MONOGRAFIES DE L'INSTITUT D'INVESTIGACIÓ
EN INTEL·LIGÈNCIA ARTIFICIAL
Number 15

Institut d'Investigació
en Intel·ligència Artificial

Consell Superior
d'Investigacions Científiques

Monografies de l'Institut d'Investigació en Intel·ligència Artificial

Num. 1	J. Puyol, *MILORD II: A Language for Knowledge–Based Systems*
Num. 2	J. Levy, *The Calculus of Refinements, a Formal Specification Model Based on Inclusions*
Num. 3	Ll. Vila, *On Temporal Representation and Reasoning in Knowledge–Based Systems*
Num. 4	M. Domingo, *An Expert System Architecture for Identification in Biology*
Num. 5	E. Armengol, *A Framework for Integrating Learning and Problem Solving*
Num. 6	J. Ll. Arcos, *The Noos Representation Language*
Num. 7	J. Larrosa, *Algorithms and Heuristics for Total and Partial Constraint Satisfaction*
Num. 8	P. Noriega, *Agent Mediated Auctions: The Fishmarket Metaphor*
Num. 9	F. Manyà, *Proof Procedures for Multiple-Valued Propositional Logics*
Num. 10	W. M. Schorlemmer, *On Specifying and Reasoning with Special Relations*
Num. 11	M. López-Sánchez, *Approaches to Map Generation by means of Collaborative Autonomous Robots*
Num. 12	D. Robertson, *Pragmatics in the Synthesis of Logic Programs*
Num. 13	P. Faratin, *Automated Service Negotiation between Autonomous Computational Agents*
Num. 14	J. A. Rodríguez, *Towards a Test-bed for Trading Agents in Electronic Auction Markets*
Num. 15	T. Alsinet, *Logic Programming with Fuzzy Unification and Imprecise Constants: Possibilistic Semantics and Automated Deduction*

Logic Programming with Fuzzy Unification and Imprecise Constants: Possibilistic Semantics and Automated Deduction

Teresa Alsinet

Foreword by Lluís Godo

2002 Consell Superior d'Investigacions Científiques
Institut d'Investigació en Intel·ligència Artificial
Bellaterra, Catalonia, Spain.

Series Editor
Institut d'Investigació en Intel·ligència Artificial
Consell Superior d'Investigacions Científiques

Foreword by
Lluís Godo
Institut d'Investigació en Intel·ligència Artificial
Consell Superior d'Investigacions Científiques

Volume Author
Teresa Alsinet
Departament d'Informática i Enginyeria Industrial
Universitat de Lleida

Institut d'Investigació
en Intel·ligència Artificial

Consell Superior
d'Investigacions Científiques

NIPO
ISBN: 84-00-08054-8
Dip. Legal: 403-02-091-3

Ordering Information: Text orders should be addressed to the Library of the IIIA, Institut d'Investigació en Intel·ligència Artificial, Campus de la Universitat Autònoma de Barcelona, 08193 Bellaterra, Barcelona, Spain.

Printed by CPDA-ETSEIB.
Avinguda Diagonal, 647.
08028 Barcelona, Spain.

Als meus pares
i al Manel.

Contents

List of Figures

Foreword

Possibilistic logic, introduced by D. Dubois, H. Prade and his collaborators in the 90's, appears nowadays as a sound and well founded logical framework to reason under uncertainty in those domains where uncertainty is basically of ordinal, qualitative nature. It is a perfect complement to other logical frameworks based on traditional uncertainty models with a more quantitative flavor.

In Artificial Intelligence, possibilistic logic has become a powerful tool, not only for knowledge representation and reasoning modelling tasks, but also in domains such as decision under uncertainty or constraint satisfaction, where one needs to represent preferences and priorities.

In this book, the author pushes possibilistic logic a big step further by defining two languages, in a logic programming style, that extend first-order possibilistic logic to allow predicates to have imprecise or fuzzy object constants. This apparently small extension has very important consequences, as it amounts to combine in a single framework truth-functional elements of many-valued logic (to account for the fuzziness) together with elements of possibility theory (to account for the uncertainty), which are non-truth functional. To deal with both aspects, uncertainty and vagueness, in a sound and accurate logical way makes things quite complex. The two languages investigated in-depth (both in semantical and proof theoretical terms), PLFC and PGL, arise respectively when one considers extending the possibilistic logic components to accommodate the possible fuzziness in the knowledge, or when one considers extending a many-valued logical system to account for possibilistic uncertainty on the knowledge. For both languages, efficient proof systems are studied and developed. Last but not least, the author has done a remarkable work in presenting complex matters in a clear and didactic form, starting from the roots and going into details when necessary.

This monograph is based on the author's Ph.D. dissertation, which I enjoyed to supervise and that has enriched me both in the scientifical and personal level.

Lluís Godo
IIIA - CSIC, Bellaterra, October 2002

Abstract

In this thesis we focus on the formalization of first-order logical systems for reasoning under possibilistic uncertainty and vague knowledge. In this framework, we first define a formal semantics and a sound resolution-style calculus by refutation for the necessity-valued fragment of first-order possibilistic logic extended with fuzzy constants and fuzzily restricted quantifiers (called PLFC). The resolution rule includes an implicit fuzzy unification mechanism of fuzzy events. Second, we formalize a general fuzzy possibilistic logic (called PGL) based on Gödel infinitely-valued logic, and we define a complete modus ponens-style calculus for the Horn-rule fragment of PGL. Third, we extend this logic with fuzzy constants and a fuzzy unification mechanism which preserves completeness for a particular class of clauses. Finally, we extend the fuzzy unification mechanism to the first-order case, and we describe how this mechanism can be implemented in the Warren Abstract Machine context.

Concerning the semantics, because of the fuzzy information, the truth evaluation of formulas is many-valued and belief states are modeled by normalized possibility distributions on a set of many-valued interpretations in both PLFC and PGL. Hence, we consider a suitable extension of the notion of necessity measure for fuzzy sets in both systems; in particular, for fuzzy sets of interpretations. However, the semantics of PLFC and PGL are not equivalent. On the one hand, the basic connectives of PLFC are negation and disjunction and of PGL are conjunction and implication, and the two sets of connectives are not inter-definable. On the other hand, the extended necessity measure for fuzzy sets used in PLFC is different from the one used in PGL, although both are extensions of the standard necessity measure for crisp sets.

Concerning the automated deduction system, the proof method for PLFC is defined by refutation through a generalized resolution rule extended with a necessity measure of matching of fuzzy events. During the proof process, a merging rule must be applied after each resolution step, and the search space consists of all possible orderings of the literals in the knowledge base. In contrast, in PGL, the merging rule is applied during a pre-processing step of the knowledge base, and the proof procedure is defined by deduction through a generalized modus ponens rule and a unification mechanism of fuzzy constants based on three inference patterns.

Acknowledgements

This book contains my PhD thesis, submitted to the *Departament de Llenguatges i Sistemes Informàtics*, *Universitat Politècnica de Catalunya*, in July 2001.
This thesis would never exist without the help and encouragement of many people. My sincerest thanks go to:

– Lluís Godo, my advisor, for his support and advice during my work on this thesis;

– Didier Dubois, Francesc Esteva, Jordi Levy, Felip Manyà and Sandra Sandri for accepting to join my examining committee and for valuable comments and suggestions;

– the *Institut d'Investigació en Intel·ligència Artificial* (IIIA-CSIC) whose members contributed with ideas, discussions and in many other ways to this thesis;

– the *Escola Universitària Politècnica* and the *Departament d'Informàtica i Enginyeria Industrial* at the *Universitat de Lleida* for providing a very special work environment;

– the CICYT and the *Universitat de Lleida* for partially supporting this research;

– my colleagues Carlos Asótegui, Ramón Béjar, Alba Cabiscol, Cèsar Fernàndez, Carles Mateu, Marga Moltó, Silvia Penas and Magda Valls for the good times we have shared;

– and especially my family and friends for their love and support over the years.

Chapter 1

Introduction

1.1 Context and objectives

Logic programming began in the early 1970's as a direct outgrowth of previous work in automatic theorem proving. Building on work of Herbrand in 1930 (Herbrand, 1967), there was much activity in theorem proving in the early 1960's by Davis and Putnam (1960), Gilmore (1960), Prawitz (1960), and many others. This effort culminated in the definition of the *resolution rule* by Robinson (1965). Subsequently, Kowalski (1974) and Comerauer et al. (1973) were led to the fundamental idea that *logic can be used as a programming language*, which is based on a *procedural interpretation* of formulas, and the first Prolog interpreter was implemented.

One of the main ideas of logic programming, which is due to Kowalski (1979a,1979b), is that an algorithm consists of two disjoint components: the logic and the control. The logic is the statement of *what* the problem is that has to be solved. The control is the statement of *how* it is to be solved and has to be provided by the logic programming system itself.

From then, logic programming languages have been applied to a wide range of areas such as artificial intelligence and deductive databases. Among these languages, Prolog is the most well-known, but it is not powerful enough for reasoning and representing knowledge in situations in which there is vague, imprecise, or incomplete information. However, as Zadeh has stated, the challenge now facing both subfields of computer science is to produce systems exhibiting common sense reasoning and human-knowledge representation, rather than purely symbolic logical deduction. The representation of human-originated information and the formalization of some forms of common sense reasoning motivated the creation and development of fuzzy set theory by Zadeh (1965). Fuzzy sets bring to logic programming a mathematical framework for capturing impreciseness as well as uncertainty.

To get insight into the domain of this thesis let us first clarify the distinction between impreciseness and uncertainty.

Impreciseness or *vagueness* is *fuzziness*, a phenomenon arising when we try to characterize phenomena encountered in the world. It is connected with delineation of classes of objects having some property, which is general imprecise or vague and which thus leads to classes with no sharp boundaries. For instance, *young* is a vague property and we cannot uniquely decide about a given person whether he or she is young or not –the less is the age of a person (measured e.g. in years), the more true is that he or she is young. The, truth of propositions (statements) involving imprecise or vague information is a matter of degree which may lay between true and false. Fuzziness can thus be characterized by some kind of many-valued logic and namely, by fuzzy logic and fuzzy sets. The word "crisp" is used as meaning "non-fuzzy".

Uncertainty is a phenomenon which accompanies the occurrence of other phenomena, mostly in time. It arises when we try to characterize phenomena under incomplete or partially inconsistent information. For instance, we may be more or less certain that a given person is 35 years old and, usually, our degree of uncertainty is due to the fact that we do not have enough information about him or her. Hence, uncertainty of occurrence is as well a matter of degree which may lay between absolute ignorance and absolute certainty. Possibility, probability and belief theories characterize different aspects of uncertainty.

A significant distinction between truth and uncertainty on the formal level consists of the compositionality. Namely, the truth of the composite property, say (φ and/or ψ)(x_0), can be computed from the truths of $\varphi(x_0)$ and $\psi(x_0)$. This does not hold for the uncertainty (and its forms – possibility, probability or belief). And, following the discussion of Dubois and Prade (1986) and Dubois et al. (1991a), one could say that uncertainty concerns precise (or imprecise) statements under incomplete information while vagueness (logic of truth) concerns imprecise statements under complete information.

The problems connected with logic programming in logics of uncertainty and fuzziness are mostly reduced to the question of how some generalization of the resolution principle can be formulated and how the automated deduction can be realized. The *fuzzy resolution principle* roots to 1972 when the paper of Lee (1972) was published. Later on, Ishizuka and Kanai (1985) implemented Lee's results in the fuzzy Prolog-ELF system. From then, new inference systems have been proposed based on a variety of non-classical logics, such as multiple-valued logics (Escalada-Imaz and Manyà, 1995; Klawonn and Kruse, 1994; Vojtáš, 2001a; Yasui and Mukaidono, 1996), possibilistic logic (Dubois et al., 1988a; Dubois and Prade, 1990; Dubois et al., 1991b), probabilistic logic (Bacchus, 1990; Heinsohn, 1994; Lukasiewicz, 1998), evidential support logic (Baldwin, 1986; Baldwin, 1987a; Baldwin et al., 1993), fuzzy operator logic (Liu and Xiao, 1985; Liu, 1989; Weigert et al., 1993). Depending on the underlying logic, some systems are more suitable for dealing with vague knowledge, and others are more appropriate for reasoning under incomplete or partially inconsistent knowledge. Therefore, most of the fuzzy logic programming systems proposed in the literature implement proof procedures for reasoning under one of the two complementary forms of indeterminacy: uncertainty and vagueness.

And, although there are some attempts to handle linguistic truth values (Alsinet and Manyà, 1996; Hinde, 1986; Li and Liu, 1990; Yasui and Mukaidono, 1998) and fuzzy unification (Arcelli et al., 1998; Bel et al., 1986; Cayrol et al., 1982; Formato et al., 2000; Gerla and Sessa, 1999; Schwartz, 1989; Umano, 1987; Virtanen, 1998; Vojtáš et al., 2001b) in the framework of fuzzy logic programming, only few inference systems, such as the ones defined by Baldwin et al. (1995) for evidential reasoning, and Dubois et al. (1996,1998b) and Godo and Vila (1995) for possibilistic logic, provide a unified framework for the treatment of vagueness and uncertainty.

The main objective of this thesis is the definition of first-order logic programming systems for reasoning under possibilistic uncertainty and vague knowledge which allows us, for instance, to model statements of the form

"it is almost certain that Peter is young",

where the vague property *young* is represented by means of a fuzzy constant, and the uncertainty expression *almost sure* is represented by means of a degree in the unit interval $[0, 1]$, for instance 0.9, expressing how much the fuzzy statement "Peter is young" is believed in terms of necessity measures.

Our departure points are: (i) the syntactic extension proposed by Dubois et al. (1996,1998b) of the necessity-valued fragment of first-order possibilistic logic dealing with fuzzy constants and fuzzily restricted quantifiers, which is called PLFC; and (ii) the propositional temporal logic defined by Godo and Vila (1995), which is based on a possibilistic semantics and a sound logical inference system allowing a partial matching between fuzzy temporal constraints.

To achieve our goal, we explore three lines of research. First, we define a formal semantics and a sound refutation proof method by resolution for PLFC. Then, following the approach proposed by Godo and Vila (1995), we define a possibilistic logic programming language with fuzzy constants based on the Horn-rule fragment of Gödel infinitely-valued logic extended with a fuzzy unification mechanism. Finally, we extend the fuzzy unification mechanism to the first-order case and we provide the resulting language with a logic programming environment based on a suitable extension of the Warren Abstract Machine (WAM) (Warren, 1983; Aït-Kaci, 1991) to our fuzzy and possibilistic context.

In the next section, we briefly describe these three lines of research.

1.2 Contributions

The **first contribution** of this thesis consists of both the formalization of PLFC itself and an automated deduction system for it by (i) providing a formal semantics; (ii) defining a sound resolution-style calculus by refutation; and (iii) describing a first-order proof procedure for PLFC clauses based on the calculus and on a novel notion of most general substitution of two literals in a resolution step.

In PLFC, formulas are pairs of the form

$$(\varphi(\bar{x}), f(\bar{y})),$$

where $\bar{x}$ and $\bar{y}$ denote sets of free and implicitly universally quantified variables, such that $\bar{y} \supseteq \bar{x}$, $\varphi(\bar{x})$ is a disjunction of literals with fuzzy constants, and $f(\bar{y})$ is a valid valuation function that expresses the certainty of $\varphi(\bar{x})$ in terms of necessity measures. Basically, valuation functions $f(\bar{y})$ are constant values and variable weights. Variable weights (Dubois et al., 1994a; Dubois et al., 1994b) are suitable for modeling statements of the form

"the more x is A (or x belongs to A), the more certain is $p(x)$",

where A is a fuzzy set with membership function $\mu_A(x)$. This is formalized in PLFC as,

"for all x, $p(x)$ is certain with a necessity of at least $\mu_A(x)$",

and is represented as $(p(x), A(x))$. When A is imprecise but not fuzzy, the interpretation of such a formula is just "$\forall x \in A$, $p(x)$". So variable weights act as (flexible if they are fuzzy) restrictions on an universal quantifier. On the other hand, fuzzy constants in PLFC can be seen as (flexible) restrictions on an existential quantifier. In general,

"$L(B)$ is certain with a necessity of at least α"

is represented in PLFC as $(L(B), \alpha)$, where L is either a positive or a negative literal and B is a fuzzy constant. For instance, if B is imprecise but not fuzzy, $p(B)$ and $\neg p(B)$ are interpreted in PLFC as

"$\exists x \in B, p(x)$" and "$\exists x \in B, \neg p(x)$",

respectively. When B is fuzzy, we get a more complex complex interpretation in terms of level-cuts. Thus, if B is a fuzzy set with a membership function μ_B, $p(B)$ is interpreted in possibilistic terms as

"$[\exists x \in [\mu_B]_\beta, p(x)]$ is certain with a necessity of at least $1 - \beta$"

for each $\beta \in [0, 1]$, where $[\mu_B]_\beta$ denotes the β-cut of μ_B.

Concerning PLFC semantics, as predicates are allowed to talk about poorly known, imprecise or fuzzy constants and a form of fuzzily restricted quantifiers is also present, we introduce three major changes with respect to the standard semantics of first-order possibilistic logic. On the one hand, all fuzzy components are interpreted as fuzzy sets, and thus, the truth value of formulas is not Boolean but many-valued and the set of truth values is the whole unit interval $[0, 1]$. On the other hand, belief states are modeled by normalized possibility distributions on a set of many-valued interpretations, and thus, we consider a suitable extension of the notion of necessity measure for fuzzy sets, in particular for fuzzy sets of interpretations. Finally, in order to measure the certainty of PLFC formulas in a possibilistic model, we cannot take into account all possible many-valued interpretations, but only those which share a common interpretation of fuzzy constants what leads us to define the notion of context.

As in classical first-order possibilistic logic, the proof method for PLFC is refutation by a generalized resolution rule between positive and negative literals.

However, as we consider fuzzy constants and variable weights in the language, we introduce two major changes with respect to classical first-order inference systems. On the one hand, we define a directional fuzzy unification mechanism between variable weights and fuzzy constants. On the other hand, in order to compute higher degrees of unification, we define a merging rule between clauses.

In logic programming languages, unification is a basic process in a proof procedure, used to match the terms of atomic formulas, and thus allowing the inference of new information through some resolution-like rule. In classical logic, the unification process has been well-established and clearly defined (Robinson, 1965); in particular, when trying to match a constant, A, to another constant, B, the unification algorithm fails whenever $A \neq B$, and thus, the unification mechanism is based on a syntactic and symmetric matching. New problems arise when we allow fuzzy constants in the system. On the one hand, we must measure the degree of matching between fuzzy events, for instance, the linguistic terms *between_22_and_25* and *young*, both defined over a common reference set `years`. On the other hand, as people between 22 and 25 years old are usually considered to be young, but not all young people are between 22 and 25 years old, the fuzzy pattern matching measure must be asymmetrical or "directional" in the sense of estimating the certainty degree of a fuzzy event based on some available fuzzy information, and thus, to compute the unification degree between two fuzzy events in a logic programming system we must know the origin of each fuzzy information in the knowledge base.

In PLFC, due to the disjunctive interpretation of fuzzy constants, the unification (in the classic sense) between fuzzy constants is not allowed. For instance, from

$$(\neg p(A) \vee \psi, 1) \quad \text{and} \quad (p(A), 1),$$

which, if A is not fuzzy, are interpreted respectively as

$$\text{“}[\exists x \in A,\ \neg p(x)] \vee \psi\text{”} \quad \text{and} \quad \text{“}\exists x \in A,\ p(x)\text{”},$$

we can infer ψ only when A is a precise constant. However, as variable weights are interpreted as conjunctive information, a partial matching between fuzzy events is allowed through variable weights and fuzzy constants. For instance, from

$$(\neg p(x) \vee \psi(x), A(x)) \quad \text{and} \quad (p(B), 1),$$

which, if A and B are not fuzzy, are interpreted respectively as

$$\text{“}\forall x \in A,\ \neg p(x) \vee \psi(x)\text{”} \quad \text{and} \quad \text{“}\exists x \in B,\ p(x)\text{”},$$

we infer $(\psi(B), N(A \mid B))$, where $N(A \mid B)$ is a necessity measure of matching between fuzzy events. Hence, in PLFC the unification mechanism consists of determining the certainty of a variable weight based on the information of a fuzzy constant.

PLFC provides a powerful framework for reasoning under disjunctive and conjunctive vague knowledge, but due to variable weights, has some computational limitations. The current proof method for PLFC (resolution-style by

refutation) is not complete and does not allow for the definition of an efficient proof algorithm. In order to stretch variable weights and compute higher degrees of unification, a merging rule must be applied after every resolution step, and the search space consists of all possible orderings of the literals in the knowledge base. On the other hand, as the calculus for PLFC is defined by refutation through a generalized resolution rule between positive and negative literals, to prove a PLFC formula involving fuzzy constants, for instance, $(p(A), 1)$, which if A is not fuzzy is interpreted as "$\exists x \in A,\ p(x)$", we must add to the knowledge base $\neg[(p(A), 1)]$ which is represented and interpreted in PLFC as

$$(\neg p(x),\ A(x)) \quad \text{and} \quad \text{"}\forall x \in A,\ \neg p(x)\text{"},$$

respectively. Hence, clauses with variable weights are strictly necessary in PLFC.

Due to these limitations, we have turned our attention to a possibilistic language based on definite clauses with fuzzy constants.

The **second contribution** of this thesis is the formalization of a possibilistic logic programming language with fuzzy constants based on the Horn-rule fragment of Gödel infinitely-valued logic which, as in classical logic programming systems, enables us to define an efficient proof algorithm based on a complete calculus and oriented to goals. The definition of this language has been done in three sequential phases. Next, we briefly describe the results achieved in each phase.

First, we formalize a general fuzzy possibilistic logic based on Gödel infinitely-valued logic (called PGL) by (i) defining a syntax; (ii) providing a well-behaved and well-featured possibilistic semantics on top of truth functions of Gödel logic; and (iii) defining a sound inference method by deduction based on a Hilbert-style axiomatization of the logic and a generalized modus ponens inference rule.

Second, we focus on the fuzzy possibilistic logic programming language that results from considering the Horn-rule fragment of PGL, and we prove for this restricted class of formulas that the modus ponens-style calculus is complete for determining the maximum degree of possibilistic belief with which a fuzzy propositional variable can be entailed from a set of formulas. In the particular case that we do not allow recursive formulas in the language, the underlying uncertainty logic of our logic programming language is syntactically equivalent to the family of multiple-valued propositional logics the interpreter defined by Escala-Imaz and Manyà in (Escalada-Imaz and Manyà, 1995; Manyà, 1996) can deal with. The interpreter is based on a backward proof algorithm for computing the maximum degree of deduction of a fuzzy propositional variable from a set of formulas whose worst-case time complexity is linear in the total number of occurrences of propositional variables in the set of formulas.

Third, we extend the Horn-rule fragment of PGL with fuzzy constants. We refer to this extension as PGL$^+$. In PGL$^+$, fuzzy constants are interpreted as disjunctive imprecise knowledge. In doing so, we are led to extend the underlying uncertainty logic at two levels. On the one hand, many-valued interpretations regard fuzzy constants as fuzzy sets, and thus, the already proposed notion of

context of PLFC is also needed. On the other hand, in order to allow a partial matching between fuzzy constants, we extend the PGL modus ponens-style calculus with a fuzzy unification mechanism based on a necessity-like measure and five deduction rules. In this context, (i) we formalize the notion of non-recursive and satisfiable PGL$^+$ program; (ii) we prove, for this restricted class of programs satisfying a context constraint, that the modus ponens-style calculus extended with the semantical unification mechanism is complete for determining the maximum degree of possibilistic belief with which a goal with fuzzy constants can be entailed; and (iii) we define an efficient as possible proof algorithm for computing the maximum degree of deduction of a fuzzy goal from a non-recursive and satisfiable PGL$^+$ program.

The proof procedure for a PGL$^+$ program clearly departs from classical propositional interpreters and is performed in four sequential steps. The first step consits in a satisfiability testing procedure of PGL$^+$ clauses. The second step consits in a pre-processing procedure which, due to the disjunctive interpretation of fuzzy constants, ensures that all (hidden) clauses of a PGL$^+$ program are considered. The third step consits in a translation procedure which takes a PGL$^+$ program extended with all (hidden) clauses and translates it into a semantically equivalent set of facts for determining the maximum degree of possibilistic belief with which a fuzzy goal can be entailed, whenever the program satisfies a context constraint. The fourth step consits in a deduction procedure, based on a necessity-like measure, which computes the maximum degree of deduction of a fuzzy goal from the equivalent set of facts.

The pre-processing step is strictly necessary in order to ensure completeness and it is conceptually equivalent to apply the merging rule in PLFC, and thus, the set of (hidden) clauses of a PGL$^+$ program is hard to be computed. However, after applying the pre-processing and the translation steps to the clauses of a PGL$^+$ program, only if new rules are added to the knowledge base this set of (hidden) clauses must be computed again. Hence, when extending the knowledge base with facts only the equivalent set of facts must be computed again, and the time complexity of the algorithm is linear in the total number of occurrences of predicates symbols in the set of clauses. Finally, the maximum degree of deduction of a fuzzy goal can be computed in a constant time complexity from the equivalent set of facts –in the sense that it is equivalent to compute the partial matching between two fuzzy constants. And, for each goal with predicate symbol q, the computed degree of deduction is the maximum degree of possibilistic entailment whenever the PGL$^+$ program satisfies a context constraint for q, which is easily checked during the translation step. Hence, for each goal the proof algorithm determines if the computed degree of deduction is the maximum degree of possibilistic entailment.

The **final contribution** of this thesis is the extension of PGL$^+$ to the first-order case. We refer to this extension as PGL$^+\forall$. In this context, (i) we extend the syntax and the semantics of PGL$^+$ to allow variables in the language; (ii) we provide the language with a sound modus ponens-style calculus by deduction through some unification rules, a fusion rule and weight weaken-

ing rule; (iii) we formalize the notions of most general directional fuzzy unifier and unification degree of a pair of atomic formulas; (iv) we develop a directional fuzzy unification algorithm based on a distinction between general and specific patterns; (v) we define a proof procedure oriented to program queries that applies the generalized modus ponens inference rule in a reverse way by using a depth-first strategy; and (vi) we define a system for translating $\mathrm{PGL}^{+}\forall$ programs into machine code by extending the WAM with the directional fuzzy unification algorithm and certainty weights.

1.3 Publications

Most of the results presented in this thesis have already been published as journal articles or in conference proceedings:

(1) T. Alsinet, F. Manyà, L. Jové and J. Morillo. A multiple-valued logic programming system: Definition and implementation. In *Actas de la VI Conferencia de la Asociación Española para la Inteligencia Artificial, CAEPIA-95*, pages 37–45, Alicante, Spain, 1995.

(2) T. Alsinet and F. Manyà. A declarative programming environment for infinitely-valued logics. In *Proceedings of the Sixth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU-96*, volume 3, pages 1205–1210, Granada, Spain, 1996.

(3) T. Alsinet. Una primera aproximación a la unificación fuzzy vía relaciones de compatibilidad. In *Actas del VII Congreso Español sobre Tecnologías y Lógica Fuzzy, ESTYLF-97*, pages 159–164, Tarragona, Spain, 1997.

(4) T. Alsinet. Algorisme d'unificació fuzzy direccional. *Butlletí de l'Associació Catalana d'Intel·ligència Artificial*, 12, pages 109–117. Actes de les Jornades d'Intel·ligència Artificial: Noves Tendències, Lleida, Spain, 1997.

(5) T. Alsinet and L. Godo. Fuzzy unification degree. In *Proceedings of the Second International Workshop on Logic Programming and Soft Computing*, pages 23–43, Manchester, UK, 1998.

(6) T. Alsinet, L. Godo and S. Sandri. On the semantics and automated deduction for PLFC, a logic of possibilistic uncertainty and fuzziness. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, UAI-99*, pages 3–12, Stockholm, Sweden, 1999.

(7) T. Alsinet and L. Godo. A complete calculus for possibilistic logic programming with fuzzy propositional variables. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, UAI-2000*, pages 1–10, Standford, CA, USA, 2000.

(8) T. Alsinet and L. Godo. A complete proof method for possibilistic logic programming with semantical unification of fuzzy constants. In *Actas del X Congreso Español sobre Tecnologías y Lógica Fuzzy, ESTYLF-2000*, pages 279–284, Sevilla, Spain, 2000.

(9) T. Alsinet and L. Godo. Adding similarity to possibilistic logic with fuzzy constants. In *Proceedings of the Ninth International Fuzzy Systems Association World Congress, IFSA-2001*, pages 1535–1540, Vancouver, British Columbia, Canada, 2001.

(10) T. Alsinet and L. Godo. A proof procedure for possibilistic logic programming with fuzzy constants. In *Proceedings of the Sixth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU-2001*, pages 760–771, Toulouse, France, 2001. Springer, LNAI 2143.

(11) T. Alsinet and L. Godo. Towards an automated deduction system for first-order possibilistic logic programming with fuzzy constants. *International Journal of Intelligent Systems*, volume 17, pages 887–924, 2002.

1.4 Structure of the thesis

The thesis is organized in eight chapters, whose contents are summarized below.

Chapter 1. Introduction

In this chapter, we first discuss and motivate the importance of defining and implementing logic programming systems for reasoning under both of the two complementary forms of indeterminacy: uncertainty and vagueness. Then, we describe the objectives, main results and structure of the thesis.

Chapter 2. Related work on inference systems of fuzzy logic

In this chapter, we provide an overview, from a formal and logic programming point of view, of some inference systems of fuzzy logic, where the term "fuzzy logic" mainly refers to truth-functional multiple-valued logics.

Chapter 3. Background on necessity-valued possibilistic logic and its extension with fuzzy constants and fuzzily restricted quantifiers

In this chapter, we first present formal and logic programming aspects of a fragment of possibilistic logic where formulas are valued by a lower bound on their degree of necessity. Then, we give a detailed description of the inference patterns proposed by Dubois, Prade and Sandri to handle fuzzy constants and fuzzily restricted quantifiers in this fragment of possibilistic logic. We refer to this extension as PLFC.

Chapter 4. On the semantics and automated deduction for PLFC

In this chapter, we provide PLFC with a formal semantics and a sound proof method by refutation using an extended version of the resolution inference rule based on a necessity-like measure for computing the partial matching between fuzzy events.

Chapter 5. A fuzzy possibilistic logic based on Gödel infinitely-valued logic

In this chapter, we first define a general fuzzy possibilistic logic based on propositional Gödel infinitely-valued logic. We refer to this logic as PGL. Then, we provide the fuzzy possibilistic logic programming language that results from considering the Horn-rule fragment of PGL with a complete modus ponens-style calculus for determining the maximum degree of possibilistic belief with which a fuzzy propositional variable can be entailed from a set of formulas.

Chapter 6. A complete calculus for PGL extended with fuzzy constants

In this chapter, we first extend the Horn-rule fragment of PGL with fuzzy constants and a semantical unification mechanism which preserves completeness for a particular class of formulas. We refer to this extension as PGL$^+$. Then, we focus on automating the computation of the maximum degree of deduction of an atomic formula with fuzzy constants from a knowledge base.

Chapter 7. An automated deduction system for a first-order extension of PGL$^+$

In this chapter, we first extend the underlying uncertainty logic of PGL$^+$ to the first-order case. Then, we focus on logic programming aspects by defining a directional fuzzy unification algorithm and a backward first-order proof procedure. Finally, we define a logic programming environment for this first-order possibilistic language with fuzzy constants.

Chapter 8. Conclusions and future work

In this chapter, we briefly summarize the main contributions of the thesis, and point out some open problems and future research perspectives that we plan to tackle in the near future.

Finally, the reader can find the references appearing in the text.

Chapter 2

Related work on inference systems of fuzzy logic

2.1 Introduction

In this chapter, we present some inference systems of fuzzy logic, where the term "fuzzy logic" mainly refers to truth-functional multiple-valued logics.

The chapter is organized as follows. In Section 2.2, we develop some deductive systems of fuzzy logic related with our framework. In Section 2.3, we focus in the field of resolution systems of fuzzy logic being the base of many fuzzy programming language proposals. In Section 2.4, we present some rule-based systems of fuzzy logic. Finally, in Section 2.5 we present some logic programming systems of fuzzy logic.

2.2 Deductive systems of fuzzy logic

In this section, following Hájek and Godo (1997) and Hájek (1998b), we describe logical properties of some truth-functional many-valued logics related with our framework. The set of truth values is the unit interval $[0, 1]$ and truth functions behave classically on extremal truth values 0 and 1, and satisfy some natural monotonicities. Thus, the truth function of conjunction (disjunction) is non-decreasing in both arguments; the truth function of implication is non-decreasing in the second argument but non-increasing in the first, i.e. the less true is the antecedent φ and the more is true the consequent ψ the more is true the implication $\varphi \to \psi$; and negation is non-increasing. This leads to the notion of a t-norm (cf. Schweizer and Sklar (1963)). A t-norm is an operation

$$* : [0, 1]^2 \to [0, 1]$$

which is commutative and associative, non-decreasing in both arguments, and having 1 as unit element and 0 as zero element, i.e.

$$
\begin{array}{l}
x * y = y * x, \\
(x * y) * z = x * (y * z), \\
x \leq x' \text{ and } y \leq y' \text{ implies } x * y \leq x' * y', \\
1 * x = x, \\
0 * x = 0.
\end{array}
$$

Usually *continuous* t-norms are considered as good candidates for truth functions of conjunctions. Each continuous t-norm $*$ determines uniquely a residuated implication $\Rightarrow$ (not necessarily continuous) satisfying that

$$x \Rightarrow y \geq z \text{ iff } x * z \leq y,$$

for each $x, y, z \in [0, 1]$; and defined as

$$x \Rightarrow y = \max\{z \in [0, 1] \mid x * z \leq y\},$$

for each $x, y \in [0, 1]$.

Three outstanding examples of fuzzy logic systems are:

Gödel logic (Gödel, 1932) with the conjunction

$$x * y = min(x, y)$$

and the corresponding residuated implication

$$x \Rightarrow y = \begin{cases} 1, & \text{if } x \leq y \\ y, & \text{otherwise.} \end{cases}$$

Łukasiewicz logic (Łukasiewicz, 1970) with the conjunction

$$x * y = max(x + y - 1, 0)$$

and the corresponding residuated implication

$$x \Rightarrow y = \begin{cases} 1, & \text{if } x \leq y \\ 1 - x + y, & \text{otherwise.} \end{cases}$$

Product logic (Hájek et al., 1996) with the conjunction

$$x * y = x \cdot y \quad \text{(product of reals)}$$

and the corresponding residuated implication

$$x \Rightarrow y = \begin{cases} 1, & \text{if } x \leq y \\ y/x, & \text{otherwise.} \end{cases}$$

In all these cases, negation, denoted $\neg$, is defined as

$$\neg x = x \Rightarrow 0.$$

One can show that each continuous t-norm is composed in a certain way from these three examples by an algebraic construct called *ordinal sum* (see e.g. Klement et al. (2000)), and the min and max operations are definable from $*$ and $\Rightarrow$. Indeed, for each continuous t-norm $*$ and its residuated implication $\Rightarrow$, the following identities are true:

$$\begin{aligned} \min(x, y) &= x * (x \Rightarrow y), \\ \max(x, y) &= \min((x \Rightarrow y) \Rightarrow y, (y \Rightarrow x) \Rightarrow x). \end{aligned}$$

Next, we present Hájek's basic many-valued propositional logic (Hájek, 1998b) since it is the logic that captures what is common to all fuzzy logic systems based on continuous t-norms. Then, we develop Gödel propositional and predicate logic since it underlies one of the logic programming systems defined in this thesis. Finally, we briefly describe some logical properties of Łukasiewicz and Product logics. In all these logical systems, deduction is based on modus ponens in the setting of a Hilbert-style axiomatization and provable formulas correspond to 1-tautologies, i.e. formulas that take the truth value 1 under any interpretation. Logics of partial truth have been studied, in a very general manner, by Pavelka (1979). The reader interested in formal aspects of these logics is invited to consult (Hájek, 1995a; Novák, 1990; Novák, 1995a). The theory of deductive systems of fuzzy logic is fully developed in (Gottwald, 1999; Gottwald, 2001; Hájek, 1998b).

2.2.1 The basic many-valued logic

The language of basic fuzzy propositional logic (denoted hereafter BL) is built in the usual way from a (countable) set of propositional variables, a (strong) conjunction $\&$, an implication $\rightarrow$ and the truth constant $\overline{0}$. When fixing a continuous t-norm $*$, we fix a propositional calculus PC($*$) (whose set of truth values is $[0, 1]$). This means that $*$ is the truth function of the (strong) conjunction $\&$, and the residuated implication $\Rightarrow$ of $*$ becomes the truth function of the implication $\rightarrow$. Further connectives are defined as follows:

$$\begin{array}{ll} \varphi \wedge \psi & \text{is } \varphi\&(\varphi \rightarrow \psi), \\ \varphi \vee \psi & \text{is } ((\varphi \rightarrow \psi) \rightarrow \psi) \wedge ((\psi \rightarrow \varphi) \rightarrow \varphi), \\ \neg\varphi & \text{is } \varphi \rightarrow \bar{0}, \\ \varphi \equiv \psi & \text{is } (\varphi \rightarrow \psi)\&(\psi \rightarrow \varphi). \end{array}$$

An *interpretation* of propositional variables is a mapping $\mathbf{I}$ assigning to each propositional variable p its truth value $\mathbf{I}(p) \in [0, 1]$. This extends to each formula via truth functions as follows:

$$\mathbf{I}(\bar{0}) = 0,$$

$$\begin{aligned}\mathbf{I}(\varphi \to \psi) &= \mathbf{I}(\varphi) \Rightarrow \mathbf{I}(\psi),\\ \mathbf{I}(\varphi\&\psi) &= \mathbf{I}(\varphi) * \mathbf{I}(\psi).\end{aligned}$$

A formula φ is a 1-tautology of PC($*$) if $\mathbf{I}(\varphi) = 1$ for all interpretation $\mathbf{I}$.

The following is an axiomatization of BL:

(A1)	$(\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi))$
(A2)	$(\varphi\&\psi) \to \varphi$
(A3)	$(\varphi\&\psi) \to (\psi\&\varphi)$
(A4)	$(\varphi\&(\psi\&\chi)) \to ((\varphi\&\psi)\&\chi)$
(A5a)	$(\varphi \to (\psi \to \chi)) \to ((\varphi\&\psi) \to \chi)$
(A5b)	$((\varphi\&\psi) \to \chi) \to (\varphi \to (\psi \to \chi))$
(A6)	$((\varphi \to \psi) \to \chi) \to (((\psi \to \varphi) \to \chi) \to \chi)$
(A7)	$\bar{0} \to \varphi$

The *deduction rule* of BL is modus ponens (from φ and $\varphi \to \psi$ derive ψ). Given this, the notions of a *proof* and of a *provable formula* in BL are defined as usual (relative to BL axioms).

All axioms of BL are 1-tautologies in each PC($*$). If φ and $\varphi \to \psi$ are 1-tautologies of PC($*$), then ψ is also a 1-tautology of PC($*$). Hence, each formula provable in BL is a 1-tautology of each PC($*$).

A *theory* over BL is a set of formulas. A *proof* in a theory T is a sequence $\varphi_1, \ldots, \varphi_n$ of formulas of which each member is either an axiom of BL, or a member of T (special axiom), or follows from some preceding members of the sequence using the deduction rule modus ponens. A formula φ is *provable* in a theory T, written $T \vdash \varphi$, if φ is the last member of of a proof in T.

The *deduction theorem* for BL reads as follows. Let T be a theory and let φ, ψ be formulas. $T \cup \{\varphi\} \vdash \psi$ iff there is an n such that $T \vdash \varphi^n \to \psi$ (where φ^n is $\varphi\& \ldots \&\varphi, n$ factors).

Now, following Hájek (1998b), we introduce some algebras corresponding to BL similarly as Boolean algebras correspond to classical logic.

A *residuated lattice* is an algebra

$$(L, \sqcap, \sqcup, *, \Rightarrow, 0, 1)$$

with four binary operations and two constants such that

(i) $(L, \sqcap, \sqcup, 0, 1)$ is a lattice with the largest element 1 and the least element 0 (with respect to the lattice ordering $\leq$);

(ii) $(L, *, 1)$ is a commutative semigroup with the unit element 1, i.e. $*$ is commutative, associative, and $1 * x = x$ for each $x \in L$; and

(iii) $*$ and $\Rightarrow$ form an adjoint pair, i.e $z \leq (x \Rightarrow y)$ iff $x * z \leq y$ for each x, y, $z \in L$.

A residuated lattice $(L, \sqcap, \sqcup, *, \Rightarrow, 0, 1)$ is a BL-*algebra* iff the following identities hold for each x, $y \in L$:

$$\begin{aligned}x \sqcap y &= x * (x \Rightarrow y),\\ (x \Rightarrow y) \sqcup (y \Rightarrow x) &= 1.\end{aligned}$$

A residuated lattice $(L, \sqcap, \sqcup, *, \Rightarrow, 0, 1)$ is *linearly ordered* if its lattice ordering is linear, i.e. for each pair x, $y \in L$ it holds that $x \sqcap y = x$ or $x \sqcap y = y$ (equivalent, $x \sqcup y = x$ or $x \sqcup y = y$). A linearly ordered residuated lattice is a BL-*algebra* iff the identity $x \sqcap y = x * (x \Rightarrow y)$ is true in it.

Let $\mathbf{L}= (L, \sqcap, \sqcup, *, \Rightarrow, 0, 1)$ be a BL-algebra. An $\mathbf{L}$-*interpretation* of propositional variables is a mapping $\mathbf{I}$ assigning to each propositional variable p an element $\mathbf{I}(p)$ of L. This extends in the obvious way to each formula using the operations on $\mathbf{L}$ as truth functions, i.e

$$\begin{aligned}
\mathbf{I}(\bar{0}) &= 0, \\
\mathbf{I}(\varphi \to \psi) &= \mathbf{I}(\varphi) \Rightarrow \mathbf{I}(\psi), \\
\mathbf{I}(\varphi \& \psi) &= \mathbf{I}(\varphi) * \mathbf{I}(\psi), \\
\mathbf{I}(\varphi \wedge \psi) &= \mathbf{I}(\varphi) \sqcap \mathbf{I}(\psi), \\
\mathbf{I}(\varphi \vee \psi) &= \mathbf{I}(\varphi) \sqcup \mathbf{I}(\psi), \\
\mathbf{I}(\neg\varphi) &= \mathbf{I}(\varphi) \Rightarrow 0.
\end{aligned}$$

A formula φ is an $\mathbf{L}$-tautology if $\mathbf{I}(\varphi) = 1$ for each $\mathbf{L}$-interpretation $\mathbf{I}$.

