Working Papers

of the

IJCAI-2013 Workshop on

Weighted Logics for Artificial Intelligence

WL4AI-2013

August 4-5, 2013 - Beijing (China)
Preface

Logics provide a formal basis for the study and development of applications and systems in Artificial Intelligence. In the last decades there has been a rapidly increasing number of logical formalisms capable of dealing with a variety of reasoning tasks that require an explicit representation of quantitative or qualitative weights associated with classical or modal logical formulas (in a form or another).

The semantics of the weights refer to a large variety of intended meanings: belief degrees, preference degrees, truth degrees, trust degrees, etc. Examples of such weighted formalisms include probabilistic or possibilistic uncertainty logics, preference logics, fuzzy description logics, different forms of weighted or fuzzy logic programs under various semantics, weighted argumentation systems, logics handling inconsistency with weights, logics for graded BDI agents, logics of trust and reputation, logics for handling graded emotions, etc. The underlying logics range from fully compositional systems, like systems of many-valued or fuzzy logic, to non-compositional ones like modal-like epistemic logics for reasoning about uncertainty, as probabilistic or possibilistic logics, or even some combination of them.

This IJCAI 2013 workshop, WL4AI-2013, is the second workshop with this name. The first edition was successfully held last year in collocation with ECAI-2012, in Montpellier (France). As in the first workshop, the aim has been to bring together researchers to discuss about the different motivations for the use of weighted logics in AI, the different types of calculi that are appropriate for these needs, and the problems that arise when putting them at work. As a result, we are very happy to gather in this proceedings volume a very interesting set of contributions on different logical formalisms that we believe are representative of the richness of the area.

Finally, we would like to express our gratitude to:

- Prof. Mingshen Ying for having accepted to give an invited talk at this workshop.
- The program committee members for their commitment to the success of this event and for their work.
- The participants of WL4AI for the quality of their contributions.

Lluis Godo, Henri Prade and Guilin Qi
Workshop Chairs

Lluis Godo IIIA-CSIC, Spain
Henri Prade IRIT- University of Toulouse, France
Guilin Qi Southeast University, China

Program Committee

Teresa Alsinet, University of Lleida, Spain
Leila Amgoud IRIT, France
Salem Benferhat University of Artois, France
Jonathan Ben Naim IRIT, France
Carlos Chesnèvar Universidad Nacional del Sur, Argentine
Laurence Cholvy ONERA, France
Petr Cintula CSAV, Czech Republic
Martine De Cock Ghent University, Belgium
Didier Dubois IRIT, France
Francesc Esteva IIA-CSIC, Spain
Tommaso Flaminio IIA-CSIC, Spain
Angelo Gilio University of Roma, Italy
Gabriele Kern-Isberner Dortmund University, Germany
Joe Halpern Cornell University, USA
Anthony Hunter University College London, UK
Manfred Jaeger Aalborg University, Denmark
Souhila Kaci University Montpellier, France
Churn-Jung Liau Academia Sinica, Taiwan
Weiru Liu Queens University Belfast, UK
Emiliano Lorini IRIT, France
Thomas Lukasiewicz Oxford University, UK
Zongmin Ma Northeast University, China
Enrico Marchioni IRIT, France
Vilém Novák University of Ostrava, Czech Republic
Zoran Ognjanović Mathematical Institute SANU, Serbia
Manuel Ojeda-Aciego University of Mlaga, Spain
Nicola Olivetti University Paul Czanne, France
Emad Saad Advanced Intelligence Research, Egypt
Steven Schockaert Cardiff University, UK
Guillermo Simari Universidad Nacional del Sur, Argentine
Umberto Straccia ISTI-CNR, Italy
V.S. Subrahmanian University of Maryland, USA
Thomas Vetterlein University of Linz, Austria
Guojun Wang Shanxi Normal University, China
Mingsheng Ying University of Technology Sidney, Australia
Table of Contents

Invited Talk

A logic for linguistic quantifiers and its applications (abstract)
M. Ying 1

Contributed papers

Answer set programs with optional rules: a possibilistic approach
K. Bauters, S. Schockaert, M. De Cock, D. Vermeir 2

Probabilistic State Estimation in the Situation Calculus
V. Belle, H.J. Levesque 10

Relating fuzzy autoepistemic logic to fuzzy modal logics of belief
M. Blondeel, T. Flaminio, L. Godo, M. De Cock 18

Modeling Reliability Varying over Time through a Labeled Argumentative Framework
M.C.D. Budán, M.J. Gómez Lucero, G.R. Simari 26

Logical representation of beliefs in the belief functions theory
L. Cholvy 34

Recursive Attack and Support in Abstract Argumentation Frameworks
A. Cohen, S. Gottifredi, A.J. García and G.R. Simari 42

Towards Instance Query Answering for Concepts Relaxed by Similarity Measures
A. Ecke, R. Peñaloza, A.-Y. Turhan 50

A Bipolar Assertion Model for Natural Language Generation
H. Eyre, J. Lawry 57

Unifying Probability and Logic for Learning
M. Hutter, J.W. Lloyd, K.S. Ng, W.T.B. Uther 65

Prime Forms in Possibilistic Logic
G. Qi, L. Perrusel 73

Modal uncertainty logics with fuzzy neighborhood semantics
R.O. Rodríguez, L. Godo 79

Scale reasoning with fuzzy-$\mathcal{EL}^+$ ontologies based on MapReduce
Z. Zhou, G. Qi, C. Liu, P. Hitzler, R. Mutharaju 87
A logic for linguistic quantifiers and its applications

MingSheng Ying

University of Technology Sidney
Australia

Abstract

Quantifiers have the ability of summarizing the properties of a class of objects without enumerating them. This talk introduces a framework for modeling quantifiers in natural languages in which each linguistic quantifier is represented by a family of non-additive measures, and the truth value of a quantified proposition is evaluated by using Sugeno’s integral. Some elegant logical properties of linguistic quantifiers can be established in this framework. We will compare our measure-theoretic model of linguistic quantifiers with other approaches. We will further discuss the possible applications of linguistic quantifiers in soft CSPs (constraint satisfaction problems) and summarization of big data. The talk will be based on a series of papers (from IS-MVL’86 to AIJ’06) and some ongoing works.
Answer set programs with optional rules: a possibilistic approach*

Kim Bauters\(^1\) and Steven Schockaert\(^2\) and Martine De Cock\(^1\) and Dirk Vermeir\(^3\)

\(^1\) Dept. of Appl. Math., Computer Science and Statistics, Universiteit Gent, Belgium, (kim.bauters, martine.decock)@ugent.be
\(^2\) School of Computer Science & Informatics, Cardiff University, United Kingdom, s.schockaert@cs.cardiff.ac.uk
\(^3\) Department of Computer Science, Vrije Universiteit Brussel, Belgium, dvermeir@vub.ac.be

Abstract

Many problems in artificial intelligence can be encoded as answer set programs (ASP) in which some rules are uncertain. ASP programs with incorrect rules may have erroneous conclusions, but due to the non-monotonic nature of ASP, omitting a correct rule may also lead to errors. To derive the most certain conclusions from an uncertain ASP program, we thus need to consider all situations in which some, none, or all of the least certain rules are omitted. This corresponds to treating some rules as optional and reasoning about which conclusions remain valid regardless of the inclusion of these optional rules. While a version of possibilistic ASP (PASP) based on this view has recently been introduced, no implementation is currently available. In this paper we propose a simulation of the main reasoning tasks in PASP using (disjunctive) ASP programs, allowing us to take advantage of state-of-the-art ASP solvers. Furthermore, we identify how several interesting AI problems can be naturally seen as special cases of the considered reasoning tasks, including cautious abductive reasoning and conformant planning. As such, the proposed simulation enables us to solve instances of the latter problem types that are more general than what current solvers can handle.

1 Introduction

Answer Set Programming (ASP), which is based on the idea of stable models [Gelfond and Lifschitz, 1988], is a form of non-monotonic declarative programming. The basic idea of ASP is to encode a problem as a set of rules, called a program, such that the models of the program correspond with the solutions of the original problem.

However, many real-world problems are affected by uncertainty, e.g. actions may have uncertain outcomes or we may not have perfect knowledge of the state of a given system. To cope with this, many extensions to ASP have been proposed over the last few years that allow for reasoning over uncertain information [Baral et al., 2006; Nicolas et al., 2006; Bauters et al., 2009].

As an example, consider the following problem. A firm has a client calling from city C to ask for an appointment the next day. The secretary knows that the sales person is either in city A (\textit{inA}) or city B (\textit{inB}), but is more confident that he is in city A. The secretary is almost certain (resp. absolutely certain) that a sales person can get from city A (resp. city B) to city C in one day (\textit{toC}), assuming there is no road block. There are some rumors of a possible road block on the route from city A to C. Cities A and B also connect to city D, which is in city A. The secretary is almost certain (resp. absolutely certain) that a sales person is in city A than in city B.

To derive the most certain conclusion, we might choose to omit the least certain rules as in possibilistic logic. Indeed, the rule (\textit{blockAC} \leftarrow) is more likely safe to be excluded since we have a low certainty that the rule is valid, whereas a rule such as (\textit{inA} \leftarrow) may not be omitted. However, if we omit the rule (\textit{blockAC} \leftarrow), then \textit{not blockAC} can be derived in the program, which may allow us to deduce ‘\textit{toC}’ when we also know ‘\textit{inA}’. Hence, due to the non-monotonic nature of ASP, both including invalid rules and excluding valid rules may lead to errors. As such, we need to consider all situations in which some, none, or all of the least certain rules are omitted. This naturally leads to the idea of an ASP program with optional rules. Given such a program with optional rules, we want to determine whether particular conclusions hold irrespective of the inclusion of the optional rules.

This idea was recently used in [Bauters et al., 2012] to develop a semantics for Possibilistic ASP (PASP), which we present in detail in Section 2.3. However, this work did not provide implementations, nor does it make explicit the link with programs with optional rules and their applications.
In this paper, we present implementation\footnote{The prototype of the solver is available online at: \url{http://www.cwi.ugent.be/kim/pasp2asp/}.} for the decision problems of ASP programs with optional rules, which we discuss in Section 2.3. Furthermore, we show that the idea of optional rules reaches beyond programs with uncertain rules. In particular, we show how cautious abductive reasoning from logic programs\cite{Eiter et al. 1997} and conformant planning (e.g.\cite{Eiter et al. 2004}) can be seen as special cases of programs with optional rules. In this way, our solver offers the first implementation of cautious abductive reasoning from logic programs, and the first single-step simulation of cautious abductive reasoning in ASP.

The remainder of this paper is organized as follows. In Section 2 we recall some notions from answer set programming and possibility theory, as well as the possibilistic semantics for ASP programs with uncertain rules. In this section, we moreover present the decision problems that will be considered in this paper. In Section 3 we introduce a number of simulations that allow us to reduce the problem of reasoning with uncertain or optional rules to cautious and brave reasoning over classical ASP programs. In Section 4 we explain how cautious abductive reasoning and conformant planning can be reduced to the decision problems of programs with optional rules. In Section 5 we provide an overview of our implementation. Related work is discussed in Section 6 and we formulate our conclusions in Section 7.

2 Preliminaries

2.1 Answer Set Programming (ASP)

A disjunctive program is a set of disjunctive rules of the form \(l_0; \ldots; l_k \leftarrow l_{k+1}, \ldots, l_m, \text{not } l_{m+1}, \ldots, \text{not } l_n\) with \(l_i\), \(0 \leq i \leq n\), a literal, i.e. either an atom ‘\(a\)’ or ‘\(\neg a\)’ and \(k \leq m \leq n\). The operator ‘not’ denotes negation-as-failure. Intuitively, \(\text{not } l\) is true when we cannot prove \(l\). A disjunctive rule \(r\) consists of a head \(H(r) = \{l_1, \ldots, l_k\}\), interpreted as a disjunction, and a body \(B(r) = B^+(r) \cup B^-(r)\) with \(B^+(r) = \{l_{k+1}, \ldots, l_m\}\) and \(B^-(r) = \{l_{m+1}, \ldots, l_n\}\), both interpreted as conjunctions. A rule is called a fact (resp. constraint) when \(B(r) = \emptyset\) (resp. \(H(r) = \emptyset\)). When \(B^-(r) = \emptyset\) we say that \(r\) is a positive disjunctive rule. When \(|H(r)| \leq 1\) we say that \(r\) is a normal rule. A normal program is a set of normal rules. \(At(P)\) is the set of atoms in program \(P\) and \(Lit(P) = At(P) \cup \{\neg a | a \in At(P)\}\).

An interpretation \(I\) is a set of literals and is consistent when there does not exist an atom \(a\) with \(a \in I\) and \(\neg a \in I\). A consistent interpretation \(I\) is a model of a positive disjunctive rule \(r\) when \(H(r) \cap I = \emptyset\) or \(B^+(r) \subseteq I\). For a positive disjunctive program, a consistent interpretation \(I\) is a model of \(P\) when \(I\) is a model of every rule in \(P\). An answer set of the positive disjunctive program \(P\) is a minimal model w.r.t. set inclusion of \(P\). The reduct \(P^I\) of a disjunctive program \(P\) w.r.t. the interpretation \(I\) is defined as the set of rules \(P^I = \{H(r) \leftarrow B^+(r) | r \in P \text{ and } B^-(r) \cap I = \emptyset\}\). An interpretation \(I\) is an answer set of \(P\) when \(I\) is an answer set of \(P^I\). A program is a consistent program when it has at least one answer set. Finally, we write \(P \models^I l\) (resp. \(P \models^I \neg l\)) to denote that \(l\) is a brave (resp. cautious) consequence of \(P\), i.e. that an answer set \(I\) of \(P\) exists with \(l \in I\) (resp. that \(l \in I\) for all answer sets \(I\) of \(P\)).

Recall that the complexity classes \(\Sigma^p_n\) and \(\Pi^p_n\) are defined as \(\Sigma^p_0 = \Pi^p_0 = P\) with \(\Sigma^p_{i+1} = \text{NP}^{\Sigma^p_i}\) and \(\Pi^p_{i+1} = \text{coNP}^{\Sigma^p_i}\), with \(\text{NP}^{\Sigma^p_i}\) the class of problems solvable in polynomial time on a non-deterministic machine with a \(\Sigma^p_i\) oracle, i.e. a procedure that can solve \(\Sigma^p_i\) problems in constant time\cite{Papadimitriou 1994}. Brave reasoning over a disjunctive (resp. normal) program is \(\Sigma^p_2\)-complete (resp. NP-complete). Cautious reasoning over a disjunctive (resp. normal) program is \(\Pi^p_2\)-complete (resp. coNP-complete)\cite{Baral 2003}.

2.2 Possibility Theory

Possibility theory is a theory for dealing with uncertainty. Let \(\Omega\) be a (finite) universe and \(\omega \in \Omega\). A possibility distribution \(\pi\) is defined as a mapping \(\pi : \Omega \rightarrow [0, 1]\) and encodes to what extent it is plausible that \(\omega\) is the actual world. By convention, \(\pi(\omega) = 0\) means that \(\omega\) is impossible and \(\pi(\omega) = 1\) means that no available information prevents \(\omega\) from being the actual world. Possibility degrees are mainly interpreted qualitatively: when \(\pi(\omega) > \pi(\omega')\), \(\omega\) is considered more plausible than \(\omega'\). When we impose constraints on a possibility distribution, we are usually only interested in the least specific possibility distribution, i.e. the greatest possibility distribution w.r.t. the ordering \(\succ\). A possibility distribution \(\pi\) induces two uncertainty measures\cite{Dubois and Prade 1988}. The possibility measure \(\Pi\) is defined as \(\Pi(A) = \max \{\pi(\omega) | \omega \in A\}\) with \(A \subseteq \Omega\) and evaluates the extent to which a world \(\omega\) in \(A\) is consistent with the beliefs expressed by \(\pi\). The dual necessity measure \(N\) is defined as \(N(A) = 1 - \Pi(A)\) and evaluates the extent to which all possible worlds belong to \(A\).

2.3 Possible and Necessary Answer Sets

Possibilistic ASP (PASP) provides a semantics for programs with uncertain rules by interpreting weights associated with rules in terms of a necessity measure. Several proposals have already been made in the literature, including\cite{Nicolas et al. 2006, Bauters et al. 2010}. The following discussion is based on the approach from\cite{Bauters et al. 2012}. A possibilistic (positive) disjunctive (resp. normal) rule is a pair \((r, c)\) with \(r\) a (positive) disjunctive (resp. normal) rule and \(c \in [0, 1]\). A possibilistic disjunctive (resp. normal) program is a set of possibilistic disjunctive (resp. normal) rules. For a possibilistic rule \(p = (r, c)\) we use \(p^*\) to denote \(r\), i.e. the classical rule. Similarly, for a possibilistic program \(P\) we use \(P^*\) to denote the set of rules \(\{p^* | p \in P\}\). The set of all weights in a possibilistic program \(P\) is denoted \(\text{cert}(P) = \{c | p = (r, c) \in P\}\). The set of all weights in a possibilistic program \(P\) is denoted \(\text{cert}(P) = \{c | p = (r, c) \in P\}\) and we also use the extended set of weights \(\text{cert}^\ast(P) = \{c | c \in \text{cert}(P)\} \cup \{1 - c | c \in \text{cert}(P)\} \cup \{0, \frac{1}{2}, 1\}\).

Semantically, the weight \(c\) is interpreted as the certainty that the rule is valid. For a given program \(P\), all the subprograms \(P' \subseteq P\) are considered. Each subprogram \(P'\) corresponds with the assumption that omitted rules are wrong and included rules are correct. The possibility of each subprogram \(P'\) (in fact, the possibility that the corresponding as-
Proposition 1. Let \( P \) be a possibilistic disjunctive program and \( P_{brave}^\Pi(l, \lambda) \) the disjunctive program defined as \( P_{\text{basic}}(\lambda) \cup \{ \xi \rightarrow \lambda \} \). Then \( \Pi(P \models \models l) \geq \lambda \) iff \( P_{brave}^\Pi(l, \lambda) \) has a classical answer set.

Proof. The condition \( \Pi(P \models \models l) \geq \lambda \) is equivalent to \( \max \{ \pi(P) \mid P' \subseteq P \text{ and } P'' \models \models \lambda \} \geq \lambda \) which states that there is some subprogram \( P' \subseteq P \) with \( P'' \models \models l \) such that \( \pi(P) \geq \lambda \). The latter condition is equivalent to \( P_{\text{eq}} \subseteq P'' \) with \( P_{\text{eq}} = \{ r \mid (r, c) \in P, c > 1 - \lambda \} \). Thus the problem reduces to determining whether for some set of rules \( P_{\text{opt}} \subseteq \{ r \mid (r, c) \in P, c \leq 1 - \lambda \} \) we have \( P_{\text{eq}} \cup P_{\text{opt}} \models \models l \). By construction of \( P_{\text{basic}}(\lambda) \), due to the rules \( \{ \xi \rightarrow \lambda \} \), we know that every rule in \( P_{\text{eq}} \) is chosen. Furthermore, every choice made in \( \Pi \) corresponds with a choice of \( P_{\text{opt}} \). Finally, the addition of the rule \( \{ \xi \rightarrow \lambda \} \) ensures that \( \Pi' \) must be a conclusion of some answer set of the simulation \( P_{brave}^\Pi(l, \lambda) \), or otherwise \( P_{brave}^\Pi(l, \lambda) \) will not have any answer sets.

Proposition 2. Let \( P \) be a possibilistic disjunctive program, \( \lambda > 0 \) and \( P_{cautious}^\Pi(l, \lambda) \) the disjunctive program defined as \( P_{\text{basic}}(1 - \lambda') \cup \{ \xi \rightarrow \lambda' \} \) with \( \lambda' \in \text{cert}^+ (P) \) such that \( \lambda' < \lambda \) and \( \exists \lambda'' \in \text{cert}^+ (P) \cdot X < \lambda'' < \lambda \). Then \( N(P \models \models l) \geq \lambda \) iff \( P_{\text{cautious}}(l, \lambda) \) has no classical answer set.

Proof. The proof is analogous to the proof of Proposition 1. To determine whether \( N(P \models \models l) \geq \lambda \) we need to verify that \( \max \{ \pi(P) \mid P' \subseteq P \text{ and } P'' \models \models \lambda \} \leq 1 - \lambda' \). In other words, we need to verify that there does not exist a subprogram \( P'' \) such that \( \lambda'' \models \models \lambda \) and \( \pi(P'') > 1 - \lambda' \). The simulation \( P_{\text{cautious}}^\Pi(l, \lambda) \) looks for a subprogram with a certainty strictly higher than \( 1 - \lambda \) in which \( \Pi' \) is false, i.e. \( l \) is not a cautious consequence. Furthermore, note that the certainty will be strictly higher than \( 1 - \lambda' \) iff \( \Pi' \) is at least \( 1 - \lambda' \). If this does not exist, i.e. if we find no answer sets, then \( N(P \models \models l) \geq \lambda \).

3 Simulation Using ASP

We show how ASP with uncertain rules can be simulated using classical ASP. We start with the simulation of \( \Pi(P \models \models l) \geq \lambda \) and \( N(P \models \models l) \geq \lambda \), which are the decision problems with the lowest complexity.

Definition 1. Let \( P \) be a possibilistic disjunctive program. We define \( P_{\text{basic}}(\lambda) \) as the set of rules:

\[
\begin{align*}
&\{ r' \leftarrow \text{not } r'' \mid (r, c) \in P, c \leq 1 - \lambda \} \\
&\cup \{ r'' \leftarrow \text{not } r' \mid (r, c) \in P, c \leq 1 - \lambda \} \quad (1) \\
&\cup \{ r' \leftarrow \text{not } r'' \mid (r, c) \in P, c > 1 - \lambda \} \quad (2) \\
&\cup \{ \text{head}(r) \leftarrow \text{body}(r) \cup \{ r' \} \mid (r, c) \in P \} \quad (3)
\end{align*}
\]

Intuitively, the rules \( (1) \) generate as many answer sets as there are choices of rules for which the resulting subprogram has a possibility of at least \( \lambda \). The most certain rules are considered to be valid in \( (2) \). The information encoded in the respective rules is applied using \( (3) \).
There is a directed positive edge from \( l_1 \) to \( l_0 \) if there is a rule \( r \in P \) such that \( H(r) = l_0 \) and \( l_1 \in B^+(r) \). Similarly, there is a directed negative edge from \( l_1 \) to \( l_0 \) if there is a rule \( r \in P \) such that \( H(r) = l_0 \) and \( l_1 \in B^-(r) \). There is a positive path from \( l_1 \) to \( l_2 \) if there is a path in the dependency graph from vertex \( l_1 \) to vertex \( l_2 \) consisting only of positive edges. A normal program \( P \) is said to be tight if it does not contain a positive cycle, i.e., a positive path starting and ending in a vertex \( l \) [Lin and Zhao, 2003].

Not every ASP program, however, is tight. As such, we will need to rely on more complex translations of ASP programs to sets of clauses, such as the translation based on a characterization in terms of level numbering presented in [Jannhunen, 2004]. Once we have the translation to a set of clauses, we generate answer sets for every subprogram and we apply saturation techniques to both validate whether a given interpretation is a valid model of the subprogram and to verify whether a given literal is a desired conclusion of the given subprogram.

**Definition 2.** Let \( P \) be a possibilistic normal program. The disjunctive program \( P_{\text{complex}}(\lambda) \) is defined as the set of rules:

\[
\{ t' \leftarrow \neg \neg t' \mid (r, c) \in P, c \leq 1 - \lambda \}
\]

\[
\{ t' \leftarrow \neg \neg t' \mid (r, c) \in P, c \leq 1 - \lambda \}
\]

\[
\{ t' \leftarrow (r, c) \in P, c > 1 - \lambda \}
\]

\[
\{ cl \leftarrow cl \in \text{cls}(P^r) \}
\]

\[
\{ \text{sat} \leftarrow a, na \mid a \in \text{at}(\text{cls}(P^r)) \}
\]

\[
\{ a \leftarrow \text{sat} \mid a \in \text{at}(\text{cls}(P^r)) \}
\]

\[
\{ \text{na} \leftarrow \text{sat} \mid a \in \text{at}(\text{cls}(P^r)) \}
\]

\[
\{ \neg \leftarrow \text{not sat} \}
\]

\[
\{ cl_1' \leftarrow cl \in \text{cls}(P^r) \}
\]

\[
\{ a', na' \mid a \in \text{at}(\text{cls}(P^r)) \}
\]

where \( \text{cls}(P) \) is a representation of the normal program \( P \) as a set of clauses (e.g. [Jannhunen, 2004]). \( P^r \) is the set of rules \( \{ \text{head}(r) \leftarrow \text{body}(r), r \in P \} \) with 'r' a fresh atom. \( C^i \) is the set of clauses obtained from \( C \) by replacing every occurrence of a negated atom \( \neg a \) with a fresh atom \( na \) except for the atoms \( r_i \) and \( at(C) \) is the set of atoms appearing in \( C \) from which we remove the atoms \( r_i \). Finally, \( cl_1' \) is obtained from a clause \( cl \) by replacing every literal from \( \text{Lit}_P \) with \( l' \).

**Proposition 3.** Let \( P \) be a possibilistic normal program and \( P_{\text{clus}}(l, \lambda) \) the disjunctive program defined as \( \Pi(\lambda) \cup \{ \text{sat} \leftarrow l \} \). Then \( \Pi(P \models^\lambda l) \geq \lambda \) iff \( P_{\text{clus}}(l, \lambda) \) has a classical answer set.

**Proof.** We want to determine whether \( \Pi(P \models^\lambda l) \geq \lambda \), i.e., whether there exists a \( P' \subseteq P \) such that \( P'^r \models^\lambda l \) and \( \pi_\lambda(P'^r) \geq \lambda \). The latter condition means that \( (r, c) \in P' \) for every \( (r, c) \in P'^r \) with \( c > 1 - \lambda \). Similar as in Proposition 3, the rules in (5) and (6) generate as many answer sets as there are subprograms \( P' \subseteq P \) for which \( \pi_\lambda(P'^r) \geq \lambda \).

For each such subprogram \( P' \) we want to determine whether \( P'^r \) has 'l' as a cautious conclusion. By construction, \( \{ cl \leftarrow cl \in \text{cls}(P^r) \} \) is equivalent to \( P^r \). In particular, every model of these rules corresponds to an answer set of \( P^r \). Since we removed classical negation in (6), however, we need to add the rules in (7) to ensure that 'sat' is contained in the answer set whenever 'a' and the opposite atom 'na' are true at the same time. The intuition of making 'sat' true is thus to indicate that this is not a valid answer set of the subprogram \( P^r \). The rule (8) is then used to block all answer sets in which 'sat' is false. In other words: unless for every answer set of \( P^r \) we have that 'l' is true in the answer set, we have not found that 'l' is a cautious consequence of \( P' \).

Thus far we have not discussed the use of the rules (8). Together with the atom 'sat', these rules are used to implement a saturation technique [Baral, 2003] over our disjunctive simulation and we refer to this work for a detailed overview of how saturation works. The intuition of saturation is that we use the property that an answer set is a minimal model. In particular, the rules in (8) will add all the atoms under consideration to the model \( M \) to try and prevent it from being an answer set. Indeed, if we find a model \( M' \subseteq M \) then clearly \( M \) cannot be an answer set. As such, we can ensure that consistent models of \( P' \) are preferred over inconsistent models, and that models of \( P' \) in which 'l' is false are preferred over models in which 'l' is true. Then, only if no consistent answer set (in which 'l' is false) exists for \( P' \), will we have that 'sat' is true in an answer set of \( P_{\text{clus}}(l, \lambda) \).

Finally, when a subprogram \( P' \) is inconsistent, then \( \pi(P') = 0 \), i.e. we do not want to consider this subprogram. Notice, however, that the rule (7) would not work as expected in this case. Indeed, if \( P' \) is inconsistent it does not have a consistent model and the saturation technique would not exclude this subprogram. As such, we repeat our simulation of the subprogram \( P' \) in (10) and use constraints in (11) to effectively block inconsistent subprograms.

**Proposition 4.** Let \( P \) be a possibilistic normal program and \( P_{\text{clus}}(l, \lambda) \) the disjunctive program defined as \( \Pi(\lambda) \cup \{ \text{sat} \leftarrow \text{not l} \} \) with \( \lambda \) defined as in Proposition 3. Then \( N(P \models^\lambda l) \geq \lambda \) iff \( P_{\text{clus}}(l, \lambda) \) has no classical answer set.

**Proof.** This proof is analogous to the proof of Proposition 3 similar as how the proof of Proposition 2 was analogous to the proof of Proposition 1.

## 4 Certain programs with optional rules

Our simulations for the decision problems \( \Pi(P \models^\lambda l) \geq \lambda \) and \( N(P \models^\lambda l) \geq \lambda \) are not only useful for reasoning with uncertain answer set programs, but can be applied to the much wider problem range of programs with optional rules. In particular, in this section we prove how two interesting AI problems, namely cautious abductive reasoning and conformant planning, can be expressed in terms of programs with optional rules. As such, we can solve these problems with the
simulation offered in Section 3 and this, in turn, forms the first implementation of cautious abductive reasoning and the first single-step implementation of conformant planning in ASP. Both problems can also trivially be extended with weights.

### 4.1 Cautious abductive reasoning

An abductive diagnosis program [Eiter et al., 1997] is encoded as a triple \( (H, T, O) \) where \( H \) is a set of literals referred to as hypotheses, \( T \) is a (normal) ASP program referred to as the theory and \( O \) is a set of literals referred to as observations. Intuitively, the theory \( T \) describes the dynamics of a system, the observations \( O \) describe the observed state of the system and the hypotheses \( H \) are those literals that can be used to try and explain such observations within the theory. Cautious abductive reasoning is concerned with the problem of finding hypotheses that could explain the observations in \( O \).

Let \( \Pi \) be the possibilistic normal program defined for an abductive diagnosis program \( (H, T, O) \) as

\[
\begin{align*}
0.5 \cdot \text{block} \cdot h_+ & \iff h \in H \\
\cup \{1: h \iff \neg \text{block} \cdot h+ | h \in H \} \\
\cup \{1: \text{goal} \iff O \} \\
\cup \{1: r | r \in T \}.
\end{align*}
\]

It holds that \( \langle H, T, O \rangle \) has a cautious explanation iff

\[
\Pi (P_{\text{abd}} \models \text{goal}) \geq 0.5. \quad \text{In particular, } E \text{ is a cautious explanation iff for } P' = P_{\text{abd}} \setminus \{ \text{block} \cdot h_+ | h \in E \} \text{ we have that } P' \models \text{goal}. \]

**Proof.** For \( \Pi (P_{\text{abd}} \models \text{goal}) \geq 0.5 \), we must have some \( P' \subseteq P_{\text{abd}} \) such that \( P' \models \text{goal} \) and \( \pi(P') \geq 0.5 \). Thus, clearly, all the rules defined in (13), (14) and (15) must be in \( P' \). It is furthermore easy to see that for every \( h \in H \setminus E \) we have that \( (h +) \in (P')^M \) for every answer set \( M \) of \( P' \) and, since \( P^+ \models \text{goal} \), that \( E \) is a cautious explanation. \( \square \)

**Example 2.** John wants to become rich. He can either choose to invest his money in stocks, or to invest it in bonds (he lacks the money to do both). He can either win or fail with his investment, where he resp. becomes rich or bankrupt. Bonds are safer and will make him rich, eventually. Whether or not he should invest will also depend on how certain he is that his investment will succeed (very certain) or fail (somewhat certain) and his confidence that investing in bonds will make him rich (somewhat certain):

\[
\begin{align*}
0.8: & \text{win} \iff \neg \text{not win}, \text{invest} \\
0.5: & \text{fail} \iff \neg \text{win}, \text{invest} \\
1: & \text{rich} \iff \text{win} \\
1: & \text{bankrupt} \iff \neg \text{fail} \\
0.5: & \text{rich} \iff \text{bonds} \\
1: & \iff \text{invest}, \text{bonds}
\end{align*}
\]

Given the hypotheses \( H = \{ \text{invest}, \text{bonds} \} \), it is clear that when we ignore the weights only \( E = \{ \text{bonds} \} \) is a cautious abductive explanation for the observation \( O = \{ \text{rich} \} \). If we take the certainties into account and \( \lambda = 0.5 \), then both \( E_1 = \{ \text{bonds} \} \) and \( E_2 = \{ \text{invest} \} \) are cautious explanations.

### 4.2 Conformant planning

Conformant planning is the problem of determining whether a plan (i.e. a series of actions) exists that always leads to the desired goal, regardless of the incompletely known initial state of the agent. Such problems are typically expressed using an action language.

An action language is built from a finite number of fluents \( f_1, \ldots, f_n \). A state is a finite set of fluents. The properties of the initial state \( s_0 \) are described by formulas of the form \( \text{initial } f_i \), which are called value propositions, with \( f \) a fluent literal, i.e. a fluent or a fluent preceded by \( \neg \). Changes of states are defined using a finite number of actions \( a_1, \ldots, a_j \). Formulas of the form \( \text{a } f_i \) are called effect propositions, with \( f_i \) fluent literals and \( a \) an action. A domain \( D \) is a finite set of value and effect propositions. A proper domain, to which we limit ourselves in this paper, is a domain in which we can determine in polynomial time what the successor state is, given the current state and an action. A plan is a sequence of actions \( [a_1, \ldots, a_m] \). The planning problem is to determine for a given domain \( D \) and a fluent literal \( f \) whether a plan exists leading from \( s_0 \) to a state in which \( f \) is true, where we call \( f \) the goal fluent. To solve a planning problem, the domain is translated to ASP. Particularly, such a translation can be written as \( P_{\text{act}} \cup P_{\text{rem}} \) where \( P_{\text{act}} \) are those rules used to describe the actions, whereas \( P_{\text{rem}} \) are the remaining rules that among others describe the (incomplete) initial state and rules to ensure inertia. Then, a plan exists when an answer set contains the goal fluent.

However, not all forms of planning problems can be solved in this way. When we say that we have an incomplete domain, this means that the initial values of some fluents are unknown. Conformant planning is the problem of determining whether for an incomplete domain and a fluent \( f \) a plan exists leading to a state in which \( f \) is true, regardless of the initial values of the unknown fluents. Only some action languages, e.g. \( K \) [Eiter et al., 2000] and \( DLV^K \) [Eiter et al., 2004], have the expressive power to describe conformant planning problems. For solving such problems, \( DLV^K \) relies on a two-step translation to ASP where a plan is generated (that is not necessarily a conformant plan) and verified to be an actual conformant plan, until an actual conformant plan is found. However, these methods are not designed to work with uncertainty and cannot, e.g. compute the most reliable plan when no conformant plan can be found.

We now show how conformant planning can be expressed in terms of a decision problem of the form \( N (P \models \text{goal}) \geq \lambda \). Note that the existence of a conformant plan can be also written as \( \exists P \forall iv \cdot P(p, iv, pp) \) where \( P(p, iv, pp) \) describes that for the planning problem \( pp \) and for all initially unknown values \( iv \) the plan \( p \) leads to the goal fluent.

**Proposition 6.** Let \( P_{\text{can}} \) be the possibilistic normal program defined for a conformant planning problem with the atom ‘goal’ the desired goal fluent. We express the domain knowledge as a normal ASP program \( P_{\text{act}} \cup P_{\text{rem}} \). Then \( P_{\text{can}} \) is:

\[
\begin{align*}
0.5: & \text{block} \cdot i_+ \iff r_i \in P_{\text{act}} \\
\cup \{1: H(r_i) \iff B(r_i) \cup \neg \text{block} \cdot i_+ | r_i \in P_{\text{act}} \} \\
\cup \{1: r | r \in P_{\text{rem}} \}
\end{align*}
\]
A conformant plan exists if $\Pi(P_{con} \models \text{goal}) \geq 0.5$.

Proof. When $\Pi(P_{con} \models \text{goal}) \geq 0.5$ then, by definition, there exists a subprogram $P' \subseteq P$ such that $(P')^* \models \text{goal}$ with $\pi(P') \geq 0.5$. Since $\pi(P') \geq 0.5$ we know that all the rules from (17), (18) and (19) are in $P'$. Thus, only rules from (16) may be in $P \setminus P'$. In that case, the corresponding rule from (17) ensures that for every answer set $M$ of $(P')^*$ we have that $H(r_i) \leftarrow B(r_i) \in ((P')^*)^M$. Thus, the action is no longer blocked and can be applied. Because of the available actions we can, regardless of the initial state described in (18), cautiously derive ‘goal’. Indeed, otherwise we know due to (19) that $M$ is not be a model. In other words: the choice made in (16) corresponds with a set of actions that form a cautious plan for the given planning problem. \qed

Example 3. Consider the example from the introduction where we add additional action rules (e.g. driveAtOC). Clearly, by adding certainties, we can compute a conformant plan with a lower certainty, even when a classical conformant program can be seen as a set of uncertain rules. Thus, only rules from (16) may be in $P \setminus P'$. In that case, the corresponding rule from (17) ensures that for every answer set $M$ of $(P')^*$ we have that $H(r_i) \leftarrow B(r_i) \in ((P')^*)^M$. Thus, the action is no longer blocked and can be applied. Because of the available actions we can, regardless of the initial state described in (18), cautiously derive ‘goal’. Indeed, otherwise we know due to (19) that $M$ is not be a model. In other words: the choice made in (16) corresponds with a set of actions that form a cautious plan for the given planning problem.

5 Implementation

We have implemented an operational prototype of the simulation presented in Section 3. The translator PASP2SAT prepares an input PASP program by removing the certainties and adding the fresh atom $r_i$ to the body of each rule $r_i \in P$. The resulting ASP program is converted to an equivalent set of clauses using the technique from [Janhunen, 2004]. The output in DIMACS CNF form is converted back into ASP rules by CLAUSE2ASP. The rules (6) -- (11) are constructed from the information of the translation to clauses, while $\lambda$ is needed to add the rules (12) -- (15) and $l$ is required to add the rules that are specific to the simulation of the decision problem as identified in Proposition 3 and 4. An overview of the implementation is given in Figure 1. The answer sets of the resulting ASP program can then be computed using an ASP solver for disjunctive programs, e.g. DLV [Eiter et al., 1999].

\[ \{ 1 : \leftarrow \text{not goal} \} \] (19)

For $BMTUC(2, 4)$, $BMTUC(3, 4)$ and $BMTUC(4, 4)$ both our solver and DLV$^{K}$ were able to find a secure plan in less than 1 second. For $BMTUC(5, 4)$, within one second, a secure plan could still be found by DLV$^{K}$ but not by our solver. For $BMTUC(6, 4)$ neither of the solvers was able to find a solution within 1 second. For other variants of the Bomb in Toilet problem, a similar pattern was observed. Although our solver is somewhat slower than DLV$^{K}$, it is competitive on most problem instances. Moreover, while DLV$^{K}$ is optimized for the problem of conformant planning, our solver is more generic and can thus exploit problem-specific heuristics. Moreover, in these experiments we did not consider optimization, based on the tightness of the programs.

6 Related Work

The combination of uncertainty with logic programming has been widely studied. One of the earliest results on how to combine probability theory and logic programming is [Lukasiewicz, 2002]. Later, in [Baral et al., 2009], a framework was proposed in which ASP is combined with probability theory. This framework, probabilistic atoms are used in addition to classical ASP to describe the probability that the atom will take on a random value given the prior knowledge.

The work in [Dubois et al., 1994] on possibilistic logic is one of the first approaches to combine possibility theory with logic programming. The first work on combining ASP and possibility theory was [Nicolas et al., 2006], which introduced the PASP framework. In this approach, ‘not $l$’ is treated as it is more certain that ‘$l$’. In [Bauters et al., 2010], an alternative was presented, where ‘not $l$’ is interpreted as the degree to which it is possible that ‘$l$’. Essentially, both approaches can be seen as a multi-valued logic, using the negation from Gödel logic and Łukasiewicz logic, respectively. These two approaches have in common that the weights associated with rules are used to obtain weighted answer sets. The approach from [Bauters et al., 2012], however, which we have adopted in this paper, uses the weights to obtain a weighted set of classical answer sets. As such, a program can be seen as a set of uncertain rules.

To the best of our knowledge, the approach presented in this paper is the first implementation of cautious abductive reasoning. Brave abductive reasoning, on the other hand, has seen a myriad of implementations and many solvers for answer set programming, e.g. [Leone et al., 2006] Gebser et al., 2011, have incorporated very performant mechanisms for brave abductive reasoning. Interestingly, [Dubois and Prade, 1995] illustrated how abductive reasoning can benefit from possibility theory, allowing to order the possible cautious explanations. Also of interest is the work from [Medinaus et al., 2001] on abduction in multi-adjoint logic programs. In multi-adjoint logic programs, it is possible to associate confidence factors with the rules. Furthermore, it allows to specify for each rule the type of the implication, i.e. Gödel, Łukasiewicz or the product. Still, no practical implementation has been suggested for this particular type of abduction.

Conformant planning, which is also known under other names such as secure planning and strong planning, has also...
seen a lot of interest. For a fixed plan length, conformant planning is a $\Sigma_2^P$-complete problem and secure checking, i.e. verifying whether a plan is a secure plan, is a $\Pi_2^P$-complete problem. If we restrict ourselves to proper planning domains, then conformant planning and secure checking is $\Sigma_2^P$-complete and coNP-complete, respectively. Many implementations of conformant planning exist, including C-Plan [Ferraris and Giunchiglia, 2000], CMBT [Cimatti and Roveri, 2004], Conformant-PE [Hoffmann and Brafman, 2006], and DLV [Eiter et al., 2004], where the latter is an ASP-based approach. The latter approach, in particular, is an ASP-based approach in which the planning problem is expressed in the action language $K$. Contrary to our approach, however, these implementations do not allow certainties.

7 Conclusions

We have considered the problem of reasoning with uncertain answer set programs and have provided the first implementations for each of the main reasoning tasks. Taking advantage of recent progress on efficient translations from ASP to SAT, our approach translates an answer set program with uncertain rules into a classical (possibly disjunctive) answer set program, such that the answer sets of the latter program correspond to solutions of the original problem. We showed the practical significance of our implementation, beyond managing uncertainty in ASP, by showing how cautious abductive reasoning and conformant planning can be naturally seen as special cases of the considered problem. To the best of our knowledge, our approach is the first implementation of cautious abductive reasoning from ASP programs and the first implementation of conformant planning that is exclusively based on a translation to a single classical ASP program.

References


Probabilistic State Estimation in the Situation Calculus

Vaishak Belle and Hector J. Levesque
Dept. of Computer Science
University of Toronto
Toronto, Ontario M5S 3H5, Canada
{vaishak, hector}@cs.toronto.edu

Abstract

Probabilistic state estimation tasks are fundamental reasoning concerns in robotic applications, where the world is uncertain, and sensors and effectors are noisy. Most systems make various assumptions about the dependencies between state variables, and especially about how these dependencies change as a result of actions. Building on a general framework by Bacchus, Halpern and Levesque for reasoning about degrees of belief in the situation calculus, and a recent extension to it for continuous domains, in this paper we illustrate probabilistic state estimation in the presence of a rich theory of actions using an example. We also show that while actions might affect prior distributions in nonstandard ways, suitable posterior beliefs are nonetheless entailed as a side-effect of the overall specification.

1 Introduction

Intelligent agents operating in dynamic and uncertain worlds face two major sorts of reasoning problems. First, because the world is dynamic, actions perpetually change the properties of the state. Second, because little in the world is definite, the agent has to modify its beliefs based on the actions it performs and its sensor measurements, both of which are prone to noise. To the best of our knowledge, the most general formalism for dealing with probabilistic belief in formulas, and how that should evolve in the presence of noisy acting and sensing, is a logical account by Bacchus, Halpern and Levesque (BHL) [1999]. In the BHL approach, besides quantifiers and other logical connectives, one has the provision for specifying the degrees of belief in formulas in the initial state. This specification may be compatible with one or very many initial distributions and sets of independence assumptions. All the properties of belief will then follow at a corresponding level of specificity.

Subjective uncertainty is captured in the BHL scheme using a possible-world model of belief [Fagin et al., 1995]. In classical possible-world semantics, a formula $\phi$ is believed to be true when $\phi$ comes out true in all possible worlds that are deemed accessible. In BHL, the degree of belief in $\phi$ is defined as a normalized sum over the possible worlds where $\phi$ is true of some nonnegative weights associated with those worlds. To reason about belief change, the BHL model is then embedded in a rich theory of action and sensing provided by the situation calculus [McCarthy and Hayes, 1969; Reiter, 2001; Scherl and Levesque, 2003]. The BHL account provides axioms in the situation calculus regarding how the weight associated with a possible world changes as the result of acting and sensing. The properties of belief and belief change then emerge as a direct logical consequence of the initial specifications and these changes in weights. In a nutshell, the formalism has the many advantages of logical representations, and is also equipped to handle belief change wrt noisy sensors and effectors in a manner similar to, but significantly more general than, probabilistic techniques such as Kalman filtering and Dynamic Bayesian Networks [Dean and Kanazawa, 1989; Dean and Wellman, 1991]. The BHL scheme, however, is limited to fluents whose values are drawn from discrete countable domains. In recent work [Belle and Levesque, 2013a], we show how with minimal additional assumptions this serious shortcoming of BHL can be lifted.

In this paper, we illustrate how that generalized BHL scheme may be utilized for probabilistic state estimation tasks using an example with various subtleties. Imagine a robot moving towards a wall, and at a certain distance $h$ to the right of the wall, as in Figure 1. The robot might initially believe that $h$ is drawn from a uniform distribution on $[2, 12]$. Among the robot’s many capabilities, we imagine the ability of moving left. A leftwards motion of 1 unit would shift the uniform distribution on $h$ to $[1, 11]$, but a leftward motion of 4 units would change the distribution more radically. The point $h=0$ would now obtain a weight of $.2$, while $h \in (0, 8]$ would retain their densities. This mixed distribution would then be

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{robot.png}
\caption{Robot operating in a 2-dimensional world, and moving towards a wall with a window.}
\end{figure}

preserved by a subsequent rightward motion. Suppose further there are two onboard sensors: a sonar unit aimed at the wall estimating \( h \), and a GPS (global positioning system) device sensing both \( h \) and the robot’s vertical position, \( v \). Each of these might be characterized by Gaussian error models, and the effect of a reading from any sensor would revise the distribution on \( h \) from uniform to an appropriate Gaussian. The robot is now left with the difficult task of adjusting its beliefs as it moves and obtains competing (perhaps conflicting) measurements from individual sensors. Lastly, in addition to the above continuous fluents, imagine the following discrete variable. Suppose there is a window on the wall, and the robot has a sensor to sense whether the window is open or closed. This sensor might be noisy but the exact nature of this noise might depend on how far the robot is from the wall. We show here that belief changes appropriately when one assumes, as above, continuous and discrete fluents, nontrivial actions that shift distributions, situation-specific error profiles, among other complications. More precisely, suitable posterior beliefs are simply entailed by the domain formalization. Since no assumptions need to be made in general regarding the kind of distributions that initial state variables are drawn from, nor about dependencies between state variables, this work illustrates how beliefs would change after acting and sensing in complex uncertain domains.

The paper is structured as follows. In the next section, we briefly review formal preliminaries, such as the situation calculus, the BHL scheme as well as the essentials of its generalization to continuous domains. We then model the robot domain and illustrate properties about belief change. In the final sections, we discuss related and future work. In the paper, sensors are allowed to be noisy but physical actions are assumed to be deterministic for simplicity.

2 Preliminaries

The language \( \mathcal{L} \) of the situation calculus [McCarthy and Hayes, 1969] is a many-sorted dialect of predicate calculus, with sorts for actions, situations and objects (for everything else, and includes the set of reals \( \mathbb{R} \) as a subset). A situation represents a world history as a sequence of actions. A set of initial situations correspond to the ways the world might be initially. Successor situations are the result of doing actions, where the term \( do(a,s) \) denotes the unique situation obtained on doing \( a \) in \( s \). The term \( do(a,s) \), where \( a \) is the sequence \( [a_1,\ldots,a_n] \) abbreviates \( do(a_n,\ldots,do(a_1,s)) \). Initial situations are defined as those without a predecessor:

\[
\text{Init}(s) \equiv \neg\exists a,a',s',s = do(a,a').
\]

We let the constant \( S_0 \) denote the actual initial situation, and we use the variable \( i \) to range over initial situations only.

In general, the situations can be structured into a set of trees, where the root of each tree is an initial situation and the edges are actions. In dynamical domains, we want the values of predicate and functions to vary from situation to situation. For this purpose, \( \mathcal{L} \) includes \textit{fluents} whose last argument is always a situation. For example, if \( f \) is a fluent, we understand \( f(s) < 12 \) as “\( f < 12 \) at situation \( s' \)”. Here we assume without loss of generality that all fluents are functional.

Basic action theory

Following [Reiter, 2001], we model dynamic domains in \( \mathcal{L} \) by means of a \textit{basic action theory} \( \mathcal{D} \), which consists of

1. axioms \( D_0 \) that describe what is true in the initial states, including \( S_0 \);
2. precondition axioms that describe the conditions under which actions are executable;
3. successor state axioms that describe the changes to fluents on executing actions;
4. domain-independent \textit{foundational} axioms, the details of which need not concern us here. See [Reiter, 2001].

An agent reasons about actions by means of the entailments of \( \mathcal{D} \), for which standard Tarskian models suffice. We assume henceforth that models \textit{also} assign the usual interpretations to \( =,\lt,\gt,0,1,+,/,-,\pi \) and \( x' \) (exponentials).

Likelihood and degree of belief

The BHL model of belief builds on a treatment of knowledge by Scherl and Levesque [2003]. Here we present a simpler variant based on just two distinguished binary fluents \( l \) and \( p \).

The term \( l(a,s) \) is intended to denote the likelihood of action \( a \) in situation \( s \). For example, suppose sonar\( (z) \) is the action of reading the value \( z \) from a sensor that measures the distance to the wall, \( h \). We might assume that this action is characterized by a Gaussian error model:

\[
l(\text{sonar}(z),s) = u \equiv (z \geq 0 \land u = \mathcal{N}(z - \mu;\sigma^2)) \lor (z < 0 \land u = 0)
\]

which stipulates that the difference between a nonnegative reading of \( z \) and the true value \( h \) is normally distributed with a variance of \( \sigma^2 \) and mean of \( \mu \). In general, the action theory \( \mathcal{D} \) is assumed to contain for each action \( A \) an additional \textit{action likelihood axiom} of the form

\[
l(A(x),s) = u \equiv \phi_A(x,u,s)
\]

where \( \phi_A \) is a formula that characterizes the conditions under which action \( A \) has likelihood \( u \) in \( s \). (Actions that have no sensing aspect should be given a likelihood of 1.)

Next, the \( p \) fluent determines a probability distribution on situations. The term \( p(s',s) \) denotes the relative weight accorded to situation \( s' \) when the agent happens to be in situation \( s \). The properties of \( p \) in initial states, which vary from domain to domain, are specified by axioms as part of \( D_0 \). The following nonnegative constraint is also included in \( D_0 \):

\[
\forall i,s.\, p(s,i) \geq 0 \land (p(s,i) > 0 \Rightarrow \text{Init}(s)) \tag{P1}
\]

except that worlds are not part of the syntax.

As usual, free variables in any of these axioms should be understood as universally quantified from the outside.

Alternatively, one could specify axioms for characterizing the field of real numbers in \( \mathcal{D} \). Whether or not reals with exponentiation is first-order axiomatizable remains a major open question.

Naturally, we assume that the value \( z \) being read is not under the agent’s control. See BHL for a precise rendering of this nondeterminism in terms of GOLOG operators [Reiter, 2001].

Note that \( \mathcal{N} \) is a continuous distribution involving \( \pi, e \), exponentiation, and so on. Therefore, BHL always consider discrete probability distributions that approximate the continuous ones.
While this is a stipulation about initial states only, BHL provide a successor state axiom for \( p \), and show that with an appropriate action likelihood axiom, the nonnegative constraint then continues to hold everywhere:

\[
p(s', do(a, s)) = u \equiv \\
\exists s'' [s' = do(a, s'') \land Poss(a, s'') \land \\
\neg \exists s'' [s' = do(a, s'') \land Poss(a, s'') \land u = p(s'', s) \times l(a, s'')] \\
\vee \neg \exists s'' [s' = do(a, s'') \land Poss(a, s'') \land u = 0]
\]

(P2)

Now if \( \phi \) is a formula with a single free variable of sort situation, then the **degree of belief** in \( \phi \) is simply defined as the following abbreviation:

\[
Bel(\phi, s) = \frac{1}{\gamma} \sum_{(s', \phi \in \{s\})} p(s', s)
\]

(B)

where \( \gamma \), the normalization factor, is understood throughout as the same expression as the numerator but with \( \phi \) replaced by \( true \). For example, here \( \gamma \) is \( \sum_{s} p(s', s) \). We do not have to insist that \( s' \) and \( s \) share histories since \( p(s', s) \) will be 0 otherwise. BHL show how summations can be expressed using second-order logic, see the appendix. That is, neither \( Bel \)'s definition nor summations are special axioms of \( D \), but simply convenient abbreviations for logical terms. To summarize, in the BHL scheme, an action theory consists of:

1. \( D_0 \) as before, but now also including (P1);
2. precondition axioms as before;
3. successor state axioms as before, but now also including one for \( p \) viz. (P2);
4. foundational domain-independent axioms as before; and
5. action likelihood axioms.

**From sums to integrals**

While the definition of belief in BHL has many desirable properties, it is defined in terms of a *summation* over situations, and therefore precludes fluents whose values range over the reals. The continuous analogue of (B) then requires integrating over some suitable space of values.

As it turns out, a suitable space can be found. First, some notation. We use a form of conditional *if-then-else* expressions, by taking some liberties with notation and the scope of variables as follows. We write \( f = If \exists x. \phi \) Then \( t_1 \) Else \( t_2 \) to mean the logical formula

\[
f = u \equiv \exists x. (\phi \land (u = t_1)) \vee [(u = t_2) \land \neg \exists x. \phi]
\]

Now, assume that there are \( n \) fluents \( f_1, \ldots, f_n \) in \( L \), and that these take no arguments other than a situation. Next, suppose that there is exactly one initial situation for any vector of fluent values [Levesque et al., 1998]:

\[
[\forall x \exists t. f_i(t) = x_i] \land [\forall t, t'. \exists t. f_i(t) = f_i(t') \supset t = t'] \quad (*)
\]

Under these assumptions, it can be shown that the summation over all situations in (B) can be recast as a summation over all possible initial values \( x_1, \ldots, x_n \) for the fluents:

\[
Bel(\phi, s) = \frac{1}{\gamma} \sum_{x} P(\bar{x}, \phi, s)
\]

(B')

where \( P(\bar{t}, \phi, s) \) is the (unnormalized) weight accorded to the successor of an initial world where \( f_i \) equals \( t_i \):

\[
P(\bar{x}, \phi, do(a, S_0)) = \\
If \exists x. \land f_i(t) = x_i \land \phi[do(a, t)] \\
Then \quad p(do(a, t), do(a, S_0))
\]

Else 0

where \( a \) is an action sequence. In a nutshell, because every situation has an initial situation as an ancestor, and because there is a bijection between initial situations and possible fluent values, it is sufficient to sum over fluent values to obtain the belief even for non-initial situations. Note that unlike (B), this one expects the final situation term \( do(a, S_0) \) mentioning what actions and observations took place to be explicitly specified, but that is just what one expects when the agent reasons about its belief after doing things, and for the projection problem in particular [Reiter, 2001].

The generalization to the continuous case then proceeds as follows. First, we observe that some (though possibly not all) fluents will be real-valued, and that \( p(s', s) \) will now be a measure of *density* not weight. Similarly, the \( P \) term above now measures (unnormalized) density rather than weight.

Now suppose fluents are partitioned into two groups: the first \( k \) take their values \( x_1, \ldots, x_k \) from \( \mathbb{R} \), while the rest take their values \( y_{k+1}, \ldots, y_n \) from countable domains, then the *degree of belief* in \( \phi \) is an abbreviation for:

\[
Bel(\phi, s) = \frac{1}{\gamma} \int \sum_{y_n} P(\bar{y}, \phi, s)
\]

The belief in \( \phi \) is obtained by ranging over all possible fluent values, and integrating and summing the densities of situations where \( \phi \) holds. In [Belle and Levesque, 2013a], it is shown that belief change in the formalism is identical, under certain conditions, to Bayesian conditioning over continuous variables [Pearl, 1988]. This is precisely what is desired to capture standard probabilistic belief update mechanisms.

The appendix shows how integrals can be formulated using second-order quantification. That is, as before, \( Bel, P, \) integrals and sums are simply convenient abbreviations, and do any set, including infinite ones. In probabilistic terms, this would correspond to having a joint probability distribution over infinitely many, perhaps uncountably many, random variables. We know of no existing work of this sort, and we have as yet no good ideas about how to deal with it.

We are assuming here that the density function is (Riemann) integrable. If it is not, belief is clearly not defined, nor should it be. Similarly, if the normalization factor is 0, which corresponds to the case of conditioning on an event that has 0 probability, belief should not be (and is not) defined.
not involve special axioms in $D$. More precisely, the continuous extension to BHL has the same components from earlier, with a single revision:

1. $D_{0}$ additionally includes $(*)$.

Note that likelihood axioms are specified as before, but we will no longer have to approximate Gaussian error models (or any other continuous models) as would BHL.

### 3 Probabilistic State Estimation

We build a basic action theory $D$ for a robot in a 2-dimensional grid. We imagine three fluents $h, v$ and $w$ in addition to $Poss, l$ and $p$. The fluent $h$ gives the distance to the wall, $v$ gives the position of the robot along the vertical axis and $w$ gives the status of the window: $w = 1$ indicates that it is open and $w = 0$ indicates that it is closed. We consider two physical actions $left(z)$ and $up(z)$, and three sensing actions $sonar(x)$, $gps(x, y)$ and $seeW(x)$.

