that

$$\text{for } G \in \mathcal{F}_A : pr(F * G) = \min(pr(G), \preceq') \quad (5)$$

($\subseteq$) Let $v \in pr(F * G)$ and keep in mind that, by $A1_{pr}$, also $v \in pr(G)$. We show for each $w \in pr(G)$ that $v \preceq' w$. Consider an arbitrary $w \in pr(G)$. Note that by $v, w \in pr(G)$ we have that $pr(f_{pr}(\{v, w\})) = \{v, w\}$. From $A5_{pr}$ and $A6_{pr}$ we get $pr(F * G) \cap \{v, w\} = pr(F * f_{pr}(pr(G) \cap \{v, w\})) = pr(F * f_{pr}(\{v, w\}))$. As by assumption $v \in pr(F * G)$ we get $v \preceq' w$ by definition of $\preceq'$. ($\supseteq$) Towards a contradiction, assume some $v \in \min(pr(G), \preceq')$ such that $v \notin pr(F * G)$ (again note that also $v \in pr(G)$ by $A1_{pr}$). By $A3_{pr}$ and the fact that $pr(G) \neq \emptyset$ there is some $w \in pr(F*G)$. From (3) we infer that $v \npreceq' w$. But the by assumption also $w \npreceq' v$. Since v and w must be incompatible by $v, w \in pr(G)$ this means $pr(F * f_{pr}(\{v, w\})) \cap \{v, w\} = \emptyset$ and by $*$ fulfilling $A1_{pr}$ even $pr(F * f_{pr}(\{v, w\})) = \emptyset$, a contradiction to $*$ satisfying $A3_{pr}$. $\square$

With Theorems 1 and 2 we have obtained a one-to-one correspondence between i-max-faithful rankings and revision operators satisfying all postulates. In particular, we can use standard revision operators from the literature which work on faithful rankings (as each faithful ranking is also i-max-faithful) to get concrete revision operators. To exemplify

the obtained result, we consider Dalal's operator (1988), customized to the three-valued setting (using the same distance measure as, for instance, in (Arieli, 2008)):

Definition 5. Given an ADF F and semantics σ, the ranking $\preceq_F^\sigma$ based on three-valued distance is defined as

$$v_1 \preceq_F^\sigma v_2 \Leftrightarrow \min_{v \in \sigma(F)} (v \triangle v_1) \leq \min_{v \in \sigma(F)} (v \triangle v_2).$$

for each $v_1, v_2 \in \mathcal{V}$. The operator $*_\sigma^D$ induced by $\preceq_F^\sigma$ returns $F *_\sigma^D G = f_{pr}(\min(\sigma(G), \preceq_F^\sigma))$ for each $G \in \mathcal{F}_A$.

It is easy to see that $\preceq_F^\sigma$ is i-max-faithful, as the minimal distance to $\sigma(F)$ is 0 for interpretations $v \in \sigma(F)$ and greater than 0 for interpretations $v \notin \sigma(F)$. Hence, by Theorem 1, $*_\sigma^D$ satisfies all postulates.

Example 3. Consider the ADF $F = \{\langle a, a \rangle, \langle b, a \rangle, \langle c, \neg a \wedge b \rangle\}$, having $pr(F) = \{\mathbf{ttf}, \mathbf{fff}\}$. First note that the minimal elements of $\preceq_F^{pr}$ coincide with $pr(F)$, i.e. $\mathbf{ttf} \approx_F^{pr} \mathbf{fff} \prec_F^{pr}$ others. Now consider the revision by the ADF G having $pr(G) = \{\mathbf{tft}, \mathbf{ttu}, \mathbf{ffu}\}$ and observe that $\mathbf{ttu} \approx_F^{pr} \mathbf{ffu} \prec_F^{pr} \mathbf{tft}$ ($\mathbf{ttu}$ and $\mathbf{ffu}$ have minimal distance to $pr(F)$ of $\frac{1}{2}$, while $\mathbf{tft}$ has 2). Therefore we get $F *_{pr}^D G = f_{pr}(\{\mathbf{ttu}, \mathbf{ffu}\})$. On the other hand consider the ADF $G' = \{\langle a, \top \rangle, \langle b, \neg a \rangle, \langle c, \neg b \rangle\}$, having $pr(G') = \{\mathbf{tft}\}$. The revision of F by G' obviously results in an ADF also having $\mathbf{tft}$ – minimal distance 2 to $pr(F)$ – as only preferred interpretation. Inspecting the set of admissible interpretations of G', which can be seen as reasonable (but not maximal) positions in the revising ADF, $ad(G') = \{\mathbf{tft}, \mathbf{tfu}, \mathbf{tuu}, \mathbf{uuu}\}$, we observe that it contains elements which are closer to $pr(F)$ than $\mathbf{tft}$. In particular, the interpretation $\mathbf{tuu}$ has distance 1 to $pr(F)$ and is even admissible in F.