The logic BL is *sound* with respect to $\mathbf{L}$-tautologies: If φ is provable in BL, then φ is an $\mathbf{L}$-tautology for each BL-algebra $\mathbf{L}$. More generally, if T is a theory over BL and T proves φ then, for each BL-algebra $\mathbf{L}$ and each $\mathbf{L}$-interpretation $\mathbf{I}$ of propositional variables assigning the value 1 to each formula of T, $\mathbf{I}(\varphi) = 1$.

Finally, Hájek's original *completeness theorem* for BL reads as follows (Hájek, 1998b). For each formula φ the following three things are equivalent:

(i) φ is provable in BL;

(ii) for each linearly ordered BL-algebra $\mathbf{L}$, φ is an $\mathbf{L}$-tautology; and

(iii) for each BL-algebra $\mathbf{L}$, φ is an $\mathbf{L}$-tautology.

This result has been improved to get *standard completeness* for BL, i.e. completeness of BL with respect to standard BL-algebras in $[0, 1]$ – the real unit interval structures defined by continuous t-norms $*$ and their corresponding residuated implications $\Rightarrow$. First results were obtained by Hájek (1998a), but it was Cignoli et al. (2000) who finally proved that a formula φ is provable in BL iff $\mathbf{I}(\varphi) = 1$ for each interpretation $\mathbf{I}$ over standard BL-algebras in $[0, 1]$, i.e. iff φ is a 1-tautology for each continuous t-norm $*$ and its corresponding residuated implication $\Rightarrow$.

2.2.2 Gödel logic

In 1932, Gödel published an extremely short paper (Gödel, 1932) concerning intuitionistic logic (a subsystem of classical logic with a different meaning of connectives, e.g. $\varphi \vee \neg\varphi$ is not provable). Gödel's aim was to show that there is no finitely valued logic for which axioms of intuitionistic logic would be complete. For this purpose he created a semantics of (possibly infinite-valued) propositional calculus which is now called Gödel logic (denoted hereafter G).

Following Hájek (1998b), the language of G is built in the usual way from a (countable) set of propositional variables, a conjunction $\wedge$, an implication $\to$ and the truth constant $\overline{0}$. Further connectives are defined as follows:

$$\begin{array}{ll} \varphi \vee \psi & \text{is } ((\varphi \to \psi) \to \psi) \wedge ((\psi \to \varphi) \to \varphi), \\ \neg\varphi & \text{is } \varphi \to \bar{0}, \\ \varphi \equiv \psi & \text{is } (\varphi \to \psi) \wedge (\psi \to \varphi). \end{array}$$

The semantics of G is given by *interpretations* $\mathbf{I}$ of propositional variables into the unit interval $[0,1]$ which are extended to arbitrary formulas by means of the following rules:

$$\begin{aligned} \mathbf{I}(\overline{0}) &= 0, \\ \mathbf{I}(\varphi \wedge \psi) &= \min(\mathbf{I}(\varphi), \mathbf{I}(\psi)), \\ \mathbf{I}(\varphi \to \psi) &= \begin{cases} 1, & \text{if } \mathbf{I}(\varphi) \le \mathbf{I}(\psi) \\ \mathbf{I}(\psi), & \text{otherwise.} \end{cases} \end{aligned}$$

For the derived connectives, truth interpretations take these forms:

$$\begin{aligned} \mathbf{I}(\varphi \vee \psi) &= \max(\mathbf{I}(\varphi), \mathbf{I}(\psi)), \\ \mathbf{I}(\neg\varphi) &= \begin{cases} 1, & \text{if } \mathbf{I}(\varphi) = 0 \\ 0, & \text{otherwise,} \end{cases} \\ \mathbf{I}(\varphi \equiv \psi) &= \begin{cases} 1, & \text{if } \mathbf{I}(\varphi) = \mathbf{I}(\psi) \\ \min(\mathbf{I}(\varphi), \mathbf{I}(\psi)), & \text{otherwise.} \end{cases} \end{aligned}$$

The following is an axiomatization of G:

(A1) $(\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi))$
(A2) $(\varphi \wedge \psi) \to \varphi$
(A3) $(\varphi \wedge \psi) \to (\psi \wedge \varphi)$
(A4) $(\varphi \wedge (\varphi \to \psi)) \to (\psi \wedge (\psi \to \varphi))$
(A5) $(\varphi \to (\psi \to \chi)) \equiv ((\varphi \wedge \psi) \to \chi)$
(A6) $((\varphi \to \psi) \to \chi) \to (((\psi \to \varphi) \to \chi) \to \chi)$
(A7) $\overline{0} \to \varphi$
(A8) $\varphi \to \varphi \wedge \varphi$

Actually, logic G is equivalent to the extension of intuitionistic logic with the pre-linearity axiom $(\varphi \to \psi) \vee (\psi \to \varphi)$. Furthermore, one can show that G is equivalently axiomatized by BL plus $\varphi \to (\varphi \& \varphi)$ – idempotence of &. It follows easily that $\varphi \& \psi$ is equivalent to $\varphi \wedge \psi$, so that & is redundant.

The *deduction rule* of G is modus ponens. The notion of *proof*, denoted $\vdash_G$, is as usual (relative to G axioms). *Deduction theorem* is valid for G: $T \cup \{\varphi\} \vdash_G \psi$ iff $T \vdash_G (\varphi \to \psi)$. G is *the only* PC($*$) having the deduction theorem, i.e. if PC($*$) satisfies the deduction theorem then $*$ is minimum.

Algebras for G (called G-*algebras*) are BL-algebras satisfying the identity $x * x = x$, i.e. with idempotent multiplication.

Completeness for G reads as follows: φ is provable in G iff $\mathbf{I}(\varphi) = 1$ for each interpretation $\mathbf{I}$. Furthermore, G enjoys *strong completeness*. Namely, let T be

an arbitrary theory over G. An interpretation $\mathbf{I}$ is a model of T iff $\mathbf{I}(\psi) = 1$ for each $\psi \in T$. Then, T proves φ iff $\mathbf{I}(\varphi) = 1$ for each interpretation $\mathbf{I}$ which is a model of T.

An interesting extension of Gödel logic is the so-called Rational Gödel logic (denoted hereafter RGL), where for each rational $r \in [0, 1]$ a truth constant $\bar{r}$ is introduced into the language[1]. *Interpretations* $\mathbf{I}$ are then extended to RGL formulas by requiring $\mathbf{I}(\bar{r}) = r$, for each rational $r \in [0, 1]$. *Axioms* of RGL are those of G plus the following book-keeping axioms for truth constants:

(RGL1) $\bar{r} \wedge \bar{s} \equiv \overline{\min(r, s)}$
(RGL2) $\bar{r} \rightarrow \bar{s} \equiv \overline{r \Rightarrow s}$

where $\Rightarrow$ is Gödel implication function.

For RGL one cannot expect a Pavelka-style completeness (i.e. provability degree = truth degree), since this style of completeness strongly relies on the continuity of the truth functions, and it is obvious that in RGL, the truth function of implication is not continuous. In (De Baets et al., 2001), the authors proved the following *finite strong completeness* result by adapting an analogous theorem proposed in (Esteva et al., 2000) for an extension of RGL: Let T be a finite theory over RGL. Then, T proves φ iff $\mathbf{I}(\varphi) = 1$ for each interpretation $\mathbf{I}$ which is a model of T. In particular, T proves $\bar{r} \rightarrow \varphi$ iff $\mathbf{I}(\varphi) \geq r$ for each interpretation $\mathbf{I}$ which is a model of T.

One potential drawback of Gödel logic to be used for approximate reasoning is that it lacks an involutive negation. In (Esteva et al., 2000), the authors studied residuated logics arising from extending SBL logic (BL logic with Gödel negation) with a new involutive negation connective $\sim$. In this framework, they define an axiomatization of Gödel logic G extended with connectives $\sim$ and Δ (called $\mathrm{G}_\sim$), where $\Delta\varphi$ is $\neg \sim \varphi$. They show that $\mathrm{G}_\sim$ is complete with respect to the so-called *standard* $\mathrm{G}_\sim$*-algebra*, the unit interval $[0, 1]$ equipped with Gödel's truth functions and with the involutive negation $\sim x = 1 - x$. Furthermore, they also consider the extension of $\mathrm{G}_\sim$ with rational truth constants preserving completeness results.

Next, we develop Gödel predicate logic (denoted hereafter G$\forall$). Following Hájek (1998b), the language of G$\forall$ consists of

- object variables;
- object constants;
- propositional variables;
- predicates symbols each together with its arity;
- connectives $\wedge$, $\rightarrow$;
- truth constants $\bar{0}$, $\bar{1}$; and

[1] In the same spirit as what was originally done by Pavelka (1979) with the infinitely-valued Łukasiewicz logic and later improved by Hájek (1998b).

- quantifiers $\forall$, $\exists$.

Other connectives ($\vee$, $\neg$, $\equiv$) are defined as in propositional calculus.

Terms are object variables and object constants. *Atomic formulas* are either of the form $p(t_1, \ldots, t_n)$, where p is a predicate symbol of arity n and $t_1, \ldots, t_n$ are terms, or just propositional variables. If φ and ψ are formulas and x is an object variable, then $\varphi \wedge \psi$, $\varphi \to \psi$, $(\forall x)\psi$, $(\exists x)\varphi$, $\bar{0}$ and $\bar{1}$ are formulas; each formula results from atomic formulas by iterated use of this rule.

An *interpretation* has the form

$$\mathbf{M} = (U, i, m),$$

where

- U is a non-empty domain (or sets of domains if we have sorts);
- i maps each propositional variable into a truth value of the unit interval $[0, 1]$ and each predicate symbol p of arity n into a n-ary relation on U $i(p) : U^n \to [0, 1]$; and
- m maps each object constant into an element of U.

An *evaluation* of object variables is a mapping v assigning to each object variable x an element $v(x) \in U$. The *truth value* of an atomic formula φ under an interpretation $\mathbf{M}$ and an evaluation of variables v, denoted $\|\varphi\|_{\mathbf{M},v}$, is just $i(p)$ if φ is a propositional variable p, and if φ is of the form $p(t_1, \ldots, t_n)$, then it is computed as

$$\|p(t_1, \ldots, t_n)\|_{\mathbf{M},v} = i(p)(\|t_1\|_{\mathbf{M},v}, \ldots, \|t_n\|_{\mathbf{M},v}),$$

where, for $j = 1, \ldots, n$, $\|t_j\|_{\mathbf{M},v}$ is $v(x)$ if t_j is an object variable x, and it is $m(c)$ if t_j is an object constant c.

The above truth value extends to the value $\|\varphi\|_{\mathbf{M},v}$, for each G$\forall$ formula φ, in the usual manner with truth functions of Gödel logic:

$$\begin{array}{rcl}
\|\bar{1}\|_{\mathbf{M},v} & = & 1, \\
\|\bar{0}\|_{\mathbf{M},v} & = & 0, \\
\|\varphi \wedge \psi\|_{\mathbf{M},v} & = & \min(\|\varphi\|_{\mathbf{M},v}, \|\psi\|_{\mathbf{M},v}), \\
\|\varphi \to \psi\|_{\mathbf{M},v} & = & \begin{cases} 1, & \text{if } \|\varphi\|_{\mathbf{M},v} \leq \|\psi\|_{\mathbf{M},v} \\ \|\psi\|_{\mathbf{M},v}, & \text{otherwise,} \end{cases} \\
\|(\forall x)\varphi\|_{\mathbf{M},v} & = & \inf\{\|\varphi\|_{\mathbf{M},v'} \mid v'(y) = v(y) \text{ for each variable } y, \text{ except } x\}, \\
\|(\exists x)\varphi\|_{\mathbf{M},v} & = & \sup\{\|\varphi\|_{\mathbf{M},v'} \mid v'(y) = v(y) \text{ for each variable } y, \text{ except } x\}.
\end{array}$$

The truth value of a G$\forall$ formula φ in $\mathbf{M}$ is

$$\|\varphi\|_{\mathbf{M}} = \inf\{\|\varphi\|_{\mathbf{M},v} \mid v \text{ is an evaluation of variables}\}.$$

Logical axioms of G$\forall$ are those of Gödel propositional logic plus the logical axioms on quantifiers:

$(\forall 1)$ $(\forall x)\varphi(x) \to \varphi(t)$ (t substitutable for x in $\varphi(x)$)
$(\exists 1)$ $\varphi(t) \to (\exists x)\varphi(x)$ (t substitutable for x in $\varphi(x)$)
$(\forall 2)$ $(\forall x)(\chi \to \varphi) \to (\chi \to (\forall x)\varphi)$ (x not free in χ)
$(\exists 2)$ $(\forall x)(\varphi \to \chi) \to ((\exists x)\varphi \to \chi)$ (x not free in χ)
$(\forall 3)$ $(\forall x)(\varphi \vee \chi) \to ((\forall x)\varphi \vee \chi)$ (x not free in χ)

Deduction rules are modus ponens and generalization (from φ derive $(\forall x)\varphi$). *Completeness* for G$\forall$ reads as follows: Let T be an arbitrary theory over G$\forall$. An interpretation $\mathbf{M}$ is a model of T iff $\|\psi\|_{\mathbf{M}} = 1$ for each $\psi \in T$. Then, T proves φ iff $\|\varphi\|_{\mathbf{M}} = 1$ for each interpretation $\mathbf{M}$ which is a model of T.

2.2.3 Łukasiewicz logic

The language of Łukasiewicz propositional logic (denoted hereafter Ł) is built in the usual way from a (countable) set of propositional variables, an implication $\to$ ($x \Rightarrow y = \min(1, 1 - x + y)$), and the truth constant $\overline{0}$. Negation and (strong) conjunction are defined as follows:

$$\begin{array}{lll} \neg\varphi & \text{is } \varphi \to \overline{0} & (\neg x = x \Rightarrow 0 = 1 - x), \\ \varphi\&\psi & \text{is } \neg(\varphi \to \neg\psi) & (x * y = max(x + y - 1, 0)). \end{array}$$

Conjunction, disjunction and (strong) disjunction are defined as follows:

$$\begin{array}{lll} \varphi \wedge \psi & \text{is } \varphi\&(\varphi \to \psi) & (x \sqcap y = min(x, y)), \\ \varphi \vee \psi & \text{is } (\varphi \to \psi) \to \psi & (x \sqcup y = max(x, y)), \\ \varphi \underline{\vee} \psi & \text{is } \neg\varphi \to \psi & (x \underline{\sqcup} y = min(x + y, 1)). \end{array}$$

The following are the original axioms of Ł (Łukasiewicz, 1970):

(L1) $\varphi \to (\psi \to \varphi)$
(L2) $(\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi))$
(L3) $(\neg\psi \to \neg\varphi) \to (\varphi \to \psi)$
(L4) $((\varphi \to \psi) \to \psi) \to ((\psi \to \varphi) \to \varphi)$

Completeness of this set of axioms was conjectured by Łukasiewicz in the thirties, but first proved by Rose and Rosser (1958).

Łukasiewicz propositional calculus can be understood as a schematic extension of BL by the schema

$$\neg\neg\varphi \to \varphi.$$

Algebras for Ł (called MV-*algebras*) are BL-algebras in which the identity $x = ((x \Rightarrow 0) \Rightarrow 0)$ is valid, and Ł is complete with respect to the standard MV-algebra in $[0, 1]$, i.e. the algebra defined by $[0, 1]$ being the continuous t-norm $*$ the Łukasiewicz t-norm ($x * y = max(x + y - 1, 0)$) and $\Rightarrow$ its corresponding residuated implication.

2.2.4 Product logic

The logic based on the product t-norm has been considerably less investigated than the two preceding ones (see Alsina et al. (1983)). Product logic was introduced by Hájek et al. (1996).

The language of product propositional logic (denoted hereafter Π) is built in the usual way from a (countable) set of propositional variables, a (strong) conjunction $\odot$, an implication $\rightarrow$ and the truth constant $\overline{0}$.

The axioms of Π are those of BL plus:

(Π1) $\quad \neg\neg\chi \rightarrow ((\varphi \odot \chi \rightarrow \psi \odot \chi) \rightarrow (\varphi \rightarrow \psi))$
(Π2) $\quad \varphi \wedge \neg\varphi \rightarrow \bar{0}$

The axiom (Π2) can be equivalently replaced by each of the following formulas:

$$\neg(\varphi \odot \varphi) \rightarrow \neg\varphi,$$
$$(\varphi \rightarrow \neg\varphi) \rightarrow \neg\varphi,$$
$$\neg\varphi \vee \neg\neg\varphi.$$

Product algebras (or Π-*algebras*) are BL-algebras satisfying

$$\neg\neg z \leq ((x * z \Rightarrow y * z) \Rightarrow (x \Rightarrow y)),$$
$$x \sqcap \neg x = 0.$$

Hájek et al. (1996) have proved completeness theorem with respect to the class of Π-algebras. Moreover, standard completeness with respect to the standard Π-algebra in $[0,1]$ has also been proved, i.e. the algebra defined by $[0,1]$ being the continuous t-norm $*$ the usual product and $\Rightarrow$ its corresponding residuated implication. Baaz et al. (1998) have showed that Łukasiewicz logic has a faithful interpretation in product logic.

2.3 Resolution-based systems of fuzzy logic

In the logical systems presented in the last section, the notion of proof was based on modus ponens in the setting of a Hilbert-like axiomatization. A very different family of fuzzy logics uses the $(\max, \min, 1 - \cdot)$ triple for modeling disjunction, conjunction and negation, the set of truth values is the unit interval $[0,1]$, and the notion of proof is related to some resolution rule, i.e. to compute the truth value $\|\psi \vee \chi\|$ from $\|\varphi \vee \psi\|$ and $\|\neg\varphi \vee \chi\|$. This trend, initiated by Lee (1972), has blossomed in the framework of logic programming and preserves simple clausal forms. Klement and Navara (1999) have compared these two fuzzy logic traditions.

The first fuzzy resolution method was defined by Lee (1972). At the syntactic level, formulas are *classical first-order formulas*, but at the semantic level, formulas have a *truth value* which may be intermediary between 0 and 1. An *interpretation* $\mathbf{M}$ is defined by an assignment of a truth value to each atomic formula, from which truth values of compound formulas are computed in the following way:

$$\begin{aligned} \|\neg\varphi\|_{\mathbf{M}} &= 1 - \|\varphi\|_{\mathbf{M}}, \\ \|\varphi \wedge \psi\|_{\mathbf{M}} &= \min(\|\varphi\|_{\mathbf{M}}, \|\psi\|_{\mathbf{M}}), \\ \|\varphi \vee \psi\|_{\mathbf{M}} &= \max(\|\varphi\|_{\mathbf{M}}, \|\psi\|_{\mathbf{M}}). \end{aligned}$$

The notions of validity, consistency and inconsistency are generalized to fuzzy logic: Let φ be a fuzzy formula. φ is *valid* iff $\|\varphi\|_{\mathbf{M}} \geq 0.5$ for each interpretation $\mathbf{M}$, i.e the set of designated truth values is $[0.5, 1]$. φ is *inconsistent* iff $\|\varphi\|_{\mathbf{M}} \leq 0.5$ for each interpretation $\mathbf{M}$. And, φ *entails* another formula ψ, denoted $\varphi \models \psi$, if $\|\psi\|_{\mathbf{M}} \geq 0.5$ for each interpretation $\mathbf{M}$ such that $\|\varphi\|_{\mathbf{M}} \geq 0.5$. Lee and Chang (1971) proved that a fuzzy formula is valid (respec. inconsistent) iff the formula is classically valid (respectively, inconsistent), i.e. considering the involved predicates and propositions as crisp; and that $\varphi \models \psi$ in fuzzy logic iff $\varphi \models \psi$ in classical logic. The resolvent of two clauses C_1 and C_2 is defined as in classical first-order logic. Lee (1972) proved that provided that C_1 and C_2 are ground clauses, and if $\min(\|C_1\|, \|C_2\|) = a > 0.5$ and $\max(\|C_1\|, \|C_2\|) = b$, then $a \leq \|R(C_1, C_2)\| \leq b$ for each resolvent $R(C_1, C_2)$ of C_1 and C_2. This is generalized to resolvents of a set of ground clauses obtained by a number of successive applications of the resolution principle. Hence, Lee's resolution is *sound*. This result also holds for intervals of truth values with a lower bound greater than 0.5. Lee's proof method does not deal with refutation, hence it is not complete (since resolution is not complete for deduction).

Many subsequent works have been based on Lee's definitions and results. In (Shen et al., 1988; Mukaidono et al., 1989) Lee's resolution principle was generalized by introducing a *fuzzy resolvent*. Let C_1 and C_2 be two clauses of fuzzy logic and let $R(C_1, C_2)$ be a classical resolvent of C_1 and C_2. Let l be the literal on the basis of which $R(C_1, C_2)$ has been obtained. Then, the *fuzzy resolvent* of C_1 and C_2 is $R(C_1, C_2) \vee (l \wedge \neg l)$ with the truth value $\max(\|R(C_1, C_2)\|, \|(l \wedge \neg l)\|)$. It is proved that a fuzzy resolvent is always a logical consequence of its parent clauses, which generalizes Lee's result. One of the drawbacks of this approach is the necessity to know a concrete interpretation using which $\|\cdot\|$ can be computed. Also, the problem is that this approach –and the ones based on elaboration of these ideas, e.g. (Bénéjam, 1986; Liau and Lin, 1988; Orlowska and Wierzchon, 1985; Yager, 1985)– is based on the language of classical logic, and thus, does not make it possible to deal with intermediate truth values at the syntactic level.

This is not the case of the *operator fuzzy logic* defined in (Liu and Xiao, 1985; Liu, 1989; Weigert et al., 1993), where formulas may be prefixed by a valuation λ which takes its value in a lattice and imbrication of valuations is admitted as in modal languages. For instance, $\lambda\varphi$, $\lambda\alpha\varphi$ and $\lambda(\varphi \wedge \alpha\psi)$ are well-formed formulas in this logic with valuations λ and α. Operator fuzzy logic is truth-functional not only for the usual connectives $\vee$, $\wedge$ and $\neg$ but also, for the imbrication of 'λ' operators. Thus, $\|\lambda\varphi\| = \lambda \circ \|\varphi\|$, where $\circ$ is a fuzzification operator which may have various interpretations (the authors propose arithmetic mean). This enables to grade inconsistency as follows. A formula φ of operator fuzzy logic is λ-*inconsistent* iff $\|\varphi\|_{\mathbf{M}} \leq \lambda$ for every crisp interpretation $\mathbf{M}$. Hence, the greater λ, the less inconsistent the formula is. Moreover, if $\lambda_{\min}$ is the minimum value for which φ is λ-inconsistent, the value $1 - \lambda_{\min}$ can be seen as a measure to which degree φ is inconsistent (maximal inconsistency corresponds to $\lambda_{\min} = 0$). A formula can be transformed into an equivalent set of clauses of the form

$\lambda_1 \dots \lambda_n l$, where l is a literal. A *λ-resolution* is defined in the following way: The higher λ, where λ acts as a threshold, the more contradictory (in terms of truth values) the complementary literals must be to enable λ-resolution (a too high threshold λ inhibits resolutions with insufficient contradictions). It is proved that a set of clauses T is λ-inconsistent iff we can get a λ-empty clause by λ-resolution from T.

In the framework of the logical systems presented in Section 2.2, a resolution rule for Łukasiewicz-based logics has been proposed in (Thiele and Lehmke, 1994; Lehmke, 1995; Klawonn and Kruse, 1994; Klawonn, 1995).

Lehmke and Thiele defined a resolution system for so-called *weighted bold clauses*. Clauses are of the form $C = l_1 \nabla \cdots \nabla l_n$, where l_i are literals in classical way (they consider only propositional logic) and ∇ is the Łukasiewicz (strong) disjunction (i.e. $\|C_1 \nabla C_2\| = \min(\|C_1\| + \|C_2\|, 1)$). They introduce the resolution rule as follows:

$$\frac{T \vdash C_1 \text{ and } p \text{ occurs in } C_1 \qquad T \vdash C_2 \text{ and } \neg p \text{ occurs in } C_2}{T \vdash ((C_1 \nabla C_2) \backslash p) \backslash \neg p},$$

where $\backslash$ denotes the operation of omitting the corresponding literal. Then, they get the following result:

$$\text{If } T \vdash C \text{ then } T \models C, \text{ and if } T \text{ has no } \mathbf{1}\text{-model then } T \vdash \square.$$

Klawonn and Kruse turned to predicate fuzzy logic. They introduce special implication clauses of the form $(\forall x_1 \dots x_n)(\varphi \Rightarrow A)$ and $(\forall x_1 \dots x_n)\varphi$, where A is an atomic formula and φ contains only "and" and "or" kinds of connectives and no quantifiers. Only a finite set of truth values is considered and implication clauses do not allow existential quantifiers. They introduce the concept of a fuzzy theory T, called here Ł-Prolog program, as a set of clauses over the set of all implication clauses F_0. Then, the resolution mechanism is defined in the following way: Let T_1 and T_2 be two Ł-Prolog programs. T_2 is *directly derivable* from T_1 if it is constructed in the following way:

1. If there is an implication clause $(\forall x_1 \dots x_n)(\varphi \Rightarrow A)$, then

$$\begin{aligned} T_2((\forall x_1 \dots x_n)(\varphi \Rightarrow A)) \;=\; & T_1((\forall x_1 \dots x_n)A) \vee \\ & T_1((\forall x_1 \dots x_n)\varphi) \& \\ & T_1((\forall x_1 \dots x_n)(\varphi \Rightarrow A)) \end{aligned}$$

   and $T_2((\forall x_1 \dots x_n)B) = T_1((\forall x_1 \dots x_n)B)$, for $B \neq A$.

2. If there is an implication clause $(\forall x_1 \dots x_n)A$ and terms $t_{i_1} \dots t_{i_r}$, $i_1, \dots, i_r \in \{1, \dots, n\}$, without variables, then A' is obtained by substituting $t_{i_1} \dots t_{i_r}$ for the variables $x_{i_1} \dots x_{i_r}$ and quantifying the remaining ones and we put $T_2(A') = T_1((\forall x_1 \dots x_n)A) \vee T_1(A')$.

On the basis of these concepts, the authors prove a *completeness theorem*: Let T be an Ł-Prolog program. $T \vdash^0_\alpha (\forall x_1 \dots x_n)\varphi$ iff $T \models^0_\alpha (\forall x_1 \dots x_n)\varphi$, where $\vdash^0_\alpha$ ($\models^0_\alpha$) means that φ is derivable (respectively, true) at least to the degree α.

Novák (1999) points out that there are two main problems related with resolution-based systems of fuzzy logic. On the one hand, a satisfactory solution of the resolution principle in fuzzy logic has been limited till now because of the lack of the formulation of a generalization of the Herbrand theorem, which is the basis of the resolution principle in classical logic. A significant step in this direction has been obtained by Novák and Perfilieva (2000) where the fuzzy version of the Herbrand theorem has been proved. On the other hand, algorithmization of the refutation procedure in fuzzy logic is a difficult problem because, in general, conjunction and disjunction operations are not distributive, and thus, it is difficult to find a universal representation of formulas in fuzzy logic in a normal conjunctive form style, as is the case of classical logic. Novák (1999) states that a solution of this problem can be expected from McNaughton (1951) theorem in Lukasiewicz logic, which establishes that every formula can be represented by a partially linear function on $[0,1]$. An algorithmic proof of it has been given by Mundici (1994) and Tonis and Perfilieva (2000).

A particular case of resolution of fuzzy logic is based on extending the modus ponens inference rule to formulas with degrees of truth. The problem is to compute the truth value $\|\psi\|$ from $\|\varphi\|$ and $\|\varphi \to \psi\|$. In this framework, in (Valverde and Trillas, 1985; Godo, 1990) a family of infinitely-valued logics was studied. These logics are chiefly characterized by the following points: the interpretation function of the conjunction connective has to fulfill the properties of t-norms, the implication connective is defined by residuation with respect to the conjunction connective, and the interpretation function of the negation connective is the unit complement function. They show that, in this case, the t-norm that gives meaning to the conjunction connective is a modus ponens generating function for the implication. Thus, let $\mathbf{I}$ be an interpretation (a mapping that assigns to every propositional variable a truth value from the unit interval $[0,1]$) and let T be a set of weighted formulas of the form

$$\begin{array}{l}(\varphi_1, \alpha_1)\\ \cdots\cdots\\ (\varphi_n, \alpha_n)\\ (\varphi_1 \wedge \cdots \wedge \varphi_n \to \psi, \alpha_R).\end{array}$$

Then,

$$\inf\{\mathbf{I}(\psi) \mid \mathbf{I} \models T\} = *(\alpha_R, \alpha_1, \ldots, \alpha_n),$$

where

$$\begin{array}{l}\mathbf{I} \models T \text{ iff } \mathbf{I}(\varphi_1 \wedge \cdots \wedge \varphi_n \to \psi) \geq \alpha_R \text{ and } \mathbf{I}(\varphi_i) \geq \alpha_i \text{ for } i = 1, \ldots, n,\\ \mathbf{I}(\varphi_1 \wedge \cdots \wedge \varphi_n) = *(\mathbf{I}(\varphi_1), \ldots, \mathbf{I}(\varphi_n)),\\ \mathbf{I}(\varphi \to \psi) = \mathbf{I}(\varphi) \Rightarrow \mathbf{I}(\psi),\end{array}$$

$\Rightarrow$ being the residuated implication of a t-norm $*$. Based on this result, a complete calculus for deducing infinitely-valued propositional atomic formulas from a set of infinitely-valued Horn clauses was defined in (Escalada-Imaz and Manyà, 1995; Manyà, 1996). They design an efficient proof procedure that could act as a propositional interpreter for infinitely-valued propositional logic

programs. Some logic programming extensions as a negation as failure rule and a cut operator are also considered. Notice that when using a modus ponens inference rule in the frame of Horn clauses, we do not need any transformation into normal forms.

A refutation proof method by resolution for a first-order language based on Horn clauses with fuzzy predicates (called fuzzy Horn clauses) was defined in (Mukaidono and Yasui, 1994; Yasui et al., 1995; Yasui and Mukaidono, 1996). They define the following resolution rule: If there exists a substitution θ such that $B_1\theta = B'_1\theta, \ldots, B_n\theta = B'_n\theta$, then the resolvent of $B'_1, \ldots, B'_n$ and $A \leftarrow B_1, \ldots B_n$ is $A\theta$, and its truth value is calculated as

$$\|A\theta\| = \|A\theta \leftarrow B_1\theta, \ldots B_n\theta\| * (\|B'_1\theta\| \wedge \ldots \wedge \|B'_n\theta\|),$$

where $*$ is the composition operation of truth values of premises and $\wedge$ is a fuzzy logic conjunction operator.

A proof method, based on a generalized modus ponens rule for a truth-functional fuzzy logic with arbitrary/flexible t-operators has been developed in (Vojtáš and Paulik, 1996; Vojtáš, 1998; Vojtáš, 2001a). The language consists of propositional variables and several conjunctions, disjunctions and implications which are interpreted as t-norms, t-conorms and residuated implications, respectively. Completeness is proved in the framework of a declarative semantics based on fixpoint theory.

Concerning resolution-based theorem proving in finitely-valued propositional logics, Baaz and Fermüller (1995) suggested a two-level approach. The first level consists of translating arbitrary formulas of a particular finitely-valued propositional logic into clause form. Given the definition of a finitely-valued logic by means of the truth tables of its connectives, it can be devised a *logic-dependent* translation calculus that derives a clause from a formula of the source logic. This translation can be done in such a way that the language of the clause forms is independent from the particular language of the source logic except for the number of truth values, i.e. clause forms are *logic-independent*. The second level consists of the application of a logic-independent resolution calculus to the clause forms obtained in the first level.

These logic-independent clause forms are known as *signed formulas in conjunctive normal form* (signed CNF formulas). A signed CNF formula is a set of signed clauses and a signed clause is a set of signed literals. A signed literal is an expression of the form $S{:}p$, where S, called the sign of the literal, is a subset of the truth value set and p is a propositional atom. An interpretation is mapping that assigns to every propositional atom an element of the truth value set. An interpretation satisfies a signed literal $S{:}p$ iff it assigns to p a truth value that appears in S. An interpretation satisfies a signed clause iff it satisfies at least one of its literals. A signed CNF formula is satisfiable iff there exists an interpretation that satisfies all its clauses; otherwise, it is unsatisfiable.

One significant subclass of signed CNF formulas is the class of *regular CNF formulas*. Roughly speaking, regular CNF formulas are those signed CNF formulas such that the signs of the literals are either of the form $\{j \in N \mid j \geq i\}$ or

of the form $\{j \in N \mid j \leq i\}$, where N is the truth value set, $\leq$ is a total order on N and $i, j \in N$. Another subclass is the class of *monosigned CNF formulas*. They are those signed CNF formulas whose signs are singletons instead of arbitrary subsets of the truth value set.

Two remarkable contributions to the first level of Baaz and Fermüller's approach to resolution-based theorem proving are the following ones:

- The system MUltlog (Baaz et al., 1993; Baaz et al., 1996) which computes an optimized logic-dependant translation calculus for deriving signed CNF formulas from the definition of the operators and quantifiers of any finitely-valued first-order logic. It produces a sequent calculus and a natural deduction system as well.

- Hähnle's method for deriving a satisfiability equivalent signed CNF formula from an arbitrary formula of any finitely-valued logic (Hähnle, 1994b). A formula A is satisfiability equivalent to a formula B if it holds that A is satisfiable iff B is satisfiable. This property is weaker than logical equivalence. For proofs by refutation, satisfiability equivalence suffices. He has described a structure-preserving method for generating short conjunctive normal forms (i.e. short signed CNF formulas) whose length is linear in the length of the input formula and polynomial in the cardinality of the truth value set.

Concerning the second level, there exist several resolution calculus for signed CNF formulas and their subclasses. The first resolution calculus for signed CNF formulas was defined by Murray and Rosenthal (1993) which consists of the following *signed parallel resolution*, *merging* and *simplification* rules:

$$\frac{S_1{:}p \vee C_1 \quad \cdots \quad S_m{:}p \vee C_m}{(S_1 \cap \cdots \cap S_m){:}p \vee C_1 \vee \cdots \vee C_m},$$

$$\frac{S_1{:}p \vee S_2{:}p \vee C}{(S_1 \cup S_2){:}p \vee C},$$

$$\frac{\emptyset{:}p \vee C}{C}.$$

Subsequently, Hähnle (1994b) defined a sequential refutation complete resolution calculus for signed CNF formulas which consists of the above simplification rule and the following *signed binary resolution* rule:

$$\frac{S_1{:}p \vee C_1 \qquad S_2{:}p \vee C_2}{(S_1 \cap S_2){:}p \vee C_1 \vee C_2}.$$

Completeness is proved using a straightforward generalization of classical semantic trees to the multiple-valued context. Hähnle (1996) points out that for obtaining refutation completeness the merging rule is not necessary.

The fact of developing separately inference rules for regular clauses appears to be very beneficial for obtaining efficient proof procedures for regular CNF

formulas. On the one hand, all regular literals have either positive or negative polarity; on the other hand, any unsatisfiable set of regular literals has an unsatisfiable subset of cardinality two, contrary to what happens with signed literals. These points enable to define refinements of signed resolution calculi which are complete for regular CNF formulas and bear a close resemblance to classical resolution versions. In this frame, Hähnle (1994b) proved that the following *regular* version of the resolution rule is refutation complete for regular CNF formulas:

$$\frac{\boxed{\geq i_1} : p \vee C_1 \quad \cdots \quad \boxed{\geq i_m} : p \vee C_m \quad \boxed{\leq j} : p \vee C}{C_1 \vee \cdots \vee C_m \vee C} \quad \text{if } \max_{1 \leq k \leq m} i_k > j.$$

As we have already noted, the merging rule is not necessary for obtaining completeness. Later, Hähnle (1996) defined the following complete regular version of *negative hyperresolution*:

$$\frac{\boxed{\leq i_1} : p_1 \vee C_1 \quad \cdots \quad \boxed{\leq i_m} : p_m \vee C_m \qquad \boxed{\geq j_1} : p_1 \vee \cdots \vee \boxed{\geq j_m} : p_m \vee C}{C_1 \vee \cdots \vee C_m \vee C}$$

provided $m \geq 1$, $i_l < j_l$ for each $1 \leq l \leq m$, and $C_1, \ldots, C_m, C$ contain only negative literals.

Manyà (1996) proved that for obtaining completeness for regular Horn formulas it is enough to resolve on regular positive unit clauses. Then, the following rule is defined as a refutation complete calculus for regular Horn formulas:

$$\frac{\begin{array}{l}\boxed{\geq j} : p \\ \boxed{\leq i} : p \vee C\end{array}}{C} \quad \text{if } i < j.$$

Beckert et al. (2000) showed that each signed CNF formula can be reduced to a satisfiability equivalent regular formula which is only polynomially larger.

In (De Baets et al., 2001), the authors considered the sublanguage $\mathcal{L}_Z$ of $\mathrm{G}_\sim$ (Gödel logic with involution) built from the set of propositional variables and only with the connectives $\wedge$, $\vee$ and $\sim$. They introduce truth constants by considering weighted formulas as pairs (φ, α), where $\varphi \in \mathcal{L}_Z$ and α is a rational from $[0,1]$ understood as a lower bound for the truth value of φ. In this framework, they formalize a semantics and define a transformation $\#$ which maps a set of weighted $\mathcal{L}_Z$-formulas T into a set $T^\#$ consisting only of weighted $\mathcal{L}_Z$-clauses and semantically equivalent to T. Then, they show how weighted $\mathcal{L}_Z$-clauses can be translated into regular clauses, and the regular version of the resolution rule is used for defining a complete proof method by refutation for weighted $\mathcal{L}_Z$-formulas.

Baaz and Fermüller (1995) studied monosigned resolution and proved that only one inference rule is needed to determine the satisfiability of monosigned CNF formulas. This rule is very close to classical binary resolution:

$$\frac{\{v_1\} : p \vee C_1 \qquad \{v_2\} : p \vee C_2}{C_1 \vee C_2} \quad \text{if } v_1 \neq v_2 \text{ with } v_1, v_2 \in N.$$

There exist some automated theorem provers for multiple-valued logics. The most representative among them is $_3T^4P$ (Beckert et al., 1996), developed at the University of Karlsruhe. It is a generic tableau-based theorem prover for finitely-valued first-order logics with sorts and equality; it was implemented in Prolog. Surveys on deduction methods for many-valued logics that contain results of signed and regular CNF formulas are (Hähnle, 1994a; Hähnle and Escalada-Imaz, 1997; Hähnle, 2001). The reader interested in an homogeneous exposition and the main references of earlier approaches to multiple-valued resolution not based on signed clauses is invited to consult Hähnle (1994a, Chapter 8). Among the resolution systems examined there are those of Morgan (1976), Orłowska (1978) and Di Zenzo (1988) for Post logics, that of Schmitt (1986) for a specific three-valued logic for use within a natural language dialogue system, those of Stachniak and O'Hearn (1990) for a wide class of multiple-valued logics and those of da Costa et al. (1990) and Lu et al. (1991) for paraconsistent logics. The theory of many-valued non-clausal resolution is fully developed in (Stachniak, 1996).

2.4 Rule-based systems

In this section, we give a brief overview of the use of rule-based systems and of the problems related with the interpretation of weights attached to logical formulas or symbolic expressions.

In the seventies, the so called *rule-based* or *expert* systems appeared which started a new and very successful field of research in the theory of artificial intelligence. The first one was MYCIN developed by Shortliffe and Buchanan (1975). The general idea of expert systems is to develop an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their resolution (Feingenbaum, 1987). According to Dubois et al. (1991a) and Bouchon-Meunier et al. (1999), a so-called "certainty" factor ranging on a numerical scale is attached to each "if ... then" rule and to each fact in the knowledge base. Then, combination and propagation operations enable the system to compute, in a compositional way, (i) the certainty factor estimated to what extent a (compound) condition is satisfied from the certainty factors attached to the satisfaction of elementary conditions; (ii) the certainty factor attached to the conclusion of a rule from the ones attached to the satisfaction of its condition part and to the rule itself; and (iii) the certainty factor of a fact obtained by fusing partial conclusions, obtained in different ways, but pertaining to the same matter.

Max and min operations were used in MYCIN (Buchanan and Shortliffe, 1984), as well as in PROSPECTOR (Duda et al., 1981), for the evaluation of disjunctive and conjunctive conditions, respectively. Fuzzy set researchers have tried to improve the MYCIN approach to rule-based inference systems by allowing fuzzy conditions in rules, applying fuzzy connectives for the combination and the propagation of the degrees attached to pieces of knowledge, and allowing fuzzy degrees rather than taking for granted that precise degrees were always

available.

When fuzzy conditions are allowed in rules and the available information in the knowledge base is also fuzzy, a fuzzy pattern matching problem arises for estimating to what extent the condition of rules is satisfied, and thus, the partial matching between fuzzy events is a matter of degree as defined in (Yager, 1980). The partial matching between fuzzy events can be approximated by degrees of possibility and necessity as proposed by Cayrol et al. (1982) and Bel et al. (1986), or by degrees of inclusion. Magrez and Smets (1989) proposed to use a degree of inclusion based on Łukasiewicz implication. Cayrol et al. (1982) and Prade (1982) proposed to extended necessity measures of matching, and Baldwin and Pilsworth (1979) possibility measures, to independent multiple conditions using minimum and maximum for conjunction and disjunction of conditions, respectively. In (Dubois et al., 1988b; Sanchez, 1989; Yager, 1989), the authors proposed to handle unequally important conditions using weighted operations, applied to pattern-matching in information retrieval. Finally, Lesmo et al. (1983) and Dubois (1989a) studied the case of several fuzzy productions rules concluding on a same fact or decision. Another application of fuzzy rule-based systems is similarity-based reasoning. In these systems, analyzed by Dubois and Prade (1991c), the pattern matching problem comes down to compute a distance between fuzzy events.

Combination and propagation issues have been investigated for rule-based systems using fully compositional connectives operations. However, several drawbacks and methodological problems may be encountered (see Dubois and Prade (1989b) and Léa Sombé, (Besnard et al. (1990, Chapter 2.1) for detailed discussions). The certainty factors were originally understood as probabilities but treated extensionally. Later on, it has been found that expert manipulation corresponds to understanding certainty factors as beliefs. However, in both cases, extensionality cannot be accepted. Another thing is a question whether fuzzy logic (logic of truth) could be helpful for rule-based systems. The weights can be understood as truth degrees and then, a logically-based procedure can be derived as proposed in (Ivánek, 1991; Novák and Ivánek, 1995b). In (Dubois et al., 1991a; Bouchon-Meunier et al., 1999) several inference patterns are analyzed. Some fuzzy logic-based inference engines defined in the mid-eighties are (Adlassnig et al., 1986; Appelbaum and Ruspini, 1985; Bonissone et al., 1987; Martin-Clouaire and Prade, 1985; Soula et al., 1986; Tong and Appelbaum, 1988; Tong and Shapiro, 1985).