$D_{0}$ includes the following domain-independent axioms: $(*)$ and (P1). Specific to the domain, imagine that $D_{0}$ includes the following for $p$:  

\[
 p(x, S_{0}) = \begin{cases} 
 0.1 \times N(v(x); 0, 16) \times 0.6 & \text{if } h(x) \in [2, 12], w(x) = 1 \\
 0.1 \times N(v(x); 0, 16) \times 0.4 & \text{if } h(x) \in [2, 12], w(x) = 0 \\
 0 & \text{otherwise}
\end{cases}
\]

This says that the value of $v$ is normally distributed about the horizontal axis with variance 16, and independently, the window is open with a .6 probability and the value of $h$ is uniformly distributed between 2 and 12.\(^{11}\) No other sentence is included in $D_{0}$.

For simplicity, we assume that actions are always executable. Therefore, $D$ will not contain any precondition axioms. $D$’s successor state axioms are the following. There is a fixed one for $p$, which is (P2). For $h$ suppose:

\[
h(\text{do}(a, s)) = u \equiv \\
\neg \exists z (a = \text{left}(z) \land u = h(s)) \\
\exists z (a = \text{left}(z) \land u = \max(0, h(s) - z)).
\]

This says an action $\text{left}(z)$ moves the robot $z$ units to the left (towards the wall) but that the motion stops if the robot hits the wall. It is also assumed that $\text{left}(z)$ is the only action that affects $h$. Of course, to move away from the wall, $z$ can be any negative value. Similarly, for $v$ let:

\[
v(\text{do}(a, s)) = u \equiv \\
\neg \exists z (a = \text{up}(z) \land u = v(s)) \\
\exists z (a = \text{up}(z) \land u = v(s) + z).
\]

This captures the upward motion of the robot, while assuming that $\text{up}(z)$ is the only action affecting $v$. We assume $w$ is not affected by any action:

\[
w(\text{do}(a, s)) = u \equiv u = w(s).
\]

\(^{10}\)Initial beliefs can also be specified for $D_{0}$ using Bel directly.

\(^{11}\)We model a simple distribution for illustrative purposes. In general, neither do the variables have to be independent, nor does the specification need to be complete in the sense of mentioning all the variables.

Finally, we specify the likelihood axioms in $D$. We will suppose that the sonar unit, which senses $h$, is quite accurate:

\[
l(\text{sonar}(z), s) = u \equiv \\
(z \geq 0 \land u = N(h(s) - z; 0, 0.25)) \\
\lor (z < 0 \land u = 0)
\]

which stipulates that the difference between a nonnegative reading of $z$ and the true value $h$ is normally distributed with a variance of .25 and mean of 0. (A mean of 0 indicates that there is no systematic bias in the reading.) For the GPS device, assuming that its absolute readings of latitude and longitude have been converted to relative readings [Hightower and Borriello, 2001] for $h$ and $v$, imagine a bivariate Gaussian error model:

\[
l(\text{gps}(x, y), s) = \begin{cases} 
 N(h(s) - x, v(s) - y; \mu_{1}, \Sigma) & \text{if } h(s) \geq 2 \\
 N(h(s) - x, v(s) - y; \mu_{2}, \Sigma) & \text{otherwise}
\end{cases}
\]

where $\Sigma$ is the $2 \times 2$ identity matrix, $\mu_{1} = [0 \ 0]^T$ and $\mu_{2} = [1 \ 2]^T$. This says that the components of the Gaussian are independent, and that there is systematic bias in the reading for $v$ when the robot is close to the wall (due to a signal obstructions).

For the window sensor, we assume the following context-sensitive error profile:

\[
l(\text{seeW}(z), s) = u \equiv \\
(h(s) \leq 12 \land z = 1 \land w(s) = z \land u = 0.8) \\
\lor (h(s) \leq 12 \land z = 1 \land w(s) = z \land u = 0.6) \\
\lor (h(s) > 12 \land z = 1 \land w(s) = z \land u = 0.7) \\
\lor (h(s) > 12 \land z = 1 \land w(s) = z \land u = 0.5) \\
\lor (z = 0 \land w(s) = z \land u = 0.9) \\
\lor (z = 0 \land w(s) = z \land u = 0.7) \\
\lor (z = 0 \land u = 0) \\
\lor (z \not= 0 \land u = 0)
\]

That is, the window sensor is slightly more accurate when it is sensing a closed window rather than an open one. Moreover, when at a distance of within 12 units from the wall, the sensor’s accuracy for an open window is .8, but declines to .7 when the distance between the robot and the wall is more than 12 units.

As mentioned earlier, physical actions such as $\text{left}(z)$ and $\text{up}(z)$ are assumed to be deterministic for this paper, so they are given trivial likelihoods:

\[
l(\text{left}(z), s) = 1, \\
l(\text{up}(z), s) = 1.
\]

This completes the specification of $D$.

**Theorem 1:** The following are logical entailments of $D$:

**Initial beliefs**

1. $Bel(\text{true}, S_{0}) = 1$.

2. $Bel(h = 2 \lor h = 3 \lor h = 4, S_{0}) = 0$

   Intuitively, we are to evaluate $\int_{x_{1}} \int_{x_{2}} \sum_{x_{3}} q(x_{1}, x_{2}, x_{3})$, where $h, v$ and $w$ take values $x_{1}, x_{2}$ and $x_{3}$ respectively. While $x_{1}$ and $x_{2}$ range over $\mathbb{R}$, for the given belief term the function $q(x_{1}, x_{2}, x_{3})$ is 0 unless $x_{1} \in \{2, 3, 4\}$.

3. $Bel(5 \leq h \leq 5.5, S_{0}) = .05$

   Here we evaluate a function that is 0 except when $5 \leq x_{1} \leq 5.5$. We get $\int_{h} \int_{h} \int_{h} \sum_{x_{2}} .1 \times N(x_{2}; 0, 16) \times$
5.3 (in green), and after finally reading 5.6 (in red).

\[ \delta(x_3) \, dx_1 \, dx_2 = \int_2^{x_3} \int_2^{x_3} .1 \times N(x_2; 0, 16) \times 0.6 + \int_2^{x_3} N(x_2; 0.16) \times .4 = .05, \]

where \( \delta(x_1) \) is the probability assigned to the different values for \( w \) initially.

4. \( Bel(w = 0, S_0) = 0.4 \)
   
   We evaluate the density function \( q(x_1, x_2, x_3) \) for all values of \( x_1 \) and \( x_2 \) but for \( x_3 = 0 \). Then we have
   
   \[ \int_2^{x_3} \int_2^{x_3} .1 \times N(x_2; 0, 16) \times .4 \, dx_2 \, dx_1 = 0.4. \]

**Sensing by sonar**

5. \( Bel(5 \leq h \leq 5.5, do(sonar(5.3), S_0)) \approx .38 \)

   Compared to item 3, belief is sharpened significantly by obtaining a reading of 5.3 on the highly sensitive sonar. This is because the \( p \) function incorporates the likelihood of a \( sonar(5.3) \) action. Starting with the density function in item 3, the sensor reading multiplies the expression to be integrated by \( N(x_1 - 5.3; 0, .25) \), as given by (3). This amounts to evaluating the expression
   
   \[ \int_{x_1} \int_{x_2} .1 \times N(x_2; 0, 16) \times \delta(x_3) \times N(x_1 - 5.3; 0, .25) \, dx_1 \, dx_2 \]


6. \( Bel(4.5 \leq h \leq 6.5, do(sonar(5.3), sonar(5.6)), S_0) \approx .99 \)

   Two successive readings around 5.5 sharpen belief within 1 unit of 5.5 to almost certainty. Compared to item 5, the density function is further multiplied by \( N(x_1 - 5.6; 0, .25) \), and integrated wrt \( x_1 \) over \([2, 12]\) for the denominator as usual but over \([4.5, 6.5]\) for the numerator. The changing densities are shown in Figure 2.

**Physical actions**

7. \( Bel(h = 0, do(left(4), S_0)) = .2 \)

   Here a continuous distribution evolves into a mixed one. By (1), \( h = 0 \) holds after the action if \( h \leq 4 \) held before.

8. \( Bel(h \leq 5, do(left(4), S_0)) = .7 \)

   Bel’s definition is amenable to a set of \( h \) values, where one value has a weight of .2, and all the other real values have a uniformly distributed density of .1. This change in weights is shown in Figure 3.

9. \( Bel(h = 4, do([left(4), left(-4)]), S_0)) = .2 \)

   \( Bel(h = 4, do([left(-4), left(4)]), S_0)) = 0 \)

   The point \( h = 4 \) has 0 weight initially (like in item 2). Moving leftwards first means many points “collapse”, and so this point (now having \( h \) value 0) gets .2 weight which is retained on moving away. But not vice versa.

10. \( Bel(-1 \leq v \leq 1, do(left(6), S_0)) = \)

    \[ Bel(-1 \leq v \leq 1, S_0) = \int_1^{1} N(x_2; 0, 16) \, dx_2 \]

   Owing to Reiter’s solution to the frame problem, belief in \( v \) is unaffected by a lateral motion. For \( v \in [-1, 1] \) it is the area between \([-1, 1]\) bounded by the specified Gaussian.

11. \( Bel(v \leq 1.5, do(up(3.5), S_0)) = Bel(v \leq -2, S_0) \)

   After the action \( up(3.5) \), the Gaussian for \( v \)‘s value has its mean “shifted” by 3.5 because the density associated with \( v = x_2 + 3.5 \) is initially now associated with \( v = x_2 + 3.5 \).

**Sensing by GPS**

12. \( Bel(-1 \leq v \leq 1, do(gps(5.1), S_0)) \approx .27 \)

   Compared to item 10, which evaluates to \( \approx .19 \), a GPS reading of .1 increases the posterior belief for \( v \in [-1, 1] \) to \( \approx .27 \). Using the error model, this is a result of

   \[ \int_2^{12} \int_{A} N(x_2; 0, 16) N(x_1 - 5, x_2 - .1; \mu_1, \Sigma) \, dx_2 \, dx_1 \]

   with \( A = [-1, 1] \) for the numerator and \( A = [-\infty, \infty] \) for the denominator.\(^{12}\)

**Competing sensors**

13. \( Bel(5 \leq h \leq 5.5, do([gps(5.1), gps(5.3), 1], S_0)) \approx .27 \)

   \( Bel(5 \leq h \leq 5.5, do([sonar(5.3), gps(5.1), S_0]) \approx .42 \)

   The sonar is more sensitive than the GPS, and so its reading is far more effective. Relating this to item 5, a GPS reading of 5 for \( h \) only slightly redistributes the density.

**Systematic bias**

14. \( h(S_0) \leq 4 \supset \)

    \[ Bel(-1 \leq v \leq 1, do([left(4), gps(1, 0)], S_0)) \approx 0 \]

   After moving left by 4 units, \( v \)’s reading from the GPS has a systematic bias of 2. Among other things, this entails that the belief in \( v \leq 1 \) is almost 0 which is much weaker than its prior from item 10.

**Sensing the window**

15. \( Bel(w = 1, do(seew(1), S_0)) \approx .85 \)

   At the initial situation, the window being sensed open means that the density function is multiplied by the error associated for the case \( h \leq 12 \). Essentially, we get:

   \[ \frac{1}{y} \int_2^{12} \int_{R} .1 \times N(x_2; 0, 16) \times .6 \times .8 \, dx_2 \, dx_1 \]

   where \( y \) is

   \[ \int_2^{12} \int_{R} .1 \times N(x_2; 0, 16) \times (.6 \times .8 + .4 \times .2) \, dx_2 \, dx_1. \]

\(^{12}\)This is a simple instance of Kalman filtering [Dean and Wellman, 1991] where the value being sensed is static. Gaussian distributions enjoy the conjugate property: multiplying Gaussians results in another Gaussian [Box and Tiao, 1973], and is easily computed.
16. \( \text{Bel}(w = 1, \text{do}([\text{left}(\neg 4), \text{seew}(1)], S_0)) = \cdot 82 \)

After moving away by 4 units, for those situations where \( h \leq 12 \) the sensor’s accuracy is \( .8 \), but for those situations where \( h > 12 \) the accuracy is \( .7 \). Basically, then, belief is calculated as follows:

\[
\int_{x_1} \int_{x_2} \sum_{x_3} \begin{cases} 
0.1 \times \mathcal{N}(x_2; 0, 16) \times 0.6 \times 0.8 & \text{if } x_1 \in [2, 12], \\
0.1 \times \mathcal{N}(x_2; 0, 16) \times 0.6 \times 0.7 & \text{if } x_1 \in [2, 12], \\
0.1 \times \mathcal{N}(x_2; 0, 16) \times 0.4 \times 0.2 & \text{if } x_1 \in [2, 12], \\
0.1 \times \mathcal{N}(x_2; 0, 16) \times 0.4 \times 0.3 & \text{if } x_1 \in [2, 12], \\
1 & \text{if } x_1 + 4 \leq 12, \\
1 & \text{if } x_1 + 4 > 12, \\
1 & \text{if } x_3 = 1, \\
0 & \text{if } x_3 = 0
\end{cases}
\]

For the numerator, this is evaluated for \( x_3 = 1 \) and for the denominator, this is evaluated for \( x_3 \in [0, 1] \). As expected, since the sensor is slightly less accurate when the robot moves away, it is slightly less confident (in comparison to the previous item) about the window being open after sensing.

**Nonstandard properties**

17. \( \text{Bel}(h > 7\nu, S_0) \approx 0.6 \)

Beliefs about any mathematical expression involving the random variables, even when that does not correspond to well known density functions, are entailed.

18. \( \text{Bel}([\exists a. s. \text{now} = \text{do}(a, s) \land h(s) > 1], \text{do}(\text{left}(4), S_0)) = 1. \)

It is possible to refer to earlier or later situations using \( \text{now} \) as the current situation. This says that after moving, there is full belief that \( (h > 1) \) held before the action.

## 4 Related Work

Sensor fusion has been a primary concern in state estimation approaches [Thrun et al., 2005]. Popular models include variants of Kalman filtering [Fox et al., 2003], where priors and likelihoods are assumed to be Gaussian. We already pointed out that entailment item 12 is a simple instance of Kalman filtering. But in general, our approach does not make any assumptions about the nature of distributions, nor about how distributions and dependencies may evolve after actions, and allows for strict uncertainty. This distinguishes the current method from numerous probabilistic formalisms [Lerner et al., 2002; Dean and Wellman, 1991; Fox et al., 2003], including those that handle explicit actions [Darwiche and Goldszmidt, 1994; Hajishirzi and Amir, 2010]. To the best of our knowledge, none of these formalisms have treated cases where state variables change in the manner indicated in the paper.

In contrast, probabilistic logical formalisms such as [Halpern, 1990; Bacchus, 1990] are equipped to handle features such as disjunctions and quantifiers, but they do not explicitly address actions. Relational probabilistic languages and Markov logics [Ng and Subrahmanian, 1992; Richardson and Domingos, 2006] also do not model actions. Recently, temporal extensions, such as [Choi et al., 2011], specifically treat Kalman filtering, but not complex actions. In this regard, action logics such as dynamic and process logics are closely related. Recent proposals, for example [Van Benthem et al., 2009], treat sensor fusion. However, these and related frameworks [Halpern and Tuttle, 1993; Kushmerick et al., 1995], are mostly propositional. In the last years, there have been extensions to the PDDL planning language, so as to account for probabilistic effects and partial observability [Younes and Littman, 2004; Sanner, 2011]. The focus in this literature, as well other probabilistic planning approaches [Kushmerick et al., 1995], is on certain sorts of initial databases rather than a specification that allows full first-order expressivity.

Finally, proposals based on the situation and fluent calculi are first-order [Bacchus et al., 1999; Poole, 1998; Boutilier et al., 2000; Mateus et al., 2001; Shapiro, 2005; Gabaldon and Lakemeyer, 2007; Fritz and McIlraith, 2009; Belle and Lakemeyer, 2011; Thielshcher, 2001], but none of them deal with continuous sensor noise, and nor do the extensions for continuous processes [Reiter, 2001; Herrmann and Thielshcher, 1996; Fox and Long, 2006].

## 5 Conclusions

This paper illustrates location estimation for a robot operating in an incompletely known world, equipped with noisy sensors. In contrast to a number of competing formalisms, where the modeler is left with the difficult task of deciding how the dependencies and distributions of state variables might evolve, here one need only specify the initial beliefs and the physical laws. Suitable posteriors are then entailed. The framework of the situation calculus, and a recent generalization to the BHL scheme, allows us to additionally specify situation-specific biases and realistic continuous error models. Our example demonstrates that belief changes appropriately even when one is interested in nonstandard properties, such as logical relationships of state variables, all of which emerges as a side-effect of the general specification.

In the future, we intend to consider the more elaborate case...
where a robot’s position will include Cartesian coordinates as well as the robot’s angular orientation, and explore state estimation in this setting. More broadly, on the representation side, features such as continuous time, exogenous actions, decision theory and durative actions have been proposed in the situation calculus [Reiter, 2001], which could be imported to our formalism. On the more computational side, we are interested in investigating formal conditions about action theories that allow us to estimate posteriors efficiently under the assumption that priors and likelihoods are drawn from tractable distributions [Box and Tiao, 1973]. We believe our formulation of belief is also amenable to regression [Reiter, 2001; Scherl and Levesque, 2003], and an extension of regression for degrees of belief is ongoing work. See [Belle and Levesque, 2013b] for preliminary results in this direction.

Appendix: Sums and Integrals in Logic

Logical formulas can be used to characterize sums and a variety of sorts of integrals. Here we show the simplest possible cases: the summing of a one variable function from 1 to n, and the definite integral from \(-\infty\) to \(\infty\) of a continuous real-valued function of one variable. Other complications are treated in a longer version of the paper.

First, sums. For any logical term \(t\) and variable \(i\), we introduce the following notation to characterize summations:

\[
\sum_{i=1}^{n} t = z \Leftrightarrow \exists f \left[ \left( f(1) = t_1 \wedge f(n) = z \wedge \forall j \left( 1 \leq j \leq n \Rightarrow f(j+1) = f(j) + t_{j+1} \right) \right) \right]
\]

where \(f\) is assumed to not appear in \(t\), and \(j\) is understood to be chosen not to conflict with any of the variables in \(t\) and \(i\).

Now, integrals. We begin by introducing a notation for limits to positive infinity. For any logical term \(t\) and variable \(x\), we let \(\lim_{x \to \infty} t\) stand for a term characterized by:

\[
\lim_{x \to \infty} t = z \Leftrightarrow \forall u \left( u > 0 \Rightarrow \exists m \forall n \left( n > m \Rightarrow z - t_{n+m} < u \right) \right).
\]

The variables \(u, m, \) and \(n\) are understood to be chosen not to conflict with any of the variables in \(x, t, \) and \(z\).

Then, for any variable \(z\) and terms \(a, b,\) and \(t\), we introduce a term \(\text{INT}[x, a, b, t]\) to stand for the definite integral of \(t\) over \(x\) from \(a\) to \(b\):

\[
\text{INT}[x, a, b, t] = \lim_{n \to \infty} h \cdot \sum_{i=1}^{n} t_{(a+i)}
\]

where \(h\) stands for \((b-a)/n\). The variable \(n\) is chosen not to conflict with any of the other variables.

Finally, we define the definite integral of \(t\) over all real values of \(x\) by the following:

\[
\int_{-\infty}^{\infty} t = \lim_{\epsilon \to 0} \lim_{\delta \to 0} \text{INT}[x, -\epsilon, \epsilon, t].
\]

The main result for this logical abbreviation is the following:

**Theorem 2:** Let \(g\) be a function symbol of \(L\) standing for a function from \(\mathbb{R}\) to \(\mathbb{R}\), and let \(c\) be a constant symbol of \(L\). Let \(M\) be any logical interpretation of \(L\) such that the function \(g^M\) is continuous everywhere. Then we have the following:

\[
\text{If } \int_{-\infty}^{\infty} g^M(x) \cdot dx = c^M \text{ then } M \models (c = \int g(x)).
\]

References


Relating fuzzy autoepistemic logic to fuzzy modal logics of belief

Marjon Blondeel § and Tommaso Flaminio † and Lluis Godo † and Martine De Cock ‡

Abstract
Fuzzy autoepistemic logic is a generalization of autoepistemic logic, an important formalism for nonmonotonic reasoning originally intended to model an ideally rational agent reflecting upon his own beliefs, and allows to represent an agent’s rational beliefs on partially true gradable propositions. Fuzzy autoepistemic logic has recently been shown to be a suitable logical framework for fuzzy answer set programming, generalizing a classical result. On the other hand, there are well-known links between autoepistemic logic and several nonmonotonic modal logic systems. In this paper, we introduce generalizations of the main classical propositional modal logics of belief based on finitely-valued Łukasiewicz calculus. We obtain completeness with respect to appropriate Kripke-style semantics and we prove NP-completeness for the satisfiability problem. Then we show how fuzzy autoepistemic logic can be approached in these many-valued modal settings. In particular we obtain a generalization of Levesque’s result on the relationship between stable expansions, belief sets and “only knowing” operators.

1 Introduction and motivation
Since its introduction in the 1980s, autoepistemic logic [Moore, 1983; Konolige, 1994; Shvarts, 1990] has been one of the main formalisms for nonmonotonic reasoning. It extends propositional logic by offering the ability to reason about an agent’s (lack of) beliefs. More precisely, these beliefs are sets of sentences in a propositional language augmented by a modal operator B. If \( \varphi \) is a formula, then B\( \varphi \), which has to be interpreted as “\( \varphi \) is believed”, is a formula as well. Hence, in this language nested modal operators are allowed; it is possible to have beliefs about beliefs.

Logic programming has had a significant impact on the development of nonmonotonic logics and vice versa (e.g. [Baral and Gelfond, 1994]). In particular, Gelfond and Lifschitz [Gelfond and Lifschitz, 1988] showed that there is a one-to-one correspondence between the answer sets of an answer set program and the stable expansions of a corresponding autoepistemic theory. In [Moore, 1983], a stable expansion of a set of autoepistemic formulas \( A \) is defined as a set of formulas \( E_A \) such that the following fix-point condition holds:

\[
E_A = \{ \varphi \mid A \cup \{ B\psi \mid \psi \in E_A \} \cup \{ \neg B\psi \mid \psi \notin E_A \} \vdash \varphi \},
\]

where \( \vdash \) denotes derivability in classical propositional logic and each formula B\( \psi \) is considered as a new propositional variable. Informally, a stable expansion of \( A \) is a closed set of beliefs of an ideal rational agent based on the premises \( A \).

In [Levesque, 1990], a modal logic account of main concepts of Moore’s autoepistemic logic is provided, and in particular a K45 modal logic of belief is expanded with a new unary modality \( O \) where \( O\varphi \) has to be interpreted as “\( \varphi \) is all that is believed” or “only \( \varphi \) is believed”. It is then shown that “only believing” (or “only knowing”) is closely related to stable expansions in autoepistemic logic.

Recently, a fuzzy generalization (from a semantical point of view) of autoepistemic logic has been defined in [Blondeel et al., 2013] where it is also shown that the important relation between autoepistemic logic and answer set programming is preserved: the answer sets of a fuzzy answer set program (e.g. [Van Nieuwenborgh et al., 2007]) can be equivalently determined by computing the fuzzy stable expansions of a corresponding set of fuzzy autoepistemic formulas.

In this paper, we introduce generalizations of the main classical propositional modal logics of belief (K45, KD45, S5) based on finitely-valued Łukasiewicz calculus in order to model the notion of belief on fuzzy propositions, in the sense of admitting partial degrees of truth between 0 (fully false) and 1 (fully true). For instance suppose the expression “I believe it is raining” has truth value 0.2. This is interpreted as “I believe to degree 0.2 that it is raining”, or as can be shown by the definitions we will introduce as “I (fully) believe that it is raining to at least degree 0.2”. For practical and technical reasons we consider truth degrees belonging to a finite
scale $S_k = \{0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1\}$. Then we show how fuzzy
autoepistemic logic can be approached using possible worlds
semantics corresponding to these many-valued modal logics.
We also consider the expansion of our many-valued K45 with
an “only believing” operator $O$ and show that Levesque’s re-
sult on the relationship between stable extensions, belief sets
and “only knowing” operators nicely extends to our frame-
work.

This paper is structured as follows. After this introduc-
tion, Section 2, we provide some necessary preliminaries
on the $(k + 1)$-valued Łuksiewicz logic $L_k$ and available
results on the minimal modal logic over $L_k$. In Section 3
we define proper generalizations of the classical modal systems
K45, KD45 and S5 and prove sound- and completeness with
respect to appropriate Kripke-style semantics, while in Sec-
tion 4 we deal with the complexity of these logics and prove
NP-completeness for two variants of the satisfiability prob-
lem. Then in Section 5 we consider possible world semantics
for the fuzzy autoepistemic logic of [Blondeel et al., 2013]
and provide a characterization of fuzzy stable expansions in
terms of many-valued valued K45 belief sets, and also in terms
of proper generalizations of stable sets. In Section 6, we gen-
eralize (from a semantical point of view) the propositional
fragment of Levesque’s “only knowing” logic and prove that
there is a characterization of fuzzy stable expansions in terms
of the belief sets involving the “only knowing” operator $O$.
We conclude with some final remarks about related work.

2 Background

Consider the propositional language $L$ whose formulas are
built from a countable set of propositional variables $V$, the
connective $\rightarrow$ (implication) and truth constants $\top$ for each
c $\in S_k = \{0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1\}$ with a fixed $k \in \mathbb{N}$. Further
connectives are defined as follows:

\[
\neg \phi = \top \\
\phi \land \psi = \phi \land (\phi \rightarrow \psi) \\
\phi \lor \psi = (\phi \rightarrow \psi) \land (\phi \rightarrow \psi) \\
\phi \leftrightarrow \psi = (\phi \rightarrow \psi) \land (\psi \rightarrow \phi)
\]

with $\phi$ and $\psi$ arbitrary formulas. A propositional evaluation
is a mapping $e : V \rightarrow S_k$ that is extended to formulas
as follows. If $\phi$ and $\psi$ are formulas and $c$ is an element in $S_k$, then

\[
e(\phi \rightarrow \psi) = e(\phi) \lor e(\psi) \land (e(\top) = c),
\]

where $x \rightarrow y = \text{min}(1, 1 - x + y)$ for $x, y \in S_k$. Note that
$(x \rightarrow y) = 1$ iff $x \leq y$. The set of all such evaluations
will be henceforth denoted by $\Omega_k$. Notice that, in particular,
for every formula $\phi$ and $\psi$ and for every $e \in \Omega_k$, we have
\n\[
e(\neg \phi) = 1 - e(\phi), e(\phi \land \psi) = \text{max}(e(\phi) + (1 - e(\psi)))
\]

A formula $\phi$ is said to be $\textit{satisfiable}$ if there is some proposi-
tional evaluation $e$ such that $e(\phi) = 1$. In such a case we
say that $e$ is a model of $\phi$. A tautology is a formula $\phi$ such
that $e(\phi) = 1$ for each propositional evaluation $e$. A formula
$\phi$ is a $\textit{semantic consequence}$ of a set of formulas $\Gamma$, written
as $\Gamma \vdash \phi$ iff it holds that if $e$ is a model of each formula
in $\Gamma$, then $e$ is also a model of $\phi$. The logic $L_k$ based on the
language $L$ has a sound and a strongly complete axiomatiza-
tion, see e.g. [Cignoli et al., 2000] for details. So if $\vdash$ denotes
the notion of proof defined from the set of axioms of $L_k$ and
modus ponens, then for any, hence possibly infinite, set of
formulas $T \cup \{\psi\}$, it holds that $T \vdash \psi$ iff $T \models \psi$. A formula
$\psi$ that can be proven from axioms and rules only is called a $\textit{theo-
rem}$ and this is written as $\vdash \psi$.

To reason with beliefs over fuzzy propositions we in-
roduce a modal operator $B$. By $L_B$ we denote the expansion of
$L$ by $B$. We base our approach on previous theoretical work
[Bou et al., 2011b] on fuzzy modal logics where the truth-
values form a finite residuated lattice. In [Bou et al., 2011b],
the authors introduce the minimal modal logic over $L_k$. Its
axioms are all the axioms of $L_k$, plus the following.

\[
(B2) \quad (B\phi \land B\psi) \rightarrow B(\phi \land \psi),
\]

\[
(B3) \quad B(\sigma \rightarrow \phi) \leftrightarrow (\sigma \land B\phi), \quad \text{for each } c \in S_k,
\]

\[
(B4) \quad (B\phi \lor B\psi) \leftrightarrow B(\phi \lor \psi).
\]

The rules are modus ponens (from $\phi$ and $\phi \rightarrow \psi$ infer $\psi$) and
monotonicity for $B$ (if $\phi \rightarrow \psi$ is a theorem then $B\phi \rightarrow B\psi$
theorem as well). In [Bou et al., 2011b], the authors show
that this is a sound and complete axiomatization with respect
to the class of Kripke frames $M = (W, e, R)$ where $W$ is a set
of possible worlds, $e : W \times V \rightarrow S_k$ is a mapping giving an
evaluation $e(w, \cdot) : V \rightarrow S_k$ for each possible world $w$ and
$R : W \times W \rightarrow S_k$ is a $S_k$-valued binary relation on possible
worlds. Given a (Kripke) frame $M = (W, e, R)$ and a world
\n$w \in W$, the truth value of a formula $\phi$ in $L_B$ is inductively
defined as follows:

\[
\begin{align*}
&\quad \text{If } \phi \text{ is a propositional variable } p, \text{ then } ||\phi||_{M, w} = e(w, p), \\
&\quad \text{If } \phi \text{ is truth-constant } \top, \text{ then } ||\phi||_{M, w} = c. \\
&\quad \text{If } \phi = B\psi, \text{ then } ||\phi||_{M, w} = \inf \{R(w, w') \Rightarrow ||\psi||_{M, w'} : w' \in W\}. \\
&\quad \text{If } \phi = \psi \rightarrow \gamma, \text{ then } ||\phi||_{M, w} = ||\psi||_{M, w} \Rightarrow ||\gamma||_{M, w}.
\end{align*}
\]

The third bullet then intuitively expresses that $\psi$ is believed
in a world $w \in W$ to the degree that $\psi$ is true in all worlds
\n$w' \text{ that are accessible (related to) from } w \text{ to a certain degree}.
A formula $\phi$ is said to be $\textit{satisfiable}$ if there exists a frame
\n$M = (W, e, R)$ and a $w \in W$ such that $||\phi||_{M, w} = 1$ and
we say that $\phi$ is satisfied by $M$. It is called a $\textit{tautology}$ if for
each frame $M = (W, e, R)$ we have $||\phi||_{M, w} = 1$ for each
$w \in W$. A formula $\phi$ is a $\textit{semantic consequence}$ of a set of
formulas $\Gamma$, written as $\Gamma \models \phi$ iff it holds that if each formula
in $\Gamma$ is satisfied by a frame $M$, that $\phi$ is also satisfied by $M$.

As it was shown in [Bou et al., 2011b], the well-known axiom
\n(K) $B(\phi \rightarrow \psi) \rightarrow (B\phi \rightarrow B\psi)$
\]

is not generally sound in the above Kripke frames, only in frames
\n$M = (W, e, R)$ where $R$ is a two-valued relation on $M$ (i.e. when
\n$R(w, w') \in \{0, 1\}$ for all $w, w' \in W$). Notice that in such Kripke frames, the truth evaluation of $B\phi$ in a
world $w \in W$ reduces to

\[
||B\phi||_{M, w} = \inf \{||\phi||_{M, w'} : R(w, w') = 1\}.
\]

In the remainder of the paper we will be interested in the class
of Kripke frames with two-valued accessibility relations. We
will denote this class by \( \mathcal{M} \). Moreover we will denote by \( \mathcal{B}_L \) the axiomatic extension of the minimal modal logic with axiom (K). Due to the presence of axiom (K), the monotonicity rule can be replaced by the usual necessitation rule: if \( \phi \) is a theorem then \( \Box \phi \) is a theorem as well.

### 3 Fuzzy modal logics of belief

When defining logics for belief, it is usual to presume that the agent has both positive and negative introspective capabilities. This is captured in the classical case by the well-known axioms (4) and (5). Moreover, sometimes belief consistency is required which is captured by axiom (D). Finally, we consider the following extensions of \( \mathcal{B}_L \): such as (4) and (5).

As was done for (fuzzy) autoepistemic logic [Blondeel et al., 2013; Moore, 1983], each formula \( \phi \) can be converted to a proof for \( \psi \) in one of the extensions of \( \mathcal{B}_L \).

#### Lemma 3.1

Let \( L \) be any of the logics \( K 45(\mathcal{L}_k) \), \( KD 45(\mathcal{L}_k) \), \( S 5(\mathcal{L}_k) \). Suppose \( T \cup \{ \psi \} \) is a set of formulas from \( \mathcal{L}_B \) and let \( \Lambda_L = \{ \phi^+ | \vdash_L \phi \} \). Then it holds that \( T \vdash \psi \iff T^* \cup \Lambda_L \vdash \psi^+ \).

**Proof.** Suppose a proof for \( \psi \) in \( L \) from \( T \) has the form \( \Gamma = (\gamma_1, \ldots, \gamma_m) \). A proof for \( \psi^+ \) in \( L \) from \( T^* \cup \Lambda_L \) is then easily obtained by replacing all formulas \( \gamma_i \) in \( \Gamma \) by \( \gamma_i^+ \).

Conversely, suppose there is a proof \( \Phi = (\phi_1, \ldots, \phi_n) \) for \( \psi^+ \) in \( L \) from \( T^* \cup \Lambda_L \). For each \( i \) we then have that \( \phi_i = \gamma_i^+ \) with either \( \gamma_i \in T \) or \( \gamma_i \) an instantiation of an axiom

In \( L \) or \( \gamma_i = \Box \alpha \) with \( \vdash \alpha \). The sequence \( \Gamma = (\gamma_1, \ldots, \gamma_m) \) obtained from \( \Phi \) is then converted to a proof for \( \psi \) in \( L \) from \( T \) as follows. If for some \( i \), \( \gamma_i \not\in T \) and \( \gamma_i \) is not an axiom in \( L \), then add a proof for \( \gamma_i = \Box \alpha \), which is possible since in this case, it must hold that \( \vdash \alpha \) from which we can then infer that \( \gamma_i = \Box \alpha \) is a theorem as well.

The next step is to define the canonical Kripke frame for a given modal logic \( L \). The following definition is general for any logic \( L \) resulting from all possible combinations of the axioms (4), (5) and (T) and hence in particular for \( L = \{ K 45(\mathcal{L}_k), K D 45(\mathcal{L}_k), S 5(\mathcal{L}_k) \} \). The \( L \)-canonical Kripke frame is defined as the Kripke frame \( M_{can}^L = (W_{can}^L, \Gamma_{can}, R_{can}^L) \), where

- \( W_{can}^L = \{ w \in \Omega_k | \forall \phi \in A_L \vdash w(\phi^+) = 1 \} \) with \( A_L = \{ \phi^+ | \vdash_L \phi \} \)
- \( \Gamma_{can} = \{ (w_1, w_2) \in \Omega_k \times \Omega_k | \forall \phi : w_1((B\phi)^+) = 1, then \( w_2(\phi^+) = 1 \} \)
- \( e_{can}^L(w, p) = w(p) \) for each variable \( p \)

We now introduce some subclasses of \( \mathcal{M} \), depending on which properties the two-valued accessibility relations in the Kripke frames (\( W, e, R \)) satisfy.

- \( \mathcal{M}_{et} \): class of Kripke frames with Euclidean² and transitive relations
- \( \mathcal{M}_{can} \): class of Kripke frames with Euclidean, serial and transitive relations
- \( \mathcal{M}_{ref} \): class of Kripke frames with reflexive, symmetric and transitive relations

In Theorem 3.4 we will show that the extensions of \( \mathcal{B}_L \) defined above are sound and complete axiomatizations for these subclasses of \( \mathcal{M} \). To show completeness, we need to prove the following truth lemma (general for any \( L \)).

**Proposition 3.2.** (Truth-lemma) For any \( \mathcal{L}_B \)-formula \( \phi \), let \( M_{can}^L = (W_{can}^L, e_{can}^L, R_{can}^L) \) be its canonical Kripke frame with \( L \in \{ K 45(\mathcal{L}_k), K D 45(\mathcal{L}_k), S 5(\mathcal{L}_k) \} \). Then it holds that \( v(\phi^+) = \| \phi \|_{M_{can}^L, v} \) for every \( v \in W_{can}^L \).

**Proof.** By using the monotonicity for \( B \) and the meet distribution property, the claim follows by an easy adaption from Lemma 4.20 in [Bou et al., 2011b].

We can now show the following properties for the canonical Kripke frames.

**Proposition 3.3.** Let \( L \in \{ K 45(\mathcal{L}_k), K D 45(\mathcal{L}_k), S 5(\mathcal{L}_k) \} \), then the following conditions hold

1. If \( L \) contains axiom (T) then \( R_{can}^L \) is reflexive.
2. If \( L \) contains axiom (4) then \( R_{can}^L \) is transitive.
3. If \( L \) contains axiom (5) then \( R_{can}^L \) is Euclidean.
4. If \( L \) contains axiom (D) then \( R_{can}^L \) is serial.

²Recall that a relation \( R \) is called Euclidean if \( R(w_1, w_2) = R(w_1, w_3) = 1 \) implies \( R(w_2, w_3) = 1 \) for each \( w_1, w_2, w_3 \in W_{can} \) and serial if for each \( w_1 \in W \) there exists \( w_2 \in W \) such that \( R(w_1, w_2) = 1 \).
Proof. 1. Let \( w \in W_{can}^L \) and \((w((B\phi))^* = 1. Since \((B\phi \to \phi)^* \in \Lambda_L\), it follows that \( 1 = w((B\phi \to \phi))^* = \|B\phi \to \phi\|_{M^w_{can},w} \leq \|\phi\|_{M^w_{can},w} = w(\phi^*)\). Therefore \(R^L_{can}(w, 1) = 1\).

2. Let \( w_1, w_2, w_3 \in W_{can}^L \) such that \( R^L_{can}(w_1, w_2) = 1 \) and \( w_1((B\phi)^* = 1. Since \( B\phi \to BB\phi)^* \in \Lambda_L\), it follows that \( 1 = w_1((B\phi \to BB\phi)^* = \|B\phi \to BB\phi\|_{M^w_{can},w} \leq \|B\phi\|_{M^w_{can},w} = w((B\phi)^*)\). Since \( R^L_{can}(w_1, w_2) = 1\), it then follows that \( w_2((B\phi)^*) = 1 \) and subsequently, since \( R^L_{can}(w_2, w_3) = 1\), that \( w_3((B\phi)^*) = 1\). Hence \( R^L_{can}(w_1, w_3) = 1\).

3. Let \( w_1, w_2, w_3 \in W_{can}^L \) such that \( R^L_{can}(w_1, w_2) = 1 \) and \( w((B\phi)^*) = 1. By definition, \( \|B\phi\|_{M^w_{can},w} = \inf\{\|\neg B\phi\|_{M^w_{can},v} \mid R^L_{can}(w_1, v) = 1\} \), hence \( \|B\phi\|_{M^w_{can},w} \leq \| \neg B\phi\|_{M^w_{can},w} \). Now since \( \|\neg B\phi\|_{M^w_{can},w} = 1 \), we obtain \(\|B\phi\|_{M^w_{can},w} = 1 - w_2((B\phi)^*) = 0\). Since \( \neg B\phi \to B\phi \) \( \in \Lambda_L\), it follows that \( 1 = w_1((\neg B\phi \to B\phi)^*) = \neg B\phi \to B\phi \) \( \in \Lambda_L\), and hence \( 1 = \|\neg B\phi\|_{M^w_{can},w} \leq \|B\phi\|_{M^w_{can},w} = w((B\phi)^*)\). Finally, since \( R^L_{can}(w_1, w_3) = 1\), it then follows that \( w_3((B\phi)^*) = 1\), and hence \( R^L_{can}(w_2, w_3) = 1\).

4. Let \( w_1 \in W_{can}^L \). Since \( \neg B\phi \) \( \in \Lambda_L\), it follows that \( 1 = w_1((\neg B\phi)^*) = \neg B\phi \to B\phi \) \( \in \Lambda_L\), and thus \( 0 = \|\neg B\phi\|_{M^w_{can},w} = \inf\{\|\neg B\phi\|_{M^w_{can},v} \mid R^L_{can}(w, v) = 1\} \). Therefore the latter set must be non-empty, and hence there must exist \( w_2 \in W_{can}^L \) such that \( R^L_{can}(w_1, w_2) = 1\).

Using Proposition 3.3, we can now show the following theorem.

Theorem 3.4. \( K45(L_k), KD45(L_k) \) and \( S5(L_k) \) are sound and complete w.r.t. the classes \( M^s\), \( M^e \) and \( M^s\) respectively.

Proof. Soundness is straightforward. We can show the completeness by proving that if there is a formula \( \phi \) such that \( R^L_{can}(w, v) = 1 \) for all \( w \in W_{can}\) where \( \|\phi\|_{M^w_{can},w} < 1\), then there must exist a Kripke frame \( M = (W, e, R) \) in the corresponding subclass of Kripke frames such that \( \|\phi\|_{M,w} \leq 1\). We show that the L-canonical Kripke frame meets this condition. The fact that each of these canonical Kripke frames belong to the correct subclass of \( M \) follows from Proposition 3.3 and by the fact that a relation that is reflexive and Euclidean is also symmetrical. By Lemma 3.1 it follows, independently on \( L \), that \( \Lambda_L \not\subseteq \phi^* \) and by the strong completeness of \( L_k \) it then follows that \( \Lambda_L \not\subseteq \phi, i.e. there exists \( v \in W_{can}\) such that \( \|\phi\|_{M^v_{can},v} = v(\phi^*) < 1\). □

As in the classical case, the logics \( K45(L_k), KD45(L_k) \) and \( S5(L_k) \) admit simpler semantics while preserving soundness and completeness. Consider the following cases of Kripke frames:

- \( M^s\) : the subclass of Kripke frames \( M = (W, e, R) \) of \( M^s\) when \( R = W \times E \) for some \( E \subseteq W \)
- \( M^e\) : the subclass of Kripke frames \( M = (W, e, R) \) of \( M^e\) when \( R = W \times E \) for some \( \emptyset \not\subseteq E \subseteq W \)
- \( M^s\) : the subclass of Kripke frames \( M = (W, e, R) \) of \( M^s\) when \( R = W \times W \)

Proposition 3.5. \( K45(L_k), KD45(L_k) \) and \( S5(L_k) \) are sound and complete w.r.t. the classes \( M^s\), \( M^e \) and \( M^s\) respectively.

Proof. We only prove the case for \( KD45(L_k)\), the other cases being easy variations. By Theorem 3.4, it is sufficient to show that \( M^s\) and \( M^e\) have the same tautologies. Since \( M^s\) is a subclass of \( M^e\), we only have to show that if \( \phi \) is a formula \( \phi \) there exists \( M = (W, e, R) \in M^s\) and \( w \in W\) such that \( \|\phi\|_{M,w} < 1\), that there exists \( M' = (W', e', R') \in M^s\) and \( w' \in W'\) such that \( \|\phi\|_{M',w'} < 1\). Suppose such a frame \( M = (W, e, R) \in M^s\) and \( w \in W\). Define \( E = \{ v \in W \mid R(w,v) = 1\} \). By seriality of \( R \) we have \( E \not\subseteq \emptyset\). We define \( M'\) as follows: \( W' = \{ w \cup E, e' = e_{W' \times W}, R' = W' \times E\). Notice that for \( v \in E \) arbitrary we have \( E \subseteq \{ z \mid R(v, z) = 1\} \subseteq E\) since \( R \) is Euclidean and \( \{ z \mid R(v, z) = 1\} \subseteq E\) since \( R \) is transitive. Hence \( E \subseteq \{ z \mid R(v, z) = 1\} \) for all \( v \in E\). We can now show by structural induction that for each \( \psi \) it holds that \( \|\psi\|_{M',v} = \|\psi\|_{M',w} \) for every \( v \in E\). The only notable case is when \( \psi = B\alpha \), but this follows by the previous remark: \( \|B\psi\|_{M,w} = \inf\{\|\alpha\|_{M,v'} \mid v' \in E\} = \|B\alpha\|_{M',w} \). Finally, it is sufficient to show that \( \|\psi\|_{M,w} = \|\phi\|_{M',w} \). We do this by structural induction, again we only show the case \( \phi = B\psi \): \( \|B\psi\|_{M,w} = \inf\{\|\psi\|_{M,v} \mid v \in E\} = \inf\{\|\psi\|_{M',v} \mid v \in E\} = \|B\psi\|_{M',w}. \) □

4 Complexity of satisfiability problems

In this section we will discuss the complexity of two satisfiability problems for \( KD45(L_k)\).

- 1-SAT: Given a formula \( \phi \), does there exist \( M = (W, e, R) \in M^s\) and \( w \in W\) such that \( \|\phi\|_{M,w} = 1\)?
- pos-SAT: Given a formula \( \phi \), does there exist \( M = (W, e, R) \in M^s\) and \( w \in W\) such that \( \|\phi\|_{M,w} > 0\)?

We will show that these problems are NP-complete and hence generalizing to the many-valued case does not imply an increase in computational complexity. See [Halpern and Moses, 1992] for results on the complexity of classical modal logics. As in the previous section, the same results can be obtained for \( K45(L_k)\) and \( S5(L_k)\).

For any formula \( \phi \) of \( L_k\), we denote by \#\phi its complexity:
- \#e = 1 for each \( e \in S_k\) and \#p = 1 for every propositional variable \( p\)
- \#(\phi \to \psi) = 1 + \#\phi + \#\psi and \#(B\phi) = 1 + \#\phi.

For a formula \( \phi \) of \( KD45(L_k)\) and a subformula \( \psi \) of \( \phi \), the depth \( d(\psi) \) of \( \psi \) in \( \phi \) is defined as usual given the tree of subformulas of \( \phi \). For instance, for a formula \( B(a \land Bb) \), we have \( d(B(a \land Bb)) = 0, d(a \land Bb) = 1, d(a) = d(Bb) = 2 \) and \( d(b) = 3\). We can then show the following finite model property:
Lemma 4.1. Let φ be a $\mathcal{L}_B$-formula. Then for every frame $M = (W, e, R) \in \mathbb{M}^*_\text{est}$, and for every $w \in W$, there exists a finite frame $M' = (W', e', R') \in \mathbb{M}^*_\text{est}$ and a world $w' \in W'$ such that $|W'| \leq \# \phi$ and $\|\phi\|_{M, w} = \|\phi\|_{M', w'}$.

Proof. Consider a frame $M = (W, e, R)$ with $R = W \times E$ and $w \in W$. The aim is to find a finite set $W'$ and a non-empty subset $E' \subseteq W'$ for which the claim holds.

Trivially, if φ is B-free, then take $W' = E' = \{w\}$, $R' = W' \times E'$ and $e'$ be defined by restriction.

Otherwise, if φ is not B-free, let $d$ be the maximum depth of the subformulas of the form of φ.

If $d = 0$, then φ = $B\psi$ and $\psi$ is B-free. Then $\|B\psi\|_{M, w} = \inf \{\|\psi\|_{M, w^*} \mid w^* \in E\}$. Now, since $\|\psi\|_{M, w^*} = e(\Psi, \psi)$ can only take a finite number of values in $S_k$, there exists a world $w^0 \in W$ in which the infimum is attained, i.e. $\|B\psi\|_{M, w} = \inf \{\|\psi\|_{M, w^*} \mid w^* \in E\} = \|\psi\|_{M, w^0} = e(\Psi, \psi)$.

In this case put $W' = \{w^0\}$, $w' = w^0$ and let $E'$ and $e'$ be defined by restriction.

If $d > 0$, let $B\psi_1, \ldots, B\psi_n$ be the subformulas of φ of depth $d$, hence each $\psi_i$ is B-free. Again, for each $\psi_i$, there exists a world $w^0_i$ such that $\|B\psi_i\|_{M, w^0} = e(\psi_i, \psi_i)$ with $w^0$ arbitrary. Now replace each subformula $B\psi_i$ by the corresponding constant and repeat the process for all levels $n = d - 1, \ldots, 0$. Put $W' = \{w\} \cup \{w^0 \mid 1 \leq l \leq d, 1 \leq j \leq r_j\}, w' = w$ and let $E'$ and $e'$ be defined by restriction to $W'$. Then, by construction, $\|\phi\|_{M, w} = \|\phi\|_{M', w'}$. Moreover, $|W'| = 1 + \sum_{l=0}^d r_l \leq \# \phi$. □

Observe that, as in the proof of Lemma 4.1, given a formula φ of $KD45(\mathcal{L}_k)$ and a frame $M = (W, e, R)$, we can construct an $\mathcal{L}_k$ formula $\phi^M$.

Theorem 4.2. The problems 1-SAT and pos-SAT for $KD45(\mathcal{L}_k)$ are NP-complete.

Proof. In order to prove NP-membership, recall that from Lemma 4.1 a formula φ is 1-SAT in a frame M iff φ is 1-SAT in a finite frame $M'$ where its cardinality is in the complexity of φ. Let us guess the frame $M' = (W', e', R')$. Since $|W'| \leq \# \phi$, the guess is polynomial in #φ. Let $\phi^M$ the formula of $\mathcal{L}_k$ obtained from $M'$ and φ applying the procedure described above, and notice that # $\phi^M$ is polynomial in #φ. Moreover, since $|W'| \leq \# \phi$ the formula # $\phi^M$ is obtained in a number of steps which is polynomial in #φ. From [Mundici, 1987] it follows that checking if $\phi^M$ is either 1-SAT or pos-SAT in $\mathcal{L}_k$ is NP, and hence our claim follows. Since each formula of $\mathcal{L}_k$ is in particular a formula of $KD45(\mathcal{L}_k)$, and since both 1-SAT and pos-SAT for $\mathcal{L}_k$ are NP-complete, the NP-hardness of our problems follows. □

Notice that in [Bou et al., 2011a], the authors considered these satisfiability problems for the minimal modal logic over $\mathcal{L}_k$, but with respect to generic Kripke frames where the accessibility relation can also be many-valued, and they proved that those problems are PSPACE-complete.

5 Relating fuzzy modal logic and fuzzy autoepistemic logic

The aim of Moore’s autoepistemic logic [Moore, 1983] is to model or characterize the set of beliefs of a rational agent with introspection capabilities by means of a set of simple properties it must fulfill. The language of fuzzy autoepistemic logic is the same as $\mathcal{L}_B$, the one of the modal logic with the belief operator $B$, so $B\phi$ is also read as “the agent believes φ”.

The basic construct is the notion stable expansion $E$ of a set of initial beliefs $A$, briefly introduced in Section 1. It can be seen as the closed set of beliefs of an ideal rational agent reflecting on his own beliefs. This notion has been generalized to the fuzzy case in [Blondeel et al., 2013] as follows. A stable fuzzy expansion of a set of $\mathcal{L}_B$-formulas $A$ is a fuzzy set $E_A : \mathcal{L}_B \rightarrow S_k$ that satisfies the following fix-point condition:

$E_A(\phi) = \inf \{v(\phi^*) \mid v \in \Omega_k, \phi \rightarrow \gamma, v(\phi^*) \leftrightarrow E_A(\phi) \mid v \in \Omega_k\}$.

Recall that $E_A(\phi)$ is the symbol corresponding to $E_A(\phi) \in S_k$.

Generalizing Moore’s result [Moore, 1984], in [Blondeel et al., 2013] it is shown that stable fuzzy expansions can also be characterized by a fuzzy Kripke-style possible world semantics. In particular, a fuzzy autoepistemic structure is a pair $(w, S)$ with $w \in \Omega_k$ representing the actual world used to evaluate B-$\phi$-formulas and $S \subseteq \Omega_k$ representing all worlds considered possible (epistemic states) used to evaluate formulas of the form $B\phi$. The class of such structures will be denoted by $M^ae$. The degree of truth of a $\mathcal{L}_B$-formula $\phi$ relative to a fuzzy autoepistemic structure $(w, S)$ is inductively defined as follows:

- If $\phi$ is a propositional formula from $\mathcal{L}$, then $\|\phi\|_{(w, S)} = w(\phi)$.
- If $\phi$ is a truth constant $\tau$, then $\|\phi\|_{(w, S)} = c$.
- If $\phi = B\psi$, then $\|B\psi\|_{(w, S)} = \inf \{\|\psi\|_{(v, S)} \mid v \in S\}$.
- If $\phi = \psi \rightarrow \gamma$, $\|\phi\|_{(w, S)} = (\|\psi\|_{(w, S)} \Rightarrow \|\gamma\|_{(w, S)})$.

Intuitively, one can think of $S$ as a set of “sources” (worlds) and we define the truth value of $B\phi$ in $S$ as the minimal value of $\phi$ such that each source supports it at least in this degree. Since the truth evaluation of formulas of the form $B\phi$ in a structure $(w, S)$ does not depend on the actual world $w$, we will also write $\|B\phi\|_S$ to denote $\|B\phi\|_{(w, S)}$. Note that if $S = \emptyset$, then $\|B\phi\|_S = 1$. Also note that, conversely, the world $w$ in $(w, S)$ is needed to evaluate non-modal formulas.

We consider the following subclasses of fuzzy autoepistemic structures of $M^ae$:

- the class $M^ae_{\emptyset}$, where only pairs $(w, S)$ with $S$ non-empty are considered, and
- the class $M^ae_{\neq}$, where only pairs $(w, S)$ with $w \in S$ are considered.

It can be shown that $K45(\mathcal{L}_k)$, $KD45(\mathcal{L}_k)$ and $S5(\mathcal{L}_k)$ are sound and complete with respect to the classes $M^ae_{\emptyset}$, $M^ae_{\neq}$ and $M^ae_{\neq}$ respectively.

Theorem 5.1. $K45(\mathcal{L}_k)$, $KD45(\mathcal{L}_k)$ and $S5(\mathcal{L}_k)$ are sound and complete w.r.t. to $M^ae_{\emptyset}$, $M^ae_{\neq}$ and $M^ae_{\neq}$ respectively.
Proof. We only show the case of \( KD45(L_k) \). The other cases are obtained by slight adaptations of the proof. By Proposition 3.5 it is sufficient to show that \( M^*_w \) and \( M^{ae} \) have the same tautologies. First suppose there exists \( M = (W, e, R) \in M^*_w \) and \( w \in W \) such that \( \| \phi \|_{M, w} < 1 \). Define \((w, S) \in M^{ae} \) such that \( R = W \times S \). We can then show by structural induction that for each formula \( \gamma \) we have \( \| \gamma \|_{M, w} = \| \gamma \|_{(w, S)} \) for all \( v \in W \). Next, suppose we have \((w, S) \in M^{ae} \) such that \( \| \phi \|_{w, S} < 1 \). Define \( M = (W, e, R) \in M^{ae} \) as follows: \( W = \{ w \} \cup S \) and \( R = W \times S \). Similar as above it follows that \( \| \gamma \|_{M, w} = \| \gamma \|_{(w, S)} \) for each formula \( \gamma \). □

In [Blondeel et al., 2013] the authors characterize stable fuzzy expansions in terms of this possible world many-valued semantics. Indeed, the fuzzy belief set \( B\mathcal{E}_S \) induced by an epistemic state described by a set of \( L_k \)-evaluations \( S \) is defined in the natural way: for each \( \mathcal{L}_B \)-formula \( \varphi \)

\[
B\mathcal{E}_S(\varphi) = \| B\varphi \|_S = \inf_{w \in S} \| \varphi \|_{(w, S)}.
\]

This notion also generalizes that of a S5-set to the many-valued case. Moreover, given a set of formulas \( A \), a set of propositional evaluations \( S_A \) is called a fuzzy autoepistemic model of \( A \) whenever

\[
S_A = \{ w \in \Omega_k \mid \| \phi \|_{(w, S_A)} = 1 \text{ for each } \phi \in A \}.
\]

Intuitively, \( S_A \) is the set of worlds or “sources” in which all formulas of \( A \) are true. Then the following result is proved in [Blondeel et al., 2013].

**Proposition 5.2.** A fuzzy set of formulas \( E : \mathcal{L}_B \to S_k \) is a stable fuzzy expansion of a set of formulas \( A \) if it is the belief set for some fuzzy autoepistemic model \( S_A \) of \( A \), i.e. \( E(\phi) = \| B\phi \|_{S_A} \) for each \( \phi \in \mathcal{L}_B \).

On the other hand, as it happens in the classical case, we can also characterize fuzzy belief sets, or equivalent stable fuzzy expansions, by means of the syntactical notion of fuzzy stable sets (c.f. [Halpern and Moses, 1992]).

**Definition 5.3.** Let \( \Gamma : \mathcal{L}_B \to S_k \) be a fuzzy set and put

\[
\hat{\Gamma} = \{ \Gamma(\varphi) \to \varphi^* \mid \varphi \text{ formula } \}. \text{ We say that } \Gamma \text{ is a fuzzy stable set if the following conditions hold:}
\]

1. \( \hat{\Gamma} \) is propositionally consistent, i.e. \( \hat{\Gamma} \not\vdash \overline{\varphi} \).
2. If \( \hat{\Gamma} \vdash \overline{\varphi} \) then \( \Gamma(\varphi) \geq c \).
3. \( \Gamma(\varphi) = \Gamma(B\varphi) \)
4. \( 1 - \Gamma(\varphi) = \Gamma(\overline{\varphi}) \)

**Proposition 5.4.** \( \Gamma \) is a fuzzy stable set iff \( \Gamma \) is a fuzzy belief set.

Proof. (1) First we show that a fuzzy belief set \( \Gamma \) is a fuzzy stable set. By definition of a fuzzy belief set we know that there exists \( S \subseteq \Omega_k \) such that \( \Gamma(\varphi) = \| B\varphi \|_S \) for each formula \( \varphi \). In order to show that \( \hat{\Gamma} \) is propositionally consistent, by the strong completeness of \( L_k \), it is sufficient to show that there exists \( v \in \Omega_k \) such that for each formula \( \varphi \) we have \( \Gamma(\varphi) \leq v(\varphi^*) \). Let \( w \in S \) be arbitrary but fixed and define \( v \) such that \( v(\varphi^*) = \| \varphi \|_{(w, S)} \) for each \( \varphi \).