3.2 Revision under Admissible Semantics

Example 3 suggests to take the admissible interpretations into account when revising with respect to the preferred interpretations. A quite radical step would be to just revise with respect to admissible interpretations instead. By the fact that $ad(F_1) \cap ad(F_2) \neq \emptyset$ for all ADFs $F_1, F_2 \in \mathcal{F}_A$ we get only one operator satisfying postulate $A2_{ad}$ and the following result immediately follows:

Theorem 3. An operator $* : \mathcal{F}_A \times \mathcal{F}_A \mapsto \mathcal{F}_A$ fulfills $A1_{ad} - A6_{ad}$ iff $*$ is defined as $F * G = f_{ad}(ad(F) \cap ad(G))$.

It is important to note that admissible semantics is closed under intersection (cf. Proposition 1), therefore $f_{ad}(ad(F) \cap ad(G))$ always realizes $ad(F) \cap ad(G)$.

Example 4. Again consider the ADFs F and G' from Example 3 and note that $ad(F) = \{\mathbf{ttf}, \mathbf{fff}, \mathbf{ttu}, \mathbf{tuf}, \mathbf{ffu}, \mathbf{tuu}, \mathbf{fuu}, \mathbf{uuu}\}$ and $ad(G') = \{\mathbf{tft}, \mathbf{tfu}, \mathbf{tuu}, \mathbf{uuu}\}$. Moreover, let $*_{ad}$ be the operator from Theorem 3. As expected, we get $F *_{ad} G' = f_{ad}(\{\mathbf{tuu}, \mathbf{uuu}\})$, i.e. the resulting ADF has $\mathbf{tuu}$ as single preferred interpretation, which was somehow seen as one of the more desired scenarios in Example 3.

But now consider the ADF G'' having $ad(G'') = \{\mathbf{utf}, \mathbf{uuu}\}$ and observe that $F *_{ad} G'' = f_{ad}(\{\mathbf{uuu}\})$. From the perspective of the preferred interpretations of F (being $\{\mathbf{ttf}, \mathbf{fff}\}$) this might not be desired, as $\mathbf{utf}$ is admissible in G'' and has a distance of only $\frac{1}{2}$ to $pr(F)$, while the result of the revision has distance $\frac{3}{2}$.

3.3 Hybrid Approach

Due to the problem illustrated in Example 4 we are interested in operators selecting out of the admissible interpretations of the revising ADF (in a sense accepting all reasonable positions as valid outcomes of the revision), but basing the amount of change on the preferred interpretations of the original ADF. To this end we reformulate the postulates to this setting:

(P1) $pr(F \star G) \subseteq ad(G)$.

(P2) If $pr(F) \cap ad(G) \neq \emptyset$, then $pr(F \star G) = pr(F) \cap ad(G)$.

(P3) If $ad(G) \neq \emptyset$, then $pr(F \star G) \neq \emptyset$.

(P4) If $ad(G) = ad(H)$, then $pr(F \star G) = pr(F \star H)$.

(P5) $pr(F \star G) \cap ad(H) \subseteq pr(F \star f_{ad}(ad(G) \cap ad(H)))$.

(P6) If $pr(F \star G) \cap ad(H) \neq \emptyset$, then $pr(F \star f_{ad}(ad(G) \cap ad(H))) \subseteq pr(F \star G) \cap ad(H)$.

(Acyc) If for $1 \leq i < n$, $pr(F * G_{i+1}) \cap ad(G_i) \neq \emptyset$ and $pr(F*G_1) \cap ad(G_n) \neq \emptyset$ then $pr(F*G_n) \cap ad(G_1) \neq \emptyset$.

As admissible semantics may give pairwise compatible interpretations, we will not restrict ourselves to i-max-faithful rankings for the representation result. However, we face another challenge, as illustrated in the following example.