In order to cope with the vagueness and imprecision of certainty factors or degrees of truth linguistically assessed by experts, several authors have attached fuzzy sets numbers (in the unit interval) to logical formulas or symbolic expressions, rather than just scalar values. For instance, Umano (1986,1989) uses the extended min operation for combining fuzzy degrees of matching with fuzzy certainty factors attached to rules, while in the inference system MILORD (Sierra, 1989; Puyol, 1994) different operations can be considered for propagating the linguistically expressed certainty values as defined in (Godo et al., 1987; Godo et al., 1989).

Finally, in (Novák, 1999) a different problem related with rule-based systems is analyzed: "In dealing with rule-based systems, we meet the problem that we have to deal both with the vagueness as well as the uncertainty". And the author states that neither *purely* fuzzy logic nor uncertainty/probability logic-based inference engines in expert systems are correct. The problem consists in the fact that a proper unifying theory is still not available. An attempt to prepare a formal frame for a theory of the whole natural language semantics has been proposed by Novák (1992). And, in (Hájek et al., 1995b; Hájek, 1998b; Godo et al., 2000; Godo et al., 2001) a formal bridge between probability, belief functions and fuzzy logic has been defined. The possibilistic system based on Gödel infinitely-valued logic defined in this thesis provides a formal frame to deal with vagueness as well as possibilistic uncertainty and could act as inference engine for expert systems.

2.5 Fuzzy logic programming

In this section, we briefly review some logic programming systems which have been defined to deal with some kind of indeterminacy (vagueness or uncertainty) in knowledge-based systems. Most of these systems are based on some of the resolution mechanisms presented in Section 2.3. Some extensions of fuzzy logic programming as linguistic truth values and fuzzy unification are also considered. Surveys on fuzzy knowledge-based systems are (Dubois et al., 1991a; Escalada-Imaz et al., 1996; Martin and Arcelli, 1998; Novák, 1999; Wagner, 1998).

The first programming language for implementing fuzzy inferences, called FUZZY, was designed and implemented by LeFaivre (1974b,1974a). However, the system is not based on a formal fuzzy logic. A version handling fuzzy truth values, called L-FUZZY, was developed by Freksa (1981).

Lee's results were implemented first by Ishizuka and Kanai (1985) in the fuzzy Prolog-ELF system. As a consequence of Lee's soundness result, only clauses having a truth value greater than 0.5 are used for inference. When several proofs exists for a same goal, the user may ask for all solutions or for the best one(s). The search strategy is blind and close to Prolog's. The user may specify a proof threshold, under which all solutions are inhibited.

The FProlog interpreter (Baldwin et al., 1987b) was developed following Baldwin's work on fuzzy relations (Baldwin, 1981). It allows fuzzy relations to be defined, with a membership value giving the degree of truth of tuples. Standard fuzzy operators (such as min/max) are used to compute truth values of formulas and to combine truth values arising from different proofs paths for a same conclusion. Truth values are not allowed for rules as well as fuzzy attribute values. The FProlog interpreter has been extended by Baldwin et al. (1993,1995) in the Fril system. It is based on the theory of mass assignments, which includes probability and fuzzy sets as special cases, and can deal with uncertainty (i) in attributes by allowing fuzzy sets as data values; (ii) in facts and rules by using support pairs which define probability intervals; and (iii) in inference by calculating a support pair for any inferences made. Fril also in-

cludes a complete Prolog-like subsystem, allowing procedural code to be written and executed in a depth-first mode. In (Baldwin and Martin, 1996), the authors have defined an object-oriented programming extension of Fril.

The Fuzzy Sets Prolog system developed by Umano (1987) allows both linguistic truth values and fuzzy constants with a discrete or continuous membership function. For instance, the fuzzy statement

"it is more or less true that Mary is about 26 years old"

may be translated by the fuzzy-valued fuzzy formula

$$age(Mary, \{0.6/25, 1/26, 0.7/27\})[\{0.7/0.8, 1/0.9, 0.8/1\}].$$

The resolution process consists of 5 steps. First, the unification of terms involved in complementary literals by means of the computation of the compatibility of the fuzzy constants involved inside the terms (by means of a function f), and the aggregation of the compatibility degrees for all terms of two matched literals (f_1). Then, the weight associated with a fact (f_2) is computed. Finally, the combination of the weights of the premises of a rule (f_3), and the combination of the obtained result with the weight associated with the rule (f_4). For instance, from the clauses

$$\begin{array}{l} p(a')[\alpha_1], \\ q(b', c')[\alpha_2], \\ p(a) \wedge q(b, c) \rightarrow r(d)[\beta], \end{array}$$

$r(d)$ can be proved with the linguistic truth value

$$f_4(\beta, f_3(f_2(\alpha_1, f(a, a')), f_2(\alpha_2, f_1(f(b, b'), f(c, c'))))).$$

By default f is the so-called compatibility measure in Zadeh's sense (Zadeh, 1979) and the combination functions are the min operation extended to fuzzy numbers. However, the author does not define a formal underlying semantics to deal with fuzzy elements.

The Fuzzy Prolog system defined by Hinde (1986) enables linguistic truth values to have another linguistic truth value in its scope, eg. $(p(a)[\alpha_2])[\alpha_1]$. This feature makes the underlying logic a bit similar to a modal logic in which modalities would be linguistic truth values. However, Hinde (1986) does not deal with such logical aspects and formulas with such imbricated truth values are truth-functionally evaluated.

Schwartz (1989) defined two inference methods for reasoning with qualitative linguistic information. One is based on a scalar distance δ between linguistic terms, i.e. given a rule $p_1 \wedge \cdots \wedge p_n \rightarrow q$ and facts $p'_1, \ldots, p'_n$, we compute $d_i = \delta(p_1, p'_1)$, for $i \in \{1, \ldots, n\}$, and $d = d_1 + \cdots + d_n$, then we infer q' such that $\delta(q, q')$ is the closest to d. Remark that a large distance between p_i and p'_i still gives a result held for certain, and thus, it may lead to rather misleading results (contrarily to the generalized modus ponens where a weak matching result leads to an uncertainty level). The second method is based on an association

of an ordered set of linguistic values to each proposition (for instance, {"very-tall", "tall", "rather-tall", "medium-tall", "rather-not-tall", ...}) and deduction is then based on an inclusion relation between them.

Orci (1989) extended Lee's resolution with a threshold, and formalized a declarative semantics. However, on the paper, he does not really specify what kind of knowledge the system is able to treat. The truth value of a clause may be a function of the logical variables involved in it, which enables an easy representation of fuzzy sets. Ray (1990) also used Lee's approach together with the explicit handling of vague predicates represented by fuzzy sets.

Mukaidono has worked on fuzzy logic programming for many years (Shen et al., 1988; Mukaidono et al., 1989; Mukaidono and Yasui, 1994; Yasui et al., 1995; Yasui and Mukaidono, 1996), his most recent contribution being the LbFP system (Yasui and Mukaidono, 1998) based on fuzzy logic with Łukasiewicz implication and product as compositional operations. In LbFP, fuzzy attribute values are not allowed as fundamental objects, and thus, no changes to the unification algorithm are therefore necessary. Facts and rules are augmented with truth values which can be specified by means of linguistic hedges. LbFP is formulated in terms of tree resolution rather than the more usual linear resolution. A different approach was proposed in PROFIL (Kikuchi and Mukaidono, 1988). Facts are weighted by an "interval truth value" containing their truth value, and rules are weighted according to Łukasiewicz implication. The combination of several refutations is done by taking the intersections of the resulting interval truth values and it enables to compute an indetermination degree (if the interval truth values are not disjoint) or a contradiction degree (if they are). Allowing imprecision on degrees of truth in PROFIL can be viewed as an attempt to deal with the uncertainty (pervading the available knowledge), and not only with the intermediary truth of vague statements with respect to precise states of knowledge.

In (Ivánek et al., 1988), the authors implemented a Prolog-like theorem prover for a Łukasiewicz logic with conjunction (min), disjunction (max), the classical involutive negation $(1-.)$, strong conjunction $(\alpha\&\beta = \max(0, \alpha+\beta-1))$ and Łukasiewicz implication. Rules have the form

$$\text{"context: } \varphi\text{; if: } \psi\text{; then: } \chi\text{; with weight: } \alpha\text{"}.$$

The contribution of a rule $(\varphi;\psi;\chi;\alpha)$ to the evaluation of the truth value of χ is truth-functionally computed, and the results given by different rules concluding χ are combined by a bounded sum operation.

Li and Liu (1990) defined a fuzzy Prolog, called f-Prolog, in a manner reminiscent of the FProlog system discussed above. Their extensions are (i) in allowing weighted first-order Horn-rules of the form

$$q \ :- \ [\ f \in (0,1]\ ]\ q_1, \ldots, q_n,$$

where the deduction degree of the conclusion q is computed as the product of f and the minimum of the deduction degrees of the goals q_i, $i \in \{1, \ldots, n\}$; (ii) in specifying a minimum threshold for a query to be satisfied; and (iii) in allowing weights to be fuzzy numbers. Fuzzy constants are not allowed.

More recently, a finite-valued Łukasiewicz logic based Prolog was defined in (Klawonn and Kruse, 1994; Klawonn, 1995). The interpreter, called LULOG, has been implemented in LISP and is based on a restricted subset of first order logic, mainly by avoiding negation. Neither fuzzy constants nor linguistic truth values have been considered.

Based on the work of Escalada-Imaz and Manyà (1994), in (Alsinet et al., 1995; Alsinet and Manyà, 1996) a first-order extension of this system was implemented by adapting the Warren Abstract Machine to the infinitely-valued context. One feature of the program clauses is that the attached truth value can be an algebraic expression instead of a value from the interval $[0, 1]$. This characteristic allows to represent fuzzy sets as a set of clauses in such a way that the attached algebraic expression corresponds to the membership function of the fuzzy set. For instance, the trapezoidal fuzzy set[2] $young = [10; 20; 30; 40]$ is represented by the following rules:

$$\begin{array}{ll}(\mu_{young}(x) & :- \ x < 10, 0),\\ (\mu_{young}(x) & :- \ x >= 10 \wedge x < 20, 0.1 * x - 1),\\ (\mu_{young}(x) & :- \ x >= 20 \wedge x < 30, 1),\\ (\mu_{young}(x) & :- \ x >= 30 \wedge x < 40, -0.1 * x + 4),\\ (\mu_{young}(x) & :- \ x >= 40, 0).\end{array}$$

Then, the fuzzy predicate *Young* can be defined as follows:

$$(Young(x) : -Age(x, y) \wedge \mu_{young}(y), 1).$$

Virtanen (1994,1998) defined a declarative and procedural semantics of a logic programming language with linguistic variables. The resolution is based on a modus ponens rule extended with fuzzy equality relations for determining the proximity between linguistic terms, i.e from the fact B' and the rule $B \rightarrow A$ we infer A', where

$$\mu_{A'}(x) = \sup_y E(\mu_B(y), \mu_{B'}(y)) \Rightarrow \mu_A(x),$$

E being a fuzzy equality relation and $\Rightarrow$ an implication relation. The author defines a unification algorithm for determining the choice of a "fuzzy equal term" for a variable that occurs more than once in a clause.

Rios-Filho and Sandri (1995) addressed the problem of unification involving fuzzy constants in systems where a separation between general and specific patterns can be made. The patterns classified in the first context are part of general information, like rules in expert systems, or ungrounded clauses in logic programming languages. The ones classified in the second context come from specific information about a problem, like facts in expert systems, or grounded clauses in logic programming languages. In this framework, they propose to use inclusion measures to compare fuzzy constants modeling different kinds of

[2]We use the representation of a trapezoidal fuzzy set as $[t_1; t_2; t_3; t_4]$ where the interval $[t_1, t_4]$ is the support and the interval $[t_2, t_3]$ is the core.

knowledge, and resemblance measures, otherwise. They also propose to aggregate distinct fuzzy constants matching a given variable.

Cao and Creasy (2000) extended annotated fuzzy logic programming with fuzzy truth values. The authors propose the following inference rule:

$$\frac{\begin{array}{l} B_1 : L_1' \wedge \cdots \wedge B_n : L_n' \\ B_1 : L_1 \wedge \cdots \wedge B_n : L_n \rightarrow H : L_0 \end{array}}{H : L_0'},$$

where L_i and L_i', for $i = \{0, \ldots, n\}$, are fuzzy truth values, and L_0' is defined as the least specific fuzzy truth value such that

$$\Delta(L_0 \mid L_0') \leq \max_{i=1,\ldots,n} \Delta(L_i \mid L_i'),$$

$\Delta(L_i \mid L_i')$ being a mismatching degree computed as

$$\Delta(L_i \mid L_i') = 1 - \inf_u \mu_{L'}(u) \Rightarrow \mu_L(u),$$

where $\Rightarrow$ is the Łukasiewicz implication. Fuzzy constants are not allowed in atomic formulas.

As we have already pointed out in Section 2.4, a different approach for computing the partial matching between terms in knowledge-based systems is based on fuzzy similarity relations. In this framework, in (Arcelli et al., 1998) three different kinds of unification in the fuzzy context were defined. The first one is based on similarity relations, the second one identifies similar objects through an equivalence relation and the last one uses "semantic constraints" for defining a more flexible unification. The authors extend classical resolution for each kind of fuzzy unification. In particular, the resolution rule based on similarity relations, which had already been used in the LIKELOG system (Formato, 1998), is the following one:

$$\frac{G_1 \wedge \cdots \wedge G_i \wedge \cdots \wedge G_n \quad A :- B_1 \wedge \cdots \wedge B_m}{\theta(G_1 \wedge \cdots \wedge G_{i-1} \wedge B_1 \wedge \cdots \wedge B_m \wedge G_{i+1} \wedge \cdots \wedge G_n)},$$

where θ is an extended unifier of G_i and A and the unification degree is computed by means of a fuzzy similarity relation over θ.

More recently, Gerla and Sessa (1999) formalized a methodology for transforming an interpreter for SLD-resolution into an interpreter that computes on abstract values which express similarity properties on the set of predicate and function symbols of a classical first-order language. In (Formato et al., 2000; Formato, 1998), the unification algorithm of Martelli and Montanari (1982) was extended to allow a partial matching between crisp constants through similarity relations.

Finally, Vinař and Vojtáš (2000) studied the problem of valid reasoning from data in situations when attributes are crisp but there is uncertainty concerning the identity of objects possessing these attributes. The authors state that this problem may become quite common in databases formed by joining heterogeneous sources and propose to use a similarity degree between object names.

In the framework of an extended possibilistic logic, Godo and Vila (1995) proposed a necessity-weighted propositional temporal logic with a Horn-rule style syntax and a possibilistic semantics. The authors define a sound inference mechanism composed of a sort of possibilistic modus ponens, i.e.

$$\frac{\begin{array}{l}(B_1, \alpha_1)\\ \cdots\cdots\\ (B_n, \alpha_n)\\ (B_1 \wedge \cdots \wedge B_n \rightarrow A, \alpha_R)\end{array}}{(A, \min(\alpha_R, \alpha_1, \ldots, \alpha_n))},$$

where A and B_i, for $i = 1, \ldots, n$, are fuzzy temporal constraints; and two inference patterns allowing a partial matching between fuzzy temporal constraints. These inference patterns provide a way to infer certain fuzzy constraints (facts) from uncertainty ones, and viceversa, i.e.

$$\frac{(A, \alpha)}{(A', 1)} \quad \text{if } \mu_{A'} \geq (\alpha \Rightarrow \mu_A)$$

and

$$\frac{(A, 1)}{(A', \alpha)},$$

where $\alpha = \inf_u \mu_A(u) \Rightarrow \mu_{A'}(u)$, $\Rightarrow$ being the Łukasiewicz implication. Similar inference patterns are used in this thesis for defining a complete calculus for a possibilistic logic programming language with fuzzy constants based on the Horn-rule fragment of Gödel infinitely-valued logic.

More recently, a syntactic extension of the necessity-valued fragment of first-order possibilistic logic dealing with fuzzy constants and fuzzily restricted quantifiers (called PLFC) was proposed in (Dubois et al., 1996; Dubois et al., 1998b). In PLFC, the partial matching between two fuzzy events A and B is implicitly computed by the resolution mechanism. In this framework, the authors define the following inference pattern:

$$\frac{(\neg p(x) \vee q(x), A(x)), (p(B) \vee r, \alpha)}{(q(B) \vee r, \min(\alpha, N(A \mid B)))},$$

where $N(A \mid B) = \inf_u \max(1 - \mu_B(u), \mu_A(u))$ is a necessity-like measure of how much certain is the fuzzy event A given the fuzzy information B. The inference patterns proposed by Dubois et al. (1996,1998b) to handle fuzzy constants and fuzzily restricted quantifiers in this fragment of possibilistic logic are fully developed in Chapter 3. In this thesis, we provide PLFC with both a formal semantics and a sound refutation by resolution proof method.

Finally, Kullman and Sandri (1999) studied, from a syntactic point of view, how PLFC logic, in the context of Horn clauses, can be handled in the annotated logic framework, and Sandri and Godo (1999) showed how to build theories in PLFC for temporal reasoning based on fuzzy temporal constraints by adapting the approach used in (Godo and Vila, 1995).

Chapter 3

Background on necessity-valued possibilistic logic and its extension with fuzzy constants and fuzzily restricted quantifiers

3.1 Introduction

The necessity-valued fragment of possibilistic logic (Dubois et al., 1994c) is a logic of uncertainty tailored for reasoning under incomplete evidence. At the syntactic level it handles formulas of propositional or first-order logic to which are attached numbers between 0 and 1, or, more generally, elements in a totally ordered set. These weights are lower bounds on so-called degrees of necessity of the corresponding formulas. The degree of necessity (or certainty) of a formula expresses to what extent the available evidence entails the truth of this formula. Degrees of necessity are closely related to fuzzy sets (Zadeh, 1965; Zadeh, 1978), and necessity-valued possibilistic logic is especially adapted to automated reasoning when the available information is pervaded with vagueness. A vague piece of evidence can be viewed as defining an implicit ordering on the possible worlds, this ordering being encoded by means of a possibility distribution. Hence, necessity-valued possibilistic logic is a tool for reasoning under uncertainty based on the idea of (complete) ordering rather than counting, contrary to probalistic logic. This idea has been extensively applied to model non-monotonic reasoning with possibilistic logic (see e.g. (Dubois et al., 1994c; Benferhat et al., 1997)). Recently, Dubois et al. (1996,1998b) have proposed a syntactic extension of this logic, called PLFC, with fuzzy constants

and fuzzily restricted quantifiers. In this chapter, we develop formal and logic programming aspects of both the necessity-valued fragment of possibilistic logic and its extension with fuzzy events.

The chapter is organized as follows. In Section 3.2, we present the language, the semantics and the proof method by refutation of the necessity-valued fragment of possibilistic logic. In Section 3.3, we establish the possibilistic interpretation of fuzzy events in PLFC and the inference patterns proposed by Dubois, Prade and Sandri for PLFC.

3.2 Necessity-valued possibilistic logic

A *necessity-valued formula* is a pair

$$(\varphi, \alpha),$$

where φ is a classical first-order, closed (without free variables) formula and $\alpha \in (0, 1]$ is a lower bound on the belief on φ in terms of necessity measures. A *necessity-valued knowledge base* is then defined as a finite set (i.e. a conjunction) of necessity-valued formulas. A formula (φ, α) is thus interpreted as a constraint $N(\varphi) \geq \alpha$, where N is a necessity measure on logical formulas, a mapping from the set of logical formulas to a totally ordered bounded scale, usually (but not necessarily) given by $[0, 1]$, characterized by the axioms:

(i) $N(\top) = 1$
(ii) $N(\bot) = 0$
(iii) $N(\varphi \wedge \psi) = min(N(\varphi), N(\psi))$
(iv) $N(\varphi) = N(\psi)$, if φ and ψ are classically equivalent

where $\top$ and $\bot$ denote respectively tautology and contradiction.

A *possibility measure* Π is associated by duality with N, namely

$$\Pi(\varphi) = 1 - N(\neg(\varphi)),$$

where $1 - (\cdot)$ is the order-reversing map of the scale. It states that the absence of certainty in favor of $\neg(\varphi)$ makes φ possible.

3.2.1 Semantics

We present here the usual (monotonic) semantics for the necessity-valued possibilistic logic (simply possibilistic logic from now on). For the sake of an easier understanding, we first consider the propositional case, and then the first-order case.

The propositional case

Let $\mathcal{L}$ be a *classical propositional language* and let $\mathcal{I}$ be the set of *classical propositional interpretations* for $\mathcal{L}$, that is, the set of interpretations $\mathbf{I}$ of the

propositional variables of $\mathcal{L}$ into the Boolean truth value set $\{0,1\}$. Each interpretation of propositional variables $\mathbf{I}$ extends to any classical propositional formula in the usual way, and thus, $\mathbf{I}(\varphi) \in \{0,1\}$ for each formula φ. For each formula φ, we write $\mathbf{I} \models \varphi$ iff $\mathbf{I}(\varphi) = 1$. We also write $[\varphi]$ to denote the set of models of φ, i.e. $[\varphi] = \{\mathbf{I} \in \mathcal{I} \mid \mathbf{I} \models \varphi\}$.

Belief states are modeled by *normalized possibility distributions* $\pi : \mathcal{I} \to [0,1]$ on the set of possible interpretations $\mathcal{I}$. A possibility distribution π is *normalized* when there is at least one $\mathbf{I} \in \mathcal{I}$ such that $\pi(\mathbf{I}) = 1$. In other words, belief states modeled by normalized distributions are consistent states, in the sense that at least one interpretation (or state or possible world) has to be fully plausible. These are our *possibilistic models*.

The *satisfaction* relation between possibilistic models (i.e. possibility distributions) and possibilistic formulas is defined as follows:

$$\pi \models (\varphi, \alpha) \text{ iff } N([\varphi] \mid \pi) \geq \alpha,$$

where $N(. \mid \pi)$ is the necessity measure induced by π on the power set of $\mathcal{I}$, defined as

$$N([\varphi] \mid \pi) = \inf_{\mathbf{I} \in \mathcal{I}} \max(1 - \pi(\mathbf{I}), \mathbf{I}(\varphi)) = \inf_{\mathbf{I} \not\models \varphi} 1 - \pi(\mathbf{I}).$$

If $\pi \models (\varphi, \alpha)$ we say that π is a *model* of (φ, α). It is interesting to notice that the set of models of (φ, α) has a *greatest element* $\pi_{(\varphi,\alpha)}$ defined as

$$\pi_{(\varphi,\alpha)}(\mathbf{I}) = \max(1 - \alpha, \mathbf{I}(\varphi)), \text{ for each } \mathbf{I} \in \mathcal{I}.$$

Hence, there is an equivalent definition of *possibilistic satisfaction*:

$$\pi \models (\varphi, \alpha) \text{ iff } \pi(\mathbf{I}) \leq \max(1 - \alpha, \mathbf{I}(\varphi)) \text{ for each } \mathbf{I} \in \mathcal{I}.$$

As usual, if K denotes a necessity-valued knowledge base, we say that π is a *model* of K iff π is a model of each necessity-valued formula in K. The *possibilistic entailment*, denoted $\models_{PL}$, is then defined as follows:

$$K \models_{PL} (\varphi, \alpha) \text{ iff } \pi \models (\varphi, \alpha) \text{ for each } \pi \text{ being model of } K.$$

The first-order case

When the language $\mathcal{L}$ is of first-order, things do not change very much. *First-order interpretations* are structures of the form

$$\mathbf{M} = (U, i, m),$$

where U is a non-empty domain (or sets of domains if we have sorts); i maps each predicate symbol of arity n to a subset of U^n; and m maps each object constant to an element of the domain U. Then, if φ is a closed first-order formula, we continue writing $\mathbf{M} \models \varphi$ to denote that φ is true in the interpretation $\mathbf{M}$. For instance,

$$\mathbf{M} \models (\forall\, x)p(x, c) \text{ iff } (u, m(c)) \in i(p) \text{ for each } u \in U.$$

Now, *possibilistic models* are possibility distributions π on the set of first-order interpretations $\mathcal{M} = \{\mathbf{M} = (U, i, m)\}$ of $\mathcal{L}$, and possibilistic semantics (possibilistic satisfaction and entailment) are then just as for the propositional case.

3.2.2 A formal system for necessity-valued possibilistic logic

Possibilistic logic is axiomatized (Hilbert-style) by the axioms of classical first-order logic weighted by 1, i.e.

(A1) $(\varphi \to (\psi \to \varphi), 1)$
(A2) $((\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi)), 1)$
(A3) $((\neg\varphi \to \neg\psi) \to ((\neg\varphi \to \psi) \to \varphi), 1)$
(A4) $((\forall\, x)\varphi(x) \to \varphi(t), 1)$ (t substitutable for x in $\varphi(x)$)
(A5) $((\forall\, x)(\varphi \to \psi) \to (\varphi \to (\forall\, x)\psi), 1)$ (x not free in φ)

together with the following *graded* versions of the usual *modus ponens* and *generalization* inference rules:

$$\frac{(\varphi, \alpha), (\varphi \to \psi, \beta)}{(\psi, \min(\alpha, \beta))},$$

$$\frac{(\varphi, \alpha)}{((\forall\, x)\varphi, \alpha)} \quad \text{if } x \text{ is not bound in } \varphi;$$

together with the following *weight weakening* inference rule:

$$\frac{(\varphi, \alpha)}{(\varphi, \beta)} \quad \text{if } \beta \leq \alpha.$$

We denote by $\vdash_{PL}$ the *notion of proof* in possibilistic logic derived from this formal system of axioms and inference rules. In Dubois et al. (1994c), the authors showed that the proposed formal system is *sound and complete* with respect to the above possibilistic semantics, i.e. for any necessity-valued knowledge base K we have

$$K \models_{PL} (\varphi, \alpha) \text{ iff } K \vdash_{PL} (\varphi, \alpha).$$

3.2.3 Automated deduction

Refutation by resolution is an automated deduction method which has been nicely adapted to possibilistic logic.

In order to extend resolution to possibilistic logic, a clausal form is first defined. A *possibilistic clause* is a pair (C, α), where C is a first-order (or propositional) clause and α is a valuation of $(0, 1]$. A *possibilistic clausal form* is a universally quantified conjunction of possibilistic clauses. Indeed, let $K = \{(\varphi_i, \alpha_i) \mid i = 1, \ldots, n\}$ be a necessity-valued knowledge base. The possibilistic clausal form $\mathcal{C}$ of K can be obtained by the following method:

1. Put each formula φ_i into clausal form, i.e. $\varphi_i = (\forall) \bigwedge_j (C_{ij})$, where C_{ij} is a universally quantified classical first-order clause.

2. $\mathcal{C} \leftarrow (\forall) \bigwedge_{i,j} \{(C_{ij}, \alpha_i)\}$.

Once a clausal form is defined for a given necessity-valued knowledge base, the resolution principle may easily be extended from classical first-order logic to possibilistic logic. The following *possibilistic resolution rule* between two possibilistic clauses (C_1, α_1) and (C_2, α_2) has been established by Dubois and Prade (1987b):

$$\frac{(C_1, \alpha_1), (C_2, \alpha_2)}{(R(C_1, C_2), \min(\alpha_1, \alpha_2))}, \tag{3.1}$$

where $R(C_1, C_2)$ is any classical resolvent of C_1 and C_2. For instance,

$$\frac{(\neg p \vee q, \alpha), (p \vee r, \beta)}{(q \vee r, \min(\alpha, \beta))}.$$

Dubois, Lang and Prade have shown (Dubois et al., 1994c) that the possibilistic resolution rule is *sound* with respect to the possibilistic entailment of possibilistic clauses: Let $\mathcal{C}$ be a set of possibilistic clauses, and let (C, α) be a possibilistic clause obtained by a finite number of successive applications of possibilistic resolution rule 3.1 to $\mathcal{C}$. Then, $\mathcal{C} \models_{PL} (C, \alpha)$. Moreover, they have shown that refutation by resolution can be used for computing the maximal necessity valuation $\bar{\alpha}$ with which a necessity-valued knowledge base K entails a classical formula φ, i.e. $\bar{\alpha} = \sup\{\alpha \in (0, 1] \mid K \models_{PL} (\varphi, \alpha)\}$. *Refutation by resolution* is extended to possibilistic logic as follows:

1. Put K into possibilistic clausal form $\mathcal{C}$.

2. Put $\neg\varphi$ into clausal form; let $C_1, \ldots, C_m$ be the obtained clauses.

3. $\mathcal{C}' \leftarrow \mathcal{C} \cup \{(C_1, 1), \ldots, (C_m, 1)\}$.

4. Search for a deduction of $(\bot, \bar{\alpha})$ by applying possibilistic resolution rule 3.1 from $\mathcal{C}'$ repeatedly, with $\bar{\alpha}$ maximal.

5. $K \models_{PL} (\varphi, \bar{\alpha})$.

If we denote the above procedure of proof by refutation through resolution by $\vdash^r_{PL}$, we have the following *soundness* and *completeness* result:

$$K \models_{PL} (\varphi, \bar{\alpha}) \text{ iff } K \vdash_{PL} (\varphi, \bar{\alpha}) \text{ iff } K \vdash^r_{PL} (\varphi, \bar{\alpha}).$$

An implementation for finding out the refutation with $\bar{\alpha}$ maximal first has been proposed in (Dubois et al., 1987a). Finally, using the $\vdash^r_{PL}$ procedure, other inference rules can be derived. For instance, the following *fusion rule*:

$$\frac{(\varphi, \alpha), (\varphi, \beta)}{(\varphi, max(\alpha, \beta))}, \tag{3.2}$$

and, when the unification mechanism employed in the resolution rule is the same as in classical logic, the following *particularization rule* is also derivable:

$$\frac{((\forall\, x)\varphi(x), \alpha)}{(\varphi(t), \alpha)} \quad \text{if } t \text{ substitutable for } x \text{ in } \varphi(x).$$

3.3 Possibilistic logic with fuzzy constants and fuzzily restricted quantifiers: PLFC

Dubois et al. (1996,1998b) defined a syntactic extension of necessity-valued possibilistic logic, always using formulas in possibilistic clausal form, where, in order to deal with fuzzy predicates and poorly known values, variable weights and generalized fuzzy constants are allowed respectively. The underlying idea of the authors was to propose an extension of possibilistic logic sticking to classical logic proof procedures as much as possible, in particular to refutation by resolution, as in standard possibilistic logic. However, there was no evaluation there about whether such a proof by refutation method can be supported by a well-defined semantics. As we have already pointed out in Section 1.2, in this thesis we provide this extension with both a formal semantics and a sound refutation by resolution proof method. In the rest of this thesis, we refer to this extension as PLFC.

3.3.1 Variable weights

One way to accommodate fuzzy predicates in the framework of possibilistic logic, is to allow for fuzzy predicates in the language and to have extended resolution rules for them (Dubois and Prade, 1990). A different way, used in PLFC, is to keep classical predicates, but to allow for *variable weights*, as we have already suggested in (Dubois et al., 1994a; Dubois et al., 1994b). Namely, the weight associated with a classical logic formula, which expresses its level of certainty and is formally interpreted in terms of a necessity measure, may itself depend on variables involved in it. Letting the weight be a function of variables involved in the formula makes the certainty of the formula depend on some proviso expressed in terms of regular, or fuzzy, predicates whose characteristic functions appear in the expression of the weight. This enables us to attach a fuzzily restricted universal quantifier to a classical formula. Thus, variable weights are employed in PLFC to enable the modeling of statements such as

"the more x is A (or x belongs to A), the more certain is $p(x)$",

where A is a fuzzy set with a membership function μ_A. This is formalized as

"for all x, $p(x)$ is true with a necessity degree of at least $\mu_A(x)$",

and represented as

$$(p(x), A(x)),$$

where it is assumed, as in standard possibilistic clauses, that all variables appearing in a clause are universally quantified. When A is imprecise but not fuzzy, the interpretation of such a formula amounts to say that

$$\text{“}\forall\ x \in A,\ p(x) \text{ is true”}.$$

So A acts as a (flexible if it is fuzzy) restriction on the universal quantifier. For instance, the statement

“the younger the person, the more certain he/she is single”

can be represented by the variable certainty weighted possibilistic clause

$$(\neg age(x,y) \vee single(x), young(y)),$$

where here $\mu_A = \mu_{young}$.

The resolution principle was extended from possibilistic logic to variable certainty weighted possibilistic logic in the following way. The instantiation of a variable certainty weight is determined by the instantiation of the involved variables. For instance, from

$$(\neg p(x) \vee q(x), A(x)) \quad \text{and} \quad (p(a), \alpha)$$

we infer

$$(q(a), \min(A(a), \alpha)),$$

where $A(a)$ is computed as $\mu_A(a)$. Hence, the particularization of variables is performed both on the logical-part and on the weight-part of a clause, as in the following example:

$$\frac{\begin{array}{l}(\neg age(x,z) \vee \neg age(y,t) \vee friend(x,y), \min(0.6, approx\_equal(z,t))) \\ (age(Mary, 20), 0.9) \\ (age(Peter, 21), 0.7)\end{array}}{(friend(Mary, Peter), \min(0.9, 0.7, 0.6, approx\_equal(20, 21)))}.$$

When applying the above resolution mechanism to possibilistic clauses with variable weights, involved variables may disappear in the logical-part of a clause, but still appear in the weight-part. For instance, from

$$(\neg p(x) \vee q(y), R(x,y)) \quad \text{and} \quad (p(x), A(x)),$$

where R and A denote a (fuzzy) relation and a (fuzzy) set, respectively, we can infer

$$(q(y), \min(R(x,y), A(x))),$$

which, in possibilistic terms, is interpreted as “for all y and $x \in X$, $q(y)$ is true with a necessity degree of at least $\min(\mu_R(x,y), \mu_A(x))$”, where X denotes the supposedly finite domain of variable x, and thus, for each $x \in X$ we get a clause with the same logical-part. Hence, the use of a particular case of the fusion

rule 3.2 was proposed to eliminate variables that only appear in the weight-part of a clause, and thus, from

$$(q(y), \min(R(x,y), A(x)))$$

we infer

$$(q(y), \max_{x \in X} \min(R(x,y), A(x))).$$

A more intuitive example is the following one. From

$$(\neg speaks(x,y) \vee visited(x, Europe), \min(0.7, Latin(y)))$$

and

$$(speaks(Mary, z), S\_F\_I(z)),$$

where $Latin$ denotes the set composed of all Latin languages and $S\_F\_I = \{Spanish, French, Italian\}$, we infer

$$(visited(Mary, Europe), \min(0.7, Latin(y), S\_F\_I(y))),$$

which, in possibilistic terms, is interpreted as "for all language y, we are certain that Mary has visited Europe with a necessity of at least $\min(0.7, Latin(y), S\_F\_I(y))$", and thus, for each language we get a clause with the same logical-part. Now, applying the fusion rule we infer

$$(visited(Mary, Europe), \max_{y \in languages} \min(0.7, Latin(y), S\_F\_I(y))),$$

which can be turned into

$$(visited(Mary, Europe), \min(0.7, \max_{y \in languages} \min(Latin(y), S\_F\_I(y)))),$$

where $\max_{y \in languages} \min(Latin(y), S\_F\_I(y)) = 1$, i.e. "we are certain that Mary has visited Europe with a necessity of at least 0.7" provided that she speaks a Latin language.

3.3.2 Fuzzy constants

In the above examples, constants are precise values as usual. Allowing for imprecise or even fuzzy constants representing poorly known values clearly enlarges the knowledge representation power of the possibilistic logic language. Fuzzy constants are used to model typical fuzzy statements of the type

"in Brazil the mean temperature in December is about_25",

represented in PLFC as

$$mean\_temp(Brazil, December, about\_25),$$

where $mean\_temp$ is a classical predicate and $about\_25$ is a generalized constant. In the case in which $about\_25$ denotes a crisp interval of temperatures, say $[24, 26]$, the above expression is interpreted as

"$\exists\, x \in [24, 26]$ such that $mean\_temp(Brazil, December, x)$ is true".

In the case in which $about\_25$ denotes a fuzzy interval of temperatures with a membership function μ_{about_25}, the above expression is interpreted in possibilistic terms as

"$\exists\, x \in [\mu_{about\_25}]_\beta$ such that $mean\_temp(Brazil, December, x)$ is certain with a necessity of at least $1 - \beta$", for each $\beta \in [0, 1]$,

where $[\mu_{about\_25}]_\beta$ denotes the β-cut of μ_{about_25}. So, a fuzzy constant can be seen as (flexible if it is fuzzy) restriction on an existential quantifier.

More formally, in PLFC, an imprecise constant

$$B = \{b_1, \ldots, b_n\}$$

expresses disjunctive information about the elements in B. Thus, the expression $L(B)$, where L is either a positive or negative literal, is equivalent to the disjunction

$$L(b_1) \vee \cdots \vee L(b_n)$$

and so, it is interpreted as

"$\exists\, b \in B$ such that $L(b)$".

On the other hand, a fuzzy constant B with membership function μ_B can be described by its β-cuts, i.e. $[\mu_B]_\beta = \{b \mid \mu_B(b) \geq \beta\}$ with $\beta \in [0, 1]$. Then, the certainty that $L([\mu_B]_\beta)$ holds is lower bounded by $1 - \beta$. Indeed, in possibility theory the necessity measure that $[\mu_B]_\beta$ is true, knowing the fuzzy information μ_B is such that

$$N([\mu_B]_\beta) = \inf_{b \notin [\mu_B]_\beta} 1 - \mu_B(b) \geq 1 - \beta.$$

Hence, when B is a fuzzy constant, $L(B)$ is equivalent to the set of possibilistic clauses with imprecise constants

$$\{(L([\mu_B]_\beta), N([\mu_B]_\beta)) \mid \beta \in [0, 1]\}.$$

This set is finite iff the number of distinct possibility levels β which are necessary for describing B is finite. Assume that these levels are ordered in the following way

$$\beta_1 = 1 > \beta_2 > \cdots > \beta_k > \beta_{k+1} = 0,$$

then

$$N([\mu_B]_{\beta_i}) = 1 - \beta_{i+1}, \text{ for } i = 1, \ldots, k.$$

In particular,

$$N([\mu_B]_{\beta_k}) = N(\{b \mid \mu_B(b) > 0\}) = 1,$$

which expresses that we are certain that the values restricted by B are in the support of B. In case the number of distinct β-cuts is infinite (which requires that the universe on which the fuzzy set B is defined is also infinite), and the

membership function μ_B is continuous, we have that $N([\mu_B]_\beta) = 1 - \beta$. Finally, since $[\mu_{B_1 \cup B_2}]_\beta = [\mu_{B_1}]_\beta \cup [\mu_{B_2}]_\beta$, we have that $L(B_1 \cup B_2) = L(B_1) \vee L(B_2)$ in the crisp and in the fuzzy cases.

Special attention must be given when reading a literal such as $\neg p(B)$ in a possibilistic clause. Indeed, if B is not fuzzy,

$$(p(B), 1) \quad \text{and} \quad (\neg p(B), 1)$$

have to be respectively interpreted as

$$\text{“}\exists\, x \in B,\, p(x)\text{”} \quad \text{and} \quad \text{“}\exists\, x \in B,\, \neg p(x)\text{”}.$$

Moreover, a formula such as

$$(\neg p(B) \vee r(C), 1),$$

where B and C are not fuzzy, should be interpreted as

$$\text{“it is certain that } [\exists\, x \in B,\, \neg p(x)] \text{ or } [\exists\, y \in C,\, r(y)]\text{”},$$

or equivalently,

$$\text{“it is certain that } [\forall\, x \in B,\, p(x)] \rightarrow [\exists\, y \in C,\, r(y)]\text{”}$$

that is completely different of the interpretation

$$\text{“it is certain that } \forall\, x \in B,\, [p(x) \rightarrow [\exists\, y \in C,\, r(y)]]\text{”},$$

or equivalently,

$$\text{“it is certain that } \forall\, x \in B,\, [\neg p(x) \text{ or } [\exists\, y \in C,\, r(y)]]\text{”},$$

which is represented as $(\neg p(x) \vee r(C), B(x))$.

It is also important to note that the proposed framework does not completely allow for a proper handling of conjunction inside the scope of an existential quantifier (i.e. an imprecise constant). For instance, we cannot represent

$$\text{“it is certain that } \exists\, x \in B,\, [\forall\, y \in C,\, p(x, y)]\text{”},$$

but only

$$\text{“it is certain that } \forall\, y \in C,\, [\exists\, x \in B,\, p(x, y)]\text{”}$$

as $(p(B, y), C(y))$. Indeed, B does not play the role of a Skolem constant here. Moreover, since “$\exists\, x \in B,\, [p(x) \text{ and } q(x)]$” only entails (but is not equivalent to) “$[\exists\, x \in B,\, p(x)]$ and $[\exists\, x \in B,\, q(x)]$”, the representation of the first formula cannot be handled in PLFC since we can only express $(p(B), 1)$ and $(q(B), 1)$, which is the exact counterpart of the second formula.

3.3.3 Inference with fuzzy constants and variable weights

In this section, we develop the inference patterns proposed by Dubois, Prade and Sandri for handling fuzzy constants and variable weights in the framework of possibilistic logic.

First, we present the pattern of reasoning with a *restricted, but non-fuzzy, quantifier* A and an *imprecise constant* B. Thus, from

$$\text{“}\forall\, x \in A,\, \forall\, y,\, \neg p(x,y) \vee q(x,y)\text{”} \quad \text{and} \quad \text{“}\exists\, x \in B,\, p(x,c)\text{”},$$

where y is a non-restricted variable and c is a precise constant, we can conclude that

$$\text{“}\exists\, x \in B \text{ such that } q(x,c)\text{”} \textit{ provided that } B \subseteq A.$$

Hence, if A and B are imprecise, but non-fuzzy, objects, we get the following possibilistic inference pattern:

$$\frac{\begin{array}{l}(\neg p(x,y) \vee q(x,y), A(x)) \\ (p(B,c),1)\end{array}}{(q(B,c), \min_{x\in B} \mu_A(x))}, \tag{3.3}$$

where $\min_{x\in B} \mu_A(x)$ is 1 iff $B \subseteq A$, and is 0, otherwise.