It follows that \( \Gamma(\varphi) \leq v(\varphi^*) \) which proves (1). Next, assume that \( \hat{\Gamma} \vdash \overline{\varphi} \to \varphi^* \), or by strong completeness of \( L_k \) that \( \Gamma \vdash \overline{\varphi} \to \varphi^* \). We show that \( \Gamma(\varphi) \geq c \). Note, similar as above, that for each \( w \in S \) we have that \( v(\varphi^*) = \| \varphi \|_{(w, S)} \) is a model of \( \hat{\Gamma} \) and hence of \( \overline{\varphi} \to \varphi^* \). Therefore \( c \leq \| \varphi \|_{(w, S)} \) for each \( w \in S \) and \( c \leq \inf_{w \in S} \| \varphi \|_{(w, S)} = \Gamma(\varphi) \).

Proving (3) follows easily by noting that \( \Gamma(\varphi) = \| B\varphi \|_S = \inf_{w \in S} \| B\varphi \|_{(w, S)} \) is a fuzzy stable set. Define \( S = \{ w \in \Omega_k \mid u(\alpha) = 1, \forall \alpha \in \hat{\Gamma} \} \). Note that \( S \) is nonempty by (1). Next, we show that for each \( w \in S \) we have by (3) that \( w(B(\varphi)^*) \geq \Gamma(\varphi) \).

For each \( w \in \Omega_k \) set \( (\overline{\varphi})^* \) such that \( ((\overline{\varphi})^*) = \inf_{w \in S} \| \overline{\varphi} \|_{(w, S)} \) for each formula \( \varphi \) and an arbitrary \( w \in S \). The only notable case is where \( \alpha = B\psi \).

By the induction hypothesis and by the definition of \( S \) we have \( \| B\psi \|_S = \inf_{w \in S} \| \psi \|_{(w, S)} \) for each formula \( \varphi \) and hence

\[
\hat{\Gamma} \vdash \overline{\varphi} \to \varphi^* \to \psi^*.
\]

By (2), it then follows that \( \Gamma(\psi) \geq u'(\psi^*) \) and hence that

\[
\Gamma(\psi) > \Gamma(\varphi), \text{ a contradiction.} \]

□

The results from this section show that the modal logics we are dealing with are suitable for reasoning in the fuzzy autoepistemic frame. In particular, the following example illustrates that real-world situations, which can be naturally expressed in the autoepistemic language, can also be framed in our fuzzy modal setting.

**Example 5.5.** Consider the following example to show how fuzzy autoepistemic logic can be used in a real world scenario. Suppose we have a wireless sensor network consisting of devices that can sense their environment and communicate wirelessly, for instance with the purpose of detecting forest fires in an early stage. We will use fuzzy autoepistemic logic to determine whether there are sensors not working optimally and if so, within what range we can assume the temperature to be.

Let \( t_i \) be the actual temperature at sensor \( i \) and \( t'_i \) the temperature measured by sensor \( i \). Suppose we have an appropriate rescaling to assure that all variables take values in \( \Omega_k \) and let \( e_i \) be the variable representing the degree to which sensor \( i \) is faulty. The formula

\[
\neg B e_i \to (t_i \leftrightarrow t'_i)
\]

then captures a fuzzy version of the classical intuition that “If we do not believe that the sensor \( i \) is broken, then it measures the right temperature”. Moreover, let us define a new connective \( d(\varphi, \psi)^3 \) as \( \neg(\varphi \leftrightarrow \psi) \) for which the semantics

\[3] The connective \( d(\varphi, \psi) \) is well known in the literature of many-
is given by the Euclidean distance $d$: for all $x, y \in [0,1]$, $d(x,y) = |x - y|$. The formula 

$$(w_{ij} \rightarrow d(t_i', t_j')) \rightarrow e_i \lor e_j$$

(2) 

then has the following intuitive meaning: “If the difference between the measurements of sensors $i$ and $j$ is above some threshold $w_{ij}$ (to a certain degree), then there must be something wrong with one of the sensors. This value $w_{ij}$ can for instance be based on the location of the sensors $i$ and $j$.

As a concrete example, suppose we have 2 sensors such that the temperature measured by sensor 1 is equal to 0.4 and by sensor 2 equal to 0.5. Suppose the threshold is equal to 0.2. For a fuzzy autoepistemic model $S$ of formulas (1) and (2), it must then hold for each $w' \in S$ that

(a) $0.4 - \inf_{w \in S} w(e_1) \leq w'(t_1) \leq 0.4 + \inf_{w \in S} w(e_1)$

(b) $0.5 - \inf_{w \in S} w(e_2) \leq w'(t_2) \leq 0.5 + \inf_{w \in S} w(e_2)$

(c) $0.9 \leq \max(w'(e_1), w'(e_2))$

where (a) and (b) follow from the fact that $(w', S)$ must satisfy equation (1), and (c) from the fact that it must satisfy equation (2). One can easily show that $S = \{ w \in \Omega_k \mid 0.9 \leq \max(w(e_1), w(e_2)), w(t_1) = 0.4, w(t_2) = 0.5 \}$ is a fuzzy autoepistemic model of formulas (1) and (2). In the corresponding stable fuzzy expansion $E$ it holds $E(t_1) = 0.4, E(t_2) = 0.5, E(e_1) = E(e_2) = 0$ and $E(e_1 \lor e_2) = 0.9$. Hence we can conclude that there is no error made by the sensors.

6 “Only knowing” operators and fuzzy stable expansions

Let us expand $L_{B}$ with a modal operator $O$ such that a formula $O\psi$ has to be interpreted as “ψ is all that is believed”. In the classical case [Levesque, 1990], the semantics is defined as follows. Given a epistemic state $S$ consisting of a set of classical evaluations, a formulas $O\psi$ is true in $S$ when $ψ$ is true in any structure $(z, S)$ with $z \in S$, and false in any structure $(z', S)$ with $z' \notin S$.

We can straightforwardly generalize this condition to the many-valued case by defining

$$[O\psi]_{(w, S)} = \min_{z \in S} \psi_{(z, S)},$$

where now $w \in \Omega_k$ and $S \subseteq \Omega_k$. Formulas of the form $B\psi$ are evaluated as in fuzzy autoepistemic logic. If we then add another modal operator $N$ whose truth evaluation in a pair $(w, S)$ is

$$[N\psi]_{(w, S)} = \inf_{z \notin S} \psi_{(z, S)},$$

then it is easy to see that the semantics of $O\psi$ is exactly that of $B\psi \land N\neg\psi$. Notice that $[N\psi]_{(w, S)} = [B\psi]_{(w, \Omega_k \setminus S)}$, so it is clear that $N$ is another K45 operator. Again, since the truth value of $O\psi$ (and $N\psi$) in a structure $(w, S)$ does not depend on $w$, we will also write $[O\psi]_S$ and $[N\psi]_S$ to denote $[O\psi]_{(w, S)}$ and $[N\psi]_{(w, S)}$ respectively.

In [Levesque, 1990], Levesque proposes a sound and complete axiomatization for the classical logic of “only knowing”. It is worth noticing that all axioms are also valid in our fuzzy framework. We will check in particular the ones involving both operators:

Lemma 6.1. The following formula schemas are valid:

(i) $\phi \rightarrow B\phi$, where all variables and constants in $\phi$ occur in the scope of an operator $O$ or $B$,

(ii) $\phi \rightarrow N\phi$, where all variables and constants in $\phi$ occur in the scope of an operator $O$ or $B$,

(iii) $\neg B\phi \lor \neg N\phi$, if $\neg \phi$ is satisfiable and does not contain any modal operators.

Proof. Schemas (i) and (ii) are easy to check. For condition (iii), suppose $\neg \phi$ is satisfiable, i.e. there exists a structure $(w', S')$ such that $w'(\phi) = \|\phi\|_{(w', S')} = 0$. For a structure $(w, S)$ we have $\|\neg B\phi \lor \neg N\phi\|_{(w, S)} = 1$ if $\max(\|\neg B\phi\|_{(w, S)}, \|\neg N\phi\|_{(w, S)}) = 1$ if $\|B\phi\|_{(w, S)} = 0$ or $\|N\phi\|_{(w, S)} = 0$ if there exists $z \in S$ such that $z(\phi) = \|\phi\|_{(z, S)} = 0$ or there exists $z \notin S$ such that $z(\phi) = \|\phi\|_{(z', S)} = 0$. The latter is satisfied by the fact that there exists $w' \in \Omega_k$ such that $w'(\phi) = 0$.

Notice that (i) and (ii) are more general than both modal axioms (4) and (5) (see Section 3). The question of whether axioms (i), (ii) and (iii), together with the minimal modal logic axioms for $B$ and for $N$, provide a complete axiomatization with respect to the above semantics as well as its complexity is left as future work. We can only advance now that the logic is decidable since a re-adaptation of Lemma 4.1 easily shows that the “only knowing” logic over finitely-valued Lukasiewicz logic has the finite model property. Nothing about the complexity can be stated up to now, since no polynomial bound on the cardinality of the model has been fixed so far.

Nonetheless, we can show that the relationship between the “only knowing” operator $O$ and Moore’s stable expansions proved in [Levesque, 1990] nicely extends to our fuzzy framework with some slight variations. First of all, observe that if a formula $O\varphi$ gets value 1 in an epistemic state $S$, then necessarily $\varphi$ must be Boolean.

Lemma 6.2. For any formula $\varphi$, if $\|O\varphi\|_S = 1$ then $\|\varphi\|_{(v, S)} = 1$ for every $v \in S$ and $\|\varphi\|_{(w', S)} = 0$ for every $w' \notin S$, hence $\varphi$ is Boolean.

Next proposition shows that the belief set $Bel_S$ for an epistemic state defined by a set of $L_k$-evaluations $S$ is indeed a stable fuzzy expansion of a Boolean premise $\varphi$ whenever $\varphi$ is all that is fully believed in the epistemic state $S$.

Proposition 6.3. For any Boolean $L_k$-formula $\varphi$, $\|O\varphi\|_S = 1$ iff $Bel_S$ is a stable fuzzy expansion of $A = \{ \varphi \}$.

Proof. Since $\|O\varphi\|_S = \min(\|B\varphi\|_S, \|N\neg\varphi\|_S)$, we have the following chain of equivalences:
∥Oϕ∥_S = 1 \iff ∥Bϕ∥_S = 1 \text{ and } ∥N\neg ϕ∥_S = 1
\iff ∃v ∈ S, ∥ϕ∥_{(e,S)} = 1 \text{ and }
\forall v \notin S, ∥ϕ∥_{(e,S)} = 0
\iff S = \{ v \in Ω_k | ∥ϕ∥_{(e,S)} = 1 \}
\iff S \text{ is a fuzzy AE model of } A
\iff \text{Bel}_S \text{ is a stable fuzzy expansion of } A.

where the last equivalence follows from Prop. 5.2. Notice that the third “iff” is valid only in case ϕ is Boolean.

7 Related work and conclusions

In this paper we have introduced Hilbert-style axiomatizations of fuzzy modal logics for belief based on finitely-valued Łukasiewicz logic, in particular of many-valued counterparts of the well-known K45, KD45 and S5 modal logics. We have shown they provide possible-world semantics for a fuzzy autoepistemic logic, generalizing some bridges established in the classical case in [Levesque, 1990]. In the last years there has been some work on fuzzy modal logics with generalized Kripke semantics, see e.g. [Bou et al., 2011b] and references therein. Here we have focused on modal systems based on a finite set of linearly ordered truth-values with Łukasiewicz logic semantics for connectives which generate the class of finite MV-algebras [Mundici, 1987]. These systems represent a good compromise between expressive power and nice logical properties. The infinitely-valued case offers some problems, see e.g. [Hájek, 2010] for the case of S5 with total accessibility relations. On the other hand, a closely related work is Maruyama’s paper [Maruyama, 2011], where modal logics for belief based on a finitely-valued Heyting algebra of truth-values and hence not necessarily linearly ordered are considered. The formalization is very similar, but he also deals with common belief. Here we rather focus on providing a formal basis for the fuzzy generalization of autoepistemic logic developed in [Blondeel et al., 2013]. Many-valued extensions of autoepistemic logic have also been addressed by Fitting [Fitting, 1992] in the context of finite Heyting algebras of truth-values, as well as by Koutras et al. [Koutras and Zachos, 2000; Koutras et al., 1999]. As for future work, we plan to fully develop the “only believing” logic with the two modal operators O and N, and extend the modal approach to possibly other fuzzy logics.

Acknowledgments Blondeel is funded by a joint Research Foundation-Flanders (FWO) project; Flaminio is partially funded by the Italian project FIRB 2010 (RBFR10DGUA-002); Godo is partially funded by the Spanish projects EdTrE (TIN2012-39348-C02-01) and AT (CONSOLIDER CSD2007-0022, INGENIO 2010).

References

Modeling Reliability Varying over Time through a Labeled Argumentative Framework

Maximiliano C. D. Budán 1,2,3 Mauro J. Gómez Lucero 1,2 Guillermo R. Simari 2
1 - Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
2 - R&D Artificial Intelligence Laboratory (LIDIA) - Universidad Nacional del Sur
3 - Departamento de Matemática - Universidad Nacional de Santiago del Estero
e-mail: mcdb,mjg,grs@cs.uns.edu.ar

Abstract

Argumentation formalisms have been applied in modeling many real world applications such as recommender systems, legal systems, and in several areas of multi-agent systems. Besides the cognitive process agents go through when performing epistemic reasoning, they also should perform practical reasoning, i.e., reasoning about what to do, obtaining decisions and attaching more information to the pieces of knowledge involved. Labels allow to introduce the representation of uncertainty, strength, time availability, or any other feature about arguments, providing a mean to assist the process of determining argument acceptability.

We will introduce the Labeled Argumentation Framework (LAF), a framework based on the Argument Interchange Format (AIF), that incorporates labels as a medium to convey meta-level information. It will handle composite labels representing, for instance, time availability of arguments, together with a measure of strength possibly varying over time. The labels associated with arguments will be combined and propagated according to argument interactions, carrying information such as time and conclusive force. Through this process, we will establish argument acceptability, where the final labels propagated to the acceptable arguments will provide additional acceptability information.

1 Introduction

Commonsense reasoning makes use of different mechanisms, many of them are not yet well understood. Nevertheless, argumentation, a form of reasoning that has received special attention in the past years, contributes with a natural form of discerning between conflicting claims. In a general sense, argumentation can be defined as the study of the interaction of arguments for and against conclusions, with the purpose of determining which conclusions are acceptable [Bench-Capon and Dunne, 2007; Besnard and Hunter, 2008; Rahwan and Simari, 2009]. Several argument-based formalisms have emerged finding application in building autonomous agents and multi-agent systems. An agent may use argumentation to perform individual reasoning to resolve conflicting evidence or to decide between conflicting goals [Amgoud and Prade, 2009; Beierle et al., 2010]; Multiple agents may also use dialectical argumentation to identify and reconcile differences between themselves, through interactions such as negotiation, persuasion, and joint deliberation [Pasquier et al., 2011; van der Weide et al., 2011; Rahwan et al., 2003b; 2003a].

Adding meta-level information to the argumentation reasoning process in the form of labels attached to arguments will enhance the representational capabilities of the framework; a reason for this extension is that, besides the all-important property of soundness of an argument, there might be other considerations to take into account, for instance, each argument may have associated particular characteristics such as its strength [Bench-Capon, 2002] or reliability varying on time [Budán et al., 2012b]. Each label can be defined by a set of characteristics that is important to associate with an argument and the interaction between arguments can affect these labels. For that, we propose the introduction of an algebraic structure, called algebra of labels to manipulate meta-information associated to arguments in the argumentation domain. Thus, the introduction of labels will allow the representation of uncertainty, strength, or any other relevant feature of the arguments, providing a useful way of improving and refining the process of determining argument acceptability. Among the potential applications for the labels is the definition of an acceptability threshold which will determine if the arguments are strong enough to be accepted in a particular query, and the specification and manipulation of preferences associated with the arguments for determining which is more important in a specific domain.

In this paper we present a framework called Labeled Argumentation Framework (LAF) based in the Argument Interchange Format (AIF) [Chesñevar et al., 2006], which integrates the handling of labels. The labels associated with arguments will be combined and propagated according to argument interactions, such as support, aggregation, and conflict; in particular, aggregation has been studied in the form of argument accrual [Prakken, 2005; Verheij, 1995; Lucero et al., 2013]. Through this process, we will establish argument acceptability, where the final labels propagated to the accepted arguments provide additional acceptability information, such as degree of justification, restrictions on justification, explanation, etc. This general framework will be instantiated with composite labels representing time availabil-
ity of arguments, together with a strength measure possibly varying on time.

Below we will present an intuitive example to motivate and illustrate the goals of our work. Then, we will introduce a formalism that will allow the algebrization of the treatment of labels associated with the arguments, we called Algebra of Labels. We will present a brief introduction to AIF to present the elements necessary for our formalization as Labeled Argumentation Framework (LAF), that we will be exemplified as an application.

2 An Illustrative Example

The aim of our work is to contribute to the successful integration of argumentation in different artificial intelligence applications, such as Autonomous Agents in Decision Support Systems, Knowledge Management, and others of similar importance. In this section we will illustrate the usefulness of our formalization for a particular agent decision-making problem, where the strength of reasons supporting decisions may vary over time. Consider the following scenario:

Thomas, a real estate agent operating in Gotham City, is considering selling a property in the Northern suburb to get the best gain possible.

To reach a decision, Thomas considers reasons pondering their different strengths; given that he is pondering reasons, his reasoning will be based on argumentation. He also has to factor in that this measure of strength may vary over time as a consequence of different events happening in the environment; therefore, the problem is to determine in the present (time 0) which is the best moment (time interval) in the future to sell the property.

Tomas considers the following arguments:

A. The Property has an excellent construction, so it’s possible to sell it at a good price. (Note that the property deteriorates over time, so it makes sense that this argument will lose strength over time.)

B. It’s possible to sell the property at a good price, because it is located in a quiet area.

C. The property is in a quiet area, because most of the neighbors are retirees and peaceful people.

D. It’s not possible to sell the property at a good price, because it is located in an unsafe area.

E. The property is located in an unsafe area because of the high rate of assaults.

F. In the Northern suburb, the police force will be reinforced within 10 months, so, the crime rate will decrease.

G. The Northern suburb becomes safer, because the crime rate decrease.

H. Within a period of 28 months, apartments for college students will be finished in the Northern suburb; thus, the number of students living there will increase.

The area will not be quiet anymore because of the increase on the number of students living there.

The example illustrates how the knowledge used to decide can be naturally structured as arguments. These arguments interact in various ways such as support (e.g., C supports B), aggregation, where different arguments for the same conclusion reinforce each other (e.g., A and B), and conflicts involving arguments that support contradictory conclusions (e.g., B and D).

In many applications of argumentation, particularly in decision support systems, it is more natural to analyze arguments with the same conclusion together than to appraise them individually; this is known as aggregation, and it is based on the intuition that having more reasons for a given conclusion makes such a conclusion more credible [Prakken, 2005; Verheij, 1995; Lucero et al., 2013]. In the example, the arguments A and B supporting the action of selling the property in the Northern suburb should be considered—and weighted—together against the set of arguments supporting the conclusion against that decision (D). The example also illustrates the need to handle meta-information associated with arguments: each argument is associated with a measure of strength at each point in time, and this information must be propagated across support, combined on aggregation, and weighted to solve conflicts among arguments.

Finally, as is common in argumentation theories, when two arguments X and Y are in conflict, and X is preferred to Y, then Y is considered to be defeated by X, and X remains unaffected by Y attack; but, some applications need a more complex treatment of conflict, capturing the intuition the strength of X is somewhat diminished by Y attack, i.e., when there is no counter-argument, the strength of the support of a decision should not be the same as when they exist. That is, it is necessary to model the weakening effect an undefeated argument when counter-arguments exist.

We will introduce a general framework allowing to represent meta-information associated with arguments using labels, and providing the capability of defining acceptability by combining and propagating these labels according to the interactions of support, aggregation, and conflict. We will finally instantiate the proposed formalization to model a decision support system for the example presented in this section.

3 Abstract Algebra and Labels

Abstraction is a tool that allows to concentrate in what is important in a particular task; thus, the process of abstraction of irrelevant details leads to more comprehensive conceptualizations. In Mathematics, theories of various structures have been created that are applied to many different objects. In Abstract Algebra, entities such as Groups, Rings, Fields, Modules, and Vector Spaces are studied as algebraic structures; these structures are sets together with operations defined over them, where these operations satisfy particular axioms.

Because of its generality, abstract algebra is used in many fields of mathematics and science. For instance, Algebraic Topology uses elements of abstract algebra to study topological spaces, among others. We will now introduce an algebraic structure defined over labels to model, combine and propagate meta-information associated to arguments.

3.1 Algebra of labels

We will introduce an algebraic formalization for the representation and handling of meta-level information through the use
of labels that are attached to the elements of which arguments are built; a collection of operators will be used to combine and propagate the labels according to the different interactions affecting arguments, such as support, conflict, and aggregation. This Algebra of Labels is used to formalize a particular instantiation allowing to model the strength of arguments as it varies over time.

**Definition 1 (Algebra of labels)** An Algebra of labels is a 5-tuple \( \mathcal{A} = (\mathcal{E}, \circ, \oplus, \ominus, N_0) \) where:

- \( \mathcal{E} \) is a set of labels called domain of labels.
- \( \circ : \mathcal{E} \times \mathcal{E} \rightarrow \mathcal{E} \) is called the support operator.
- \( \oplus : \mathcal{E} \times \mathcal{E} \rightarrow \mathcal{E} \) is called the conflict operator.
- \( \ominus : \mathcal{E} \times \mathcal{E} \rightarrow \mathcal{E} \) is called the aggregation operator.
- \( N_0 \) is an identity element for the operators \( \circ \) and \( \oplus \).

The carrier set of this algebra is a set of argument labels; thus, the support operator will be used to obtain the meta-information associated with the conclusion of an inference by combining the meta-information (label) associated with the premises. The aggregation operator will be used to obtain the meta-information (label) that represents the collective strengthening of two or more reasons supporting the same conclusion shared by two or more arguments, reflecting an increased credibility for that conclusion. Finally, the conflict operator defines the meta-information (label) associated with a conclusion after considering the possible arguments that support conflicting claims.

We seek to show how the labels allow to represent uncertainty, reliability, time availability, or any other feature concerning arguments whose interpretation could provide assistance in determining argument acceptability. Next we will present the core ontology of the Argument Interchange Format (AIF); then, we will introduce a framework, called Labeled Argumentation Framework (LAF) that is based in this ontology.

### 4 Argument Interchange Format (AIF)

The Argument Interchange Format (AIF) core is currently specified as a core ontology that can be specialized to capture the representational requirements of a variety of argumentation formalisms. Argument entities are represented as nodes in a directed graph, informally called an argument network (full details in [Chesnèvar et al., 2006]).

![Figure 1: AIF Core Ontology](image)

There are two types of nodes, namely information nodes (I-nodes) and scheme application nodes (S-nodes) depicted with boxes and cans respectively in Figure 1. I-nodes are used to represent propositional information contained in an argument, such as a claim, premise, data, etc. S-nodes capture the application of schemes (i.e., patterns of reasoning). The present ontology has three different types of scheme nodes: rule of inference schemes, conflict schemes, and preference schemes; they respectively yield to three types of S-nodes: rule application nodes (RA-nodes), denoting applications of an inference rule or scheme, conflict application nodes (CA-nodes), denoting a specific conflict, and preference application nodes (PA-nodes), denoting specific preferences.

Nodes may possess different attributes such as title, creator, type (e.g., decision, action, goal, belief), creation date, evaluation (or strength, or conditional evaluation table), acceptability, and polarity (e.g., values such as pro or con); these attributes may vary and are not part of the core ontology. Most attributes are primitive, that is, essential to the node itself, while others are derived. It is conceivable a derived attribute such as acceptability may be obtained from node-specific attributes through calculation. In this case, the acceptability of an argument may be obtained from evaluation through mechanical inference. Nodes are used to build an AIF argument network, which is defined as follows.

**Definition 2 (Argument Network [Rahwan et al., 2007])**

An AIF argument network is a digraph \( G = (V, E) \), where:

- \( V = I \cup RA \cup CA \cup PA \) is the set of nodes in \( G \), where \( I \) is a set of I-Nodes, \( RA \) is a set of RA-Nodes, \( CA \) is a set of CA-Nodes, and \( PA \) is a set of PA-Nodes; and
- \( E \subseteq V \times V \setminus I \times I \), is the set of the edges in \( G \).

In the context of an argument network representing argument-based concepts and relations, a node \( A \) is said to support node \( B \) if and only if there is an edge running from \( A \) to \( B \). Edges do not need to be explicitly marked, labelled, or otherwise supplied with semantic pointers; basically, there are two types of edges: Scheme edges emanate from S-nodes and are meant to support conclusions that follow from the S-node, these conclusions may either be I-nodes or S-nodes, and Data edges emanating from I-nodes but necessarily ending in S-nodes, and are meant to supply data, or information to scheme applications. Thus, we can speak of I-to-S edges (information or data supplying edges), S-to-I edges (conclusion edges) and S-to-S edges (warrant edges). The table in Figure 2 describes the relations associated with the semantics of support as proposed in [Chesnèvar et al., 2006].

Here we will concentrate only in the relationships that are relevant to our formalization (these relationships are highlighted in figure 2). Notice that edges connecting I-nodes are forbidden, because I-nodes cannot be connected without an explanation that justifies that connection. There is always a scheme, justification, inference or rationale behind a relation between two or more I-nodes that is captured through an S-node. Moreover, only I-nodes can have zero incoming edges, as all S-nodes relate two or more components (for RA-nodes, at least one antecedent is used to support at least one conclusion; and for CA-nodes, at least one claim is in conflict with at least one other).
5 Labeled Argumentation based on AIF

We will introduce a framework based on the proposed standard Argument Interchange Format (AIF) that will provide the ability to represent special characteristics about claims through labels; we will refer to this framework as Labeled Argumentation Framework (LAF). Labels associated with the AIF’s elements will be also combined and propagated according to the argument interactions, such as support, conflict, and aggregation. Through this process, labels will participate in argument acceptability by providing additional information, such as degree of justification, restrictions on justification, explanations, among others.

Definition 3 (Labeled Argumentation Framework) A Labeled Argumentation Framework (LAF) is a 5-tuple $\Phi = \langle L, R, K, A, F \rangle$ where:

- $L$ is a logical language allowing to represent claims about the domain of discourse; in particular, a set of expressions involving the distinguished symbol “$\sim$” that denotes strong negation, is such that there is no element in $L$ containing a subexpression “$\sim\sim a$” (i.e., the consecutive application of two or more “$\sim$” is not permitted), and $L$ is closed by complement with respect to “$\sim$”, (i.e., if $s \in L$ then $\overline{s} \in L$, where $\overline{s}$, the complement of $s$, is $\sim s$).

- $R$ is a set of domain independent inference rules $R = \{R_1, R_2, \ldots, R_n\}$ defined in terms of $L$ (i.e., with premises and conclusion in $L$).

- $K$ is the knowledge base, such that $K \subseteq L$ describing the knowledge about the domain of discourse.

- $A$ is an algebra of labels (Definition 1), where the support operator $\otimes$ and the aggregation operator $\oplus$ are commutative and associative.

- $F$ is a function that assigns a label to each element of $K$, i.e., $F : K \rightarrow L$.

Observation: The use of two or more consecutive “$\sim$” in $L$ is not allowed to simplify the definition of conflict between claims; since the “$\sim\sim$” represents strong negation, this decision does not limit the expressive power of the language or the generality of the representation.

From a pragmatic point of view, the language $L$ is used to specify knowledge corresponding to a particular domain. The inferences that can be made from knowledge expressed in $L$ will be specified by the inference rules. In LAF, inference rules represent domain-independent patterns of reasoning, such as deductive inference rules (e.g., modus ponens, modus tollens) or non-deductive or defeasible inference rules (e.g., defeasible modus ponens).

In the application of the algebra of labels on LAF, the support operator will be used to obtain the label associated with the conclusion of an inference from the set of labels associated with the premises, where this set of premises does not have a specific order; therefore, we define the support and aggregation operators as binary operations that must satisfy commutativity and associativity. Next is an instantiation of LAF modeling the running example described in Section 2.

Example 1 Let $\Phi = \langle L, R, K, A, F \rangle$ be a LAF, where:

- $L$ is a first-order language.
- $R = \{dMP\}$, where the inference rule dMP is defined as follows:

\[
dMP : A_1, \ldots, A_n \quad A_1 \quad \ldots \quad A_n \Rightarrow A
\]

(Defeasible Modus Ponens)

- $A = \langle \mathcal{E}, \otimes, \oplus, N_\emptyset \rangle$ is an Algebra of labels, instantiated to represent and manipulate argument strength varying in time, in the following way:

  The domain of labels $\mathcal{E}$ is such that each label $\alpha \in \mathcal{E}$ has the form $\alpha = \langle (a_1, \tau_1), \ldots, (a_i, \tau_i), \ldots, (a_n, \tau_n) \rangle$, where $a_i \in \mathbb{N} \cup \{0\}$ is a strength value and $\tau_i$ is a subset of $\mathbb{R}$, i.e., $\tau_i \in 2^{\mathbb{R}}$ that represent a time interval. Also, $\tau_i$ are disjoint pairwise, i.e., $\tau_i \cap \tau_j = \emptyset$ for $i, j = 1, \ldots, n$ with $i \neq j$; this last condition establishes that no point in time should be associated with more than one strength value.

  In this particular case, a label $\alpha$ represents the strength that an argument has over different time intervals. In the example time is measure in months.

Let $\alpha, \beta \in \mathcal{E}$ be two labels, the operators of support, conflict and aggregation over labels are thus specified:

\[
\alpha \otimes \beta = \begin{cases} 
N_\emptyset, & \text{if } \beta \in (a, \tau_a) \in \alpha, (b, \tau_b) \in \beta, \text{ such that } \tau_a \cap \tau_b = \emptyset, \\
\left\{(\min(a, b), \tau_a \cap \tau_b) \mid (a, \tau_a) \in \alpha, (b, \tau_b) \in \beta \text{ and } \tau_a \cap \tau_b \neq \emptyset\right\}, & \text{otherwise.}
\end{cases}
\]

The operator $\otimes$ is based on the weakest link rule. Which is applied in strength and time (with respect to strength is analogous to P-DeLP).

\[
\alpha \oplus \beta = \begin{cases} 
\alpha, & \text{if } \beta = \emptyset, \\
\bigcup_{i=1}^{n} \text{Diff}((a_i, \tau_i), (b, \tau_b)) \oplus (\beta \setminus \{(b, \tau_b)\}), & \text{otherwise, where } (b, \tau_b) \in \beta.
\end{cases}
\]

where $\text{Diff}((a, \tau_a), (b, \tau_b))$ is defined as follows:

\[
\text{Diff}((a, \tau_a), (b, \tau_b)) = \begin{cases} 
\{(\max(a-b, 0), \tau_a \cap \tau_b); (a, \tau_a - \tau_b)\}, & \text{if } \tau_a - \tau_b \neq \emptyset \text{ and } \tau_a \cap \tau_b \neq \emptyset, \\
\{(\max(a-b, 0), \tau_a)\}, & \text{if } \tau_a - \tau_b = \emptyset, \\
\{(a, \tau_a)\}, & \text{otherwise.}
\end{cases}
\]

The operator $\oplus$ reflects weakening strength between conflicting literal, considering the time interval in which both are available.
Definition 4 (Argumentation graph)

sentence of tentative analysis derived from a LAF. We assume that each element of the availability of the arguments in the time.

The identity element \( N_\oplus = \{0, [0, \infty)\} \).

- \( K \) is the following knowledge base, showing next to each claim, the strength measure varying in time associated by \( r \). The value attached to each rule represents the strength of the connection between the antecedent and consequent of the rule.

\[
\begin{align*}
\text{r}_1 &: \text{goodPrice}(S, C) \leftrightarrow \text{goodConstr}(S, C) : \{(35, [0, -120])\} \\
\text{r}_2 &: \text{goodPrice}(S, C) \leftrightarrow \text{quiet}(S, C) : \{(40, [0, -120])\} \\
\text{r}_3 &: \text{quiet}(S, C) \leftrightarrow \text{goodNeighbors}(S, C) : \{(40, [0, -120])\} \\
\text{r}_4 &: \text{moreStuds}(S, C) \leftrightarrow \text{newStudApts}(S, C) : \{(50, [0, -120])\} \\
\text{r}_5 &: \text{newStudApts}(S, C) \leftrightarrow \text{safe}(S, C) : \{(50, [0, -120])\} \\
\text{r}_6 &: \text{highAssaults}(S, C) \leftrightarrow \text{lessCrime}(S, C) : \{(30, [0, -120])\} \\
\text{r}_7 &: \text{lessCrime}(S, C) \leftrightarrow \text{goodNeighbors}(S, C) : \{(30, [0, -120])\} \\
\text{r}_8 &: \text{goodNeighbors}(northern, gotham) : \{(50, [0, 60])\} \\
\text{r}_9 &: \text{highAssaults}(northern, gotham) : \{(50, [0, 120])\} \\
\text{r}_{10} &: \text{morePolice}(northern, gotham) : \{(50, [0, 120])\} \\
\text{r}_{11} &: \text{newStudApts}(northern, gotham) : \{(30, [28, -120])\}
\end{align*}
\]

Note that, the values the labels associated with the knowledge in \( K \) are chosen to illustrate representative cases of our approach.

Argumentation graphs will be used to represent the argumentative analysis derived from a LAF. We assume that there exist not two nodes in a given graph labelled with the same sentence of \( L \), so we will use the labeling sentence to refer to the I-node in the graph.

Definition 4 (Argumentation graph) Let \( \Phi \) be a LAF. The argumentation graph \( G_\Phi \) associated with \( \Phi \) is an AIF digraph \( G = (N, E) \), where \( N \) is the set of nodes, \( E \) is the set of the edges and the following conditions hold:

- each element of \( K \) is represented as an I-node \( X \in N \).
- for each application of an inference rule defined in \( \Phi \), there exists a RA-node \( R \in N \) such that:
  i) the inputs are all I-nodes \( P_1, P_2, \ldots, P_n \in N \) representing the premises required by \( R \).
  ii) the output is an I-node \( C \in N \) representing the conclusion.
- if \( X, \sim X \in N \), then there exists a CA-node such that its inputs/outputs are the I-nodes \( X \) and \( \sim X \).

Example 2 From the knowledge base \( K \) presented in the Example 1 we obtain the argumentation graph in Fig. 3.

\[ \text{Figure 3: Representation of an argument graph} \]

Once the argumentation graph \( G \) is constructed, we will associate two labels to each I-node in \( G \) representing the aggregation and conflict values for this I-node. The resulting graph is called Labeled argumentation graph, and the labeling process is captured in the following definition.

Definition 5 (Labeled argumentation graph) A labeled argumentation graph is an argumentation graph where each I-node \( X \) has two labels (or attributes): \( \alpha_X \) and \( \beta_X \) that holds the state of the claim after taking conflict into account. Let \( X \) be an I-node, then:

- if \( X \) has no inputs (and so stands for an element of \( K \)), it holds that \( \alpha_X = \beta_X = \mathcal{F}(X) \).
- otherwise, if an I-node \( X \) has inputs from RA-nodes \( R_1, \ldots, R_k \), where each \( R_i \) has premises \( X_1^{R_i}, \ldots, X_n^{R_i} \), then:
  \[ \alpha_X^+ = \left( \alpha_{X_1} \uplus \cdots \uplus \alpha_{X_n} \right) \uplus \cdots \uplus \left( \alpha_{X_1} \uplus \cdots \uplus \alpha_{X_n} \right) \]
- if a CA-node has as input an I-node \( X \) and I-node \( \sim X \), then \( \alpha_{X} = \alpha^+_X \oplus \alpha^+_{\sim X} \) and \( \alpha_{\sim X} = \alpha^+_{\sim X} \oplus \alpha_X \).

Once the I-nodes are labeled, we will define their acceptability status as follows:

Definition 6 (Acceptability status) For each I-node we assign one of three possible acceptability status according to their associated labels:

- strictly accepted if \( \alpha^+ = \alpha^- \)
• weakened accepted iff \( \alpha^+ \neq \alpha^- \) and \( \alpha^- \neq N_\beta \).
• not accepted iff \( \alpha^+ \neq \alpha^- \) and \( \alpha^- = N_\beta \).

**Example 3** Going back to Example 2, next we will calculate the labels for each claim (or I-node) of the argumentation graph, according to Definition 5.

<table>
<thead>
<tr>
<th>Claim</th>
<th>Label ( \alpha^+ )</th>
<th>Label ( \alpha^- )</th>
</tr>
</thead>
<tbody>
<tr>
<td>goodPrice((n,g))</td>
<td>((75, [0-28]); (45, [28-60]); (45, [60-120]))</td>
<td>((25, [0-10]); (55, [10-28]); (25, [28-60]); (25, [60-120]))</td>
</tr>
<tr>
<td>(\sim)goodPrice((n,g))</td>
<td>((50, [0-10]);(20, [10-120]))</td>
<td>((0, [0-10]); (40, [10-28]); (0, [28-60]); (0, [60-120]))</td>
</tr>
<tr>
<td>goodConstr((n,g))</td>
<td>((50, [0-60]); (40, [60-120]))</td>
<td>((50, [0-60]); (40, [60-120]))</td>
</tr>
<tr>
<td>(r_1)</td>
<td>((35, [0-120]))</td>
<td>((35, [0-120]))</td>
</tr>
<tr>
<td>quiet((n,g))</td>
<td>((40, [0-120]))</td>
<td>((40, [0-120]))</td>
</tr>
<tr>
<td>goodNeighbors((n,g))</td>
<td>((50, [0-120]))</td>
<td>((50, [0-120]))</td>
</tr>
<tr>
<td>(r_2)</td>
<td>((40, [0-120]))</td>
<td>((40, [0-120]))</td>
</tr>
<tr>
<td>moreStuds((n,g))</td>
<td>((30, [28-60]))</td>
<td>((30, [28-60]))</td>
</tr>
<tr>
<td>(r_3)</td>
<td>((30, [0-120]))</td>
<td>((30, [0-120]))</td>
</tr>
<tr>
<td>(\sim)quiet((n,g))</td>
<td>((30, [28-60]))</td>
<td>((0, [28-60]))</td>
</tr>
<tr>
<td>newStudApts((n,g))</td>
<td>((30, [28-60]))</td>
<td>((30, [28-60]))</td>
</tr>
<tr>
<td>(r_4)</td>
<td>((50, [0-120]))</td>
<td>((50, [0-120]))</td>
</tr>
<tr>
<td>(\sim)safe((n,g))</td>
<td>((50, [0-120]))</td>
<td>((50, [0-120]))</td>
</tr>
<tr>
<td>(r_5)</td>
<td>((50, [0-120]))</td>
<td>((50, [0-120]))</td>
</tr>
<tr>
<td>highAssaults((n,g))</td>
<td>((50, [0-120]))</td>
<td>((50, [0-120]))</td>
</tr>
<tr>
<td>(r_6)</td>
<td>((50, [0-120]))</td>
<td>((50, [0-120]))</td>
</tr>
</tbody>
</table>

\(\alpha^-_{\text{quiet}(n,g)} \boxplus \alpha^-_{\text{safe}(n,g)} = \{(\min(40, 40), [0-28]) \cap [10-120])\} = \{(40, [0-28])\} = \{(10, 28-120)\}\)

\(\alpha^+_{\text{goodPrice}(n,g)} = (N_\alpha^{\text{goodConstr}(n,g)} \boxplus \alpha^-_{\text{quiet}(n,g)}) \boxplus (\alpha^-_{\text{quiet}(n,g)} \boxplus \alpha^-_{\text{safe}(n,g)}) = \{(35, [0-60])\}; (35, [60-120])\) \boxplus \{(40, [0-28])\}; (10, [28-60])\) \boxplus \{(75, [0-28])\}; (45, [28-60])\); (45, [60-120])\}

\(\alpha^-_{\text{lessCrime}(n,g)} = \alpha^-_{\text{safe}(n,g)} \boxplus \alpha^+_{\text{safe}(n,g)} = \text{Diff}((50, [0-120]), (30, [10-120])\} = \{(50, [0-10])\}; (20, [10-120])\)\)

Next, we specify the acceptability status for each node in the Labeled argumentation graph.

- \(S = S^A \cup S^W\) is the set of accepted claims, where
  \(S^A = \{\text{goodNeighbors}(n,g), \text{goodConstr}(n,g), \text{highAssaults}(n,g), \text{morePolice}(n,g), \text{newStudApts}(n,g), \text{moreStuds}(n,g), \text{lessCrime}(n,g)\}\)
  is the set of strictly accepted claims, because \(\alpha^+ = \alpha^-\).

- \(S^W = \{\text{goodPrice}(n,g), \text{quiet}(n,g), \text{safe}(n,g)\}\)
  is the set of weakened accepted claims, because \(\alpha^+ \neq \alpha^- \) and \(\alpha^- \neq N_\beta\).
- \(S^D = \{\text{lessCrime}(n,g), \text{quiet}(n,g), \text{safe}(n,g)\}\)
  is the set of defeated claims, since \(\alpha^+ \neq \alpha^- \) and \(\alpha^- = N_\beta\).

Due to space limitations, Fig. 4 shows a relevant part of the labelled argumentative graph.

![Figure 4: Partial representation of a label argumentation graph.](image)

Note that, in the label argumentative graph the label on the left of an I-node corresponds to a \(\alpha^+\), and the one on the right, to a \(\alpha^-\). Finally, we obtain the strength of the argument “goodPrice\((n,g)\)” which is crucial to decide whether to sell the property, and responds to the following distribution over time: \(\{\text{25, [0-10]), (55, [10-28]), (25, [28-60]), (25, [60-120])\}\). These strength variations are consequence of the existence of reasons for against selling the property. Thus, all the information was taken into account and affected the acceptability status of the arguments. In conclusion, the best moment to sell the property is the time interval [10-28] with a strength of 55, reflecting that in this interval the property is likely to have the higher price.

The answer found expresses the degree in which the property has a good price through different time intervals; the decision of whether or not to sell and when, it is being obtained from the system based on the provided information.

### 6 Discussion and Related Work

The source of the initial motivation for our work was in the foundational work of Dov Gabbay on Labeled Deductive Systems (LDS) [Gabbay, 1993; 1996]. In [Gabbay, 1993] the handling of the labels as an algebra was first proposed. Gabbay introduced a general formalism to address the solution of problems requiring the use of logical frameworks together with other information that is dealt with the introduction of labeled deduction capabilities; thus, problems in amenable areas including temporal logics, database query languages, and defeasible reasoning systems can be handled better. In a labeled deductive system, labeled formulæ represented as \(L : \phi\),
where $L$ represents a label associated with the logical formula $\phi$, are used to produce labeled deductions; the attachment of labels to formulas gives the possibility of making the representation language more capable as a representational tool, with the addition of further flexibility and adaptability to different problems.

In [Chesnèvar and Simari, 2007], the formalism of labeled deductive systems was applied to argumentation systems. Using that tool, the proposal formally characterized different argument-based inference mechanisms putting them under the unifying structure of LDS. This work considers two inference operators and models dialectical analysis as it is performed in argumentation systems, the building of arguments, and the construction of the dialectical structure that appears in the form of trees.

Certainly inspired in the intuitions and formalisms contained in the aforementioned lines of research, we have introduced the use of labels whose application was characterized through an algebra of labels. Nevertheless, our aim was focused on a slightly different direction; we do not seek to formally compare different logics, but to extend the representational capabilities of argumentation frameworks by introducing the capability of representing additional information specific to the domain of application. Although it can be argued that, due to its extreme generality, Gabbay’s framework might also be somehow instantiated to achieve this purpose, we have proposed a specific way, advancing in a formal mechanism to propagate labels for the particular case of argument interactions, such as support, conflict and aggregation.

In recent works, T. J. M. Bench-Capon and J. L. Pollock introduced systems that are very influential in argumentation community. In [Bench-Capon, 2002], Bench-Capon persuasively posits that in situations involving practical reasoning, it is impossible to demonstrate conclusively that either party is wrong; thus, in such cases the role of argument is to persuade rather than to prove, demonstrate or refuse. In his own words: “The point is that in many contexts the soundness of an argument is not the only consideration: arguments also have a force which derives from the value they advance or protect.” [Bench-Capon, 2002]. In [Pollock, 2010], Pollock points out the fact that, in defeasible reasoning, most semantics ignore the issue of the inner force of arguments, i.e., that some arguments support their conclusions more strongly. But once we acknowledge that arguments can differ in strength and conclusions can differ in their degree of justification, things become more complicated. In particular, he introduces the notion of diminishes, which are defeaters that cannot completely defeat their target, but instead lower the degree of justification of that argument.

Another forerunner of our work is found on P-DeLP [Chesnèvar et al., 2004], where the elements of the language are labeled with possibilistic values and propagated to a final value for arguments that are constructed from these elements. In that work there was no attempt to further combine values between arguments.

Recently, in [Godo et al., 2012] Lluís Godo explored the possibility of expressing the uncertainty of temporal rules and events, and how this uncertainty may change over time. This allows the possibility of formalizing arguments and the corresponding defeat relations among them combining both temporal criteria introduced in t-DeLP [Pardo and Godo, 2011] and the belief strength criteria presented in P-DeLP [Alsinet et al., 2008b; 2008a].

Based on the intuitions of these three research lines, we formalize an argumentative framework, integrating AIF into the picture: in this system the labels are the way to represent the features of the arguments, generalizing the notion of value.

Clearly, the interaction between arguments can affect the labels they have associated, so that these changes can cause strengthening (through a form of accrual) and weakening (a form of diminishing) among arguments. It is worth to remark that the characteristics or properties associated with an argument can vary over time and be affected by various characteristics that influence in the real world, as, for instance, the reliability of a given source does [Buñán et al., 2012b; 2012a; Godo et al., 2012]. Using this framework, we established argument acceptability, where the final labels propagated to the accepted arguments provide additional acceptability information, such as degree of justification, explanation, and others.

7 Conclusions and Future Work

In many applications of argumentation, it is beneficial to associate meta-information to arguments for different purposes. For instance, in the implementation of an agent, it would be beneficial to establish the degree of success to be obtained by reaching a given objective; or, in the domain of recommender systems, it is interesting to provide recommendations together with a degree of reliability associated with it.

Our work was focussed on the development of a Labeled Argumentation Framework, (LAF), which is based on the Argument Interchange Format (AIF). In LAF, there is the possibility of representing uncertainty, reliability, time availability, degree of success, strength measure, appropriated combination of those, or any other feature that could help in representing an important characteristic of the arguments. In the example presented here, we represent strength measures varying over time associated with the arguments an agent takes as a basis for decision making. The reason of the variability of this measure may come from events occurring in the environment or the result of interaction between arguments, such as support, aggregation (or accrual), and conflict. Each of these relationships has an operation associated in the algebra of labels, which establishes how to propagate meta-information in the argumentation graph. From the labeling of the argumentation graph, it is possible to determine the acceptability of arguments and the resulting meta-data associated with them. A peculiarity of the conflict operation defined in the algebra is that it allows the weakening of arguments; this weakening contributes to a better representation of the real world for some application domains.

Currently, we are studying the formal properties of the operations of the algebra of labels we have proposed, considering how the acceptability relation is affected. We will also develop an implementation of LAF in the existing DeLP system ¹ as a basis; the resulting implementation will be exercised in different domains requiring to model extra informations.

¹See http://lidia.cs.uns.edu.ar/delp
tion associated with the arguments, taking as motivation studies and analysis of P-DeLP.

References


Logical representation of beliefs in the belief functions theory∗

Laurence Cholvy
ONERA
2 avenue Edouard Belin
31055 France
cholvy@onera.fr

Abstract

This paper presents a version of the belief functions theory in which masses are assigned to propositional formulas and it provides two combination rules which apply within this framework. It proves that belief function theory and its extension to non-exclusive hypotheses are two particular cases of this proposal. It finally shows that, even if this framework is not more expressive than the belief function theory as defined by Dempster and Shafer, its interest resides in the fact that it offers the user a rich language to express its beliefs, i.e. a propositional language.

1 Introduction

As defined in [Dempster, 1968] and [Shafer, 1976], the belief functions theory considers a frame of discernment which is a finite set of exclusive hypotheses and masses are assigned to elements of the power set of the frame i.e. to unions of hypotheses. [Dezert, 2002] extends this theory by relaxing the assumption of hypotheses exclusivity. Masses are then assigned to elements of a set named the hyper power set of the frame which are intersections of unions of hypotheses.

[Cholvy, 2012] showed that (i) in both case, the frame of discernment can be seen as a propositional language; (ii) under the assumption of hypotheses exclusivity, expressions on which masses are assigned are equivalent to propositional positive clauses; (iii) when relaxing this assumption, expressions on which masses are assigned are equivalent to some particular kind of conjunctions of positive clauses.

The question we ask in this present paper is the following: why can’t we be more general and consider expressions which are equivalent to conjunctions of clauses? I.e., why don’t we assign masses to expressions which are equivalent to propositional formulas? This question is motivated by the following example.

Example 1 We consider the example given by Shafer in [Shafer, 1990]: my friend Betty reports her observation about something which fell on my car.

∗This work has been supported by ONERA grant 20194.01F.

• (a) Betty tells me that a limb fell on my car. Suppose that from this testimony alone, I can justify a 0.9 degree of belief that a limb fell on my car. Modelling this leads to consider a frame of discernment with two exclusive hypotheses \{limb, nolimb\} and to consider the mass function: \(m(\{\text{limb}\}) = 0.9, \ m(\{\text{nolimb}\}) = 0.1\).

• (b) Assume now that Betty can distinguish between oaks, ashes and birches and that she tells me that an oak limb fell on my car. Modelling this within Dempster-Shafer Theory can be done by considering 4 exclusive hypotheses: \(H_1\) which represents the fact that an oak limb fell, \(H_2\) which represents the fact that a ash limb fell, \(H_3\) which represents the fact that a birch limb fell, \(H_4\) which represents the fact that no limb fell. My beliefs are then modelled by the mass function: \(m(\{H_1\}) = 0.9, \ m(\{H_1, H_2, H_3, H_4\}) = 0.1\). But we can also use Dezert’s model and consider the frame of discernment \{limb, nolimb, oak, ash, birch\} in which the pairs of exclusive hypotheses are: \(\{\text{limb, nolimb}\}, \{\text{oak, ash}\}, \{\text{oak, birch}\} \) and \(\{\text{ash, birch}\}\) only. My beliefs are then modelled by: \(m(\{\text{limb} \cap \{\text{oak}\}) = 0.9, \ m(\{\text{limb, nolimb, oak, ash, birch}\}) = 0.1\).

• (c) Consider finally that Betty tells me that a limb fell on my car and it was not an oak limb. Modelling this within Dempster-Shafer theory leads to the following mass function: \(m(\{H_2, H_3\}) = 0.9, \ m(\{H_1, H_2, H_3, H_4\}) = 0.1\). Modelling this within Dezert’s model leads to consider the mass function: \(m(\{\text{limb} \cap \{\text{ash, birch}\}) = 0.9, \ m(\{\text{limb, nolimb, oak, ash, birch}\}) = 0.1\).

Let us come back to point (c). We can notice that in both theories, the modeler has to reformulate the information it has to model. Indeed, the information Betty tells me is that a non-oak limb fell. Within the first model, this information is reformulated as: an ash limb fell or a birch limb fell (i.e., \(H_2, H_3\)). Within the second model, this information is reformulated as: a limb of an tree, which is an ash or a birch, fell (i.e., \(\{\text{limb} \cap \{\text{ash, birch}\}\}).

1Let us mention that, according to Shafer [Shafer, 1990], this degree is a consequence of the fact that my subjective probability that Betty is reliable is 0.9. However, in this paper, we do not discuss the meaning of such values nor we discuss the intuition behind the combination rules.
Our suggestion is to propose the modeller a language which allows it to express this information without any re-formulation.

For doing so, we will abandon set theory and use instead propositional logic for expressing information. More precisely, in this paper, we will allow the modeler to express information by means of a propositional language. This will lead it to express its beliefs as any kind of propositional formulas.

In the previous example, this will lead to consider the propositional language whose letters are limb, oak, ash, birch. The first focal element will then be modelled by limb \& \neg oak which is the very information reported by Betty. Notice that this propositional formula is a conjunction of two atomic clauses, the second one being negative.

This paper is organized as follows. Section 2 presents the propositional logic, the belief functions theory and its extension to non-exhaustive hypotheses. Section 3 describes this framework is not more expressive than belief function, presents two combination rules and it shows that belief functions theory and its extension to non-exclusive hypotheses are used when.

2 Propositional logic and belief function

2.1 Propositional logic

Let us first recall some definitions ans results that will be useful in the rest of the paper.

- A propositional language \( \Theta \) is defined by a set of propositional letters, connectives \( \neg, \land, \lor, \rightarrow, \leftrightarrow \) and parenthes. In what follows, it is sufficient to consider a finite language i.e a language composed of a finite set of letters.

- The set of formulas, denoted FORM, is the smallest set of words built on this alphabet such that: if \( a \) is a letter, then \( a \) is a formula; \( \neg A \) is a formula if \( A \) is a formula; \( A \land B \) is a formula if \( A \) and \( B \) are formulas. Other formulas are defined by abbreviation. More precisely, \( A \lor B \) denotes \( \neg(\neg A \land \neg B) \); \( A \rightarrow B \) denotes \( \neg A \lor B \); \( A \leftrightarrow B \) denotes \( (A \rightarrow B) \land (B \rightarrow A) \).

- A literal is a letter or the negation of a letter. In the first case it is positive, in the second it is negative.

- A clause is a disjunction of literals.

- A positive clause is a clause whose literals are positive.

- A clause \( C_1 \) subsumes a clause \( C_2 \) iff the literals of \( C_1 \) are literals of \( C_2 \).

- A formula is under minimal positive conjunctive normal form (denoted \( +mcnf \)) if it is a conjunction of positive clauses such that no clause subsumes another one.

• An interpretation \( i \) is a mapping from the set of letters to the set of truth values \( \{0, 1\} \). An interpretation \( i \) can be extended to the set of formulas by: \( i(\neg A) = 1 \) iff \( i(A) = 0 \); \( i(A \land B) = 1 \) iff \( i(A) = 1 \) and \( i(B) = 1 \).

Consequently, \( i(A \lor B) = 1 \) iff \( i(A) = 1 \) or \( i(B) = 1 \), and \( i(A \rightarrow B) = 1 \) iff \( i(A) = 0 \) or \( i(B) = 1 \); \( i(A \leftrightarrow B) = 1 \) iff \( i(A) = i(B) \).

• The set of interpretations of the language \( \Theta \) will be denoted \( I_\Theta \).

• The interpretation \( i \) is a model of formula \( A \) iff \( i(A) = 1 \). We say that \( i \) satisfies \( A \).

• The set of models (or truth-set) of formula \( A \) is denoted \( Mod(A) \). We have: \( Mod(A) \in 2^{I_\Theta} \).

• \( A \) is satisfiable iff \( Mod(A) \neq \emptyset \).

• \( A \) is a tautology iff \( Mod(A) = I_\Theta \). Tautologies are denoted by true.

• \( A \) is a contradiction iff \( Mod(A) = \emptyset \). Contradictions are denoted by false.

Let \( A \) and \( B \) be two formulas. \( B \) is a logical consequence of \( A \) iff \( Mod(A) \subseteq Mod(B) \). It is denoted \( A \models B \).

Let \( A \) and \( B \) two formulas. \( A \) and \( B \) are logically equivalent (or equivalent) iff \( Mod(A) = Mod(B) \). It is denoted \( A \equiv B \).

Any propositional formula is equivalent to a conjunction of clauses in which no clause is subsumed.

Let \( \sigma \) be a satisfiable formula. An equivalence relation denoted \( \equiv \) is defined on FORM by: \( A \equiv B \) iff \( \sigma \models A \leftrightarrow B \). Thus, two formulas are related by relation \( \equiv \) if and only if they are equivalent when \( \sigma \) is true.

Conventional: We will say that all the formulas of a given equivalent class of \( \equiv \) are identical.

For instance, if \( \sigma = \neg(a \land b) \) then \( c \lor (a \land b) \) and \( c \) are identical; \( a \lor b \) and \( b \land a \) as well.

Consequently, if \( \Theta \) is finite and if \( \sigma \) is a satisfiable formula, then we can consider the set of formulas is finite. It is denoted \( FORM^\sigma \).

For instance, consider that the two letters of \( \Theta \) are \( a \) and \( b \). If \( \sigma_1 = true \) then \( FORM^\sigma_1 = \{true, false, a, b, \neg a, \neg b, a \lor b, a \lor \neg b, \neg a \lor \neg b, a \land b, a \land \neg b, \neg a \land b, a \lor b \land \neg a \lor \neg b\} \). By convention any other formulas is identical to one of these. If \( \sigma_2 = \neg(a \land b) \) then \( FORM^\sigma_2 = \{true, false, a, b, \neg a, \neg b, a \lor \neg b, \neg a \lor \neg b, a \land \neg b, \neg a \land b, (a \lor \neg b) \land \neg (a \lor \neg b)\} \). By convention any other formula is identical to one of these.