Example 5. Consider the ranking $\mathbf{ff} \prec others \prec \mathbf{tu} \approx \mathbf{ut} \prec \mathbf{tt} \prec \mathbf{uu}$ and the ADFs $F = \{\langle a, \bot\rangle, \langle b, \bot\rangle\}$, $G = \{\langle a, \top\rangle, \langle b, \top\rangle\}$, and $H = \{\langle a, \neg a \vee b\rangle, \langle b, a \vee \neg b\rangle\}$. We have $pr(F) = \{\mathbf{ff}\}$, $ad(G) = \{\mathbf{uu}, \mathbf{ut}, \mathbf{tu}, \mathbf{tt}\}$, and $ad(H) = \{\mathbf{uu}, \mathbf{tt}\}$. It can be seen that $\preceq$ is a faithful ranking for F. However, the revision operator $\star$ induced by $\preceq$ gives us $F \star G = f_{pr}(\{\mathbf{ut}, \mathbf{tu}\})$ and we further get

- $pr(F \star G) \cap ad(H) = \{\mathbf{uu}\}$, but

- $pr(F \star f_{ad}(ad(G) \cap ad(H))) = \{\mathbf{tt}\}$.

Therefore $\star$ violates $P5$. The problem is somehow hidden in the fact that $\mathbf{ut}$ and $\mathbf{tu}$ are compatible. That is, the set of interpretations $\{\mathbf{ut}, \mathbf{tu}\}$ cannot be realized under preferred semantics, hence $pr(f_{pr}(\{\mathbf{ut}, \mathbf{tu}\})) = \{\mathbf{uu}\}$.

To overcome this issue we introduce the concept of compliance, generalizing similar notions from (Delgrande *et al.*, 2013; Delgrande and Peppas, 2015; Diller *et al.*, 2015).

Definition 6. A preorder $\preceq$ is σ-τ-compliant if, for every ADF $F \in \mathcal{F}_A$, $\min(\tau(F), \preceq)$ is realizable under σ.

In general, this condition depends on the concrete capabilities in terms of realizability of σ and τ. Fortunately, we can capture *pr-ad*-compliance with conditions on the ranking.

Proposition 3. A preorder $\preceq$ is *pr-ad*-compliant iff: if $v_1, v_2 \in \mathcal{V}$ are compatible and $v_1 \approx v_2$ then $\exists v_3 \in cl(v_1, v_2) : v_3 \prec v_1, v_2$.

We will make use of the following properties of the adm-closure in the following results.

Lemma 1. For each $V, V_1, V_2 \subseteq \mathcal{V}$ and $v, v' \in \mathcal{V}$ it holds:

1. $cl(V) = cl(cl(V))$ (idempotence)

2. $V_1 \subseteq V_2 \Rightarrow cl(V_1) \subseteq cl(V_2)$ (monotonicity)

3. $\forall v'' \in cl(v, v') : cl(v, v'') \subseteq cl(v, v')$.

Proof. Note that $V \subseteq cl(V)$ for any $V \subseteq \mathcal{V}$ is clear by definition. (1) $cl(V) \subseteq cl(cl(V))$ follows from the initial observation. Assume there is some $v \in cl(cl(V))$ with $v \notin cl(V)$. The latter means that $\exists a \in (v^{\mathbf{t}} \cup v^{\mathbf{f}}) \exists v_2 \in [v]_2$ s.t. $\nexists v' \in V : v' \leq_i v_2 \wedge v'(a) = v(a)$. Now for this particular a and v_2 it holds, by $v \in cl(cl(V))$, that $\exists w \in cl(V) : w \leq_i v_2 \wedge w(a) = v(a)$. Hence $\exists w' \in V : w' \leq_i v_2 \wedge w'(a) = w(a)$. We have $w'(a) = w(a) = v(a)$, a contradiction.
(2) Let $v \in cl(V_1)$ and consider some $a \in v^{\mathbf{t}} \cup v^{\mathbf{f}}$ and $v_2 \in [v]_2$. There is some $v' \in V_1$ s.t. $v' \leq_i v_2$ and $v(a) = v'(a)$. As $V_1 \subseteq V_2$ by assumption, also $v \in V_2$, hence $v \in cl(V_2)$.
(3) Consider some $v'' \in cl(v, v')$, i.e. $\forall a \in v''^{\mathbf{t}} \cup v''^{\mathbf{f}} \forall v_2 \in [v'']_2 (v \leq_i v_2 \wedge v(a) = v''(a)) \vee (v' \leq_i v_2 \wedge v'(a) = v''(a))$. Assume there is some $w \in cl(v, v'')$ and $w \notin cl(v, v')$. The latter means that $\exists a \in w^{\mathbf{t}} \cup w^{\mathbf{f}} \exists w_2 \in [w]_2$ s.t. $\neg(v \leq_i w_2 \wedge v(a) = w(a)) \wedge \neg(v' \leq_i w_2 \wedge v'(a) = w(a))$. Hence, by $w \in cl(v, v'')$, we get for this particular a and w_2 that $v'' \leq_i w_2$ and $v''(a) = w(a)$. From $a \in w^{\mathbf{t}} \cup w^{\mathbf{f}}$ and $v''(a) = w(a)$ it follows that $a \in v''^{\mathbf{t}} \cup v''^{\mathbf{f}}$ and from $v'' \leq_i w_2$ we get $w_2 \in [v'']_2$. Therefore, from $v'' \in cl(v, v')$ and $\neg(v' \leq_i w_2 \wedge v'(a) = w(a))$, we get $v \leq_i w_2$ and $v(a) = v''(a)$ and, consequently, $v(a) = w(a)$, a contradiction. $\square$