Pattern 3.3 can be readily extended to the case of a *fuzzily restricted quantifier*, where $\mu_A(x)$ is the membership function of a fuzzy set A. Indeed, if $B = \{b_1, \ldots, b_n\}$,

$$(p(B,c),1)$$

is equivalent to

$$(p(b_1,c) \vee \cdots \vee p(b_n,c),1).$$

Then, from

$$(\neg p(x,y) \vee q(x,y), A(x)), \text{ applied to } x = b_1, \ldots, x = b_n,$$

and

$$(p(b_1,c) \vee \cdots \vee p(b_n,c),1),$$

we can conclude that

$$(q(b_1,c) \vee \cdots \vee q(b_n,c), \min(\mu_A(b_1), \ldots, \mu_A(b_n))),$$

which is precisely pattern 3.3 when A is fuzzy and B is non-fuzzy.

Now, we consider the pattern of reasoning with a *restricted, but non-fuzzy, quantifier* A and a *fuzzy constant* B with membership function μ_B, having a finite number k of β-cuts, such that $\beta_1 = 1 > \beta_2 > \cdots > \beta_k > \beta_{k+1} = 0$. In possibilistic terms, it reads

$$\begin{array}{l}(\neg p(x,y) \vee q(x,y), A(x)) \\ (p([\mu_B]_{\beta_i}, c), 1 - \beta_{i+1}), \text{ for } i = 1, \ldots, k\end{array} \tag{3.4}$$

Thus, we can apply pattern 3.3, looking for β_i such that $[\mu_B]_{\beta_i} \subseteq A$ with $1-\beta_{i+1}$ as large as possible (since we are interested in the most certain conclusion). Hence, from 3.4, we can conclude

$$(q(B,c), \max_{i=1,\ldots,k} \min(1-\beta_{i+1}, \min_{x\in[\mu_B]_{\beta_i}} \mu_A(x))).$$

It can be shown, in the infinite case with continuous membership functions (e.g. Prade (1982)), that

$$\begin{aligned} \sup_\beta \min(1-\beta, \min_{x\in[\mu_B]_\beta} \mu_A(x)) &= 1 - \inf\{\beta \mid [\mu_B]_\beta \subseteq A\} \\ &= \inf_{x\notin A} 1-\mu_B(x). \end{aligned}$$

When B has a finite number k of level cuts, we also have

$$\max_{i=1,\ldots,k} \min(1-\beta_{i+1}, \min_{x\in[\mu_B]_{\beta_i}} \mu_A(x)) = \min_{x\notin A} 1-\mu_B(x).$$

Hence, from 3.4 we can conclude

$$(q(B,c), \min_{x\notin A} 1-\mu_B(x)).$$

The general case, where *both* A and B are *fuzzy*, can be handled in terms of level cuts as well. Indeed, we have the possibilistic clauses

$$([\forall\, x \in [\mu_A]_\alpha,\ \neg p(x,y) \vee q(x,y)], \alpha), \text{ for each } \alpha\text{-cut of } A,$$

and

$$([\exists\, x \in [\mu_B]_{\beta_i},\ p(x,c)], 1-\beta_{i+1}), \text{ for each } \beta\text{-cut of } B.$$

Then, we should maximize $\min(\alpha, 1-\beta_{i+1})$ under the constraint $[\mu_B]_{\beta_i} \subseteq [\mu_A]_\alpha$. Prade (1982) proved that in the infinite case for continuous membership functions

$$\sup\{\min(\alpha, 1-\beta) \mid [\mu_B]_\beta \subseteq [\mu_A]_\alpha\} = \inf_x \max(\mu_A(x), 1-\mu_B(x)).$$

In the finite case, we still have

$$\max\{\min(\alpha, 1-\beta_{i+1}) \mid [\mu_B]_{\beta_i} \subseteq [\mu_A]_\alpha\} = \min_x \max(\mu_A(x), 1-\mu_B(x)).$$

From this result, Dubois, Prade and Sandri established the following general pattern for PLFC:

$$\frac{\begin{array}{l}(\neg p(x,y) \vee q(x,y), A(x)) \\ (p(B,c), 1)\end{array}}{(q(B,c), N(A \mid B))}, \tag{3.5}$$

where

$$N(A \mid B) = \inf_x \max(\mu_A(x), 1-\mu_B(x))$$

is the necessity measure of the fuzzy event A based on the fuzzy information B.

Several remarks are in order here. The first one is that, obviously, this pattern generalizes the cases where only A or B are fuzzy. The second one is that with this definition, in general, we can only ensure that $\frac{1}{2} \leq N(A \mid A) \leq 1$. Furthermore, $N(A \mid A) = 1$ iff A is not a fuzzy constant. And the third one is that the equivalence

$$N(A \mid B) = 1 \text{ iff the core of } A \supseteq \text{the support of } B$$

is also a consequence.

When A is a *fuzzy relation*, pattern 3.5 was generalized as follows. Let $A = A_1 \times \cdots \times A_m$ be a fuzzy relation with membership function $\mu_A(x_1, \ldots, x_m)$. Then, if x_i is unified with a fuzzy constant B_i, for $i = 1, \ldots, m$, $N(A \mid B)$ in 3.5 is changed into

$$N(A \mid B_1 \times \cdots \times B_m) = \min_{i=1,\ldots,m} N(A_i \mid B_i).$$

Finally, pattern 3.5 produces conclusions which are all the stronger as A is large and B is small. Indeed,

$$N(A \mid B) \geq N(A' \mid B) \text{ if } \mu_A \geq \mu_{A'}$$

and

$$N(A \mid B) \geq N(A \mid B') \text{ if } \mu_B \leq \mu_{B'}.$$

This points out that it is interesting to have possibilistic clauses with the greatest possible weight. In order to get larger variable weights, the fusion rule 3.2 was extended in the following way:

$$\frac{(\neg p(x) \vee q, A(x)) \quad (\neg p(x) \vee r, B(x))}{(\neg p(x) \vee q \vee r, [A \cup B](x))}. \tag{3.6}$$

Then, for instance, from

$$(\neg p(x) \vee q \vee r, [A \cup B](x)) \quad \text{and} \quad (p(C), 1)$$

using pattern 3.5 we can conclude

$$(q \vee r, N([A \cup B] \mid C)),$$

and we may have that $N([A \cup B] \mid C) > 0$ while $N(A \mid C) = N(B \mid C) = 0$.

Chapter 4

On the semantics and automated deduction for PLFC

4.1 Introduction

In Chapter 3, we first presented formal and logic programming aspects of the necessity-valued fragment of possibilistic logic (Dubois et al., 1994c), a well-known graded logic of uncertainty suitable to reason under incomplete information, built upon classical first-order logic, and with a sound and complete proof procedure by refutation through resolution. Then, we gave a detailed description of the syntactic extension, proposed by Dubois et al. (1996,1998b), of this fragment of possibilistic logic with fuzzy constants and variable weights (called PLFC).

Now in this chapter we tackle both the formalization of PLFC itself and an automated deduction system for it by (i) providing a formal semantics; (ii) defining a sound resolution-style calculus by refutation; and (iii) describing a first-order proof procedure for PLFC clauses based on (ii) and on a novel notion of most general substitution of two literals in a resolution step. In contrast to standard possibilistic semantics, the truth value of a formula with fuzzy constants is many-valued instead of Boolean (two-valued), and consequently, an extended notion of possibilistic uncertainty is also needed.

The chapter is organized as follows. In Section 4.2, we provide possibilistic logic extended with fuzzy constants with formal semantical grounds which leads to a qualitative jump from Boolean to many-valued semantics of the underlying representational language and, in turn, to an increasing of complexity of the uncertainty model. In Section 4.3, we formalize the syntax and the semantics of PLFC clauses. To this end, in the first part of this section we focus on PLFC clauses with constants weights, and then, we take into account variable weights.

In Section 4.4, we define a sound proof method by refutation using an extended version of the possibilistic resolution rule. Finally, based on them, in Section 4.5 we describe an automated deduction method for PLFC.

4.2 Extending standard possibilistic semantics

We are concerned in providing PLFC with a sound semantics, extending the one provided in Section 3.2 for possibilistic logic. So, the matter is what has to be modified in the possibilistic logic semantics to support the extension of the logical constructs of PLFC, where predicates are allowed to talk about poorly known, imprecise or fuzzy constants, and where a form of fuzzily restricted quantifiers is also present.

For instance, consider the following predicate instance:

$$mean\_temp(Brazil, december, about\_25)$$

denoting an imprecision about what the actual mean value of the temperatures is. Our intended interpretation is that *about_25* is a fuzzy set of temperatures describing temperatures around 25^oC, in the range from -50^oC to 50^oC, and with a particular membership function

$$\mu_{about\_25} : [-50, 50] \rightarrow [0, 1].$$

In doing so, we are introducing two major changes in the standard semantics of possibilistic logic: (i) the truth value of predicates and formulas are no longer Boolean but many-valued, and thus, the set of truth values becomes the whole unit interval $[0, 1]$; and (ii) the certainty evaluation of formulas in a possibilistic model has to be extended in a suitable manner, that is, we have to define what does $N([\varphi] \mid \pi)$ mean when φ contains fuzzy constants.

(i) From Boolean to many-valued. With respect to standard possibilistic logic, the main difference of truth values is that now, for example, in a particular interpretation, the predicate instance $mean\_temp(Brazil, december, about\_25)$ can be true (1), false (0), but can also take some intermediate truth degree, depending on how much the actual temperature in the interpretation fits with the fuzzy set *about_25*. For instance, consider an interpretation

$$\mathbf{M} = (U, i, m),$$

where

- $U = U_1 \times U_2 \times U_3$, where U_1 is a set of countries, U_2 is the set of months and U_3 is the crisp interval $[-50, 50]$.
- i maps the predicate symbol *mean_temp* into a crisp relation $i(mean\_temp) \subseteq U_1 \times U_2 \times U_3$. For instance,

$$\begin{aligned} i(mean\_temp) \;=\; \{\; & (Brazil, january, 30), \ldots, (Brazil, december, 27), \\ & \cdots\cdots \\ & (Spain, january, 5), \ldots, (Spain, december, 10)\;\}. \end{aligned}$$

- m not only assigns elements of U to usual object constants, but also a fuzzy set (i.e. a membership function) to each fuzzy constant, in this case to *about_25*. For instance, $m(about\_25) = [20; 25; 25; 30]$.

Then, it seems natural to take as truth degree of the predicate instance $mean\_temp(Brazil, december, about\_25)$, in the interpretation $\mathbf{M}$, the value

$$\begin{aligned} \|mean\_temp(Brazil, december, about\_25)\|_{\mathbf{M}} &= \mu_{m(about\_25)}(27) \\ &= \mu_{[20;25;25;30]}(27) \\ &= 0.6. \end{aligned}$$

(ii) Certainty-evaluation. According to (i), we assume from now on that the truth value of PLFC formulas φ in each interpretation $\mathbf{M}$ is a value $\|\varphi\|_{\mathbf{M}} \in [0, 1]$. Therefore, each PLFC formula does not induce anymore a crisp set of interpretations, but a fuzzy set of interpretations $[\varphi]$, defining

$$\mu_{[\varphi]}(\mathbf{M}) = \|\varphi\|_{\mathbf{M}}, \text{ for each interpretation } \mathbf{M} \in \mathcal{M}.$$

Hence, if we want to continue measuring the uncertainty induced on a PLFC formula by a possibility distribution on a set of interpretations, we have to consider some extension for fuzzy sets (of interpretations) of the standard notion of necessity measure.

The basic question is, given a belief state modeled by a possibility distribution π, how to establish the possibilistic semantics of statements of the type

$$\varphi \text{ is } \alpha - \text{certain},$$

where φ represents vague, incomplete or imprecise knowledge about the real world (i.e. a fuzzy set). Thus, we have to define a measure $N(\cdot \mid \pi)$, extension of the necessity measure introduced in Section 3.2 for classical sets, in such a way that a possibility distribution π supports the statement iff $N([\varphi] \mid \pi) \geq \alpha$. This question has already been tackled by Dubois and Prade (1991c) (see e.g. Dubois et al. (1994c)) where they propose to use this index:

$$N^*([\varphi] \mid \pi) = \inf_{\mathbf{M} \in \mathcal{M}} \pi(\mathbf{M}) \Rightarrow \mu_{[\varphi]}(\mathbf{M}),$$

where $\Rightarrow$ is the reciprocal of Gödel's many-valued implication, defined as

$$x \Rightarrow y = \begin{cases} 1, & \text{if } x \leq y \\ 1 - x, & \text{otherwise.} \end{cases}$$

Several remarks are in order here. The first one is that it is indeed an extension of the classical definition, in the sense that we recover it whenever φ model precise knowledge about the real world (i.e. a crisp set). The second one is that with this definition, the condition

$$\pi(\mathbf{M}) \leq \max(1 - \alpha, \mu_{[\varphi]}(\mathbf{M})) \text{ for each } \mathbf{M} \in \mathcal{M},$$

is again equivalent to

$$N^*([\varphi] \mid \pi) \geq \alpha.$$

And the third one is that the equivalence

$$N^*([\varphi] \mid \pi) = 1 \text{ iff } \pi(\mathbf{M}) \leq \mu_{[\varphi]}(\mathbf{M}) \text{ for each } \mathbf{M} \in \mathcal{M}$$

is also a nice consequence.

But, the bad news about this candidate semantics is that proof by refutation using the resolution rule 3.5 (the general pattern for PLFC proposed by Dubois et al. (1996,1998b)) is not sound, even though the resolution rule 3.5 itself can be proved to be sound.

Let us consider the following instances of predicate *mean_temp*:

A1: $mean\_temp(Brazil, december, \mu_{T_1})$

A2: $mean\_temp(Brazil, december, \mu_{T_2})$

where μ_{T_1} and μ_{T_2} are trapezoidal fuzzy sets of temperatures in the range from -50^oC to 50^oC defined as

$$\mu_{T_1} = [20; 24; 26; 30] \text{ and } \mu_{T_2} = [20; 25; 25; 30].$$

It is easy to check that

$$\inf_{t \in [-50,50]} \mu_{T_1}(t) \Rightarrow \mu_{T_2}(t) = 0,$$

and thus, $(mean\_temp(Brazil, december, \mu_{T_2}), \alpha)$ cannot be a logical consequence of $(mean\_temp(Brazil, december, \mu_{T_1}), 1)$ if $\alpha > 0$. On the other hand, by refutation, using the resolution rule 3.5, we get that

$$\frac{\begin{array}{l}(mean\_temp(Brazil, december, \mu_{T_1}), 1) \\ (\neg mean\_temp(Brazil, december, x), \mu_{T_2}(x))\end{array}}{(\bot, N(\mu_{T_2} \mid \mu_{T_1}))},$$

where the PLFC clause $(\neg mean\_temp(Brazil, december, x), \mu_{T_2}(x))$ stands for $\neg(mean\_temp(Brazil, december, \mu_{T_2}), 1)$ and

$$N(\mu_{T_2} \mid \mu_{T_1}) = \inf_{t \in [-50,50]} \max(1 - \mu_{T_1}(t), \mu_{T_2}(t)) = 4/9 > 0.$$

However, there is an alternative notion of necessity of fuzzy events which is commonly used in Possibility Theory as a measure of pattern matching (Dubois and Prade, 1998a), and which corresponds to the necessity measure proposed by Dubois et al. (1996,1998b) for computing the partial matching between fuzzy events in PLFC (see Section 3.3.3). Thus, to define

$$N([\varphi] \mid \pi) = \inf_{\mathbf{M} \in \mathcal{M}} \max(1 - \pi(\mathbf{M}), \mu_{[\varphi]}(\mathbf{M})).$$

This definition obviously extends the standard notion of necessity degree when φ is crisp, and we have that $N([\varphi] \mid \pi) = 1$ only when every plausible interpretation $\mathbf{M}$ (i.e. $\pi(\mathbf{M}) > 0$) makes φ totally true (i.e. $\mu_{[\varphi]}(\mathbf{M}) = 1$). Now, the condition $N([\varphi] \mid \pi) \geq \alpha$ becomes equivalent to the inequality

$$\pi(\mathbf{M}) \leq \max(1 - \alpha, \mu_{[\varphi]_\alpha}(\mathbf{M})) \text{ for each } \mathbf{M} \in \mathcal{M},$$

where $[\varphi]_\alpha$ denotes the α-cut of $[\varphi]$, i.e. $[\varphi]_\alpha = \{\mathbf{M} \in \mathcal{M} \mid \mu_{[}\varphi](\mathbf{M}) \geq \alpha\}$. In Section 4.4, we show that using this semantics a sound refutation by resolution proof method for PLFC clauses can be defined. Moreover, there is a nice axiomatization for the above defined necessity measure for fuzzy sets. Namely, let Ω be a set and let $N : [0,1]^\Omega \to [0,1]$ be a measure on the set of fuzzy sets of Ω. Consider the following postulates:

N1 $N(\Omega) = 1$
N2 $N(\emptyset) = 0$
N3 $N(A \cap B) = \min(N(A), N(B))$
($N(\cap_{i \in I} A_i) = \inf_{i \in I} N(A_i)$)
N4 if A is crisp, then $N(A \cup \alpha) = \max(\alpha, N(A))$

where $\mu_{A \cap B}(w) = \min(\mu_A(w), \mu_B(w))$ for each $w \in \Omega$, and $A \cup \alpha$ denotes the fuzzy set defined by the membership function $\mu_{A \cup \alpha}(w) = \max(\alpha, \mu_A(w))$, for each $w \in \Omega$.

Theorem 4.1 *If N satisfies the above postulates, then there exists $\pi : \Omega \to [0,1]$ such that, for every fuzzy subset A of Ω,*

$$N(A) = \inf_{w \in \Omega} \max(1 - \pi(w), \mu_A(w)).$$

Proof: Let A be an arbitrary fuzzy subset of Ω. Define $\pi(w) = 1 - N(\overline{\{w\}})$. Since A can be put as $A = \cap_{w \in \Omega}\{\overline{\{w\}} \cup \mu_A(w)\}$, by N3, we have $N(A) = \inf_{w \in \Omega} N(\overline{\{w\}} \cup \mu_A(w))$, but by N4, $N(\overline{\{w\}} \cup \mu_A(w)) = \max(\mu_A(w), N(\overline{\{w\}}))$. Hence, $N(A) = \inf_{w \in \Omega} \max(\mu_A(w), N(\overline{\{w\}}))$, that is, $N(A) = \inf_{w \in \Omega} \max(1 - \pi(w), \mu_A(w))$. ■

4.3 Formalizing PLFC

According with Section 3.3, in general, PLFC clauses are pairs of the form

$$(\varphi(\bar{x}), f(\bar{y})),$$

where $\bar{x}$ (respectively $\bar{y}$) denotes a set of variables x_i (respectively y_j), for instance, $(p(A, x) \vee q(y), \min(\alpha, \mu_B(x), \mu_C(y)))$. The left-hand side of the pair $\varphi(\bar{x})$ is a disjunction of literals with classical predicates, free variables $\bar{x}$ and possibly with fuzzy constants. The right-hand side $f(\bar{y})$, $\bar{y} \supseteq \bar{x}$, consists of a valuation function, defined for a superset of the free variables in the left-hand

side, denoting a (variable) lower bound for the necessity value of the formula of the left-hand side.

In the next section, we describe the language and the many-valued semantics of logical formulas appearing in the left-hand side of PLFC clauses. We refer to them as *base formulas*, and their language, *base language*, denoted hereafter as PLFC*. This is needed in the rest of the section where we define a possibilistic semantics of PLFC clauses.

4.3.1 The base language of PLFC: Syntax and many-valued semantics

The *basic components* of the language PLFC* are:

- *Sorts* of variables and object constants. We distinguish a *basic sort* σ from its corresponding (fuzzy) *extended sort* $f\sigma$. A *type* is a tuple of sorts.
- A set $\mathcal{X}$ of *object variables* and a set $\mathcal{C}$ of *object constants*, each having its sort. We distinguish a *precise* object constant c of a basic sort σ from an imprecise and a fuzzy constant A of an extended sort $f\sigma$. Furthermore, if $\mathcal{FC}$ is the set of *fuzzy constants*, then there is a set $\mathcal{FC}_{\text{cuts}}$ of *imprecise constants* corresponding to the α-cuts of the fuzzy constants of $\mathcal{FC}$, i.e. if $A \in \mathcal{FC}$, then $[A]_\alpha \in \mathcal{FC}_{\text{cuts}}$, for each $0 \leq \alpha \leq 1$.
- A set $Pred$ of *regular predicates*, each one having a type.
- *Connectives* $\neg$ and $\vee$.

Definition 4.1 (term) *A term is either an object variable from $\mathcal{X}$ or an object constant from $\mathcal{C}$ (precise, fuzzy or imprecise constant).*

Definition 4.2 (atomic formula) *An atomic formula is of the form $p(x_1, ..., x_n)$, where p is a predicate symbol from $Pred$ and $t_1, ..., t_n$ are terms such that the sorts of $t_1, ..., t_n$ correspond to the type of p.*

Definition 4.3 (literal) *A literal is of the form $p(x_1, ..., x_n)$ (positive) or $\neg p(x_1, ..., x_n)$ (negative).*

Definition 4.4 (PLFC* formula) *A PLFC* formula is a disjunction of literals, either positive or negative, such that all variables are free and implicitly universally quantified.*

Next we define the semantics of PLFC* formulas, which, due to the presence of fuzzy constants, is many-valued, instead of Boolean (two-valued).

Definition 4.5 (many-valued interpretation) *A many-valued interpretation $\mathbf{M} = (U, i, m)$ maps:*

1. *each basic sort σ into a non-empty domain U_σ and each extended sort $f\sigma$ into the set $F(U_\sigma)$ of fuzzy sets of U_σ;*

2. *a predicate* p *of type* $(\sigma_1, \ldots, \sigma_k, f\sigma_{k+1}, \ldots, f\sigma_n)$ *into a* crisp *relation* $i(p) \subseteq U_{\sigma_1} \times \cdots \times U_{\sigma_n}$*; and*

3. *a precise object constant* c *of sort* σ *into a value* $m(c) \in U_\sigma$*, a fuzzy object constant* A *of sort* $f\sigma$ *into a (normalized) fuzzy set* $m(A) \in F(U_\sigma)$*, and an imprecise object constant* $[A]_\alpha$ *corresponding to the* α*-cut of a fuzzy constant* A *into the* α*-cut of* $m(A)$*. We denote the membership function of* $m(\cdot)$ *by* $\mu_{m(\cdot)}$*. The value* $m(c) \in U_\sigma$ *is also represented by a fuzzy set given by*

$$\mu_{m(c)}(u) = \begin{cases} 1, & \text{if } u = m(c) \\ 0, & \textit{for each } u \in U_\sigma \textit{ such that } u \neq m(c). \end{cases}$$

Remark that a PLFC many-valued interpretation $\mathbf{M} = (U, i, m)$ can be a conjunctive interpretation in the sense that if p is a predicate of type $(\sigma_1, \ldots, \sigma_k, f\sigma_{k+1}, \ldots, f\sigma_n)$, then for each value $u \in U_{\sigma_1}$ there can exist multiple values $(u_2, \ldots, u_n) \in U_{\sigma_2} \times \cdots \times U_{\sigma_n}$ such that $(u, u_2, \ldots, u_n) \in i(p)$.

Conjunctive interpretations are needed in PLFC to model the semantics of conjunctive information expressed by both variable weights and negative literals with fuzzy constants. For instance, to define a formal semantics for the following statements:

"Mary speaks Spanish, French and Italian"

and

"If Mary speaks Spanish, French and Italian, Mary has visited Europe",

which are respectively represented in PLFC as:

$$(speaks(Mary, x), \{Spanish, French, Italian\}(x))$$

and

$$(\neg speaks(Mary, \{Spanish, French, Italian\}) \vee visited(Mary, Europe), 1),$$

we need that interpretations (or possible states of the real world) enable Mary to speak more than one language; otherwise, for instance, the conjunctive information

"Mary speaks Spanish, French and Italian"

would be false in each possible state of the world. Moreover, since in PLFC a variable weight is interpreted as a fuzzily restricted universal quantifier, variables of PLFC* formulas have to be mapped (or evaluated) into atomic values of the domain U.

Definition 4.6 (evaluation of variables) *An evaluation of object variables is a mapping* v *assigning to each object variable* $x \in \mathcal{X}$*, of sort* σ *or* $f\sigma$*, an element* $v(x) \in U_\sigma$*.*

As above, we denote the membership function of $v(x)$ by $\mu_{v(x)}$, where $v(x)$ is always an element of U_σ, i.e.

$$\mu_{v(x)}(u) = \begin{cases} 1, & \text{if } u = v(x) \\ 0, & \text{for each } u \in U_\sigma \text{ such that } u \neq v(x). \end{cases}$$

Definition 4.7 (truth value of a PLFC* formula) *We define the truth value of a PLFC* formula under an interpretation $\mathbf{M} = (U, i, m)$ and an evaluation of variables v by cases:*

1. *For positive literals:*

   $$\|p(\ldots, x, \ldots, c, \ldots)\|_{\mathbf{M},v} = \sup_{(\ldots,u,\ldots,w,\ldots)\in i(p)} \min(\ldots, \mu_{v(x)}(u), \ldots, \mu_{m(c)}(w), \ldots).$$

2. *For negative literals:*

   $$\|\neg p(\ldots, x, \ldots, c, \ldots)\|_{\mathbf{M},v} = \sup_{(\ldots,u,\ldots,w,\ldots)\notin i(p)} \min(\ldots, \mu_{v(x)}(u), \ldots, \mu_{m(c)}(w), \ldots).$$

3. *For disjunctions of literals:*

   $$\|L_1 \vee \cdots \vee L_r\|_{\mathbf{M},v} = \max(\|L_1\|_{\mathbf{M},v}, \ldots, \|L_r\|_{\mathbf{M},v}),$$

   where $L_1, \ldots, L_r$ are (positive or negative) literals.

Finally, the truth value of a PLFC* formula φ under an interpretation $\mathbf{M}$ is defined as

$$\|\varphi\|_{\mathbf{M}} = \inf\{\|\varphi\|_{\mathbf{M},v} \mid v \textit{ is an evaluation of variables}\}.$$

It is clear that $\|\varphi\|_{\mathbf{M}}$ may take any intermediate value between 0 and 1 as soon as φ contains some fuzzy constant. Moreover, notice that the negation in this semantics is not truth-functional.

Example 4.1 Let $price(\cdot,\cdot)$ be a binary predicate of type $(\texttt{product\_name}, f\texttt{product\_price})$, let *trousers*, *coat* and *shoes* be three object constants of type `product_name`, and let *about_85* be an object constant of type f`product_price`. Further, let $\mathbf{M} = (U, i, m)$ be an interpretation such that:

1. $U = \{\ U_{\texttt{product\_name}} = \{trousers, coat, shoes\},$
   $U_{\texttt{product\_price}} = [0, 200]\text{(euros)}\ \};$
2. $i(price) = \{\ (trousers, 80),$
   $(trousers, 83),$
   $(coat, 150),$
   $(shoes, 75)\ \};$

3. $m(trousers) = trousers$,
   $m(coat) = coat$,
   $m(shoes) = shoes$,
   $m(x) = x$, for each real $x \in [0, 200]$, and
   $m(about\_85) = [80; 85; 85; 90]$.

Consider the atomic formula $price(trousers, x)$ and the following two evaluations of variable x:

$$v_1(x) = 80 \quad \text{and} \quad v_2(x) = 85.$$

Then, we have the following truth value of the formula $price(trousers, x)$ in $\mathbf{M}$ under the corresponding variable evaluations:

- $\|price(trousers, x)\|_{\mathbf{M}, v_1}$

$$\begin{aligned}
&= \sup_{(u,w)\in i(price)} \min(\mu_{m(trousers)}(u), \mu_{v_1(x)}(w)) \\
&= \max(\min(\mu_{m(trousers)}(trousers), \mu_{\{80\}}(80)), \\
&\quad \min(\mu_{m(trousers)}(trousers), \mu_{\{80\}}(83)), \\
&\quad \min(\mu_{m(trousers)}(coat), \mu_{\{80\}}(150)), \\
&\quad \min(\mu_{m(trousers)}(shoes), \mu_{\{80\}}(75))) \\
&= \max(\min(1,1), \min(1,0), \min(0,0), \min(0,0)) \\
&= 1.
\end{aligned}$$

- $\|price(trousers, x)\|_{\mathbf{M}, v_2}$

$$\begin{aligned}
&= \max(\min(1,0), \min(1,0), \min(0,0), \min(0,0)) \\
&= 0.
\end{aligned}$$

Now consider the grounded atomic formula $price(trousers, about\_85)$. In this case, we do not have any free variable and the truth value in $\mathbf{M}$ of the corresponding positive and negative literals are the following ones:

- $\|price(trousers, about\_85)\|_{\mathbf{M}}$

$$\begin{aligned}
&= \sup_{(u,w)\in i(price)} \min(\mu_{m(trousers)}(u), \mu_{m(about_85)}(w)) \\
&= \max(\min(\mu_{m(trousers)}(trousers), \mu_{m(about_85)}(80)), \\
&\quad \min(\mu_{m(trousers)}(trousers), \mu_{m(about_85)}(83))) \\
&= \max(\min(1, \mu_{[80;85;85;90]}(80)), \min(1, \mu_{[80;85;85;90]}(83))) \\
&= \max(\min(1,0), \min(1,0.6)) \\
&= 0.6.
\end{aligned}$$

- $\|\neg price(trousers, about\_85)\|_{\mathbf{M}}$

$$\begin{aligned}
&= \sup_{(u,w)\notin i(price)} \min(\mu_{m(trousers)}(u), \mu_{m(about_85)}(w)) \\
&= \sup_{x\notin\{80,83\}} \min(\mu_{m(trousers)}(trousers), \mu_{m(about_85)}(x)) \\
&= \min(1, \mu_{[80;85;85;90]}(85)) \\
&= \min(1,1) \\
&= 1.
\end{aligned}$$

and thus, $\|\neg price(trousers, about\_85)\|_{\mathbf{M}}$ cannot be computed just from $\|price(trousers, about\_85)\|_{\mathbf{M}}$ (i.e. negation in PLFC is not truth-functional). □

4.3.2 Possibilistic semantics for PLFC

A PLFC *clause* is a pair of the form

$$(\varphi(\bar{x}), f(\bar{y})),$$

where $\bar{x}$ and $\bar{y}$ denote sets of free and implicitly universally quantified variables such that $\bar{y} \supseteq \bar{x}$, $\varphi(\bar{x})$ is a PLFC* formula, and $f(\bar{y})$ is a valid valuation function which expresses the certainty of $\varphi(\bar{x})$ in terms of necessity measures. Basically, valuation functions $f(\bar{y})$ are either constant values in the real interval $[0, 1]$, or membership functions of fuzzy sets, or max-min combinations of them, or necessity measures on them. We refer to them as *valid valuation* functions.

In this section, we define a possibilistic semantics of PLFC clauses, first with constant weights, and later with variable weights.

Semantics for PLFC clauses with constant weights

Let us consider PLFC clauses of the form (φ, α), where α is a constant weight, i.e. $\alpha \in [0, 1]$. Assume that φ contains fuzzy constants, for instance take φ to be $p(A)$, where A is a fuzzy constant. Now according to Section 4.3.1, the truth value of the literal $p(A)$ under an interpretation $\mathbf{M} = (U, i, m)$ depends not only on the crisp relation $i(p)$ assigned to p, but on the fuzzy set $m(A)$. In order to measure the certainty of $p(A)$ in a possibilistic model, we need to assume that A has a fixed interpretation, in terms of its membership function, otherwise we would not be able to compute its necessity degree. Therefore, when defining the possibilistic models as possibility distributions over interpretations, we cannot take into account all possible interpretations, but only those which share a common and particular interpretation of the fuzzy constants, and hence they also have to share the domain. This leads us to define the notion of context.

Definition 4.8 (context) *Let U be a collection of non-empty domains and let m be an interpretation of object constants over U (or over $[0, 1]^U$ in the case of fuzzy constants). We further assume that U and m are such that m interprets each precise constant into a different element of the domain U, and each fuzzy constant into a fuzzy set with a normalized and left continuous membership function over U. We define the context determined by U and m, denoted $\mathcal{M}_{U,m}$, as the set of many-valued interpretations having U as a domain and m as an interpretation of object constants. Thus,*

$$\mathcal{M}_{U,m} = \{\mathbf{M} \in \mathcal{M} \mid \mathbf{M} = (U, i, m)\},$$

where $\mathcal{M}$ is the set of all many-valued interpretations.

Let us briefly discuss the reason for defining the notion of context by means of an example.

Example 4.2 Let $age(\cdot,\cdot)$ be a binary predicate of type $(\texttt{person\_name}, f\texttt{years\_old})$, let $Mary$ be an object constant of type $\texttt{person\_name}$, and let $around\_19$ be an object constant of type $f\texttt{years\_old}$. Further, let $\mathbf{M}_0 = (U, i, m_0)$ and $\mathbf{M}_1 = (U, i, m_1)$ be two interpretations such that

1. $U = \{\ U_{\texttt{person\_name}} = \{Mary\},$
   $U_{\texttt{years\_old}} = [0, 120](\text{years})\ \};$
2. $i(age) = \{(Mary, 20)\};$
3. $m_0(Mary) = m_1(Mary) = Mary,$
   $m_0(around\_19) = [17; 18; 20; 21]$, and
   $m_1(around\_19) = [18; 19; 19; 20]$.

Although Mary's age is the same value in both interpretations, we have a different truth value in each interpretation depending on the membership function of the fuzzy set assigned to the fuzzy constant $around\_19$. Thus,

$$\|age(Mary, around\_19)\|_{\mathbf{M}_0} = \mu_{[17;18;20;21]}(20) = 1$$

and

$$\|age(Mary, around\_19)\|_{\mathbf{M}_1} = \mu_{[18;19;19;20]}(20) = 0.$$

Then, for instance, the possibility distributions π on the set of all many-valued interpretations $\mathcal{M}$ that satisfy the PLFC clause

$$(age(Mary, around\_19), 1),$$

i.e. that satisfy the constraint $N([age(Mary, around\_19)] \mid \pi) \geq 1$, are those such that $\pi(\mathbf{M}_0) \leq 1$ and $\pi(\mathbf{M}_1) = 0$, and thus, $\mathbf{M}_0$ can be fully plausible while $\mathbf{M}_1$ is inadmissible, however, Mary is twenty years old in both interpretations. □

Therefore, when fixing a particular context we are ensuring that belief states modeled by normalized possibility distributions on a set of possible interpretations (or possible states) are consistent, in the sense that possible states are sharing a common view of the real world.

Now we are ready to introduce the notion of possibilistic model and possibilistic satisfiability of a PLFC clause in a context $\mathcal{M}_{U,m}$.

Definition 4.9 (possibilistic model) *Given a context $\mathcal{M}_{U,m}$, a possibilistic model is a normalized possibility distribution $\pi : \mathcal{M}_{U,m} \to [0, 1]$ on the set of many-valued interpretations $\mathcal{M}_{U,m}$.*

Definition 4.10 (necessity evaluation) *Let $\pi : \mathcal{M}_{U,m} \to [0,1]$ be a possibilistic model on a context $\mathcal{M}_{U,m}$. The necessity evaluation of a PLFC* formula φ given by π is defined as:*

$$N([\varphi] \mid \pi) = \inf_{\mathbf{M} \in \mathcal{M}_{U,m}} \max(1 - \pi(\mathbf{M}), \|\varphi\|_{\mathbf{M}}).$$

Definition 4.11 (possibilistic satisfiability) *Given a context $\mathcal{M}_{U,m}$, a possibilistic model $\pi : \mathcal{M}_{U,m} \to [0,1]$ satisfies a PLFC clause with a constant weight (φ, α), written $\pi \models^{U,m}_{PLFC} (\varphi, \alpha)$, iff $N([\varphi] \mid \pi) \geq \alpha$. Thus,*

$$\pi \models^{U,m}_{PLFC} (\varphi, \alpha) \text{ iff } \pi(\mathbf{M}) \leq \max(1 - \alpha, \mu_{[\varphi]_\alpha}(\mathbf{M}))$$

for each interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$, where

$$\mu_{[\varphi]_\alpha}(\mathbf{M}) = \begin{cases} 1, & \text{if } \|\varphi\|_{\mathbf{M}} \geq \alpha \\ 0, & \text{otherwise.} \end{cases}$$

Example 4.3 Consider the context $\mathcal{M}_{U,m}$ where U and m are as in Example 4.1, i.e.

1. $U = \{U_{\texttt{product\_name}} = \{trousers, coat, shoes\},$
   $U_{\texttt{product\_price}} = [0, 200](\text{euros})\};$
2. $m(trousers) = trousers,$
   $m(coat) = coat,$
   $m(shoes) = shoes,$
   $m(x) = x$, for each real $x \in [0, 200]$, and
   $m(about\_85) = [80; 85; 85; 90]$.

Now consider the following predicate interpretation mappings:

$$i_0(price) = \{(trousers, 83), (coat, 150), (shoes, 75)\}$$

$$i_1(price) = \{(trousers, 85), (coat, 145), (shoes, 70)\}$$

$$i_2(price) = \{(trousers, 90), (coat, 150), (shoes, 80)\}$$

and denote $\mathbf{M}_0 = (U, i_0, m)$, $\mathbf{M}_1 = (U, i_1, m)$, and $\mathbf{M}_2 = (U, i_2, m)$. Finally, consider the following possibilistic model π on the context $\mathcal{M}_{U,m}$:

$$\pi(\mathbf{M}) = \begin{cases} 0.6, & \text{if } \mathbf{M} = \mathbf{M}_0 \\ 1, & \text{if } \mathbf{M} = \mathbf{M}_1 \\ 0.2, & \text{if } \mathbf{M} = \mathbf{M}_2 \\ 0, & \text{otherwise} \end{cases}$$

Let us now compute how much π makes the atomic formula $price(trousers, about\_85)$ certain. For each interpretation $\mathbf{M}_i$, $i = 0, \ldots 2$, we have the following truth values:

$$\|price(trousers, about\_85)\|_{\mathbf{M}_0} = \mu_{m(about\_85)}(83) = 0.6$$

$$\|price(trousers, about\_85)\|_{\mathbf{M}_1} = \mu_{m(about\_85)}(85) = 1$$

$$\|price(trousers, about\_85)\|_{\mathbf{M}_2} = \mu_{m(about\_85)}(90) = 0$$

Then,

$$\begin{aligned} &N([price(trousers, about\_85)] \mid \pi) \\ &\quad = \inf_{\mathbf{M}\in\mathcal{M}_{U,m}} \max(1-\pi(\mathbf{M}), \|(price(trousers, about\_85)\|_{\mathbf{M}}) \\ &\quad = \min(0.6, 1, 0.8, 1) \\ &\quad = 0.6. \end{aligned}$$

Therefore, for instance, in the current context

$$\pi \models^{U,m}_{PLFC} (price(trousers, about\_85), 0.6),$$

but

$$\pi \not\models^{U,m}_{PLFC} (price(trousers, about\_85), 0.7).$$

□

Some very interesting and remarkable consequences of the above definition of possibilistic satisfiability of PLFC clauses with constants weights are the following ones.

Proposition 4.1 *In each context $\mathcal{M}_{U,m}$, under general* continuity conditions[1] *of the interpretation m of fuzzy constants, it holds that:*

(i) $\pi \models^{U,m}_{PLFC} (p(A), \alpha)$ *iff* $\pi \models^{U,m}_{PLFC} (p([A]_\alpha), \alpha)$

(ii) $\pi \models^{U,m}_{PLFC} (p(A, B), \alpha)$ *iff* $\pi \models^{U,m}_{PLFC} (p([A]_\alpha, [B]_\alpha), \alpha)$

(iii) $\pi \models^{U,m}_{PLFC} (p(A) \vee q(B), \alpha)$ *iff* $\pi \models^{U,m}_{PLFC} (p([A]_\alpha) \vee q([B]_\alpha), \alpha)$

where p and q can be positive or negative literals, and $[A]_\alpha$ and $[B]_\alpha$ denote the imprecise constants corresponding to the α-cuts of the fuzzy constants A and B, respectively.

Proof:

(i) According to Definition 4.11, $\pi \models^{U,m}_{PLFC} (p(A), \alpha)$ iff, for each interpretation $\mathbf{M} = (U, i, m) \in \mathcal{M}_{U,m}$, $\pi(\mathbf{M}) \leq \max(1-\alpha, \mu_{[p(A)]_\alpha}(\mathbf{M}))$, where $\mu_{[p(A)]_\alpha}(\mathbf{M}) = 1$, if $\|p(A)\|_{\mathbf{M}} \geq \alpha$; and $\mu_{[p(A)]_\alpha}(\mathbf{M}) = 0$, otherwise. But $\|p(A)\|_{\mathbf{M}} = \sup_{c\in i(p)} \mu_{m(A)}(c)$. Now, assuming a *continuity condition* on $\mu_{m(A)}$, from $\|p(A)\|_{\mathbf{M}} \geq \alpha$, we can infer that there exists $c \in i(p)$ such that $\mu_{m(A)}(c) \geq \alpha$. Then, we have $\|p(A)\|_{\mathbf{M}} \geq \alpha$ iff there exists c such that $c \in [m(A)]_\alpha$. Hence, iff $\|p([A]_\alpha)\|_{\mathbf{M}} = 1$.

[1] It refers to the assumption that m interprets fuzzy constants into left continuous membership functions. This is the case, for instance, when using trapezoidal membership functions and their cuts.

(ii) It easily follows from (i) by noticing that, under the continuity condition of the interpretation m of fuzzy constants, $\|p(A,B)\|_{\mathbf{M}} \geq \alpha$ iff there exists $(c_1, c_2) \in i(p)$ such that $\mu_{m(A)}(c_1) \geq \alpha$ and $\mu_{m(B)}(c_2) \geq \alpha$.

(iii) It follows from (i) as well as due to the fact that $\mu_{[p(A)\vee q(B)]_\alpha}(\mathbf{M}) = \max(\mu_{[(p(A)]_\alpha}(\mathbf{M}), \mu_{[q(B)]_\alpha}(\mathbf{M}))$.

■

These properties have important consequences since it means that in PLFC with (only) fuzzy constants we can in a way forget about fuzzy constants as such and focus only on imprecise but crisp constants.