With the previous convention, we denote \( CL^\sigma \) the finite set of clauses that can be built if one considers \( \sigma \). I.e., \( CL^\sigma = \{A : A \) is a clause and there is a formula \( \phi \) in FORM such that \( A \) is identical to \( \phi \} \).
• With the previous convention, we denote $+MCNF^\sigma$ the (finite) set of formulas which are under minimal positive conjunctive normal form. I.e., $+MCNF^\sigma = \{ A : A \in FORM^\sigma \text{ and } A \text{ is a } +mcnf \}$.

2.2 Belief function theory

Belief functions theory considers a finite frame of discernment $\Theta = \{ \theta_1, ... \theta_n \}$ whose elements, called hypotheses, are exhaustive and exclusive.

A basic belief assignment (or mass function) $m : 2^\Theta \rightarrow [0, 1]$ is a function such that: $m(\emptyset) = 0$ and $\sum_{A \in \Theta} m(A) = 1$.

Given a mass function $m$, one can define a belief function $Bel : 2^\Theta \rightarrow [0, 1]$ by: $Bel(A) = \sum_{B \subseteq A} m(B)$. One can also define a plausibility function $Pl : 2^\Theta \rightarrow [0, 1]$ such that $Pl(A) = 1 - Bel(\bar{A})$.

Let $m_1$ and $m_2$ be two mass functions on the frame $\Theta$. Dempster’s combination rule defines a mass function denoted $m_1 \oplus m_2$, from $2^\Theta$ to $[0, 1]$ by: $m_1 \oplus m_2(\emptyset) = 0$ and for any $C \neq \emptyset$:

$$m_1 \oplus m_2(C) = \frac{\sum_{A \cap B = C} m_1(A)m_2(B)}{\sum_{A \cap B \neq \emptyset} m_1(A)m_2(B)} \text{ for any } C \neq \emptyset$$

if $\sum_{A \cap B \neq \emptyset} m_1(A).m_2(B) \neq 0$

2.3 Extension to non-exclusive hypotheses

The formalism described by [Dezert, 2002], named DSmT, assumes that frames of discernment are finite sets of hypotheses which are exhaustive but not necessarily exclusive. It has then been extended in [Smarandache and Dezert, 2006] by considering integrity constraints. Here, we summarize this extended version, called hybrid model in [Smarandache and Dezert, 2006].

Let $\Theta = \{ \theta_1, ... \theta_n \}$ be a frame of discernment. An expression is under reduced conjunctive normal form (rcnf-expression) iff it is an intersection of unions of hypotheses of $\Theta$ such that no union contains another one.

The hyper-power-set, $D^\Theta$, is the set of all the expressions under reduced conjunctive normal form that can be built.

Integrity constraints are represented by $IC$, an expression of the form $E = \emptyset$, where $E \in D^\Theta$.

Taking integrity constraints into account comes down to considering a restriction of the hyper power set $D^\Theta$. This restricted set contains fewer elements than in the general case and we will denote it $D^\Theta_{IC}$ to indicate the fact that this new set depends on $IC$.

Example 2 If $\Theta = \{ \theta_1, \theta_2 \}$ then $D^\Theta = \{ \emptyset, \theta_1, \theta_2, \theta_1 \cap \theta_2, \theta_1 \cup \theta_2 \}$. Consider now the constraint $IC = (\theta_1 \cap \theta_2 = \emptyset)$. Then we get $D^\Theta_{IC} = 2^\Theta = \{ \emptyset, \theta_1, \theta_2, \theta_1 \cup \theta_2 \}$.

The mass functions are then defined on elements of the hyper-power-set $D^\Theta_{IC}$. Definitions of belief functions and plausibility functions are unchanged.

Finally, several combination rules have been defined in this theory in particular, the so called Proportional Conflict redistribution Rules (PCR) which redistribute the mass related to conflicting beliefs proportionally to expressions responsible of this conflict. They are several ways for redistributing\(^2\). Let us focus on the fifth rule called PCR5. This rule takes two mass functions $m_1$ and $m_2$, and builds a third mass function: $m_1 \oplus m_2 : D^\Theta_{IC} \rightarrow [0, 1]$ defined by:

$$m_1 \oplus m_2(\emptyset) = 0$$

And for any $X \neq \emptyset$, $m_1 \oplus m_2(X) = m_{12}(X)$ with

$$m_{12}(X) = \sum_{X_i \cap X_2 = X} m_1(X_1)m_2(X_2)$$

and

$$m_{12}(X) = \frac{m_1(X)^2m_2(Y) + m_2(X)^2m_1(Y)}{m_1(X) + m_2(Y) + m_2(X) + m_1(Y)}$$

(where any fraction whose denominator is 0 is discarded)

3 Our proposal

In this section, we present a formalism that allows a modeller to represent beliefs by assigning masses to propositional formulas.

3.1 Logical mass functions, logical belief functions, logical plausibility functions

We introduce new definitions for mass functions, belief functions and plausibility functions in order to apply them to propositional formulas.

Let $\Theta$ be a finite propositional language, $\sigma$ be a satisfiable formula.

Definition 1 A logical mass function is a function $m : FORM^\sigma \rightarrow [0, 1]$ such that: $m(\text{false}) = 0$ and $\sum_{A \in FORM^\sigma} m(A) = 1$.

Definition 2 Given a logical mass function, the logical belief function $Bel$ is defined by: $Bel : FORM^\sigma \rightarrow [0, 1]$ by:

$$Bel(A) = \sum_{B \in FORM^\sigma \text{ s.t. } B \rightarrow A} m(B)$$

Definition 3 Given a logical mass function, the logical plausibility function $Pl$ is defined by: $Pl : FORM^\sigma \rightarrow [0, 1]$ such that $Pl(A) = 1 - Bel(\neg A)$.

Theorem 1

$$Pl(A) = \sum_{(\sigma \land B \rightarrow A) \text{ is satisfiable}} m(B)$$

Proof 1 For readability, proofs of theorems are given in appendix

\(^2\)In this paper, we do not discuss the intuition behind the combination rules
Example 3 Let $\Theta$ be the propositional language whose letters are: limb, oak, ash, birch respectively meaning “what fell is a limb”, “what fell comes from an oak”, “what fell comes from an ash”, “what fell comes from an birch”.

Consider $\sigma = \text{limb} \rightarrow (\text{ash} \lor \text{ash} \lor \text{limb}) \land \neg (\text{oak} \land \text{limb}) \land \neg (\text{limb} \land \text{ash})$ and denoted $\sigma$, expressing that if a limb fell then it is comes from an oak, an ash or a birch, and these three types of trees are different.

Consider the following logical mass function: $m(\text{limb} \land \neg \text{oak}) = 0.8$, $m(\text{ash}) = 0.2$, $m(\text{A}) = 0$ for any other formula $A$. This expresses that my degree of belief in the piece of information “a non-oak limb fell” is $0.8$ and my degree of belief in the piece of information “what fell comes from an ash” is $0.2$.

The logical belief function is then defined by: $\text{Bel}(\text{limb} \land \neg \text{oak}) = 0.8$, $\text{Bel}(\text{ash}) = 0.2$, $\text{Bel}(\text{ash} \lor \text{birch}) = 1$, $\text{Bel}(\neg \text{oak}) = 1$, $\text{Bel}(\neg \text{birch}) = 0.2$, $\text{Bel}(\text{birch}) = 0$, $\text{Bel}(\text{limb}) = 0.8$, $\text{Bel}(\neg \text{limb}) = 0$, etc....

The logical plausibility function is then defined by: $\text{Pl}(\text{limb} \land \neg \text{oak}) = 1$, $\text{Pl}(\text{ash}) = 1$, $\text{Pl}(\text{ash} \lor \text{birch}) = 1$, $\text{Pl}(\neg \text{oak}) = 1$, $\text{Pl}(\neg \text{birch}) = 1$, $\text{Pl}(\text{birch}) = 0.8$, $\text{Pl}(\text{limb}) = 1$, $\text{Pl}(\neg \text{limb}) = 0.2$, etc....

3.2 Combination rules

Here we address the question of combining two logical mass functions. As it is done the belief functions community, we could define plenty of rules, according to the meanings of the masses or according to the properties that the combination rule must fulfill. But the purpose of this paper is not to define a new combination rule nor to comment the drawbacks or advantages of such or such existing rule.

This is why we arbitrarily propose two rules for combining two logical mass functions. The first one is called “logical DS rule”, denoted $\odot_L$; the second one is called “logical PCR5 rule” or “logical Proportional Conflict Redistribution Rule 5” and denoted $\oplus_L$. They are defined by the following definitions and illustrated on example 4.

Definition 4 Let $m_1$ and $m_2$ be two logical mass functions. The logical DS rule, denoted $\odot_L$, defines the logical mass function $m_1 \odot_L m_2 : \text{FORM}^\sigma \rightarrow [0,1]$ by:

$m_1 \odot_L m_2(\text{false}) = 0$ and for any $C \not\in \text{false}$

$$m_1 \odot_L m_2(C) = \sum_{\sigma \land (A \land B) \text{ satisfiable}} m_1(A) \cdot m_2(B)$$

if

$$\sum_{\sigma \land (A \land B) \text{ satisfiable}} m_1(A) \cdot m_2(B) \neq 0$$

Thus, according to this definition, the logical DS rule defines a logical mass function $m_1 \odot_L m_2$ such that the mass assigned to any formula $C$ which is not a contradiction, is the normalized sum of the product $m_1(A) \cdot m_2(B)$ where $A \land B$ and $C$ are identical (i.e., equivalent when $\sigma$ is true).

Definition 5 Let $m_1$ and $m_2$ be two logical mass functions. The logical PCR5 rule defines the logical mass function $m_1 \oplus_L m_2 : \text{FORM}^\sigma \rightarrow [0,1]$ by:

$m_{\oplus_L}(\text{false}) = 0$

$$m_{\oplus_L}(X) = m_1(X) + m_2(X) \text{ if } X \not\in \text{false}$$

with

$$m_1(X) = \sum_{X_1, X_2 \in \text{FORM}^\sigma} m_1(X_1) \cdot m_2(X_2)$$

and

$$m_2(X) = \sum_{X_1, X_2 \in \text{FORM}^\sigma} \left( \frac{m_1(X) \cdot m_2(Y)}{m_1(X) + m_2(Y)} \right) \cdot m_2(X) + m_1(Y)$$

(where any fraction whose denominator is 0 is discarded)

Thus, according to this definition, the logical PCR5 rule defines a logical mass function $m_1 \oplus_L m_2$ such that the mass assigned to any formula $X$ which is not a contradiction, is the sum of two numbers: the first number is the sum of the products $m_1(X_1) \cdot m_2(X_2)$ where $X_1 \land X_2$ and $X$ are identical (i.e., equivalent when $\sigma$ is true); the second number depends on the masses of the propositions which (under $\sigma$) contradict $X$.

Example 4 Let us consider $\Theta$ and $\sigma$ as in example 3. We consider here two witnesses: Betty and Sally. Suppose that Betty tells that a limb fell on my car but it was not an oak limb and that this testimony alone implies that I have the following degrees of belief: $m_1(\text{limb} \land \neg \text{oak}) = 2/3$, $m_1(\text{true}) = 1/3$.

Suppose that Sally tells that what fell on my car was not a limb but it comes down from an ash and that this testimony alone implies that I have the following degrees of belief: $m_2(\neg \text{limb} \land \text{ash}) = 2/3$, $m_2(\text{true}) = 1/3$.

1. Let us see what the logical DS rule provides.

First, notice that $\sigma \land (\text{limb} \land \neg \text{oak}) \land (\neg \text{limb} \land \text{ash})$ is not satisfiable. Thus $N = 5/9$. Consequently, the logical DS rule provides the following logical mass function:

$m_1 \odot_L m_2(\text{limb} \land \neg \text{oak}) = 2/5$, $m_1 \odot_L m_2(\neg \text{limb} \land \text{ash}) = 2/5$, $m_1 \odot_L m_2(\text{true}) = 1/5$.

Thus, the logical belief function $\text{Bel}$, associated with $m_1 \odot_L m_2$ is so that: $\text{Bel}(\text{limb}) = 2/5$, $\text{Bel}(\neg \text{limb}) = 2/5$, $\text{Bel}(\text{oak}) = 0$, $\text{Bel}(\neg \text{oak}) = 4/5$, $\text{Bel}(\text{ash}) = 2/5$, $\text{Bel}(\neg \text{ash}) = 0$, $\text{Bel}(\text{birch}) = 0$, $\text{Bel}(\neg \text{birch}) = 2/5$ etc.

And the logical plausibility function $\text{Pl}$ associated with $m_1 \odot_L m_2$ is so that: $\text{Pl}(\text{limb}) = 3/5$, $\text{Pl}(\neg \text{limb}) = 3/5$, $\text{Pl}(\text{oak}) = 1/5$, $\text{Pl}(\neg \text{oak}) = 1$, $\text{Pl}(\text{ash}) = 1$, $\text{Pl}(\neg \text{ash}) = 3/5$, $\text{Pl}(\text{birch}) = 3/5$, $\text{Pl}(\neg \text{birch}) = 1$ etc.

2. Let us consider now the logical PCR5 rule. This rule provides the following logical mass function:

$m_1 \oplus_L m_2(\text{limb} \land \neg \text{oak}) = 4/9$, $m_1 \oplus_L m_2(\neg \text{limb} \land \text{ash}) = 4/9$, $m_1 \oplus_L m_2(\text{true}) = 1/9$.

Thus the logical belief function which is associated with $m_1 \oplus_L m_2$ is so that:
Here, we consider a first particular case of our logical proposition. Then we show that if we consider another particular case then the logical framework reduces to the belief function theory.

Under the previous assumptions:

\[ m_1 \oplus_L m_2 \text{ is so that:} \]

- \[ P(limb) = 5/9, P(\neg\text{limb}) = 5/9, P(\text{oak}) = 5/9, \]
- \[ P(\text{limb} \land \neg\text{oak}) = 4/9, P(\text{ash} \lor \text{birch}) = 8/9, \text{etc.} \]

And the logical plausibility function which is associated with \( m_1 \oplus_L m_2 \) is so that:

- \[ P(limb) = 5/9, P(\neg\text{limb}) = 5/9, P(\text{oak}) = 5/9, \]
- \[ P(\text{limb} \land \neg\text{oak}) = 4/9, P(\text{ash} \lor \text{birch}) = 8/9, \text{etc.} \]

3.3 Particular cases

In this paragraph, we consider two particular cases of the logical framework previously introduced. We first show that if we consider a particular \( \sigma \) which expresses that hypotheses are exclusive and exhaustive and restrict formulas to clauses, then the logical framework reduces to the belief function theory. More precisely, we show that any logical mass function is the “logical” version of a mass function of the belief function. Then we show that if we consider another particular \( \sigma \) and restrict formulas to those of \(+MCNF^\sigma\), then the logical framework reduces to the extension of the belief function theory to non-exclusive hypotheses. Again, we show that any logical mass function is the “logical” version of a mass function of the extended version of belief function with non-exclusive hypotheses. In both case, we consider a frame of discernment which is isomorphic to the propositional language.

First particular case

Here, we consider a first particular case of our logical proposal in which:

- \( \sigma = (H_1 \lor \ldots \lor H_n) \land \bigwedge_{i \neq j} \neg(H_i \land H_j) \) where \( H_1, \ldots, H_n \) are the letters of the propositional language \( \Theta \). Thus, considering \( \sigma \) as true leads to consider that one and only one \( H_i \) is true.
- logical mass functions are restricted to those whose focal elements are clauses only. I.e., here, a logical mass function is a function \( m : CL^\sigma \rightarrow [0,1] \) such that:
  \[ m(\text{false}) = 0 \text{ and } \sum_{A \in \text{cl}^\sigma} m(A) = 1. \]

Definition 6 With \( \Theta \), we can associate a frame of discernment with \( \sigma \) exhaustive and exclusive hypotheses. These hypotheses are denoted \( f(H_1), \ldots, f(H_n) \) and the frame is denoted \( f(\Theta) \).

Theorem 2 Under the previous assumptions:

1. Any clause \( C \) of \( CL^\sigma \) can be associated with an unique expression of \( 2^{h(\Theta)} \).
2. Any logical mass function \( m \) whose focal elements are \( C_1, \ldots, C_k \) can be associated with an unique mass function denoted \( f(m) \) whose focal elements are \( f(C_1), \ldots, f(C_k) \) and so that each \( f(C_1) \) is given the mass \( m(C_1) \).
3. Given two logical mass functions \( m_1 \) and \( m_2 \), then for any clause \( C \) in \( CL^\sigma \), we have
  \[ m_1 \oplus_L m_2(C) = f(m_1) \oplus f(m_2)(f(C)) \]

Proof 2 See appendix.

Second particular case

Here, we consider a second particular case of our logical proposal in which:

- \( \sigma = (H_1 \lor \ldots \lor H_n) \land IC \) where \( H_1, \ldots, H_n \) are the letters of the propositional language \( \Theta \) and \( IC \) a propositional formula different from \( (H_1 \lor \ldots \lor H_n) \).
- logical mass functions are restricted to those whose focal elements are \(+mcnf\) formulas only. I.e., here, a logical mass function is a function \( m : +MCNF^\sigma \rightarrow [0,1] \) so that:
  \[ m(\text{false}) = 0 \text{ and } \sum_{A \in +MCNF^\sigma} m(A) = 1. \]

Definition 7 With \( \Theta \), we can associate a frame of discernment with \( \sigma \) exhaustive hypotheses. These hypotheses are denoted \( g(H_1), \ldots, g(H_n) \) and the frame is denoted \( g(\Theta) \).

Theorem 3 Under the previous assumptions:

1. Any formula \( F \) of \(+MCNF^\sigma \) can be associated with an unique expression of \( D_\sigma(\Sigma)^{g(\Theta)} \).
2. Any logical mass function \( m \) whose focal elements are \( F_1, \ldots, F_k \) can be associated with an unique mass function denoted \( g(m) \) whose focal elements are \( g(F_1), \ldots, g(F_k) \) and so that each \( g(F_i) \) is given the mass \( m(F_i) \).
3. Given two logical mass functions \( m_1 \) and \( m_2 \), then for any \(+mcnf\) formula \( F \) in \(+MCNF^\sigma \), we have
  \[ m_1 \oplus_L m_2(F) = g(m_1) \oplus g(m_2)(g(F)) \]

Proof 3 See appendix.

4 Comparison with belief function

4.1 Expressivity

According to theorems 2 and 3, the belief function theory and its extension to non-exclusive hypotheses can be expressed as two particular cases of the framework presented in section 3. However, we cannot conclude that the logical framework is more expressive than the belief function theory. Indeed in this paragraph, we show that any logical mass function can be reformulated into a mass function of the belief function theory, on condition that we consider a frame of discernment which may not be isomorphic to the language.

Let \( \Theta \) a propositional language whose letters are \( H_1, \ldots, H_n \) and \( \sigma \) a satisfiable formula.

Definition 8 With \( \Theta \), we can associate a frame of discernment whose hypotheses are exhaustive and exclusive. This frame, denoted \( h(\Theta) \), is isomorphic to \( \text{Mod}(\sigma) \) and the hypotheses are denoted \( h_1, \ldots, h_n \) being the size of \( \text{Mod}(\sigma) \).

Let us remark for instance that if \( \sigma \) is true, then the size of \( h(\Theta) \) is \( 2^n \). If \( \sigma = (H_1 \lor \ldots \lor H_n) \land \bigwedge_{i \neq j} \neg(H_i \land H_j) \), then the size of \( h(\Theta) \) is \( n. \) And if \( \sigma = (H_1 \lor \ldots \lor H_n) \land \bigwedge_{i \neq j} \neg(H_i \land H_j) \land IC \), for any \( IC \), then the size of \( h(\Theta) \) is strictly less than \( n \).

Theorem 4

1. Any formula \( F \) of \( \text{FORM}^\sigma \) can be associated with an unique expression of \( 2^{h(\Theta)} \) denoted \( h(F) \).
2. Any logical mass function \( m \) whose focal elements are \( F_1, \ldots, F_k \) can be associated with an unique mass function denoted \( h(m) \) whose focal elements are \( h(F_1), \ldots, h(F_k) \) and so that each \( h(F_i) \) is given the mass \( m(F_i) \).
3. Given two logical mass functions \( m_1 \) and \( m_2 \), then for any formula \( F \) in \( FORM_L \), we have
\[
m_1 \oplus_L m_2(F) = h(m_1) \oplus h(m_2)(h(F))
\]
\[
m_1 \oplus_L m_2(F) = h(m_1) \oplus h(m_2)(h(F))
\]

Example 5 Let us come back to example 4. Remember that \( \sigma = \text{limb} \rightarrow (\text{oak} \lor \text{ash} \lor \text{birch}) \land \neg(\text{oak} \land \text{ash}) \land \neg(\text{oak} \land \text{birch}) \). The two logical mass functions are:
\[
m_1(\text{limb} \land \neg \text{oak}) = 2/3, m_1(\text{true}) = 1/3 \text{ and } m_2(\neg \text{limb} \land \text{ash}) = 2/3, m_2(\text{true}) = 1/3.
\]

Then \( m_1 \) and \( m_2 \) can be associated with the following mass functions:
\[
h(m_1)(w_2, w_3) = 2/3, h(m_1)(w_1, ..., w_6) = 1/3 \text{ and } h(m_2)(w_5) = 2/3, h(m_2)(w_1, ..., w_6) = 1/3.
\]

We can easily check that \( m_1 \oplus_L m_2(F) = h(m_1) \oplus h(m_2)(h(F)) \) and \( m_1 \oplus_L m_2(F) = h(m_1) \oplus h(m_2)(h(F)) \).

Thus, the logical framework we introduced is not more expressive than the belief function theory: any logical mass function can be translated into a mass function within the belief function theory and combining two logical functions with the logical DS rule (resp, logical PCR5) is equivalent to combining with the Dempster’s rule (resp PCR5) the two corresponding mass functions.

4.2 Direct expression of beliefs
However, even if not more expressive than belief function, this framework allows one to directly express (i.e. with no translation) complex beliefs.

To illustrate the interest of this point, we consider the problem of uncertain semantic beliefs merging problem [Bellenger et al., 2011]. In this paper the authors deal with uncertain semantic beliefs i.e. mass functions defined on frames of discernment whose hypotheses are concepts of an ontology and are thus constrained by relations (relations of inclusions, intersections). One of the problems addressed in [Bellenger et al., 2011] is the combination of two such mass functions. For solving this problem, the authors propose to consider a new frame of discernment and to reformulate the two mass functions within this new frame as follows: if, according to the ontology, two hypotheses \( H_1 \) and \( H_2 \) intersect, then \( H_1 \) is reformulated by \( \{h_1, h_12\} \) and \( H_2 \) is reformulated by \( \{h_2, h_12\} \). If, according to the ontology, \( H_1 \) is included in \( H_2 \) then \( H_1 \) is reformulated into \( \{h_1\} \) and \( H_2 \) is reformulated into \( \{h_1, h_2\} \). This reformulation leads to a frame of discernment whose hypotheses are exclusive. Then the authors propose to apply Dempster’s rule of combination.

For illustrating their work, the authors consider the following example.

Example 6 Two information sources (a radar and an human) observe an object to be identified. The radar can only distinguish a land vehicle from an aircraft. The human can detail land vehicles. We thus consider the following ontology: Objects may only be aircrafts (A) or land vehicles (LV) but not both. Cars (C) and fire vehicle (FV) are land vehicles (FV).

The authors propose the following uncertain beliefs:
\[
m_1(\{L\}) = 0.6, m_2(\{A\}) = 0.1, m_2(\{LV, A\}) = 0.3
\]
\[
m_3(\{C\}) = 0.2, m_2(\{FV\}) = 0.4, m_2(\{LV\}) = 0.4
\]

Following the previous method, \( A \) is reformulated in \( \{C, FV\} \), \( FV \) in \( \{C, FV\} \), \( LV \) in \( \{C, FV\} \), \( FV \). Thus, the authors propose to apply the logical model introduced in this paper can be used for solving this problem directly, without any reformulation of expressions and by using the logical combination rule. The following example shows it.

Example 7 We choose a propositional language whose letters are: \( A, LV, C, FV \) (resp “the object is an airplane”, “the object is a land vehicle”, “the object is a car”, “the object is a fire vehicle”). The ontology is described by:
\[
\sigma = \neg (A \land LV) \land (A \lor FV) \rightarrow LV.
\]
Thus, the two logical mass functions are:
\[
m_1(LV) = 0.4, m_2(A) = 0.1, m_2(LV \lor A) = 0.3
\]
\[
m_3(C) = 0.2, m_2(FV) = 0.4, m_2(LV) = 0.4
\]

The logical DS rule produces the mass function:
\[
m(C) = 0.2, m(FV) = 0.4, m(LV) = 0.4
\]

Obiously, we get the same results. But the expressions on which masses are assigned are more compact and their meaning is clearer: the three different masses are assigned to the fact that the object is a car, a fire vehicle or a land vehicle. And obviously, no translation has been needed.

4.3 Computation complexity
However, what we gain from one side is lost from another. Indeed, even if complex beliefs can be expressed directly in this framework, without translation, computations for implementing combination rules are very expensive. They are based on several satisfiability checking tests (mainly of the form \( C \models (A \land B) \) or \( \sigma (A \land B) \) satisfiable) and these tests are highly complex. [Cook, 1971].

However, this complexity remains to be compared to the complexity of the translation introduced in definition 8.

5 Related works
We have not found in the literature any work proposing the logical extension described previously or something equivalent.
However, the relation between focal elements of belief function theory and propositional clauses is known for a long time.

Indeed, the link between expressions in set theory and propositional formulas was early mentioned in Shafer’s book (page 37) even if nothing more was done with this.

In [Provan, 1990], Provan proposes a logical view of Dempster Shafer theory. He shows that Dempster-Shafer theory can be modeled in terms of propositional logic so that focal elements of a mass function are modeled by clauses of a propositional language. Notice that this result is produced as a particular case of our model in section 3.3.

In [Kwisthout and Dastani, 2005], Kwisthout and Dastani aim to incorporate uncertainty to agent programming language. They ask the question: can belief function theory be applied to the beliefs of an agent if they are represented by logical formulas in an agent programming language? So their motivation is very close to ours. However, instead of remaining in logic and re-defining a logical version of belief function theory as we do here, they translate the problem in “classical” belief function theory. They show that modelling an agent’s uncertain beliefs about the propositional formula leads to define a mass function such that . But by defining such mass function, the problems of reasoning and combining are solved in the “classical” belief function theory. For instance, they use Dempster’s rule of combination to combine masses.

In [Lehmann, 2007], Lehmann is interested in reducing the cost of computations in belief function theory, due to the high numbers of variables. He proposes a compact representation of focal elements based on bitstrings. For doing so, he considers hypotheses which are interpretations of propositional formulas. And he obviously shows that, instead of assigning masses to unions of interpretations (i.e truth sets) one can assign masses to formulas. However, even if Lehmann’s motivation for finding a more compact representation is very interesting and meets ours (see example the analysis of the example in 1), he did not address the question of combination rules in this compact representation.

Finally, notice that the framework we introduced is not a logic and cannot be compared with the belief-function logic introduced in [Saffiotti, 1992] nor with the fuzzy modal logic for belief functions defined by [Godo et al., 2001].

6 Conclusion

This paper presented a version of the belief functions in which masses can be assigned to propositional formulas. Two combination rules have also been defined. It showed that belief function and its extension to non-exclusive hypotheses can be seen as particular cases. However, this framework is not more expressive than belief function since any logical mass function can be translated into a mass function. More, the logical combination rules which have been introduced are very expansive since they are based on many satisfiable tests but this complexity remains to be compared with the complexity of the mentioned translation. However, the interest of this logical version is that it allows to directly express the beliefs using logic. No translation is needed.

Another interest we see for this framework is purely formal: we guess that this framework can help people who are familiar with logic only, to understand belief function and the many combination rules that have been introduced. Indeed, even if we have focused on only two combination rules, we could give a logical version of any combination rule which has been introduced in the belief function community. But this remains to be done.

The framework we introduced here could be extended to first order logic beliefs. This would allow one to assign masses to first order formulas. We could then manipulate logical mass functions like: .

Acknowledgments

I sincerely thank the anonymous reviewers whose questions and comments helped me to improve this paper.

References


Appendix: proofs

Proof 1  
It is worth noticing that: $\sigma \models B \rightarrow \neg A$ is equivalent to $\sigma \land B \land A$ unsatisfiable. Thus, 
\[ 1 - \sum_{B \models \text{FORM}^o} m(B) = \sum_{B \models \text{FORM}^o} m(B) \] 
\[ (\sigma \land B \land A) \text{ is satisfiable} \]

Proof 2  
1. Any clause $C$ in $CL^o$ can be associated with an element in $2^{F\cup L}$, named $f(C)$ as follows:
- $f(\text{false}) = \emptyset$
- $f(\text{true}) = \{ f(H) \}$
- $f(\neg H_i) = \{ f(H_1), ... , f(H_{i-1}), f(H_{i+1}), ... f(H_n) \}$
- $f(H_1 \lor H_2 \lor ... \lor H_m) = f(H_1^r) \lor f(H_2^r) \lor ... \lor f(H_m^r)$ for $(m \geq 2)$

2. Let $m$ be a logical mass function defined by: $m(C_1) = m_1, ..., m(C_k) = m_k$. Then the function, $f(m)$ on $2^{F\cup L}$ defined by: $f(m)(f(C_1)) = m_1, ..., f(m)(f(C_k)) = m_k$ is a mass function.

3. Let $m_1$ and $m_2$ be two logical mass functions. First, we notice that $m_1 \oplus L m_2(\text{false}) = 0 = f(m_1) \oplus f(m_2)(\emptyset)$

Now consider $C, \not\models f(C)$ false. We show that:
- If two clauses $A$ and $B$ are so that $\sigma \land (A \land B)$ satisfiable then $f(A) \cap f(B) \neq \emptyset$. Thus $\sum_{A \land B \text{ satisfies}} m_1(A), m_2(B) = \sum_{f(A) \cap f(B) \neq \emptyset} f(m_1)(f(A)), f(m_2)(f(B))$.
- If $A$ and $B$ are two clauses so that $C, \not\models (A \land B)$ then $f(C) = f(A) \cap f(B)$. Thus $\sum_{C \not\models (A \land B)} m_1(A), m_2(B) = \sum_{f(C) = f(A) \cap f(B)} f(m_1)(f(A)), f(m_2)(f(B))$.

Consequently, $m_1 \oplus L m_2(C) = f(m_1) \oplus f(m_2)(f(C))$

Proof 3  
1. Any formula $F$ in $+MCFNF^o$ can be associated with an expression named $g(F)$ as follows:
- $g(F_1 \land F_2 ... \land F_k) = g(F_1) \cap g(F_2) \land ... \land g(F_k)$
- $g(H_1^r \lor H_2^r \lor ... \lor H_m^r) = g(H_1^r) \lor g(H_2^r) \lor ... \lor g(H_m^r)$ for $(m \geq 2)$
- $g(H_i) = \{ g(H_i) \}$
- $g(\text{false}) = \emptyset$

We can show that: $g(F)$ belongs to $D_{\sigma}^o$. Indeed, since $F$ is $+mcf$, $g(F)$ is under reduced conjunctive form thus belongs to $D_{\sigma}^o$. Furthermore, since $F$ belongs to $FORM^o$, then $g(F)$ belongs to the restriction of $D_{\sigma}^o$ by $g(\sigma)$, i.e., $D_{\sigma}^o$.

2. Let $m$ be a logical mass function defined by: $m(F_1) = m_1, ..., m(F_k) = m_k$. Then we can define a mass function, $g(m)$ on $D_{\sigma}^o$ by: $g(m)(g(F_1)) = m_1, ..., g(m)(g(C_k)) = m_k$

3. Let $m_1$ and $m_2$ be two logical mass functions. First, we have $m_1 \oplus L m_2(\text{false}) = 0 = g(m_1) \oplus g(m_2)(\emptyset)$

Now consider $F, \not\models f(F)$ false. We show that:
- If $F_1$ and $F_2$ are two $+mcf$ formulas so that $F = (F_1 \lor F_2)$ then $g(F) = g(F_1) \cap g(F_2)$.
- Thus, $\sum_{F \models (F_1 \lor F_2)} m_1(F_1), m_2(F_2) = \sum_{F \models (F_1 \lor F_2)} g(m_1)(g(F_1)), g(m_2)(g(F_2))$
- If two $+mcf$ formulas $F_1$ and $F_2$ are so that $\sigma \models (F_1 \lor F_2)$ a subset of $\sigma \models (F_1 \lor F_2)$ satisfiable then $g(F_1) \cap g(F_2) \neq \emptyset$. Thus $\sum_{F \models (F_1 \lor F_2) \text{ satisfies}} m_1(F_1), m_2(F_2) = \sum_{F \models (F_1 \lor F_2) \text{ satisfies}} g(m_1)(g(F_1)), g(m_2)(g(F_2))$

Consequently, $m_1 \oplus L m_2(F) = g(m_1) \oplus g(m_2)(g(F))$
Recursive Attack and Support in Abstract Argumentation Frameworks

Andrea Cohen, Sebastian Gottifredi, Alejandro J. García and Guillermo R. Simari

Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Artificial Intelligence Research and Development Laboratory,
Department of Computer Science and Engineering,
Universidad Nacional del Sur, Bahía Blanca, Argentina
E-mail: \{ac, sg, ajg, grs\} @cs.uns.edu.ar

Abstract

This work introduces the Attack-Support Argumentation Framework (ASAF), a novel approach to abstract argumentation extending the Argumentation Framework with Recursive Attacks (AFRA) in two ways. Firstly, it adds a support relation enabling to express support for arguments, for attacks, and for the support relation itself. Secondly, it extends AFRA’s attack relation allowing attacks to the support relation. Thus, the ASAF provides a unified framework for representing attack and support for arguments, as well as attack and support for the attack and support relations at any level.

1 Introduction

Argumentation is an attractive and effective paradigm for conceptualizing common-sense reasoning [Bench-Capon and Dunne, 2007; Besnard and Hunter, 2008; Rahwan and Simari, 2009]. Briefly, argumentation is a form of reasoning where a piece of information (claim) is accepted or rejected after considering the reasons (arguments) for and against that acceptance providing a reasoning mechanism capable of handling contradictory, incomplete and/or uncertain information.

Several approaches were proposed to model argumentation: on an abstract basis [Dung, 1995], using classical logics [Besnard and Hunter, 2001], or using logic programming [García and Simari, 2004]. Additionally, the argumentation process has been employed in various applications and domains such as decision making and negotiation [Amgoud et al., 2000; Black and Hunter, 2009], and multi-agent systems [Parsons et al., 1998; Amgoud et al., 2002].

Toulmin’s foundational work [Toulmin, 1958] introduced a model for the layout of arguments distinguishing between data, claim, warrant, backing, rebuttal, and qualifier; we can identify two forms of interaction among these elements. First, in addition to the data supporting the claim, the backing provides support for the warrant. Second, the presence of a rebuttal leads to the rejection of the claim through the defeat of the argument.

Starting from Dung’s approach [Dung, 1995], later works put aside the notion of support, focusing on abstract argumentation formalisms that only consider an attack relation between arguments. Several extensions of Dung’s frameworks were proposed over the years, including the consideration of attacks to the attack relation [Modgil, 2009; Baroni et al., 2009; 2011]. Recently, the study of the notion of support between arguments regained attention. The Bipolar Argumentation Frameworks [Amgoud et al., 2004; Cayrol and Lagasquie-Schiex, 2005] were the first to extend Dung’s work adding a general support relation between arguments. Then, several approaches with different interpretations of support such as evidential [Oren and Norman, 2008], deductive [Boella et al., 2010; Villata et al., 2012], necessity [Nouioua and Risch, 2010; 2011; Boudhar et al., 2012] and backing [Cohen et al., 2012] were proposed.

This substantial body of research shows that having attack and support relations between arguments is relevant; furthermore, adding attacks to attacks where preferences over conflicting arguments are expressed has also been proved useful, as in [Modgil, 2009; Baroni et al., 2009; 2011]. Also, allowing attacks to the support gives the possibility of overriding the acceptability constraints imposed by the support relation [Boella et al., 2010; Villata et al., 2012].

We will introduce the Attack-Support Argumentation Framework (ASAF), an abstract argumentation framework considering recursive attack and support between arguments, as well as the combination of the attack and support relations. Thus, this work provides a novel unified framework for representing attack and support for arguments, as well as attack and support for the attack and support relations at any level.

The rest of this work is organized as follows. Section 2 briefly presents the Argumentation Frameworks with Recursive Attacks [Baroni et al., 2009; 2011] and the Argumentation Frameworks with Necessities [Nouioua and Risch, 2010; 2011; Boudhar et al., 2012]. Section 3 formally introduces the Attack-Support Argumentation Framework (ASAF). Section 4 presents some properties of the ASAF. Finally, in Section 5 some related work and conclusions are discussed.

2 Background

Now we will introduce the background notions necessary for the proposed formalization of the Attack-Support Argumentation Framework, assuming Dung’s well-known framework [Dung, 1995]. First, we briefly review the Argumentation Frameworks with Recursive Attacks (AFRA) [Baroni et al., 2009; 2011]. Then, we will present the Argumentation
Frameworks with Necessities (AFN) [Nouioua and Risch, 2010; 2011; Boudhar et al., 2012].

**Definition 1 (AFRA).** An Argumentation Framework with Recursive Attacks (AFRA) is a pair \( \langle A, R \rangle \) where:

- \( A \) is a set of arguments; and
- \( R \) is a set of attacks, namely pairs \( \langle A, X \rangle \) s.t. \( A \in A \) and either \( X \in A \) or \( X \in R \).

Given an attack \( \alpha = \langle A, X \rangle \in R \), \( A \) is the source of \( \alpha \), denoted as \( \text{src}(\alpha) = A \), and \( X \) is the target of \( \alpha \), denoted as \( \text{trg}(\alpha) = X \).

Defeats to an argument or an attack in an AFRA are determined analyzing the recursive attack relation. Thus, defeats on arguments are also spread to the attacks they originate.

**Definition 2 (Defeat in AFRA).** Let \( \langle A, R \rangle \) be an AFRA, \( \alpha, \beta \in A \) and \( X \in A \cup R \).

- \( \alpha \) directly defeats \( X \) iff \( \text{trg}(\alpha) = X \).
- \( \alpha \) indirectly defeats \( \beta \) iff \( \text{trg}(\alpha) = X \) and \( X = \text{src}(\beta) \).

A graph-like representation for AFRA can be provided, where an attack \( \alpha = \langle A, X \rangle \) is denoted as \( A \rightarrow X \). To illustrate this, let us consider the following example presented in [Baroni et al., 2009].

**Example 1.** Suppose Bob is deciding about his Christmas holidays. He always buys last minute offers and he knows there are offers for going to a ski resort in Gstaad (\( G \)) or to a beach in Cuba (\( C \)). When possible, he prefers to go skiing (\( P \)); however, the weather report informs that it has not snowed in Gstaad (\( N \)). Notwithstanding this, Bob is informed that the ski resort in Gstaad has a good amount of artificial snow which makes it possible to ski (\( A \)). This situation can be represented by the AFRA \( \Gamma_1 = \langle \{G, C, P, N, A\}, \{\alpha, \beta, \gamma, \delta, \epsilon\} \rangle \), where \( \alpha = \langle G, C \rangle \), \( \beta = \langle C, G \rangle \), \( \gamma = \langle P, \beta \rangle \), \( \delta = \langle N, \gamma \rangle \), and \( \epsilon = \langle \alpha, N \rangle \). The graphical representation of \( \Gamma_1 \) is included below.

\[
\begin{align*}
A & \xrightarrow{\epsilon} N \xrightarrow{\delta} P \\
& \xleftarrow{\alpha} G \xleftarrow{\beta} C
\end{align*}
\]

Here, we have that \( \epsilon \) directly defeats \( N \) and indirectly defeats \( \delta \); \( \delta \) directly defeats \( \gamma \); \( \gamma \) directly defeats \( \beta \); \( \beta \) directly defeats \( G \) and indirectly defeats \( \alpha \); \( \alpha \) directly defeats \( C \) and indirectly defeats \( \beta \).

The authors in [Baroni et al., 2009; 2011] show that there exists a correspondence between an AFRA and a Dung’s Argumentation Framework (AF). Thus, by expressing an AFRA in terms of an AF they are able to reuse the properties and theoretical results available for Dung’s framework. As a result, they compute the acceptable arguments of an AFRA by using the associated Dung AF.

**Definition 3 (AF associated to an AFRA).** Let \( \Gamma = \langle A, R \rangle \) be an AFRA. The associated AF \( \tilde{\Gamma} = \langle \tilde{A}, \tilde{R} \rangle \) is defined as follows:

- \( \tilde{A} = A \cup R \);
- \( \tilde{R} = \{ (X, Y) \mid X, Y \in A \cup R \text{ and } X \text{ directly or indirectly defeats } Y \} \).

For instance, the AF \( \tilde{\Gamma}_1 \) associated to the AFRA \( \Gamma_1 \) is depicted below.

\[
\begin{align*}
A & \xrightarrow{\epsilon} N \xrightarrow{\delta} P \\
& \xleftarrow{\alpha} G \xleftarrow{\beta} C
\end{align*}
\]

According to the preferred semantics [Dung, 1995], the accepted set of arguments of \( \tilde{\Gamma}_1 \) (i.e., its only preferred extension) is \( \{A, P, \tilde{G}, \gamma, \epsilon, \alpha\} \).

Note that attacks in the original AFRA are arguments in the associated AF. In that way, an attack \( A \rightarrow X \) in an AFRA is represented as \( A \rightarrow X \) in the associated AF. Therefore, attacks in the original AFRA may appear in the extensions of the associated AF, because they have not been made ineffective by another attack. In addition, observe that the associated AF does not provide any connection between the arguments of the AFRA and the attacks they originate. For instance, given the AFRA \( \Gamma_1 \) of Example 1, in the associated AF \( \tilde{\Gamma}_1 \) there is no relation between argument \( P \) and the attack \( \gamma \) it originates.

In the following, a short presentation of the Argumentation Frameworks with Necessities (AFN) [Nouioua and Risch, 2010; 2011; Boudhar et al., 2012] is included.

**Definition 4 (AFN).** An Argumentation Framework with Necessities (AFN) is a tuple \( \langle A, R, N \rangle \) where \( A \) is a set of arguments, \( R \subseteq A \times A \) is an attack relation and \( N \subseteq A \times A \) is an irreflexive and transitive necessity relation.

The attack relation \( R \) is the same as in Dung’s argumentation frameworks. On the other hand, the support relation \( N \) has a necessity interpretation, establishing some acceptability constraints on the arguments it relates: if \( A \not\ni B \), it holds that if \( B \) is accepted, then \( A \) is also accepted and, conversely, if \( A \) is not accepted then \( B \) is not accepted either. From the original attacks and supports, additional attacks can be obtained.

**Definition 5 (Extended Attack).** Let \( \langle A, R, N \rangle \) be an AFN and \( A, B \in A \). There is an extended attack from \( A \) to \( B \), noted as \( ARB \), iff \( \exists C \in A \) s.t. either \( ARC \) or \( C \) and \( CNDA \). The direct attack \( A RB \) is considered as a particular case of extended attack.

An AFN can be graphically represented by a directed graph where nodes are arguments, and there are two kinds of edges: \( A \rightarrow B \) denotes \( A \triangleright B \) (attack), and \( A \equiv B \) denotes \( ANB \) (necessity). To illustrate these notions let us consider the following example.

**Example 2.** Consider the AFN \( N_2 \) depicted below on the left.
The graph above on the left depicts the attack and support relations of $AFN_2$. Note that, since the necessity relation is transitive, $A$ is necessary for $C$, and $D$ is necessary for $F$.

However, to simplify the graphical representation, the necessities derived by transitivity will not be explicitly shown, but they can be obtained by following the support paths on the graph. The graph above on the right summarizes the extended attacks obtained from $AFN_2$ which are depicted using dashed arrows except for those that are, in particular, direct attacks. Given that $D$ attacks $A$, and $A$ is necessary for $B$ and $C$, there are extended attacks from $D$ to $B$ and $C$. In addition, since $D$ attacks $A$ and is necessary for $E$ and $F$, there are also extended attacks from $E$ and $F$ to $A$.

In [Nouioua and Risch, 2011], the authors propose an alternative for computing the accepted arguments of an AFN, which consists on transforming an AFN into a Dung AF.

**Definition 6 (AF associated to an AFN).** Let $(A, R, N)$ be an AFN. The associated AF is $(A, \mathbb{R}^+)$. Note that the AF associated to an AFN takes the extended attack relation $\mathbb{R}^+$ into consideration. Recall that, since direct attacks are considered as a particular case of the extended attacks (see Definition 5), the associated AF contemplates both direct and extended attacks on the arguments of the original AFN. Then, by computing the acceptable arguments of the associated AF we obtain the acceptable arguments of the original AFN. For instance, given the $AFN_2$ depicted in Example 2 on the left, its associated AF would be as depicted on the right. Therefore, the only preferred extension of $AFN_2$ is $\{D, E, F\}$.

### 3 Attack-Support Argumentation Frameworks

Having briefly presented the required background material, now we will introduce the Attack-Support Argumentation Framework (ASAF), which is a novel approach extending the AFRA in two ways: i) it incorporates a support relation enabling to express support for arguments, for attacks, and for the support relation itself; ii) it extends the attack relation by allowing attacks to the support relation. The support relation in an ASAF has a necessity interpretation as presented in Section 2 and, therefore, it will impose some acceptability constraints on the elements it relates.

**Definition 7 (ASAF).** An Attack-Support Argumentation Framework is a tuple $(A, R, S)$ where $A$ is a set of arguments, $R \subseteq A \times (A \cup R \cup S)$ is an attack relation, $S \subseteq A \times (A \cup R \cup S)$ is an irreflexive and transitive support relation, and the attack and support relations are such that $R \cap S = \emptyset$.

The attack and support relations of an ASAF must be disjoint to prevent inconsistent arguments from belonging to the framework. That is, if there was $A \in A$ such that $(A, X) \in R$ and $(A, X) \in S$, argument $A$ would be inconsistent because it simultaneously attacks and supports $X$.

An ASAF can be graphically represented using a graph-like notation, where nodes are arguments and ‘$\rightarrow$’ and ‘$\leftarrow$’ respectively denote the attack and support relations. To simplify the notation, the attack (resp. support) from an argument $B$ to an element $\alpha = (A, X)$ in the attack (resp. support) relation of an ASAF will be represented as $(B, \alpha)$. Note that, since the attack and support relations of an ASAF are disjoint, a pair $\alpha = (A, X)$ in the attack relation or the support relation will be unequivocally identified by $\alpha$. Thus, when referring to $\alpha$, it will be possible to determine the attack or support it represents without problems.

For instance, let us consider the following example adapted from [Prakken, 2005], where attacks to the attack and support relations occur.

**Example 3.** Let us consider the following arguments:

- $A$ : “Car $C$ has airbags”
- $S$ : “Car $C$ is safe”
- $N$ : “Newspapers recently reported on airbags expanding without cause”
- $U$ : “Newspaper reports are very unreliable sources of technological information”

The arguments presented above and their connections can be represented by the ASAF $ASAF_3 = (A_3, R_3, S_3)$ with $A_3 = \{S, A, N, U\}$, $R_3 = \{\beta, \gamma\}$, and $S_3 = \{\alpha\}$ where $\alpha = (A, S)$, $\beta = (N, \alpha)$ and $\gamma = (U, \beta)$.

The graphical representation of $ASAF_3$ is depicted below.

\[
\begin{align*}
A & \rightarrow^\alpha S \\
& \leftarrow^\beta U \\
& \leftarrow^\gamma N
\end{align*}
\]

Here, the argument stating that the airbags expand without reason attacks the fact that having an airbag is necessary for the car to be safe. Similarly, that the newspapers are unreliable sources of technological information does not attack the argument regarding the information provided by the newspapers, but it attacks the authority of newspapers regarding technological information.

The following example presents a scenario where the support relation itself receives support.

**Example 4.** Suppose that having snow-chains (C) is necessary for driving a car on the route (D); however, this is necessary when the road is covered by snow (S). This situation can be characterized by the ASAF $ASAF_4$ depicted below.

\[
\begin{align*}
C & \leftarrow^\gamma S \\
& \rightarrow^\varepsilon D
\end{align*}
\]
Here, the nested support $S \leadsto \delta$ is giving the context under which the necessary support from $C$ to $D$ holds.

Acceptability of arguments in an ASAF is computed using an extension-based approach [Dung, 1995]. In order to do so, we will provide a transformation from an ASAF into an AFN. Each argument in the original ASAF will be an argument in the associated AFN. In addition, following the spirit of [Baroni et al., 2009; 2011] and [Boella et al., 2010], the attack and support relations in the original ASAF will be represented by arguments, attacks and supports in the associated AFN. Then, the resulting AFN will be used to obtain the accepted arguments of the ASAF. Next, we will present the intuitions behind this transformation, followed by its formal definition.

Attacks and supports in an ASAF are partly represented by attack-arguments and support-arguments in the associated AFN. Therefore, since they may appear in the extensions of the AFN, we need to determine what it means to have an attack/support-argument accepted. An attack $\alpha = (A, X)$ in the original ASAF is identified by an attack-argument $\alpha$ in the associated AFN. Thus, if $\alpha$ belongs to an extension of the associated AFN, then it means that the attack is active. For instance, given $\alpha = (A, X)$, if $\alpha$ belongs to an extension $E$ of the associated AFN, then the attack from $A$ to $X$ is active, implying that if $A$ is accepted (i.e., it belongs to $E$), then $X$ is not accepted (i.e., it does not belong to $E$).

Since the support relation of an ASAF has a necessity interpretation, a support $\beta = (B, Y)$ encodes the following constraints: if $Y$ is accepted, then $B$ is accepted and, conversely, if $B$ is not accepted, then $Y$ is not accepted either. We refer to these constraints as the positive and negative constraint respectively. Therefore, a support $\beta = (B, Y)$ in the original ASAF is related to two arguments in the associated AFN: the support-argument $\beta^+$, which represents that the support relation is active and corresponds to the positive constraint; and the support-argument $\beta^-$, which represents that the support relation is active and corresponds to the negative constraint. For instance, given $\beta = (B, Y)$, if $\beta^+$ belongs to an extension $E$ of the associated AFN, then it means that the necessary support between $B$ and $Y$ holds and it satisfies the positive constraint, implying that if $Y$ is accepted (i.e., it belongs to $E$), then $B$ is also accepted (i.e., it belongs to $E$ too).

In addition to the attack-arguments and the support-arguments described above, the attack and support relations of the original ASAF are represented in the associated AFN through a combination of attacks and supports. First, we analyze a simple scenario where an argument attacks another, as depicted below on the left.

\[
\begin{align*}
A &\alpha \rightarrow C \\
B &\beta \rightarrow \beta^+ \rightarrow \beta^-
\end{align*}
\]

In this case, the support from $B$ to $\alpha$ in the original ASAF expresses that $B$ is necessary for the attack from $A$ to $C$ to hold. Therefore, the non-acceptability of $B$ in the associated AFN will lead to the non-acceptability of $\alpha$, meaning that the attack from $A$ to $C$ is not active.

Regarding attacks to the support relation in an ASAF, let us consider the situation depicted below on the left.

\[
\begin{align*}
A &\alpha \rightarrow C \\
B &\beta \rightarrow \beta^+ \rightarrow \beta^-
\end{align*}
\]

In this case, the support from $B$ to $\alpha$ in the original ASAF expresses that $B$ is necessary for the attack from $A$ to $C$ to hold. Therefore, the non-acceptability of $B$ in the associated AFN will lead to the non-acceptability of $\alpha$, meaning that the attack from $A$ to $C$ is not active.

Taking the translation of attacks to arguments into account, an attack to an attack in an ASAF, as depicted below on the left, is directly translated by attacking the corresponding attack-argument, as depicted on the right.

\[
\begin{align*}
A &\alpha \rightarrow C \\
B &\beta \rightarrow \beta^+ \rightarrow \beta^-
\end{align*}
\]
The attacks from $\beta$ to $\alpha^+$ and $\alpha^-$ in the associated AFN depicted above on the right express that if the attack from $B$ to $\alpha$ in the original ASAF is active, then the support from $A$ to $C$ does not hold, overriding both positive and negative constraints of the necessary support relation.

Analogously, the support to a support relation in an ASAF, as depicted below on the left, is translated by making use of the support-arguments.

### Definition 8 (AFN associated to an ASAF)

Let $(A, R, S)$ be an ASAF. The associated AFN is $(\mathcal{A}, \mathcal{R}, \mathcal{N}, \mathcal{S})$, where:

- $\mathcal{A}^* = \mathcal{A} \cup \{ \alpha \mid (A, X) \in \mathcal{R} \} \cup \{ \beta^+, \beta^- \mid \beta = (B, Y) \in \mathcal{S} \}$.
- $\mathcal{R}^*$ and $\mathcal{N}^*$ are such that:
  1. if $\alpha = (A, X) \in \mathcal{R}$ and $X \in \mathcal{A} \cup \mathcal{R}$, then:
     - $(A, \alpha) \in \mathcal{N}^*$; and
     - $(\alpha, X) \in \mathcal{R}^*$.
  2. if $\beta = (B, Y) \in \mathcal{S}$ and $Y \in \mathcal{A} \cup \mathcal{R}$, then:
     - $(B, \beta^+) \in \mathcal{N}^*$;
     - $(\beta^+, \beta^-) \in \mathcal{R}^*$; and
     - $(\beta^-, Y) \in \mathcal{R}^*$.
  3. if $\alpha = (A, \beta) \in \mathcal{R}$, where $\beta = (B, Y) \in \mathcal{S}$, then:
     - $(A, \alpha) \in \mathcal{N}^*$;
     - $(\alpha, \beta^+) \in \mathcal{R}^*$; and
     - $(\alpha, \beta^-) \in \mathcal{R}^*$.
  4. if $\beta = (B, \gamma) \in \mathcal{S}$, where $\gamma = (C, Z) \in \mathcal{S}$, then:
     - $(B, \beta^+) \in \mathcal{N}^*$;
     - $(\beta^+, \beta^-) \in \mathcal{R}^*$;
     - $(\beta^-, \gamma^+) \in \mathcal{R}^*$; and
     - $(\beta^-, \gamma^-) \in \mathcal{R}^*$.

Note that the attacks to an argument or to an attack in an ASAF are grouped together in Definition 8. This is because arguments and attacks in an ASAF have one associated argument in the resulting AFN. Likewise, supports to an argument or to an attack are also grouped together. Finally, given an ASAF and its associated AFN, $E$ will be an ASAF’s extension if it is an extension of the associated AFN.

### Definition 9 (ASAF Extensions)

Let ASAF be an attack-support argumentation framework and AFN its associated argumentation framework with necessities. If $E$ is an extension of AFN under a given semantics, then $E$ is an extension of ASAF under the same semantics.

As in [Baroni et al., 2009; 2011], an extension of an ASAF might contain, in addition to arguments of the ASAF, attack-arguments and support-arguments of the associated AFN. In particular, the attack/support-arguments indicate respectively the attack/support relations active in the original ASAF. To illustrate this, let us consider the following examples.

#### Example 5

Let ASAF be the attack-support argumentation framework depicted below. The AFN associated to ASAF is depicted below.

In this case, there are no extended attacks between the arguments of ASAF. On the other hand, $U$ makes the attack $\beta$ from $N$ to the support $\alpha$ ineffective. Therefore, the support-argument $\alpha^+$ is accepted in AFN, meaning that the positive constraint of the support relation holds. Following the preferred semantics, the only preferred extension of AFN, and hence the preferred extension of ASAF, is $E_5 = \{ U, \gamma, N, A, \alpha^+, S \}$.

Let us now consider the following example, where support for an attack relation occurs.

#### Example 6

Let ASAF be the attack-support argumentation framework depicted below. The argumentation framework with necessities associated to ASAF is AFN, and is depicted below on the right.

The attack $\alpha$ from $E$ to $F$ is supported by $G$; thus, since $H$ attacks $G$, we have that $\alpha$ has lost its support. Following the preferred semantics, the only preferred extension of AFN is $E_5 = \{ H, \gamma, \beta^-, E, F \}$. As a result, the support-argument $\beta^-$ is accepted in AFN, meaning that the negative constraint of the support relation holds. Note that, in this case, the attack $\alpha$ has been made ineffective not by another attack, but by leaving it without the required support.
4 Attack and Support Semantics

The results in this section hold for complete, preferred, stable and grounded semantics (see [Dung, 1995] for details). From here on, when referring to a given semantics \( s \), it will correspond to one of Dung’s semantics mentioned above.

We will say that an attack \( \alpha = (A, X) \) is an active attack with respect to an extension \( E \) of an ASAF under a semantics \( s \) iff \( \alpha \) belongs to \( E \). Similarly, we will say that a support \( \beta = (B, Y) \) is an active support with respect to \( E \) iff one of the support-arguments \( \beta^+ \) or \( \beta^- \) belongs to \( E \). Note that, by Definition 8, it cannot be the case that both \( \beta^+ \) and \( \beta^- \) belong to \( E \) since they are conflicting arguments of the associated AFN. For instance, given the ASAF \( \gamma \) the following proposition shows that the attack relation of the ASAF is not an active attack, and \( \alpha \) is an active support with respect to the extension \( E_5 \).

It is important to remark that, given an attack \( \alpha = (A, X) \), the acceptance of \( A \) affects the activation of \( \alpha \), as shown by the following proposition.

\textbf{Proposition 1.} Let \( \Delta = \langle A, R, S \rangle \) be an ASAF and \( \alpha = (A, X) \in \mathbb{R} \). If \( A \) does not belong to an extension \( E \) of \( \Delta \) under a semantics \( s \), then \( \alpha \) does not belong to \( E \) either.