We now show the representation result for our hybrid operators which work on the admissible interpretations of the revising ADF but basing the distance measure on the preferred interpretations of the original ADF. The first direction follows similar to Theorem 1 with the help of *pr-ad*-compliance.

Theorem 4. Let F be an ADF and $\preceq_F$ a *pr-ad*-compliant, faithful ranking for F. Define operator $\star : \mathcal{F}_A \times \mathcal{F}_A \mapsto \mathcal{F}_A$ by $F \star G = f_{pr}(\min(ad(G), \preceq_F))$. Then $\star$ satisfies postulates $P1 - P6$ and Acyc.

Theorem 5. Let $\star$ be a revision operator satisfying $P1 - P6$ and Acyc. Then there is an assignment mapping each ADF F to a faithful ranking $\preceq$ for F that is *pr-ad*-compliant and $pr(F \star G) = \min(ad(G), \preceq)$ for every ADF G.

Proof. Given a revision operator $\star$ satisfying $P1 - P6$ and *Acyc*, let F be an arbitrary ADF. We will gradually define the ranking $\preceq$ and show that it is faithful and *pr-ad*-compliant and it indeed simulates $\star$. First, we define $\preceq'$ as

$$v_1 \preceq' v_2 \Leftrightarrow v_1 \in pr(F \star f_{ad}(cl(v_1, v_2)))$$

for each $v_1, v_2 \in \mathcal{V}$. Note that $\preceq'$ is reflexive, but neither transitive nor total. This is because there might be interpretations $v_1, v_2 \in \mathcal{V}$ for which $pr(F \star f_{ad}(cl(v_1, v_2))) \cap \{v_1, v_2\} = \emptyset$ due to $cl(v_1, v_2) \supset \{v_1, v_2\}$. After showing three properties of $\preceq'$ we will extend it first to the transitive $\preceq^t$ and then to the desired ranking $\preceq$.

for $v_1, v_2 \in \mathcal{V}$ s.t. $v_1 \preceq' v_2, G \in \mathcal{F}_A$:
$$v_1 \in ad(G) \wedge v_2 \in pr(F \star G) \Rightarrow v_1 \in pr(F \star G) \quad (6)$$

Let $G \in \mathcal{F}_A$, $v_1 \in ad(G)$, $v_2 \in pr(F \star G)$ with $v_1 \preceq' v_2$. First, we get $v_2 \in ad(G)$ from $P1$. Moreover, from $P5$ and $P6$ we get $pr(F \star G) \cap cl(v_1, v_2) = pr(F \star f_{ad}(ad(G) \cap cl(v_1, v_2)))$. As both $v_1, v_2 \in ad(G)$ we get that $cl(v_1, v_2) \subseteq ad(G)$ from Lemma 1.2, hence $pr(F \star G) \cap cl(v_1, v_2) = pr(F \star f_{ad}(cl(v_1, v_2)))$. Now as $v_1 \preceq' v_2$ by assumption it must hold that $v_1 \in pr(F \star f_{ad}(v_1, v_2))$, hence $v_1 \in pr(F \star G)$.