Semantics for possibilistic clauses with variable weights

PLFC clauses with variable weights are of the general form

$$(\varphi(\bar{x}), f(\bar{y})),$$

where $\varphi(\bar{x})$ is a PLFC* formula with free variables $\bar{x}$ and $f(\bar{y})$ is a valid valuation function with values on $[0,1]$ depending on a set of variables $\bar{y}$, $\bar{y} \supseteq \bar{x}$, in the sense that it becomes computable in a given context as soon as the variables $\bar{y}$ are instantiated. As an example, consider the general PLFC clause

$$(p(A,x) \vee q(y), \min(\alpha, B(x), C(y))),$$

where A, B and C are fuzzy constants. But to be meaningful, first of all, such a PLFC clause has to be understood under a particular context determined by a collection of non-empty domains U and an interpretation of object constants m such that m provides meaning not only to the fuzzy constant A of the PLFC* formula but also to the fuzzy constants B and C of the valuation function by assigning them membership functions $\mu_{m(B)}$ and $\mu_{m(C)}$, respectively. Second, as we have already noted, free variables are assumed to be universally quantified. Thus, we should understand the above PLFC clause as the following collection of instantiated clauses with constant weights:

$$\{(p(A,c) \vee q(d), \min(\alpha, B(c), C(d))) \mid c \in X \text{ and } d \in Y\}$$

where c and d vary on the set of precise object constants X and Y (of the corresponding sort), inducing on possibilistic models $\pi : \mathcal{M}_{U,m} \to [0,1]$ the set of constraints:

$$N([p(A,c) \vee q(d)] \mid \pi) \geq \min(\alpha, \mu_{m(B)}(m(c)), \mu_{m(C)}(m(d))).$$

Notice that $\mu_{m(B)}(m(c)) \in [0,1]$, for each precise object constants c of sort of variable x, as well as $\mu_{m(C)}(m(d))$.

PLFC clauses of the form $(\varphi(x_1, \ldots, x_n), \alpha)$, i.e. PLFC clauses with constant weights, can be considered as a special kind of variable weight clauses if we establish the convention of considering

1. an implicit set

$$\{X_i \text{ object constant of sort of variable } x_i \mid i = 1, \ldots, n\}$$

such that m interprets each object constant X_i, for $i = 1, \ldots, n$, as $\mu_{m(X_i)}(u) = 1$ for each $u \in U_{\text{sort of variable } x_i}$; and

2. an implicit variable weight $\min(\alpha, X_1(x_1), \ldots, X_n(x_n))$ such that the resulting clause is of the form

$$(\varphi(x_1, \ldots, x_n),\ min(\alpha, X_1(x_1), \ldots, X_n(x_n))).$$

This convention can be extended to PLFC clauses with variable weights such that some variables of the base formula do not appear in the valuation function, which allows us to ensure that PLFC clauses are of the general form $(\varphi(\bar{x}), f(\bar{y}))$ with $\bar{y} \supseteq \bar{x}$. For instance, following this convention, the PLFC clauses

$$(p(x), \alpha) \quad \text{and} \quad (p(x,y), \min(\alpha, A(x)))$$

respectively correspond with

$$(p(x), \min(\alpha, X(x)) \quad \text{and} \quad (p(x,y), \min(\alpha, A(x), Y(y))),$$

where X and Y are object constants of sort of variables x and y, respectively.

Definition 4.12 (possibilistic satisfiability) *Given a context $\mathcal{M}_{U,m}$, a possibilistic model $\pi : \mathcal{M}_{U,m} \to [0,1]$ satisfies a PLFC clause with a variable weight $(\varphi(x), A(x))$, written $\pi \models^{U,m}_{PLFC} (\varphi(x), A(x))$, iff $\pi \models^{U,m}_{PLFC} (\varphi(c), A(c))$ for each precise object constant $c \in X$ (of the corresponding sort). Thus,*

$$\pi \models^{U,m}_{PLFC} (\varphi(x), A(x)) \text{ iff } N([\varphi(c)] \mid \pi) \geq \mu_{m(A)}(m(c))$$

for each object constant $c \in X$.

Proposition 4.1 extends to variable weight clauses in the following way.

Proposition 4.2 *In each context $\mathcal{M}_{U,m}$, under general* continuity conditions *of the interpretation m of fuzzy constants, it holds that:*

(i) The PLFC clause $(p(A,x), \min(\alpha, B(x)))$ is semantically equivalent to $(p([A]_{\min(\alpha,B(x))}, x), \min(\alpha, B(x)))$.

(ii) The PLFC clause $(p(A) \vee r(x), \min(\alpha, B(x)))$ is semantically equivalent to $(p([A]_{\min(\alpha,B(x))}) \vee r(x), \min(\alpha, B(x)))$.

(iii) If $\pi \models^{U,m}_{PLFC} (p([A]_\alpha, x), \min(\alpha, B(x)))$, then

$$\pi \models^{U,m}_{PLFC} (p(A,x), \min(\alpha, B(x))).$$

Proof:

(i) Direct consequence of (ii) of Proposition 4.1.

(ii) Again it is an easy consequence of Proposition 4.1, taking into account that, for each precise object constant $c \in X$ (of the corresponding sort) and for each interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$, $\|p(A) \vee r(c)\|_{\mathbf{M}} = \max(\|(p(A)\|_{\mathbf{M}}, \|r(c)\|_{\mathbf{M}})$.

(iii) It follows from (i) by noticing that, for each precise object constant $c \in X$, $[A]_\alpha \subseteq [A]_{\min(\alpha, B(c))}$.

■

4.4 Resolution and refutation in PLFC

In this section, we define a sound refutation by resolution proof method for PLFC clauses. The possibilistic entailment relation between a PLFC knowledge base and a PLFC clause is defined as in standard possibilistic logic, but is related to a particular context.

Definition 4.13 (possibilistic entailment) *Let K be a set of PLFC clauses, i.e. a PLFC knowledge base $K = \{(\varphi_i, f_i) \mid i = 1, \ldots, n\}$. We say that K entails a PLFC clause (φ, f) in a context $\mathcal{M}_{U,m}$, written $K \models^{U,m}_{PLFC} (\varphi, f)$, iff, for each possibilistic model π on $\mathcal{M}_{U,m}$, $\pi \models^{U,m}_{PLFC} (\varphi, f)$ whenever $\pi \models^{U,m}_{PLFC} (\varphi_i, f_i)$ for each $i = 1, \ldots, n$.*

From now on, we assume a particular context $\mathcal{M}_{U,m}$ to be given, and thus, the notion of soundness is relative to the context. Furthermore, we assume that $\mathcal{M}_{U,m}$ provides interpretations of fuzzy constants fulfilling the previously mentioned general continuity conditions.

In Section 3.3, we showed that when applying a resolution mechanism to PLFC clauses with variable weights, involved variables may disappear in the logical-part of a clause, but still appear in its valuation side. The *fusion rule* proposed by Dubois et al. (1996,1998b) to deal with this situation can be generalized in the following way:

$$\frac{(\varphi(\bar{x}), f(\bar{x}, y))}{(\varphi(\bar{x}), \max_{c \in Y} f(\bar{x}, c))} \text{ [GF]},$$

where $\varphi(\bar{x})$ is a PLFC* formula involving variables $\bar{x}$, $f(\bar{x}, y)$ is a valid valuation function involving variables $\bar{x}$ and y, and c vary on the set of precise object constants Y (of the corresponding sort).

The generalized fusion rule GF is obviously sound with respect to the notion of PLFC entailment in a particular context $\mathcal{M}_{U,m}$. However, this is not the case of the resolution-like rule 3.5 proposed by Dubois et al. (1996,1998b). Instead,

given a context $\mathcal{M}_{U,m}$, it can be shown that in PLFC the following *general resolution rule* holds:

$$\frac{\begin{array}{l}(\neg p(x,b,y) \vee \psi(x,y), \min(\beta, A(x), B(y))) \\ (p(C,z,r) \vee \varphi(z,r), \min(\alpha, D(z), E(r)))\end{array}}{(\psi(C,r) \vee \varphi(b,r), \min(\beta, \alpha, \mu_{m(D)}(m(b)), \delta(r)))} \text{ [GR]},$$

where

$$\delta(r) = \min(B(r), E(r), N(m(A)|[m(C)]_{\min(\alpha, \mu_{m(D)}(m(b)), E(r))})).$$

Obviously, $\delta(r)$ becomes computable in a given context $\mathcal{M}_{U,m}$ as soon as variable r is instantiated to some object constant, and then, $N(m(A)|[m(C)]_{\min(\alpha,\mu_{m(D)}(m(b)),E(r))})$ is the necessity measure of the fuzzy event $m(A)$ based on the imprecise, but non-fuzzy, information $[m(C)]_{\min(\alpha,\mu_{m(D)}(m(b)),E(r))}$ computed as

$$\begin{aligned} & N(m(A)|[m(C)]_{\min(\alpha,\mu_{m(D)}(m(b)),E(r))}) \\ & \quad = \inf_{u \in U} \max(1 - \mu_{[m(C)]_{\min(\alpha,\mu_{m(D)}(m(b)),E(r))}}(u), \mu_{m(A)}(u)) \\ & \quad = \inf_{u \in [m(C)]_{\min(\alpha,\mu_{m(D)}(m(b)),E(r))}} \mu_{m(A)}(u). \end{aligned}$$

Theorem 4.2 (soundness of the GR inference rule) *The general resolution rule GR is sound in a context $\mathcal{M}_{U,m}$ with respect to the possibilistic entailment of PLFC clauses.*

Proof: Given a context $\mathcal{M}_{U,m}$, we want to show that, for each possibilistic model $\pi : \mathcal{M}_{U,m} \to [0,1]$, if $\pi \models^{U,m}_{PLFC} (\neg p(x,b,y) \vee \psi(x,y), \min(\beta, A(x), B(y)))$ and $\pi \models^{U,m}_{PLFC} (p(C,z,r) \vee \varphi(z,r), \min(\alpha, D(z), E(r)))$, then

$$\pi \models^{U,m}_{PLFC} (\psi(C,r) \vee \varphi(b,r), \min(\beta, \alpha, \mu_{m(D)}(m(b)), \delta(r))),$$

where $\delta(r) = \min(B(r), E(r), N(m(A)|[m(C)]_{\min(\alpha,\mu_{m(D)}(m(b)),E(r))}))$.

Therefore, suppose that[2]

$$\max(1 - \pi(\mathbf{M}), 1 - \|p(c_1,b,c_2)\|_{\mathbf{M}}, \|\psi(c_1,c_2)\|_{\mathbf{M}}) \geq \min(\beta, A(c_1), B(c_2)),$$

for each precise object constants c_1 of sort of variable x, denoted hereafter as $\sigma_{A,C}$, and c_2 of sort of variable y, denoted hereafter as $\sigma_{B,E}$, and for each interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$, and that

$$\max(1 - \pi(\mathbf{M}), \|p(C,b',c_2)\|_{\mathbf{M}}, \|\varphi(b',c_2)\|_{\mathbf{M}}) \geq \min(\alpha, D(b'), E(c_2)),$$

for each precise object constants b' of sort of variable z, denoted hereafter as σ_D, and c_2 of sort $\sigma_{B,E}$, and for each interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$.

Let $\mathbf{M} = (U,i,m)$ be an interpretation of $\mathcal{M}_{U,m}$. Taking $b' = b$, we have that $\|p(C,b,c_2)\|_{\mathbf{M}} = \sup_{(u,m(b),m(c_2)) \in i(p)} \mu_{m(C)}(u)$, for each precise object constant c_2 of sort $\sigma_{B,E}$. Now,

[2]For the sake of a simpler notation, and since a context $\mathcal{M}_{U,m}$ is assumed, we simply write $A(\cdot)$, $B(\cdot)$, $D(\cdot)$ and $E(\cdot)$ instead of $\mu_{m(A)}(m(\cdot))$, $\mu_{m(B)}(m(\cdot))$, $\mu_{m(D)}(m(\cdot))$ and $\mu_{m(E)}(m(\cdot))$, respectively, in valuation functions.

- if $\|p(C,b,c_2)\|_{\mathbf{M}} < \min(\alpha, D(b), E(c_2))$, then

$$\max(1-\pi(\mathbf{M}), \|\varphi(b,c_2)\|_{\mathbf{M}}) \geq \min(\alpha, D(b), E(c_2)).$$

Thus,

$$\begin{array}{c}\max(1-\pi(\mathbf{M}), \|\psi(b,c_2)\|_{\mathbf{M}}, \|\varphi(b,c_2)\|_{\mathbf{M}}) \\ \geq \\ \min(\beta, \alpha, D(b), B(c_2), E(c_2), N(m(A)|[m(C)]_{\min(\alpha,D(b),E(c_2))})),\end{array}$$

for each precise object constant c_2 of sort $\sigma_{B,E}$ such that $\|p(C,b,c_2)\|_{\mathbf{M}} < \min(\alpha, D(b), E(c_2))$.

- if $\|p(C,b,c_2)\|_{\mathbf{M}} \geq \min(\alpha, D(b), E(c_2))$ and we assume general continuity conditions of the interpretation m of fuzzy constants, we can infer that there exists $(m(c_1), m(b), m(c_2)) \in i(p)$ such that

$$\|p(C,b,c_2)\|_{\mathbf{M}} = \sup_{(u,m(b),m(c_2))\in i(p)} \mu_{m(C)}(u) = \mu_{m(C)}(m(c_1)).$$

Thus, $\mu_{m(C)}(m(c_1)) \geq \min(\alpha, D(b), E(c_2))$, i.e.

$$m(c_1) \in [m(C)]_{\min(\alpha,D(b),E(c_2))} \text{ and } \|p(c_1,b,c_2)\|_{\mathbf{M}} = 1.$$

When $\|p(c_1,b,c_2)\|_{\mathbf{M}} = 1$, the first constraint turns into

$$\max(1-\pi(\mathbf{M}), \|\psi(c_1,c_2)\|_{\mathbf{M}}) \geq \min(\beta, A(c_1), B(c_2)).$$

Now,

  - if $(m(c_1), m(c_2)) \in i(\psi)$, then

$$\begin{array}{l}\|\psi(C,c_2)\|_{\mathbf{M}} \\ \quad = \sup_{(u,m(c_2))\in i(\psi)} \mu_{m(C)}(u) \\ \quad = \max(\mu_{m(C)}(m(c_1)), \sup_{(u,m(c_2))\in i(\psi) \text{ with } u\neq m(c_1)} \mu_{m(C)}(u)) \\ \quad \geq \min(\alpha, D(b), E(c_2)),\end{array}$$

    which implies that $\max(1-\pi(\mathbf{M}), \|\psi(b,c_2)\|_{\mathbf{M}}, \|\varphi(b,c_2)\|_{\mathbf{M}}) \geq \min(\alpha, D(b), E(c_2))$.

  - if $(m(c_1), m(c_2)) \notin i(\psi)$, then $1-\pi(\mathbf{M}) \geq \min(\beta, A(c_1), B(c_2))$. We know that $m(c_1) \in [m(C)]_{\min(\alpha,D(b),E(c_2))}$, then

$$\mu_{m(A)}(m(c_1)) \geq \inf_{u\in[m(C)]_{\min(\alpha,D(b),E(c_2))}} \mu_{m(A)}(u),$$

    where

$$\inf_{u\in[m(C)]_{\min(\alpha,D(b),E(c_2))}} \mu_{m(A)}(u) = N(m(A)|[m(C)]_{\min(\alpha,D(b),E(r))}).$$

Thus,

$$\begin{aligned} 1-\pi(\mathbf{M}) &\geq \min(\beta, A(c_1), B(c_2)) \\ &\geq \min(\beta, N(m(A)|[m(C)]_{\min(\alpha,D(b),E(r))}), B(c_2)), \end{aligned}$$

which implies that $\max(1-\pi(\mathbf{M}), \|\psi(b,c_2)\|_{\mathbf{M}}, \|\varphi(b,c_2)\|_{\mathbf{M}}) \geq \min(\beta, N(m(A)|[m(C)]_{\min(\alpha,D(b),E(r))}), B(c_2))$.

Merging both constraints we get that

$$\begin{gathered} \max(1-\pi(\mathbf{M}), \|\psi(b,c_2)\|_{\mathbf{M}}, \|\varphi(b,c_2)\|_{\mathbf{M}}) \\ \geq \\ \min(\beta, \alpha, D(b), B(c_2), E(c_2), N(m(A)|[m(C)]_{\min(\alpha,D(b),E(c_2))})), \end{gathered}$$

for each precise object constant c_2 of sort $\sigma_{B,E}$ such that $\|p(C,b,c_2)\|_{\mathbf{M}} \geq \min(\alpha, D(b), E(c_2))$.

Then, we have that

$$\begin{gathered} \max(1-\pi(\mathbf{M}), \|\psi(b,c_2)\|_{\mathbf{M}}, \|\varphi(b,c_2)\|_{\mathbf{M}}) \\ \geq \\ \min(\beta, \alpha, D(b), B(c_2), E(c_2), N(m(A)|[m(C)]_{\min(\alpha,D(b),E(c_2))})), \end{gathered}$$

for each precise object constant c_2 of sort $\sigma_{B,E}$ and interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$. Hence, we have that, for each precise object constant c_2 of sort $\sigma_{B,E}$,

$$\pi \models^{U,m}_{PLFC} (\psi(C,c_2) \vee \varphi(b,c_2), \min(\beta, \alpha, D(b), \delta(c_2))),$$

where $\delta(c_2) = \min(B(c_2), E(c_2), N(m(A)|[m(C)]_{\min(\alpha,D(b),E(c_2))}))$, and thus, we have showed that $\pi \models^{U,m}_{PLFC} (\psi(C,r) \vee \varphi(b,r), \min(\beta, \alpha, \mu_{m(D)}(m(b)), \delta(r)))$, as desired. ■

It is easy to see that GR inference rule recovers possibilistic logic resolution rule 3.1 when fuzzy constants and variable weights are not present. Moreover, given a context $\mathcal{M}_{U,m}$, a couple of particular interesting cases of GR inference rule are the following ones:

$$\frac{\begin{array}{l}(\neg p(x) \vee \psi(x), \min(\beta, A(x))) \\ (p(B), \alpha)\end{array}}{(\psi(B), \min(\beta, \alpha, N(m(A) \mid [m(B)]_\alpha)))} \ [\mathrm{GR}_1]$$

$$\frac{\begin{array}{l}(\neg p(x) \vee \psi, \min(\beta, A(x))) \\ (p(y) \vee \varphi(z), \min(\alpha, B(y), C(z)))\end{array}}{(\psi \vee \varphi(z), \min(\beta, \alpha, A(y), B(y), C(z)))} \ [\mathrm{GR}_2]$$

Notice that GR_1 inference rule is the analog of the resolution-like rule 3.5 proposed by Dubois et al. (1996,1998b) when considering a context $\mathcal{M}_{U,m}$, but differs from it in the term

$$N(m(A) \mid [m(B)]_\alpha),$$

which in pattern 3.5 was

$$N(m(A) \mid m(B)).$$

On the other hand, if we apply the generalized fusion rule GF to the resolvent of GR_2, what we get is just

$$(\psi \vee \varphi(z), \min(\beta, \alpha, C(z), Pos(m(A) \mid m(B))),$$

where

$$Pos(m(A) \mid m(B)) = \sup_{u \in U_\sigma} \min(\mu_{m(A)}(u), \mu_{m(B)}(u)).$$

One of the main advantages of the present semantics for PLFC is that provides a sound refutation mechanism based on the following properties.

Theorem 4.3 (refutation) *Let K be a set of PLFC clauses. In each context $\mathcal{M}_{U,m}$, under general* continuity conditions *of the interpretation m of fuzzy constants, it holds that:*

(i) $K \cup \{(\neg p(x), A(x))\} \models^{U,m}_{PLFC} (\bot, \beta)$ *iff* $K \models^{U,m}_{PLFC} (p(A), \beta)$.

(ii) If A is an imprecise but non-fuzzy constant, then

$$K \cup \{(\neg p(A), 1)\} \models^{U,m}_{PLFC} (\bot, \beta) \text{ iff } K \models^{U,m}_{PLFC} (p(x), \min(\beta, A(x))).$$

(iii) If $K \cup \{(\neg p(A_{>0}), 1)\} \models^{U,m}_{PLFC} (\bot, \beta)$, *then*

$$K \models^{U,m}_{PLFC} (p(x), \min(\beta, A(x))),$$

where $A_{>0}$ denotes the support of A, i.e.

$$\mu_{m(A_{>0})}(u) = \begin{cases} 1, & \text{if} \mu_{m(A)}(u) > 0 \\ 0, & \text{otherwise} \end{cases}$$

Remark that in this case we have that

$$K \models^{U,m}_{PLFC} (p(x), \min(\beta, A_{>0}(x))).$$

(iv) If $K \cup \{(\neg p(x, B_{>0}), A(x))\} \models^{U,m}_{PLFC} (\bot, \beta)$, *then*

$$K \models^{U,m}_{PLFC} (p(A, x), \min(\beta, B(x))).$$

(v) If $K \cup \{(\neg p(x), A(x)), (\neg q(B_{>0}), 1)\} \models^{U,m}_{PLFC} (\bot, \beta)$, *then*

$$K \models^{U,m}_{PLFC} (p(A) \vee q(x), \min(\beta, B(x))).$$

Proof:

(i) Let π be a possibility distribution on $\mathcal{M}_{U,m}$.

($\Rightarrow$) Suppose that $\pi \models^{U,m}_{PLFC} K$ and $\pi \models^{U,m}_{PLFC} (\neg p(x), A(x))$. This means that $\pi \models^{U,m}_{PLFC} (\neg p(c), A(c))$, for each precise object constant c of sort of variable x, denoted hereafter as σ_A, and thus, $\max(1 - \pi(\mathbf{M}), 1 - \|p(c)\|_{\mathbf{M}}) \geq \mu_{m(A)}(m(c))$, for each precise object constant c of sort σ_A and interpretation $\mathbf{M} = (U, i, m) \in \mathcal{M}_{U,m}$, or equivalently, $1 - \pi(\mathbf{M}) \geq \mu_{m(A)}(m(c))$, for each $m(c) \in U_{\sigma_A}$ such that $m(c) \in i(p)$, that is, $1 - \pi(\mathbf{M}) \geq \sup_{m(c) \in i(p)} \mu_{m(A)}(m(c))$. Now, if this condition has to imply that $\pi \models^{U,m}_{PLFC} (\bot, \beta)$, i.e. $1 - \pi(\mathbf{M}) \geq \beta$ for each interpretation $\mathbf{M}$, it must be that $1 - \pi(\mathbf{M}) \geq \max(\beta, \sup_{m(c) \in i(p)} \mu_{m(A)}(m(c)))$ for each interpretation $\mathbf{M}$, which is equivalent to force that $\pi \models^{U,m}_{PLFC} (p(A), \beta)$.

($\Leftarrow$) Suppose now that $\pi \models^{U,m}_{PLFC} (p(A), \beta)$. Moreover, suppose that $\pi \models^{U,m}_{PLFC} K$ and $\pi \models^{U,m}_{PLFC} (\neg p(x), A(x))$. Analogously to above, this means that, $\max(1 - \pi(\mathbf{M}), \|p(A)\|_{\mathbf{M}}) \geq \beta$, for each interpretation $\mathbf{M} = (U, i, m) \in \mathcal{M}_{U,m}$. Therefore,

- if $\|p(A)\|_{\mathbf{M}} < \beta$, it must be that

$$1 - \pi(\mathbf{M}) \geq \beta > \sup_{m(c) \in i(p)} \mu_{m(A)}(m(c)).$$

- if $\|p(A)\|_{\mathbf{M}} \geq \beta$, by general continuity conditions, it must exist $c \in \sigma_A$ such that $m(c) \in i(p)$ and $\mu_{m(A)}(m(c)) \geq \beta$. Hence, $\|p(c)\|_{\mathbf{M}} = 1$, and thus,

$$\max(1 - \pi(\mathbf{M}), 1 - \|p(c)\|_{\mathbf{M}}) = 1 - \pi(\mathbf{M}) \geq \mu_{m(A)}(m(c)) \geq \beta.$$

Therefore, in both cases it holds that $\pi \models^{U,m}_{PLFC} (\bot, \beta)$.

(ii) Let π be a possibility distribution on $\mathcal{M}_{U,m}$ and let A be an imprecise but non-fuzzy constant, i.e. $\mu_{m(A)}(m(c)) \in \{0, 1\}$ for each precise object constant c of sort σ_A. Assume that $\pi \models^{U,m}_{PLFC} K$. Now, $\pi \models^{U,m}_{PLFC} (\neg p(A), 1)$ iff $\pi(\mathbf{M}) = 0$ for each interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$ such that $\|\neg p(A)\|_{\mathbf{M}} = 0$, that is, for each interpretations $\mathbf{M} \in \mathcal{M}_{U,m}$ such that $\|p(c)\|_{\mathbf{M}} = 1$ for each $c \in \sigma_A$ such that $\mu_{m(A)}(m(c)) = 1$. Thus, this condition implies $\pi \models^{U,m}_{PLFC} (\bot, \beta)$ (i.e. $\pi(\mathbf{M}) \leq 1 - \beta$ for each interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$) iff one has $\pi(\mathbf{M}) \leq 1 - \beta$ for each interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$ such that there exists $c \in \sigma_A$, with $\mu_{m(A)}(m(c)) = 1$, such that $\|p(c)\|_{\mathbf{M}} = 0$. But this is exactly the condition for having $\pi \models^{U,m}_{PLFC} (p(x), \min(\beta, A(x)))$, namely, for each $c \in \sigma_A$ such that $\mu_{m(A)}(m(c)) = 1$, $\max(1 - \pi(\mathbf{M}), \|p(c)\|_{\mathbf{M}}) \geq \beta$, and thus, $1 - \pi(\mathbf{M}) \geq \beta$ if there exists $c \in \sigma_A$, with $\mu_{m(A)}(m(c)) = 1$, such that $\|p(c)\|_{\mathbf{M}} = 0$.

(iii) It is a direct consequence of (ii), taking into account that $\mu_{m(A_{>0})}(m(c)) \geq \mu_{m(A)}(m(c))$ for each precise object constant c of sort σ_A.

(iv) It is a consequence of (i) and (iii).

(v) Again it is an easy consequence of (i) and (iii), taking into account that $\|p(A) \vee q(c)\|_{\mathbf{M}} = \max(\|p(A)\|_{\mathbf{M}}, \|q(c)\|_{\mathbf{M}})$ for each precise object constant c of sort σ_A and interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$.

■

Based on these soundness results, in the next section, we define a refutation by resolution proof method that checks whether a PLFC knowledge base K entails, in a given context $\mathcal{M}_{U,m}$, a PLFC clause (φ, f). Roughly speaking, the method, as usual, consists of negating the query (φ, f), adding it to the knowledge base K, and checking to what degree the contradiction is deduced by applying some inference rules. Hence, in order to extend refutation by resolution to PLFC, *negation* of PLFC queries is first defined.

Definition 4.14 (PLFC query) *A PLFC query is a PLFC clause of the form*

$$(\varphi(\bar{x}), f(\bar{x})),$$

where $\bar{x}$ denotes a set of free and implicitly universally quantified variables, $\varphi(\bar{x})$ is a PLFC formula, and $f(\bar{x})$ is either a constant value in the real interval $[0,1]$, or membership functions of fuzzy sets, or min combinations of them.*

Remark that a PLFC clause of the form $(\varphi(\bar{x}), f(\bar{y}))$, with $\bar{y} \supseteq \bar{x}$, can be transformed into $(\varphi(\bar{x}), f(\bar{x}))$ by applying the GF inference rule.

Definition 4.15 (negation of a PLFC query) *We define the negation of a PLFC query (φ, f) by cases:*

1. *When φ is a grounded literal:*

$$\begin{aligned}
\neg(p(A), f) \;&\textit{ is }\; (\neg p(x), A(x)),\\
\neg(\neg p(A), f) \;&\textit{ is }\; (p(x), A(x)),\\
\neg(p(A_1, \ldots, A_p), f) \;&\textit{ is }\; (\neg p(x_1, \ldots, x_p), \min(A_1(x_1), \ldots, A_p(x_p))),\\
\neg(\neg p(A_1, \ldots, A_p), f) \;&\textit{ is }\; (p(x_1, \ldots, x_p), \min(A_1(x_1), \ldots, A_p(x_p))).
\end{aligned}$$

2. *When φ is a literal with free variables:*

$$\begin{aligned}
\neg(p(y), \min(C(y), f)) \;&\textit{ is }\; (\neg p(C_{>0}), 1),\\
\neg(\neg p(y), \min(C(y), f)) \;&\textit{ is }\; (p(C_{>0}), 1),\\
\neg(p(A, y), \min(C(y), f)) \;&\textit{ is }\; (\neg p(x, C_{>0}), A(x)),\\
\neg(\neg p(A, y), \min(C(y), f)) \;&\textit{ is }\; (p(x, C_{>0}), A(x)),\\
\neg(p(A_1, \ldots, A_p, y_1, \ldots, y_l), \min(C_1(y_1), \ldots, C_l(y_l), f))&\\
\textit{is}&\\
(\neg p(x_1, \ldots, x_p, C_{1>0}, \ldots, C_{l>0}), \min(A_1(x_1), \ldots, A_p(x_p))),&\\
\neg(\neg p(A_1, \ldots, A_p, y_1, \ldots, y_l), \min(C_1(y_1), \ldots, C_l(y_l), f))&\\
\textit{is}&\\
(p(x_1, \ldots, x_p, C_{1>0}, \ldots, C_{l>0}), \min(A_1(x_1), \ldots, A_p(x_p))).&
\end{aligned}$$

where $C_{>0}$ and $C_{i>0}$, for $i = 1, \ldots, l$, denote the support of C and C_i, respectively.

3. *When φ is a disjunction of literals:*

$$\neg(L_1 \vee \cdots \vee L_r, f) \quad is \quad \{\neg(L_i, f) \mid i = 1, \ldots, r\},$$

where $L_1, \ldots, L_r$ are (positive or negative) literals.

In cases 1 and 2, the valuation function f can be assumed to be a constant value, while in case 3 it denotes a general query valuation function.

Example 4.4 Let us consider five particular but illustrative cases. Namely, if we want to check whether a knowledge base K entails (satisfies), with a necessity of at least β (threshold), the following information (statements):

(i) "Mary speaks Spanish, French or Italian"

(ii) "Mary speaks Spanish, French and Italian"

(iii) "For each language y in $\{Spanish, French, Italian\}$, either Mary or Peter speaks y"

(iv) "[Mary does not speak Spanish, French or Italian] or [Mary speaks Spanish, French and Italian]"

(v) "[For each language y in $\{Spanish, French, Italian\}$, either Mary or Peter do not speak y] or [Mary speaks either Spanish or French or Italian, and so does Peter]"

we have to

1. negate, respectively, the following PLFC queries:

   (i) $(speaks(Mary, \{Spanish, French, Italian\}), \beta)$

   (ii) $(speaks(Mary, y), \min(\beta, \{Spanish, French, Italian\}(y)))$

   (iii) $(speaks(\{Mary, Peter\}, y), \min(\beta, \{Spanish, French, Italian\}(y)))$

   (iv) $(\neg speaks(Mary, \{Spanish, French, Italian\}) \vee$
   $speaks(Mary, y), \min(\beta, \{Spanish, French, Italian\}(y)))$

   (v) $(\neg speaks(\{Mary, Peter\}, y) \vee speaks(z, \{Spanish, French, Italian\}),$
   $\min(\beta, \{Spanish, French, Italian\}(y), \{Mary, Peter\}(z)))$

2. add, respectively, to K the following PLFC clauses:

   $\neg$(i) $(\neg speaks(x, y), \min(Mary(x), \{Spanish, French, Italian\}(y)))$
   Since Mary is a precise object constant, this PLFC clause is semantically equivalent to

$$(\neg speaks(Mary, y), \{Spanish, French, Italian\}(y))$$

   $\neg$(ii) $(\neg speaks(Mary, \{Spanish, French, Italian\}), 1)$

¬(iii) $(\neg speaks(x, \{Spanish, French, Italian\}), \{Mary, Peter\}(x))$
Remark that this PLFC clause expresses that

"Mary does not speak Spanish, French or Italian, and so does Peter"

which does not exactly correspond with the "*logical negation*" of statement (iii), i.e.

"for some language y in $\{Spanish, French, Italian\}$,
neither Mary nor Peter speak y",

which cannot be expressed in PLFC (see Section 3.3).
However, if $K \cup \{\neg(\text{iii})\}$ entails $(\bot, \beta)$, intuitively, K should entail, with a necessity of at least β, the "*logical negation*" of statement ¬(iii), i.e.

"Mary speaks Spanish, French and Italian, or Peter does"

which, again, cannot be expressed in PLFC, but which, obviously, entails statement (iii).

¬(iv) $\{ (speaks(Mary, y), \{Spanish, French, Italian\}(y)),$
$(\neg speaks(Mary, \{Spanish, French, Italian\}), 1) \}$
Remark that these PLFC clauses express the following contradictory statements:

"Mary speaks Spanish, French and Italian"

and

"Mary does not speak Spanish, French or Italian",

respectively. On the other hand, applying the resolution rule GR to ¬(iv), we get $(\bot, 1)$. Hence, by the soundness of the GR inference rule and Theorem 4.3, we have that (v) is absolutely certain in any context.

¬(v) $\{ (speaks(x, \{Spanish, French, Italian\}, \{Mary, Peter\}(x)),$
$(\neg speaks(\{Mary, Peter\}, t), \{Spanish, French, Italian\}(t)) \}$
Remark that statement (v) is certain in any context, and thus, the "*logical negation*" of (v) should express contradictory information. But, like in case (iii), ¬(v) does not exactly correspond with the "*logical negation*" of (v). However, when variable t of

$$\neg speaks(\{Mary, Peter\}, t)$$

takes the value $\{Spanish, French, Italian\}$, we get

$$\exists x \in \{Mary, Peter\},\ \neg(speaks(x, \{Spanish, French, Italian\}))$$

which is interpreted as:

$$\exists x \in \{Mary, Peter\}\, \forall y \in \{Spanish, French, Italian\}, \neg speaks(x, y).$$

Hence, ¬(v) PLFC clauses (with variable t instantiated to $\{Spanish, French, Italian\}$) express the following contradictory statements:

"Mary speaks Spanish, French or Italian, and so does Peter"

and

"Mary or Peter do not speak neither Spanish nor French nor Italian",

respectively. On the other hand, applying the resolution rule GR to ¬(v), we get $(\bot, 1)$. Hence, by the soundness of the GR inference rule and Theorem 4.3, we have that (v) is absolutely certain in each context.

3. and, finally, Theorem 4.3 guarantees that if the knowledge base K augmented with ¬(i) (respectively, with ¬(ii), ¬(iii), ¬(iv) and ¬(v)) derives, by applying some (sound) inference rules, $(\bot, \beta)$, then (i) (respectively, (ii), (iii), (iv) and (v)) is a logical consequence of K. However, nothing is said about the converse.

□

4.5 Automated deduction

In this section, we define an automated deduction method for PLFC based on refutation through resolution. We then need an algorithm that let us know when two literals $p(t_1, \ldots, t_n)$ and $\neg p(s_1, \ldots, s_n)$ can be resolved. Moreover, we need an algorithm that automatically computes a set of substitutions that must be applied on the resolvent clause. But, in this framework, we cannot borrow the unification concept used in classical first-order logic programming systems. Let us consider one illustrative example. For instance, from

$$(\neg p(A) \vee \psi, 1) \quad \text{and} \quad (p(A), 1),$$

which, if A is not fuzzy, are interpreted respectively as

$$\text{“}[\exists x \in A,\ \neg p(x)] \vee \psi\text{”} \quad \text{and} \quad \text{“}\exists x \in A,\ p(x)\text{”},$$

we can infer ψ iff A is a precise constant. Then, resolution for $\neg p(A)$ and $p(A)$ must fail unless A is a precise constant, even though, obviously, $p(A)\theta = p(A)\theta$ for each (classical) substitution of variables θ. Therefore, from now on, we refer ourselves to a *most general substitution of two literals in a resolution step*. The first part of this section is about the formalization and computation of a most general substitution of two literals in a resolution step, the second one extends the calculus of the logic with a *merging* and a *transformation* inference rules, and the last one describes a proof procedure by refutation through (i) the general resolution rule GR; (ii) the generalized fusion rule GF already proposed

by Dubois et al. (1996,1998b) and applied to a resolvent clause when some variables have disappeared in its base formula but still appear in its valuation side; and (iii) the new merging and transformation rules.

4.5.1 Most general substitution

In the following, we formally define a most general substitution of two literals in a resolution step, we describe how it must be applied to a PLFC clause and, finally, we give an algorithm for its automatic computation.

Definition 4.16 (substitution term) *A substitution term of a variable is either a variable, a precise constant or an imprecise but non-fuzzy constant.*

Notice that fuzzy constants are not substitution terms. This is due to Propositions 4.1 and 4.2 which safely allow us to focus only on crisp constants.

Definition 4.17 (substitution) *A substitution is a mapping from variables to substitution terms, and is written as* $\theta = \{x_1/t_1, ..., x_n/t_n\}$, *where the variables* $x_1, ..., x_n$ *are different and* $x_i \neq t_i$, *for* $i = 1, \ldots, n$.

Substitutions operate on expressions. By an expression we mean a term, a PLFC$*$ formula, or a PLFC clause with a constant or a variable weight.

Definition 4.18 (instance of an expression) *Let* E *be an expression and let* θ *be a substitution. The instance of* E *given by* θ, *written* $E\theta$, *stands for the result of applying* θ *to* E *which is obtained by simultaneously replacing each occurrence in* E *of a variable from the domain of* θ *by the corresponding substitution term. After applying a substitution to a PLFC clause we can obtain in the valuation side expressions like* $f_1(B)$ *or* $f_2(B_1, \ldots, B_n)$, *where* f_1 *and* f_2 *are valid valuation functions in the model and* $B, B_1, \ldots, B_n$ *are imprecise but non-fuzzy constants. Then, given a context* $\mathcal{M}_{U,m}$, $f_1(B)$ *is computed as*

$$N(m(f_1) \mid m(B)) = \inf_{u \in U_{\sigma_B}} \max(1 - \mu_{m(B)}(u), \mu_{m(f_1)}(u)),$$

and $f_2(B_1, \ldots, B_n)$ *as*

$$\begin{aligned} & N(m(f_2) \mid \min(m(B_1), \ldots, m(B_n))) \\ & \quad = \inf_{(u_1,\ldots,u_n) \in U_{\sigma_{B_1}} \times \cdots \times U_{\sigma_{B_n}}} \max(1 - \min(\mu_{m(B_1)}(u_1), \ldots, \\ & \qquad \mu_{m(B_n)}(u_n)), \mu_{m(f_2)}(u_1, \ldots, u_n)) \\ & \quad = \inf_{(u_1,\ldots,u_n) \in U_{\sigma_{B_1}} \times \cdots \times U_{\sigma_{B_n}}} \max(1 - \mu_{m(B_1)}(u_1), \ldots, \\ & \qquad 1 - \mu_{B_n}(u_n), \mu_{m(f_2)}(u_1, \ldots, u_n)), \end{aligned}$$

where $\mu_{m(f_1)}$ *and* $\mu_{m(f_2)}$ *are the membership function of the fuzzy set that results from applying the interpretation function* m *to the object constants involved in* f_1 *and* f_2, *respectively, and* σ_B *and* σ_{B_i}, *for* $i = 1, \ldots, n$, *are the basic sorts of* B *and* B_i, *respectively.*

Definition 4.19 (composition of substitutions) *Let* $\theta = \{x_1/t_1, ..., x_n/t_n\}$ *and* $\eta = \{y_1/s_1, ..., y_m/s_m\}$ *be two substitutions. The composition of* θ *and* η*, written* $\theta\eta$*, is a substitution defined by removing from the set* $\{x_1/t_1\eta, ..., x_n/t_n\eta, y_1/s_1, ..., y_m/s_m\}$ *those pairs* $x_i/t_i\eta$ *for which* $x_i = t_i\eta$ *and those pairs* y_i/s_i *for which* $y_i \in \{x_1, ..., x_n\}$.

Definition 4.20 (most general substitution) *Let* θ *and* η *be two substitutions.* θ *is more general than* η *if for some substitution* γ *we have* $\eta = \theta\gamma$.

Definition 4.21 (substitution of two atomic formulas in a context) *Let* φ *and* ϕ *be two atomic formulas with predicate symbol* p *of arity* n*, let* θ *be a substitution and let* $\mathcal{M}_{U,m}$ *be a context. If* $\varphi\theta$ *is of the form* $p(s_1, \ldots, s_n)$ *and* $\phi\theta$ *is of the form* $p(t_1, \ldots, t_n)$*, we say that* θ *is a substitution of* φ *and* ϕ *in the context* $\mathcal{M}_{U,m}$ *if for each pair* (s_i, t_i)*,* $i = 1, \ldots, n$*, it holds that either* $s_i = t_i$*, or* s_i *and* t_i *are object constants and* $\mu_{m(s_i)} = \mu_{m(t_i)}$*. Furthermore, we say that* θ *is a most general substitution of* φ *and* ϕ *in the context* $\mathcal{M}_{U,m}$ *(or mgs for short) if it is more general than any other substitution of* φ *and* ϕ *in the context* $\mathcal{M}_{U,m}$*. In fact, mgs's are unique modulo renaming of variables and object constants (i.e. object constants with a same interpretation).*

The next algorithm takes two PLFC literals and a context, and produces an mgs of the atomic formulas in the context if the literals can be resolved (i.e. if literals express contradictory information); otherwise, reports an error message. We follow the presentation of Apt (1990), based upon Herbrand's original algorithm, first presented by Martelli and Montanari (1982), which deals with solutions of finite sets of term equations.

Algorithm 4.1 mgs of two literals in a resolution step

Input:

- Two literals with predicate symbol p of arity n of the form $\neg p(s_1, \ldots, s_n)$ and $p(t_1, \ldots, t_n)$, where each term is either a variable, a precise constant, or an imprecise but non-fuzzy constant, and such that they do not have any variable in common.
- A context determined by a collection of non-empty domains U and an interpretation m of object constants over U.

Output: An mgs θ of $p(s_1, \ldots, s_n)$ and $p(t_1, \ldots, t_n)$ in the context determined by U and m if the literals can be resolved; otherwise, an error message.

Initialization: From $p(s_1, ..., s_n)$ and $p(t_1, ..., t_n)$ we construct a set of substitutions S of the form $\{s_1/t_1, ..., s_n/t_n\}$.

Method: Choose a substitution $\{s_i/t_i\}$ from S and perform the associated action until either S remains unchanged or the algorithm fails:

1. If s_i and t_i are object constants, then

- if $\mu_{m(s_i)} = \mu_{m(t_i)}$ and s_i, t_i are precise constants then, delete the substitution $\{s_i/t_i\}$ from S;
- otherwise, fail.