\textbf{Proof.} Since \( A \notin E \), there exists an attack from \( Y \) to \( A \) in the associated AFN and there is no \( Z \in E \) such that \( Z \) attacks \( Y \) in the associated AFN. By Definition 8, the attack \( \alpha = (A, X) \) in the ASAF is represented by a sequence \( A \rightarrow X \) in the associated AFN. Therefore, by Definition 5, the attack from \( Y \) to \( A \) leads to an extended attack from \( Y \) to \( \alpha \) in the associated AFN, preventing \( \alpha \) from belonging to \( E \).

Note that this proposition captures the behavior modeled by the indirect defeat of the Argumentation Frameworks with Recursive Attacks (see Definition 2).

The following proposition shows that the AFN associated to an ASAF satisfies the acceptability constraints imposed by the attack relation of the ASAF. That is, if \( A \) attacks \( X \) then \( A \) and \( X \) cannot be simultaneously accepted.

\textbf{Proposition 2.} Let \( \Delta = \langle A, R, S \rangle \) be an ASAF and \( \alpha = (A, X) \in \mathbb{R} \). If \( A \) belongs to an extension \( E \) of \( \Delta \) under a semantics \( s \) and \( \alpha \) is an active attack with respect to \( E \), then \( X \) does not belong to \( E \).

\textbf{Proof.} By Definition 8, the attack \( \alpha = (A, X) \) is translated as a sequence \( A \rightarrow X \) in the associated AFN. Therefore, since by hypothesis \( \alpha \in E \), all arguments attacked by \( \alpha \) will not belong to \( E \); in particular, \( X \) will not belong to \( E \).

Similarly to Proposition 2, the following proposition shows that the AFN associated to an ASAF satisfies the acceptability constraints imposed by the support relation of the ASAF. That is, if \( B \) supports \( Y \) then it holds that: \( i) \) if \( Y \) is accepted then \( B \) is also accepted, and \( ii) \) if \( B \) is not accepted then \( Y \) is not accepted either.

\textbf{Proposition 3.} Let \( \Delta = \langle A, R, S \rangle \) be an ASAF and \( \beta = (B, Y) \in S \). If \( \beta \) is an active support with respect to an extension \( E \) of \( \Delta \) under a semantics \( s \), then it holds that:

\( i) \) if \( Y \in E \), then \( B \in E \); and

\( ii) \) if \( B \notin E \), then \( Y \notin E \).

\textbf{Proof.} Since \( i) \) and \( ii) \) are logically equivalent, it suffices to prove \( ii) \).

\( ii) \) If \( B \notin E \), then there exists an attack from \( X \) to \( B \) in the associated AFN, and there is no \( Z \in E \) such that \( Z \) attacks \( X \) in the associated AFN. By Definition 8, the support \( \beta = (B, Y) \) in \( \Delta \) is represented as \( B \rightarrow \beta^+ \rightarrow \beta^- \rightarrow Y \) in the associated AFN. Thus, by Definition 5 there is an extended attack from \( X \) to \( \beta^- \), making \( \beta^+ \notin E \). As a result, it must be the case that \( \beta^- \notin E \) since by hypothesis \( \beta \) is active with respect to \( E \). Finally, every argument attacked by \( \beta^- \) in the associated AFN will not be in \( E \), in particular, \( Y \notin E \).

Finally, the following proposition shows that the ASAF extends the AFRA. That is, when considering an ASAF with an empty support relation, it is equivalent to an AFRA.

\textbf{Proposition 4.} Let \( \Delta = \langle A, R, \emptyset \rangle \) be an ASAF and \( \Gamma = \langle A, R \rangle \) be an AFRA. \( E \) is an extension of \( \Delta \) under a given semantics iff \( \emptyset \) is an extension of \( \Gamma \) under the same semantics.

\textbf{Proof Sketch.} We prove the equivalence by considering the transformations of the AFRA \( \Gamma \) and the ASAF \( \Delta \). By Definition 3, \( \Gamma \) is translated into a classical argumentation framework \( AF_1 \); and the extensions of \( AF_1 \) are the extensions of \( \Gamma \). By Definition 8, \( \Delta \) is translated into an argumentation framework with necessities \( AF_{N_\Delta} \); and the extensions of \( AF_{N_\Delta} \) are the extensions of \( \Delta \). Then, by Definition 6, \( AF_{N_\Delta} \) is translated into a classical argumentation framework \( AF_{\Delta} \); and the extensions of \( AF_{\Delta} \) are the extensions of \( AF_{N_\Delta} \). Thus, the extensions of \( AF_{\Delta} \) are the extensions of \( \Delta \). We show that, given \( \Gamma \) and \( \Delta \), the classical argumentation frameworks \( AF_1 \) and \( AF_{\Delta} \) are equal. The transformations of \( \Gamma \) and \( \Delta \) are such that given an attack \( \alpha = (A, X) \in \mathbb{R} \), \( A \) and \( X \) are arguments of \( AF_1 \) and \( AF_{\Delta} \). In addition, by Definition 3 the attack relation of \( AF_1 \) contains the direct and indirect defeats of \( \Gamma \). On the other hand, by definitions 8 and 6, the attack relation of \( AF_{\Delta} \) contains the direct attacks of \( AF_{N_\Delta} \) (the direct attacks of \( \Delta \), which correspond to the direct defeats of \( \Gamma \)) and the extended attacks of \( AF_{N_\Delta} \). Then, as shown by Proposition 1, the extended attacks of \( AF_{N_\Delta} \) capture the intuition of the indirect defeats in \( \Gamma \).

5 Conclusions and Related Work

We have introduced the Attack-Support Argumentation Framework (ASAF), which provides a unified setting for representing attack and support for arguments, as well as attack and support for the attack and support relations at any level. The ASAF extends the Argumentation Framework with Recursive Attack (AFRA) [Baroni et al., 2009; 2011] in two ways: \( i) \) it incorporates a support relation enabling to express support for arguments, for attacks, and for the support relation itself; \( ii) \) it extends the attack relation by allowing attacks to the support relation.

It was shown that acceptability of arguments in an ASAF is computed by translating the ASAF into an Argumentation Framework with Necessities (AFN) [Nouioua and Risch,
In particular, this transformation takes advantage from the fact that both ASAF and AFN make use of a support relation with a necessity interpretation. It was shown that this transformation is sound, in the sense that the associated AFN satisfies the acceptability constraints imposed by the attack and support relations of the original ASAF. Finally, it was also shown that the ASAF extends the AFRA; that is, an ASAF with an empty support relation is equivalent to an AFRA.

One of the first approaches to the representation of support and attack relations is DEFLOG [Verheij, 2003]. Briefly, its logical language has two connectives $\times$ and $\sim$. The dialectical negation $\sim S$ of a statement $S$ expresses that $S$ is defeated. The primitive implication $\sim$ is a binary connective used to express that one statement supports another and validates modus ponens. In DEFLOG is possible to combine and nest the connectives $\times$ and $\sim$ to obtain more complex statements, allowing to represent attacks to supports and attacks, as well as support to supports and attacks. However, DEFLOG is a sentence-based approach, whereas the ASAF proposed in this paper is an argument-based theory. In addition, the semantics of DEFLOG’s primitive implication differs from the semantics of the support relations used in abstract argumentation approaches. In particular, the support relation of the ASAF encodes a necessity between the elements it relates, whereas the primitive implication in DEFLOG is used to obtain new sentences through the use of modus ponens.

The study of attacks to attacks in abstract argumentation frameworks was addressed by several approaches such as [Modgil, 2009; Baroni et al., 2009; 2011; Boella et al., 2010; Villata et al., 2012]. However, the work by [Baroni et al., 2009; 2011] was the only to consider a recursive attack relation which enables to represent attacks to attacks at any level. On the other hand, the meta-argumentation approach of [Boella et al., 2010; Villata et al., 2012] also accounted for a support relation between arguments, allowing for the representation of attacks to the support relation. However, this interaction was fixed, thus not being able to combine and nest the attack and support relations at any level.

Starting from [Amgoud et al., 2004; Cayrol and Lagasque-Schiex, 2005], the study of a support relation in abstract argumentation frameworks regained attention among the researchers. Several formalizations were proposed in the literature, where different interpretations of support such as evidential [Oren and Norman, 2008], deductive [Boella et al., 2010; Villata et al., 2012], necessity [Nouioua and Risch, 2010; 2011; Boudhar et al., 2012] and backing [Cohen et al., 2012] were considered. In particular, the support relation of the ASAF proposed in this work follows the necessity interpretation of the Argumentation Frameworks with Necessities. Finally, it is important to remark that none of the existing approaches so far have addressed the possibility of allowing support for the support relation itself. Moreover, they have neither considered all the alternatives for combining the attack and support relations, as considered in the ASAF.

References


Towards Instance Query Answering for Concepts Relaxed by Similarity Measures

Andreas Ecke$^*$

Theoretical Computer Science,
TU Dresden, Germany

Rafael Peñaloza$^†$

Theoretical Computer Science,
TU Dresden, Germany

Anni-Yasmin Turhan$^‡$

Theoretical Computer Science,
TU Dresden, Germany

Center for Advancing Electronics Dresden

Abstract

In Description Logics (DL) knowledge bases (KBs) information is typically captured by crisp concept descriptions. However, for many practical applications querying the KB by crisp concepts is too restrictive. A controlled way of gradually relaxing a query concept can be achieved by the use of similarity measures.

To this end we formalize the task of instance query answering for crisp DL KBs using concepts relaxed by similarity measures. We identify relevant properties for the similarity measure and give first results on a computation algorithm.

1 Introduction

Description Logics (DLs) are a family of knowledge representation formalisms that have unambiguous semantics. A particular DL is characterized by a set of concept constructors, which allow to build complex concept descriptions. Intuitively, concept descriptions characterize categories from an application domain. In addition, binary relations on the domain of interest can be captured by roles. These in turn can be used in concept descriptions. The terminological knowledge of an application domain is stored in the TBox, where complex concept descriptions can be assigned to concept names. Facts from the application domain and relations between them are represented by individuals in the ABox. TBox and ABox together form the DL knowledge base (KB).

The formal semantics of DLs allow the definition of a variety of reasoning services. The most prominent ones are subsumption, i.e. to compute whether a sub-concept relationship holds between two concept descriptions and instance query answering, where for a given concept description all individuals from an ABox that are instances of the concept are computed. These reasoning services are implemented in highly optimized reasoning systems, see for example [Tsarkov and Horrocks, 2006; Kazakov et al., 2012; Haarslev et al., 2012].

DLs of varying expressivity are the underlying logics for the W3C standardized ontology language OWL 2 and its profiles [Motik et al., 2009]. This has led to an increased use of DLs and DL reasoning systems in the recent years in many application areas. By now there is a large collection of KBs written in these languages. However, many applications need to query the knowledge base in a less strict fashion.

In the application area of service matching OWL TBoxes are employed to describe types of services. Here, a user request for a service specifies several conditions for the desired service. These conditions are represented by a concept description. For such a concept description the OWL ABox that contains the individual services is searched for a service matching the specified request by employing instance query answering. In cases where an exact match with the provided requirements is not possible, a “feasible” alternative needs to be retrieved from the ABox containing the services. This means that those individuals from the ABox should be retrieved for the given query concept that fulfill the main conditions, while for some conditions only a relaxed variant is fulfilled.

A natural idea on how to relax the notion of instance query answering is to simply employ fuzzy DLs and perform query answering on a fuzzy variant of the initial query concept. However, on the one hand reasoning in fuzzy DLs easily becomes undecidable [Borgwardt et al., 2012; Borgwardt and Peñaloza, 2012; Cerami and Straccia, 2013] and on the other hand depending on the user and on the request, different ways of relaxing the query concept are needed. For instance, for a request to a car rental company to rent a particular car model in Beijing, it might be acceptable to get an offer for a similar car model to be rented in Beijing, instead of getting the offer to rent the requested car model in London. Whereas for a handicapped user in a wheelchair it might not be acceptable to relax the requested car model from a two-door one to a four-door one. Here fuzzy concepts would relax the initial concept in an unspecific and uniform way. Ideally, relaxed instance query answering should allow to

1. choose which aspects of the query concept can be relaxed and

$^*$Supported by the German Research Foundation (DFG) Graduiertenkolleg 1763 (QuantLA).

$^†$Partially supported by DFG within the Cluster of Excellence ‘cfaED’

$^‡$Partially supported by DFG in the Collaborative Research Center 912 “Highly Adaptive Energy-Efficient Computing”.
2. choose the degree to how much these aspects can be relaxed.

The reasoning service addressed in this paper is a relaxed notion of instance querying, such that it allows for a given query concept the selective and gradual extension of the answer set of individuals. We develop a formal definition of this reasoning service in Section 3.

Our approach for achieving selective and gradual extension of the answer sets is to employ concept similarity measures to relax the query concept. A concept similarity measure yields, for a pair of concept descriptions, a value from the interval \([0, 1]\) — indicating how similar the concepts are. The goal is to compute for a given concept \(C\), a concept similarity measure \(\sim\) and a degree \(t (t \in [0, 1])\), a set of concept descriptions such that each of these concepts is similar to \(C\) by a degree of at least \(t\), if measured by \(\sim\), and finding all their instances.

For DLs there is whole range of similarity measures defined (see for example [Borgida et al., 2005; d’Amato et al., 2005; Lehmann and Turhan, 2012]), which could be employed for this task. In particular the similarity measures generated by the framework described in [Lehmann and Turhan, 2012] allow users to specify which part of the vocabulary used in their knowledge base is to be regarded more important when it comes to the assessment of similarity of concepts. Thus, these measures naturally allow to select which aspect of the query concept to relax.

The core reasoning problem encountered in our algorithm for relaxed instance query answering is to compute for an individual \(a\) and the query concept description \(C\) a concept description \(C'\) that mimics \(C\), i.e. a concept description that is ‘sufficiently similar’ to \(C\) w.r.t. the used similarity measure \(\sim\) and the degree \(t\).

We propose in this paper an algorithm to compute the above mentioned reasoning service of relaxed instance query answering in the lightweight DL \(\mathcal{EL}\). For instance, for the Gene ontology [Gene Ontology Consortium, 2000], which is written in \(\mathcal{EL}\) and is used (among other things) to solve the task of finding genes that realize similar functionality [Lord et al., 2003], a proliferation of different similarity measures has been defined [Lord et al., 2003; Schlicker et al., 2006; Mistry and Pavlidis, 2008; Alvarez and Yan, 2011]. In principle these measures could be used in our approach to query ABoxes. We identify properties of concept similarity measures that allow to compute relaxed instances of concepts.

The paper is organized as follows: after introducing basic notions on DLs and concept similarity measures in Section 2, we develop a formal notion of relaxed instances in Section 3. In order to compute relaxed instances it is necessary, as we shall see, to compute mimics of a concept and an individual. An way of finding a mimic and its application to construct an algorithm that computes all relaxed instances of a query concept is provided in Section 4. As customary, the paper ends with conclusions and future work.

### Table 1: Concept definitions, TBox axioms and ABox assertions for \(\mathcal{EL}\).

<table>
<thead>
<tr>
<th>Concept Definition</th>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>top concept</td>
<td>(\top)</td>
<td>(\top \equiv \Delta_I)</td>
</tr>
<tr>
<td>conjunction</td>
<td>(C \sqcap D)</td>
<td>((C \sqcap D)I = C \sqcap D)</td>
</tr>
<tr>
<td>existential restriction</td>
<td>(\exists r.C)</td>
<td>({d \in \Delta_I \mid \exists e. (d, e) \in r_I \land e \in C_I})</td>
</tr>
<tr>
<td>concept definition</td>
<td>(A \equiv C)</td>
<td>(A \equiv C)</td>
</tr>
<tr>
<td>role assertion</td>
<td>(r(a, b))</td>
<td>((a_I, b_I) \in r)</td>
</tr>
</tbody>
</table>

2. Preliminaries

In this section we introduce the basic notions of Description Logics and similarity measures between concepts. For a thorough introduction to Description Logics, see [Baader et al., 2003]. While we try to formalize the notion of relaxed instances of a concept w.r.t. a similarity measure independently from a specific DL, Section 4 will show how instance querying for relaxed concepts can be computed in the restricted DL \(\mathcal{EL}\).

Let \(N_C, N_R,\) and \(N_I\) be non-empty, disjoint sets of concept names, role names, and individual names. A concept description (or short concept) is constructed from concept names by applying concept constructors such as conjunction, negation, quantification, or the top concept \(\top\). In particular, \(\mathcal{EL}\) only admits the concept constructors conjunctions, existential restrictions and the top concept, as seen in Table 1. We denote the set of all \(\mathcal{L}\)-concept descriptions constructed is such a way by \(C(C)\).

For example, using the following \(\mathcal{EL}\)-concept description, one can describe a service which currently waits for requests, but runs on an overloaded server:

\[
\text{Service} \sqcap \exists \text{has-state.WaitingForRequest} \\
\sqcap \exists \text{runs-on} (\text{Server} \sqcap \exists \text{has-condition.Overloaded})
\]

The semantics of concept descriptions is defined by means of interpretations \(I = (\Delta_I, I)\) consisting of a non-empty domain \(\Delta_I\) and an interpretation function \(I\) that assigns binary relations on \(\Delta_I\) to role names, subsets of \(\Delta_I\) to concept names, and elements of \(\Delta_I\) to individual names. The interpretation function can be recursively extended to \(\mathcal{EL}\)-concept descriptions as shown in Table 1.

An \(\mathcal{EL}\)-knowledge base \((\text{KB})\) \(K = (T, A)\) consists of an \(\mathcal{EL}\)-TBox \(T\), which captures the terminological knowledge, and an \(\mathcal{EL}\)-ABox \(A\), which contains the assertions about specific individual. In this paper we only consider unfoldable TBoxes, i.e., sets of concept definitions such that each concept name occurs at most once on the left-hand side of a concept definition and there are no cyclic dependencies between defined concepts. An ABox is a set of concept and role assertions. The semantics of interpretations is extended to concept definitions and assertions as shown in Table 1. We say that an interpretation \(I\) is a model of a TBox \(T\) (ABox \(A\)), if it satisfies all concept definition in \(T\) (assertions in \(A\)). \(I\) is a model of a knowledge base \(K = (T, A)\) if it is a model for both \(T\) and \(A\).

There exists a number of inferences for DLs. Three com-
the triangle inequality was found to be hard to achieve for similarity. However, all definitions and results can easily be assured, since they better capture our intuitive understanding of being.

Besides these standard reasoning tasks, other inferences have been developed for certain applications. The most specific concept, first introduced in [Nebel, 1990], is such a non-standard inference. This inference computes a concept description that describes an individual \( a \) from the knowledge base as exact as it is possible in the used DL.

**Definition 1.** Let \( \mathcal{L} \) be a DL and \( \mathcal{K} = (\mathcal{T}, \mathcal{A}) \) be an \( \mathcal{L} \)-KB. The concept description \( C \) is the most specific concept of an individual \( a \) w.r.t. \( \mathcal{K} \) (denoted \( \text{msc}(a) \)) iff

- \( a \) is an instance of \( C \), and
- for all concept descriptions \( D \in \mathcal{C}(\mathcal{L}) \), if \( a \) is an instance of \( D \), then \( C \sqsubseteq_T D \).

**Similarity measures.** For a DL \( \mathcal{L} \), a concept similarity measure \( \sim : \mathcal{C}(\mathcal{L}) \times \mathcal{C}(\mathcal{L}) \rightarrow [0, 1] \) is a function that assigns a similarity value \( C \sim D \) to each pair \( C, D \) of \( \mathcal{L} \)-concept descriptions. A value \( C \sim D = 0 \) means that \( C \) and \( D \) are totally dissimilar, while a value \( C \sim D = 1 \) means that \( C \) and \( D \) are totally similar.

A collection of properties for concept similarity measures is given in [Lehmann and Turhan, 2012]. In particular, a similarity measure \( \sim \) for \( \mathcal{L} \)-concept descriptions is:

1. **symmetric** iff \( C \sim D = D \sim C \) for all \( C, D \in \mathcal{C}(\mathcal{L}) \);
2. fulfilling the triangle inequality iff
   
   \[ 1 + D \sim E \geq D \sim C + C \sim E \]
   for all \( C, D, E \in \mathcal{C}(\mathcal{L}) \);
3. **equivalence invariant** iff for all \( C, D, E \in \mathcal{C}(\mathcal{L}) \) with \( C \equiv D \), it holds that \( C \sim E = D \sim E \);
4. **equivalence closed** iff \( C \sim D = 1 \iff C \equiv D \).

In this paper, we only consider symmetric similarity measures, since they better capture our intuitive understanding of similarity. However, all definitions and results can easily be extended to asymmetric similarity measures. Furthermore, the triangle inequality was found to be hard to achieve for similarity measures for even restricted DLs like \( \mathcal{EL} \), and thus will not be discussed here.

Observe that the property `equivalence closed` interacts with relaxed instances of a query concept \( C \) in the following way: clearly, if we want only relaxed instances with a similarity of exactly 1, then equivalence closed similarity measures should result in exactly the instances of \( C \), while similarity measures that are not equivalence closed might result in additional individuals.

Most previously proposed concept similarity measures can be divided into two groups: structural measures, which are defined using the syntax of the concepts, and interpretation based measures, which are defined using interpretations and cardinality instead of the syntax. We later describe a result for structural similarity measures, therefore we will describe these in more detail: Basically, a similarity measure \( \sim \) on \( \mathcal{L} \)-concepts descriptions is called structural, if it computes the similarity of two concepts \( C \) and \( D \) recursively by computing the similarity of concept names in \( C \) and \( D \) and the similarity of the existential restrictions occurring in \( C \) and \( D \) and combining these values monotonically to the overall similarity.

For structural similarity measures to be equivalence invariant, the concepts often need to be transformed into a normal form before comparing them [Lehmann and Turhan, 2012]. For a similarity measure \( \sim \), we call the normal form used for the computation of the similarity the \( \sim \)-normal form.

### 3 Relaxed Instances

In this section we introduce the main reasoning problems that we want to solve, as well as a first approach for obtaining a solution.

Our main goal is to generalize query answering to allow for more relaxed solutions. Intuitively, given a concept \( C \), we are interested in finding all the certain instances of \( C \), but also in finding those individuals that are close to being instances of \( C \); we call these individuals the relaxed instances of \( C \). To emphasize the contrast, we some times call the instances of \( C \) certain instances of \( C \).

Before we can try to compute these relaxed instances, we need to formalize the notion of relaxed instances of a query concept. In principle there are many ways to do so and we discuss next some of these options.

One natural approach would be to try to decide which individuals are similar to any of the certain instances of \( C \). However, this method would require the definition of a similarity measure on the elements of the domain, rather than on the concepts. Such a DL with a similarity measure on the domain elements was introduced in [Lutz et al., 2003]. However, for this DL the similarity measure (or more precisely, a distance metric) is part of the interpretation and cannot be adjusted to different user needs.

A different idea that has been proposed is to simply generalize the concept \( C \) by considering named concepts that subsume \( C \). Thus for a named concept \( C \), consider its direct subsumers in the concept hierarchy. This idea is easy to implement and understand, but provides only very rough approximations to the concept \( C \) determined by the set of concept names only. Moreover, users have no control on the quality of the approximation provided; in fact even the direct subsumers might describe a concept that is already very dissimilar to \( C \).

We follow a different approach, in which we ask for the instances of those concepts that are similar to \( C \). We can then control how inclusive the relaxed instance solutions should be, by adjusting the degree \( t \) of similarity allowed.

**Definition 2** (relaxed instance). Let \( \mathcal{L} \) be some DL, \( C \) be an \( \mathcal{L} \)-concept, \( \sim \) a similarity measure over \( \mathcal{L} \)-concepts, and \( t \in (0, 1] \). The individual \( a \in N_I \) is a relaxed instance of \( C \) w.r.t. the \( \mathcal{L} \)-knowledge base \( \mathcal{K} \), \( \sim \) and the threshold \( t \), denoted \( a \in \sim^t C \), iff there exists a concept description \( X \in \mathcal{C}(\mathcal{L}) \) such that \( C \sim X \geq t \) and \( a \in X^t \) for all models \( \mathcal{I} \) of \( \mathcal{K} \).
For brevity, we will denote as Relax\(\_\sim_t(C)\) the set of all relaxed instances of \(C\) w.r.t. \(K\), \(\sim\) and \(t\). Clearly, the elements of Relax\(\_\sim_t(C)\) depend strongly on the value of \(t\), but also on the similarity measure \(\sim\), as shown in Figure 1. For a fixed similarity measure \(\sim\), if \(t \leq t'\), then it holds that Relax\(\_\sim_t(C) \subseteq\) Relax\(\_\sim_{t'}(C)\). In the figure, the central circle is shown the interpretation of Relax\(\_\sim_t(C)\) with darker lines gradually representing large values \(t\). We use two different kinds of lines (continuous vs. dashed) to represent two different similarity measures, that relax the concepts based on different features. As can be seen, the sets obtained can greatly differ from each other.

As mentioned before, our goal is to find all the instances in Relax\(\_\sim_t(C)\). Following Definition 2, this task could be performed by first computing all concepts \(X\) that are similar to \(C\) with degree at least \(t\), and then obtaining all the instances of these concepts \(X\); in symbols,

\[
\text{Relax}_{\sim_t}(C) = \bigcup_{C \sim X \geq t} \{a \mid a \text{ is an instance of } X\}
\]

However, this approach suffers from two main drawbacks. First, the set of all concepts that are similar to \(C\) with degree at least \(t\) might be infinite, thus requiring an infinite number of queries to obtain Relax\(\_\sim_t(C)\), even though this set contains only finitely many individuals. Second, it is not known how to compute the similar concepts \(X\). Similarity measures tell us only how similar two given concepts are, but not how to build a concept that is similar to another with at least some given degree.

To avoid these issues, we consider a different reasoning problem, that considers the computation of a concept that has a given individual \(a\) as an instance and resembles \(C\) most. We call this the mimic of \(C\) w.r.t. \(a\).

**Definition 3 (mimic).** Let \(\mathcal{L}\) be a DL, \(K\) be an \(\mathcal{L}\)-knowledge base, \(a \in N_I\) be an individual name, \(C\) be an \(\mathcal{L}\)-concept description, and \(\sim\) be a similarity measure. An \(\mathcal{L}\)-concept \(D\) is called a mimic of \(C\) w.r.t. \(a\), denoted \(\text{M}(C, a)\), iff the following two conditions hold:

- \(a\) is an instance of \(D\), i.e., \(a^I \in D^I\) for all models \(I\) of \(K\), and
- for all \(\mathcal{L}\)-concept descriptions \(E\) holds, if \(a\) is an instance of \(E\), then \(C \sim D \geq C \sim E\).

Intuitively, a mimic of \(C\) w.r.t. \(a\) is a concept that is as similar to \(C\) as possible, while still having \(a\) as an instance. As for relaxed instances, the mimic strongly depends on the similarity measure chosen. Figure 2 depicts the idea of mimics. In the figure, \(a\) and \(b\) are two named individuals. The former is an instance of \(C\) while the latter is not. The dotted lines depict their most specific concepts. Since \(a\) is an instance of \(C\), \(C\) is also a mimic of \(C\) w.r.t. \(a\): \(C \sim C = 1\). The dashed line depicts a mimic of \(C\) w.r.t. \(b\). Notice that this mimic must contain the msc of \(b\), but need not be a subsumer of \(C\).

We must point out that the mimic of \(C\) w.r.t. an individual \(a\) need not be unique, even modulo concept equivalence. For example, let \(K\) be a knowledge base consisting of the empty TBox \(T\) and the ABox \(A = \{A \cap B(a)\}\), and \(\sim\) be a similarity measure with \(A \sim C = 0.5\), \(B \sim C = 0.5\) and \(A \cap B \sim C = \max\{A \sim C, B \sim C\} = 0.5\). Then \(A\), \(B\), and \(A \cap B\), are all mimics of \(C\) w.r.t. \(a\), as they all have a similarity value of 0.5 to \(C\). In fact, there can be infinitely many such mimics for a given concept \(C\) and individual \(a\). As we will see, it suffices to compute one of them.

Using mimics, we can compute the relaxed instances of a concept. The idea is to compute, for each individual \(a\) appearing in the knowledge base \(K\), the mimic of \(C\) w.r.t. \(a\). If this mimic has similarity at least \(t\) with \(C\), then \(a\) is a relaxed instance of \(C\); otherwise, it cannot be a relaxed instance, as no concept can have a greater similarity degree with \(C\) while still containing \(a\). This is formalized in the following proposition. The proof is a simple consequence of the arguments given above.

**Proposition 4.** Let \(K\) be a knowledge base, \(a\) be an individual occurring in \(K\), \(C\) be a concept description, \(\sim\) be a similarity measure and \(t \in [0, 1]\). Then \(a \in \text{Relax}_{\sim_t}(C)\) iff there is a mimic \(D\) of \(C\) w.r.t. individual \(a\) such that \(C \sim D \geq t\).

In the next section we will study the problem of computing a mimic for a given concept \(C\) w.r.t. an individual \(a\). Since all mimics must have the same degree of similarity w.r.t. \(C\), a simple similarity computation provides us with a decision whether \(a\) is a relaxed instance of \(C\) or not, up to degree \(t\). As computing a mimic may be an expensive task, we also provide an optimization criterion: if a mimic \(D\) of \(C\) w.r.t. \(a\) is similar to \(C\) to degree at least \(t\), then all instances of \(D\) must also be relaxed instances of \(C\), and hence there is no need of computing their corresponding mimics.
In general there are infinitely many concepts, for which an individual \( a \) is an instance of, and thus enumerating them and computing the similarity to \( C \) to find the mimic is not a feasible option. However, under some circumstances we can limit the number of concepts that need to be tested in order to find a mimic.

Recall that the notion of a mimic combines a property that is based on the semantics (it must have \( a \) as an instance) and a syntactic property (it must be similar to \( C \)). The semantic property gives us a starting point on how to find a mimic. A mimic \( D \) w.r.t. \( a \) must always have \( a \) as an instance, and hence, by definition of the msc, \( \text{msc}(a) \sqsubseteq_T D \) holds. For equivalence invariant similarity measures the idea is to use the msc\((a)\) as a lower bound for the mimic guaranteeing the semantic property, and to only consider concept descriptions that can be obtained from syntactic manipulations of msc\((a)\) that result in a generalized concept, i.e., by removing some concept names or existential restrictions.

**Definition 5** (generalized concept). Let \( C \) be a concept description of the form
\[
C = \prod_{i \in I} A_i \sqcap \prod_{j \in J} \exists r_j . E_j ,
\]
with \( A_i \in NC \) for all \( i \in I \), and \( r_j \in NR \), \( E_j \) is a concept description for all \( j \in J \). Then a concept description \( D \) is a generalized concept of \( C \) iff it has the form
\[
D = \prod_{i' \in I'} A_i' \sqcap \prod_{j' \in J'} \exists r_{j'} . E_{j'}'
\]
with \( I' \subseteq I \), \( J' \subseteq J \) and \( E_{j'}' \) is a generalized concept of \( E_j \) for \( j \in J' \).

This idea, however, only works if the msc is given in a particular syntactic form. It needs to be fully expanded.

**Definition 6** (fully expanded concept). Let \( T \) be an \( \text{EL} \)-TBox. A concept description \( C \) is fully expanded w.r.t. \( T \) iff for all concept definitions \( D = E \in T \) with \( C \sqsubseteq_T D \) we have that \( E \) is a generalized concept of \( C \).

The idea is that \( C \) contains all its subsumers explicitly as sub-concept descriptions. Now, we can show that the mimic of \( C \) w.r.t. \( a \) must be a generalized concept of the fully expanded most specific concept of \( a \).

**Lemma 7.** Let \( K = (T, A) \) be an \( \text{EL} \)-knowledge base, \( a \) be an individual from \( A \), \( C \) be an \( \text{EL} \)-concept description, and \( \sim \) be an equivalence invariant similarity measure. Let further \( E = \text{msc}(a) \) be the fully expanded most specific concept of \( a \). Then there is a mimic \( D = \text{msc}(C, a) \) of \( C \) w.r.t. \( K \) that is a generalized concept of \( E \).

**Proof.** We show that any concept \( F \) which has \( a \) as an instance must be equivalent to a generalized concept of the fully expanded msc. Since the mimic of \( C \) w.r.t. \( a \) has \( a \) as an instance and \( \sim \) is equivalence invariant, the lemma follows.

Let \( F \) be a concept description with \( a^+ \in F^2 \) for all models \( I \) of \( K \). Then \( E \sqsubseteq_K F \) by definition of the msc. Since \( E \) is fully expanded and contains all its subsumers explicitly, any part of the concept description \( F \) must also be part of the concept description \( E \). Thus \( F \) is a generalized concept of \( E \).

In general, the msc may contain a chain of infinitely nested existential restrictions for cyclic ABoxes, and hence describing it as a concept would require infinite size. Then there are still infinitely many generalized concepts (of finite size) that need to be checked to find a mimic. This means that Lemma 7 does not always provide a solution to the problem. However, the query concept \( C \) (in \( \sim \)-normal form) has always a finite role-depth, and most structural similarity measures used in practice compute the similarity recursively between concepts at the same role-depth. Therefore, for these similarity measures, it is possible to limit the role-depth of the most specific concept and still get the same result.

**Definition 8.** Let \( K \) be an \( \text{EL} \)-KB. By \( \text{rd}(C) \) we denote the role-depth of a concept \( C \), i.e. the maximal number of nested quantifiers.

The \( \text{EL} \)-concept description \( C \) is the role-depth bounded most specific concept (denoted \( k \)-msc\((a)\)) of an individual \( a \) w.r.t. \( K \) and the role-depth bound \( k \) iff

- \( \text{rd}(C) \leq k \),
- \( a^+ \in C^2 \) for all models \( I \) of \( K \),
- for all \( \text{EL} \)-concepts \( D \in C(\mathcal{L}) \) with \( \text{rd}(D) \leq k \) and all \( a^+ \in D^2 \) for all models \( I \) of \( K \) it holds that \( C \sqsubseteq_T D \).

The role-depth bounded msc is a commonly used approximation of the msc, since it always exists and is unique. An algorithm to compute the \( k \)-msc in the \( \text{EL} \)-family, even w.r.t. general TBoxes, has been introduced in [Peñaloza and Turhan, 2011] and [Ecke et al., 2013]. Using this, we can now show that for structural similarity measures we can find the mimic always as a generalized concept of the role-depth bounded msc.

**Lemma 9.** Let \( K = (T, A) \) be an \( \text{EL} \)-knowledge base, \( a \) be an individual from \( A \), \( C \) be an \( \text{EL} \)-concept description in \( \sim \)-normal form, and \( \sim \) be a structural, equivalence invariant similarity measure with the following property:
\[
X \sim \prod_{i \in I} A_i \geq X \sqcap \exists r . B \sim \prod_{j \in J} A_i.
\]
Let further \( k = \text{rd}(C) \) and \( E = \text{msc}(a) \) be the fully expanded role-depth bounded most specific concept of \( a \). Then there is a mimic \( D = \text{msc}(C, a) \) of \( C \) w.r.t. \( a \) that is a generalized concept of \( E \).

**Proof.** By Lemma 7 we know that there exists a mimic \( F \) of \( C \) w.r.t. \( a \) that is a generalized concept of the (possibly infinite) msc\((a)\). Since \( E \) is the fully expanded \( k \)-msc of \( a \), \( F \) must also be a generalized concept of \( E \) up to role-depth \( k \) (but of course, it may contain additional existential restrictions which increase the role-depth of \( F \)). We show by induction on \( k \), that there is a generalized concept \( F' \) of \( E \) with \( F' \sim C \geq F \sim C \). This will imply that \( F' \) is a mimic of \( C \) w.r.t. \( a \), which proves the lemma.

For the case \( k = 0 \), \( C = \prod_{i \in I} A_i \) and \( E = \prod_{j \in J} B_j \) are conjunctions of concept names and since \( F \) is a generalized concept of \( E \) up to role-depth \( k = 0 \), we know that \( F \) is of the form \( F = \prod_{i \in I} B_j \sqcap \prod_{h \in H} \exists r_h . F_h \) with \( J' \subseteq J \). But then property (1) yields for \( F' = \prod_{j \in J} B_j \):
\[
F' \sim C \geq F' \sqcap \prod_{h \in H} \exists r_h . F_h \sim C = F \sim C.
\]
Figure 3: Computation algorithm for relaxed instances in $\mathcal{EL}$.

For the case $k > 0$, $C = \prod_{i \in I} A_i \sqcap \prod_{h \in H} \exists h. C_h$ and $E = \prod_{j \in J} B_j \sqcap \prod_{l \in L} \exists l. E_l$ are conjunctions of concept names and existential restrictions with $rd(C_h), rd(E_l) \leq k-1$ for $h \in H$, $l \in L$. Once again, since $F$ is a generalized concept of $E$ up to role-depth $k$, it must be of the form $F = \prod_{j' \in J'} B_{j'} \sqcap \prod_{l' \in L'} \exists l'. F_l'$ with $J' \subseteq J$, $L' \subseteq L$ and each $F_l'$ is a generalized concept of $E_l$ up to role-depth $k-1$. But then, the induction hypothesis yields for each $h \in H$ and $l \in L'$ that $F_l' \sim C_h \geq F_l \sim C_h$ for generalized concepts $F_l'$ of $E_l$. Then also $F' = \prod_{j' \in J'} B_{j'} \sqcap \prod_{l' \in L'} \exists l'. F_l'$ is a generalized concept of $E$ and since the similarity measure $\sim$ is structural, this yields: $F' \sim C \geq F \sim C$.

We have now identified some constraints on the similarity measure such that we can always find the mimic of $C$ w.r.t. $a$ from a finite set of concept descriptions: the generalized concepts of the fully expanded role-depth bounded msc of the individual $a$.

Instead of computing the mimic $D = \exists \mathbb{H}(C, a)$ of $C$ w.r.t. $a$ and testing whether the similarity between the $C$ and $D$ is at least $t$, it is enough to find any concept $D'$ with $a$ as an instance and $C \sim D' \geq t$ to show that $a$ is a relaxed instance of $C$. Such an non-deterministic algorithm that, given an $\mathcal{EL}$-knowledge base $\mathcal{K}$, an individual $a$, an $\mathcal{EL}$-concept description $C$, a similarity measure $\sim$, and a similarity degree $t$, computes whether $a$ is a relaxed instance of $C$ w.r.t. $t$ and given in Figure 3. The algorithm works by computing the $k$-msc of $a$ with $k = rd(C)$ and then guessing a generalized concept $F$ of $E$ with similarity $F \sim C \geq t$, if such a concept exists.

**Corollary 10.** Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{EL}$-knowledge base, $C$ be an $\mathcal{EL}$ concept in $\approx$-normal form, $a$ be an individual in $\mathcal{K}$, $\approx$ be a structural equivalence invariant similarity measure fulfilling Property 1 from Lemma 9 and $t \in [0, 1]$. Then relaxed-instance?(a, C, K, \sim , t) computes whether $a \in\mathcal{K}_t$ w.r.t. $K$.

**Proof.** Lemma 9 shows that a mimic of $C$ w.r.t. $a$ is a generalized concept of $E = k$-msc$(a)$ for $k = rd(C)$. Thus, if the algorithm returns false, we know that no generalized concept $F$ exists with $C \sim F \geq t$, and in particular also the mimic of $C$ w.r.t. $a$ must have a similarity of less than $t$ to $C$. Thus no concept that has $a$ as an instance is similar enough to $C$ and thus $a \notin\mathcal{K}_t$. If the algorithm returns true, the guessed concept $F$ shows $a \in\mathcal{K}_t$ since $a$ is an instance of $F$ and $F \sim C \geq t$.

Guessing a generalized concept $F$ of a concept description $E$ can be done in time linear to size $\|E\|$ of $E$ by recursively guessing for each concept name and each existential restriction in $E$ whether they should occur in $F$ or not. However, the size of $E = k$-msc$(a)$ can be exponential in $k$ and polynomial in $\|K\|$ [Peñaloza and Turhan, 2011]. Since $k = rd(C)$ is bounded linearly by $\|K\|$, the algorithm runs in NEXP-time (provided that $\sim$ can be computed in NEXP-time). However, the algorithm runs in NP-time in $\|K\|$ (provided that $\sim$ can be computed in NP), and since $C$ is an input concept, its role-depth can be assumed to be rather low. Hence, we conjecture that the exponential blow-up of the msc usually plays only a minor role in practical applications.

To obtain a deterministic algorithm, the mimic of $C$ w.r.t. $a$ can be computed by enumerating all generalized concepts of $k$-msc$(a)$ and taking one with the maximal similarity to $C$. Of course, there are a few optimizations possible: if the individual $a$ belongs to $C$, we can directly return true, since the mimic will always be $C$ itself. If we find a generalized concept $F$ with $C \sim F \geq t$, we can stop to search for even more similar concepts and return true. And finally, if we find a mimic $D$ for an individual $a$ with $C \sim D \geq t$, we know that all other instances of $D$ besides $a$ will be relaxed instances of $C$ as well, without needing to compute their mimics.

## 5 Conclusions

In this paper we have studied a new inference service for description logics, which consists in computing the relaxed instances of a given query concept $C$ w.r.t. a similarity measure $\sim$ and a similarity degree $t$. This problem is relevant to the field of artificial intelligence in general, and to knowledge representation and reasoning in particular, as it provides a formal and unambiguous method for computing answers for a relaxed notion of instance query. Thus it is useful for ontology-based applications that need to obtain answers that fit the query criteria only to a certain degree.

The inference has two main degrees of freedom: in the choice of the similarity measure, and in the degree of relaxation of the concept. The similarity degree $t$ allows the user to tune how strict or relaxed the answers provided are: a degree closer to 1 will yield only a few additional individuals that do not belong to $C$, while relaxing to a level closer to 0 yields almost all individuals in the ontology as relaxed instances. The similarity measure provides also criteria on how the relaxed instances are obtained. Intuitively, different similarity measures yield different weights on specific criteria. For example, one could require that small changes inside existential restrictions produce a high level of dissimilarity.

As a step for computing the relaxed instances of a concept $C$, we introduced the problem of finding a mimic of the query concept $C$ w.r.t. a given individual $a$. Such a mimic is a concept $D$ that contains $a$ as instance, and has the highest similarity possible to $C$; i.e., it is a concept that tries to imitate $C$ while containing $a$. Computing mimics w.r.t. all individuals appearing in an ontology provides a method for finding the relaxed instances of $C$.
The problem of finding a mimic is non-trivial. We have provided an algorithm capable of finding such a mimic, based on the msc of an individual \( a \) for certain structural similarity measures. While this computation is expensive, some obvious optimizations can be used to reduce the number of times these mimics are constructed.

As future work, we plan to expand on the two main inference problems described in this paper. First, we intend to improve the algorithms that compute the mimics. On the one hand, we will try to find one such mimic efficiently. On the other, it would also be beneficial to compute the most general mimic, if it exists; this concept would have the most possible instances, and hence would be useful as an optimization approach. Second, we will try to find tight complexity bounds on the problems of computing relaxed instances and finding mimics for a given concept. Third, we plan to obtain a better understanding on the properties of similarity measures that can impact (positively or negatively) on the complexity and run-time of solving these problems. As we have mentioned before, both inferences depend strongly on the similarity measure chosen. However, we do not know precisely which measures would allow for better results, be it in terms of execution time, or in terms of precision and fine-grained tuning.

References


A Bipolar Assertion Model for Natural Language Generation

Henrietta Eyre and Jonathan Lawry
University of Bristol
UK
henrietta.eyre@bristol.ac.uk

Abstract
A fundamental problem in natural language generation and dialogue systems is that of identifying the most appropriate assertion in a given context. This is usually viewed as a decision problem in which a single statement must be selected from amongst a set of competing possibilities. Furthermore, natural language is also inherently vague, allowing for assertions with differing levels of semantic precision. Indeed, vagueness is a multi-faceted phenomenon incorporating typicality, semantic uncertainty and indeterminism. In this paper we propose a decision model for selecting assertions to described elements drawn from an underlying conceptual space. This model is developed within a concept representation framework which combines random set theory and prototype theory, and which also incorporates bipolarity (indeterminism).

1 Introduction

1.1 Natural Language Generation as a Choice Problem
Natural Language Generation (NLG) systems take inputs in the form of non-linguistic data, and map them onto a linguistic output. For example in [Goldberg et al., 1994] weather data is mapped to a weather report, and in [Portet et al., 2007] data from patient records is mapped onto a medical (textual) summary. A principal feature of any NLG system is the process of identifying the most appropriate assertion to provide in a given context. In this context, NLG systems are moving away from fully scripted systems, and identifying the most appropriate assertion is now often viewed as a decision problem in which a single statement must be selected from a set of possible statements [Van Deemter, 2009].

1.2 Vagueness in Natural Language
In selecting an appropriate statement, an NLG system must take input data which may be quantitative (for example an air temperature of $25^\circ C$) and produce a natural language output (for example “it is warm today”). In doing so, care must be taken to account for the presence of vague predicates in natural language. If we consider the example of air temperature, then we may expect most people to agree with a statement of “it is hot today” given a temperature of $45^\circ C$. However if the air temperature was $15^\circ C$ we may not expect a uniform agreement across a population. Furthermore it is not obvious that we can identify a clear boundary between two predicates (such as cold and warm).

Vagueness is common in natural language [Russell, 1923]. We continually encounter situations where a lexical structure does not determine a unique interpretation, but rather leaves open an interval or collection of possible meanings. Intuition would suggest that such a phenomenon is not without purpose, and humans have not evolved to communicate using vague predicates without them providing some form of communicative advantage [Lipman, 2009]. Van Deemter has proposed a number of ways in which vagueness could be a positive feature of natural language [Van Deemter, 2009]. Of particular relevance to the current context is the suggestion that risk may be minimised by using vague assertions in the presence of conflicting aims and objectives. For example, he argues that making crisp promises or forecasts could be a risky strategy for politicians, possibly resulting in an electoral punishment. Similarly, in weather forecasting systems crisp forecasts are more likely to be clearly wrong, resulting in a loss of confidence in the system.

1.3 Indeterminism and Semantic Uncertainty in NLG
The methods proposed in this paper focus on two particular aspects of vagueness: indeterminism and semantic uncertainty. It is important that we make a clear distinction between these terms, and indeed clarify how each is interpreted.

For vague categories a sharp boundary cannot be identified between those cases which definitely belong to the category and those which do not. This feature has inspired different theories of vagueness such as supervaluationism [Fine, 1975] and fuzzy logic [Zadeh, 1996]. The epistemic view of vagueness [Williamson, 1994] assumes that objectively correct sharp category boundaries do does exist, but argues that in practice it is impossible to know where these boundaries lie. A slightly weaker form of this argument does leave agents with the possibility of flexible categories. Rather than claiming the existence of some objective boundary between
a concept and its negation we instead envisage a distributed model of language learning in which a shared meaning of terms evolves through repeated interactions between individuals in a population of communicating agents [Lawry and Tang, 2009]. In this context then when facing a decision problem about what to assert, or with a category learning goal, an agent may find it useful to assume that sharp category boundaries do exist. In other words, agents behave as if the epistemic view of vagueness is correct. This strategy is referred to as the *epistemic stance* [Lawry, 2008b; Lawry and Tang, 2009]. We refer to the uncertainty regarding category boundaries resulting from the distributed manner in which concept definitions are learnt as *semantic uncertainty*. Vague concepts inherently admit borderline cases where we would not readily classify either as definitely belonging to or definitely not belonging to a particular concept. *Indeterminism* refers to the existence of an explicit borderline region between the extension of the concept and that of its negation. Parikh argues that indeterminism can be observed in the assertability of expressions [Parikh, 2008]. In deciding what to assert, some expressions would be considered definitely true and some definitely not true, but there are also those which we would not consider to be neither, or borderline true. Indeed if somebody else were to use such a borderline sentence we would not not “reproach” them.

Authors such as Dorothy Edgington have argued that to deal with the many issues which arise when modelling vagueness we must consider probability theory in conjunction with the more traditional logical models [Edgington, 1992; 2001]. In the following we present a conceptual model which combines random set and prototype theory where categories are modelled as random sets within a conceptual space. Each category has two boundaries modelled as probabilistic distances from a prototype, providing an explicit representation of both semantic uncertainty and indeterminism. Section 3 builds on the theory of label semantics [Lawry and Tang, 2009; Tang and Lawry, 2012] to introduce a model of assertion, where agents decide between strong or weak versions of a concept and its negation we instead envisage a distributed model of language learning in which a shared meaning of terms evolves through repeated interactions between individuals in a population of communicating agents [Lawry and Tang, 2009]. In this context then when facing a decision problem about what to assert, or with a category learning goal, an agent may find it useful to assume that sharp category boundaries do exist. In other words, agents behave as if the epistemic view of vagueness is correct. This strategy is referred to as the *epistemic stance* [Lawry, 2008b; Lawry and Tang, 2009]. We refer to the uncertainty regarding category boundaries resulting from the distributed manner in which concept definitions are learnt as *semantic uncertainty*. Vague concepts inherently admit borderline cases where we would not readily classify either as definitely belonging to or definitely not belonging to a particular concept. *Indeterminism* refers to the existence of an explicit borderline region between the extension of the concept and that of its negation. Parikh argues that indeterminism can be observed in the assertability of expressions [Parikh, 2008]. In deciding what to assert, some expressions would be considered definitely true and some definitely not true, but there are also those which we would not consider to be neither, or borderline true. Indeed if somebody else were to use such a borderline sentence we would not not “reproach” them.

This naturally results in a typical ordering on possible examples of a category.

We define a set of n labels \( LA = \{L_1, ..., L_n\} \), from which a set of compound expressions \( LE \) can be generated through recursive applications of the logical connectives \( \land, \lor, \neg \) to the labels in \( LA \). Each label \( L_i \) represents a word which may be used to described elements or collections of elements in the conceptual space \( \Omega \). As such, basic labels \( L_i \) correspond to descriptive words (adjectives or nouns) for which the expression “\( x \) is \( L_i \)” is meaningful for any \( x \in \Omega \). For example, if we let \( \Omega \) be a colour space such as CIELAB [Connolly and Fleiss, 1997], then \( LA \) would consist of the basic colour labels red, blue etc, and \( LE \) would be the compound expressions generated from these such as red and blue, not green, orange and not black etc.

**Definition 1 (Label Expressions).** Given a finite set of basic labels \( LA \) we define the set of label expressions \( LE \) recursively as follows:

- If \( L_i \in LA \) then \( L_i \in LE \).
- If \( \theta, \phi \in LE \), then \( \neg \theta \in LE, \theta \lor \phi \in LE, \theta \land \phi \in LE \)

We assume a distance function \( d \) defined on \( \Omega \) so that \( d : \Omega^2 \to [0, \infty) \) which quantifies conceptual distance (or similarity). A label \( L_i \) is appropriate to describe element \( x \in \Omega \) if \( x \) is sufficiently similar or close to a prototypical element \( P_i \) for \( L_i \). As such we identify thresholds for the distance \( d(x, P_i) \) under which the label \( L_i \) may be applied. More formally, for each \( L_i \in LA \), we identify a pair of thresholds \( \epsilon_i \leq \tau_i \) such that \( L_i \) is absolutely appropriate to describe \( x \) if \( d(x, P_i) \leq \epsilon_i \), and \( L_i \) is absolutely inappropriate to describe \( x \) if \( d(x, P_i) > \tau_i \). As such an agent may then identify a borderline region for each concept where objects from the conceptual space are neither absolutely appropriately or absolutely inappropriately described by the concept label. For a label \( L_i \), the borderline region for the category for \( L_i \) is described by the set \( \{ x \in \Omega : \epsilon_i < d(x, P_i) \leq \tau_i \} \). Now for each \( i = 1 \ldots n \) the semantic uncertainty associated with the definition of \( L_i \) is quantified by a joint probability density function \( \delta_i(\epsilon_i, \tau_i) \) on \( \mathbb{R}^2 \). This gives agents an uncertain measure on category boundaries.

Each agent has a cognitive model of the \( n \) categories where each category (with category label \( L_i \)) is defined by a prototype \( P_i \in \Omega \), a lower threshold \( \epsilon_i \), and an upper threshold \( \tau_i \). We formally say that each agent is characterised by an interpretation \( I \), defined as follows:

**Definition 2 (Interpretation).** An interpretation of a label set \( LA \) is a tuple \( I = (\Omega, d, \vec{P}, \vec{\epsilon}, \vec{\tau}) \), where \( \Omega \) is a conceptual space with associated metric \( d \), \( \vec{P} = \langle P_1, \ldots, P_n \rangle \) is a vector of prototypes where each \( P_i \in \vec{P} \) defines the prototype (or prototypes) for label \( L_i \), and \( \vec{\epsilon} = ((\epsilon_1, \tau_1), \ldots, (\epsilon_n, \tau_n)) \) is a vector of upper and lower boundary variables. Also, each boundary variable pair \((\epsilon_i, \tau_i)\) has an associated joint probability density \( \delta_i \).

\(^1\) \( P_i \) may be a point or collection of points in \( \Omega \)
Given an interpretation \( I \), the constraints \( d(x, P_i) \leq \xi_i \) and \( d(x, P_i) \leq \tau_i \) can be used to define lower and upper neighbourhood regions of \( \Omega \) within which all points can be absolutely appropriate or not absolutely inappropriate to be described by \( L_i \). We may think of these regions as extensions of \( L_i \) for interpretation \( I \). For single labels extensions are a nested pair of neighbourhoods about a prototype \( P_i \) with boundary distances \( \xi_i \) and \( \tau_i \) respectively (see figure 1). This can then be recursively extended from basic labels to label expressions in \( LE \). These regions, or neighbourhoods, are defined as follows:

**Definition 3** (Neighbourhoods for label expressions). The lower and upper neighbourhoods for label expressions in \( LE \) are defined recursively as follows:

- \( \mathcal{N}_{\theta}^I = \left\{ x \in \Omega : d(x, P_i) \leq \xi_i \right\} \)
- \( \mathcal{N}_{\phi}^I = \left\{ x \in \Omega : d(x, P_i) \leq \tau_i \right\} \)
- \( \mathcal{N}_{\theta}^{\phi} = \mathcal{N}_{\theta}^I \cap \mathcal{N}_{\phi}^I \)
- \( \mathcal{N}_{\phi}^{\theta} = \mathcal{N}_{\theta}^I \cup \mathcal{N}_{\phi}^I \)
- \( \mathcal{N}_{\theta \phi}^I = \mathcal{N}_{\theta}^I \times \mathcal{N}_{\phi}^I \)
- \( \mathcal{N}_{\phi \theta}^I = \mathcal{N}_{\phi}^I \times \mathcal{N}_{\theta}^I \)

where \( I \) is an interpretation of \( LA \).

According to definition 3 we may now say that for an any expression \( \theta \in LE \), and any \( x \in \Omega \), \( \theta \) is absolutely appropriate to describe \( x \) under interpretation \( I \) if \( x \in \mathcal{N}_{\theta}^I \), and \( \theta \) is not absolutely inappropriate to describe object \( x \) under interpretation \( I \) if only if \( x \in \mathcal{N}_{\phi}^I \). The definition of the neighbourhood for the negation of an expression is built on the intuition that \( \neg \theta \) is absolutely appropriate to describe \( x \) if and only if \( \theta \) is absolutely inappropriate to describe \( x \).

Given a particular expression \( \theta \in LE \), definition 3 identifies the set of objects \( x \in \Omega \) which are appropriately described by \( \theta \). We may also take a different perspective and, for any object \( x \in \Omega \), consider which labels from \( LA \) are appropriate to describe \( x \). We define \( \mathcal{D}_x \) and \( \mathcal{D}_x \) to be the subset of labels in \( LA \) which are absolutely appropriate and not absolutely inappropriate to describe a particular instance of the conceptual space \( x \in \Omega \) respectively. Formally:

**Definition 4.** \( \forall x \in \Omega, \mathcal{D}_x^I = \{ L_i : d(x, P_i) \leq \xi_i \}, \mathcal{D}_x^I = \{ L_i : d(x, P_i) \leq \tau_i \} \)

Now under our definition of an interpretation (definition 2) the boundaries \( (\xi_i, \tau_i) \) are random variables given by the joint density \( \delta_i \). This results in the sets \( \mathcal{N}_{\theta}^I, \mathcal{N}_{\phi}^I, \mathcal{D}_x^I \) and \( \mathcal{D}_x^I \) being uncertain sets, or random sets [Matheron, 1975]. As such, we do not classify set members with binary (Tarski) truth values, but rather set members have membership with associated probabilities. And so we define a mass function \( m_x(\mathcal{E}, \mathcal{F}) \), the probability that, for a particular \( x \in \Omega \), the set of labels which are absolutely appropriate to describe \( x \) is given by \( \mathcal{E} \), and the set of labels which are not absolutely inappropriate is given by \( \mathcal{F} \).

**Definition 5** (Mass function). Let \( \delta(x) \) be a joint density on the vector of pairs of thresholds with marginals \( \delta_i((\xi_i, \tau_i)) \). For an interpretation \( I \), for \( \mathcal{E} \subseteq LA, \mathcal{F} \subseteq LA \), the mass function \( m_x(\mathcal{E}, \mathcal{F}) = 2^{LA} \times 2^{LA} \to [0, 1] \) is defined as follows:

\[
m_x(\mathcal{E}, \mathcal{F}) = \delta((x) : \mathcal{D}_x^I = \mathcal{E}, \mathcal{D}_x^I = \mathcal{F})
\]

An agent’s subjective belief that an expression \( \theta \in LE \) can be used to describe an object \( x \in \Omega \) is given in terms lower and upper appropriateness measures. These correspond to the probabilities that the lower and upper distance thresholds satisfy \( x \in \mathcal{N}_{\theta}^I \) and \( x \in \mathcal{N}_{\phi}^I \) respectively.