We proceed with

$$\text{for } G \in \mathcal{F}_A : \min(ad(G), \preceq') = pr(F \star G) \qquad (7)$$

($\subseteq$): To the contrary, assume some $v_1 \in \min(ad(G), \preceq')$ with $v_1 \notin pr(F \star G)$. From $P3$ we know $pr(F \star G) \neq \emptyset$, so assume an arbitrary $v_2 \in pr(F \star G)$. From (6) we follow that $v_1 \not\preceq' v_2$ and, consequently, from $v_1 \in \min(ad(G, \preceq')$ also $v_2 \not\preceq' v_1$. By the definition of $\preceq'$ and considering $P3$ there must then be some $v_3 \in pr(F \star f_{ad}(cl(v_1, v_2)))$. From $P1$ it follows that $v_3 \in f_{ad}(cl(v_1, v_2))$, i.e. $v_3 \in cl(v_1, v_2)$. Then from $P5$ and $P6$ we get $pr(F \star f_{ad}(cl(v_1, v_2))) \cap cl(v_1, v_3) = pr(F \star f_{ad}(cl(v_1, v_2) \cap cl(v_1, v_3)))$. From Lemma 1.3 it follows that $cl(v_1, v_3) \subseteq cl(v_1, v_2)$, hence $pr(F \star f_{ad}(cl(v_1, v_2))) \cap cl(v_1, v_3) = pr(F \star f_{ad}(cl(v_1, v_3)))$. Therefore $v_1 \notin pr(F \star f_{ad}(cl(v_1, v_3)))$ and $v_3 \in pr(F \star f_{ad}(cl(v_1, v_3)))$, hence $v_3 \prec' v_1$. Finally, note that $cl(v_1, v_2) \subseteq ad(G)$, hence $v_3 \in ad(G)$ contradicting $v_1 \in \min(ad(G, \preceq')$. ($\supseteq$): Let $v_1 \in pr(F \star G)$ and consider an arbitrary $v_2 \in ad(G)$. Observing $v_1 \in ad(G)$ by $P1$ we get $pr(F \star G) \cap cl(v_1, v_2) = pr(F \star f_{ad}(ad(G) \cap cl(v_1, v_2)))$ by $P5$ and $P6$. Moreover, $cl(v_1, v_2) \subseteq ad(G)$ by Lemma 1.2, hence $pr(F \star G) \cap cl(v_1, v_2) = pr(F \star f_{ad}(cl(v_1, v_2)))$ and, consequently, $v_1 \in pr(F \star f_{ad}(cl(v_1, v_2)))$, meaning $v_1 \preceq' v_2$. Therefore, recalling that v_2 was chosen arbitrarily, $v_1 \in \min(ad(G), \preceq')$.

The following can be shown similarly as (1).

$$\text{for } v_1, \dots, v_n \in \mathcal{V} : v_1 \preceq' \cdots \preceq' v_n \preceq' v_1 \Rightarrow v_1 \preceq' v_n \quad (8)$$

Now we define $\preceq^t$ to be the transitive closure of $\preceq'$. As a consequence of (8) we infer

$$\text{for } v_1, v_2 \in \mathcal{V} : v_1 \prec' v_2 \Rightarrow v_1 \prec^t v_2 \qquad (9)$$

Defining, for any set of interpretations V, $\max(V, \preceq^t)$ as the set $\{v_1 \in V \mid \not\exists v_2 \in V : v_1 \prec^t v_2\}$ we get, by (8) and the fact that $\mathcal{V}$ is finite, that

$$\text{for } V \subseteq \mathcal{V} : V \neq \emptyset \Rightarrow \max(V, \preceq^t) \neq \emptyset \qquad (10)$$

We are now ready to define $\preceq$. To this end consider the sequence of sets of interpretations $V_0, V_1, \dots$ defined as

$$V_0 = \max(\mathcal{V}, \preceq^t),$$
$$V_1 = \max\left(\mathcal{V} \setminus V_0, \preceq^t\right),$$
$$V_i = \max\left(\mathcal{V} \setminus \bigcup_{0 \le j < i} V_j, \preceq^t\right) \text{ for } i > 1.$$

Since $\mathcal{V}$ is finite we conclude from (10) that the sequence will reach the empty set of interpretations at some point and each of the following elements will also be empty. The sequence $V_1, \dots, V_m$ of non-empty sets of interpretation then forms a partition of $\mathcal{V}$. Based on this we define $\preceq$ as

$$v_1 \preceq v_2 \Leftrightarrow \exists V_i, V_j \text{ s.t. } v_1 \in V_i, v_2 \in V_j, i \ge j$$

for each $v_1, v_2 \in \mathcal{V}$. It is easy to see that $\preceq$ is total, reflexive, and transitive. Its minimal elements coincide with $\preceq'$:

$$\text{for } G \in \mathcal{F}_A : \min(ad(G), \preceq) = \min(ad(G), \preceq') \quad (11)$$