2. If s_i is an object constant and t_i is a variable, then replace $\{s_i/t_i\}$ by $\{t_i/s_i\}$ in S.
3. If s_i is a variable, then
   - if $s_i = t_i$ then delete the substitution $\{s_i/t_i\}$ from S;
   - else, if s_i has another occurrence in S, then
     - if s_i appears in t_i, then fail;
     - otherwise, perform the substitution $\{s_i/t_i\}$ in every other term in S.

Final treatment: If $\{s'_1/t'_1, \ldots, s'_k/t'_k)\}$ is the resulting set of substitutions, then $\theta := \{s'_1/t'_1, ..., s'_k/t'_k\}$.

Example 4.5 Let us consider the PLFC clauses used in the general resolution rule GR of Section 4.4:

s1: $(\neg p(x, b, y) \vee \psi(x, y), \min(\beta, A(x), B(y)))$

s2: $(p(C, z, r) \vee \varphi(z, r), \min(\alpha, D(z), E(r)))$

By Proposition 4.2, **s2** is equivalent to:

s2': $(p([C]_{f(z,r)}, z, r) \vee \varphi(z, r), f(z, r))$

where

$$f(z, r) = \min(\alpha, D(z), E(r)).$$

The initial set of substitutions computed by Algorithm 4.1 for $\neg p(x, b, y)$ and $p([C]_{f(z,r)}, z, r)$ is:

$$S = \{x/[C]_{f(z,r)}, b/z, y/r\}.$$

Since b is an object constant and z is a variable, the substitution $\{b/z\}$ is replaced by $\{z/b\}$ in S, and thus,

$$S = \{x/[C]_{f(z,r)}, z/b, y/r\}.$$

Since $\{z/b\} \in S$ and variable z has another occurrence in S, the substitution $\{z/b\}$ is performed in every other term in S, and thus, $\{z/b\}$ is performed over the substitution term $[C]_{f(z,r)}$. Hence, the output of Algorithm 4.1 for $\neg p(x, b, y)$ and $p([C]_{f(z,r)}, z, r)$ is an mgs

$$\theta = \{x/[C]_{f(b,r)}, z/b, y/r\}.$$

On the other hand, the resolvent of **s1** and **s2'** before applying the mgs θ is

s3: $(\psi(x, y) \vee \varphi(z, r), \min(\beta, \alpha, A(x), B(y), D(z), E(r)))$

Now, applying θ to **s3** we get

s3': $(\psi([C]_{f(b,r)}, r) \vee \varphi(b,r), \min(\beta, \alpha, A([C]_{f(b,r)}), B(r), D(b), E(r)))$

where, in a particular context $\mathcal{M}_{U,m}$,

$$D(b) = N(m(D) \mid m(b)) = \mu_{m(D)}(m(b))$$

and

$$A([C]_{f(b,r)}) = N(m(A) \mid [m(C)]_{\min(\alpha, \mu_{m(D)}(m(b)), E(r))}).$$

Finally, by Proposition 4.2, **s3'** is equivalent to

s3": $(\psi(C,r) \vee \varphi(b,r), \min(\beta, \alpha, \mu_{m(D)}(m(b)), \delta(r))),$

where

$$\delta(r) = \min(N(m(A)|[m(C)]_{\min(\alpha, \mu_{m(D)}(m(b)), E(r))}), B(r), E(r)),$$

which is exactly the resolvent of GR when applied to **s1** and **s2**. □

Example 4.6 Suppose that we are interested in resolving the following two PLFC clauses:

s1: $(\neg p(x, B) \vee \psi(x), \min(\beta, A(x)))$

s2: $(p(C, y), \min(\alpha, D(y)))$

By Proposition 4.2, **s1** and **s2** are equivalent, respectively, to

s1': $(\neg p(x, [B]_{\min(\beta, A(x))}) \vee \psi(x), \min(\beta, A(x)))$

s2': $(p([C]_{\min(\alpha, D(y))}, y), \min(\alpha, D(y)))$

The initial set of substitutions computed by Algorithm 4.1 for $\neg p(x, [B]_{\min(\beta, A(x))})$ and $p([C]_{\min(\alpha, D(y))}, y)$ is:

$$S = \{x/[C]_{\min(\alpha, D(y))}, [B]_{\min(\beta, A(x))}/y\}.$$

Since $[B]_{\min(\beta, A(x))}$ is an object constant and y is a variable, the substitution $\{[B]_{\min(\beta, A(x))}/y\}$ is replaced by $\{y/[B]_{\min(\beta, A(x))}\}$ in S, and thus,

$$S = \{x/[C]_{\min(\alpha, D(y))}, y/[B]_{\min(\beta, A(x))}\}.$$

Choosing $\{x/[C]_{\min(\alpha, D(y))}\} \in S$, since variable x has another occurrence in S, the substitution $\{x/[C]_{\min(\alpha, D(y))}\}$ is performed over the substitution term $[B]_{\min(\beta, A(x))}$, and thus,

$$S = \{x/[C]_{\min(\alpha, D(y))}, y/[B]_{\min(\beta, A([C]_{\min(\alpha, D(y))}))}\}.$$

Now, taking $\{y/[B]_{\min(\beta, A([C]_{\min(\alpha, D(y))}))}\} \in S$, since variable y has another occurrence in S, the substitution $\{y/[B]_{\min(\beta, A([C]_{\min(\alpha, D(y))}))}\}$ should be performed in every other term in S. However, the substitution term of y depends on y itself. Therefore, the algorithm halts with failure in this case. □

4.5.2 Generalized merging rule

We have seen that in PLFC, due to the disjunctive interpretation of fuzzy constants, the unification (in the classical sense) between fuzzy constants is not allowed. However, as variable weights are interpreted as conjunctive information, (fuzzy) unification is allowed between variable weights and fuzzy constants and it is implicitly performed by the general resolution rule GR and computed through a necessity measure of matching between fuzzy events. For instance, given a context $\mathcal{M}_{U,m}$, applying the GR inference rule to

$$(\neg p(x) \vee \varphi, A(x)) \quad \text{and} \quad (p(B), \alpha),$$

we conclude

$$(\varphi, \min(\alpha, N(m(A) \mid [m(B)]_\alpha))),$$

where

$$N(m(A) \mid [m(B)]_\alpha) = \inf_{u \in [m(B)]_\alpha} \mu_{m(A)}(u).$$

Hence, as already pointed out by Dubois et al. (1996,1998b), resolution in PLFC produces conclusions which are all the stronger as variable weights are large. Indeed, with the resolution rule GR

$$N(m(A) \mid [m(B)]_\alpha) \geq N(m(A') \mid [m(B)]_\alpha)$$
$$\text{if}$$
$$\mu_{m(A)}(u) \geq \mu_{m(A')}(u) \text{ for each } u \in [m(B)]_\alpha.$$

Before developing the refutation by resolution proof procedure, we stress that in PLFC an extension of the merging rule 3.6 proposed by Dubois et al. (1996,1998b) holds. Let us first introduce two new definitions about substitutions.

Definition 4.22 (renaming) *A substitution of the form $\theta = \{x_1/t_1, ..., x_n/t_n\}$ is called a renaming if the substitution terms $t_1, \ldots, t_n$ are different variables.*

Definition 4.23 (variant) *Let $E_1 = (\varphi(\overline{x}), f_1)$ and $E_2 = (\varphi(\overline{y}), f_2)$ be two PLFC clauses, where f_1 and f_2 are valid valuations functions involving at least the set of variables $\overline{x}$ and $\overline{y}$, respectively. E_1 is a variant of E_2 if there exits a renaming θ such that no variable of E_1 omitted in the domain of θ appears in the range of θ, and $\varphi(\overline{x})\theta = \varphi(\overline{y})$.*

The following *generalized merging* rule is an extension of the one proposed by Sandri and Godo (1999) and it corresponds to the following pattern:

$$\frac{(\varphi(\overline{x}), f_1), (\varphi(\overline{y}), f_2)}{(\varphi(\overline{y}), \max(f_1\theta, f_2))} \text{ [GM]},$$

where θ is a renaming such that $(\varphi(\overline{x}), f_1)$ is a variant of $(\varphi(\overline{y}), f_2)$. It can be proved that this inference rule is sound, but we can show more.

Proposition 4.3 *If θ is a renaming such that $(\varphi(\overline{x}), f_1))$ is a variant of $(\varphi(\overline{y}), f_2)$, then, for each context $\mathcal{M}_{U,m}$, the set of PLFC clauses $\{(\varphi(\overline{x}), f_1), (\varphi(\overline{y}), f_2)\}$ is semantically equivalent to the PLFC clause $(\varphi(\overline{y}), \max(f_1\theta, f_2))$.*

Proof: If θ is a renaming, then, for each possibilistic model $\pi : \mathcal{M}_{U,m} \to [0,1]$, $\pi \models^{U,m}_{PLFC} (\varphi(\overline{x}), f_1)$ iff $\pi \models^{U,m}_{PLFC} (\varphi(\overline{x}), f_1)\theta$ iff $\pi \models^{U,m}_{PLFC} (\varphi(\overline{x})\theta, f_1\theta)$ iff $\pi \models^{U,m}_{PLFC} (\varphi(\overline{y}), f_1\theta)$.

Now, $\pi \models^{U,m}_{PLFC} \{(\varphi(\overline{y}), f_1\theta), (\varphi(\overline{y}), f_2)\}$ iff, for each $\mathbf{M} \in \mathcal{M}_{U,m}$, $\max(1 - \pi(\mathbf{M}), \|\varphi(\overline{y})\|_{\mathbf{M}}) \geq \max(f_1\theta, f_2)$. Thus, iff $\pi \models^{U,m}_{PLFC} (\varphi(\overline{y})), \max(f_1\theta, f_2))$. ■

The usefulness of the GM inference rule can be verified by means of a simple example.

Example 4.7 Let A, B and C be three imprecise object constants of sort σ, and let $\mathcal{M}_{U,m}$ be a context such that

1. $U = \{U_\sigma = \mathbb{N}\}$;
2. $m(A) = \{1, 2\}$,
   $m(B) = \{2, 3\}$, and
   $m(C) = \{1, 2, 3\}$.

Let us suppose we have the following PLFC clauses:

s1: $(p(x), A(x))$

s2: $(p(y), B(y))$

s3: $(\neg p(C), 1)$

Resolving **s1** and **s3** we conclude $(\bot, 0)$, which is the same result we obtain when we resolve **s2** and **s3**. Therefore, without the use of the merging rule, we only obtain $(\bot, 0)$ as final result. However, **s1** is a variant of **s2** then, if **s1** and **s2** are fused together we obtain

$$(p(y), f(y)), \text{ with } f(y) = \max(A(y), B(y))$$

which resolved with **s3** finally yields

$$(\bot, f([C]_1)),$$

where

$$\begin{aligned} f([C]_1) &= N(\max(m(A), m(B)) \mid [m(C)]_1) \\ &= \inf_{u \in m(C)} \max(\mu_{m(A)}(u), \mu_{m(B)}(u)) \\ &= \inf_{u \in \{1,2,3\}} \max(\mu_{\{1,2\}}(u), \mu_{\{2,3\}}(u)) \\ &= 1. \end{aligned}$$

□

Notice in the previous example that **s1** and **s2** can be either clauses of a knowledge base, or the resolvents from a previous resolution step. Therefore, the generalized merging rule GM must be applied over each pair of clauses of a knowledge base, before starting the refutation proof procedure, and after each resolution step during the proof procedure.

Finally, precise object constants appearing in the logic component of PLFC clauses can be transformed into variable weights. Indeed, a PLFC clause of the form

$$(\varphi(\bar{x}, a), f(\bar{y})),$$

where a is a precise object constant and $\bar{x} \subseteq \bar{y}$, is semantically equivalent to the PLFC clause

$$(\varphi(\bar{x}, t), \min(f(\bar{y}), a(t))),$$

where t is a variable such that $t \notin \bar{y}$. On the other hand, we have seen that in PLFC (i) unification (in the classical sense) between precise and non-precise object constants is not allowed; (ii) (fuzzy) unification is performed between object constants and variable weights; and (iii) the merging rule GM can be used to get larger variable weights. This points out that before applying the GM inference rule to a knowledge base, it is interesting to transform each precise object constant (appearing in the logic component of clauses) into variable weights. Hence, given a context $\mathcal{M}_{U,m}$, we define the following *transformation* rule:

$$\frac{(\varphi(\bar{x}, a), f(\bar{y}))}{(\varphi(\bar{x}, t), \min(f(\bar{y}), a(t)))} \text{ [TR]},$$

if t is a variable such that $t \notin \bar{y}$ and a is a precise object constant, i.e. $\mu_{m(a)}(u) = 0$, for each $u \in U$ such that $u \neq m(a)$, and $\mu_{m(a)}(m(a)) = 1$.

The usefulness of the TR inference rule can be verified by means of a simple example.

Example 4.8 We extend Example 4.7 with a precise object constant a of sort σ such that

$$\mu_{m(a)}(u) = \begin{cases} 1, & \text{if } u = 1 \\ 0, & \text{otherwise.} \end{cases}$$

Suppose now that we have the following PLFC clauses:

s1: $(p(a), 1)$

s2: $(p(y), B(y))$

s3: $(\neg p(C), 1)$

where, in the context of Example 4.7,

$$m(B) = \{2, 3\} \quad \text{and} \quad m(C) = \{1, 2, 3\}.$$

On the one hand, Algorithm 4.1 fails for $p(a)$ and $\neg p(C)$, and thus, **s1** and **s3** cannot be resolved. On the other hand, resolving **s2** and **s3** we conclude $(\bot, 0)$, and clauses **s1** and **s2** cannot be merged. However, **s1** is semantically equivalent to the variable weight clause:

s1': $(p(x), a(x))$

Now merging **s1'** with **s2** we obtain

$$(p(y), \max(a(y), B(y))),$$

which resolved with **s3** finally yields $(\bot, 1)$. □

4.5.3 Refutation procedure

Given a context $\mathcal{M}_{U,m}$, *refutation by resolution* proof method is extended to PLFC as follows. Let $K = \{(\varphi_i, f_i) \mid i = 1, \ldots, n\}$ be a set of PLFC clauses and let (φ, f) be PLFC query.

1. Negate (φ, f) as stated in Definition 4.15; let $(C_1, f_1'), \ldots, (C_m, f_m')$ be the obtained PLFC clauses.
2. $K' \leftarrow K \cup \{C_1, f_1'), \ldots, (C_m, f_m')\}$.
3. Search for a deduction, in the context determined by U and m, of $(\bot, \beta)$, by applying the GR, GF, GM and TR inference rules from K' repeatedly.
4. $K \models^{U,m}_{PLFC} (\varphi, \min(\beta, f))$, where $\beta \in [0, 1]$ can be seen as a proof threshold.

We design the PLFC refutation proof procedure as a function described next:

function *Refutation_Procedure*

input
K : Set of PLFC clauses
U : Collection of non-empty domains
m : Interpretation of object constants over U `/* context` $\mathcal{M}_{U,m}$ `*/`
(φ, f) : PLFC query
β : Necessity degree `/* proof threshold */`

output
entails : boolean

auxiliary variables
K' : Set of PLFC clauses `/* K extended with` $\neg(\varphi, f)$ `*/`
RL : Set of PLFC literals `/* set of resolved literals */`

begin
$K' := K \cup$ *Negation*$((\varphi, f))$;
$K' :=$ *Fusion*(K');
$K' :=$ *Threshold*(K', U, m, β);
$K' :=$ *Equivalent*(K');
$K' :=$ *Transformation*(K');
$K' :=$ *Merging*(K');
$RL := \emptyset$;
entails := *Proof_Procedure*(K', U, m, β, RL);

return(*entails* **)**
end function *Refutation_Procedure*

Function *Negation* returns the set of clauses obtained by negating the query (φ, f) following patterns of Definition 4.15; function *Fusion* applies the generalized fusion rule GF to those clauses of the knowledge base such that some variables appear in the valuation side but not in the base formula; function *Threshold* eliminates from the knowledge base the clauses such that the valuation function cannot be evaluated to a value $\delta \geq \beta$; function *Equivalent* transforms, following Propositions 4.1 and 4.2, each fuzzy constant of a base formula of the knowledge base into a imprecise but non-fuzzy constant; function *Transformation* transforms all precise constants present in base formulas of the knowledge base into variable weights; function *Merging* applies the generalized merging rule GM over the knowledge base; and finally, function *Proof_Procedure* verifies, by successively applying the GR, GF, GM and TR inference rules, if the necessity of obtaining the contradiction $\perp$ from the knowledge base is $\geq \beta$. Hence, if the function call

$$\textit{Refutation\_Procedure}(K,\ U,\ m,\ (\varphi, f),\ \beta)$$

returns *true*, we have that the necessity of the query (φ, f) being true, given the knowledge in K and the context $\mathcal{M}_{U,m}$, is $\geq \beta$, i.e.

$$K \models^{U,m}_{PLFC} (\varphi, \min(\beta, f)).$$

Notice that the *Refutation_Procedure* function can be easily adapted for providing the highest necessity degree of the query being true given the knowledge in K.

As we have already pointed out (see Example 4.7), during the refutation process the GM inference rule must be applied after each resolution step. Therefore, the *Proof_Procedure* function cannot be oriented to a resolvent clause, and thus, the search space consists of all possible orderings of the literals in the knowledge base. Next we describe the *Proof_Procedure* function which, for each resolution step, is based on chronological backtracking and the search strategy is depth-first.

function *Proof_Procedure*

input
K : Set of PLFC clauses
`/* knowledge base extended with the negation of the query */`
U : Collection of non-empty domains
m : Interpretation of object constants over U `/* context` $\mathcal{M}_{U,m}$ `*/`
β : Necessity degree `/* proof threshold */`
RL : Set of PLFC literals `/* set of resolved literals */`

output
derives : boolean

```
auxiliary variables
  C1, C2, C : PLFC clause
  L1, L2 : PLFC literal
  θ : mgs
  RL' : Set of PLFC literals
  K' : Set of PLFC clauses
begin
  if ( (⊥, δ) ∈ K and δ ≥ β ) then derives : = true;
  else
    for ( each clause C1 ∈ K ) do
      for ( each literal L1 ∈ C1 such that L1 ∉ RL ) do
 /*   assume that C1 has the general form (L1 ∨ φ, f1),
      where f1 is a valid valuation function   */
        for ( each clause C2 ∈ K ) do
          for ( each literal L2 ∈ C2 ) do
 /*   assume that C2 has the general form (L2 ∨ ψ, f2),
      where f2 is a valid valuation function   */
            if ( θ = mgs(L1, L2, U, m) and θ ≠ fail ) then
              C := ((φ ∨ ψ)θ, min(f1, f2)θ);
              C := Fusion_r(C);
              if ( Threshold_r(C, U, m, β) ) then
 /*   even some variables of C have not been instantiated,
      function Threshold_r returns false as soon as the
      valuation function of C cannot be evaluated, in the
      context M_{U,m}, to a necessity degree ≥ β   */
                C := Equivalent_r(C);
                C := Transformation_r(C);
                K' := Merging_r(K, C);
                RL' := RL ∪ {L1};
                if ( Proof_Procedure(K', U, m, β, RL') ) then
                  derives : = true;
                end if
              end if
            end if
          end for
        end for
      end for
    derives : = false;
  end if
  return( derives )
end function Proof_Procedure
```

Function *Fusion_r* applies, if necessary, the GF inference rule to the resolvent clause of C_1 and C_2; function *Threshold_r* determines if the valuation function of the resolvent clause can be evaluated to a value $\geq \beta$; function *Equivalent_r*

transforms, following Propositions 4.1 and 4.2, each fuzzy constant that should be present in the base formula of the resolvent clause into an imprecise but non-fuzzy constant; function *Transformation_r* transforms each precise constant of the base formula of the resolvent clause into a variable weight; and finally, function *Merging_r* applies the GM inference rule over the knowledge base and the resolvent clause. Hence, if the function call

$$Proof\_Procedure(K,\ U,\ m,\ \beta,\ \emptyset)$$

returns *true*, we have that $(\bot, \delta)$, with $\delta \geq \beta$, can be derived from K, in the context determined by U and m, by successively applying the GR, GF, TR and GM inference rules.

Proposition 4.4 *Given a particular context $\mathcal{M}_{U,m}$, the notion of proof in PLFC by refutation using the GR, GF, TR and GM inference rules, written $\vdash^{U,m}_{PLFC}$, is sound wrt the PLFC semantics, that is, for each set of PLFC clauses K, a PLFC query (φ, f) and a threshold β*

$$\text{if } K \vdash^{U,m}_{PLFC} (\varphi, \min(f, \beta)), \text{ then } K \models^{U,m}_{PLFC} (\varphi, \min(f, \beta)).$$

Proof: Direct consequence of the soundness of refutation in PLFC and the soundness of the GR, GF, TR and GM inference rules. ■

Example 4.9 PLFC can be used for temporal reasoning. Let us consider a rule for the diagnosis of a disorder called *Brucellosis* described by Godo and Vila (1995) and already used in the context of PLFC by Sandri and Godo (1999). To facilitate comprehension, the following abbreviations are used:

B_I	for	Begin of the Inoculation
E_I	for	End of the Inoculation
B_IP	for	Begin of the Initial Period
E_IP	for	End of the Initial Period
B_OFP	for	Begin Undulating Fever Period
E_OFP	for	End of the Undulating Fever Period
B_IA	for	Begin of the Intervertebral Affection
E_IA	for	End of the Intervertebral Affection

We represent in PLFC the rule diagnosing *Brucellosis* as:

$$\begin{aligned}\mathbf{r:}\quad & (\neg FuzzDist(B\_I, B\_IP, z_1) \quad \vee \\ & \neg FuzzDist(B\_IP, E\_IP, z_2) \quad \vee \\ & \neg FuzzDist(B\_IP, B\_OFP, z_3) \quad \vee \\ & \neg FuzzDist(B\_OFP, E\_OFP, z_4) \quad \vee \\ & \neg FuzzDist(B\_OFP, B\_IA, z_5) \quad \vee \\ & \neg FuzzDist(B\_IA, E\_IA, z_6) \quad \vee \\ & Brucellosis,\ \min(0.9, f(z_1, z_2, z_3, z_4, z_5, z_6)))\end{aligned}$$

where $FuzzDist(\cdot, \cdot, \cdot)$ is a predicate expressing the "fuzzy temporal distance"

between two events, z_i, for $i = 1, \ldots, 6$, are variables of sort `days`, and

$$\begin{aligned} f(z_1, z_2, z_3, z_4, z_5, z_6) \;=\; & \min(less\_month(z_1), less\_half\_month(z_2), \\ & some\_period(z_3), between\_week\_two\_month(z_4), \\ & less\_two\_years(z_5), between\_month\_year(z_6)). \end{aligned}$$

Let us now suppose that we have a patient with the following data: "It is almost certain that the initial period of his disease began about 20-25 days after the inoculation and it lasted 10 days. The patient remembers he went through an undulating fever period of approximately one month, starting more or less one month after the initial period. Finally, the patient began to suffer from an intervertebral pain approximately one year and a half after the fever began and he is sure it went on for almost 1 year".

This data can be represented in PLFC as follows:

s1: $(FuzzDist(B\_I, B\_IP, about\_20-25\_days), 0.9)$

s2: $(FuzzDist(B\_IP, E\_IP, about\_10\_days), 1)$

s3: $(FuzzDist(B\_OFP, E\_OFP, approx\_one\_month), 0.8)$

s4: $(FuzzDist(B\_IP, B\_OFP, more\_or\_less\_one\_month), 0.7)$

s5: $(FuzzDist(B\_OFP, B\_IA, approx\_one\_year\_and\_a\_half), 0.7)$

s6: $(FuzzDist(B\_IA, E\_IA, almost\_one\_year), 1)$

Now, taking the knowledge base $K = \{\mathbf{r}, \mathbf{s1}, \mathbf{s2}, \mathbf{s3}, \mathbf{s4}, \mathbf{s5}, \mathbf{s6}\}$, the query $(Brucellosis, 1)$, and the context

1. $U = \{U_{\texttt{days}} = [0, 1080]\}$;

2. $m(less\_month) = [1, 30]$,
   $m(less\_half\_month) = [1, 15]$,
   $m(some\_period) = [1, 1080]$,
   $m(between\_week\_two\_month) = [7, 60]$,
   $m(less\_two\_years) = [1, 660]$,
   $m(between\_month\_year) = [30, 360]$,
   $m(about\_20-25) = [17; 20; 25; 28]$,
   $m(about\_10) = [8; 9; 11; 12]$,
   $m(approx\_one\_month) = [20; 25; 35; 40]$,
   $m(more\_or\_less\_one\_month) = [12; 25; 50; 63]$,
   $m(approx\_one\_year\_and\_a\_half) = [500; 520; 560; 580]$, and
   $m(almost\_one\_year) = [325; 340; 350; 365]$.

The *Refutation_Procedure* function return *true* for each threshold $\beta \leq 0.7$. This means that from K and $(\neg Brucellosis, 1)$, in the context determined by U

and m, the *Proof_Procedure* function derives $(\bot, 0.7)$, where 0.7 stands from $\min(0.9, 0.9, 1, 0.7, 0.8, 0.7, 1, \delta)$ with

$$\begin{aligned}\delta &= \min(N([1,30] \mid [17;20;25;28]_{0.9}), N([1,15] \mid [8;9;11;12]_1),\\ &\quad N([1,1080] \mid [12;25;50;63]_{0.7}), N([7,60] \mid [20;25;35;40]_{0.8}),\\ &\quad N([1,660] \mid [500;520;560;580]_{0.7}), N([30,360] \mid [325;340;350;365]_1))\\ &= 1.\end{aligned}$$

We then have proved that *Brucellosis* explains the patient's symptoms with a necessity of at least 0.7. □

Chapter 5

A fuzzy possibilistic logic based on Gödel infinitely-valued logic

5.1 Introduction

PLFC provides a powerful framework for reasoning under possibilistic uncertainty and representing disjunctive and conjunctive vague knowledge (see Section 3.3), but due to variable weights, has some computational limitations. On the one hand, the current proof method for PLFC (refutation by resolution) is not complete (see Section 4.4), and does not allow us to define an efficient proof procedure (see Section 4.5). Indeed, during the proof process, in order to get higher unification degrees, a merging rule must be applied after each resolution step, and thus, the proof procedure cannot be oriented to a resolvent clause and the search space consists of all possible orderings of the literals in the knowledge base. On the other hand, since unification (in the classical sense) between fuzzy constants is not allowed and it is performed through variable weights, clauses with variable weights are strictly necessary in PLFC. And finally, in case we transform PLFC clauses into a Horn-rule syntax-style, the interpretation of object constants depends on whether they appear in the antecedent or in the consequent of a rule. For instance, the PLFC clauses

s1: $(\neg q \vee p(B), 1)$

s2: $(\neg p(B) \vee q, 1)$

which, if B is an imprecise but non-fuzzy constant, are interpreted as

s1: "$\neg q \vee [\exists\, x \in B,\ p(x)]$"

s2: "$[\exists\, x \in B,\ \neg p(x)] \vee q$"

have to be represented in a Horn-rule syntax-style as

s1': $(q \rightarrow p(B), 1)$

s2': $(p(B) \rightarrow q, 1)$

and thus, they are interpreted as:

s1': "$q \rightarrow [\exists\, x \in B,\ p(x)]$"

s2': "$[\forall\, x \in B,\ p(x)] \rightarrow q$"

Therefore, in this case, (fuzzy) constants are interpreted as conjunctive information if they appear in the antecedent of a Horn-rule, and as disjunctive information, otherwise.

Due to these limitations, we turn our attention to a possibilistic language based on definite clauses with fuzzy constants expressing (disjunctive) vague knowledge. Within this restricted framework our aim is to define a logic programming language for reasoning under possibilistic uncertainty and representing vague knowledge, but as in classical logic programming systems, this language enables us to define an efficient proof procedure based on a complete calculus and oriented to goals.

In this chapter, we first define a general fuzzy possibilistic logic based on the propositional Gödel infinitely-valued logic (called PGL). Then, we focus our attention on the possibilistic logic programming language with fuzzy propositional variables that results from considering the Horn-rule fragment of PGL. In the next chapter, we extend this fragment of PGL with both fuzzy constants and a fuzzy unification mechanism based on a necessity-like measure which preserves completeness for a particular class of formulas. In our opinion, this is a key feature that justifies by itself the interest of such a logic programming system.

The chapter is organized as follows. In Section 5.2, we recall the syntax and the many-valued semantics of the underlying fuzzy logic, i.e. the propositional Gödel infinitely-valued logic. In Section 5.3, we extend the language of propositional Gödel logic to allow possibilistic reasoning. In Section 5.4 we describe the uncertainty sublogic that our proof method can deal with. Finally, in Section 5.5 we prove that the proof method is complete for determining the maximum degree of possibilistic entailment of fuzzy propositional variables.

5.2 The underlying fuzzy logic: Gödel logic

Our aim in this chapter is to define a propositional logic programming language for reasoning under possibilistic uncertainty and vague knowledge. Moreover, such a language must enable us to define an efficient proof method, based on a complete calculus, for determining the maximum degree of possibilistic belief with which a fuzzy propositional variable can be entailed from a knowledge base.

On the one hand, fuzzy propositional variables provide us a suitable representation model in situations in which there is vague, incomplete or imprecise information about the real world. For instance, the fuzzy statement "Peter is *about_35* years old" can be represented by a fuzzy propositional variable *Peter_is_about_35*. Thus, if *about_35* denotes a crisp interval of ages, say $[34, 36]$, the fuzzy propositional variable *Peter_is_about_35* is interpreted in possibilistic terms as

$$\text{"}\exists\, x \in [34, 36] \text{ such that Peter is } x \text{ years old."}$$

So, as fuzzy constants in PLFC, fuzzy propositional variables can be seen as (flexible) restrictions on an existential quantifier.

On the other hand, since we want to deal with fuzzy propositional variables in the language, the truth evaluation of formulas must be many-valued rather than Boolean, and thus, again as in PLFC, the possibilistic logic programming language has to be based on a many-valued logic. We take Gödel logic us underlying fuzzy logic.

The reason for choosing Gödel logic as the underlying many-valued logic to model fuzziness is two-fold: first, truth-functions of Gödel logic are purely ordinal; that is, they are definable just from the ordering of the truth scale (see Section 2.2), no further algebraic operations are required, and thus, the use of this logic is in accordance with the simplest understanding, in terms of an ordering, of what a fuzzy, gradual property can be; and second, and non-negligible, we show that Gödel logic is fully compatible with an already proposed and suitable extension of necessity measures for fuzzy events, in the sense that Gödel logic allows us to define a well-behaved and well-featured possibilistic semantics on top of it.

Next we briefly recall the syntax and the many-valued semantics of propositional Gödel fuzzy logic (see Section 2.2 for an extended overview).

The *language* of propositional Gödel fuzzy logic is built in the usual way from

- a (countable) set of (fuzzy) *propositional variables*, for instance, *Peter_is_about_35*;
- *connectives* $\wedge$ and $\rightarrow$; and
- the *truth* constant $\overline{0}$.

The *semantics* of propositional Gödel fuzzy logic is given by *interpretations* $\mathbf{I}$ of propositional variables into the unit interval $[0, 1]$ which are extended to arbitrary formulas by means of the following rules:

$$\begin{array}{lcl} \mathbf{I}(\overline{0}) & = & 0, \\ \mathbf{I}(\varphi \wedge \psi) & = & \min(\mathbf{I}(\varphi), \mathbf{I}(\psi)), \\ \mathbf{I}(\varphi \rightarrow \psi) & = & \begin{cases} 1, & \text{if } \mathbf{I}(\varphi) \leq \mathbf{I}(\psi) \\ \mathbf{I}(\psi), & \text{otherwise.} \end{cases} \end{array}$$

5.3 Possibilistic reasoning over Gödel logic: PGL

We have seen that fuzzy propositional variables are suitable for representing vague information as in the statement "Peter is *about_35* years old". Now, we are interested in extending the fuzzy propositional language to allow fuzzy reasoning under possibilistic uncertainty which leads us to a more expressive language. For instance, the statement

"it is almost certain that Peter is *about_35* years old"

is represented in this setting by a *certainty-weighted fuzzy proposition*

$$(Peter\_is\_about\_35, 0.9),$$

where the certainty value 0.9 expresses how much the fuzzy statement "Peter is *about_35* years old" is believed in terms of necessity measures.

As in PLFC, certainty weights are employed to model statements of the form

"φ is alpha-certain",

where φ represents vague, incomplete or imprecise knowledge about the real world. In this framework, this is formalized as

"φ is certain with a necessity of at least α"

and is represented by a certainty-weighted Gödel logic formula (φ, α). We refer to Gödel fuzzy logic extended with certainty-weights as possibilistic Gödel logic, denoted hereafter as PGL.

Definition 5.1 (PGL formula) *A PGL formula is a pair of the form (φ, α), where φ is a Gödel logic formula and $\alpha \in [0,1]$ is a lower bound on the belief on φ in terms of necessity measures.*

Within the standard possibilistic model of uncertainty, belief states are modeled by normalized possibility distributions on a set of Boolean interpretations. However, as in PLFC, the truth evaluation of a Gödel logic formula φ in each interpretation $\mathbf{I}$ is a value $\mathbf{I}(\varphi) \in [0,1]$, and thus, each formula does not induce a crisp set of interpretations, but a fuzzy set of interpretations $[\varphi]$, defining $\mu_{[\varphi]}(\mathbf{I}) = \mathbf{I}(\varphi)$, for each interpretation $\mathbf{I}$. Therefore, in this setting, possibilistic models are normalized possibility distributions on the set $\mathcal{I}$ of all possible interpretations $\mathbf{I}$ of propositional variables into the unit interval $[0,1]$. Then, to measure the uncertainty induced on a Gödel logic formula φ by a possibilistic model $\pi : \mathcal{I} \to [0,1]$, we have to consider some extension of the notion of necessity measure for fuzzy sets.

In order to provide PLFC with a formal possibilistic semantics, in Section 4.2, we considered two different extensions of the standard notion of necessity measure. Namely,

(i) the necessity measure for fuzzy sets proposed by Dubois and Prade (1991c), i.e.

$$N^*([\varphi] \mid \pi) = \inf_{\mathbf{I} \in \mathcal{I}} \pi(\mathbf{I}) \Rightarrow \mu_{[\varphi]}(\mathbf{I}),$$

where $\Rightarrow$ is the reciprocal of Gödel's many-valued implication defined as $x \Rightarrow y = 1$ if $x \leq y$, and $x \Rightarrow y = 1 - x$, otherwise; and

(ii) the necessity measure commonly used in Possibility Theory as a measure of pattern matching of fuzzy events (Dubois and Prade, 1998a), i.e.

$$N([\varphi] \mid \pi) = \inf_{\mathbf{I} \in \mathcal{I}} \max(1 - \pi(\mathbf{I}), \mu_{[\varphi]}(\mathbf{I})).$$

In Section 4.2, we showed that N^* is not a good candidate semantics[1] for PLFC. However, it allows us to define a well-behaved and well-featured possibilistic semantics on top of Gödel logic, in the sense that it allows us to define a complete modus ponens-style calculus for the Horn-rule fragment of Gödel logic (see Section 5.5).

Notice the difference of N^* with respect the necessity measure N adopted for PLFC. For example, if *about_35* denotes a fuzzy interval with a membership function $\mu_{about\_35} : [0, 120] \to [0, 1]$, the formula

$$(Peter\_is\_about\_35, 0.9)$$

is to be interpreted in PGL as "$\exists\, x \in [\mu_{about\_35}]_\beta$ such that Peter is x years old" is certain with necessity of at least $\min(0.9, 1 - \beta)$, for each $\beta \in [0, 1]$, where $[\mu_{about\_35}]_\beta$ denotes the crisp interval of ages associated with the β-cut of the fuzzy set μ_{about_35}.

It is not difficult to see that the necessity measure N^* on fuzzy sets is characterized by the following set of axioms. Namely, let Ω be an arbitrary set and let $N^* : [0, 1]^\Omega \to [0, 1]$ be a measure on the set of fuzzy sets of Ω. Consider the following postulates:

N*1 $N^*(\Omega) = 1$

N*2 $N^*(\emptyset) = 0$

N*3 $N^*(A \cap B) = \min(N^*(A), N^*(B))$
($N^*(\cap_{i \in I} A_i) = \inf_{i \in I} N^*(A_i)$)

N*4 if A is crisp, then $N^*(A \cup \alpha) = \begin{cases} 1, & \text{if } 1 - \alpha \leq N^*(A) \\ N^*(A), & \text{otherwise} \end{cases}$

where $\mu_{A \cap B}(w) = \min(\mu_A(w), \mu_B(w))$, for each $w \in \Omega$, and $A \cup \alpha$ denotes the fuzzy set defined by the membership function $\mu_{A \cup \alpha}(w) = \max(\alpha, \mu_A(w))$, for each $w \in \Omega$.

Indeed, the characterization of the necessity measure N^* differs from the characterization of N just in axiom **N*4** (see Section 4.2), which for N is

$$N(A \cup \alpha) = \max(\alpha, N(A)).$$

[1] Recall that the possibilistic semantics for PLFC is based on N.

Theorem 5.1 *If N^* satisfies the above postulates, then there exists $\pi : \Omega \to [0,1]$ such that, for all fuzzy subset A of Ω,*

$$N^*(A) = \inf_{w \in \Omega} \pi(w) \Rightarrow \mu_A(w),$$

where $\Rightarrow$ is the reciprocal of Gödel's many-valued implication.

Proof: Let A be an arbitrary fuzzy subset of Ω. Define $\pi(w) = 1 - N^*(\overline{\{w\}})$. Since A can be put as $A = \cap_{w \in \Omega}\{\overline{\{w\}} \cup \mu_A(w)\}$, by N*3, we have that

$$N^*(A) = \inf_{w \in \Omega} N^*(\overline{\{w\}} \cup \mu_A(w)),$$

but by N*4,

$$N^*(\overline{\{w\}} \cup \mu_A(w)) = \begin{cases} 1, & \text{if } 1 - \mu_A(w) \leq N^*(\overline{\{w\}}) \\ N^*(\overline{\{w\}}), & \text{otherwise.} \end{cases}$$

But, $N^*(\overline{\{w\}}) = 1 - \pi(w)$, then

$$N^*(\overline{\{w\}} \cup \mu_A(w)) = \begin{cases} 1, & \text{if } \pi(w) \leq \mu_A(w) \\ 1 - \pi(w), & \text{otherwise} \end{cases}$$

and thus, $N^*(\overline{\{w\}} \cup \mu_A(w)) = \pi(w) \Rightarrow \mu_A(w)$. Hence, $N^*(A) = \inf_{w \in \Omega} \pi(w) \Rightarrow \mu_A(w)$. ■

Now let us go into formal definitions.

Definition 5.2 (possibilistic model) *Let $\mathcal{I}$ be the set of (many-valued) Gödel interpretations over a given (countable) set of fuzzy propositional variables. A possibilistic model is a normalized possibility distribution $\pi : \mathcal{I} \to [0,1]$ on the set of interpretations $\mathcal{I}$.*

Definition 5.3 (necessity evaluation) *Let $\pi : \mathcal{I} \to [0,1]$ be a possibilistic model. The necessity evaluation of a Gödel logic formula φ given by π is defined as:*

$$N^*([\varphi] \mid \pi) = \inf_{\mathbf{I} \in \mathcal{I}} \pi(\mathbf{I}) \Rightarrow \mu_{[\varphi]}(\mathbf{I}),$$

where $\mu_{[\varphi]}(\mathbf{I}) = \mathbf{I}(\varphi)$ and $\Rightarrow$ is the reciprocal of Gödel's many-valued implication.

Definition 5.4 (possibilistic satisfiability) *A possibilistic model $\pi : \mathcal{I} \to [0,1]$ satisfies a PGL formula (φ, α), written $\pi \models_{PGL} (\varphi, \alpha)$, iff $N^*([\varphi] \mid \pi) \geq \alpha$. Thus,*

$$\pi \models_{PGL} (\varphi, \alpha) \text{ iff } \pi(\mathbf{I}) \leq \max(1 - \alpha, \mu_{[\varphi]}(\mathbf{I}))$$

for each interpretation $\mathbf{I} \in \mathcal{I}$.

Definition 5.5 (possibilistic entailment) *Let K be a set of PGL formulas. We say that K entails another PGL formula (φ, α), written $K \models_{PGL} (\varphi, \alpha)$, iff every possibilistic model $\pi : \mathcal{I} \to [0,1]$ satisfying all the formulas in K also satisfies (φ, α).*

We refer to a possibilistic model π satisfying all the PGL formulas in a set K as a model of K.

We propose now a Hilbert-style axiomatization[2] of possibilistic Gödel logic. Axioms of PGL are:

- *axioms of Gödel logic* weighted by 1 (see Section 2.2) plus
- the *triviality axiom* $(\varphi, 0)$; and

PGL inference (deduction) rules are:

- *generalized modus ponens:*

$$\frac{\begin{array}{l}(\varphi \to \psi, \alpha)\\ (\varphi, \beta)\end{array}}{(\psi, \min(\alpha, \beta))}.$$

- *weakening:*

$$\frac{(\varphi, \alpha)}{(\varphi, \beta)} \quad \text{if } \beta \leq \alpha.$$

Then, the notion of proof in PGL is as usual (deduction relative to PGL axioms and rules) and it is denoted as $\vdash_{PGL}$. The soundness of this axiomatic system is given in the next theorem.

Theorem 5.2 (soundness of PGL) *PGL is sound with respect to the possibilistic entailment. Thus,*

$$\textit{if } K \vdash_{PGL} (\varphi, \alpha) \textit{ then } K \models_{PGL} (\varphi, \alpha).$$

Proof: Soundness of the axioms and the weakening rule is straightforward, and thus, we only prove the soundness of the modus ponens rule. This reduces to check, for each possibilistic model π on $\mathcal{I}$, that if $N^*([\varphi \to \psi] \mid \pi) \geq \alpha$ and $N^*([\varphi] \mid \pi) \geq \beta$, then $N^*([\psi] \mid \pi) \geq \min(\alpha, \beta)$.