**Definition 6** (Lower and upper appropriateness measures). \( \forall \theta \in LE, \forall x \in \Omega \) the lower and upper appropriateness measures are defined as follows:

- \( \mu_\theta(x) = \delta(x) : x \in \mathcal{N}_{\theta}^I \)
- \( \mu_\phi(x) = \delta(x) : x \in \mathcal{N}_{\phi}^I \)

**Example 2.1.** Consider a language \( LA = \{ L_1, L_2 \} \), and let \( \mathcal{E}_\theta = ((\xi_1, \tau_1), ..., (\xi_n, \tau_n)) \) be the vector of lower and upper boundary thresholds with joint density \( \delta \). First assume that \( (\xi_1, \tau_1) \) is independent of \( (\xi_2, \tau_2) \) so that \( \delta(x) = \delta_1(\xi_1, \tau_1) \times \delta_2(\xi_2, \tau_2) \). Now consider an expression \( \theta = L_1 \land L_2 \), so that we have

\[
\mu_\theta(x) = \delta((x) : d(x, P_1) \leq \xi_1, d(x, P_2) \leq \xi_2)
\]

\[
= \delta_1((\xi_1, \tau_1) : d(x, P_1) \leq \xi_1) \times \delta_2((\xi_2, \tau_2) : d(x, P_2) \leq \xi_2)
\]

\[
= \mu_{L_1}(x) \times \mu_{L_2}(x)
\]

Similarly, it also holds that \( \mu_\phi(x) = \mu_{L_1}(x) \times \mu_{L_2}(x) \).

Alternatively suppose that we have a strongly dependent boundary model, where \( \xi_1 = k_1 \xi_2 \) and \( \tau_1 = k_1 \tau_2 \) for \( k_1, k_2 \in \mathbb{R}^+ \) and \( k_1 \leq k_2 \). In other words, we assume that
each boundary variable $\xi_i$ is a re-scaling of an underlying variable $\xi$. (Similarly for $\tau_i$.) In this case:
\[
\underline{\mu}(\theta) = \delta((\xi, \tau) : d(x, P_1) \leq k_1 \xi, d(x, P_2) \leq k_2 \xi) = \delta((\xi, \tau) : \xi \geq \max \left(\frac{d(x, P_1)}{k_1}, \frac{d(x, P_2)}{k_2}\right)) = \min(\delta((\xi, \tau) : \xi \geq \frac{d(x, P_1)}{k_1}), \delta((\xi, \tau) : \xi \geq \frac{d(x, P_2)}{k_2})).
\]

Similarly, it also holds that $\overline{\mu}(x) = \min(\overline{\mu}_{L_1}(x), \overline{\mu}_{L_2}(x))$.

As an alternative to definition 6 the lower and upper appropriateness measures may be explicitly calculated as sums of the mass function $m_x$ over certain sets $\lambda(\theta)$ for $\theta \in LE$. Assume we have a set of prototypes $P = \{P_1, ..., P_n\}$, where each $P_i$ is a single element of $\Omega$. Furthermore, let $\xi_i = \xi$ and $\tau_i = \tau, \forall i = 1, ..., n$. In other words, the lower and upper boundaries are defined by the same variables for each label $L_i \in LA$. This is a special case of the strong dependency assumption outlined in example 2.1. Consider a permutation of $\Omega, (L_1, ..., L_n)$ such that $d(x, P_1) \leq ... \leq d(x, P_n)$. We may assume w.l.o.g. that this permutation is given by $\{L_1, ..., L_n\}$, so that $d(x, P_i) \leq ... \leq d(x, P_n)$. Furthermore let $F_i = \{L_1, ..., L_i\}$ for $i = 1, ..., n$, and $F_0 = \emptyset$. Then we obtain the following result [Tang and Lawry, 2012]:

**Theorem 2.** For $x \in \Omega$, by assuming w.l.o.g. that $d(x, P_1) \leq ... \leq d(x, P_n)$, let $F_i = \{L_1, ..., L_i\}$ and $F_0 = \emptyset$. Then for any $i$ and $j$ such that $0 \leq i \leq j \leq n$ the following holds:
\[
m_x(F_i, F_j) = \left\{ \begin{array}{ll}
\int_{d(x, P_1)}^{d(x, P_i)} \int_{d(x, P_j)}^{d(x, P_{i+1})} \delta((\xi, \tau))d\xi d\tau & \text{if } d(x, P_j) > d(x, P_i) \\
\int_{d(x, P_i)}^{d(x, P_{i+1})} \int_{d(x, P_j)}^{d(x, P_{i+1})} \delta((\xi, \tau))d\xi d\tau & \text{if } d(x, P_j) = d(x, P_i)
\end{array} \right.
\]

where for notational convenience we take $d(x, P_0) = 0$, and $d(x, P_{n+1}) = \infty$.

We now give an extension of theorem 2 to the general strong dependency case outlined in example 2.1.

More formally, let the category boundary variables follow the strongly dependent model introduced in example 2.1. Let $\xi_i = k_i \xi$ and $\tau_i = k_i \tau$, for $k_i, \overline{k_i} \in \mathbb{R}^+$. Furthermore, assume that $\overline{k_i} = \eta k_i$, where $\eta \in (1, 2]$ for each $i = 1, ..., n$.

We now define two re-orderings of $\Omega$. Consider an ordering $(L_1, ..., L_n)$ such that $\frac{d(x, P_1)}{k_1} \leq ... \leq \frac{d(x, P_n)}{k_n}$. Furthermore, a second ordering $(L'_1, ..., L'_n)$ such that $\frac{d(x, P'_1)}{k_1} \leq ... \leq \frac{d(x, P'_n)}{k_n}$, and then $\frac{d(x, P'_l)}{k_1} \leq ... \leq \frac{d(x, P'_n)}{k_n}$.

Now we can write expressions for the lower and upper appropriateness measures in terms of the mass function as follows [Tang and Lawry, 2012]:
\[
\underline{\mu}(x) = \sum_{(S, T) \in \lambda(\theta)} m_x(S, T) \quad \overline{\mu}(x) = \sum_{(S, T) \in \lambda(\theta)} m_x(T, S)
\]

In the following we introduce an assertion model. For practical implementation of this model it is useful to have an explicit formula to calculate the mass function $m_x$. This can be a non-trivial task, but under some assumptions we may give formulae which allow for such explicit calculation.
where, again, for notational convenience we take \( d(x, P_0) = 0 \), and \( d(x, P_{n+1}) = \infty \).

**Example 2.3.** Consider a conceptual space \( \Omega = [0, 1] \).
Let \( LA = \{ L_1, L_2 \} \), where \( L_1 \) has a single prototype \( P_1 = 0.4 \), and \( L_2 \) has prototype \( P_2 = 0.8 \). Let the underlying boundaries \((\varepsilon, \tau)\) be given by \( \delta \), the two dimensional uniform distribution on the region defined by the inequalities \( 0 \leq \varepsilon \leq 1 \) and \( \varepsilon \leq \tau \leq 1 + \varepsilon \). Let \( k_1 = 0.5, k_1 = 0.75, k_2 = 0.7, k_2 = 1.05 \), (so \( \eta = 1.5 \)).

Suppose we are considering an object \( x = 0.6 \), so \( d(x, P_1) = d(x, P_2) = 0.2 \). Then we have

\[
\begin{align*}
\frac{d(x, P_2)}{k_2} &= 0.287 < 0.4 = \frac{d(x, P_1)}{k_1} \\
\frac{d(x, P_2)}{k_2} &= 0.1905 < 0.2667 = \frac{d(x, P_1)}{k_1}
\end{align*}
\]

which gives our re-ordering of \( LA \) as \( (L_2, L_1) \).
Now we know that to search for non-zero values for the mass function we must consider only \( (\emptyset, \emptyset), (\emptyset, \{L_2\}), (\emptyset, \{L_2, L_1\}), (\{L_2\}, \{L_2, L_1\}) \).
In this way we can calculate the mass function for \((\emptyset, \emptyset)\) as follows:

\[
m_x(\emptyset, \emptyset) = \frac{d(x, P_2)}{k_2} \int_0^{\frac{\varepsilon}{k_1}} d\varepsilon \int_0^{\frac{\tau}{k_2}} d\tau
\]

Now we are considering an integral on the region \( R \) given by the area of the subregion of \( R \) bounded above by the lines \( \varepsilon = 0.2 \) and \( \tau = 0.2 \). This is in fact the area of an equilateral triangle with side length 0.1905. The mass function may be similarly calculated for all subsets of \( LA \) (indeed we only need to pay attention to \( m_x(S, T) \)) where \( S \subseteq T \). Figure 2 shows the domain of \( \delta \) divided into subregions sufficient to fully calculate the mass function. Figure 2 shows the values of a full calculation of the mass function.

### 3 An Assertion Model

In this section we propose a model where agents may give two different types of assertion: "\( x \) is \( \theta \)" denoting "\( x \) is strongly \( \theta \)" or "\( x \) is \( \bar{\theta} \)" denoting "\( x \) is weakly \( \theta \)". An assertion of "\( x \) is \( \theta \)" for an agent with interpretation \( I \) is taken to mean \( x \in N^I_\theta \). Similarly, "\( x \) is \( \bar{\theta} \)" means that \( x \in N^I_{\bar{\theta}} \).

Our framework gives agents a potentially infinite pool of possible assertions generated recursively from \( LA \) using the connectives \( \land, \lor, \bar{\lor} \) (i.e. the strong and weak assertions

<table>
<thead>
<tr>
<th>( \emptyset )</th>
<th>{( L_2 )}</th>
<th>{( L_2, L_1 )}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0181</td>
<td>0.0174</td>
<td>0.2502</td>
</tr>
<tr>
<td>{( L_2 )}</td>
<td>×</td>
<td>0</td>
</tr>
<tr>
<td>{( L_2, L_1 )}</td>
<td>×</td>
<td>0.1143</td>
</tr>
<tr>
<td>{( L_2, L_1 )}</td>
<td>×</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Figure 2: The value of the mass function \( m_x \) for example 2.3

of any \( x \) in \( LE \), but in practise only a small subset of these would ever be asserted. For example, it is unlikely that the expression \( \neg L \) would be asserted over \( L \) although the two are logically equivalent (adhering to Grice’s maxim of quantity [Grice, 1975]).

In addition, an agent may view certain types of expressions as being a priori more advantageous to assert than others. For example, in a situation where one had a shelf of 2 books, one clearly red and one clearly orange, we would expect to describe the colour of the red book as “red” rather than “not orange”, although the appropriateness measures may be high for both expressions. On the other hand in a situation where there is a book shelf with many books, all of many different shades of red and orange, the negated expression “not red” may be useful in identifying the correct book within a search. In view of these restrictions we allow a finite set of permitted assertions \( AS \subseteq \{ \emptyset, \emptyset : \emptyset \in LE \} \).

We then define a prior distribution on the set \( AS \) to capture a priori preferences between assertions.

We may use the \( \lambda \)-mapping (definition 7) to identify \( A \), the set of appropriate expressions to describe an object given appropriate label sets \( (L, F) \), as follows:

\[
A(L, F) = \{ \emptyset \in AS : (L, F) \in \lambda(\emptyset) \} \cup \{ \bar{\emptyset} \in AS : (L, F) \not\in \lambda(\emptyset) \} \]

Now since the sets \( P^I_\emptyset \) and \( P^I_{\bar{\theta}} \) are uncertain with mass function \( m_x \), we naturally obtain a mass function \( m_{ax} \) on the set
of appropriate assertions to describe $x$ as follows:

**Definition 8** (Mass Assignment on Assertions). $ma_x : 2^{AS} \times 2^{AS} \to [0, 1]$ is defined such that, $\forall G \subseteq AS$,

$$ma_x(G) = \sum_{F \subseteq LA, F \subseteq LA : A(F, F) = G} x(F, F)$$

Hence, $ma_x(G)$ is the probability that $G$ is the complete set of assertions appropriate to describe $x$.

For example consider an assertion set of $AS = \{L_1, \overline{L}_1, \overline{L}_2, \overline{L}_i : i = 1, ..., n\}$. Let $LA = \{L_1, L_2\}$, and consider the following mass function $m$:

\[
\begin{align*}
  m_x(\{L_1\}, \{L_1, L_2\}) &= 0.6 \\
  m_x(\{L_1, L_2\}, \{L_1, L_2\}) &= 0.4
\end{align*}
\]

From this we obtain the following mass assignment on assertions $ma_x$:

\[
\begin{align*}
  ma_x(\{L_1, \overline{L}_2, \overline{L}_1, L_2\}) &= 0.6 \\
  ma_x(\{L_1, L_2, \overline{L}_1, \overline{L}_2\}) &= 0.4
\end{align*}
\]

We now define a prior distribution on the set of permitted assertions so that $P(A)$ is the prior probability that a given expression $A \in AS$ should be selected to describe any object in $\Omega$ before that object has been encountered. Then we may formulate a probability distribution characterising the probability of asserting an expression $A$ given the encountered object $x$ as follows:

**Definition 9** (Probability of an Assertion). Given a prior probability distribution $P$ in $AS$, we can determine a posterior distribution on $AS$ given an encountered element $x \in \Omega$ as follows: $\forall \theta \in LE, \forall x \in \Omega$,

$$P(A = A|x) = \sum_{G \subseteq AS : A \in G} ma_x(G)P(A = A|A \in G)$$

$$= P(A) \sum_{G \subseteq AS : A \in G} ma_x(G) \frac{P(G)}{P(A)}$$

### 3.1 An Illustrative Example

Consider a language $LA$, and an interpretation $I = (\Omega, \delta, \overline{\delta}, \overline{\delta})$ where $\overline{\delta} = (\overline{\delta}_1, ..., \overline{\delta}_n)$ and for each $i = 1, ..., n$, $\overline{\delta}_i \sim \delta_i(\overline{\delta}_i, \overline{\delta})$. In other words, each label $L_i$ has boundaries independent of all other label boundaries.

Now suppose that we have an assertion set of $AS = \{L_1, \overline{L}_1, \overline{L}_2, \overline{L}_i : i = 1, ..., n\}$ (i.e. we are allowing strong and weak assertions of both basic labels and their negations.) Then

$$A(F, F) = \{L_i : L_i \in F\} \cup \{\overline{L}_i : L_i \in \overline{F}\}$$

Using the notation introduced in theorem 2 and letting $F_i = \{L_1, ..., L_i\}$, $i = 1, ..., n$, and $F_0 = \emptyset$, then we now have that, for $k \leq j$,

$$A(F_k, F_j) = \{L_1, ..., L_k, \overline{L}_1, ..., \overline{L}_{k+1}, ..., \overline{L}_j, \overline{L}_{j+1}, ..., \overline{L}_n\}$$

Now assume that we have a prior on assertions given by

$$P(L_i) = \frac{p}{n}, P(\overline{L}_i) = \frac{q}{n}, P(\overline{L}_i) = \frac{\overline{q}}{n}$$

where $p + q + \overline{q} = 1$. Then obtain:

$$P(L_i|A(F_k, F_j)) = \frac{p}{kp + j\overline{p} + (n-k)\overline{q} + (n-j)q}$$

$$P(\overline{L}_i|A(F_k, F_j)) = \frac{p}{kp + j\overline{p} + (n-k)\overline{q} + (n-j)q}$$

$$P(\overline{L}_i|A(F_k, F_j)) = \frac{q}{kp + j\overline{p} + (n-k)\overline{q} + (n-j)q}$$

This gives us assertion probabilities of the form:

$$P(A = L_i|x) = \sum_{k=0}^{n} \frac{m_x(F_k, F_j)\overline{p}}{kp + j\overline{p} + (n-k)\overline{q} + (n-j)q}$$

$$P(A = L_i|x) = \sum_{k=0}^{n} \frac{m_x(F_k, F_j)\overline{p}}{kp + j\overline{p} + (n-k)\overline{q} + (n-j)q}$$

$$P(A = \overline{L}_i|x) = \sum_{k=0}^{n} \frac{m_x(F_k, F_j)\overline{q}}{kp + j\overline{p} + (n-k)\overline{q} + (n-j)q}$$

$$P(A = \overline{L}_i|x) = \sum_{k=0}^{n} \frac{m_x(F_k, F_j)\overline{q}}{kp + j\overline{p} + (n-k)\overline{q} + (n-j)q}$$

**Example 3.1.** Consider a conceptual space $\Omega = [0, 1]$, a language $LA = \{L\}$ where $L$ has prototype $P = 0$. Suppose $(\overline{\delta}, \delta)$, the underlying boundary variables, are distributed as in example 2.3, and let $\overline{k}_1 = 0.5, \overline{k}_1 = 1$.

Label $L$ then has lower and upper appropriatenesses as shown in figure 4. At the prototype $P = 0$ we have maximal lower and upper appropriateness measures for $L$. Given that $LA = \{L\}$, we have an assertion set of $AS = \{L, \overline{L}, \overline{\overline{L}}\}$, and a prior of $P(L) = p$, $P(\overline{L}) = \overline{p}$, $P(\overline{\overline{L}}) = \overline{q}$. Initially consider a prior where $p = \overline{p} = q = \overline{q} = 0.25$.

Figure 5 shows the assertion probabilities for $L$ and $\overline{L}$ for $x \in \Omega$, and figure 6 shows the assertion probabilities for $\overline{\overline{L}}$. From this it follows that for this assertion prior, $P(A = \theta|x) < P(A = \overline{\theta}|x)$ for $\theta \in \{L, \overline{L}\}$. In other words, the probability of a weak assertion is always greater than the corresponding strong assertion. If an agent chooses to assert $\arg \max_A P_x(A)$, the assertion with maximal
probability, then a strong assertion will never be chosen in preference to weak assertion.

We may observe that if we have \( p \leq \bar{p}, q \leq \bar{q} \) then
\[
P(A = L|x) < P(A = \bar{L}|x) \quad \text{and} \quad P(A = \bar{L}|x) < P(A = L|x)
\]
Indeed, if we want, for some \( x \in \Omega \), the assertion with greatest probability to be a strong assertion, we must chose a prior \( p > \bar{p} \) and \( q > \bar{q} \).  

Consider now an assertion prior of \( p = 0.3, \bar{p} = 0.2, q = 0.3, \bar{q} = 0.2 \). Now we have \( p > \bar{p} \) and \( q > \bar{q} \). The assertion probabilities for this prior are shown in figures 7 and 8. We now observe that there is a region within \( \Omega \) where
\[
P(A = \theta|x) < P(A = \bar{\theta}|x), \quad \text{and a region for which} \quad P(A = \theta|x) > P(A = \bar{\theta}|x) \quad \text{for} \quad \theta = L, \bar{L}.
\]

First consider an assertion of \( A = L \) or \( A = \bar{L} \). Then figure 7 shows that when we are close to the prototype \( P = 0 \), a strong assertion of \( L \) has a higher probability than a weak assertion \( \bar{L} \). As \( x \) increases and is further from the prototype \( P \), the probability of asserting a weak assertion \( \bar{L} \) exceeds that of the strong assertion \( L \). Similar behaviour also holds for weak and strong assertions of \( \bar{L} \) in figure 8. At maximal distance from the prototype (i.e. at \( x = 1 \)), it holds that
\[
P(A = \bar{L}|x) > P(A = L|x) \quad \text{and} \quad P(A = L|x) > P(A = \bar{L}|x).
\]
In other words the probability of strongly asserting the negation of \( L \) is greater than the probability of weakly asserting the negation of \( L \). If we consider a region closer to the prototype, for example at \( x = 0.5 \), then the probability of weakly asserting \( \bar{L} \) is greater than the probability of strongly asserting \( \bar{L} \). Hence, an agent would choose a strong positive assertion \( (L) \) at the prototype and a region close to the prototype, and a strong negative assertion \( (\bar{L}) \) at maximal distance from the prototype \( (x = 1) \) and a region

\[\mu_L(x) \leq \mu_{\bar{L}}(x) \forall x.\]

This can be shown to result from the constraint that \( \mu_L(x) \leq \mu_{\bar{L}}(x) \forall x \).

Figure 4: Lower and upper appropriateness measures of \( L \) with prototype \( P = 0 \), and boundary parameters \( k = 0.5, \bar{k} = 1 \), for \( x \in [0, 1] \).

Figure 5: Probability of asserting \( L \) and \( \bar{L} \) with prior probabilities of \( \underline{p} = \bar{p} = \underline{q} = \bar{q} = 0.25 \).

Figure 6: Probability of asserting \( \bar{L} \) and \( L \) with prior probabilities of \( \underline{p} = \bar{p} = \underline{q} = \bar{q} = 0.25 \).

4 Conclusion

In this paper we have argued that Natural Language Generation systems may benefit from an ability to use natural language in a more flexible manner. More specifically, incorporating an explicit model of vagueness can allow NLG systems to better identify assertions appropriate to a particular situation or context.

We have introduced a model of assertion based on a bipolar extension of label semantics which incorporates both indeterminism and semantic uncertainty. Within this framework we have proposed a model of assertion where
agents choose the most appropriate description given their underlying conceptual model, the object being described, and a prior distribution on possible assertions. The example presented in section 3.1 highlights the constraints which must be satisfied in order to ensure both weak and strong assertions will be asserted for objects in a conceptual space.

References


Abstract

Uncertain knowledge can be modeled by using graded probabilities rather than binary truth-values, but so far a completely satisfactory integration of logic and probability has been lacking. In particular the inability of confirming universal hypotheses has plagued most if not all systems so far. We address this problem head on. The main technical problem to be discussed is the following: Given a set of sentences, each having some probability of being true, what probability should be ascribed to other (query) sentences? A natural wish-list, among others, is that the probability distribution (i) is consistent with the knowledge base, (ii) allows for a consistent inference procedure and in particular (iii) reduces to deductive logic in the limit of probabilities being 0 and 1, (iv) allows (Bayesian) inductive reasoning and (v) learning in the limit and in particular (vi) allows confirmation of universally quantified hypotheses/sentences. We show that probabilities satisfying (i)-(vi) exist, and present necessary and sufficient conditions (Gaifman and Cournot).

The theory is a step towards a globally consistent and empirically satisfactory unification of probability and logic.

Keywords

expressive languages; probability on sentences; Gaifman; Cournot; Bayes; induction; confirmation; learning; prior; knowledge; entropy.

"The study of probability functions defined over the sentences of a rich enough formal language yields interesting insights in more than one direction."

— Haim Gaifman (1982)

1 Hutter et al. [2013] contains all technical details and proofs and more discussion.

1 Introduction

Motivation. Sophisticated computer applications generally require expressive languages for knowledge representation and reasoning. In particular, such languages need to be able to represent both structured knowledge and uncertainty [Nilsson, 1986; Halpern, 2003; Muggleton, 1996; De Raedt and Kersting, 2003; Richardson and Domingos, 2006; Hájek, 2001; Williamson, 2002].

A key goal of this research is that of integrating logic and probability; a problem that has a history going back around 300 years and for which at least three main threads can be discerned: The oldest by far is the philosophical/mathematical thread that can be traced via Boole in 1854 back to Jacob Bernoulli in 1713. An extensive historical account of this thread can be found in [Hailperin, 1996]; the idea of putting probabilities on sentences goes back to before [Łos, 1955] which contains references to even earlier material; the important Gaifman condition appeared in [Gaifman, 1964] and was further developed in [Gaifman and Snir, 1982]; in [Scott and Krauss, 1966] the theory is developed for infinitary logic; overviews of more recent work from a philosophical perspective can be found in [Hájek, 2001; Williamson, 2002; 2008b]. The second thread is that of the knowledge representation and reasoning community in artificial intelligence, of which [Nilsson, 1986; Halpern, 1990; Fagin and Halpern, 1994; Halpern, 2003; Shirazi and Amir, 2007] are typical works. The third thread is that of the machine learning community in artificial intelligence, of which [Muggleton, 1996; De Raedt and Kersting, 2003; Richardson and Domingos, 2006; Milch and Russell, 2007; de Salvo Braz, 2007; Kersting and De Raedt, 2007; Pfeffer, 2007; Goodman et al., 2008] are typical works. We admit that this categorization is rather terse, coarse, and incomplete.

An important and useful technical distinction that can be made between these various approaches is that the combination of logic and probability can be done externally or internally [Williamson, 2008b]: in the external view, probabilities are attached to sentences in some logic; in the internal view, sentences incorporate statements about probability. One can even mix the two cases so that probabilities appear both internally and externally [Halpern, 1990]. This paper takes the
external view, leaving the combination with the internal view for future work.

**Main aim.** These considerations lead to the main technical issue studied in this paper:

Given a set of sentences, each having some probability of being true, what probability should be ascribed to other (query) sentences?

We build on the work of Gaifman [1964] whose paper with Snir [1982] develops a quite comprehensive theory of probabilities on sentences in first-order Peano arithmetic. We take up these ideas, using non-dogmatic priors [Gaifman and Snir, 1982] and additionally the minimum relative entropy principle as in [Williamson, 2008a], but for general theories and in a higher-order setting. We concentrate on developing probabilities on sentences in a higher-order logic. This sets the stage for combining it with the probabilities inside sentences approach [Ng and Lloyd, 2009; Ng et al., 2008].

**Summary of key concepts.** Section 3 gives the definition of probabilities on sentences (Definition 1) and shows their close connection with probabilities on interpretations. Gaifman [1964] (generalized in Definition 6) introduced a condition, called Gaifman in [Scott and Krauss, 1966], that connects probabilities of quantified sentences to limits of probabilities of finite conjunctions. In our case, it effectively restricts probabilities to separating interpretations while maintaining countable additivity.

While generally accepted in probability theory (Definition 2), some circles argue that countable additivity (CA) does not have a good philosophical justification, and/or that it is not needed since real experience is always finite, hence only non-asymptotic statements are of practical relevance, for which CA is not needed. On the other hand, it is usually much easier to first obtain asymptotic statements which requires CA, and then improve upon them. Furthermore we will show that CA can guide us in the right direction to find good finitary prior probabilities.

Another principle which has received much less attention than CA but is equally if not more important is that of Cournot [Cournot, 1843; Shafer, 19 May 2006]: An event of probability (close to) zero will remain small even after a physical experiment is possible; or conversely, an event of probability 1 will physically happen for sure. In short: zero probability means impossible; or conversely, an event of probability 1 will physically happen for sure. In short: zero probability means impossible; or conversely, an event of probability 1 will physically happen for sure. In short: zero probability means impossibility. The history of the semantics of probability is stony [Fine, 1973]. Cournot’s “forgotten” principle is one way of giving meaning to probabilistic statements like, “the relative frequency of heads of a fair coin converges to 1/2 with probability 1”.

The contraposition of Cournot is that one must assign non-zero probability to possible events. If “events” are described by sentences and “possible” means it is possible to satisfy these sentences, i.e. they possess a model, then we arrive at the strong Cournot principle that satisfiable sentences should be assigned non-zero probability. This condition has been appropriately called ‘non-dogmatic’ in [Gaifman and Snir, 1982]. As long as something is not proven false, there is a (small) chance it is true in the intended interpretation. This non-dogmatism is crucial in Bayesian inductive reasoning, since no evidence (however strong) can increase a zero prior belief to a non-zero posterior belief [Rathmanner and Hutter, 2011].

The Gaifman condition is inconsistent with the strong Cournot principle, but consistent with a weaker version (Definition 8). Probabilities that are Gaifman and (plain, not strong) Cournot allow learning in the limit (Theorem 9 and Corollary 11).

A standard way to construct (general / Cournot / Gaifman) probabilities on sentences is to construct (general / non-dogmatic / separating) probabilities on interpretations, and then transfer them to sentences (Proposition 4). At the same time we give model-theoretic characterizations of the Gaifman condition (Theorem 7). We also give a particularly simple construction of a probability that is Cournot and Gaifman (Theorem 10) and a complete characterization of general/Cournot/Gaifman probabilities in [Hutter et al., 2013].

At the end of Section 3 we briefly and in [Hutter et al., 2013] we fully consider the important practical situation of whether and how a real-valued function on a set of sentences can be extended to a probability on all sentences; a method for determining such probabilities is given. Prior knowledge and data constrain our (belief) probabilities in various ways, which we need to take into account when constructing probabilities. Prior knowledge is usually given in the form of probabilities on sentences like “the coin has head probability 1/2”, or facts like “all electrons have the same charge”; or non-logical axioms like “there are infinitely many natural numbers”. They correspond to requiring their probability to be 1/2, extremely close to 1, and 1, respectively. It is therefore necessary to be able to go from probabilities on sentences to probability on interpretations (Proposition 3). Seldom does knowledge constrain the probability on all sentences to be uniquely determined. In this case it is natural to choose a probability that is least dogmatic or biased [Nilsson, 1986; Williamson, 2008a]. The minimum relative entropy principle can be used to construct such a unique minimally more informative probability that is consistent with our prior knowledge.

Section 4 outlines how the developed theory might be used and approximated in autonomous reasoning agents. In particular, certain knowledge, learning in the limit (Corollary 11) and the infamous black raven paradox are discussed. Section 5 contains a brief summary and future research directions.

We start with some preliminaries in the following Section 2.

2 Preliminaries

This section sets the stage for the subsequent theoretical development and applications. We introduce the black raven hypothesis, used as a running example to illustrate and motivate the theory. Then we state a natural wish-list for the prior probability distribution, and the technical requirements they translate into. This also allows us to describe the intuition behind our main results, before delving into technicalities in Section 3. Finally the used logic is outlined.

**Induction example: black ravens.** As discussed, the main goal of this paper is to unify probability and logic for learn-
ing. We illustrate and motivate the theory developed in this section by a running example, namely the confirmation of universal hypotheses. The black raven hypothesis is an infamous instantiation [Earman, 1993; Maher, 2004]. It is technically very simple, while still most reasoning systems fail on it.

Consider a sequence of ravens identified by positive integers. Let $B(i)$ denote the fact that raven $i$ is black. $i = 1, 2, 3,...$. We see a lengthening sequence of black ravens. Consider the hypothesis “all ravens are black”, that is $\forall x. B(x)$. Intuitively, observing more and more black ravens with no counter-examples increases our confidence in the hypothesis. So a plausible requirement on any inductive reasoning system is that $\Pr(\forall x. B(x) | B(1) \land ... \land B(n))$ tends to $1$ for $n \to \infty$.

Real-world problems are much more complex, but most reasoning systems fail already on this apparently simple example. For instance, Bayes/Laplace rule and Carnap’s confirmation theory fail, but Solomonoff induction works [Rathmanner and Hutter, 2011]. A more complex example is given in Section 4. Finally note that the (full) black raven paradox is more complicated and will not be discussed here.

### Wish-list

Expressive logic languages are ideally suited for representing and reasoning about structured knowledge. Uncertain knowledge can be modeled by assigning graded probabilities rather than binary truth-values to sentences. Together this suggests to put probabilities on sentences. As stated in the introduction, the main technical problem considered is:

- allowing to confirm universally quantified hypotheses
- allowing for a consistent inference procedure and in particular
- learning in the limit and in particular
- learning to confirm universally quantified hypotheses
- (vi) allow to confirm universally quantified hypotheses

### Technical requirements

We will see that this wish-list translates into the following technical requirements for a prior probability: It needs to be

1. (P) consistent with the standard axioms of Probability,
2. (CA) including Countable Additivity,
3. (CA) non-dogmatic $\equiv$ Cournot
   $\equiv$ zero probability means impossibility
   $\equiv$ whatever is not provably false is assigned probability larger than 0.
4. (G) separating $\equiv$ Gaifman
   $\equiv$ existence is always witnessed by terms $\exists \text{ logical quantifiers over variables can be replaced by meta-logical quantification over terms.}$

### Main results

In the next section we will give suitable formalizations of all requirements. We give one explicit “construction” of such probabilities. Proofs that they satisfy all our criteria, general characterizations of probabilities that satisfy some or all of the criteria, and various (counter) examples of (strong) (non)Cournot and/or Gaifman probabilities and (non)separating interpretations can be found in [Hutter et al., 2013].

We also give necessary and sufficient conditions for extending beliefs about finitely many sentences to suitable probabilities over all sentences. Seldom does knowledge induce a unique probability on all sentences. In this case it is natural to choose a probability that is least dogmatic or least biased. We show that the probability of minimum entropy relative to some Cournot and Gaifman prior (1) exists, and is (2) consistent with our prior knowledge, (3) minimally more informative, (4) unique, and (5) suitable for inductive inference. Section 4 outlines how to use and approximate the theory for autonomous reasoning agents.

### On the choice of logic

In practice, ignoring computational considerations, the more expressive the logic the better. Higher-order logic, also called simple type theory (STT), is such an expressive logic. In Hutter et al. [2013] we fully develop the theory for STT with Henkin semantics without description operator for countable alphabet.

The major ideas though work in many logics (e.g. first order), but there are important and subtle pitfalls to be avoided. Due to limited space, we will here abstract away from and gloss over the details of the used logic.

As usual we have boolean operations $\top, \bot, \land, \lor, \rightarrow$, quantifiers $\forall x, \exists y$, closed terms $t$, sentences $\varphi, \chi$, formula $\psi(x)$ with a single free variable $x$, universal hypothesis/sentence $\forall x. \psi(x)$, usually equality $\equiv$, and in STT abstraction $\lambda z$, but this is not needed here.

### 3 Theory

We define probabilities $\mu$ over sentences $\varphi$ in the usual way [Halpern, 1990]: $\mu(\varphi)$ is the probability that $\varphi$ is true in the intended interpretation, or $\mu(\varphi)$ is the subjective probability held by an agent that sentence $\varphi$ holds in the real world. It should satisfy the basic axioms of probability, and hence has the usual properties. Only countable Additivity (CA) enters later and differently, since finitary logics lack infinite conjunctions of sentences.

### Definition 1 (probability on sentences)

A probability (on sentences) is a non-negative function $\mu : S \to \mathbb{R}$ satisfying the following conditions:

- If $\varphi$ is valid, then $\mu(\varphi) = 1$.
- If $\neg(\varphi \land \chi)$ is valid, then $\mu(\varphi \lor \chi) = \mu(\varphi) + \mu(\chi)$.
- Conditional probability: $\mu(\varphi | \chi) := \frac{\mu(\varphi \land \chi)}{\mu(\chi)}$.

A sentence $\varphi$ is said to be valid, if it is true in all (Henkin) interpretations. We also define probabilities on interpretations, which is closer to conventional measure theory. Let
\(mod(\varphi)\) be the class of (Henkin) interpretations in which \(\varphi\) is true, and \(\mathcal{I} := mod(\top)\) be the class of all (Henkin) interpretations, and \(B\) be the \(\sigma\)-algebra generated by \(\{mod(\varphi) : \varphi \in S\}\). Then:

**Definition 2 (probability on interpretations)** A function \(\mu^* : B \to \mathbb{R}\) is a (CA) probability on \(\sigma\)-algebra \(B\) if \(\mu^*(\emptyset) = 0\) and \(\mu^*(\mathcal{I}) = 1\) and for all countable collections \(\{A_i\}_{i \in \mathbb{I}} \subseteq B\) of pairwise disjoint sets it holds that \(\mu^*(\bigcup_{i \in \mathbb{I}} A_i) = \sum_{i \in \mathbb{I}} \mu^*(A_i)\).

**Probability on sentences \(\Leftrightarrow\) interpretations.** There is a close relationship between probabilities on sentences and probabilities on interpretations. This allows us to exploit (some) results from measure theory, valid for the latter, also for the former.

**Proposition 3 \((\mu \Rightarrow \mu^*)\)** Let \(\mu : S \to \mathbb{R}\) be a probability on \(S\). Then there exists a unique probability \(\mu^* : B \to \mathbb{R}\) such that \(\mu^*(mod(\varphi)) = \mu(\varphi)\) for each \(\varphi \in S\).

The proof uses compactness of the class of (Henkin) interpretations \(\mathcal{I}\) and Caratheodory’s unique-extension theorem. The converse is elementary:

**Proposition 4 \((\mu^* \Rightarrow \mu)\)** Let \(\mu^* : B \to [0, 1]\) be a probability on \(B\). Define \(\mu : S \to \mathbb{R}\) by \(\mu(\varphi) = \mu^*(mod(\varphi))\) for each \(\varphi \in S\). Then \(\mu\) is a probability on \(S\).

**Problems.** Consider the black raven example: Intuitively, knowledge of \(\{B(1), B(2), \ldots\} \equiv \{B(i) : i \in \mathbb{N}\}\) should imply \(\forall x.B(x)\).

Problem is that this is not true in all models. There are non-standard models of the natural numbers in which \(x = n\) is invalid for all \(n = 1, 2, 3, \ldots\). The reason is that the natural numbers have neither a categorical axiomatization in first order logic, nor in STT with Henkin semantic. They do in STT with normal semantics, but there compactness and hence the crucial Proposition 3 fails. So in either case we have a problem.

The solution is to exclude such unwanted interpretations. The natural generalization of “1, 2, 3, ...” for general theories is “all terms \(t\).”

**Definition 5 (separating interpretation)** An interpretation \(I\) is separating iff for all formulas \(\psi(x)\) the following holds: If \(I\) is a model of \(\exists x.\psi(x)\), then there exists a closed term \(t\) such that \(I\) is a model of \(\psi(x/t)\), where \(\psi(x/t)\) is \(\psi\) with all free \(x\) replaced by \(t\).

Informally this means that existence is always witnessed by terms. For objects to exist we must be able to name them. It is important to note that our vocabulary from which the closed terms are constructed is fixed up front and the same for all \(I\). Otherwise we could trivially make every interpretation separating by adding sufficiently many new constants to the theory, as e.g. done in Henkin’s construction. We need to avoid such new constants since they would ruin induction.

In complete analogy to above, let \(\overline{mod}(\varphi)\) be the set of separating models of \(\varphi, \mathcal{I} = \overline{mod}(\top)\) be the set of all separating interpretations, and \(\overline{B}\) be the \(\sigma\)-algebra generated by \(\{\overline{mod}(\varphi) : \varphi \in S\}\). Note that all \(\overline{mod}(\varphi)\) are \(\overline{B}\)-measurable.

Next we effectively avoid non-separating interpretations by requiring the probability on them to be zero:

**Definition 6 (Gaifman condition)** We call \(\mu\) Gaifman iff

\[
\mu(\forall x.\psi(x)) = \lim_{n \to \infty} \mu(\bigwedge_{i=1}^n \psi(x/t_i))
\]

for all \(\psi\), where \(t_1, t_2, \ldots\) is an enumeration of (representatives of) all closed terms (of same type as \(x\)).

Informally this means that logical quantifiers over variables can be replaced by meta-logical quantification over terms: With ‘representative’ we mean that one term per \(\forall\)-equivalence class is sufficient. For the theory of natural numbers, all terms (of type Nat), equal \(1\) or \(2\) or ..., e.g. \(t = \frac{2}{3} + \frac{3}{2}\) equals \(\frac{13}{6}\), hence does not need to be listed separately.

**Theorem 7 (\(\mu^*(\mathcal{I} \setminus \mathcal{I}) = 0 \Leftrightarrow \mu \text{ is Gaifman})**

For any probability \(\mu : S \to \mathbb{R}\) on sentences and probability \(\mu^* : B \to \mathbb{R}\) on interpretations (one-to-one) related by \(\mu^*(mod(\varphi)) = \mu(\varphi)\) it holds that: \(\mu^*(\mathcal{I} \setminus \mathcal{I}) = 0 \Leftrightarrow \mu \text{ is Gaifman.}\)

**Induction still does not work.** Unfortunately, even \(\mu\) satisfying the Gaifman condition may fail to confirm universal hypotheses. The reason is that \(\mu(\forall x.B(x)) = 0\) if \(\mu(\forall x.B(x)) = 0\). This is the infamous Zero-Prior problem in philosophy of induction. If your prior excludes some hypothesis, no amount of evidence can confirm it. Carnap’s and most other confirmation theories fail, since they (implicitly & unintentionally) have \(\mu(\forall x.B(x)) = 0\). Why is this problem hard? “Naturally” \(\mu(\forall x.B(x)) \leq \mu(B(1) \land \ldots \land B(n))\) \(\to 0\). Think of independent events with probability \(p < 1\), then \(p \cdot p \cdot p \cdots \to 0\). But it’s not hopeless: We just demand \(\mu(\forall x.\psi(x)) > 0\) for all \(\psi\) for which this is possible and/or reasonable, which turns out to be the \(\varphi\) that have separating models.

We call this Cournot’s principle: Informally stated, probability zero/one means impossibility/certainty, or whatever is not provably false is assigned probability larger than 0, or all (sensible) prior probabilities should be non-zero, or be as non-dogmatic as possible. Formally:

**Definition 8 (Cournot probability)** A probability \(\mu : S \to \mathbb{R}\) is Cournot if, for each \(\varphi \in S\), \(\varphi\) has a separating model implies \(\mu(\varphi) > 0\).

We cannot drop the ‘separating’, since this would then conflict with the Gaifman condition. Note that Cournot requires sentences, not interpretations, to have strictly positive probability, so is applicable even for uncountable model classes.

**Black ravens – again.** Consider a theory in which all terms (of type Nat) represent natural numbers. Let \(\mu\) be Cournot and Gaifman, then:

\[
\begin{align*}
\mu(\forall x.B(x)) &= \mu(\forall x.B(x)) \\
&= \mu(B(1) \land \ldots \land B(n)) \\
&\overset{\text{Def. of } \mu(\psi)}{=} \mu(\forall x.B(x)) \\
&\overset{n \to \infty}{\approx} \mu(\forall x.B(x)) \\
&= 1
\end{align*}
\]

[\(\mu\) is Gaifman]

[\(\mu\) is Cournot]
Finally induction works! This example generalizes: The Cournot and Gaifman conditions are sufficient and necessary for confirming universal hypotheses.

**Theorem 9 (confirmation of universal hypotheses)** $\mu$ can confirm all universal hypotheses that have a separating model $\Leftrightarrow$ $\mu$ is Cournot and Gaifman.

What remains to be shown is whether such $\mu$ actually exist. General characterizations are given in [Hutter et al., 2013]. A particularly simple “construction” is as follows:

**Theorem 10 (Constructing a Cournot and Gaifman prior $\mu$)**

The following $\mu$ is Cournot and Gaifman:

- Enumerate the countable set of sentences that have a separating model, $\chi_1, \chi_2, \ldots$
- For each sentence, $\chi_i$, choose a separating interpretation that makes it true.
- Assign probability mass $\frac{1}{i(i+1)}$ to that interpretation.
- Define $\mu^*$ to be the probability on this countable set of interpretations.
- Define $\mu$ to be the corresponding distribution over sentences.

Alternatively one can enumerate all sentences $\varphi_1, \varphi_2, \varphi_3, \ldots$, and in an infinite binary tree label each left (right) branch at depth $n$ with $\neg \varphi_n (\varphi_n)$ and assign probabilities to each node as detailed in [Hutter et al., 2013, Thm.52], which in turn defines $\mu$.

Very powerful Cournot and Gaifman (C&G) probabilities can be constructed as follows: Let $\mathcal{M} = \{\nu_1, \nu_2, \ldots\}$ be any finite or countable class of Gaifman probabilities of interest. These are usually priors that are potentially true, e.g. i.i.d. probabilities such as $\nu(B(1) \land \ldots \land B(n)) = \left(\frac{1}{2}\right)^n$ which is not Cournot. Now define the mixture $\xi(\varphi) := \sum_{\nu \in \mathcal{M}} \nu(\varphi)^{x/\chi}$, which is also Gaifman and mimics Solomonoff’s construction [Solomonoff, 1964; Hutter, 2005]. If for every sentence $\chi$ that has a separating model there exists a $\nu \in \mathcal{M}$ such that $\nu(\chi) > 0$, then $\xi$ is also Cournot, since $\xi(\chi) > \nu(\chi) > 0$. If this is not the case, we can simply add some one/any C&G prior $\mu$ to $\mathcal{M}$, e.g. the one from Theorem 10, which makes $\xi$ C&G. Since $\xi$ dominates all $\nu \in \mathcal{M}$, the Merging-of-Opinions theorem [Blackwell andDubins, 1962] guarantees that $\xi$ converges to $\nu$ in total variation with $\nu$ probability 1 for any $\nu \in \mathcal{M}$. This means, while the Cournot condition rules out e.g. i.i.d. distributions, there are C&G probabilities $\xi$ that converge to them, provided the data warrant it, and that is usually all we need.

While asymptotic convergence works equally for any C&G probability, the degree of confirmation from finite sample size depends on the specific construction. To achieve fast convergence for $\mu$ constructed in Theorem 10 one should sort sentences in decreasing order of “relevance” and pick “natural” models. note that $1/i(i+1)$ is nearly as uniform as possible, hence the order dependence is benign compared to e.g. $2^{-x}$.

**Minimum more informative probability.** Knowledge is usually given as constraints on some probability distribution $\rho$. Hard facts have $\rho(\text{fact}) = 1$, while uncertain knowledge has $0 < \rho < 1$. This still leaves many choices for $\rho$. In our context it is natural to start with some C&G prior $\mu$, and find a “minimally more informative” $\xi$ consistent with the knowledge base. A natural notion of “minimally more informative” is the minimum relative entropy.

More formally, the task is: Given a C&G prior distribution $\xi$ over sentences, and a self-consistent set of constraints on probabilities, $\rho(\varphi_1) = a_1, \ldots, \rho(\varphi_n) = a_n$ given for some sentences $\varphi_1, \ldots, \varphi_n$. Find the distribution $\rho$ that minimizes $KL(\rho||\xi)$ under the constraints.

For example, given a prior distribution $\xi$, minimally adjust it so that it obeys the constraints:

A) $\rho(\forall x. \forall y. x < 6 \Rightarrow y > 6) = 0.7$
B) $\rho(\{\text{flies Tweety}\}) = 0.9$
C) $\rho(\{\text{commutative +}\}) = 0.9999$

The solution consists of the following steps: (i) choose a prior $\xi$, e.g. the one in Theorem 10. (ii) determine the consistency of the knowledge base $\{\rho(\varphi_i) = a_i\}$. Sufficient conditions are given in [Hutter et al., 2013]. (iii) $KL(\rho||\xi)$ can be defined as $KL(\rho || \xi^*)$, where the latter is the standard measure-theoretic definition. We have derived explicit finite expression of $KL(\rho || \xi)$ without reference to probabilities on interpretations, and finite equation systems for minimizing $KL(\rho || \xi)$ w.r.t. $\rho$ under constraints $\{\rho(\varphi_i) = a_i\}$.

In effect, the constraints partition the space of (separable) interpretations $\tilde{\mathcal{I}}$, and the $\rho^*$ corresponding to the distribution $\rho^* = \arg \min_{\rho} KL(\rho || \xi)$, which is also Cournot, since $\xi(\varphi) > \rho(\varphi)$ for any $\varphi \in \mathcal{I}$. That minimizes the relative entropy $\mathcal{K}$ is a multiplicative re-weighting of $\xi$, with constant weight across each partition. This is depicted in the example below, where pixels correspond to interpretations, their intensity to their probability, and each (mixed) color to a region with uniform multiplicative re-weight. All derivations and equations can be found in [Hutter et al., 2013].

4 User Manual

This section outlines how (approximations of) the theory developed in Section 3 might be used in autonomous reasoning agents. We discuss the special case of certain knowledge and how it can be used to make inferences about statements that are not logical implications of the knowledge base. For instance, if our agent has observed a large number of ravens which are all black without exception, how strongly should it believe in the hypothesis that “all ravens are black”? We
construct an agent that can learn in the limit in the usual time-series forecasting setting with an observation sequence indexed by natural numbers.

**Certain knowledge.** A common case of knowledge is a set of sentences $\varphi_i$, each having degree of belief 1 (that is, $\mu_0(\varphi_i) = 1$, for $i = 1, \ldots, n$). In other words, there is certainty that each $\varphi_i$ is valid in the intended interpretation. This corresponds to non-logical axioms in a theory. Let $\xi$ be a Cournot probability and suppose that $\mu$ is minimally more informative than $\xi$ given $\mu_0$. For this situation, one can show that $\mu$ satisfies

$$\mu(\varphi) = \xi(\varphi | \varphi_1 \land \cdots \land \varphi_n), \quad (1)$$

for $\varphi \in S$. Consequently, either $\varphi_1 \land \cdots \land \varphi_n$ is satisfiable (leading directly to the above definition for $\mu$) or else it is not, in which case there are no solutions and $\mu$ cannot be defined at all.

A further special case beyond the one just considered is when $\varphi$ is a logical consequence of $\varphi_1 \land \cdots \land \varphi_n$. In this case, $\mu(\varphi) = \xi(\varphi | \varphi_1 \land \cdots \land \varphi_n)$, as one would expect. Similarly when $\neg \varphi$ is logical consequence, then $\mu(\varphi) = 0$.

Note that, while it is important that the prior $\xi$ be Cournot, it is just as important that the posterior $\mu$ be allowed not to be Cournot. The prior should be Cournot so that the KL divergence is as widely defined as possible or, more intuitively, to make sure sentences having a separating model are not forced to have $\mu$-probability 0. On the other hand, the probability $\mu$ should be allowed to be 0 on sentences having a separating model since the evidence in the form of the probabilities on $\varphi_1, \ldots, \varphi_n$ may imply this. This is apparent, for example, for the case where each $\varphi_i$ has probability 1: according to this evidence, any sentence (even one having a separating model) that is disjunctive from $\varphi_1 \land \cdots \land \varphi_n$ must have $\mu$-probability 0.

**Black ravens.** Let the evidence consist of the sentences $B(1), \ldots, B(n)$, whence $\varphi_i \equiv B(i)$, for $i = 1, \ldots, n$. Let $\mu_0 : \{B(1), \ldots, B(n)\} \rightarrow [0, 1]$ be defined by $\mu_0(B(i)) = 1$, for $i = 1, \ldots, n$. Thus the degree of belief that the $i$th raven is black is 1, for $i = 1, \ldots, n$. Suppose that $\xi$ is an uninformative prior that is C&G. Since a-priori there are no constraints (on $B$), this implies that $\xi(\forall x. B(x)) > 0$. Let $\mu$ be a probability that is minimally more informative than $\xi$ given $\mu_0$. Thus $\mu$ is given by (1).

Now consider the sentence $\forall x. B(x)$. This is clearly not a logical consequence of the evidence, but one can use $\mu$ to ascribe a degree of belief that it is true and, furthermore, investigate what happens to this probability as the number of black ravens increases. Equation (1) and $\mu_0(B(i)) = 1$, for $i = 1, \ldots, n$, and then Theorem 9 applied to C&G $\xi$ show that $\mu(\forall x. B(x)) = \xi(\forall x. B(x) | B(1) \land \cdots \land B(n)) \xrightarrow{n \to \infty} 1$.

Thus, as the number of observed black ravens increases, the degree of belief that all ravens are black approaches 1. Of course this also implies the weaker statement that our belief in the next raven being black tends to one:

$$\xi(B(n+1) | B(1) \land \cdots \land B(n)) \xrightarrow{n \to \infty} 1$$

**Naive black ravens.** Continuing the preceding example, suppose given the evidence $B(1), \ldots, B(n)$, each having probability 1, one wants to know the degree of belief for $B(n+1)$. Most probabilistic reasoning systems, if they have at all the ability to provide prior distributions, give $\xi(B(1) \land \cdots \land B(n)) = \frac{1}{n}^n$ or similar, which can usually be traced back to a (naive) application of the maximum entropy or indifference principle, and/or to first assigning probabilities to quantifier-free probabilities and then extending them to quantified formulas. In this case

$$\xi(B(n+1) | B(1) \land \cdots \land B(n)) = \frac{\xi(B(1) \land \cdots \land B(n) \land B(n+1))}{\xi(B(1) \land \cdots \land B(n))} = \frac{1}{2^2}.$$

Thus, for this prior, knowing the evidence so far, even for large $n$, does not give any information about $B(n+1)$. But it gets worse: Assume $\xi$ is somehow extended to a probability on all $S$. Then for any $m \geq n$,

$$\xi(\forall x. B(x) | B(1) \land \cdots \land B(n)) \leq \xi(B(1) \land \cdots \land B(n) | B(1) \land \cdots \land B(n)) = \frac{1}{2^{m-n}}$$

hence $\xi(\forall x. B(x) | B(1) \land \cdots \land B(n)) \equiv 0$ for all $n$, i.e. universal hypotheses can not be confirmed. Even more seriously, we would be absolutely sure that non-black ravens exist

$$\xi(\exists x. \neg B(i) | B(1) \land \cdots \land B(n)) \equiv 1$$

and no number of observed black ravens $n$ without any counter examples will ever convince us otherwise. The crucial requirement to avoid these problems was to include quantified sentences when constructing a prior and ensure it is Cournot (even when only making inferences about unquantified sentences like $B(n+1)$).

**Corollary 11 (learning in the limit)** Let $\psi$ be a formula with free variable $x$ of type Nat, $\mu$ be a Gaifman probability on sentences, and $\mu(\forall x. \psi(x)) > 0$. Then

$$\lim_{n \to \infty} \mu(\forall x. \psi(x) | \psi(0) \land \cdots \land \psi(n)) = 1$$

This generalizes the black raven example and follows from Theorem 9. In particular, learning in the limit is possible for the C&G probability constructed in Theorem 10, provided $\forall x. \psi(x)$ has a separating model.

The proof crucially exploits that $0, 1, 2, \ldots$ are representatives of all terms of type Nat. As discussed in [Hutter et al., 2013], this would no longer be true had we introduced a description operator into our logic. Corollary 11 would break down and universal hypotheses over the natural numbers could not be inductively confirmed, not even asymptotically.

**Approximations.** The construction of C&G $\mu$ in Theorem 10 required to determine particular separating models for $\mathcal{X}_i$ and
to determine whether they are also models of other sentences \( \varphi \).

Assume we had some calculus to determining whether sentences have (no) separating model. Even an asymptotic or approximate or incomplete calculus may be of use. Fix a sequence on-the-fly of all sentences \( \varphi_1, \varphi_2, \varphi_3, \ldots \) (once and for all). Determine the subsequence of all sentences \( \chi_1 = \varphi_1, \chi_2 = \varphi_2, \ldots \) with separating models (on the fly).

In order to determine \( \mu \) to accuracy \( \varepsilon > 0 \) for some finite number of sentences \( \{ \varphi_1, \ldots, \varphi_m \} \) of interest, we have to assign probability \( \frac{1}{\max(i+1)} \) “only” to \( \chi_i \) for \( i \leq m := \max\{ \frac{1}{\varepsilon}, i_1, \ldots, i_n \} \), i.e. determine finitely many cases. If a new sentence \( \varphi_{m+1} \) of interest “arrives” or higher precision is needed, \( m \) can be increased appropriately (that’s what was meant with on-the-fly).

**Work flow example for a simple inductive reasoning agent.** Below we present an example of a fictitious inductive reasoning agent. It is fictitious, since many operations are incomputable. In practice one needs to employ approximations to handle probabilities outside sentences.

1. **Assume the agent has been endowed with some background knowledge e.g. about kinetics, colors, biology, birds, etc.** Its knowledge is represented in the form of a finite set of ground knowledge e.g. about kinetics, colors, biology, birds, etc. Its knowledge is represented in the form of a finite set of sentences \( \{ \varphi_1, \ldots, \varphi_n \} \) that hold for sure \((\mu_0(\varphi_i) = 1 \text{ for some } i)\) or with some probability \(0 < \mu_0(\varphi_i) < 1\) for the other \( i \).

2. In [Hutter et al., 2013] we derive sufficient conditions (hierarchical, sub-additive, eligible) for \(\mu_0\) to be consistent. This task is akin to the general problem of maintaining consistent knowledge bases.

3. Next, use an approximation of a C&G \( \xi \) prior, e.g. as defined in Theorems 10 or the mentioned tree constructions as outlined above and detailed in [Hutter et al., 2013]. The agent now constructs the minimally more informative probability \( \mu \), which has been shown to exist and be Gaifman.

4. Let \( o_1, o_2, o_3, \ldots \) be the agent’s life-time sequence of past and future observations of all kinds of objects, ravens and otherwise, all it has/ever will observe, e.g. \( o_n \) is what the agent sees \( n \) seconds after it has been switched on.

5. Assume current time is \( t \) and the agent needs to hypothesize about the world to decide its next action, e.g. whether some observed regularity is “real”. For instance, “if observation at time \( k \) is a raven, is it also black?” We can formalize this with a monadic predicate \( \psi \) for type \( Nat \) with the intended interpretation of \( \psi(\bar{\xi}) \) as “if observation at time \( \bar{\xi} \) is a raven, it is black”.

6. Of course the answer to \( \psi(1), \ldots, \psi(n) \) is immediate, since \( o_1, \ldots, o_n \) have already been observed. If they are all true, the agent may start to wonder whether “all ravens are black”, or formally, whether \( \forall x. \psi(x) \) is true. Note that non-raven observations in the sequence are allowed.

7. If the agent is equipped with our inductive reasoning system, its degree of belief in this hypothesis is \(\mu(\forall x. \psi(x)) = \psi(1) \land \cdots \land \psi(n) \).

8. This result can be the basis for some decision process maximizing some utilities resulting in an informed action.

Is the degree of belief derived in Step 7 and used in Step 8 reasonable? At least asymptotically Corollary 11 ensures that in the limit the agent’s belief tends to 1, which is very reasonable. So our system of inductive reasoning at least passes this test. Most other inductive reasoning systems have difficulties in getting this right [Rathmanner and Hutter, 2011].

**5 Conclusion**

This paper provided much of the foundation for the design of an integrated probabilistic reasoning system that can handle probabilities outside sentences.

We have shown that a function from sentences to \( \mathbb{R} \) that is a well defined probability distribution with all of our criteria exists. In particular we gave a theoretical construction for a prior that meets the conditions, and showed that minimum relative entropy inference is well defined in this setting.

Besides proofs and more details and discussion, Hutter et al. [2013] additionally give general characterizations of probabilities that meet some or all of our criteria, and give various (counter) examples of (strong) (non)Cournot and/or Gaifman probabilities and (non)separating interpretations.

Overall, the results are a step towards a globally consistent and empirically satisfactory unification of probability and logic for learning.

There is much left for future research: To combine probabilities inside and outside sentences as in [Halpern, 1990], to incorporate ideas from Solomonoff induction to get optimal priors [Rathmanner and Hutter, 2011], to include the description operator(s) \( \langle \cdot, \varepsilon \rangle \), and to investigate a number of other theoretical questions. The main challenge for the future lies in the discovery of reasonable approximation schemes for the different currently incomputable aspects of the general theory.