Let V_k be the last set in the sequence $V_0, \dots, V_m$ such that $V_k \cap ad(G) \neq \emptyset$. By definition of $\preceq$, $\min(ad(G), \preceq) = V_k \cap ad(G)$. Hence we have to show that $V_k \cap ad(G) = $

$\min(ad(G), \preceq')$. ($\subseteq$): Assume there is some $v \in V_k \cap ad(G)$ such that $v \notin \min(ad(G), \preceq')$. From the latter it follows that $\exists v_0 \in ad(G) : v_0 \prec' v$. From (9) we get $v_0 \prec^t v$, hence $v_0 \notin \max(V_k, \preceq^t)$. As V_k is the last set with $V_k \cap ad(G) \neq \emptyset$ it must hold that $v_0 \in V_j$ with $j < k$, i.e. $v_0 \in \max(\mathcal{V} \setminus \bigcup_{0 \le i < j} V_i, \preceq^t)$. Therefore, recalling $v_0 \prec^t v, v \notin \mathcal{V} \setminus \bigcup_{0 \le i < j} V_i$, contradicting $v \in V_k$ and $j < k$. ($\supseteq$): Assume there is some $v_0 \in \min(ad(G), \preceq')$ such that $v_0 \notin V_k \cap ad(G)$. That means $v_0 \in ad(G)$ and $v_0 \notin V_k$ and further that $v_0 \in V_j$ for $j < k$. Now let $v_1 \in V_k \cap ad(G)$. As $j < k$ hence $v_1 \in \mathcal{V} \setminus \bigcup_{0 \le i < j} V_i$. Since v_0 is maximal wrt. $\preceq^t$ in this set, $v_0 \not\prec^t v_1$ and further by (9) $v_0 \not\prec' v_1$. It holds that $v_0 \in pr(F \star f_{ad}(cl(v_0, v_1)))$ and therefore $v_0 \preceq' v_1$ though. We show this by assuming, towards a contradiction, that $v_0 \notin pr(F \star f_{ad}(cl(v_0, v_1)))$. Hence $v_0 \not\preceq' v_1$. As $v_0 \in \min(ad(G), \preceq')$ by assumption, then also $v_1 \not\preceq' v_0$. By $P3$ there has to be some $v_2 \in pr(F \star f_{ad}(cl(v_0, v_1)))$. As also $v_2 \in cl(v_0, v_2)$ we get by $P5$ and $P6$ that $v_2 \in pr(F \star f_{ad}(cl(v_0, v_1) \cap cl(v_0, v_2)))$. From Lemma 1.2 we infer that $cl(v_0, v_2) \subseteq cl(v_0, v_1)$, hence $v_2 \in pr(F \star f_{ad}(v_0, v_2))$, meaning that $v_2 \preceq' v_0$. Moreover, $v_0 \notin pr(F \star f_{ad}(v_0, v_2))$, hence even $v_2 \prec' v_0$. As $v_2 \in ad(G)$ from $v_0, v_1 \in ad(G)$ and $cl(v_0, v_1) \subseteq cl(ad(G)) = ad(G)$, we get a contradiction to $v_0 \in \min(ad(G), \preceq')$. Now consider an arbitrary $v_3 \in \mathcal{V} \setminus \bigcup_{0 \le i < j} V_i$ such that $v_1 \preceq^t v_3$. From $v_0 \preceq' v_1 \preceq^t v_3$ we get $v_0 \preceq^t v_3$. But since $v_0 \in \max(\mathcal{V} \setminus \bigcup_{0 \le i < j} V_i, \preceq^t)$ it must also hold that $v_3 \preceq^t v_0$, meaning, together with $v_0 \preceq' v_1$, that $v_3 \preceq^t v_1$. As v_3 was chosen arbitrarily we have that $v_1 \in \max(\mathcal{V} \setminus \bigcup_{0 \le i < j} V_i, \preceq^t)$, i.e. $v_1 \in V_j$, a contradiction to $v_1 \in V_k$ and $j < k$.