The two conditions amount to, for each interpretation $\mathbf{I} \in \mathcal{I}$, $\pi(\mathbf{I}) \Rightarrow \mathbf{I}(\varphi \to \psi) \geq \alpha$ and $\pi(\mathbf{I}) \Rightarrow \mathbf{I}(\varphi) \geq \beta$. Thus, $\pi(\mathbf{I}) \Rightarrow \min(\mathbf{I}(\varphi \to \psi), \mathbf{I}(\varphi)) \geq \min(\alpha, \beta)$. But, by Gödel semantics, $\min(\mathbf{I}(\varphi \to \psi), \mathbf{I}(\varphi)) = \min(\mathbf{I}(\psi), \mathbf{I}(\varphi)) \leq \mathbf{I}(\psi)$. Therefore, $\pi(\mathbf{I}) \Rightarrow \mathbf{I}(\psi) \geq \min(\alpha, \beta)$, for each interpretation $\mathbf{I} \in \mathcal{I}$, and thus, $N^*([\psi] \mid \pi) \geq \min(\alpha, \beta)$. ■

Unfortunately we have not been able so far to prove whether PGL is complete or incomplete. So it remains as an open question to be solved in the near future.

[2]Notice the analogy with the classical possibilistic logic (Dubois et al., 1994c). Indeed, here we have just replaced the axioms of classical propositional logic with those of Gödel fuzzy logic.

5.4 A possibilistic logic programming language

In the previous section, we have defined PGL, a general possibilistic logic over Gödel fuzzy logic. Our aim in this section is, as in classical propositional logic programming systems, to define a sublanguage for logic programming which enables us to design an efficient proof algorithm, based on a complete calculus for computing the maximum degree of possibilistic entailment of a fuzzy propositional variable, called *goal*, from a set of formulas.

To this end, we restrict ourselves to a *Horn-rule sublanguage* of Gödel fuzzy logic, i.e. to formulas of the form:

$$p_1 \wedge \cdots \wedge p_k \rightarrow q$$

with $k \geq 0$, where $p_1, \ldots, p_k, q$ are (fuzzy) propositional variables, in the traditional logic programming style. As usual, we refer to the conclusion q and the set of premises $p_1, \ldots, p_k$ as the *head* and the *body*, respectively. We distinguish between two types of formulas in this sublanguage: *facts* when $k = 0$ (empty body) and are simply written q, and *rules*, otherwise. Finally, we define a PGL *clause* as a PGL formula (φ, α) such that φ is either a fact or a rule.

For PGL clauses we develop a simple and efficient calculus which does not need the whole logical apparatus of the general possibilistic logic PGL of the previous section. But before, we need to introduce some extra definitions and results.

Definition 5.6 (maximum degree of possibilistic entailment) *The maximum degree of possibilistic entailment of a goal q from a set of PGL clauses P, denoted as $\|q\|_P$, is the greatest lower bound $\alpha \in [0,1]$ on the belief on q such that $P \models_{PGL} (q, \alpha)$. Thus, $\|q\|_P = \sup\{\alpha \in [0,1] \mid P \models_{PGL} (q, \alpha)\}$.*

Proposition 5.1 *The maximum degree of possibilistic entailment of a goal q from a set of PGL clauses P is the least necessity evaluation of q given by the models of P. Thus, $\|q\|_P = \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} P\}$.*

Proof: We define $\alpha_1 = \sup\{\alpha \in [0,1] \mid P \models_{PGL} (q, \alpha)\}$ and $\alpha_2 = \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} P\}$.

- $\alpha_2 \geq \alpha_1$: As $\alpha_1 = \sup\{\alpha \in [0,1] \mid P \models_{PGL} (q, \alpha)\}$, we have that $P \models_{PGL} (q, \alpha)$ for each $\alpha < \alpha_1$. Then, for each model π of P, $N^*([q] \mid \pi) \geq \alpha$ for each $\alpha < \alpha_1$. Thus, $\alpha_2 = \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} P\} \geq \alpha$ for each $\alpha < \alpha_1$. Hence, $\alpha_2 \geq \alpha_1$.

- $\alpha_2 \leq \alpha_1$: As $\alpha_2 = \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} P\}$, we have that $N^*([q] \mid \pi) \geq \alpha_2$ for each model π of P, that is, $P \models_{PGL} (q, \alpha_2)$, and thus, $\alpha_2 \leq \sup\{\alpha \in [0,1] \mid P \models_{PGL} (q, \alpha)\} = \alpha_1$.

■

Corollary 5.1 *Let P be a set of PGL clauses and let s be a propositional variable. Then, $P \models_{PGL} (s, \|s\|_P)$.*

Proof: By Proposition 5.1, $\|s\|_P = \inf\{N^*([s] \mid \pi) \mid \pi \models_{PGL} P\}$, and thus, $N^*([s] \mid \pi) \geq \|s\|_P$ for each model π of P. Therefore, $P \models_{PGL} (s, \|s\|_P)$. ■

To provide our possibilistic logic programming language with a complete calculus for determining the maximum degree of possibilistic entailment we only need the triviality axiom and a particular instance of the generalized modus ponens rule introduced in the previous section:

Axiom: $(\varphi, 0)$.

Generalized modus ponens:

$$\frac{(p_1 \wedge \cdots \wedge p_k \rightarrow q, \alpha) \quad (p_1, \beta_1), \ldots, (p_k, \beta_k)}{(q, \min(\alpha, \beta_1, \ldots, \beta_k))} \; [MP].$$

Obviously, the axiom is a valid PGL clause and the MP rule is sound as we have already proved in the previous section.

Definition 5.7 (degree of deduction) *A goal q is deduced with a degree of deduction α from a set of PGL clauses P, denoted as $P \vdash^*_{PGL} (q, \alpha)$, iff there exists a finite sequence of PGL clauses $C_1, \ldots, C_m$ such that $C_m = (q, \alpha)$ and, for each $i \in \{1, \ldots, m\}$, it holds that $C_i \in P$, C_i is an instance of the axiom or C_i is obtained by applying the MP rule to previous clauses in the sequence.*

Next, we define the syntactic counterpart of maximum degree of possibilistic entailment.

Definition 5.8 (maximum degree of deduction) *The maximum degree of deduction of a goal q from a set of PGL clauses P, denoted $|q|_P$, is the greatest $\alpha \in [0, 1]$ such that $P \vdash^*_{PGL} (q, \alpha)$.*

As the only inference rule of our proof method is the generalized modus ponens, within the framework of logic programming in which P is always a finite set of PGL clauses, there exists a finite number of proofs of a goal q from P, and thus, the above definition turns into

$$|q|_P = \max\{\alpha \in [0, 1] \mid P \vdash^*_{PGL} (q, \alpha)\}.$$

5.5 Completeness of the proof method

In this section, we prove that the PGL deduction mechanism, based on the triviality axiom and the generalized modus ponens rule, is complete for determining the maximum degree of possibilistic entailment of a goal q from a set of PGL clauses P. But before, we need to prove some extra results.

Proposition 5.2 *If the only formulas in P with head q are recursive*[3]*, then $\|q\|_P = |q|_P = 0$.*

[3] A recursive formula is of the form $p_1 \wedge \cdots \wedge p_k \wedge q \rightarrow q$ with $k \geq 0$.

Proof: Let $\mathbf{I}_0$ be an interpretation such that $\mathbf{I}_0(q) < 1$ and $\mathbf{I}_0(p) = 1$ for each propositional variable $p \neq q$. Now, let π be a possibility distribution with the following definition:

$$\pi(\mathbf{I}) = \begin{cases} 1, & \text{if } \mathbf{I} = \mathbf{I}_0 \\ 0, & \text{otherwise.} \end{cases}$$

Since the only formulas in P with head q are recursive, we have that $\pi \models_{PGL} P$ and $N^*([q] \mid \pi) = 0$. Therefore, by Proposition 5.1, $\|q\|_P = \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} P\} = 0$. On the one hand, for each propositional variable s, $|s|_P \geq 0$, and thus, $|q|_P \geq 0$ and $|q|_P \geq \|q\|_P$. On the other hand, by the soundness of the modus ponens rule, $\|s\|_P \geq |s|_P$ for each propositional variable s, and thus, $\|q\|_P \geq |q|_P$. Then, $\|q\|_P = |q|_P = 0$. ■

Proposition 5.3 *Let (p, β) and $(p \to q, \gamma)$ be two PGL clauses such that $p \neq q$. Then, $\|q\|_{\{(p,\beta),(p\to q,\gamma)\}} = \min(\beta, \gamma)$.*

Proof:

1. $\|q\|_{\{(p,\beta),(p\to q,\gamma)\}} \geq \min(\beta,\gamma)$: By the soundness of the modus ponens rule, $\{(p,\beta),(p \to q,\gamma)\} \models_{PGL} (q, \min(\beta,\gamma))$, and thus, $\|q\|_{\{(p,\beta),(p\to q,\gamma)\}} = \sup\{\alpha \in [0,1] \mid \{(p,\beta),(p \to q,\gamma)\} \models_{PGL} (q,\alpha)\} \geq \min(\beta,\gamma)$.

2. $\|q\|_{\{(p,\beta),(p\to q,\gamma)\}} \leq \min(\beta,\gamma)$: By Proposition 5.1, $\|q\|_{\{(p,\beta),(p\to q,\gamma)\}} = \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} \{(p,\beta),(p \to q,\gamma)\}\}$. We show that for some model π of $\{(p,\beta),(p \to q,\gamma)\}$, we have that $N^*([q] \mid \pi) = \min(\beta,\gamma)$, and thus, $\|q\|_{\{(p,\beta),(p\to q,\gamma)\}} \leq \min(\beta,\gamma)$. We distinguish two cases.

   Case $\gamma \leq \beta$**.** Let $\mathbf{I}_0$ and $\mathbf{I}_1$ be two interpretations such that $\mathbf{I}_0(p) = 1$ and $\mathbf{I}_0(q) < 1 - \gamma$, and $\mathbf{I}_1(p) = \mathbf{I}_1(q) = 1$. Now, let π be a possibility distribution with the following definition:

$$\pi(\mathbf{I}) = \begin{cases} 1, & \text{if } \mathbf{I} = \mathbf{I}_1 \\ 1-\gamma, & \text{if } \mathbf{I} = \mathbf{I}_0 \\ 0, & \text{otherwise.} \end{cases}$$

   It is easy to check that $\pi \models_{PGL} (p,\beta)$ and $\pi \models_{PGL} (p \to q,\gamma)$ and $N^*([q] \mid \pi) = \gamma = \min(\beta,\gamma)$.

   Case $\beta \leq \gamma$**.** Let us consider a further interpretation $\mathbf{I}_2$ such that $\mathbf{I}_2(p) \leq \mathbf{I}_2(q) < 1 - \beta$. Now, let us define the following possibility distribution:

$$\pi(\mathbf{I}) = \begin{cases} 1, & \text{if } \mathbf{I} = \mathbf{I}_1 \\ 1-\beta, & \text{if } \mathbf{I} = \mathbf{I}_2 \\ 0, & \text{otherwise.} \end{cases}$$

   Again it is easy to check that $\pi \models_{PGL} (p,\beta)$ and $\pi \models_{PGL} (p \to q,\gamma)$ and $N^*([q] \mid \pi) = \beta = \min(\beta,\gamma)$.

■

Proposition 5.4 *Let P be a set of PGL clauses, let γ be a value of the real interval $[0,1]$, and let $\mathbf{I}_0$ and $\mathbf{I}_1$ be two interpretations such that, for each propositional variable s,*

$$\mathbf{I}_0(s) = \begin{cases} 0, & \text{if } \|s\|_P \leq \gamma \\ 1, & \text{if } \|s\|_P > \gamma \end{cases}$$

and $\mathbf{I}_1(s) = 1$. If π is a possibility distribution with the following definition:

$$\pi(\mathbf{I}) = \begin{cases} 1, & \text{if } \mathbf{I} = \mathbf{I}_1 \\ 1-\gamma, & \text{if } \mathbf{I} = \mathbf{I}_0 \\ 0, & \text{otherwise} \end{cases}$$

then $\pi \models_{PGL} P$.

Proof: Suppose that $\pi \not\models_{PGL} P$. If $\pi \not\models_{PGL} P$, it must exist at least a clause of P, say (φ, β), such that $\pi \not\models_{PGL} (\varphi, \beta)$. We distinguish two cases:

Case $(\varphi, \beta) = (q, \beta)$**.** On the one hand, because of the definition of π, if $\pi \not\models_{PGL} (q, \beta)$, it must be that $\pi(\mathbf{I}_0) \Rightarrow \mathbf{I}_0(q) < \beta$, and thus, it must be that $\mathbf{I}_0(q) = 0$ and $\gamma < \beta$. On the other hand, as $(q, \beta) \in P$, we have that $P \models_{PGL} (q, \beta)$ and $\|q\|_P = \sup\{\alpha \in [0,1] \mid P \models_{PGL} (q, \alpha)\} \geq \beta$. Now, as $\beta > \gamma$, we have that $\|q\|_P > \gamma$. Therefore, because of the definition of $\mathbf{I}_0$, we get that $\mathbf{I}_0(q) = 1$, i.e. $\pi \models_{PGL} (q, \beta)$.

Case $(\varphi, \beta) = (r \to q, \beta)$**.** On the one hand, because of the definition of π, if $\pi \not\models_{PGL} (r \to q, \beta)$ it must be that $\pi(\mathbf{I}_0) \Rightarrow \mathbf{I}_0(r \to q) < \beta$, and thus, $\mathbf{I}_0(r) = 1$ and $\mathbf{I}_0(q) = 0$ and $\gamma < \beta$. On the other hand, because of the definition of $\mathbf{I}_0$, if $\mathbf{I}_0(r) = 1$, $\|r\|_P > \gamma$. If we define $\alpha_r = \|r\|_P$, by Corollary 5.1, $P \models_{PGL} (r, \alpha_r)$ and $\alpha_r > \gamma$. Moreover, as $(r \to q, \beta) \in P$, we have that $P \models_{PGL} (r \to q, \beta)$. Therefore, by the soundness of the modus ponens rule, $P \models_{PGL} (q, \min(\alpha_r, \beta))$, and thus, $\|q\|_P = \sup\{\alpha \in [0,1] \mid P \models_{PGL} (q, \alpha)\} \geq \min(\alpha_r, \beta)$. Now, as $\alpha_r > \gamma$ and $\beta > \gamma$, we have that $\|q\|_P \geq \min(\alpha_r, \beta) > \gamma$. Then, because of the definition of $\mathbf{I}_0$, we get that $\mathbf{I}_0(q) = 1$, i.e. $\pi \models_{PGL} (r \to q, \beta)$.

■

Proposition 5.5 *Let P be a set of PGL clauses and let $(r \to q, \gamma)$ be a clause of P. If $\|q\|_P > \|q\|_{P \setminus \{(r \to q, \gamma)\}}$, then $\|q\|_P = \|q\|_{\{(r, \|r\|_P), (r \to q, \gamma)\}}$.*

Proof: We define $\alpha_r = \|r\|_P$.

1. $\|q\|_P \geq \|q\|_{\{(r,\alpha_r),(r \to q,\gamma)\}}$: On the one hand, as $\alpha_r = \|r\|_P$, by Corollary 5.1, $P \models_{PGL} (r, \alpha_r)$. On the other hand, as $(r \to q, \gamma) \in P$, we have that $P \models_{PGL} (r \to q, \gamma)$. Then, $\pi \models_{PGL} \{(r, \alpha_r), (r \to q, \gamma)\}$ for each model π of P. By Proposition 5.1, $\|q\|_P = \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} P\}$ and $\|q\|_{\{(r,\alpha_r),(r \to q,\gamma)\}} = \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} \{(r, \alpha_r), (r \to q, \gamma)\}\}$. Therefore, we get that $\|q\|_P \geq \|q\|_{\{(r,\alpha_r),(r \to q,\gamma)\}}$.

2. $\|q\|_P \leq \|q\|_{\{(r,\alpha_r),(r\to q,\gamma)\}}$: By Proposition 5.3, $\|q\|_{\{(r,\alpha_r),(r\to q,\gamma)\}} = \min(\alpha_r,\gamma)$. Then, we must prove that $\|q\|_P \leq \min(\alpha_r,\gamma)$. To this end, we define $P' = P\backslash\{(r\to q,\gamma)\}$ and we prove first that $\min(\alpha_r,\gamma) > \|q\|_{P'}$.

   (a) $\alpha_r > \|q\|_{P'}$: Suppose that $\alpha_r \leq \|q\|_{P'}$. As $P' = P\backslash\{(r\to q,\gamma)\}$, $\|r\|_P \geq \|r\|_{P'}$ and $\|r\|_{P'} \leq \|q\|_{P'}$. Then, defining $\alpha_{r'} = \|r\|_{P'}$, we have that $\min(\alpha_{r'},\gamma) \leq \|q\|_{P'}$. Now by Proposition 5.3, $\|q\|_{\{(r,\alpha_{r'}),(r\to q,\gamma)\}} = \min(\alpha_{r'},\gamma)$.
   Then, if $\alpha_r \leq \|q\|_{P'}$, we have that $\|q\|_{\{(r,\alpha_{r'}),(r\to q,\gamma)\}} \leq \|q\|_{P'}$, and thus, $\max(\|q\|_{P'}, \|q\|_{\{(r,\alpha_{r'}),(r\to q,\gamma)\}}) \leq \|q\|_{P'}$, i.e. $\|q\|_{P'\cup\{(r\to q,\gamma)\}} \leq \|q\|_{P'}$. Therefore, $\alpha_r > \|q\|_{P'}$.

   (b) $\gamma > \|q\|_{P'}$: If $\|q\|_P > \|q\|_{P'}$, there must exist at least a possibility distribution π such that $\pi \models_{PGL} P'$ and $\pi \not\models_{PGL} (r\to q,\gamma)$. If $\pi \not\models_{PGL} (r\to q,\gamma)$, there must exist at least an interpretation $\mathbf{I}$ such that $\mathbf{I}(r) > \mathbf{I}(q)$ and $\pi(\mathbf{I}) > \mathbf{I}(q)$ and $1-\pi(\mathbf{I}) < \gamma$. Now, by Proposition 5.1, $\|q\|_{P'} = \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} P'\} = \inf\{1-\pi(\mathbf{I}) \mid \pi \models_{PGL} P' \wedge \pi(\mathbf{I}) > \mathbf{I}(q)\}$, and thus, $\|q\|_{P'} < \gamma$.

   Now, let $\mathbf{I}_0$ and $\mathbf{I}_1$ be two interpretations such that, for each propositional variable s,

$$\mathbf{I}_0(s) = \begin{cases} 0, & \text{if } \|s\|_{P'} \leq \min(\alpha_r,\gamma) \\ 1, & \text{if } \|s\|_{P'} > \min(\alpha_r,\gamma) \end{cases}$$

   and $\mathbf{I}_1(s) = 1$. And, let π be a possibility distribution with the following definition:

$$\pi(\mathbf{I}) = \begin{cases} 1, & \text{if } \mathbf{I} = \mathbf{I}_1 \\ 1-\min(\alpha_r,\gamma), & \text{if } \mathbf{I} = \mathbf{I}_0 \\ 0, & \text{otherwise.} \end{cases}$$

   Because of the definition of π and Proposition 5.4, $\pi \models_{PGL} P'$. And, as $\|q\|_{P'} < \min(\alpha_r,\gamma)$, we have that $\mathbf{I}_0(q) = 0$ and $N^*([q] \mid \pi) = \min(\alpha_r,\gamma)$. Finally, if $\gamma \leq \alpha_r$, it is easy to check that $\pi \models_{PGL} (r\to q,\gamma)$ even though $\mathbf{I}_0(r) = 1$. And, if $\alpha_r \leq \gamma$, then $\|r\|_{P'} \leq \alpha_r$ and $\mathbf{I}_0(r) = 0$, and thus, $\pi \models_{PGL} (r\to q,\gamma)$ as well.

   Therefore, $\pi \models_{PGL} P$ and $N^*([q] \mid \pi) = \min(\alpha_r,\gamma)$. And, by Proposition 5.1, $\|q\|_P = \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} P\} \leq \min(\alpha_r,\gamma)$.

■

Proposition 5.6 *Let P be a set of PGL clauses and let $(q\to p,\gamma)$ be a clause of P. It holds that $\|q\|_P = \|q\|_{P\backslash\{(q\to p,\gamma)\}}$.*

Proof: We define $P' = P\backslash\{(q\to p,\gamma)\}$. Then, we must prove that $\|q\|_P = \|q\|_{P'}$.

1. $\|q\|_P \geq \|q\|_{P'}$: By Proposition 5.1, $\|q\|_P = \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} P\}$ and $\|q\|_{P'} = \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} P'\}$. Now, as $P' = P\backslash\{(q\to p,\gamma)\}$, if $\pi \models_{PGL} P$ then $\pi \models_{PGL} P'$, and thus, $\inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} P\} \geq \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} P'\}$.

2. $\|q\|_P \leq \|q\|_{P'}$: We proceed by induction on n, where n is the number of clauses of P'.

   If $n = 1$, then it must be that P' contains only either one certainty-weighted fact or rule. We assume that q occurs in P' as the head of a non recursive formula; otherwise, by Proposition 5.2, $\|q\|_{P'} = \|q\|_{P' \cup \{(q \to p, \gamma)\}} = 0$.

   Suppose that P' contains only the the certainty-weighted fact (q, β). Then, $\|q\|_{P'} = \sup\{\alpha \in [0,1] \mid P' \models_{PGL} (q, \alpha)\} \geq \beta$. Now, let $\mathbf{I}_0$ and $\mathbf{I}_1$ be two interpretations such that $\mathbf{I}_0(q) < 1-\beta$ and $\mathbf{I}_0(p) = 1$ for each propositional variable $p \neq q$, and $\mathbf{I}_1(s) = 1$ for each propositional variable s. And, let π be a possibility distribution with the following definition:

$$\pi(\mathbf{I}) = \begin{cases} 1, & \text{if } \mathbf{I} = \mathbf{I}_1 \\ 1-\beta, & \text{if } \mathbf{I} = \mathbf{I}_0 \\ 0, & \text{otherwise.} \end{cases}$$

   It is easy to check that $\pi \models_{PGL} (q, \beta)$ and $\pi \models_{PGL} (q \to p, \gamma)$ and $N^*([q] \mid \pi) = \beta$. Then, by Proposition 5.1, $\|q\|_P = \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} \{(q, \beta), (q \to p, \gamma)\}\} \leq \beta$, and thus, $\|q\|_P \leq \|q\|_{P'}$.

   Suppose that P' contains only the the certainty-weighted rule $(r \to q, \beta)$. Let $\mathbf{I}_0$ be an interpretation such that $\mathbf{I}_0(r) \leq \mathbf{I}_0(q) < 1$ and $\mathbf{I}_0(p) = 1$. And, let π be a possibility distribution with the following definition:

$$\pi(\mathbf{I}) = \begin{cases} 1, & \text{if } \mathbf{I} = \mathbf{I}_0 \\ 0, & \text{otherwise.} \end{cases}$$

   It is easy to check that $\pi \models_{PGL} (r \to q, \beta)$ and $\pi \models_{PGL} (q \to p, \gamma)$ and $N^*([q] \mid \pi) = 0$. Then, by Proposition 5.1, $\|q\|_P = \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} \{(r \to q, \beta), (q \to p, \gamma)\}\} = 0$ and $\|q\|_{P'} = \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} (r \to q, \beta)\} = 0$, and thus, $\|q\|_P = \|q\|_{P'} = 0$.

   Suppose now that for each set P'' that contains n clauses it holds that $\|r\|_{P''} \geq \|r\|_{P'' \cup \{(r \to s, \delta)\}}$ for each propositional variable r, and suppose that the program P' contains $n+1$ clauses. Then, we prove that $\|q\|_{P'} \geq \|q\|_P$, where $P = P' \cup \{(q \to p, \gamma)\}$.

   Since we are assuming that q occurs in P' as the head of a non recursive formula, let (φ, β) be a non recursive clause of P' such that q is the head of φ and $\beta \geq \delta$, for each non recursive clause (ψ, δ) of P' such that q is the head of ψ. We distinguish two cases:

   Case $(\varphi, \beta) = (q, \beta)$**.** As $(q, \beta) \in P'$, we have that $\|q\|_{P'} = \sup\{\alpha \in [0,1] \mid P' \models_{PGL} (q, \alpha)\} \geq \beta$. Now, let $\mathbf{I}_0$ and $\mathbf{I}_1$ be two interpretations such that $\mathbf{I}_0(q) < 1-\beta$, $\mathbf{I}_0(p) = 1$ for each propositional variable $p \neq q$, and $\mathbf{I}_1(s) = 1$ for each propositional variable s. And, let π be a possibility distribution with the following definition:

$$\pi(\mathbf{I}) = \begin{cases} 1, & \text{if } \mathbf{I} = \mathbf{I}_1 \\ 1-\beta, & \text{if } \mathbf{I} = \mathbf{I}_0 \\ 0, & \text{otherwise.} \end{cases}$$

On the one hand, we know that $\beta \geq \delta$ for each non recursive clause (ψ, δ) of P' such that q is the head of ψ, and thus, $\pi \models_{PGL} P'$. On the other hand, it is easy to check that $\pi \models_{PGL} (q \rightarrow p, \gamma)$ and $N^*([q] \mid \pi) = \beta$. Then, $\pi \models_{PGL} P$ and, by Proposition 5.1, we have that $\|q\|_P = \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} P\} \leq \beta$, and thus, $\|q\|_P \leq \|q\|_{P'}$.

Case $(\varphi, \beta) = (r \rightarrow q, \beta)$. For each program P' it follows that $\|q\|_{P'} \geq \|q\|_{P' \setminus \{(r \rightarrow q, \beta)\}}$. Now,

- If $\|q\|_{P'} = \|q\|_{P' \setminus \{(r \rightarrow q, \beta)\}}$, by the induction hypothesis, $\|q\|_{P'} \geq \|q\|_{P' \cup \{(q \rightarrow p, \gamma)}$.
- If $\|q\|_{P'} > \|q\|_{P' \setminus \{(r \rightarrow q, \beta)\}}$, by Proposition 5.5, $\|q\|_{P'} = \|q\|_{\{(r, \alpha_r), (r \rightarrow q, \beta)\}}$, where $\alpha_r = \|r\|_{P'}$, and, by Proposition 5.3, $\|q\|_{\{(r, \alpha_r), (r \rightarrow q, \beta)\}} = \min(\alpha_r, \beta)$. Now, let $\mathbf{I}_0$ and $\mathbf{I}_1$ be two interpretations such that, for each propositional variable s,

$$\mathbf{I}_0(s) = \begin{cases} 0, & \text{if } \|s\|_{P'} \leq \min(\alpha_r, \beta) \\ 1, & \text{if } \|s\|_{P'} > \min(\alpha_r, \beta) \end{cases}$$

and $\mathbf{I}_1(s) = 1$. And, let π be a possibility distribution with the following definition:

$$\pi(\mathbf{I}) = \begin{cases} 1, & \text{if } \mathbf{I} = \mathbf{I}_1 \\ 1 - \min(\alpha_r, \beta), & \text{if } \mathbf{I} = \mathbf{I}_0 \\ 0, & \text{otherwise.} \end{cases}$$

Because of the definition of π and Proposition 5.4, we have that $\pi \models_{PGL} P'$. And, as $\|q\|_{P'} = \min(\alpha_r, \beta)$, we have that $\mathbf{I}_0(q) = 0$ and $\mathbf{I}_0(q \rightarrow p) = 1$, and thus, $\pi \models_{PGL} (q \rightarrow p, \gamma)$ and $N^*([q] \mid \pi) = \min(\alpha_r, \beta)$. Therefore, $\pi \models_{PGL} P$ and, by Proposition 5.1, $\|q\|_P = \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} P\} \leq \min(\alpha_r, \beta)$, and thus, $\|q\|_P \leq \min(\alpha_r, \beta) = \|q\|_{P'}$.

■

Theorem 5.3 (completeness) *Let P be a set of PGL clauses and let q be a goal. Then, $\|q\|_P = |q|_P$.*

Proof: By the soundness of the modus ponens inference rule, $\|q\|_P \geq |q|_P$. Therefore, we must prove $\|q\|_P \leq |q|_P$ and we proceed by induction on n, where n is the number of clauses of P.

If $n = 1$, then it must be that P contains only either one certainty-weighted fact or rule. We assume that q occurs in P as the head of a non recursive formula; otherwise, by Proposition 5.2, $\|q\|_P = |q|_P = 0$.

Suppose that P contains only the certainty-weighted fact (q, γ). Let $\mathbf{I}_0$ and $\mathbf{I}_1$ be two interpretations such that $\mathbf{I}_0(q) < 1 - \gamma$ and $\mathbf{I}_1(q) = 1$. Now, let π be a possibility distribution with the following definition:

$$\pi(\mathbf{I}) = \begin{cases} 1, & \text{if } \mathbf{I} = \mathbf{I}_1 \\ 1 - \gamma, & \text{if } \mathbf{I} = \mathbf{I}_0 \\ 0, & \text{otherwise.} \end{cases}$$

It is easy to check that $\pi \models_{PGL} (q,\gamma)$ and $N^*([q] \mid \pi) = \gamma$. Then, by Proposition 5.1, $\|q\|_P = \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} (q,\gamma)\} \leq \gamma$. On the other hand, $|q|_P = \max\{\alpha \in [0,1] \mid (q,\gamma) \vdash^*_{PGL} (q,\alpha)\} = \gamma$, and thus, $\|q\|_P \leq |q|_P$.

Suppose that P contains only the certainty-weighted rule $(r \rightarrow q,\gamma)$. Let $\mathbf{I}_0$ be an interpretation such that $\mathbf{I}_0(r) \leq \mathbf{I}_0(q) < 1$ and let π be a possibility distribution with the following definition:

$$\pi(\mathbf{I}) = \begin{cases} 1, & \text{if } \mathbf{I} = \mathbf{I}_0 \\ 0, & \text{otherwise.} \end{cases}$$

It is easy to check that $\pi \models_{PGL} (r \rightarrow q,\gamma)$ and $N^*([q] \mid \pi) = 0$. Then, by Proposition 5.1, $\|q\|_P = \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} (r \rightarrow q,\gamma)\} = 0$, and thus, $\|q\|_P \leq |q|_P$.

Suppose now that for each set P' that contains n clauses it holds that $\|s\|_{P'} \leq |s|_{P'}$ for each propositional variable s, and suppose that P contains $n+1$ clauses. Since we are assuming that q occurs in P as the head of a non recursive formula, let (φ,γ) be a clause of P such that φ is a non recursive formula with head q and $\gamma \geq \delta$, for each clause (ψ,δ) of P such that ψ is a non recursive formula with head q. We distinguish two cases:

Case $(\varphi,\gamma) = (q,\gamma)$. Let $\mathbf{I}_0$ and $\mathbf{I}_1$ be two interpretations such that $\mathbf{I}_0(q) < 1-\gamma$ and $\mathbf{I}_0(p) = 1$ for each propositional variable $p \neq q$, and $\mathbf{I}_1(s) = 1$ for each propositional variable s. Now, let π be a possibility distribution with the following definition:

$$\pi(\mathbf{I}) = \begin{cases} 1, & \text{if } \mathbf{I} = \mathbf{I}_1 \\ 1-\gamma, & \text{if } \mathbf{I} = \mathbf{I}_0 \\ 0, & \text{otherwise.} \end{cases}$$

Since $\gamma \geq \delta$ for each non recursive clause (ψ,δ) of P such that q is the head of ψ, we have that $\pi \models_{PGL} P$ and $N^*([q] \mid \pi) = \gamma$. Therefore, by Proposition 5.1, $\|q\|_P = \inf\{N^*([q] \mid \pi) \mid \pi \models_{PGL} P\} \leq \gamma$. On the other hand, $|q|_P = \max\{\alpha \in [0,1] \mid P \vdash^*_{PGL} (q,\alpha)\} \geq \gamma$, and thus, $\|q\|_P \leq |q|_P$.

Case $(\varphi,\gamma) = (p \rightarrow q,\gamma)$. We define $P' = P \backslash \{(p \rightarrow q,\gamma)\}$. Then, $\|q\|_P \geq \|q\|_{P'}$. Now,

- If $\|q\|_P = \|q\|_{P'}$, by the induction hypothesis, $\|q\|_{P'} \leq |q|_{P'}$, and thus, $\|q\|_P \leq |q|_P$.
- If $\|q\|_P > \|q\|_{P'}$, by Proposition 5.5, $\|q\|_P = \|q\|_{\{(p,\alpha_p),(p\rightarrow q,\gamma)\}}$, where $\alpha_p = \|p\|_P$, and, by Proposition 5.3, $\|q\|_{\{(p,\alpha_p),(p\rightarrow q,\gamma)\}} = \min(\alpha_p,\gamma)$. Now, by Proposition 5.6, $\|p\|_P = \|p\|_{P'}$ and, by the induction hypothesis, $\|p\|_{P'} \leq |p|_{P'}$. Since $|p|_{P'} \leq |p|_P$ and $(p \rightarrow q,\gamma) \in P$, applying the modus ponens inference rule, we get $P \vdash^*_{PGL} (q, \min(|p|_P,\gamma))$, with $\min(|p|_P,\gamma) \geq \min(\alpha_p,\gamma)$. Then, $|q|_P = \max\{\alpha \in [0,1] \mid P \vdash^*_{PGL} (q,\alpha)\} \geq \min(|p|_P,\gamma)$, and thus, $|q|_P \geq \min(\alpha_p,\gamma) = \|q\|_P$.

■

In the particular case that we do not allow recursive formulas in the language, the underlying uncertainty logic of our logic programming system is syntactically equivalent to the family of infinitely-valued propositional logics the interpreter defined by Escalada-Imaz and Manyà (1995) can deal with. The interpreter is based on a backward proof algorithm for computing the maximum degree of deduction of a propositional variable from a set of formulas whose worst-case time complexity is linear in the total number of occurrences of propositional variables in the set of formulas. We show below an example of PGL clauses the interpreter can deal with.

Example 5.1 The maximum degree of deduction of the goal $friend\_Mary\_John$ from the set of PGL clauses

$$\begin{aligned} P = \{\ & (Mary\_is\_young, 0.8), \\ & (John\_is\_young, 0.9), \\ & (Mary\_is\_young \ \wedge \ John\_is\_young \rightarrow friend\_Mary\_John, 0.6)\ \}, \end{aligned}$$

is 0.6 which corresponds with the deduction degree computed by the interpreter when taking as triangular norm the min-conjunction function and as implication Gödel's many-valued implication function. □

Chapter 6

A complete calculus for PGL extended with fuzzy constants

6.1 Introduction

In Chapter 5, we defined PGL, a general possibilistic logic with fuzzy propositional variables based on Gödel infinitely-valued logic, and we provided the Horn-rule fragment of PGL with a complete modus ponens-style calculus for determining the maximum degree of possibilistic entailment of a fuzzy propositional variable from a set of clauses. Now in this chapter we extend the Horn-rule fragment of PGL with both fuzzy constants and a semantical unification mechanism based on a necessity evaluation of fuzzy events which preserves completeness for a particular class of set of clauses. We refer to this extension as PGL^+.

In this framework, we formalize the notion of non-recursive and satisfiable PGL^+ program and we prove, for this restricted class of programs further satisfying a context constraint, that the modus ponens-style calculus extended with the semantical unification mechanism is complete for determining the maximum degree of possibilistic belief with which an atomic formula with fuzzy constants can be entailed.

The chapter is organized as follows. In Section 6.2, we present the syntax, the semantics and the logical inference of PGL^+. In Section 6.3 we formalize the notion of non-recursive and satisfiable PGL^+ program. In Section 6.4, we prove that the PGL^+ calculus is complete for determining the maximum degree of possibilistic entailment of an atomic formula with fuzzy constants for the mentioned class of programs satisfying a context constraint. And, in Section 6.5, we define an efficient proof algorithm for determining the maximum degree of deduction of an atomic formula with fuzzy constants from a non-recursive and satisfiable PGL^+ program, which is the the maximum degree of possibilistic

entailment whenever the program satisfies the context constraint.

6.2 Adding fuzzy unification: PGL$^+$

As we have already pointed out, our aim in this chapter is to extend the modus ponens-style calculus of PGL to allow a semantical matching between fuzzy knowledge based on a necessity evaluation of fuzzy events. For instance, given the set of PGL clauses

$$\begin{aligned} P = \{\ & (Mary\_is\_middle\_age, 0.8), \\ & (Peter\_is\_about\_35, 0.9), \\ & (Mary\_is\_middle\_age \ \wedge \ Peter\_is\_middle\_age \rightarrow friends,\ 0.6)\ \}, \end{aligned}$$

where $Peter\_is\_about\_35$ and $Peter\_is\_middle\_age$ are two fuzzy propositional variables, the maximum degree of deduction of the goal $friends$ from P is 0 unless we were able to compute the necessity evaluation of a person to be $middle\_age$ from the fact that the person is $about\_35$ years old.

In this setting, for the sake of a more clear notation, we represent fuzzy propositional variables by means of typed unary regular predicates[1] and sorted fuzzy constants. Indeed, as object variables and function symbols are not allowed in PGL$^+$, the language still remains propositional and unary predicates give us a more simple representation model without loss of expressiveness. For instance, the fuzzy propositional variable $Peter\_is\_about\_35$ is represented in PGL$^+$ by the atomic formula

$$age\_Peter(about\_35),$$

where age_Peter is a classical predicate of type (`years_old`) and $about\_35$ is a fuzzy constant defined over the domain `years_old`.

In PLFC and in PGL$^+$, fuzzy constants can be seen as (flexible) restrictions on an existential quantifier. In the case in which $about\_35$ denotes a crisp interval of ages, say $[34, 36]$, the *certainty-weighted* formula

$$(age\_Peter(about\_35), 0.9)$$

is interpreted in both systems as "$\exists x \in [34, 36]$ such that $age\_Peter(x)$" is certain with a necessity of at least 0.9. In the case in which $about\_35$ denotes a fuzzy interval with a membership function $\mu_{about\_35} : [0, 120] \rightarrow [0, 1]$, because in each system the necessity measure used for defining the possibilistic semantics is different, in each system the certainty-weighted formula has a different interpretation. In PLFC it is interpreted as "$\exists\ x \in [\mu_{about\_35}]_{0.9}$ such that $age\_Peter(x)$" is certain with a necessity of at least 0.9, and in PGL$^+$ as "$\exists\ x \in [\mu_{about\_35}]_\alpha$ such that $age\_Peter(x)$" is certain with a necessity of at least $\min(0.9, 1 - \alpha)$, for each $\alpha \in [0, 1]$, where $[\mu_{about\_35}]_\alpha$ denotes the α-cut of μ_{about_35}.

$(age\_Peter([\mu_{about\_35}]_\beta)$ is certain with necessity of at least $\min(0.9, 1 - \beta)$, for each $\beta \in [0, 1]$.

[1] In the next chapter, we extend PGL$^+$ to the first-order case, and thus, as we did for PLFC, we consider typed regular predicates of arity n.

In the next section, we formalize the syntax and the semantics of PGL$^+$, while in Section 6.2.2 we develop an inference mechanism for PGL$^+$ based on a possibilistic pattern matching measure which preserves completeness for a particular class of clauses.

6.2.1 The PGL$^+$ language: Syntax and semantics

The *basic components* of PGL$^+$ are:

- *Sorts* of constants. A *type* is a tuple of sorts.
- A set $\mathcal{C}$ of object *constants* (crisp and fuzzy constants), each having its sort.
- A set *Var* of *primitive propositions*.
- A set *Pred* of unary *regular predicates*, each one having a type.
- *Connectives* $\wedge$ and $\rightarrow$.

Definition 6.1 (PGL$^+$ atomic formula) *A PGL$^+$ atomic formula is either a primitive proposition from Var or of the form $p(A)$, where p is a predicate symbol from Pred, A is an object constant from $\mathcal{C}$ and the sort of A corresponds to the type of p.*

Definition 6.2 (PGL$^+$ clause) *A PGL$^+$ clause is a pair of the form (φ, α), where φ is either a fact or a rule built on PGL$^+$ atomic formulas, and $\alpha \in [0,1]$ is a lower bound on the belief on φ in terms of necessity measures. We talk of a* unit *PGL$^+$ clause when φ is a fact and of a* general *PGL$^+$ clause, otherwise.*

With respect to the underlying fuzzy logic described in Section 5.2, the main differences are that now we extend the language with fuzzy constants and regular predicates, and that we attach fuzzy constants with a *sort*. In doing so, we are introducing some minor changes in the semantics. Namely, interpretations should map a sort into a non-empty domain, a fuzzy constant into a fuzzy set, and a predicate symbol into a value of the domain attached with its type. Hence, we need to provide a new notion of interpretation.

Definition 6.3 (PGL$^+$ interpretation) *A PGL$^+$ interpretation $\mathbf{M} = (U, i, m)$ maps:*

1. *each sort σ into a non-empty domain U_σ;*
2. *a primitive proposition into a truth value of the unit interval $[0,1]$;*
3. *a predicate p of type (σ) into a value $i(p) \in U_\sigma$; and*

4. *an object constant A (crisp or fuzzy constant) of sort σ into a normalized fuzzy set $m(A) : U_\sigma \to [0,1]$. We denote the membership function of $m(A)$ by $\mu_{m(A)}$. When A is a precise constant, $m(A)$ is a value of the domain U_σ and it is also represented by a fuzzy set given by*

$$\mu_{m(A)}(u) = \begin{cases} 1, & \text{if } u = m(A) \\ 0, & \text{for each } u \in U_\sigma \text{ such that } u \neq m(A). \end{cases}$$

Remark that a PGL$^+$ interpretation $\mathbf{M} = (U, i, m)$ is a disjunctive interpretation in the sense that $i(p)$ is a unique value of the domain for each predicate symbol p. Indeed, in contrast to PLFC, in PGL$^+$ fuzzy constants always express disjunctive information. For instance, the following rule:

$$speaks\_Mary(\{Spanish, French\}) \to visited\_Mary(\{Spain, France\})$$

is interpreted in PGL$^+$ as

"if Mary speaks either Spanish or French,
then Mary has visited either Spain or France",

and thus, in PGL$^+$ each imprecise constant is interpreted as disjunctive information. However, in PLFC the above rule should be interpreted as:

"if Mary speaks Spanish and French,
then Mary has visited either Spain or France",

and thus, conjunctive interpretations can be modeled in PLFC but not in PGL$^+$.