**Acknowledgements.** We thank the reviewers for their feedback. The research was partly supported by the Australian Research Council Discovery Project DP0877635 “Foundations and Architectures for Agent Systems”. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

**References**


Prime Forms in Possibilistic Logic

Guilin Qi
School of Computer Science and Engineering
Southeast University, China

Laurent Perrussel
IRIT-Université Toulouse 1 Capitole
France

Abstract
Possibilistic logic is a weighted logic used to represent uncertain and inconsistent knowledge. Its semantics is often defined by a possibility distribution, which is a function from a set of interpretations to a totally ordered scale. In this paper, we consider a new semantic characteristic of knowledge bases in possibilistic logics (or possibilistic knowledge bases) by a generalized notion of propositional prime implicant, which we call prioritized prime implicant. We first consider several desirable properties of a prioritized prime implicant for characterizing possibilistic knowledge bases. Some examples show that existing generalizations of prime implicants in possibilistic logic do not satisfy all of these properties. We then provide a novel definition of prioritized prime implicant, which is a set of weighted literals that may be inconsistent. We show that the prioritized prime implicants satisfy all the desirable properties. Finally, we discuss the problem of computing prioritized prime implicants of a possibilistic knowledge base.

1 Introduction
Possibilistic logic [Dubois et al., 1994] or possibility theory offers a convenient tool for handling uncertain or prioritized formulas and coping with inconsistency. At the syntactic level, it is a weighted logic which attaches to each formula with a weight belonging to a totally ordered scale, such as \([0,1]\), where the weight is interpreted as the certainty level of the formula. A possibilistic knowledge base is a set of weighted formulas. At the semantic level, it is based on the notion of a possibility distribution, which is a mapping from the set of interpretations \(\Omega\) to interval \([0,1]\). In the last 20 years, possibilistic logic plays an important role in knowledge representation and reasoning. It has been shown in [Dubois and Prade, 1991] that necessity measures defined over formulas in possibilistic logic is actually the numerical counterparts of epistemic entrenchment relations defined in belief revision [Gardenfors, 1988]. Because of this correspondence, many revision operators or merging operators have been proposed in possibilistic logic, such as those in [Dubois and Prade, 1992; Benferhat et al., 2002b; 2002a; Benferhat and Kaci, 2003; Qi et al., 2004; Qi, 2007].

The notion of prime implicants (or its dual notion called prime implicates) has been widely investigated in classical logic. It plays an important role in automated reasoning, knowledge compilation [Cadoli and Donini, 1997] and consequence finding [Marquis, 2000]. Prime implicants have also been to define revision operators or merging operators in propositional logic [Marchi et al., 2010; Marchi and Perrussel, 2011]. In [Qi et al., 2010], the authors generalize the notion of prime implicants to possibilistic logic, called weighted prime implicants, and define measures of conflict and agreement between two possibilistic knowledge bases using weighted prime implicants. They also apply weighted prime implicants to define merging operators in possibilistic logic in [Liu et al., 2006]. However, the definition of weighted prime implicants is problematic. It has been shown in [Qi and Wang, 2012] that weighted prime implicants cannot be used to characterize a possibilistic knowledge base. That is, it is not the case that a possibilistic formula can be inferred from a possibilistic knowledge base if and only if the formula can be inferred from the disjunction of all weighted prime implicants of the knowledge base. Even worse, there may exist a possibilistic knowledge base that does not have any weighted prime implicants. A modified definition of weighted prime implicants is then given in [Qi and Wang, 2012] that satisfies some desirable properties falsified by the previous definition. However, the weighted prime implicants defined in [Qi and Wang, 2012] still cannot be used to characterize a possibilistic knowledge base.

In this paper, we provide a novel definition of prime forms in possibilistic logic, called prioritized prime implicants. Unlike existing definitions of a (weighted) implicant, a prioritized implicant is a set of weighted literals that may be inconsistent. To define a prioritized implicant, a paraconsistent consequence relation for a set of weighted literals is given. A prioritized prime implicant is then defined as a prioritized implicant that is minimal w.r.t. a lexicographic order. We show that the prioritized prime implicants can be used to characterize a possibilistic knowledge base and satisfy some other desirable properties. Finally, we discuss the problem of computing prioritized prime implicants of a possibilistic knowledge base. Our method for computing prioritized prime implicants is a modification of the method for computing weighted prime implicants given in [Qi et al., 2010]. The main difference is that the modified method computes prioritized prime implicants, which may be sets of inconsistent weighted literals.
but the original method does not generate a set of inconsistent weighted literals. We show that the modified method is syntax-independent and can compute all the prioritized prime implicants of a possibilistic knowledge base.

The rest of the paper is organized as follows. In Section 2, we introduce some basic notions of propositional logic and possibilistic logic. We then discuss the problems of existing definitions of prime forms in possibilistic logic in Section 3. After that, we give a new definition of prime forms in possibilistic logic in Section 4. Finally, we conclude this paper in Section 5.

2 Preliminaries

2.1 Propositional logic

We consider a propositional language \( \mathcal{L}_{PS} \) defined from a finite set of propositional variables (also called atoms) \( PS \) and the usual connectives. Formulas are denoted by Greek letters \( \phi, \psi, \ldots \). The classical consequence relation is denoted as \( \vdash L \). An interpretation is a total function from \( PS \) to \( \{true, false\} \). The definition of an interpretation can be extended to formulas in a standard way. An interpretation is a model of a formula if it assigns truth value true to the formula. A knowledge base \( K \) is a finite set of propositional formulas. An interpretation is a model of a knowledge base if it satisfies all the formulas in it. \( K \) is consistent if it has a model. Two knowledge bases \( K_1 \) and \( K_2 \) are equivalent, denoted \( K_1 \equiv K_2 \), if they have the same set of models.

A literal is either an atom or the negation of an atom. Let \( l \) be a literal, we denote by \( l^c \) the complement of \( l \). A clause \( \phi \) is a disjunction of literals: \( \phi = l_1 \lor \ldots \lor l_n \) and its dual clause, or term \( D \), is a conjunction of literals: \( D = l_1 \land \ldots \land l_n \). A term \( D \) is an implicant of formula \( \phi \) iff \( D \vdash L \phi \) and \( D \) does not contain two complementary literals. A prime implicant of knowledge base \( K \) is an implicant of \( D \) such that for every other implicant \( D' \) of \( K \), \( D \not\subseteq D' \).

2.2 Possibilistic logic

We introduce the syntax of possibilistic logic [Dubois et al., 1994]. A possibilistic formula is a pair \((\phi, a)\), where \( \phi \) is a propositional formula and \( a \in [0,1] \). A possibilistic literal is a pair \((l, a)\), where \( l \) is a literal and \( a \in [0,1] \); a possibilistic term is a set of possibilistic literals. In this paper, we assume that there do not exist two pairs \((l, a)\) and \((l, b)\) such that \( a \neq b \) in a possibilistic term. The uncertain or prioritized pieces of information can be represented by a possibilistic knowledge base which is a finite set of possibilistic formulas of the form \( B = \{(\phi_1, a_1): i = 1, \ldots, n\} \). The classical base associated with \( B \) is \( B^* = \{\phi_i|(\phi_i, a_i) \in B\} \). A possibilistic knowledge base \( B \) is consistent iff its classical base \( B^* \) is consistent.

The semantics of possibilistic logic is based on the notion of a possibility distribution \( \pi \) which is a mapping from the set of interpretations to interval \([0,1]\). The possibility degree \( \pi(\omega) \) represents the degree of compatibility (resp. satisfaction) of \( \omega \) with the available beliefs about the real world. From a possibility distribution \( \pi \), the necessity degree of formula \( \phi \) is defined as \( N_\pi(\phi) = 1 - \Pi_\pi(\neg \phi) \), where \( \Pi_\pi(\phi) = \max \{\pi(\omega): \omega \in \Omega, \omega \models \phi\} \). The interpretation of a possibilistic formula \((\phi, a)\) is that the necessity degree of \( \phi \) is at least equal to \( a \), i.e. \( N(\phi) \geq a \).

**Definition 1.** Let \( B \) be a possibilistic knowledge base, and \( a \in [0,1] \). The a-cut (resp. strict a-cut) of \( B \) is \( B_{\geq a} = \{\phi_i \in B^*: |(\phi_i, b_i) \in B \text{ and } b_i \geq a\} \) (resp. \( B_{> a} = \{\phi_i \in B^*: |(\phi_i, b_i) \in B \text{ and } b_i > a\} \).

There are two entailment relations in possibilistic logic.

**Definition 2.** Let \( B \) be a possibilistic knowledge base. A possibilistic formula \((\phi, a)\) is a weak possibility consequent of \( B \) if it satisfies all the formulas in \( B \), denoted by \( B \vdash \pi (\phi, a) \), where \( \pi(\phi, a) = \max \{a_i: (\phi_i, b_i) \in B \text{ and } b_i \geq a\} \). A possibilistic formula \((\phi, a)\) is a possibility consequent of \( B \), denoted \( B \vdash _\pi (\phi, a) \), if (i) \( B_{\geq a} \) is consistent; (ii) \( B_{\geq a} \vdash \pi \phi \); (iii) \( \forall b > a, B_{> b} \vdash \pi \phi \).

Two possibilistic knowledge bases \( B \) and \( B' \) are said to be equivalent, denoted by \( B \equiv B' \), iff \( \forall a \in [0,1], B_{\geq a} \equiv B'_{\geq a} \).

The disjunction of two possibilistic terms is defined in [Qi and Wang, 2012] as follows: \( D_1 \lor D_2 = \{(l_i \lor l_j, a_i a_j) | (l_i, a_i) \in D_1, (l_j, b_j) \in D_2\} \). Since \( \lor \) is associative and commutative, the disjunction of more than two possibilistic terms can be easily defined.

3 Problems with Existing Definitions

In this section, we introduce the definitions of weighted prime implicants of a possibilistic knowledge base defined in [Qi et al., 2010] and [Qi and Wang, 2012] and discuss their problems.

Let \( B = \{(\phi_1, a_1), \ldots, (\phi_n, a_n)\} \) be a possibilistic knowledge base where each \( \phi_i \) is a clause. In [Qi et al., 2010], a weighted implicant of \( B \) is a possibilistic term \( D = \{(\psi_1, b_1), \ldots, (\psi_k, b_k)\} \) such that \( D \vdash _\pi B \), where \( \psi_i \) are literals such that no two complementary literals exist. Let \( D \) and \( D' \) be two weighted implicants of \( B \), \( D \) is said to be subsumed by \( D' \), denoted as \( D \preceq D' \), iff \( D \equiv D' \) for all \( (\psi_i, a_i) \in D \), there exists \( (\psi_i, b_i) \in D' \) with \( b_i \leq a_i \) (\( b_i \) is 0 if \( \psi_i \in D' \) but \( \psi_i \not\in D'^* \)). The relation \( \preceq \) is used to compare two weighted implicants. A weighted prime implicant of \( B \) is a weighted implicant that is not subsumed by any other weighted implicant of \( B \). Formally, we have the following definition.

**Definition 3.** A weighted prime implicant (WPI) of \( B \) is a weighted implicant of \( B \) such that there does exist another weighted implicant \( D' \) of \( B \) such that \( D \) is subsumed by \( D' \).

It has been shown in [Qi and Wang, 2012] that WPIs do not satisfy the following two desirable properties:

**Property 1.** for any consistent possibilistic knowledge base, it has at least one consistent WPI.

**Property 2.** for any consistent possibilistic knowledge base \( B \) and any formula \( \phi \), \( B^* \vdash \phi \) iff \( \bigvee_{D_i \in \text{WPI}(B)} D_i \vdash \phi \).

To see why Property 1 is violated, let us consider \( B = \{(p \lor q, 0.8)\} \). To infer \((p \lor q, 0.8)\), any WPI \( D \) under previous definition should contain either \((p, 0.8)\) or \((q, 0.8)\). If it contains \((p, 0.8)\), then we do not have \( D \vdash _\pi (p \lor q, 0.8) \).

1. A possibilistic formula of the form \((\phi_1 \land \ldots \land \phi_n, a)\) can be equivalently decomposed into a set of formulas \((\phi_1, a), \ldots, (\phi_n, a)\) due to the min-decomposability of necessity measures.
Property 2 is violated. A weak weighted implicant of a possibilistic knowledge base \( B \) is a possibilistic term \( D = (\ell_1, b_1), \ldots (\ell_k, b_k) \), such that \( D \models B \) such that no two complementary literals exist. To define their notion of WPIs, they simply replace weighted implicants in the definition of WPI given in [Qi et al., 2010] by weak weighted implicant. The following proposition from [Qi et al., 2010] shows that Property 2 holds for the new definition.

**Proposition 1.** \( B^+ \models \phi \iff \forall D_1 \in \text{WPI}(B) \ D_1^+ \models \phi. \)

Unfortunately, this definition does not satisfy the following property, which states that WPIs should be used for compiling a possibilistic knowledge base [Qi and Wang, 2012].

**Property 3:** for any consistent possibilistic knowledge base \( B, B \equiv_s D_{i \in \text{WPI}(B)} D_i. \)

**Example 1.** [Qi and Wang, 2012] Consider \( B = \{(q \lor r, 0.9), (-r, 0.8)\}. \) Any WPI \( D \) of \( B \) must include \((-r, 0.8)\), thus \( q \) must appear in it. Since \( D \) can infer \((q \lor r, 0.9)\), it must contain \((q, 0.9)\). Thus \( D \models (q, 0.9) \) and \( D \) is the only WPI of \( B \). However, we can check that \( B \models (q, 0.8) \).

### 4 Prime Forms in Possibilistic Logic

In this section, we define our notion of prime implicants in possibilistic logic and discuss its properties.

#### 4.1 Prioritized prime implicant

As previously shown, none of the existing definitions of weighted prime implicants can be used to compile a possibilistic knowledge base. Consider again Example 1: \( r \) is not allowed to be in any WPI of \( B \) because \(-r\) must belong to all “reasonable” WPIs of \( B \). This enforces that \((q, 0.9)\) is inferred. Complying to Property 3 entails that there must exist a WPI of \( B \) which contains \((r, 0.9)\). This will then force us to consider \( D = \{(r, 0.9), (-r, 0.8)\} \) as a WPI of \( B \). Now, the question is, since \( \{r, 0.9\}, (-r, 0.8) \) is inconsistent, what will be its logical consequences? It is clear that we cannot use possibilistic inference as \((-r, 0.8)\) will be blocked. Based on our previous discussion, we know that both \((r, 0.9)\) and \((-r, 0.8)\) should be inferred from \( D \). Thus, a paraconsistent semantics should be considered here. This semantics should lead to infer \((q \lor r, 0.9)\) and also \((-r, 0.8)\) from term \( D = \{(r, 0.9), (-r, 0.8)\} \). Clearly, it means that as we use pair \((-r, 0.8) \in D \) to infer \(-r\) with value 0.8, we should not consider \((r, 0.9)\). In more formal terms, the paraconsistent semantics is defined by a paraconsistent \( a \)-cut as follows.

**Definition 4.** Let \( D \) be a possibilistic term. The paraconsistent \( a \)-cut of \( D \) is

\[
D_{\geq a} = \{l | (l, b) \in D, b \geq a, \exists (l', b') \in D \text{ s.t. } a \leq b' < b\}
\]

A possible formula \( (\phi, a) \) is a consequence of \( D \), denoted as \( D \vdash_p (\phi, a) \) if \( D_{\geq a} \models (\phi, a) \).

Our definition of a paraconsistent \( a \)-cut modifies the definition of a \( a \)-cut by considering inconsistent terms. The idea is that, suppose \( l \) and \( l' \) both appear in \( D_{\geq a} \) with \( (l, b) \in D \) and \((l', b') \in D\), where \( b' > b \), then \( l' \) should not be included in the paraconsistent \( a \)-cut \( D_{\leq b} \). Note that \( l' \) is included \( D_{\geq b} \), so it is used to infer formulas whose weights are equal to \( b' \). Consider again Example 1 and term \( D = \{(r, 0.9), (-r, 0.8)\} \). It is the case that \( D \vdash_p (q \lor r, 0.9) \) since \( D_{\geq 0.9} = \{(r, 0.9)\} \) and \( D \vdash_p (-r, 0.8) \) since \( D_{\leq 0.8} = \{(-r, 0.8)\} \).

We then get a new definition of implication form of a possibilistic knowledge base by generalizing the definition of implicant in propositional logic. To avoid confusion of the notations, we call it a prioritized implicant.

**Definition 5.** A prioritized implicant of a possibilistic knowledge base \( B \) is a possibilistic term \( D = \{(l_1, b_1), \ldots (l_k, b_k)\} \), such that \( D \vdash_p (\phi, a) \) for all \( (\phi, a) \in B \), such that there does not exist two complementary literals with the same weight.

According to the definition, a prioritized implicant can be an inconsistent set of possibilistic literals. However, there does not exist a pair of conflicting literals in the form of \( l \) and \( l' \) with the same weight. This requirement is important for two reasons. First, if there are two conflicting literals with the same weight \( a \), then they both will be included in the paraconsistent \( a \)-cut set. Thus, the inference of the \( a \)-cut is trivialized. Second, if this requirement is violated, then the notion of prioritized implicant is not reduced to the notion of implicant.

**Example 2.** (originally from [Qi and Wang, 2012]) Suppose there are four atoms \( p, q, r \) and \( s \), where

- \( p \) represents “red light is on”
- \( q \) represents “green light is off”
- \( r \) represents “press the button”
- \( s \) represents “yellow light is on”

Suppose we have a possibilistic knowledge base \( B = \{(-q \rightarrow r, 0.8), (p \rightarrow -r, 0.7), (q, 0.7), (-s \rightarrow -r, 0.6)\} \) that consists of three uncertain rules. Then \( D = \{(q, 0.8), (-p, 0.7), (s, 0.6)\} \) and \( D' = \{(q, 0.8), (-p, 0.7), (-r, 0.7)\} \) are two prioritized implicants of \( B \).

We now generalize the notion of prime implicant. To define the notion of prioritized prime implicant, we need to take into account of the weights associated with literals. We first stratify the prioritized implicants, and define a lexicographic ordering over prioritized implicants: the intuitive idea is that literals with the highest values should be considered at first. A stratified set \( A \) is the union of sets \( A_1 \cup \ldots \cup A_n \) such that any two elements in \( A_i \) have the same priority and every element in \( A_i \) has higher priority than every element in \( A_j \) with \( i < j \). For two stratified sets \( A = A_1 \cup \ldots \cup A_n \) and \( B = B_1 \cup \ldots \cup B_m \), lexicographic ordering is defined as:

\[
A \preceq_{\text{lex}} B \text{ iff (i) there exists } i \text{ such that } A_i \subseteq B_i \text{ and (ii) for all } 1 \leq j < i, A_j = B_j. \]

We write \( A \preceq_{\text{lex}} B \) if \( A \preceq_{\text{lex}} B \) or \( A = B \).
We consider set inclusion instead of cardinality to define the lexico-
graphic ordering. This is because set inclusion is used to compare two
implicants in propositional logic.

Let us now stratify prioritized implicants. Suppose \( b_i, i = 1, \ldots, m \) are all the distinct weights appearing in \( B \) such that \( b_1 > b_2 > \ldots > b_m \) and \( D \) a prioritized implicant of \( B \). Then \( D = S_1 \cup S_2 \cup \ldots \cup S_m \) where \( S_i = \{ l : (l, b_i) \in D \} \).

Based on this stratification step, we now define prioritized prime
implicant (PPI).

**Definition 6.** A prioritized prime implicant \( D \) of \( B \) is a prior-
itized implicant of \( B \) such that there exists no other prioritized
implicant \( D' \) of \( B \) such that \( D' \triangleright_{lex} D \). We denote by \( PPI(B) \) the
set of all the PPIs of \( B \).

**Example 3.** (Example 2 continued) \( D \) is stratified as
\( A_1 \cup A_2 \cup A_3 \), where \( A_1 = \{(\neg q, 0.8)\} \), \( A_2 = \{(\neg p, 0.7), (q, 0.7)\} \), and \( A_3 = \{(s, 0.6)\} \). Set \( B \) is stratified as \( B_1 \cup B_2 \cup B_3 \), where \( B_1 = \{(\neg q, 0.8)\} \), \( B_2 = \{(\neg p, 0.7), (q, 0.7), (\neg r, 0.7)\} \), and \( B_3 = \emptyset \). Since \( A_1 = B_1 \) and \( A_2 \subset B_2 \), we have \( D \triangleright_{lex} D' \). In fact, \( D \) is a prioritized
prime implicant of \( B \).

### 4.2 Properties of prioritized prime implicants

Let us characterize the behaviour of prioritized prime
implicant. The first proposition shows that PPIs reduce to
propositional prime implicants when the knowledge base is
flat, i.e., every formula in it has weight 1.

**Proposition 2.** Suppose \( B = \{(\phi_i, 1) : i = 1, \ldots, n\} \) then for
any \( D \in PPI(B) \), \( D^* \) is a prime implicant of \( B^* \).

**Proof.** Since all formulas have the same weight, each prior-
itized implicant cannot contain two complementary literals.
Thus, the definition of prioritized implicant is reduced to the
definition of implicant of a classical knowledge base. Simi-
larly, since all formulas have the same weight, the lexico-
graphic ordering is reduced to the ordering defined by set
inclusion. Thus, the definition of prioritized implicant is
reduced to the definition of prime implicant.

The next proposition shows that for any consistent possi-
bilistic knowledge base, there exists at least one consistent
prioritized prime implicant. It enforces Property 1. This
property is a key one since PPI definition tolerates inconsis-
tency.

**Proposition 3.** Let \( B \) be a possibilistic knowledge base. There
exists a consistent prioritized prime implicant \( A \in PPI(B) \)
iff \( B \) is consistent.

**Proof.** Suppose \( A \) is a prime implicant of \( B^* \). Let \( B^{\leq k} = \{(\phi, a) \in B : a = b_k\} \). We first find a minimal subset \( A_1 \) of
\( A \) such that \((B^{\leq k})^*\) is inferred. We then attach weight \( b_k \) to all literals in \( A_1 \). We then find a minimal subset \( A_2 \) of \( A \setminus A_1 \) such that \( A_1 \cup A_2 \) can entail \((B^{\leq k+1})^*\), and we attach weight \( b_{k+1} \) to literals in \( A_2 \), and so on. It is easy to show that \( D \) obtained
in this way is a prioritized prime implicant of \( B \).

Conversely, suppose \( B \) is consistent. By Proposition 2, there
is a prime implicant of \( B^* \) and from it we can construct
a prioritized prime implicant of \( B \) which is consistent.

Finally, we show that Property 2 and Property 3 also hold:
PPI can be used to compile a possibilistic knowledge base.
We first show that \( a \)-cut concept behaves soundly for PPI.

**Lemma 1.** For any possibilistic knowledge base \( B \), we have
\( B_{\geq a} = \{ \{D_i \geq a \mid D_i \in PPI(B) \} \} \) for any \( a \in [0, 1] \).

**Proof.** For any \( D_i \in PPI(B) \), we show \( (D_i)_{\geq a} \vdash (B_{\geq a})^* \).
Suppose \( (D_i)_{\geq a} \) is inconsistent, then this trivially holds. Other-
wise, since \( D_j \vdash (\phi, b) \) for all \( (\phi, b) \in B \), and \( (D_i)_{\geq a} \) is
a prioritized implicant, we have \( (D_i)_{\geq a} \vdash B_{\geq a} \). So \( \{ \{D_i \geq a \mid D_i \in PPI(B) \} \} \vdash B_{\geq a} \).

Conversely, for any prime implicant \( A \) of \( B_{\geq a} \), by Propo-

tion 1 and Proposition 3, we can construct a prioritized prim
implicant \( D \) of \( B \) such that \( A = D_{\geq a} \). Thus \( A \vdash B_{\geq a} \). It
follows that \( A \vdash \{ \{D_i \geq a \mid D_i \in PPI(B) \} \} \). This completes
the proof.

The consequence is that any conclusion from a possibilistic
KB \( B \) can also be inferred from its prime form (and vice-
versa).

**Theorem 1.** For any consistent possibilistic knowledge base
\( B \), \( B \equiv_{D_i \in PPI(B)} D_i \).

**Proof.** We need to show that \( B_{\geq a} = \{ \{D_i \geq a \mid D_i \in PPI(B) \} \} \) for
any \( a \in [0, 1] \). By Lemma 1, we only need to show that for
\( \{ \{D_i \geq a \mid D_i \in PPI(B) \} \} \) for any \( a \in [0, 1] \). Assume that
\( PPI(B) = \{D_1, \ldots, D_n\} \).

Suppose \( \phi \in \{D_i \in PPI(B) \} \geq a \), then there exists
\( \{l_i, a_i\} \subseteq D_i \) such that \( \phi = l_i \land \ldots \land l_n \) and
\( a_i \geq a \) for all \( i \). Thus \( \{D_i \geq a \mid D_i \in PPI(B) \} \).

Conversely, suppose \( \phi \in \{D_i \geq a \mid D_i \in PPI(B) \} \), then
there exists \( \{l_i, a_i\} \subseteq D_i \) such that \( \phi = l_i \land \ldots \land l_n \).

It follows that \( \phi \in \{D_i \geq a \mid D_i \in PPI(B) \} \).

### 4.3 Computing prioritized prime implicants

Let us now detail how we can transform a possibilistic
KB in a set of PPIs. The key idea is to proceed in an incre-
mental way as in [Qi et al., 2010]. Given a possibilistic
KB \( B = \{(\phi_i, a) : i = 1, \ldots, n\} \) and a possibilistic term \( D \) such
that the weight of any literal in \( D \) is greater than \( a \), a
possibilistic term \( D' = \{l_i, a : i = 1, n\} \) is said to be a \( D \-
extended prioritized implicant of \( B \) if \( D \vdash D' \vdash (\phi, a) \), for
any \( (\phi, a) \in B \). We further say that \( D' \) is a \( D \)-extended PPI
of \( B \) if \( D' \) is a \( D \)-extended implicant of \( B \) and there does not
exist another \( D \)-extended implicant of \( B \) such that \( D'' \subset D \).

Let \( B^{\leq k} = \{(\phi, b) \in B : a \geq b_k\} \) and \( B^{= k} = \{(\phi, a) \in B : a = b_k\} \). We give a method for computing all the PPIs
of \( B \). The procedure works as follows. We first compute all
the \( \emptyset \)-extended PPIs of prime implicants of \( B^{= 1} \). This
is achieved by computing all the prime implicants of \( B^{= 1} \)
and attach weight \( b_1 \) to every literal in any of such prime
implicant. Then, for each obtained \( \emptyset \)-extended PPI \( D \) of \( B^{= 1} \), we compute all the \( \emptyset \)-extended PPIs \( D' \) of \( B^{= 2} \), and so on.

Finally, we that Property 2 and Property 3 also hold:
PPI can be used to compile a possibilistic knowledge base.
We first show that \( a \)-cut concept behaves soundly for PPI.

**Definition 7.** \( PI-Ext(B^{= k}) \) is defined by induction as:
1. PI-Ext(B^{-1}) = \{D| D^* is a prime implicant of B^{-1} and every literal in D is attached with weight b_1\};

2. PI-Ext(B^{\leq k}) = \{D_1 \cup D_2|D_1 \in PI-Ext(B^{\leq k-1}) and D_2 is a D_1-extended prime implicant of B^{\leq k}\} for k > 1.

Note that it is necessary to consider B^{\leq k-1} instead of B^{\leq k-1} when we define PI-Ext(B^{\leq k}). Otherwise the method is not syntax-independent. Consider B = \{(q \lor r, 0.9), (\neg r, 0.8)\} and B' = \{(q \lor r, 0.9), (\neg r, 0.8)\}. It is easy to check that B and B' are equivalent according to the possibilistic inference. There are two \(\subseteq\)-extended PPIs of B^{\leq 1}, i.e., D_1 = \{(q, 0.9)\} and D_2 = \{(r, 0.9)\}. So PI-Ext(B^{\leq 1}) = \{D_1, D_2\}. Clearly, PI-Ext(B^{(\leq 1)}) = \{D_1, D_2\}. The D_1-extended PPI of B^{\leq 2} (resp. B^{\leq 2}) is \{(\neg r, 0.8)\} (resp. \{(\neg r, 0.8)\}) and the D_2-extended PPI of B^{\leq 2} (resp. B^{\leq 2}) is \{(\neg r, 0.8), (q, 0.8)\} (resp. \{(\neg r, 0.8), (q, 0.8)\}). Thus, B^{\leq 2} is a prime implicant of \{r, \neg q\}. Formally, we have the following result showing that PI-Ext(B^{\leq k}) is syntax-independent.

Proposition 4. Suppose B \equiv_s B' and there are n distinct weights appearing in B. For each 1 \leq k \leq n, we have PI-Ext(B^{\leq k}) = PI-Ext(B^{\leq k}).

The proof of Proposition 4 is easy to see because B \equiv_s B' infers that B^{\leq k} \equiv_s B'^{\leq k}.

The following proposition shows that the above method actually computes all the prioritized implicants of B. We denote by PPI(B^{\leq k}) the set of all the prioritized implicants of B^{\leq k}.

Proposition 5. For any possibilistic term D, D \in PI-Ext(B^{\leq k}) iff D \in PPI(B^{\leq k}).

Proof. For the “Only if” direction. We show it by induction over k.

When k = 1. Suppose D \in PI-Ext(B^{=1}). Then D^* is a prime implicant of (B^{=1})^*. Thus, D is a prioritized prime implicant of B^{=1}.

Assume that the proposition holds for k = 1. Suppose D \in PI-Ext(B^{\leq k}). Then D = D_1 \cup D_2, where D_1 \in PI-Ext(B^{\leq k-1}) and D_2 is a D_1-extended prioritized prime implicant of B^{\leq k}. Since D_1 \in PI-Ext(B^{\leq k-1}), by assumption, D_1 \in PPI(B^{\leq k-1}). Thus, D_1 \vdash_p (\phi, a) for all (\phi, a) \in B^{\leq k-1}. Since D \in PI-Ext(B^{\leq k}), D \vdash_p (\phi, b_k) for all (\phi, b_k) \in B^{\leq k}. So D \vdash_p (\phi, a) for all (\phi, a) \in B^{\leq k}.

That is, D is a prioritized implicant of B^{\leq k}. Assume D \notin PPI(B^{\leq k}). Then there exists D' \in PPI(B^{\leq k}) such that D' \prec_{lex} D. Suppose D (resp. D') is stratified as S_1 \cup \ldots \cup S_n (S'_1 \cup \ldots \cup S'_n) as in Definition 5. Then there exists i such that S'_i \subseteq S_i and S'_i = S_i for all j < i. i can only be k as D_k \in PPI(B^{\leq k-1}). Thus S'_k \subseteq S_k and S_k = S'_k for all j < k. This means D' = D_1 \cup D_2', where (D'_2)'' = S'_k and the weight of literals in D'_2 is equal to the weight of literals in D_2. We have D'_2 \subseteq D_2. However, D' \in PPI(B^{\leq k}) implies that D' \vdash_p (\phi, b_k) for all (\phi, b_k) \in B^{\leq k}. This contradicts the fact that D \in PI-Ext(B^{\leq k}) as D_2 \subseteq D_2.

For the “If direction”. We also show it by induction over k.

For k = 1. Suppose D \in PPI(B^{=1}). Then D^* is a prime implicant of (B^{=1})^*. Thus, D is a \(\subseteq\)-extended prioritized prime implicant of B^{=1}.

Assume that the proposition holds for k = 1. Suppose D \in PPI(B^{=k}). Let D = D_1 \cup D_2, where D_1 = \{(l, a) \in D|a \geq b_{k-1}\} and D_2 = \{(l, a) \in D|a = b_k\}. We can show that D_1 is a prioritized prime implicant of B^{\leq k} by induction over k. Since D \in PPI(B^{\leq k}), D \vdash_p (\phi, a) for all (\phi, a) \in B^{\leq k}. So D_2 is a D_1-extended prioritized implicant of B^{\leq k}. Suppose on the contrary that D \not\in PI-Ext(B^{\leq k}). Then we can find a D_1-extended prime implicant D'_2 of B^{\leq k} such that D'_2 \subseteq D_2. Then D_1 \cup D'_2 \vdash_p (\phi, a) for all (\phi, a) \in B^{\leq k}. This contradicts with the fact that D \in PPI(B^{\leq k}). As D \prec_{lex} D'.

This proposition also stresses up that the set of prioritized prime implicants is unique.

Example 4. (Example 2 continued) There are three distinct weights in B. We have B^{=1} = \{(q \lor r, 0.8), B^{=2} = \{(q \lor r, 0.7), (\neg p \lor \neg r, 0.7), (q, 0.7)\} and B^{=3} = \{(q \lor r, 0.6), (\neg p \lor \neg r, 0.6), (q, 0.6), (s \lor r, 0.6)\}. It is easy to see that PI-Ext(B^{=1}) = \{D_1, D_2\}, where D_1 = \{(q, 0.8)\} and D_2 = \{(r, 0.8)\}. D_1-extended prime implicants of B^{\leq 2} are (\neg p, 0.7) and (\neg r, 0.7) and D_2-extended prime implicants of B^{\leq 2} are (\neg p, 0.7), (q, 0.7) and (\neg r, 0.7), (q, 0.7). So PI-Ext(B^{\leq 2}) = \{D_3, D_4, D_5, D_6\}, where

D_3 = \{(q, 0.8), (\neg p, 0.7)\},
D_4 = \{(q, 0.8), (\neg r, 0.7)\},
D_5 = \{(r, 0.8), (\neg p, 0.7), (q, 0.7)\} and
D_6 = \{(r, 0.8), (\neg r, 0.7), (q, 0.7)\}.

Finally, D_3-extended prime implicants of B^{\leq 3} are \{(s, 0.6)\} and \{(\neg r, 0.6)\}, D_5-extended prime implicants of B^{\leq 3} are \{(s, 0.6)\} and \{(\neg r, 0.6)\}, there is no D_4-extended prime implicant of B^{\leq 3} and D_6-extended prime implicant of B^{\leq 3}. Thus PPI(B^{=1}) = PI-Ext(B^{\leq 3}) = \{D_3', D_4', D_4, D_5', D_5, D_6\}, where

D_3' = \{(q, 0.8), (\neg p, 0.7), (s, 0.6)\},
D_4' = \{(q, 0.8), (\neg r, 0.7), (\neg p, 0.6)\},
D_5' = \{(r, 0.8), (\neg p, 0.7), (q, 0.7)\} and
D_6' = \{(r, 0.8), (\neg r, 0.7), (q, 0.7), (\neg r, 0.6)\}.

5 Conclusion and Future Work

In this paper, we considered the problem of defining prime forms of a possibilistic knowledge base. Existing definitions of prime forms of a possibilistic knowledge base are not desirable because they cannot be used to recover the possibilistic knowledge base. Our study shows that we have to drop the common assumption of a prime form of a formula or a knowledge base, i.e., a prime implicant or its generalization is a consistent formula. We defined a prioritized implicant of a possibilistic knowledge base as a set of weighted literals that may be inconsistent and provided a paraconsistent semantics for it. The notion of a prioritized prime implicant is then defined by considering a lexicographic ordering. We
show that our new definition is desirable by showing that prioritized prime implicants show some desirable properties. Finally, we presented a method for computing prioritized prime implicants of a possibilistic knowledge base.

Prime forms of a possibilistic knowledge base play an important role in knowledge management in possibilistic logic. The have been used to define revision operators and merging operators in possibilistic logic (see [Qi and Wang, 2012] and [Liu et al., 2006]). However, it is nontrivial to apply our new definition of prime forms of a possibilistic knowledge base to define a revision operator a merging operator in possibilistic logic. The revision operators defined in [Qi and Wang, 2012] is based on a distance function between two weighted prime implicants. However, a prioritized prime implicant can be an inconsistent set of possibilistic literals, and it is not clear how to define a distance function between two inconsistent set of possibilistic literals. We will leave this problem as future work.

Acknowledgement
Guilin Qi is partially supported by NSFC (61272378) and Doctoral Discipline Foundation for Young Teachers in the Higher Education Institutions of Ministry of Education (No. 20100092120029).

References


Modal uncertainty logics with fuzzy neighborhood semantics

Ricardo Oscar Rodriguez
Departamento de Computación, FCEyN - UBA, 1428 Buenos Aires, Argentina

Lluis Godo
IIIA - CSIC, Campus de la UAB s/n, 08193 Bellaterra, Spain

Abstract
In the classical setting neighborhood structures are the standard semantic tool used for non-normal modal logics. In this paper we introduce a new fuzzy version of this semantics and we prove that it is a natural generalization in the sense that fuzzy Kripke models can be identified as a special kind of our proposed models. In addition, we characterize this notion in a syntactic way and describe conditions on fuzzy neighborhood models sufficient for the validation of different and well known schemes. Finally, we show how this approach can be used to provide an axiomatization of a many-valued modal system capturing possibilistic reasoning over Łukasiewicz logic propositions.

Keywords: many-valued modal logic, fuzzy modal logic, fuzzy neighborhood semantics.

1 Introduction
In approximate reasoning, sometimes one needs to simultaneously deal with both fuzziness of propositions and modalities, for instance one may try to assign a degree of truth to propositions like “John is possibly tall” or “John is necessarily tall”, where “John is tall” is considered as a fuzzy proposition. In this sense, extensions of fuzzy logic systems can be considered as a suitable tool to model not only vagueness but also other kinds of information features like certainty, belief or similarity, which have a natural interpretation in terms of modalities. Moreover, although notions of vagueness (at propositional level) and uncertainty are not the same, there are close links between them and in many occasions they need to live together. For example, as mentioned in [9], if all we know is that “John is tall” (i.e. a vague knowledge about John’s height) then, about the (Boolean) truth of the sentence “John’s height is 1.80 m”, one can only say that it is more or less possible. More formally, Dubois and Prade in [10] propose to understand each fuzzy assertion of the sort of “X is tall” (where tall is a fuzzy subset of a domain U and X is a variable taking values in U) as a constraint on the unknown possibility of the crisp assertions X = x, with x ∈ U, of the form Π(X = x) ≤ μ_{tall}(x). This example makes it clear that vague, incomplete information also produces a form of uncertainty.

Therefore, it is natural to consider a combination of many-valued logics and modal logics in order to be capable of dealing with uncertainty and vagueness in the same representation language. Thus, we have to face the problem of searching a syntactical characterization of many-valued modal logics that works in most of the cases. Unfortunately, the well known Kripke semantics does not work out well because in many cases the K axiom is not valid, and it is not known a general method to axiomatize many-valued modal logics given by those semantics. Indeed, it turns out that the only minimal logics axiomatized in the literature are the ones where the base many-valued logic is the one corresponding to a finite Heyting algebra [11; 12], the standard (infinite) Gödel algebra [5] or a finite residuated algebra [2] (in particular finite Łukasiewicz linearly ordered algebras).

In order to overcome this difficulty we propose an alternative semantics which is a generalization of the classical neighborhood semantics, whose main ideas are recalled in Section 2. Hence, we understand modal many-valued logics as logics defined by neighborhood frames (possibly with a many-valued neighborhood function) where each world follows the rules of a many-valued logic, this many-valued logic being the same for every world. The reader will find the details of this approach in Section 3. In Section 4, we point out some more semantical considerations. In Section 5, we introduce several many-valued modal calculus, whereas in Section 6 contains completeness results of all those calculus with respect to an associated subclass of neighborhood models for each one. In Section 7, we study under which conditions this new semantics converges to a Kripke semantics. Finally, in Section 8, as an application we focus on a modal formalization of possibilistic reasoning over fuzzy propositions of a finitely-valued Łukasiewicz logic. We end up with some conclusions and future work.

2 Classical Neighborhood Semantics
For some applications, the relational Kripke semantics for modal logics is too strong. For example, suppose that the intended interpretation of □ϕ is “ϕ is assigned ‘high’ probability”, where “high” probability means above a certain

1This example is taken from the notes of course Neighborhood Semantics for Modal Logic: An Introduction, May 12 - 17, ESSLLI 2007, by Eric Pacuit
threshold \( \alpha \in [0, 1] \). Under this interpretation, the formula \( \square \varphi \land \square \psi \rightarrow \square (\varphi \land \psi) \) is not valid. To see this, suppose that \( \varphi \) and \( \psi \) represent probabilistically independent events, and suppose that the threshold is \( \alpha = \frac{3}{4} \). Suppose that both \( \varphi \) and \( \psi \) hold true (i.e., their probabilities are above the threshold). However, the probability of \( \varphi \land \psi \) is \( \frac{9}{16} \), which is below the threshold. Thus, \( \square (\varphi \land \psi) \) does not hold. Therefore, the formula \( \square \varphi \land \square \psi \rightarrow \square (\varphi \land \psi) \) is not valid under this particular interpretation.

However, as is well-known, \( \square \varphi \land \square \psi \rightarrow \square (\varphi \land \psi) \) is a valid formula under Kripke semantics (it holds true in any Kripke model). This kinds of situations have led to see for weaker semantics. Perhaps the best known of these is the so-called neighborhood semantics. The key idea of this semantics has a topological flavour: each world \( w \) in a model is associated with a collection of subsets of worlds, the neighborhood of \( w \). The key idea of this semantics relates each world \( w \) to a set of subsets of \( W \), that is, \( N : W \rightarrow \varphi (\varphi (W)) \).

The interpretation of boxed formulas is then as follows:

\[
\forall w \models \square \varphi \Leftrightarrow \{ v \in W | v \models \varphi \} \in N(w),
\]

and the dual definition for diamonds is:

\[
\forall w \models \Diamond \varphi \Leftrightarrow \{ v \in W | v \notmodels \varphi \} \notin N(w).
\]

Neighborhood semantics is a generalization of Kripke semantics, i.e. given any Kripke model \( \mathcal{M} = (W, R, e) \) we can define a neighborhood model \( N^\mathcal{M} = (W, N^\mathcal{M}, e) \) by stipulating for each \( w \in W \) that \( N^\mathcal{M}(w) \) is the (singleton) set of \( R \)-accessible worlds from \( w \). Thus, we can turn any Kripke model into an equivalent neighborhood model.

In fact, neighborhood semantics falsifies many principles valid in Kripke semantics, as for instance the axiom \( K \), or the necessitation inference rule for \( \square \). Indeed, all that remains in the basic neighborhood semantics is the weaker principle of logical equivalence preservation: \( \models \varphi \Leftrightarrow \psi \) then \( \models \square \varphi \Leftrightarrow \square \psi \). However, by imposing restrictions on the neighborhood function, it is possible to recover all those principles one by one. For example, demanding that neighborhood be closed by supersets, the axioms \( \square (\varphi \land \psi) \rightarrow (\square \varphi \land \square \psi) \) becomes valid. Alternatively, forcing that neighborhood be closed under intersections then our original formula \( \square \varphi \land \square \psi \rightarrow \square (\varphi \land \psi) \) becomes valid. In particular, superset closed neighborhood models that in addition satisfy the condition \( \bigwedge_{w' \in W} N(w') \in N(w) \) for every world \( w \) are called augmented and they agree (i.e. they are in 1-1 relationship) with the Kripke semantics.

### 3 General Framework

In this section we provide the definition of the modal many valued logics \( \text{Log}_{\mathbb{M}}(A, N) \) associated with an algebra of truth-values \( A \) and a class of \( A \)-valued Neighborhood frames \( N \).

All algebras considered in this paper will be residuated lattices. An algebra \( A = (A, \land, \lor, \top, \bot, \rightarrow, 1, 0) \) is a residuated lattice if and only if the reduct \( (A, \land, \lor, 1, 0) \) is a bounded lattice with maximum 1 and minimum 0 (its order is denoted by \( \leq \), the reduct \( (A, \cdot, 1) \) is a commutative monoid, and the fusion operation \( \odot \) (sometimes also called the intensional conjunction or strong conjunction) is residuated, with \( \Rightarrow \) being its residual; that is, for all \( a, b, c \in A \)

\[
a \odot b \leq c \iff a \leq b \Rightarrow c
\]

In addition, we will require the residuated lattices to be complete, that is, algebras where all suprema and infima (even of infinite subsets of the domain) exist. It is well known that complete residuated lattices satisfy the law

\[
a \odot \bigvee_{i \in I} b_i = \bigvee_{i \in I} (a \odot b_i)
\]

for arbitrary sets of indices \( I \).

We stress that these requirements are not very strong since a lot of well-known classes of algebras in the algebraic logic setting satisfy them, for instance, complete \( FL \)-algebras \cite{16} and complete \( BL \)-algebras \cite{14}. Hence, in particular we can consider that \( A \) is any of the three basic continuous \( t \)-norm algebras: Lukasiewicz algebra \( [0, 1]_L \), product algebra \( [0, 1]_P \) and Gödel algebra \( [0, 1]_G \).

The language of the logic \( \text{Log}_{\mathbb{M}}(A, N) \) is, by definition, the propositional language generated by a set \( \text{Var} \) of propositional variables\(^2\) together with the connectives given by the algebraic signature of \( A \), i.e. \( \land, \lor, \neg, \Rightarrow, \top \) and \( \bot \) (the latter also denoted \( \bot \) and \( \bot \), expanded with a new unary operator: the necessity operator \( \square \). The set of formulas of the resulting language will be denoted by \( \text{Fm}_{\mathbb{M}} \).

An \( A \)-valued Neighborhood frame is a pair \( \mathfrak{F} = (W, N) \) where \( W \) is a set (of worlds) and \( N \) is an \( A \)-valued binary function \( N : F(W) \times W \rightarrow A \), where \( F(W) = A^W \), i.e. the set of mappings \( f : W \rightarrow A \), called neighborhood function. Whenever \( A \) is fixed, we will denote by \( \text{Fr} \) the class of all \( A \)-valued Neighborhood frames.

**Definition 1** An \( A \)-valued Neighborhood model is a triple \( N = (W, N, e) \) where \( (W, N) \) is an \( A \)-valued Neighborhood frame and \( e : \text{Var} \times W \rightarrow A \) is a map, called valuation, assigning to each variable in \( \text{Var} \) and each world in \( W \) an element of \( A \). The map \( e \) can be uniquely extended to a map \( \overline{e} : \text{Fm}_{\mathbb{M}} \times W \rightarrow A \) satisfying that:

- \( \overline{e} \) is an algebraic homomorphism, in its first component, for the connectives in the algebraic signature of \( A \), and
- \( \overline{e}(\square \varphi, w) = N(\mu_{\varphi, w}) \), where \( \forall w' \in W, \mu_{\varphi, w'} = e(\varphi, w') \).

The function \( N \) determines for each world \( w \) and for formula \( \varphi \) the degree of necessity of \( \varphi \) at \( w \). Note that we make no assumptions about the nature of \( N \). Also, although the functions \( e \) and \( \overline{e} \) are different there will be no confusion between them, and so sometimes we will use the same notation \( e \) for both.

Following the Boolean modal case \cite{7, 6}, the notion of satisfiability in a model is formalized as follows:

\(^2\)In most cases it is assumed that \( \text{Var} = \{ p_0, p_1, p_2, \ldots \} \).

\(^3\)By abusing the notation, we shall use the same symbols for denoting both connectives in the language and in the algebra

\(^4\)Later on we will provide some hints about how to develop these ideas with the possibility operator \( \diamond \).
Definition 2 Let \( w \) be a world in a neighborhood model \( N = (W, N, e) \) then:

\[
(N, w) \models \varphi \text{ iff } e(\varphi, w) = 1.
\]

In particular, note that \((N, w) \models \varphi \iff e(\varphi, w) \leq e(\psi, w)\).

The notions of a formula being valid in a model and in a class of models is as usual.

Definition 3 A formula \( \varphi \) is valid in a model \( N \), written \( N \models \varphi \), iff for every world \( w \) in \( N \) it holds that \((N, w) \models \varphi \). A formula \( \varphi \) is valid in a class of models \( C \), written \( \models_C \varphi \), if it is valid in every model \( N \in C \).

Now we are ready to introduce the modal many-valued logic \( \text{Log}_{\varphi}(A, N) \). It is defined as the set of formulas \( \varphi \in \text{Fm}_{\varphi} \) satisfying that for every \( A \)-valued Neighborhood model \( \langle W, N, e \rangle \) over a frame \( \langle W, N \rangle \) in \( N \) and for every world \( w \) in \( W \), it holds that \( e(\varphi, w) = 1 \).

Remark 1 For the sake of simplicity in this paper we restrict ourselves to adding the necessity operator \( \square \), but analogously we could have considered a possibility operator \( \Diamond \) by taking a Neighborhood bi-modal model as \( \langle W, N, P, e \rangle \) and by ruling the truth condition as:

\[
e(\Diamond \varphi, w) = P(\mu_\varphi, w).
\]

where \( P : F(W) \times W \mapsto A \).

We stress that for the case that \( A \) is the Boolean algebra of two elements \( \{0, 1\} \), all previous definitions correspond to the standard terminology in the field of modal logic (cf. [7]). As far as the authors know, this extension to the modal many-valued setting is new.

Let us close this section with the following theorem, which states that validity in a class of neighborhood models is preserved by a congruence rule:

Theorem 1 Let \( N \) be a class of \( A \)-valued neighborhood models. Then:

\[
\models N \varphi \iff \models N \square \varphi \iff \models N \Diamond \psi.
\]

Proof: Suppose that \( N \) is a class of \( A \)-valued models such that \( \models N \varphi \iff \models N \square \varphi \iff \models N \Diamond \psi \) so that for any world \( w \) in any model \( N \) in \( N \), \( e(\varphi, w) = e(\psi, w) \) which means that \( \mu_\varphi = \mu_\psi \). Hence, for any world \( w \) in any model \( N \) in \( N \), \( e(\square \varphi, w) = e(\Diamond \psi, w) \) and then \( \models N \square \varphi \iff \models N \Diamond \psi \).

In the rest of this paper, we will assume that an underlying algebra \( A \) is fixed.

4 More semantical considerations

One of the main slogans from the main text on modal logic [1] is to describe the modal language as a language for talking about graphs, or relational structures. The key idea is that some modal formulas can be shown to define interesting properties of the accessibility relation in a Kripke frame. Similarly, in the current setting, modal formulas can be understood as expressing properties of the neighborhood function. In this section we pay attention to special and interesting classes of neighborhood frames which are defined by formulae. It is worth noticing that neighborhood semantics are easier to adapt to axioms than relational Kripke semantics, i.e., in general, given an axiom it is usually possible to find the property on the neighborhood function covering it.

Let us consider the following schemes:

\[
M^{(2)}_2 \quad \Box(\varphi \lor \psi) \rightarrow (\Box \varphi \lor \Box \psi),
\]

\[
C^{(2)}_2 \quad (\Diamond \varphi \lor \Box \psi) \rightarrow \Box(\varphi \lor \psi).
\]

\[
N \quad \Box \top,
\]

\[
K \quad \Box(\varphi \lor \psi) \rightarrow (\Box \varphi \lor \Box \psi).
\]

The following theorem states that each of these schemes has a counterexample in a neighborhood semantic.

Theorem 2 None of the schemes \( M^{(2)}_2, C^{(2)}_2, N \) and \( K \) is valid in the class of all neighborhood models.

As it has been earlier indicated, we can obtain subclass of neighborhood frames \( \langle W, N, e \rangle \) by putting conditions over the function \( N \). Some of them may be the following:

\[
m^{(2)}_2(N(\mu_\varphi \circ \mu_\psi, w) \leq N(\mu_\varphi, w) \circ N(\mu_\psi, w)).
\]

\[
c^{(2)}_2(N(\mu_\varphi, w) \circ N(\mu_\psi, w) \leq N(\mu_\varphi \circ \mu_\psi, w)).
\]

\[
(n) \quad N(\mu_T, w) = 1.
\]

for every world \( w \). Here when we write \( \mu_\varphi \circ \mu_\psi \) we mean function resulting from the pointwise application of the \( \circ \) operation of the algebra \( A \), that is, for every \( w \in W \), \( (\mu_\varphi \circ \mu_\psi)(w) = \mu_\varphi(w) \circ \mu_\psi(w) \).

Depending on whether the function \( N \) in a \( A \)-valued neighborhood frame satisfies conditions \( m^{(2)}_2 \), \( c^{(2)}_2 \) or \( (n) \), we say that the frame is \( \circ \)-distributive, is closed under \( \circ \), or contains the unit, respectively. When a model satisfies the first two conditions we say that it is quasi-filter. When all three properties are met, we call the frame to be a filter.

Theorem 3 Given an algebra \( A \), the schemas \( M^{(2)}_2, C^{(2)}_2 \) and \( N \) are valid in the subclasses of \( A \)-valued neighborhood frames that are \( \circ \)-distributive, closed under \( \circ \), and contain the unit, respectively.

We can also consider other schemas and their respective subclass of \( A \)-valued neighborhood frames. For example:

\[
T^{(2)}_\Box \quad \Box \varphi \rightarrow \varphi.
\]

\[
0^{(2)}_\Box \quad \neg \Box \top.
\]

\[
4^{(2)}_\Box \quad \Box \varphi \rightarrow \Box \Box \varphi.
\]

\[
D^{(2)}_\Box \quad \Box \Box \varphi \rightarrow \Box \varphi.
\]

\[
E^{(2)}_\Box \quad \neg \Box \varphi \rightarrow \neg \Box \varphi.
\]

\[
R^{(2)}_\Box \quad \neg \neg \Box \varphi \rightarrow \neg \Box \varphi.
\]

None of these is neither valid in the class of all \( A \)-valued neighborhood frames. But we can identify correspondence subclasses of \( A \)-valued neighborhood frames that validate them. Indeed, consider the following conditions on a \( A \)-valued neighborhood frame \( N = (W, N) \), for every world \( w \) and membership function \( \mu_\varphi \) with \( \varphi \) in \( Fm_\Box \):

\[
(t^{(2)}_\Box) \quad N(\mu_\varphi, w) \leq \mu_\varphi(w).
\]

\[
i^{(2)}_\Box \quad N(\mu_T, w) = 0.
\]

\[
(i^{(2)}_\Box) \quad N(\mu_\varphi, w) \leq N(\mu_{\lnot \varphi}, w).
\]

\[
d^{(2)}_\Box \quad N(\mu_{\varphi \circ \psi}, w) \leq N(\mu_\varphi, w).
\]

\[
e^{(2)}_\Box \quad N(\mu_{\varphi \circ \psi}, w) \leq N(\mu_\varphi, w).
\]

\[
r^{(2)}_\Box \quad N(\mu_{\lnot \varphi}, w) \leq N(\mu_\varphi, w).
\]

where \( 0 \) denotes the null constant function and \( \circ \) the complement function in the algebra \( A \). Depending on whether the
function \( N \) in a \( \mathbf{A} \)-valued neighborhood frame satisfies conditions \((t_2)\) to \((r_3)\) above, we say that the model is reflexive, null-preserving, transitive, dense, euclidean or reverse-euclidean, respectively.

**Theorem 4** The following statements hold:

1. \( T_\square \) is valid in the subclass of \( \mathbf{A} \)-valued neighborhood frames satisfying the condition \((t_2)\).
2. \( 0_\square \) is valid in the subclass of \( \mathbf{A} \)-valued neighborhood frames satisfying the condition \((z_2)\).
3. \( A_\square \) is valid in the subclass of \( \mathbf{A} \)-valued neighborhood frames satisfying the condition \((iv_\square)\).
4. \( Ds_\square \) is valid in the subclass of \( \mathbf{A} \)-valued neighborhood frames satisfying the condition \((ds_\square)\).
5. \( iE_\square \) is valid in the subclass of \( \mathbf{A} \)-valued neighborhood frames satisfying the condition \((e_\square)\).
6. \( R_\square \) is valid in the subclass of \( \mathbf{A} \)-valued neighborhood frames satisfying the condition \((r_\square)\).

## 5 Fuzzy Classical Systems

As usual we present different kinds of logical systems such that each of them will determine a different subfamily of fuzzy neighborhood models. Let \( \mathcal{A} \) be a fixed axiomatic calculus for a given algebra \( \mathbf{A} \). We consider the formal systems \( \mathcal{A}_\square \) on the language \( \text{Fm}_\square \) which are obtained by adding to \( \mathcal{A} \) the following rule:

\[ \text{RE}_\square: \quad \text{From } \varphi \leftrightarrow \psi \text{ infer } \square \varphi \leftrightarrow \square \psi \]

We call this class of many-valued modal systems “fuzzy classical”. The smallest fuzzy classical system will be called \( \mathcal{A}E_\square \). In addition, \( \vdash_{\mathcal{A}E_\square} \) will express theoremhood in these logics. Proof with assumptions will be allowed, with the restriction that \( \text{RE}_\square \) is to be applied to theorems only. For the sake of simplicity, we will also consider a fuzzy system \( \mathcal{A}_\square \) as the set of all its theorems, i.e., \( \mathcal{A}_\square = \{ \psi \mid \vdash_{\mathcal{A}_\square} \psi \} \). If \( \vdash_{\mathcal{A}_\square} \varphi \) will express that there is such a proof of \( \varphi \) with assumptions from the set \( T \). Note that, by Theorem 1, \( \text{RE}_\square \) preserves validity in any given neighborhood model.

We are going to introduce three further classes of fuzzy modal logics: monotonic, regular and normal. For this reason, we are interested in logics having the rules:

\[ \text{RM}_\square: \quad \text{From } \varphi \rightarrow \psi \text{ infer } \square \varphi \rightarrow \square \psi \]
\[ \text{RR}_\square: \quad \text{From } (\varphi \circ \theta) \rightarrow \psi \text{ infer } (\square \varphi \circ \square \theta) \rightarrow \square \psi \]

Then, a fuzzy classical system is:

- **monotonic** if it is closed under \( \text{RM}_\square \).
- **\( \circ \)-regular** if it is closed under \( \text{RR}_\square \).
- **normal** if it is regular and it contains the scheme \( \mathbf{N} \).