The fact that $\preceq$ indeed simulates $\star$ is now obtained from (7) and (11): We get that $pr(F \star G) = \min(ad(G), \preceq)$ for each ADF G. This also makes $\preceq$ *pr-ad*-compliant. To show that $\preceq$ is faithful for F assume $pr(F) \neq \emptyset$ (otherwise faithfulness is trivial). By $P2$ it holds that $pr(F \star f_{ad}(\mathcal{V})) = pr(F)$, hence $pr(F) = \min(\mathcal{V}, \preceq)$, meaning that (i) $v_1 \approx v_2$ for $v_1, v_2 \in pr(F)$ and (ii) $v_1 \prec v_2$ for $v_1 \in pr(F)$ and $v_2 \notin pr(F)$. $\quad \square$

With the insights from Theorems 4 and 5 we obtain concrete operators from faithful and *pr-ad*-compliant rankings. For instance, a valid operator is induced from the ranking $\preceq_F$ where $pr(F)$ are the minimal elements and all other interpretations form a $\prec_F$-chain. The three-valued version of Dalal's operator (cf. Definition 5) is not directly applicable here, as $\preceq_F^{pr}$ does not yield a *pr-ad*-compliant ranking for every ADF:

Example 6. Consider the ADF $F = \{\langle a, a \wedge b \rangle, \langle b, a \wedge b \rangle\}$ having $pr(F) = \{\mathbf{tt}, \mathbf{ff}\}$. It yields the ranking $\mathbf{tt} \approx_F^{pr} \mathbf{ff} \prec_F^{pr} \mathbf{tu} \approx_F^{pr} \mathbf{ut} \approx_F^{pr} \mathbf{uf} \approx_F^{pr} \mathbf{fu} \prec_F^{pr} \mathbf{tf} \approx_F^{pr} \mathbf{ft} \approx_F^{pr} \mathbf{uu}$. Now consider the compatible interpretations $\mathbf{tu}$ and $\mathbf{uf}$ and observe that all $v \in cl(\mathbf{tu}, \mathbf{uf}) = \{\mathbf{uu}, \mathbf{tu}, \mathbf{ut}, \mathbf{tf}\}$ have $v \not\approx_F^{pr} \mathbf{tu}, \mathbf{uf}$. Therefore, according to Proposition 3, $\preceq_F^{pr}$ is not *pr-ad*-compliant. In practice, this means that $F *_{pr}^D G$, where $ad(G) = \{\mathbf{uu}, \mathbf{tu}, \mathbf{uf}, \mathbf{tf}\}$, would yield $f_{pr}(\{\mathbf{tu}, \mathbf{uf}\})$; but as $\{\mathbf{tu}, \mathbf{uf}\}$ is not realizable under *pr* we do net get the preferred interpretations prescribed by the postulates.

A refinement of the distance measure in order to result in *pr-ad*-compliant rankings is subject to future work.

4 Discussion

Summary. We have characterized operators for the revision of ADFs under preferred semantics. Using recent insights on realizability we showed that rankings giving rise to concrete operators underlie milder conditions than in classical AGM revision (i-max-faithful versus faithful). We have exemplified these results by a three-valued version of Dalal's operator. While admissible semantics yield a single rational operator, we have proposed an alternative family of revision operators combining admissible and preferred semantics. Their representation by rankings is based on *pr-ad*-compliance.

Other semantics. First consider complete semantics and recall that there might be ADFs F and G such that their common complete interpretations might not be realizable under complete semantics. Therefore, when revising F by G, it is impossible to satisfy $A2_{co}$ since it would require the resulting ADF to have exactly $co(F) \cap co(G)$ as complete interpretations. The same applies to grounded semantics. Therefore it holds that for $\sigma \in \{co, gr\}$ there is no operator $* : \mathcal{F}_A \times \mathcal{F}_A \mapsto \mathcal{F}_A$ satisfying $A1_\sigma - A6_\sigma$.

For supported models, on the other hand, we observe that they have the same expressiveness as propositional logic, therefore results from classical AGM revision carry over.

Finally, stable models have similar sufficient conditions for realizability as preferred semantics, namely that a set of interpretations V is realizable if (1) $V \subseteq st(F)$ for some $F \in \mathcal{F}_A$, (2) $V = \{v_1, v_2\}$ for $v_1, v_2 \in \mathcal{V}^2$ with v_1^t and v_2^t being $\subseteq$-incomparable, or (3) $V = \{v\}$ with $v \in \mathcal{V}^2$. Therefore we expect to get similar representation results as for preferred semantics, just with slightly different conditions on the ranking.

Future work. While in this work we only dealt with the semantic outcome of operators, we also plan to study syntactic aspects of revision. Moreover, we want to study the computational complexity of Dalal's operator under preferred semantics, given that the complexity of reasoning tasks in ADFs is studied comprehensively (Strass and Wallner, 2015; Gaggl *et al.*, 2015). Finally, we want to see how gained insights carry over to the revision of AFs: operators combining preferred and admissible semantics as well as revision under three-valued semantics (Caminada and Gabbay, 2009).