We denote the smallest fuzzy monotonic system by \( \mathcal{A}M_\square \), the smallest fuzzy \( \circ \)-regular system by \( \mathcal{A}R_\square \) and the smallest fuzzy normal system by \( \mathcal{A}K_\square \).

It is easy to prove that:

- Every fuzzy monotonic system is classical.
- Every fuzzy regular is monotonic and hence classical.
- Every fuzzy normal is regular and hence monotonic and classical.

In particular, then, the smallest systems \( \mathcal{A}E_\square, \mathcal{A}M_\square, \mathcal{A}R_\square, \) and \( \mathcal{A}K_\square \) are increasingly inclusive. In fact, these inclusions are proper.

## 6 Completeness Results

In this section, fixed an underlying algebra \( \mathbf{A} \), we connect fuzzy classical systems and \( \mathbf{A} \)-valued neighborhood models by way of weak completeness theorems. In order to prove weak completeness we will reduce the problem to pure underlying \( \mathbf{A} \) logic, and then we are going to define a \( \mathbf{A} \)-canonical model with the property that for any formula \( \varphi \) such that \( \forall \mathbf{A}_\square \varphi \), there exists a world \( w \) in the model which assigns a value less than 1 to \( \varphi \).

Now we define the \( \mathbf{A} \)-canonical model. Let \( \square \text{Fm}_\square = \{ \square \theta : \theta \in \text{Fm}_\square \} \) be the set of formulas in \( \text{Fm}_\square \) which start with the connective \( \square \). Then any formula in \( \text{Fm}_\square \) may be seen as a formula of the propositional \( \mathbf{A} \)-language built from the extended set of propositional variables \( X = \text{Var} \cup \square \text{Fm}_\square \) by means of the \( \circ, \land, \forall, \Rightarrow, \bot \) connectives. That is, we may consider the formulas in \( \square \text{Fm}_\square \) as additional propositional variables for the underlying \( \mathbf{A} \) logic.

**Definition 4** **Canonical model**: \( \mathbf{N}^*_\square = (\mathcal{W}^*, \mathcal{N}^*, e^*) \) where

- The set of worlds \( \mathcal{W}^* \) will consist of those valuations \( v : \text{Var} \cup \square \text{Fm}_\square \rightarrow \mathbf{A} \) which satisfy \( \pi(\mathcal{A}_\square) = 1 \) when extended to \( \pi : \text{Fm}_\square = \text{Fm}(\text{Var} \cup \square \text{Fm}_\square) \rightarrow \mathbf{A} \) according to the \( \mathbf{A} \) interpretation of \( \circ, \land, \forall, \Rightarrow, \bot \).

- The neighborhood function is given by
  \[ e^*(\mu, v) = v(\varphi). \]

- The valuation associated to the world \( v \) will be \( v \upharpoonright \text{Var} \).
  That is, \( e^*(p, v) = v(p) \) for any \( p \in \text{Var} \).

For the sake of simplicity, we will write from now on \( v(\varphi) \) for \( \pi(\varphi) \). It is clear that \( \mathbf{N}^*_\square \) belongs to the class of \( \mathbf{A} \)-valued neighborhood models.

**Lemma 1** **(Truth Lemma)** For any world \( v \) in the canonical model \( \mathbf{N}^*_\square \) and any formula \( \varphi \),

\[ e^*(\varphi, v) = v(\varphi). \]

**Proof**: This is proved by induction in the complexity of \( \varphi \) seen again as a formula of \( \text{Fm}_\square = \text{Fm}(\text{Var} \cup \square \text{Fm}_\square) \) The atomic step and the inductive steps for the \( \mathbf{A} \) connectives being straightforward, it is enough to verify inductively \( e^*(\varphi, \psi) = v(\varphi) \). But it is obvious from both the second point of Definition 4 and the definition of \( \mathbf{A} \)-valued neighborhood model at Section 2.

**Theorem 5** The logic \( \mathcal{A}E_\square \) is sound and weak complete with respect to the class of all neighborhood frames.

**Proof**: The proof is standard and so will only be sketched. Soundness is straightforward. For weak completeness, the proof is by contraposition. Suppose that it is the case that \( \forall \mathcal{A}E_\square \varphi \). Then, by completeness of the underlying logic \( \mathcal{A} \), there is a valuation \( v : \text{Var} \cup \square \text{Fm}_\square \rightarrow \mathbf{A} \) such that \( v(\varphi) < 1 \). Hence, by lemma 1, \( \varphi \) is not valid in the canonical model, because \( (\mathbf{M}^*_\square, v) \not\models \varphi. \)

---

IJCAI-13 Workshop on Weighted Logics for Artificial Intelligence (WL4AI-2013)

82
In general, to prove the completeness of a modal system with respect to a class of models it is sufficient to show that the canonical model for this system is contained in the class. In this sense we give the next result.

**Theorem 6** Let $N^*_{\mathcal{A}}$ be the canonical model for the system $\mathcal{A}$. Then:

1. $N^*_{\mathcal{A}}$ is $\circ$-distributive if $\mathcal{A}$ contains $M_{\circ}^0$.
2. $N^*_{\mathcal{A}}$ is closed under $\circ$ if $\mathcal{A}$ contains $C_{\circ}^0$.
3. $N^*_{\mathcal{A}}$ contains the unit if $\mathcal{A}$ contains $N$.
4. $N^*_{\mathcal{A}}$ is reflexive if $\mathcal{A}$ contains $T_{\circ}$.
5. $N^*_{\mathcal{A}}$ is null preserving if $\mathcal{A}$ contains $0_{\circ}$.
6. $N^*_{\mathcal{A}}$ is transitive if $\mathcal{A}$ contains $4_{\circ}$.
7. $N^*_{\mathcal{A}}$ is dense if $\mathcal{A}$ contains $D_{\circ}$.
8. $N^*_{\mathcal{A}}$ is euclidean if $\mathcal{A}$ contains $E_{\circ}$.
9. $N^*_{\mathcal{A}}$ is reverse-euclidean if $\mathcal{A}$ contains $R_{\circ}$.

**Proof:** It is trivial. Only it is necessary to apply the definitions.

Moreover, consider the following further properties of $\mathcal{A}$-valued neighborhood frames $\mathfrak{G} = (W,N)$:

- $\mathfrak{G}$ is monotonic if $N(\mu_x,w) \leq N(\mu_{\psi},w)$ for all $w \in W$, whenever $\varphi \rightarrow \psi$ is valid in $\mathfrak{G}$.
- $\mathfrak{G}$ is $\circ$-regular if $N(\mu_{\varphi},w) \cap N(\mu_{\psi},w) \leq N(\mu_{\chi},w)$ for all $w \in W$, whenever $\varphi \circ \chi \rightarrow \psi$ is valid in $\mathfrak{G}$.
- $\mathfrak{G}$ is normal if it is $\circ$-regular and for all $w \in W$, $N(\mu_{\top},w) = 1$.

**Theorem 7** The logics $\mathcal{A}M_{\circ}, \mathcal{A}K_{\circ}$ and $\mathcal{A}K_{\circ}$ are sound and weak complete with respect to the subclass of monotonic, $\circ$-regular and normal neighborhood frames, respectively.

7 Relation between $\mathcal{A}$-valued neighborhood and $\mathcal{A}$-valued Kripke models

Another well-known semantics for modal systems is the one based on Kripke frames and Kripke models. The notion of $\mathcal{A}$-valued Kripke models is as follows.

**Definition 5** A $\mathcal{A}$-Kripke model ($\mathcal{A}$-model) is a structure $\mathcal{M} = (W,S,e)$ where:

1. $W$ is a non-empty set of objects that we call worlds of $\mathcal{M}$.
2. $S : W \times W \rightarrow \mathcal{A}$ is an arbitrary function $(x,y) \mapsto S(x,y)$.
3. $e : Var \times W \rightarrow \mathcal{A}$ is an arbitrary function $(p,x) \mapsto e(p,x)$.

The evaluations $e(-,x) : Var \rightarrow \mathcal{A}$ are extended simultaneously to all formula in $FM_{\mathcal{A}}$ by defining inductively at each world $x$: $e(\bot,x) := 0.$ $e(\varphi \circ \psi,x) := e(\varphi,x) \circ e(\psi,x).$ $e(\varphi \land \psi,x) := e(\varphi,x) \land e(\psi,x).$ $e(\varphi \lor \psi,x) := e(\varphi,x) \lor e(\psi,x).$

The notions of a formula $\varphi$ being true at a world $x$, valid in a model $\mathcal{M} = (W,S,e)$, or universally valid, are the usual ones:

- $\varphi$ is true in $\mathcal{M}$ at $x$, written $\mathcal{M} \models_{\circ} \varphi$, iff $e(\varphi,x) = 1$.
- $\varphi$ is valid in $\mathcal{M}$, written $\mathcal{M} \models \varphi$, iff $\mathcal{M} \models_{\circ} \varphi$ at any world $x$ of $\mathcal{M}$.
- $\varphi$ is $\mathcal{A}$-valid, written $\models_{\mathcal{A}} \varphi$, if it is valid in all the $\mathcal{A}$-models.

We will write $C_{\mathcal{A}}$ to denote the class of all $\mathcal{A}$-valued Kripke models.

**Observation 1** When $\mathcal{A} = [0,1]$ is the Łukasiewicz standard algebra in real unit interval $[0,1]$, then the logic $\mathcal{A}$ corresponds to the well-known infinitesimal Łukasiewicz logic $\mathcal{L}$. Then the following schemata are not valid in the class $C_{\mathcal{L}}$:

$\mathcal{M}_{\circ} : \Box(\varphi \circ \psi) \rightarrow (\Box \varphi \circ \Box \psi)$

$\mathcal{C}_{\circ} : (\Box \varphi \circ \Box \psi) \rightarrow (\Box \varphi \circ \Box \psi)$

$\mathcal{K} : \square(\varphi \rightarrow \psi) \rightarrow (\Box \varphi \rightarrow \Box \psi)$

**Theorem 8** The following schemata are valid in the class $C_{\mathcal{A}}$ for any residuated lattice $\mathcal{A}$:

$\mathcal{M}_{\circ} : \Box(\varphi \land \psi) \rightarrow (\Box \varphi \land \Box \psi)$

$\mathcal{C}_{\circ} : (\Box \varphi \land \Box \psi) \rightarrow (\Box \varphi \land \Box \psi)$

$\mathcal{K} : \Box(\varphi \rightarrow \psi) \rightarrow (\Box \varphi \rightarrow \Box \psi)$

The reader will have noticed that the difference between a $\mathcal{A}$-valued neighborhood model and a $\mathcal{A}$-valued Kripke model is due to the functions $N$ and $S$. It should be clear that with neighborhood models, there is more freedom in which collection of sets can be necessary at a particular state. On the other hand, in Kripke models this information is presented in a simple and elegant fashion. A natural question to ask is under what circumstances a neighborhood model and a relational model represent the same information or satisfy the same formulae.

The next theorem shows the embedding of the class of $\mathcal{A}$-valued Kripke models into a particular subclass of $\mathcal{A}$-neighborhood models.

**Theorem 9** For every $\mathcal{A}$-Kripke model $\mathcal{M} = (W,S,e)$ there is a pointwise equivalent $\mathcal{A}$-neighborhood model $N = (W,N,S,e)$ in the sense that for any world $x \in W$ and any formula $\varphi \in Fm_{\mathcal{A}}$:

$\bar{e}_{\mathcal{M}}(\varphi,x) = \bar{e}_{\mathcal{N}}(\varphi,x)$

**Proof:** We have only to define the neighborhood function in the following way:

$N_{\mathcal{S}}(\mu_{\varphi},x) = \inf_{y \in W} \{S(x,y) \Rightarrow \mu_{\varphi}(y)\}$

and to check the equality is satisfied. The details are left to the reader.
This particular class of $A$-valued neighborhood models should be defined in a such way as it is suggested by the last proof.

**Definition 6** A $A$-valued neighborhood model $N = ⟨W, N, e⟩$ is augmented if and only if for each $x$ in $W$, there exists a function $f_x ∈ A^W$ such that for any formula $φ ∈ FM_Ł$:

$$N(μ_φ, x) = \inf_{y ∈ W} [f_x(y) ⇒ μ_φ(y)]$$

**Theorem 10** The class of $A$-valued Kripke models is isomorphic to the class of $A$-augmented neighborhood models. Both classes are isomorphic in the sense that for each model in one of them there exists another model in the other class such that all formulae are satisfied with the same degree in both models.

**Proof:** One direction is given by Theorem 9. For the converse direction, we have only to define $S(x, y) = f_x(y)$ where $f_x$ is as postulated in Definition 6.

A much more interesting relationship between $A$-valued neighborhood models and $A$-valued Kripke models is the following one. Let $N = ⟨W, N, e⟩$ be a $A$-valued neighborhood model. Then we define its associated $A$-valued Kripke model as $M_N = ⟨W, S^N, e⟩$ where:

$$S^N(x, y) = \inf_{φ ∈ FM_Ł} \{N(μ_φ, x) ⇒ μ_φ(y)\}$$

In general, it is neither the case that $S = S^{N_Ł}$ nor $N = N_S$. However, there are some interesting cases where the last definition agree with the one used in the proof of Theorem 10. For instance, if $A$ is the standard MV-algebra $[0, 1]_Ł$ then, by using the continuity of its residuum, it is easy to prove it. Another interesting example is taking $A$ as the standard Gödel algebra $[0, 1]_G$ and $S$ is optimal in the sense of [5]. In fact, we are able to generalise that result in the following way.

**Definition 7** Given a $A$-valued Kripke-model $M = ⟨W, S, e⟩$, define a new accessibility relation as follows:

$$S^+(x, y) = \inf_{φ ∈ FM_Ł} \{e(□φ, x) ⇒ e(φ, y)\}$$

Call $M$ optimal whenever $S^+ = S$.

The following lemma shows that any Kripke-model is equivalent to an optimal one.

**Lemma 2** The model $M^+ = ⟨W, S^+, e⟩$ is optimal. Moreover, if $e^+$ is the extension of $e$ in $M^+$, then $e^+(φ, x) = e(φ, x)$ for any $x ∈ W$.

**Proof:** The first claim follows from the second (which implies $S^+ = S^+$), and the second is proven by induction on the complexity of formulas. The only non trivial step is that of the modal connectives. Notice first that $S(x, y) ≤ S^+(x, y)$, because $e(□φ, x) ≤ (S(x, y) ⇒ e(φ, y))$ thus $S(x, y) ≤ (e(□φ, x) ⇒ e(φ, y))$. Now, assume $e^+(φ, x) = e(φ, x)$ for all $y$, then by the previous observation and the induction hypothesis: $e^+(□φ, x) = \inf_y \{S^+(x, y) ⇒ e^+(φ, y)\} ≤ \inf_y \{S(x, y) ⇒ e(φ, y)\} = e(□φ, x)$. But $S^+(x, y) ≤ (e(□φ, x) ⇒ e(φ, y))$ by definition of $S^+$ and thus $e(□φ, x) ≤ (S^+(x, y) ⇒ e(φ, y)) = (S^+(x, y) ⇒ e^+(φ, y))$ which yields $e(□φ, x) ≤ e^+(□φ, x)$.

**Corollary 1** Let $M = ⟨W, S, e⟩$ and $N_S = ⟨W, N_S, e⟩$ be a Kripke model and its associated neighborhood model, respectively. Then for any $φ ∈ FM_Ł$:

$$\models M φ ⇔ \models N_S φ$$

Let $N_S = ⟨W, N, e⟩$ and $M^N = ⟨W, S^N, e⟩$ be an augmented neighborhood model and its associated Kripke model, respectively. Then for any $φ ∈ FM_Ł$:

$$\models N φ ⇔ \models M^N φ$$

This proves that the class of $A$-valued augmented neighborhood models validates the same set of formulas that the class of $A$-valued Kripke models. In fact, like in the classical case, every $A$-valued augmented neighborhood model is essentially an $A$-valued Kripke model, and vice versa. Furthermore, that result may be easily extended to subclasses satisfying additional properties like: reflexivity, transitivity, $⊙$-distributivity, closed under $⊙$, etc.

**8 Modal formalizations of possibilistic reasoning on fuzzy propositions**

In the last section, we have introduced the notion of augmented neighborhood semantics. A first question could be: why is this subclass of neighborhood models interesting? First, we have to note that a particular class of augmented neighborhood models is equivalent to (in the sense the previous section) the class of possibilistic models $(W, π, e)$ (see below). This case is very interesting and relevant for us because their logical approaches are usually introduced as modal extensions of Łukasiewicz logic which do not satisfy the $K$ axiom (see e.g. [2]) and, hence, their proofs of completeness do not admit a canonical technique. Then, augmented neighborhood models become a natural general semantics for reasoning with possibilistic measures of uncertainty (see e.g. [8; 13]).

Indeed, the semantics of possibilistic logic over Łukasiewicz logic (c.f. [13]) is based on structures $(W, π, e)$ where $e$ assigns to each world a $[0, 1]$-Łukasiewicz valuation, $π : W \mapsto [0, 1]$ is a possibility distribution, and the interpretation of necessity operator is as follows:

$$e(□φ, x) := \inf_{y ∈ W} \{π(y) ⇒ Ł e(φ, y)\}$$

where $⇒_Ł$ is Łukasiewicz implication function, i.e. $a ⇒_Ł b = \min(1, 1 - a + b)$ for all $a, b ∈ [0, 1]$.

The associated augmented neighborhood model $(W, N, e)$ equivalent to $(W, π, e)$ is obtained by defining:

$$N(e(φ, x), x) := \inf_{y ∈ W} \{π(y) ⇒ Ł e(φ, y)\}$$

This augmented neighborhood model, together with canonicity results in Theorem 6, allows us to generalize the result given in [13] in the sense that, unlike [13], we are able to consider here a general language with nested modalities (see also [15] for another axiomatization using another approach). For that, we need to restrict our approach to finitely-valued interpretations, since this allows us to get completeness with respect to augmented models. We consider the
The next step is to define a filtration with respect to a set of formulae \( \Phi \) which is closed under sub-formulae.

The \( \Phi \)-filtration of \( N' = \{ W, N, e \} \) through \( \Phi \) is the neighborhood model \( N^* = \{ W^*, N^*, e^* \} \) defined as follows:

1. \( W^* = \{ [w] \mid w \in W \} \), where \( [w] \) denotes the equivalence class of \( w \) with respect to \( \Phi \).
2. \( e^* \) is defined as the original \( e \), restricted to the atomic formulae in \( \Phi \) and the worlds in \( W^* \). Thus, \( e(\tau, w) = r \) for each truth-value \( r \), and \( e^*(p, [w]) = e(p, w) \) if \( p \) is a propositional variable in \( \Phi \) and \( w \) is a world in \( W^* \). Besides, \( e^* \) is truth functional with respect to Łukasiewicz logic connectives \( \rightarrow, \neg \).

3. For every \( w \) in \( N \) and for every formula \( \Box \phi \in \Phi \), \( N^*(\mu^*_o, [w]) = N(\mu_o, w) \), where \( \mu^*_o([w']) = e^*(\phi, [w']) \) for every \([w'] \in W^*\).

We can now state and prove the fundamental theorem related to the filtration:

**Proposition 1** Suppose \( N' = \{ W, N, e \} \) is any model, \( \Phi \) is any set of formulae which is closed under sub-formulae, and \( N^* = \{ W^*, N^*, e^* \} \) is any filtration of \( N' = \{ W, N, e \} \) through \( \Phi \). Then for every formula \( \phi \in \Phi \) and for every \( w \in W \), \( e^*(\phi, [w]) = e(\phi, w) \).

**Proof:** The proof is by induction on the complexity of a modal formula \( \phi \in \Phi \). If \( \phi \) is a propositional variable, the theorem holds by the definition of \( e^* \) in a filtration. The relevant induction step is when \( \phi = \Box \theta \). But then, by condition 3 above of the above definition we simply have, for every \( w \in W \), \( e(\Box \theta, w) = N(\mu_o, w) = N^*(\mu_o, [w]) = e^*(\Box \theta, [w]) \).

Now, we come back to our canonical model \( N^*_o = \{ W^*, N^*, e^* \} \) such that \( e^*(\phi, [w]) < 1 \). Let \( N^*_r = \{ W^*, N^r, e^r \} \) be the filtration of \( N^*_o \) through the set \( \Phi_r \) of subformulas of \( \phi \), as specified in Definition 8. Then we consider a modified neighbourhood model, defined as \( N^o = \{ W^*, N^o, e^o \} \) where \( \forall w \in W, N^o([w]) = N^*(\mu_o, [w]) = e^*(\Box \theta, [w]) \).

On the other hand, if \( A = \{ 0, \frac{1}{n}, \ldots, \frac{n-1}{n}, 1 \} \), each function \( f : A^{W^*} \to A \) can be written as:

\[
f = \bigwedge_{[w] \in W^*} \overline{1 - f([w])} \Rightarrow [w]^{c}
\]

where \( \overline{1 - f(w)} \) denotes the constant function of value \( 1 - f(w) \), and \( [w]^c : W^* \to A \) is the characteristic function of the complement of the singleton function \( \{ [w] \} \), i.e. \( [w]^c(x) = 1 \) if \( x \neq [w] \) and \( [w]^c([w]) = 0 \). Next, for \( \mu^*_o \), by applying the axioms of our calculus we obtain that

\[
N^o(\mu^*_o, [w]) = \bigwedge_{[w] \in W^*} \overline{1 - \mu^*_o([w])} \Rightarrow L N^o([w^c], v).
\]

Now, by defining \( \pi([w]) = 1 - N^o([w^c]) \), the above expression can be rewritten as

\[
N^o(\mu^*_o, v) = \bigwedge_{[w] \in W^*} \pi([w]) \Rightarrow L \mu^*_o([w])
\]

Then, the possibilistic completeness is a consequence of Corollary 1 since the neighborhood model \( N^o \) is actually an

---

3Recall that \( e(\neg \phi, w) = 1 - e(\phi, w) \).

4\( N^* \) is well defined, since if \( [w] = [w'] \), by definition of the equivalence relation, it holds that \( e(\Box \phi, w) = e(\Box \phi, w') \), i.e. \( N(\mu_o, w) = N(\mu_o, w') \).
augmented model associated to the optimal possibilistic ones. Note that, in fact, the neighborhood function $N^0$ is independent of $w$. In addition, it is worth to remark that $\neg \Box \perp$ is a theorem in our system and it guarantees that $\pi$ is a normalized possibility distribution on $W^*$. Indeed:

$$1 = e^* (\neg \Box \perp, [w]) = 1 - \inf_{[w'] \in W^*} \{ \pi([w']) \Rightarrow e^*(\perp, [w']) \} = \sup_{[w'] \in W^*} \pi([w']) .$$

In addition, since $N^0$ is finite, the previous result determines that the logic is decidable.

9 Conclusions and future work

In this paper we have addressed the study of fuzzy modal logics in the context of $A$-valued neighborhood semantics. In particular, one of the main goals was to get an axiomatization of a modal logic over infinitely-valued Łukasiewicz logic to capture possibilistic semantics, done in Section 8. However, a lot of problems are left open. The followig is a collection of what are in our opinion are the main open questions concerning the framework discussed in this paper:

- The first one is the development of a general theory where $\Box$ and $\Diamond$ are simultaneously in the language.
- Is it possible to provide strongly complete axiomatizations for some $\mathcal{A}_A$ systems? Which ones? For instance, if $\vdash_A$ is not strongly complete then it is well known that there is not a such result. On the other hand, if the underlying algebra $A$ is finite, then it seems to be feasible.
- How can we include constants for all the truth values in the language?
- The question on the decidability and complexity of $\mathcal{A}_E^\Box$ and its extensions is also left unanswered since these logics do not seem to have the finite model property under $A$-valued neighborhood semantics.
- How can we characterize those $A$-valued neighborhood frames equivalent to $A$-valued Kripke frames?
- Other possible steps will be to study the proposed logics with other techniques: algebraic semantics\(^1\), sequent calculus, etc.
- Finally, in this paper we have addressed as an application the axiomatization of a logic of generalised necessity measures for fuzzy propositions. It remains to study other generalised uncertainty logics (probabilistic, DS-belief, etc.) in the framework of other classes of neighborhood models.

Acknowledgments

The authors are grateful to the anonymous reviewers for their helpful comments. They also acknowledge partial support by the FP7 PIRSES-GA-2009-247584 project MaToMUVI. Besides, Rodriguez has been supported by Argentinian project PIP CONICET 112-200801-02543 2009-2012 and PICT-PRH 263, and Godo by the Spanish CONSOLIDER- INGENIO 2010 project Agreement Technologies (CSD2007-00022) and the MINECO project EdcTRI (TIN2012-39348-C02-01).

\(^1\)Recall that equivalence preserving expansions of algebraizable logics are algebraizable [4].

References


Scale reasoning with fuzzy-$\mathcal{EL}^+$ ontologies based on MapReduce

Zhangquan Zhou\textsuperscript{1} and Guilin Qi\textsuperscript{1} and Chang Liu\textsuperscript{2} and Pascal Hitzler\textsuperscript{3} and Raghava Mutharaju\textsuperscript{3}

\textsuperscript{1}Southeast University, China
\{quanzz, gqig\}@seu.edu.cn
\textsuperscript{2}Shanghai Jiao Tong University, China
liuchang@apex.sjtu.edu.cn
\textsuperscript{3}Wright State University, United States
{pascal.hitzler, mutharaju.2}@wright.edu

Abstract

Fuzzy extension of Description Logics (DLs) allows the formal representation and handling of fuzzy or vague knowledge. In this paper, we consider the problem of reasoning with fuzzy-$\mathcal{EL}^+$, which is a fuzzy extension of EL+. We first identify the challenges and present revised completion classification rules for fuzzy-$\mathcal{EL}^+$ that can be handled by MapReduce programs. We then propose an algorithm for scale reasoning with fuzzy-$\mathcal{EL}^+$ ontologies using MapReduce. Some preliminary experimental results are provided to show the scalability of our algorithm.

1 Introduction

The Web Ontology Language OWL, which is essentially based on the description logics, has been designed as one of the major standards for formal knowledge representation and automated reasoning in Semantic Web. OWL 2 EL based on description logic $\mathcal{EL}^+$, a restricted language of OWL 2, stands out for its positive complexity results and the sufficient expressive power for many real ontologies, such as the medical ontology Snomed-CT.

However, description logics are not able to represent fuzzy information, which is available in some applications, such as multimedia and bioinformatics. Fuzzy extension of description logics has been proposed using fuzzy sets and fuzzy logics [Klir and Yuan, 1995] to provide more expressive power. One of the challenging problems of fuzzy description logics is reasoning with large scale fuzzy ontologies. Such ontologies can be extracted from different sources, such as multimedia (see [Dalakleidi et al., 2011]).

Parallel reasoning is an obvious choice to easily achieve the scalability goal. There have been some works covering it. One of the most successful attempts is WebPIE [Urbani et al., 2010], an efficient inference engine for large amount of RDF triples under $pD^*$ semantics [ter Horst, 2005] using MapReduce framework. This work is further extended in [Liu et al., 2011] to handle fuzzy knowledge. In [Mutharaju et al., 2010] a parallel classification algorithm using MapReduce is given for classical $\mathcal{EL}^+$. However, this algorithm is not optimized for implementation and cannot handle reasoning in fuzzy ontologies. A concurrent method based on multi-core system is discussed in [Kazakov et al., 2011] to reason with classical $\mathcal{EL}$ ontologies, which makes use of multiple cores and the implemented system, ELK performs well in reasoning with large ontologies. A parallel reasoner for $\mathcal{ALC}$ is introduced in [Wu and Haarslev, 2012], which is a tableau-based description logic reasoner.

In this paper, we consider a fuzzy extension of $\mathcal{EL}^+$, called fuzzy-$\mathcal{EL}^+$, which is introduced in [Stoilos et al., 2008]. Although a polynomial time algorithm is given to classify fuzzy-$\mathcal{EL}^+$ ontologies, no experimental evaluation is reported in that work. In order to provide scalable reasoning in fuzzy-EL+, we consider using MapReduce. We first identify the difficulties and challenges to do fuzzy-$\mathcal{EL}^+$ classification using MapReduce framework. We then revise the completion fuzzy-$\mathcal{EL}^+$ rules that can be handled by MapReduce programs and implement a prototype system. We provide some experimental evaluations and prove that our algorithm can scale to fuzzy-EL+ ontologies.

2 Preliminaries

2.1 fuzzy-$\mathcal{EL}^+$

fuzzy-$\mathcal{EL}^+$ is a fuzzy extension of the description logic $\mathcal{EL}^+$, which is introduced in [Stoilos et al., 2008]. Concepts in fuzzy-$\mathcal{EL}^+$ are defined according to the following grammar:

\[ C, D ::= \top | A | C \sqcap D | \exists r.C \]

where $A$ ranges over the set of concept names (CN) and $r$ over the set of role names (RN). A fuzzy-$\mathcal{EL}^+$ ontology is a finite set of fuzzy general concept inclusions (F-GCIs) of the form $(C \sqsubseteq D, n)$, where $n \in \{0, 1\}$, and role inclusions (RIs) of the form $r_1 \circ \ldots \circ r_k \sqsubseteq s$, where $k$ is a positive integer. Note that the role inclusions axioms are not fuzzified in [Stoilos et al., 2008].

A polynomial algorithm is given to perform classification of fuzzy-$\mathcal{EL}^+$ ontologies, i.e., it computes all fuzzy subsumptions between concepts of the input ontology $O$. The algorithm first transforms the given ontology $O$ into normal form, where all concept inclusions are one of the forms:

\[ (A_1 \sqcap \ldots \sqcap A_k \sqsubseteq B, n) \]
\[ (A \sqsubseteq \exists r.B, n) \]
\[ (\exists r.B \sqsubseteq A, n) \]
and all role inclusions are of the form \( r_1 \circ r_2 \subseteq s \) or \( r \subseteq s \).

The normalization can be done in linear time [Stoilos et al., 2008]. In the following, we assume that an input ontology \( \mathcal{O} \) is in normal form.

The algorithm is formulated by two mappings \( S \) and \( R \), where \( S \) ranges over subsets of \( CN \times [0,1] \) and \( R \) over subsets of \( CN \times CN \times [0,1] \). Intuitively, \( \langle B, n \rangle \in S(A) \) implies \( A \sqsubseteq B \) and \( \langle A, B, n \rangle \in R(r) \) implies \( A \sqsubseteq \exists r.B \) and \( S(A) \) and \( R(r) \) are initialized as follows:

\[
S(A) = \{ (A, 1), (\top, 1) \}, \text{ for each class name } A \text{ in the input ontology } \mathcal{O}.
\]

\[
R(r) = \emptyset, \text{ for each role name } r \text{ in } \mathcal{O}.
\]

Then the two sets \( S(A) \) and \( R(r) \) are extended by applying the completion rules in Table 3 until no more rules can be applied.

This algorithm runs in polynomial time and it is sound and complete [Stoilos et al., 2008], i.e., after termination on the given ontology \( \mathcal{O} \), \( A \sqsubseteq B \) if and only if \( \langle B, m \rangle \in S(A) \) holds, where \( n, m \in [0,1] \) and \( m \geq n \).

### 2.2 An example of fuzzy-\( ELC^+ \) reasoning

We use an example to illustrate the procedure of fuzzy-\( ELC^+ \) reasoning. A simple medical ontology is given in Table 1 and it is already in normalized form. \( \alpha_1, \alpha_4 \) are F-GCI axioms. They express that an elbow joint is a joint (\( \alpha_1 \)) and has location in elbow (\( \alpha_2 \)), a joint is a part of body (\( \alpha_3 \)), a stuff which is a part of elbow is also a part of arm (\( \alpha_4 \)). Each F-GCI axiom has a fuzzy value. The last axiom (\( \alpha_5 \)) is a RI axiom which means has-location is more specific than is-part-of.

In Table 2 we show how the consequence subsumptions (ElbowJoint \( \sqsubseteq \) PartOfArm, 0.8) (corresponding to (7)) and (ElbowJoint \( \sqsubseteq \) isPartOf.Body, 0.6) (corresponding to (4)) can be derived using the reasoning rules in Table 3.

### 2.3 MapReduce

MapReduce is a programming model for parallel processing over huge data sets [Dean and Ghemawat, 2004]. A MapReduce task consists of two main phases: map phase and reduce phase. Several tasks complete a MapReduce job which solves a specific problem.

In map phase, a user-defined map function receives a key/value pair and outputs a set of key/value pairs. All the pairs sharing the same key are grouped and passed to reduce phase. Then a user-defined reduce function is set up to process the grouped pairs. The outcome of reduce nodes may be the results of the overall job or the intermediate input of following tasks. The grouping procedure between map and reduce phase is called shuffle which is the key factor to determine the efficiency of a task. The functionalities of map and reduce nodes can be formulated as:

**Map:** (key1, value1) \( \mapsto \) list(key2, value2),  
**Reduce:** (key2, list(value2)) \( \mapsto \) list(value3).

We give an example on how to use MapReduce programs to apply a rule. We use the previous example and consider the rule R2 in Table 3 on \( \alpha_3 \) and (3). In a naive reasoning program, the map function first scans the dataset. When the axiom \( \langle \text{Joint} \sqsubseteq \text{isPartOf.Body}, 0.6 \rangle \) is scanned, it generates a key/value pair (key='Joint', value={\{Joint \sqsubseteq \text{isPartOf.Body}, 0.6\}}). When (Joint, 0.9) \( \in S(\text{ElbowJoint}) \) is scanned, it generates a pair (key='Joint', value={\{Joint, 0.9 \} \in S(\text{ElbowJoint})}). Then the reduce function processes outputs of map function that share the same key. It iterates the collected values and computes the fuzzy value. Finally the reduce function generates the consequence subsumption (ElbowJoint \( \sqsubseteq \) isPartOf.Body, 0.6).

Here are some principles for designing an efficient MapReduce program:

- Always keep in mind to make full use of the throughout capacity of clusters.
- Do not increase the burden of shuffle phase unless necessary.

### Table 1: A simple medical ontology

| \( \alpha_1 \) | ElbowJoint | \( \sqsubseteq \) | Joint | 0.9 |
| \( \alpha_2 \) | ElbowJoint | \( \sqsubseteq \exists \) | hasLocation, Elbow | 0.8 |
| \( \alpha_3 \) | Joint | \( \sqsubseteq \exists \) | isPartOf.Body | 0.6 |
| \( \alpha_4 \) | \exists \) | isPartOf.Elbow | \( \sqsubseteq \) | PartOfArm | 0.8 |
| \( \alpha_5 \) | hasLocation | \( \sqsubseteq \) | isPartOf |

### Table 2: A fragment of reasoning

<table>
<thead>
<tr>
<th>Rule</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>( \langle \text{ElbowJoint}, 1 \rangle \in S(\text{ElbowJoint}) )</td>
</tr>
<tr>
<td>(2)</td>
<td>( \langle \text{Elbow}, 1 \rangle \in S(\text{Elbow}) )</td>
</tr>
<tr>
<td>(3)</td>
<td>( \langle \text{Joint}, 0.9 \rangle \in S(\text{ElbowJoint}) )</td>
</tr>
<tr>
<td>(4)</td>
<td>( \langle \text{ElbowJoint.Body}, 0.6 \rangle \in R(\text{isPartOf}) )</td>
</tr>
<tr>
<td>(5)</td>
<td>( \langle \text{ElbowJoint.Elbow}, 0.8 \rangle \in R(\text{hasLocation}) )</td>
</tr>
<tr>
<td>(6)</td>
<td>( \langle \text{ElbowJoint.Elbow.0.8} \rangle \in R(\text{isPartOf}) )</td>
</tr>
<tr>
<td>(7)</td>
<td>( \langle \text{PartOfArm.0.8} \rangle \in S(\text{ElbowJoint}) )</td>
</tr>
</tbody>
</table>

### Table 3: Completion rules for fuzzy-\( ELC^+ \)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>If ( \langle A_1, n_1 \rangle \in S(X), ..., \langle A_i, n_i \rangle \in S(X), \langle A_1 \sqcap ... \sqcap A_i \sqsubseteq B, k \rangle \in O ) and ( \langle B, m \rangle \notin S(X) ), then ( S(X) := S(X) \cup (B, m) ), where ( m = \min(n_1, ..., n_i, k) )</td>
</tr>
<tr>
<td>R2</td>
<td>If ( \langle A, n \rangle \in S(X), \langle A \sqsubseteq \exists r.B, k \rangle \in O ), and ( \langle B, m \rangle \notin S(X) ), then ( R(r) := R(r) \cup (B, m) ), where ( m = \min(n, k) )</td>
</tr>
<tr>
<td>R3</td>
<td>If ( \langle X, Y, n_1 \rangle \in R(r), (A, n_2) \in S(Y), \langle r.A \sqsubseteq B, n_3 \rangle \in O ), and ( \langle B, m \rangle \notin S(X) ), then ( S(X) := S(X) \cup (B, m) ), where ( m = \min(n_1, n_2, n_3) )</td>
</tr>
<tr>
<td>R4</td>
<td>If ( \langle X, Y, n \rangle \in R(r), r \subseteq s \in O ), and ( \langle X, Y, n \rangle \notin S(s) ), then ( S(s) := S(s) \cup (X, Y, n) )</td>
</tr>
<tr>
<td>R5</td>
<td>If ( \langle X, Y, n_1 \rangle \in R(r), (Y, Z, n_2) \in R(s), r \subseteq s \subseteq t \in O ), and ( \langle X, Z, n \rangle \notin S(t) ), then ( R(t) := R(t) \cup (X, Z, n) ), where ( m = \min(n_1, n_2) )</td>
</tr>
</tbody>
</table>
Completion Rule For MapReduce

R1

R5

R2

R3

R4

R5

Multiple joints. In Table 3, the rules R1, R3 and R5 have multiple joints (more than one) in their preconditions. As mentioned above, these rules can be seen as multi-way joins (for example R3 can be formulated as a 3-way join $\mathcal{R} \bowtie \mathcal{S} \bowtie \mathcal{K}$). It is easy for MapReduce to handle 2-way joins like the example in section 2.3 but not for multi-way joins. In our case, R2 and R4 can be directly handled by MapReduce programs and the remaining rules need to be modified. We give revised fuzzy-$\mathcal{EL}^+$ rules in Table 5 and in following sections, we will discuss why we adopt these modifications.

3.2 Handling R1

The rule R1 handles the concept conjunction inclusion (CCI) axioms. A CCI axiom is in the form of $\langle A_1 \cap \ldots \cap A_i \sqsubseteq B, n \rangle$, where $i \geq 1$. The application of R1 is a complex multi-way join on tables $\mathcal{S}$ and $\mathcal{H}$ and it is not intuitive to split the multi-way join into several 2-way joins.

In order to handle R1 using MapReduce programs, we first introduce a function $I$ and a mapping $T$. The function $I$ assigns each CCI axiom in input ontology $\mathcal{O}$ an integer that is used as the identifier of the axiom. To create $T$, we first set $T(X)$ as 0 for each $X$ in $\mathcal{N}$. Then for every CCI axiom like $\langle A_1 \cap \ldots \cap A_i \sqsubseteq B, n \rangle$, each concept $A_j (i \leq j \leq l)$ occurring in the left side of $B$ is checked and its corresponding set $T(A_j)$ is extended as:

$$T(A_j) := T(A_j) \cup \{i, b, l, n\}$$

where $i = I(\langle A_1 \cap \ldots \cap \mathcal{A}_j \sqsubseteq B, n \rangle)$.

In this way, $\langle A_1 \cap \ldots \cap A_i \sqsubseteq B, n \rangle$ can be replaced by $\langle i, b, l, n \rangle \in T(A_1), \ldots, \langle i, b, l, n \rangle \in T(A_i)$.

We get $T$ and $I$ before reasoning and use a new mapping $P$ to split R1 into R1-1 and R1-2 in Table 5. $\langle i, A_j, b, l, m \rangle \in P(X)$ means that $X$ is subsumed by $A_j$ with the fuzzy value $m$ and $A_j$ occurs in the concept conjunction of the axiom.

The tradeoff between these principles lead us to design and optimize our algorithms in following work.
Table 6: Effects of intermediate results

<table>
<thead>
<tr>
<th>ontology</th>
<th>( \mathcal{Q}_{\text{Rois}} )</th>
<th>( \mathcal{Q}_{\text{RoisK}} )</th>
<th>( \mathcal{Q}_{\text{SoidK}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>GALEN</td>
<td>no. of tuples/M</td>
<td>utilization ratio</td>
<td>total cost time of R3/minutes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>163</td>
<td>1.61%</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>1638</td>
<td>1.36%</td>
<td>580</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>12.81%</td>
<td>27</td>
</tr>
</tbody>
</table>

\( \langle A_1 \cap \ldots \cap A_j \cap \ldots \cap A_l \subseteq B, n \rangle \). \( P(X) \) contains the intermediate or incomplete derived information that is used only in R1-2 to complete the work of R1. The length of conjunction \( l \) is recorded in \( P(X) \) and used in R1-2 to check whether all concepts in conjunction \( \langle A_1 \cap \ldots \cap A_l \rangle \) are collected.

If we transform the mappings \( T \) and \( P \) to tables \( T \) and \( P \), the application of R1 can be represented by a sequence of two 2-way joins, respectively \( S \propto T \) and \( P \propto P \), which can be easily handled by MapReduce programs, i.e., we can use two MapReduce tasks to handle R1-1 and R1-2. The keys \( A, i \) in the left side of R1-2 means that \( A \) and \( i \) are both used to construct the key.

### 3.3 Handling R3

As mentioned in section 3.1, the application of R3 can be seen as a 3-way join on tables \( R, S \) and \( K \), formulated as:

\[ R(r, X, Y, n_1) \propto S(Y, A, n_2) \propto K(r, A, B, n_3) \]

This 3-way join can be completed by two 2-way joins because the two joins in the 3-way join can be done sequentially, i.e., we can first join two of the three tables and then join the intermediate result with the third one. Since any two tables share a common joint, there are three join orders to split the 3-way join into two steps as follows:

1. \( R(r, X, Y, n_1) \propto S(Y, A, n_2) \propto K(r, A, B, n_3) \)
2. \( R(r, X, Y, n_1) \propto K(r, A, B, n_3) \propto S(Y, A, n_2) \)
3. \( S(Y, A, n_2) \propto K(r, A, B, n_3) \propto R(r, X, Y, n_1) \)

The distinctions in performance among those three orders are just decided by the intermediate join results, because in the first hand, the tables \( R, S \) and \( K \) will be read once and joined with other tables in reduce function, thus the overheads these three tables contribute is in equality (formulated as \(|R| + |S| + |K|\)). In the second hand, the three join orders will output the same results, so the overheads contributed by the final results are also in equality. We use tables \( \mathcal{Q}_{\text{Rois}}, \mathcal{Q}_{\text{RoisK}} \) and \( \mathcal{Q}_{\text{SoidK}} \) to respectively denote the intermediate results of \( \text{Rois} \), \( \text{RoisK} \) and \( \text{SoidK} \) (namely the results of first joins in brackets). We did an experiment to investigate the overheads contributed by \( \mathcal{Q}_{\text{Rois}}, \mathcal{Q}_{\text{RoisK}} \) and \( \mathcal{Q}_{\text{SoidK}} \). The experimental results on GALEN are listed in Table 6.

In this experiment, we accumulate the number of tuples of \( \mathcal{Q}_{\text{Rois}}, \mathcal{Q}_{\text{RoisK}} \) and \( \mathcal{Q}_{\text{SoidK}} \) generated in all iterations of reasoning. The total size of the three tables is listed in first row. The second row shows the utilization ratio of the three tables, which denotes the percentage of the tuples used in second joins. The total cost times of R3 are collected in the third row.

From the experiment results, we find that \( \mathcal{Q}_{\text{SoidK}} \) has the smallest size and the highest utilization ratio, and its corresponding join order costs minimal time compared to the other two orders. We have the same results on other ontologies.

We give a brief analysis here. The size of \( \mathcal{Q}_{\text{SoidK}} \) is \( |\mathcal{Q}_{\text{SoidK}}| \), where \( p \) is the probability of two tuples from \( S \) and \( K \) agreeing on their common joint. Since \( S \propto K \) is on the joint \( A \), namely the named concepts, \( p \) can be estimated as

\( p = \frac{|S|}{|CN|} \). Therefore, we estimate that \( |\mathcal{Q}_{\text{SoidK}}| \approx \frac{|\mathcal{Q}_{\text{RoisK}}|}{|CN|} \). \( \mathcal{R} \propto K \) is on \( r \) (roles), we can estimate that \( |\mathcal{Q}_{\text{RoisK}}| \approx \frac{|\mathcal{Q}_{\text{RoisK}}|}{|CN|} \). As we see, the table \( K \) keeps unmodified, so its size \(|K|\) is fixed. In another respect, \( S \) and \( J \) are expanding and generally larger than \( K \) during the reasoning. Thus \( |\mathcal{Q}_{\text{RoisK}}| \) is always bigger than \( |\mathcal{Q}_{\text{SoidK}}| \). We have an observation that the number of roles is much less than that of concepts \(|CN| \) in most real ontologies, so \( |\mathcal{Q}_{\text{RoisK}}| \) is always the biggest one among the three intermediate results, which is also consistent with the experimental results.

We adopt the join order R2 for \( R2 \) and introduce a new mapping \( Q \) to split R3 into R3-1 and R3-2 in Table 5. \( Q \) records the intermediate result of R3 (corresponding to \( \mathcal{Q}_{\text{SoidK}} \)). Intuitively, \( Y, B, n \) in \( Q(r) \) implies \( \exists A \subseteq B, n \). \( Q(r) \) is initially set to \( \emptyset \) for each role \( r \). R3-1 and R3-2 can be handled by MapReduce programs.

### 3.4 Loading role inclusion axioms into memory

The Application of R5 can be seen as a 3-way join on two tables \( R \) and \( M \) as:

\[ R(r, X, Z, n_1) \propto M(s, Y, n_2) \]

This rule is unmodified because we have the observation that the number of role inclusion axioms (resp. roles) of the form \( r \subseteq s \) or \( s \subseteq t \) is much less than that of the concept inclusion axioms (resp. concepts) in some real ontologies like Snowed-CT and GALEN.

Therefore we assume the role inclusion axioms fit in memory, so that we parallelize the axioms of the form \( X, Y, n \) in \( R(r) \) into different map nodes and load the axioms of property chain into memory to complete the application of R5. To illustrate the process, we give Algorithm 1 and Algorithm 2 which correspond respectively to map function and reduce function of R5. For simple understanding, these two algorithms are described using the rule in Table 5.

**Algorithm 1** Map function for R5

**Input:** key: \( \langle X, Y, n_1 \rangle \in R(r) \) as the value

1. for each \( r \circ s \subseteq t \in \mathcal{O} \) do
2. emit(key:y, value:\( \langle X, Y, n_1 \rangle \in R(r) \))
3. end for
4. for each \( s \circ r \subseteq t \in \mathcal{O} \) do
5. emit(key:x, value:\( \langle X, Y, n_1 \rangle \in R(r) \))
6. end for

The map function completes two joins \( (R(r, X, Z, n_1) \propto M(s, Y, n_2) \propto M(r, s, t)) \). The results of the two joins are processed in reduce function. Therefore we
can use only one MapReduce task to handle R5, which helps reduce the numbers of MapReduce tasks. Inspired by the treatment for R5, we can also further optimize the application of R4. We first compute the role inclusion closure (RIC) which stands for the reflexive transitive closure of the axiom $r \subseteq s$ in $O$. When any new axiom $(X, Y, n) \in R(r)$ is obtained, namely after applying R2 and R5, we call Algorithm 3 to do further process with the loaded RIC. We use $r \subseteq s$ to describe that role $r$ is semantically subsumed by the role $s$, which is explicit in RIC.

Algorithm 2 Reduce function for R5

**Input:** key, iterator values

1: for each $(X, Z, n_1) \in R(r)$ in values do
2: for each $(Z, Y, n_2) \in R(s)$ in values do
3: for each $(X, Y, m) \in R(t)$ do
4: $m := \min(n_1, n_2)$
5: emit($(X, Y, m) \in R(t)$)
6: end for
7: end for
8: end for

This method allows R4 being omitted from the reasoning iteration, thus there is no need to consider the I/O overheads and map-out of R4. Since we choose not to fuzzify role axioms as well as [Stoilos et al., 2008], the application of RIC has no effects on the fuzzy values of other derived axioms. We call the function applyRIC in the reduce function of R2 and R5 to complete the inference task of R4. The reduce function of R2 is given by Algorithm 4 to illustrate how to finish the application of R4.

Algorithm 3 applyRIC for R2 and R5

**Input:** $(X, Y, n) \in R(r)$ : the inferences of R2 and R5

1: for each $s$ in $O$ do
2: if $r \subseteq s$ is in RIC then
3: emit($(X, Y, n) \in R(s)$)
4: end if
5: end for

Algorithm 4 Reduce function for R2

**Input:** key, iterator values

1: for each $(A, n_1) \in S(X)$ in values do
2: for each $(A \sqsubseteq \exists r.B, n_2) \in O$ in values do
3: $m := \min(n_1, n_2)$
4: emit($(X, B, m) \in R(r)$)
5: applyRIC($(X, B, m) \in R(r)$)
6: end for
7: end for

3.5 Overview of the reasoning algorithm

We first discuss the rationales of these revised rules. Rules R2, R4 and R5 are almost unchanged except the preconditions like $(B, m) \notin S(X)$ or $(X, B, m) \notin R(r)$ are omitted, as they are only used for termination judgment. Since we will consider the termination condition in our reasoning algorithm, there is no difference between these rules in Table 3 and Table 5. Rule R1 (resp. rule R3) is replaced by R1-1 and R1-2 (resp. R3-1 and R3-2). The outputs of R1-1 (resp. R3-1) are only used in the precondition of R1-2 (resp. R3-2), so it does not have any effect on final results.

We then give the reasoning algorithm based on the revised fuzzy-EL$^+$ rules.

Before reasoning, we first transform all input axioms to normalized forms and initializes $S$, $R$, $P$ and $Q$. The main part of the reasoning work is given by Algorithm 5, which consists of two phases. The first phase is preprocessing, in which Algorithm 5 creates the mapping $T$ and computes the complete role inclusion closure (RIC). The second phase is reasoning, in which Algorithm 5 iteratively applies the fuzzy-EL$^+$ rules until a fix point is reached. At the end of each iteration, a MapReduce task is used to delete the duplicates and get the greatest fuzzy value for an axiom obtained from completion rules. When there is no new axiom generated, the algorithm terminates.

Algorithm 5 Fuzzy-EL$^+$ reasoning

1: create the mapping $T$;
2: $RIC :=$ computeRIC();
3: $firstTime :=$ true;
4: $derived = 0$;
5: while $firstTime$ or $derived \geq 0$ do
6: $derived :=$ applyRules();
7: $firstTime :=$ false;
8: end while

The application of each rule can be handled by a MapReduce task. In map phase, each axiom which satisfies one of the preconditions of the rule is given as output in form of a key/value pair, where key is concept or role as shown in the left part of Table 5. All axioms having the same key are grouped from different map nodes and passed to one reduce node. The conclusions of the rule can be achieved in reduce phase. Since we can load the axioms of property chain into different nodes, the application of R5 can be done in one MapReduce task. We use RIC in the reduce phases of R2 and R5 to complete the inference task of R4.

4 Experiments

We implemented a prototype system based on a popular implementation of MapReduce model, Hadoop\(^3\), which is an open-source Java implementation project under the Apache Foundation.

Since there is no optimized fuzzy DL system for fuzzy-EL$^+$, we validate the correctness of our system against jCEL which is a reasoner handling EL ontologies. We run our system on the revised versions of test ontologies, i.e., we manually add fuzzy values to each axiom in these ontologies. Our system can produce the same results as jCEL without considering fuzzy values.

\(^3\)http://hadoop.apache.org/
The experiments were run in a Hadoop cluster containing 8 nodes. Each node is a PC machine with a 2-core, 3GHz, E8400 CPU, 2GB main-memory and 500G hard disk. In the cluster, each node is assigned two processes to run map tasks, and two processes to run reduce tasks. So the cluster allows program running on 16 mappers or 16 reducers simultaneously.

4.1 Test datasets

To compare our system with other reasoners and test its scalability, we generate fuzzy-\(\mathcal{EL}^+\) ontologies based on GALEN, called f-GALEN, for experimental purpose. In detail, for the normalized GALEN we assign a random fuzzy value \(f\) to each GCI axiom and keep RI axioms unfuzziified.

In order to validate the scalability of our algorithms, we need to run our system on datasets with different sizes to see the relation between the data volume and the throughput. For this purpose, we use a simple method which uses GALEN as a core and generates different number of copies based on the core, and these copies are independent. For the GALEN copies (here n-GALENs denotes to n copies of GALEN) with different sizes, we add fuzzy values to them and get the fuzzy versions (f-n-GALENs).

4.2 Comparison with memory-based reasoners

We compared the reasoning time with three memory-based reasoners ELK, jCEL and Pellet. ELK is a concurrent reasoner using multiple cores [Kazakov et al., 2011]. jCEL is a java implementation of CEL [Baader et al., 2006] for EL reasoning [Mendez, 2012]. Pellet is a tableau-based reasoner for OWL DL [Sirin and Parsia, 2004]. We ran these reasoners in one node of the cluster. In comparison we ran our system in the cluster without considering fuzzy values. The experimental results are given in Table 7. From the results we can see that for memory-based reasoners, the classification will finish when the input datasets fit in memory. For 8-GALENs, none of the three reasoners can finish classification with such memory that is given to them. Our system will finish reasoning on the four datasets using this cluster.

4.3 Scalability tests

To test the scalability of our algorithms, we ran two experiments. The first experiment ran on the cluster with 8 nodes (16 processing units), and handles four datasets with different sizes, they are f-1-GALEN, f-2-GALENs, f-4-GALENs and f-8-GALENs. We give the experimental results in Table 8 to show the relation between the data volume and the throughput. In the second experiment we ran our system on f-1-GALEN with different number of processing units (mappers and reducers) to see the relation between the processing units and the throughput.

| Table 7: Comparison of reasoning time (in seconds) |
|-----------------|--------|--------|-----------------|
| Test Datasets   | ELK    | jCEL   | Pellet          |
| 1-GALEN         | 2.3    | 116.2  | 742.4           |
| 2-GALENs        | 5.5    | 243.7  | -               |
| 4-GALENs        | 11.6   | -      | -               |
| 8-GALENs        | -      | -      | -               |
| (8 nodes)       |        |        | (8 nodes)       |

| Table 8: Scalability over data volume |
|-----------------|--------|--------|-----------------|
| Test Datasets   | Input (No. of axioms (K)) | Output (No. of axioms (K)) | Time (hours) | Throughput (Atoms(K)/minutes) |
| f-1-GALEN       | 90     | 6,838  | 1.82           | 62.62        |
| f-2-GALENs      | 178    | 13,680 | 3.32           | 68.67        |
| f-4-GALENs      | 352    | 27,349 | 5.53           | 82.42        |
| f-8-GALENs      | 703    | 54,699 | 10.63          | 85.76        |

Figure 1: Time versus number of copies

Figure 2: Time versus inverse of number of mappers

From the results of the first experiment, we can see that the throughput increases while the the size of datasets increases. Specially, when the test dataset changes from f-2-GALENs to
f-4-GALENs, the throughput increases significantly and the throughput while handling f-8-GALENs is 37% higher than the throughput while handling f-1-GALEN.

Since the cluster has overheads in startup, data transmission and processing, the speedup is non-linear shown in the results of the second experiments (see Table 9). Without considering the overheads and ignoring the constant from the time dimension, we can see that the reasoning time is proportional to the number of copies (see Figure 1) and inversely proportional to the number of units (see Figure 2). For the test datasets, the scalability of our system is validated from the experiments.

5 Conclusion and future work

In this paper, we proposed MapReduce algorithms for classifying ontologies based on fuzzy-$\mathcal{EL}^+$ (it is an extension of $\mathcal{EL}^+$ with fuzzy vagueness). We identified two main challenges using MapReduce for fuzzy-$\mathcal{EL}^+$ reasoning and proposed our solutions for tackling them. We revised the original rules and gave the classification algorithms using MapReduce framework. Furthermore, we implemented a prototype system for the evaluation. The experimental results show that this system has scalability and it can finish the work of classification through adding nodes in the cluster when ontologies do not fit in memory.

The memory-based reasoners mentioned in experiments cannot handle SNOMED-CT in single node because of the memory limit. Our system can process SNOMED-CT, although it cost nearly two days to finish the whole classification in our cluster.

In our next step, we will test the scalability using a larger cluster on the copies of SNOMED-CT and the ontologies in which SNOMED-CT is merged with other medical ontologies. We will also further optimize our algorithm based on following analysis of the limits of our system. 1) Since our system is based on fixed-point algorithms, it will scan the whole rules to check whether there are new axioms generated in each iteration. This costs overheads for reasoning. 2) Our system will set up same nodes for every rule in each iteration, however some rules will do more work than others on special datasets. We can get the statistic information of input ontologies and balance the usage of nodes for rules. 3) Hadoop will rescan all axioms and collect them for application of one rule R in every iteration. However these collected axioms do not need to be rescanned for R. So they can be processed in local node when applying R.

For test data, we would like to use the tool LogMap\(^3\) to get a fuzzy-$\mathcal{EL}^+$ ontology, i.e., we merge two ontologies with the mapping results and use the measures of similarity as fuzzy values. We also consider to extend fuzzy-$\mathcal{EL}^+$ to process ABox datasets [Ren et al., 2011], since ABox datasets are always beyond the memory capacity.

Acknowledgments

Guilin Qi is partially supported by the NSFC grant 61272378. Pascal Hitzler and Raghava Mutharaju are partially supported by the National Science Foundation under award 1017225 III: Small: TROn - Tractable Reasoning with Ontologies.

References


\(^3\)http://www.cs.ox.ac.uk/isg/projects/LogMap/