References

Latifa Al-Abdulkarim, Katie Atkinson, and Trevor J. M. Bench-Capon. A methodology for designing systems to reason with legal cases using abstract dialectical frameworks. *Artif. Intell. Law*, 24(1):1–49, 2016.

Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: partial meet contraction and revision functions. *J. Symb. Log.*, 50(2):510–530, 1985.

Ofer Arieli. Distance-based paraconsistent logics. *Int. J. Approx. Reasoning*, 48(3):766–783, 2008.

Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Giovanni Guida. AFRA: Argumentation framework with recursive attacks. *Int. J. Approx. Reasoning*, 52(1):19–37, 2011.

Trevor J. M. Bench-Capon and Paul E. Dunne. Argumentation in artificial intelligence. *Artif. Intell.*, 171(10-15):619–641, 2007.

Richard Booth. Judgment aggregation in abstract dialectical frameworks. In *Advances in KR, Logic Programming, and Abstract Argumentation*, volume 9060 of *LNCS*, pages 296–308, 2015.

Gerhard Brewka and Stefan Woltran. Abstract dialectical frameworks. In *Proc. KR*, pages 102–111, 2010.

Gerhard Brewka, Stefan Ellmauthaler, Hannes Strass, Johannes P. Wallner, and Stefan Woltran. Abstract dialectical frameworks revisited. In *Proc. IJCAI*, pages 803–809, 2013.

Gerhard Brewka, Sylwia Polberg, and Stefan Woltran. Generalizations of Dung frameworks and their role in formal argumentation. *IEEE Intelligent Systems*, 29(1):30–38, 2014.

Martin Caminada and Dov M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2):109–145, 2009.

Claudette Cayrol and Marie-Christine Lagasquie-Schiex. On the acceptability of arguments in bipolar argumentation frameworks. In *Proc. ECSQARU*, pages 378–389, 2005.

Sylvie Coste-Marquis, Sébastien Konieczny, Jean-Guy Mailly, and Pierre Marquis. On the revision of argumentation systems: minimal change of arguments statuses. In *Proc. KR*, pages 72–81, 2014.

Mukesh Dalal. Investigations into a theory of knowledge base revision. In *Proc. AAAI*, pages 475–479, 1988.

James P. Delgrande and Pavlos Peppas. Belief revision in Horn theories. *Artif. Intell.*, 218:1–22, 2015.

James P. Delgrande, Pavlos Peppas, and Stefan Woltran. AGM-Style Belief Revision of Logic Programs under Answer Set Semantics. In *Proc. LPNMR*, pages 264–276, 2013.

Martin Diller, Adrian Haret, Thomas Linsbichler, Stefan Rümmele, and Stefan Woltran. An extension-based approach to belief revision in abstract argumentation. In *Proc. IJCAI*, pages 2926–2932, 2015.

Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995.

Paul E. Dunne, Wolfgang Dvořák, Thomas Linsbichler, and Stefan Woltran. Characteristics of multiple viewpoints in abstract argumentation. *Artif. Intell.*, 228:153–178, 2015.

Marcelo A. Falappa, Alejandro J. García, Gabriele Kern-Isberner, and Guillermo R. Simari. On the evolving relation between belief revision and argumentation. *Knowledge Eng. Review*, 26(1):35–43, 2011.

Sarah A. Gaggl, Sebastian Rudolph, and Hannes Strass. On the computational complexity of naive-based semantics for abstract dialectical frameworks. In *Proc. IJCAI*, pages 2985–2991, 2015.

Hirofumi Katsuno and Alberto O. Mendelzon. Propositional knowledge base revision and minimal change. *Artif. Intell.*, 52(3):263–294, 1991.

Sanjay Modgil. Reasoning about preferences in argumentation frameworks. *Artif. Intell.*, 173(9-10):901–934, 2009.

Sylwia Polberg. Extension-based semantics of abstract dialectical frameworks. In *Proc. STAIRS*, pages 240–249, 2014.

Jörg Pührer. Realizability of three-valued semantics for abstract dialectical frameworks. In *Proc. IJCAI*, pages 3171–3177, 2015.

Hannes Strass and Johannes P. Wallner. Analyzing the computational complexity of abstract dialectical frameworks via approximation fixpoint theory. *Artif. Intell.*, 226:34–74, 2015.

Hannes Strass. Approximating operators and semantics for abstract dialectical frameworks. *Artif. Intell.*, 205:39–70, 2013.

Hannes Strass. Expressiveness of two-valued semantics for abstract dialectical frameworks. *J. Artif. Intell. Res.*, 54:193–231, 2015.