Definition 6.4 (truth value of a PGL$^+$ atomic formula) *The truth value of a PGL$^+$ atomic formula φ under an interpretation $\mathbf{M} = (U, i, m)$, denoted by $\|\varphi\|_\mathbf{M}$, is just the truth value $i(q)$ if φ is a primitive proposition q, and it is computed as $\mu_{m(A)}(i(p))$ if φ is of the form $p(A)$.*

The truth value of PGL$^+$ atomic formulas extend to rules in the usual way by means of the min-conjunction and Gödel's many-valued implication:

$$\|p_1 \wedge \cdots \wedge p_k \to q\|_\mathbf{M} = \begin{cases} 1, & \text{if } \min(\|p_1\|_\mathbf{M}, \ldots, \|p_k\|_\mathbf{M}) \leq \|q\|_\mathbf{M} \\ \|q\|_\mathbf{M}, & \text{otherwise.} \end{cases}$$

Remark that the truth value of a PGL$^+$ atomic formula $p(A)$ under an interpretation $\mathbf{M} = (U, i, m)$ depends not only on the value $i(p)$ assigned to p, but on the fuzzy set $m(A)$ assigned to the object constant A. Therefore, as in PLFC, in order to measure the certainty of an atomic formula with fuzzy constants in a possibilistic model, we cannot take into account all possible PGL$^+$ interpretations, but only those which share a common interpretation of object constants, and hence which also share their domain. This leads us to define the notion of PGL$^+$ context (cf. Section 4.3.2).

Let U be a collection of non-empty domains and let m be an interpretation of object constants over U (or over $[0,1]^U$ in the case of fuzzy constants). We

define the PGL$^+$ *context* determined by U and m, denoted by $\mathcal{M}_{U,m}$, as the set of PGL$^+$ interpretations having U as a domain and m as an interpretation of object constants.

Now we are ready to introduce the notion of possibilistic model and possibilistic entailment in a PGL$^+$ context $\mathcal{M}_{U,m}$.

Given a PGL$^+$ context $\mathcal{M}_{U,m}$, a PGL$^+$ *possibilistic model* is a normalized possibility distribution $\pi : \mathcal{M}_{U,m} \to [0,1]$ on the set of PGL$^+$ interpretations $\mathcal{M}_{U,m}$. A possibilistic model $\pi : \mathcal{M}_{U,m} \to [0,1]$ *satisfies* a PGL$^+$ clause (φ, α), written $\pi \models^{U,m}_{PGL^+} (\varphi, \alpha)$, iff $N^*([\varphi] \mid \pi) \geq \alpha$, where

$$N^*([\varphi] \mid \pi) = \inf_{\mathbf{M} \in \mathcal{M}_{U,m}} \pi(\mathbf{M}) \Rightarrow \|\varphi\|_{\mathbf{M}},$$

$\Rightarrow$ being the reciprocal of Gödel's many-valued implication[2].

Finally, a set of PGL$^+$ clauses P *entails* another PGL$^+$ clause (φ, α) in the context determined by U and m, written $P \models^{U,m}_{PGL^+} (\varphi, \alpha)$, iff every possibilistic model $\pi : \mathcal{M}_{U,m} \to [0,1]$ satisfying all the clauses in P also satisfies (φ, α). We refer to a possibilistic model $\pi : \mathcal{M}_{U,m} \to [0,1]$ satisfying all the clauses in a set P as a model of P.

As in PGL, our aim is to provide PGL$^+$ with a complete calculus for determining the maximum degree of possibilistic entailment of a PGL$^+$ atomic formula, called PGL$^+$ *goal*, from a set of PGL$^+$ clauses. The notion of maximum degree of possibilistic entailment of PGL can be easily extended to PGL$^+$ in the following way.

Given a PGL$^+$ context $\mathcal{M}_{U,m}$, *the maximum degree of possibilistic entailment* of a PGL$^+$ goal $q(C)$ from a set of PGL$^+$ clauses P, denoted by $\|q(C)\|^{U,m}_P$, is the greatest lower bound $\alpha \in [0,1]$ on the belief on $q(C)$ such that P entails $(q(C), \alpha)$ in the context determined by U and m. Thus,

$$\|q(C)\|^{U,m}_P = \sup\{\alpha \in [0,1] \mid P \models^{U,m}_{PGL^+} (q(C), \alpha)\}.$$

Then, Proposition 5.1 and Corollary 5.1 extend to PGL$^+$ in the following way.

Proposition 6.1 *Given a PGL$^+$ context $\mathcal{M}_{U,m}$,*

$$\|q(C)\|^{U,m}_P = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P\}.$$

Corollary 6.1 *Given a PGL$^+$ context $\mathcal{M}_{U,m}$, $P \models^{U,m}_{PGL^+} (q(C), \|q(C)\|^{U,m}_P)$.*

6.2.2 Extended inference with possibilistic pattern matching

To provide PGL$^+$ with a fuzzy unification mechanism we need a measure for computing the necessity evaluation of a fuzzy constant B based on a different fuzzy constant A, both of a same sort. Furthermore, this measure must enable us

[2]Remember that $x \Rightarrow y = 1$, if $x \leq y$, and $x \Rightarrow y = 1 - x$, otherwise.

to extend the modus ponens-style calculus of PGL in order to keep completeness for determining the maximum degree of possibilistic entailment of a PGL$^+$ goal from a set of PGL$^+$ clauses in a particular context.

Again there are several alternatives. After a careful analysis we have chosen the same type of measure used when defining the possibilistic semantics for PGL$^+$. Namely, given a PGL$^+$ context $\mathcal{M}_{U,m}$, the necessity evaluation of B based on a fuzzy constant A, both of a sort σ, is defined as

$$N^*(m(B) \mid m(A)) = \inf_{u \in U_\sigma} \mu_{m(A)}(u) \Rightarrow \mu_{m(B)}(u),$$

where $\Rightarrow$ is the reciprocal of Gödel's many-valued implication.

At this point we are ready to extend the modus ponens-style calculus to allow a semantical unification of fuzzy constants through the above possibilistic pattern matching measure. On the one hand, one can prove that the semantical unification between fuzzy constants can only be performed on unit PGL$^+$ clauses. On the other hand, the possibilistic pattern matching measure produces unification degrees which are all the stronger as $\mu_{m(B)}$ is large and $\mu_{m(A)}$ is small. Indeed, given a PGL$^+$ context $\mathcal{M}_{U,m}$

$$N^*(m(B) \mid m(A)) \geq N^*(m(B') \mid m(A)) \text{ if } \mu_{m(B)} \geq \mu_{m(B')}$$

and

$$N^*(m(B) \mid m(A)) \geq N^*(m(B) \mid m(A')) \text{ if } \mu_{m(A)} \leq \mu_{m(A')}.$$

This points out that it is interesting to have PGL$^+$ clauses with the largest possible body and the smallest possible head. Therefore, although there may exist several approaches, we have finally decided to extend the calculus with the following inference rules.

Given a PGL$^+$ context $\mathcal{M}_{U,m}$, the calculus for PGL$^+$ has the following triviality axiom and inference rules:

Axiom: $(\varphi, 0)$.

Generalized resolution:

$$\frac{(p \wedge s \rightarrow q(A), \alpha) \quad (q(B) \wedge t \rightarrow r, \beta)}{(p \wedge s \wedge t \rightarrow r, \min(\alpha, \beta))} \text{ [RE]}$$

if $\mu_{m(A)} \leq \mu_{m(B)}$.

Fusion:

$$\frac{(p(A) \wedge s \rightarrow q(D), \alpha) \quad (p(B) \wedge t \rightarrow q(E), \beta)}{(p(C) \wedge s \wedge t \rightarrow q(F), \min(\alpha, \beta))} \text{ [FU]}$$

if $\mu_{m(C)} \leq \max(\mu_{m(A)}, \mu_{m(B)})$ and $\mu_{m(F)} \geq \max(\mu_{m(D)}, \mu_{m(E)})$.

Semantical unification:

$$\frac{(p(A), \alpha)}{(p(B), \min(\alpha, N^*(m(B) \mid m(A))))} \text{ [SU]},$$

where $N^*(m(B) \mid m(A)) = \inf_{u \in U_\sigma} \mu_{m(A)}(u) \Rightarrow \mu_{m(B)}(u)$, $\Rightarrow$ being the reciprocal of Gödel's many-valued implication.

Intersection:

$$\frac{\begin{array}{c}(p(A), \alpha) \\ (p(B), \beta)\end{array}}{(p(C), \min(\alpha, \beta))} \text{ [}IN\text{]}$$

if $\mu_{m(C)} \geq \min(\mu_{m(A)}, \mu_{m(B)})$.

Resolving uncertainty:

$$\frac{(p(A), \alpha)}{(p(B), 1)} \text{ [UN]}$$

if $\mu_{m(B)} \geq \max(1 - \alpha, \mu_{m(A)})$.

The following *generalized modus ponens* inference rule is the result of repeatedly applying the RE inference rule to a general PGL$^+$ clause and a set of unit PGL$^+$ clauses, which can be obtained from a different set of unit PGL$^+$ clauses by applying the SU inference rule.

$$\frac{\begin{array}{c}(p_1 \wedge \cdots \wedge p_k \rightarrow q, \alpha) \\ (p_1, \beta_1), \ldots, (p_k, \beta_k)\end{array}}{(q, \min(\alpha, \beta_1, \ldots, \beta_k))} \text{ [MP]}.$$

Theorem 6.1 (soundness of the PGL$^+$ inference rules) *For each PGL$^+$ context $\mathcal{M}_{U,m}$, the RE, FU, SU, IN and UN inference rules are sound with respect to the possibilistic entailment of PGL$^+$ clauses.*

Proof:

RE: Given a PGL$^+$ context $\mathcal{M}_{U,m}$, we must prove, for each possibilistic model $\pi : \mathcal{M}_{U,m} \rightarrow [0,1]$, that if $\pi \models^{U,m}_{PGL^+} (p \wedge s \rightarrow q(A), \alpha)$ and $\pi \models^{U,m}_{PGL^+} (q(B) \wedge t \rightarrow r, \beta)$ with $\mu_{m(A)} \leq \mu_{m(B)}$, then $\pi \models^{U,m}_{PGL^+} (p \wedge s \wedge t \rightarrow r, \min(\alpha, \beta))$.

Assume that $\pi \models^{U,m}_{PGL^+} (p \wedge s \rightarrow q(A), \alpha)$ and $\pi \models^{U,m}_{PGL^+} (q(B) \wedge t \rightarrow r, \beta)$. This means that $N^*([p \wedge s \rightarrow q(A)] \mid \pi) \geq \alpha$ and $N^*([q(B) \wedge t \rightarrow r] \mid \pi) \geq \beta$. The two conditions amount to, for each PGL$^+$ interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$, $\pi(\mathbf{M}) \Rightarrow \|p \wedge s \rightarrow q(A)\|_{\mathbf{M}} \geq \alpha$ and $\pi(\mathbf{M}) \Rightarrow \|q(B) \wedge t \rightarrow r\|_{\mathbf{M}} \geq \beta$. Therefore, $\pi(\mathbf{M}) \Rightarrow \min(\|p \wedge s \rightarrow q(A)\|_{\mathbf{M}}, \|q(B) \wedge t \rightarrow r\|_{\mathbf{M}}) \geq \min(\alpha, \beta)$, and thus, $\pi(\mathbf{M}) \Rightarrow \min(\min(\|p\|_{\mathbf{M}}, \|s\|_{\mathbf{M}}) \rightarrow \|q(A)\|_{\mathbf{M}}, \min(\|q(B)\|_{\mathbf{M}}, \|t\|_{\mathbf{M}}) \rightarrow \|r\|_{\mathbf{M}}) \geq \min(\alpha, \beta)$. As $\mu_{m(A)} \leq \mu_{m(B)}$, we have that $\|q(A)\|_{\mathbf{M}} \leq \|q(B)\|_{\mathbf{M}}$ for each PGL$^+$ interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$. Hence, $\pi(\mathbf{M}) \Rightarrow \min(\min(\|p\|_{\mathbf{M}}, \|s\|_{\mathbf{M}}) \rightarrow$

$\|q(B)\|_{\mathbf{M}}, \min(\|q(B)\|_{\mathbf{M}}, \|t\|_{\mathbf{M}}) \to \|r\|_{\mathbf{M}}) \geq \min(\alpha, \beta)$. But, by residuation[3], $\min(\|q(B)\|_{\mathbf{M}}, \|t\|_{\mathbf{M}}) \to \|r\|_{\mathbf{M}} = \|q(B)\|_{\mathbf{M}} \to (\|t\|_{\mathbf{M}} \to \|r\|_{\mathbf{M}})$ and $\min(\min(\|p\|_{\mathbf{M}}, \|s\|_{\mathbf{M}}) \to \|q(B)\|_{\mathbf{M}}, \|q(B)\|_{\mathbf{M}} \to (\|t\|_{\mathbf{M}} \to \|r\|_{\mathbf{M}})) \leq \min(\|p\|_{\mathbf{M}}, \|s\|_{\mathbf{M}}) \to (\|t\|_{\mathbf{M}} \to \|r\|_{\mathbf{M}}) = \min(\|p\|_{\mathbf{M}}, \|s\|_{\mathbf{M}}, \|t\|_{\mathbf{M}}) \to \|r\|_{\mathbf{M}} = \|p \wedge s \wedge t \to r\|_{\mathbf{M}}$. Then, $\pi(\mathbf{M}) \Rightarrow \|p \wedge s \wedge t \to r\|_{\mathbf{M}} \geq \min(\alpha, \beta)$ for each PGL$^+$ interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$. Hence, $N^*([p \wedge s \wedge t \to r] \mid \pi) \geq \min(\alpha, \beta)$, and thus, $\pi \models^{U,m}_{PGL^+} (p \wedge s \wedge t \to r, \min(\alpha, \beta))$ as well.

FU: Given a PGL$^+$ context $\mathcal{M}_{U,m}$, we must prove, for each possibilistic model $\pi : \mathcal{M}_{U,m} \to [0,1]$, that if $\pi \models^{U,m}_{PGL^+} (p(A) \wedge s \to q(D), \alpha)$ and $\pi \models^{U,m}_{PGL^+} (p(B) \wedge t \to q(E), \beta)$, then $\pi \models^{U,m}_{PGL^+} (p(C) \wedge s \wedge t \to q(F) \min(\alpha, \beta))$ whenever $\mu_{m(C)} \leq \max(\mu_{m(A)}, \mu_{m(B)})$ and $\mu_{m(F)} \geq \max(\mu_{m(D)}, \mu_{m(E)})$.

Assume that $\pi \models^{U,m}_{PGL^+} (p(A) \wedge s \to q(D), \alpha)$ and $\pi \models^{U,m}_{PGL^+} (p(B) \wedge t \to q(E), \beta)$. This means that $N^*([p(A) \wedge s \to q(D)] \mid \pi) \geq \alpha$ and $N^*([p(B) \wedge t \to q(E)] \mid \pi) \geq \beta$. The two conditions amount to, for each PGL$^+$ interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$, $\pi(\mathbf{M}) \Rightarrow \|p(A) \wedge s \to q(D)\|_{\mathbf{M}} \geq \alpha$ and $\pi(\mathbf{M}) \Rightarrow \|p(B) \wedge t \to q(E)\|_{\mathbf{M}} \geq \beta$. Therefore, $\pi(\mathbf{M}) \Rightarrow \min(\|p(A) \wedge s \to q(D)\|_{\mathbf{M}}, \|p(B) \wedge t \to q(E)\|_{\mathbf{M}}) \geq \min(\alpha, \beta)$, and thus, $\pi(\mathbf{M}) \Rightarrow \min(\min(\|p(A)\|_{\mathbf{M}}, \|s\|_{\mathbf{M}}) \to \|q(D)\|_{\mathbf{M}}, \min(\|p(B)\|_{\mathbf{M}}, \|t\|_{\mathbf{M}}) \to \|q(E)\|_{\mathbf{M}}) \geq \min(\alpha, \beta)$.

Now,

$$
\begin{aligned}
&\min(\min(\|p(A)\|_{\mathbf{M}}, \|s\|_{\mathbf{M}}) \to \|q(D)\|_{\mathbf{M}},\\
&\min(\|p(B)\|_{\mathbf{M}}, \|t\|_{\mathbf{M}}) \to \|q(E)\|_{\mathbf{M}})\\
&\qquad \leq \min(\min(\|p(A)\|_{\mathbf{M}}, \|s\|_{\mathbf{M}}) \to \max(\|q(D)\|_{\mathbf{M}}, \|q(E)\|_{\mathbf{M}}),\\
&\qquad\quad \min(\|p(B)\|_{\mathbf{M}}, \|t\|_{\mathbf{M}}) \to \max(\|q(D)\|_{\mathbf{M}}, \|q(E)\|_{\mathbf{M}}))\\
&\qquad \leq \max(\min(\|p(A)\|_{\mathbf{M}}, \|s\|_{\mathbf{M}}),\\
&\qquad\quad \min(\|p(B)\|_{\mathbf{M}}, \|t\|_{\mathbf{M}})) \to \max(\|q(D)\|_{\mathbf{M}}, \|q(E)\|_{\mathbf{M}}).
\end{aligned}
$$

Then, as

$$
\begin{gathered}
\max(\min(\|p(A)\|_{\mathbf{M}}, \|s\|_{\mathbf{M}}), \min(\|p(B)\|_{\mathbf{M}}, \|t\|_{\mathbf{M}}))\\
\geq\\
\min(\max(\|p(A)\|_{\mathbf{M}}, \|p(B)\|_{\mathbf{M}}), \|s\|_{\mathbf{M}}, \|t\|_{\mathbf{M}}),
\end{gathered}
$$

we have that

$$
\begin{gathered}
\max(\min(\|p(A)\|_{\mathbf{M}}, \|s\|_{\mathbf{M}}), \min(\|p(B)\|_{\mathbf{M}}, \|t\|_{\mathbf{M}})) \to\\
\max(\|q(D)\|_{\mathbf{M}}, \|q(E)\|_{\mathbf{M}})\\
\leq\\
\min(\max(\|p(A)\|_{\mathbf{M}}, \|p(B)\|_{\mathbf{M}}), \|s\|_{\mathbf{M}}, \|t\|_{\mathbf{M}}) \to\\
\max(\|q(D)\|_{\mathbf{M}}, \|q(E)\|_{\mathbf{M}}),
\end{gathered}
$$

and thus, $\pi(\mathbf{M}) \Rightarrow (\min(\max(\|p(A)\|_{\mathbf{M}}, \|p(B)\|_{\mathbf{M}}), \|s\|_{\mathbf{M}}, \|t\|_{\mathbf{M}}) \to \max(\|q(D)\|_{\mathbf{M}}, \|q(E)\|_{\mathbf{M}})) \geq \min(\alpha, \beta)$.

[3] If $(\otimes, \Rightarrow)$ is a residuated pair, it holds that: (i) $((x \otimes y) \Rightarrow z) = (x \Rightarrow (y \Rightarrow z))$; and (ii) $(x \Rightarrow y) \otimes (y \Rightarrow z) \leq x \Rightarrow z$.

Finally, as $\|p(C)\|_{\mathbf{M}} \leq \max(\|p(A)\|_{\mathbf{M}}, \|p(B)\|_{\mathbf{M}})$ and $\|q(F)\|_{\mathbf{M}} \geq \max(\|q(D)\|_{\mathbf{M}}, \|q(E)\|_{\mathbf{M}})$, we have that

$$\pi(\mathbf{M}) \Rightarrow (\min(\|p(C)\|_{\mathbf{M}}, \|s\|_{\mathbf{M}}, \|t\|_{\mathbf{M}}) \rightarrow \|q(F)\|_{\mathbf{M}}) \geq \min(\alpha, \beta).$$

Hence, $\pi(\mathbf{M}) \Rightarrow \|p(C) \wedge s \wedge t \rightarrow q(F)\|_{\mathbf{M}} \geq \min(\alpha, \beta)$ for each PGL$^+$ interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$, and thus, $\pi \models^{U,m}_{PGL^+} (p(C) \wedge s \wedge t \rightarrow q(F), \min(\alpha, \beta))$ as well.

SU: Given a PGL$^+$ context $\mathcal{M}_{U,m}$, we must prove, for each possibilistic model $\pi : \mathcal{M}_{U,m} \rightarrow [0,1]$, that if $\pi \models^{U,m}_{PGL^+} (p(A), \alpha)$, then $\pi \models^{U,m}_{PGL^+} (p(B), \min(\alpha, N^*(m(B) \mid m(A))))$.

Assume that $\pi \models^{U,m}_{PGL^+} (p(A), \alpha)$. This means that, $N^*([p(A)] \mid \pi) \geq \alpha$, and thus, $\pi(\mathbf{M}) \Rightarrow \|p(A)\|_{\mathbf{M}} \geq \alpha$ for each PGL$^+$ interpretation $\mathbf{M} = (U, i, m) \in \mathcal{M}_{U,m}$. Then, we have the following consecutive inequalities:

1. $\pi(\mathbf{M}) \Rightarrow \mu_{m(A)}(i(p)) \geq \alpha$
2. $\min(\pi(\mathbf{M}) \Rightarrow \mu_{m(A)}(i(p)), \mu_{m(A)}(i(p)) \Rightarrow \mu_{m(B)}(i(p))) \geq \min(\alpha, \mu_{m(A)}(i(p)) \Rightarrow \mu_{m(B)}(i(p)))$
3. $\min(\pi(\mathbf{M}) \Rightarrow \mu_{m(A)}(i(p)), \mu_{m(A)}(i(p)) \Rightarrow \mu_{m(B)}(i(p))) \leq \pi(\mathbf{M}) \Rightarrow \mu_{m(B)}(i(p))$
4. $\pi(\mathbf{M}) \Rightarrow \mu_{m(B)}(i(p)) \geq \min(\alpha, \mu_{m(A)}(i(p)) \Rightarrow \mu_{m(B)}(i(p)))$
5. $\inf_{\mathbf{M} \in \mathcal{M}_{U,m}} \pi(\mathbf{M}) \Rightarrow \mu_{m(B)}(i(p)) \geq \inf_{\mathbf{M} \in \mathcal{M}_{U,m}} \min(\alpha, \mu_{m(A)}(i(p)) \Rightarrow \mu_{m(B)}(i(p)))$
6. $\inf_{\mathbf{M} \in \mathcal{M}_{U,m}} \pi(\mathbf{M}) \Rightarrow \mu_{m(B)}(i(p)) \geq \min(\alpha, \inf_{\mathbf{M} \in \mathcal{M}_{U,m}} \mu_{m(A)}(i(p)) \Rightarrow \mu_{m(B)}(i(p)))$
7. $N^*([p(B)] \mid \pi) \geq \min(\alpha, \inf_{u \in U_\sigma} \mu_{m(A)}(u) \Rightarrow \mu_{m(B)}(u))$
8. $N^*([p(B)] \mid \pi) \geq \min(\alpha, N^*(m(B) \mid m(A)))$

Thus, $\pi \models^{U,m}_{PGL^+} (p(B), \min(\alpha, N^*(m(B) \mid m(A))))$ as well.

IN: Given a PGL$^+$ context $\mathcal{M}_{U,m}$, we must prove, for each possibilistic model $\pi : \mathcal{M}_{U,m} \rightarrow [0,1]$, that if $\pi \models^{U,m}_{PGL^+} (p(A), \alpha)$ and $\pi \models^{U,m}_{PGL^+} (p(B), \beta)$, then $\pi \models^{U,m}_{PGL^+} (p(C), \min(\alpha, \beta))$ whenever $\mu_{m(C)} \geq \min(\mu_{m(A)}, \mu_{m(B)})$.

The two conditions amount to, for each PGL$^+$ interpretation $\mathbf{M} = (U, i, m) \in \mathcal{M}_{U,m}$, $\pi(\mathbf{M}) \Rightarrow \|p(A)\|_{\mathbf{M}} \geq \alpha$ and $\pi(\mathbf{M}) \Rightarrow \|p(B)\|_{\mathbf{M}} \geq \beta$. Therefore, $\pi(\mathbf{M}) \Rightarrow \min(\|p(A)\|_{\mathbf{M}}, \|p(B)\|_{\mathbf{M}}) \geq \min(\alpha, \beta)$ and thus, $\pi(\mathbf{M}) \Rightarrow \min(\mu_{m(A)}(i(p)), \mu_{m(B)}(i(p))) \geq \min(\alpha, \beta)$. Then, $\pi(\mathbf{M}) \Rightarrow \mu_{m(C)}(i(p)) \geq \min(\alpha, \beta)$, and thus, $\pi \models^{U,m}_{PGL^+} (p(C), \min(\alpha, \beta))$ as well.

UN: Given a PGL$^+$ context $\mathcal{M}_{U,m}$, we must prove, for each possibilistic model $\pi : \mathcal{M}_{U,m} \rightarrow [0,1]$, that if $\pi \models^{U,m}_{PGL^+} (p(A), \alpha)$, then $\pi \models^{U,m}_{PGL^+} (p(B), 1)$ whenever $\mu_{m(B)} \geq \max(1 - \alpha, \mu_{m(A)})$.

Assume that $\pi \models^{U,m}_{PGL^+} (p(A), \alpha)$. This means that, $N^*([p(A)] \mid \pi) \geq \alpha$, and thus, $\pi(\mathbf{M}) \leq \max(1-\alpha, \mu_{m(A)}(i(p)))$ for each PGL^+ interpretation $\mathbf{M} = (U, i, m) \in \mathcal{M}_{U,m}$. Therefore, $\pi(\mathbf{M}) \leq \mu_{m(B)}(i(p))$, and thus, $\pi \models^{U,m}_{PGL^+} (p(B), 1)$ as well.

■

As for the Horn-rule fragment of PGL, the PGL^+ proof method is defined by derivation but relative to the PGL^+ inference rules. Thus, given a PGL^+ context $\mathcal{M}_{U,m}$, a fact or rule φ is *deduced with a degree of deduction* α from a set of PGL^+ clauses P, denoted by $P \vdash^{U,m}_{PGL^+} (\varphi, \alpha)$, iff there exists a finite sequence of clauses $C_1, \ldots, C_m$ such that $C_m = (\varphi, \alpha)$ and, for each $i \in \{1, \ldots, m\}$, it holds that either $C_i \in P$, C_i is an instance of the triviality axiom or C_i is obtained by applying the RE, MP, FU, SU, IN or UN inference rules to previous clauses in the sequence.

Finally, the syntactic counterpart of maximum degree of possibilistic entailment is defined as follows. Given a PGL^+ context $\mathcal{M}_{U,m}$, *the maximum degree of deduction* of a PGL^+ goal $q(C)$ from a set of PGL^+ clauses P, denoted by $|q(C)|^{U,m}_P$, is the greatest $\alpha \in [0,1]$ for which there is a proof of $(q(C), \alpha)$ from P. Thus,

$$|q(C)|^{U,m}_P = \sup\{\alpha \in [0,1] \mid P \vdash^{U,m}_{PGL^+} (q(C), \alpha)\}.$$

Example 6.1 Let *around_16*, *between_14_16* and *between_16_18* be three fuzzy constants of sort `years_old`, and let *age_John* be a predicate symbol of type (`years_old`). Furthermore, let $\mathcal{M}_{U,m}$ be a PGL^+ context such that

1. $U = \{U_{\texttt{years\_old}} = [0, 120](\text{years})\}$;
2. $m(around\_16) = [13; 15; 17; 19]$,
   $m(between\_14\_16) = [12; 14; 16; 18]$, and
   $m(between\_16\_18) = [14; 16; 18; 20]$.

Now, given the set of PGL^+ clauses

$$P = \{\ (age\_John(between\_14\_16), 1),\ (age\_John(between\_16\_18), 1)\ \},$$

we have that

$$N^*(m(around\_16) \mid m(between\_14\_16)) = 0$$

and

$$N^*(m(around\_16) \mid m(between16\_18)) = 0,$$

and thus, applying the SU inference rule to the clauses of P we can only prove the PGL^+ goal $age\_John(around\_16)$ with a necessity degree of 0. However, each model $\pi : \mathcal{M}_{U,m} \to [0,1]$ of P verifies that

$$\pi(\mathbf{M}) \leq \min(\|age\_John(between\_14\_16)\|_{\mathbf{M}}, \|age\_John(between\_16\_18)\|_{\mathbf{M}}),$$

for each PGL$^+$ interpretation $\mathbf{M} = (U, i, m) \in \mathcal{M}_{U,m}$, and thus,

$$\pi(\mathbf{M}) \leq \min(\mu_{m(between\_14\_16)}(i(age\_John)), \mu_{m(between\_16\_18)}(i(age\_John))).$$

Therefore, as $\mu_{m(around\_16)}(u) \geq \min(\mu_{m(between\_14\_16)}(u), \mu_{m(between\_16\_18)}(u))$ for each $u \in [0, 120]$(years), we have that

$$\pi(\mathbf{M}) \leq \mu_{m(around\_16)}(i(age\_John))$$

for each PGL$^+$ interpretation $\mathbf{M} = (U, i, m) \in \mathcal{M}_{U,m}$. Hence,

$$P \models^{U,m}_{PGL^+} (age\_John(around_{16}), 1),$$

and thus, $\|age\_John(around\_16)\|^{U,m}_P = 1$. On the other hand, applying the IN inference rule to the unit clauses $(age\_John(between\_14\_16), 1)$ and $(age\_John(between\_16\_18), 1)$ we can conclude

$$(age\_John(around\_16), 1),$$

and thus, $|age\_John(around\_16)|^{U,m}_P = \|age\_John(around\_16)\|^{U,m}_P$. □

6.3 PGL$^+$ programs

As we have already mentioned, the PGL$^+$ proof method defined in the last section is complete for determining the maximum degree of possibilistic entailment of a PGL$^+$ goal in the frame of a particular class of sets of clauses or programs.

Definition 6.5 (PGL$^+$ program) *A PGL$^+$ program is a triple $\mathcal{P} = (P, U, m)$, where P is a finite set of PGL$^+$ clauses; U is a collection of non-empty domains; and m is an interpretation of object constants over $[0,1]^U$ such that for each object constant B appearing in P there exist $u, v \in U_\sigma$, σ being the sort of B, such that $\mu_{m(B)}(u) = 0$ and $\mu_{m(B)}(v) = 1$.*

Notice that U and m determine a particular PGL$^+$ context $\mathcal{M}_{U,m}$ in the sense of Section 6.2.1. Moreover, given a PGL$^+$ program $\mathcal{P} = (P, U, m)$ and a PGL$^+$ atomic formula $p(B)$ appearing in P, there exist at least two interpretations $\mathbf{M}_0, \mathbf{M}_1 \in \mathcal{M}_{U,m}$ such that $\|p(B)\|_{\mathbf{M}_0} = 0$ and $\|p(B)\|_{\mathbf{M}_1} = 1$.

Definition 6.6 (non-recursive program) *Let $\mathcal{P} = (P, U, m)$ be a PGL$^+$ program. We say that $\mathcal{P}$ is a non-recursive program if P does not contain recursive formulas. A recursive formula is of the form $p_1 \wedge \cdots \wedge p_k \wedge q(B) \rightarrow q(C)$, with $k \geq 0$, or is the result of combining two or more formulas of the form $s_1 \wedge \cdots \wedge s_m \wedge p(A) \rightarrow q(B)$ and $r_1 \wedge \cdots \wedge r_l \wedge q(C) \rightarrow p(D)$, with $m, l \geq 0$.*

Definition 6.7 (satisfiable program) *Let $\mathcal{P} = (P, U, m)$ be a PGL$^+$ program. We say that the set of clauses P is satisfiable in the context determined by U and m if there exists a normalized possibility distribution $\pi : \mathcal{M}_{U,m} \rightarrow [0,1]$ that satisfies all the clauses in P.*

The idea is that we restrict ourselves to non-recursive and satisfiable PGL$^+$ programs. Next we justify this choice.

On the one hand, given a program context $\mathcal{M}_{U,m}$, in the next section we prove that the sets of clauses $P = \{(p(A), \alpha), (p(A) \to q(B), \beta)\}$ and $P' = \{(q(B), \min(\beta, \alpha))\}$ are equivalent as far as we are interested in the entailment degree of a goal $q(C)$, i.e. $\|q(C)\|_P^{U,m} = \|q(C)\|_{P'}^{U,m}$. However, this intuitive behavior may be lost when we consider programs with recursive formulas, i.e.

$$\|q(C)\|^{U,m}_{\{(q(A),\alpha),(q(A)\to q(B),\beta)\}} = \|q(C)\|^{U,m}_{\{(q(A),\alpha),(q(B),\min(\beta,\alpha))\}}$$

iff either $\beta \leq \alpha$ or for each $u \in U_{\sigma_q}$ such that $\mu_{m(B)}(u) < 1 - \alpha$, (σ_q) being the type of q, it holds that $\mu_{m(A)}(u) \leq \mu_{m(B)}(u)$. Thus, let $\mathcal{M}_{U,m}$ be a program context and let A, B and B' be three object constants of sort σ_q such that, for each $u \in U_{\sigma_q}$,

$$\mu_{m(B')}(u) = \begin{cases} 1, & \text{if } \mu_{m(A)}(u) \leq \mu_{m(B)}(u) \\ \mu_{m(B)}(u), & \text{otherwise.} \end{cases}$$

Then, the PGL$^+$ clause $(q(A) \to q(B), \beta)$ is logically equivalent to the unit PGL$^+$ clause $(q(B'), \beta)$, and thus,

$$\|q(C)\|^{U,m}_{\{(q(A),\alpha),(q(A)\to q(B),\beta)\}} = \|q(C)\|^{U,m}_{\{(q(A),\alpha),(q(B'),\beta)\}}.$$

Moreover, for any object constant E of sort σ_q,

$$\|q(C)\|^{U,m}_{\{(q(E),\alpha),(q(A)\to q(B),\beta)\}} = \|q(C)\|^{U,m}_{\{(q(E),\alpha),(q(B'),\beta)\}},$$

and thus, the recursive clause $(q(A) \to q(B), \beta)$ does not depend on $\|q(A)\|^{U,m}_{\{(q(E),\alpha)\}}$. Let us see this result by means of an example which can be easily computed.

Example 6.2 Consider the set of PGL$^+$ clauses

$$\begin{aligned} P = \{\ & (age\_Mary(young) \wedge Mary\_studies(university) \to \\ & age\_Mary(between\_19\_26),\ 0.6), \\ & (age\_Mary(between\_35\_40), 1), \\ & (Mary\_studies(university), 1)\ \}, \end{aligned}$$

and the PGL$^+$ goal $age\_Mary(about\_40)$, where *young*, *between_19_26*, *between_35_40* and *about_40* are object constants of sort `years_old`. Further, consider a PGL$^+$ context $\mathcal{M}_{U,m}$ such that

1. $U = \{U_{\texttt{years\_old}} = [0, 120](\text{years})\}$;
2. $m(young) = [14; 17; 35; 40]$,
   $m(between\_19\_26) = [18; 19; 26; 27]$,
   $m(between\_35\_40) = [34; 35; 40; 41]$, and
   $m(about\_40) = [39; 40; 40; 41]$.

Now, if *extended_between_19_26* is an object constant of sort `years_old` and

$$\mu_{m(extended\_between\_19\_26)}(u) = \begin{cases} 1, & \text{if } u \leq 14 \text{ or } u \geq 40 \\ \mu_{m(between\_19\_26)}(u), & \text{otherwise,} \end{cases}$$

we have that

$$\|age\_Mary(about\_40)\|_P^{U,m} = \|age\_Mary(about\_40)\|_{P'}^{U,m},$$

where

$$\begin{aligned} P' = \{ & (Mary\_studies(university) \rightarrow \\ & age\_Mary(extended\_between\_19\_26),\ 0.6), \\ & (age\_Mary(between\_35\_40), 1), \\ & (Mary\_studies(university), 1) \}. \end{aligned}$$

And, in Section 6.4.2 we prove that

$$\|age\_Mary(about\_40)\|_{P'}^{U,m} = \|age\_Mary(about\_40)\|_{P''}^{U,m} = 0.6,$$

where

$$\begin{aligned} P'' = \{ & (age\_Mary(extended\_between\_19\_26), 0.6), \\ & (age\_Mary(between\_35\_40), 1) \}. \end{aligned}$$

What happens here is that, a recursive Horn-rule of the form $q(A) \wedge p \rightarrow q(B)$ is logically equivalent, in Gödel's logic, to the formula $p \rightarrow (q(A) \rightarrow q(B))$ and, in our framework, this is equivalent to $p \rightarrow p(B')$, where the fuzzy constant B' is point-wisely defined as

$$\mu_{m(B')}(u) = \mu_{m(A)}(u) \Rightarrow_G \mu_{m(B)}(u),$$

with $\Rightarrow_G$ being Gödel's many-valued implication, and thus, each recursive clause can be transformed into a non-recursive one. Then, as

$$\mu_{m(extended\_between\_19\_26)} = \mu_{m(young)} \Rightarrow_G \mu_{m(between\_19\_26)},$$

the recursive clause is logically equivalent to the clause

$$(Mary\_studies(university) \rightarrow age\_Mary(extended\_between\_19\_26), 0.6),$$

and thus, the fact that

$$\|age\_Mary(young)\|_{\{(age\_Mary(between\_35\_40),1)\}}^{U,m} = 0$$

is not considered for determining $\|age\_Mary(about\_40)\|_P^{U,m}$. However, our intention was to express that "Mary can be assumed to be between 19 and 26 years old (with a necessity ≥ 0.6) whenever we know (with a necessity ≥ 0.6) she is young and studies at the university", and thus, to be able to compute $\|age\_Mary(about\_40)\|_P^{U,m}$ solely from the set of clauses

$$P^* = \{(age\_Mary(between\_19\_26), 0),\ (age\_Mary(between\_35\_40), 1)\},$$

hence to obtain

$$\|age\_Mary(about\_40)\|_P^{U,m} = \|age\_Mary(about\_40)\|_{P^*}^{U,m} = 0.$$

□

Therefore, even though it would be possible to define an inference pattern for transforming recursive formulas into non-recursive ones based on the above result, the system user could be negatively surprised by some of the computations done by the system, as we have seen in the example.

On the other hand, satisfiable PGL$^+$ programs enjoy the following result.

Proposition 6.2 *Let $\mathcal{P} = (P, U, m)$ be a satisfiable PGL$^+$ program and let (φ, α), with $\alpha > 0$, be a PGL$^+$ clause. If P entails (φ, α) in the context determined by U and m, there exists at least an interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$ such that $\|\varphi\|_{\mathbf{M}} = 1$.*

Proof: On the one hand, if $\mathcal{P}$ is satisfiable there exists at least a normalized possibility distribution on the set of PGL$^+$ interpretations $\mathcal{M}_{U,m}$, say π_0, that satisfies all the clauses in P, and thus, there exists at least a PGL$^+$ interpretation in $\mathcal{M}_{U,m}$, say $\mathbf{M}_0$, such that $\pi_0(\mathbf{M}_0) = 1$. On the other hand, if $P \models^{U,m}_{PGL^+} (\varphi, \alpha)$ with $\alpha > 0$, we have that $\pi_0 \models^{U,m}_{PGL^+} (\varphi, \alpha)$, and thus, $\|\varphi\|_{\mathbf{M}_0} = 1$. ■

In contrast to the classical case, PGL$^+$ programs are not always satisfiable. Moreover, as we show in the next example, the satisfiability of a set of PGL$^+$ clauses depends on the interpretation of object constants.

Example 6.3 Let $\mathcal{P} = (P, U, m)$ be a PGL$^+$ program such that

$$P = \{\ (age\_Mary(about\_18) \rightarrow age\_Peter(around\_19), 0.7),\\ (age\_Peter(about\_18), 0.8),\\ (age\_Mary(about\_18), 0.9)\ \}$$

and $m(about\_18) = [17; 18; 18; 19]$(years). Then, for instance, P is satisfiable in the context determined by U and m if the object constant $around\_19$ is interpreted as the trapezoidal fuzzy set $[17; 18; 20; 21]$(years), but it is not if its interpretation is $[17; 19; 19; 21]$(years). □

Hence, in order to define a sound and coherent logic programming system, we restrict ourselves to non-recursive and satisfiable programs, simply refereed in the rest of the chapter as PGL$^+$ programs.

6.4 Completeness of the PGL$^+$ proof method

In this section, we describe and discuss two kinds of constraints we argue our PGL$^+$ programs must satisfy so that the modus ponens-style calculus extended with the semantical unification mechanism to be complete. First, we focus on what we call *modularity* constraint, which, in contrast to PLFC, we show it can be fulfilled by a pre-processing step of programs by means of the GR and FU inference rules. Then, we provide the completeness result for programs satisfying what we call *context* constraint.

For the sake of simplicity, from now on, we assume that given a collection of non-empty domains U and an interpretation m of object constants, for each

normalized fuzzy set $F : U_\sigma \to [0,1]$ there exists an object constant A of sort σ such that $m(A) = F$. However, only a numerable set of normalized fuzzy sets is needed.

6.4.1 Modularity constraint

The satisfaction of the modularity constraint by a PGL$^+$ program ensures that all (explicit and hidden) clauses of programs are considered. Indeed, since fuzzy constants are interpreted as (flexible) restrictions on an existential quantifier, PGL$^+$ atomic formulas clearly express disjunctive information. For instance, when $A = \{a_1, \ldots, a_n\}$, $p(A)$ is equivalent to the disjunction $p(a_1) \vee \cdots \vee p(a_n)$. Therefore, when parts of this (hidden) disjunctive information occur in the body of several program formulas we also have to consider all those new formulas that can be obtained through a completion process of the program which is based on the RE and FU inference rules. Let us briefly discuss this requirement by means of one example.

Example 6.4 Let $\mathcal{P} = (P, U, m)$ be a PGL$^+$ program such that

- $P = \{(p(A) \to q, 1), (p(B) \to r(C), 1), (r(C') \to q, 1), (p(D), 1)\}$;
- $m(A) = \{a_1, \ldots, a_n\}$,
  $m(B) = \{b_1, \ldots, b_m\}$,
  $m(C) = \{c_1, \ldots, c_k\}$,
  $m(C') = \{c_1, \ldots, c_k, c_{k+1}, \ldots, c_l\}$,
  $m(D) = \{a_1, b_1\}$, and
  $m(A \cup B) = m(A) \cup m(B)$.

We can easily check that the maximum degree with which q can be deduced from P by applying the MP and SU inference rules is 0. However, by the soundness of the RE and FU inferences rules, we have that

$$\{(p(B) \to r(C), 1), (r(C') \to q, 1)\} \models^{U,m}_{PGL^+} (p(B) \to q, 1)$$

and

$$\{(p(A) \to q, 1), (p(B) \to q, 1)\} \models^{U,m}_{PGL^+} (p(A \cup B) \to q, 1).$$

Hence, $(p(A \cup B) \to q, 1)$ can be seen as a valid (hidden) clause of P and can be obtained through the RE and FU inferences rules. When completing P with $(p(A \cup B) \to q, 1)$, the maximum degree with which q can be deduced by applying the MP and SU inference rules is 1. □

Therefore, the requirement of the modularity constraint of a PGL$^+$ program can be seen as a requirement of a completion process of the set of PGL$^+$ clauses which can be performed as a pre-processing step based on the RE and FU inference rules.

At this point we are ready to formalize the *modularity constraint* of a PGL$^+$ program.

Definition 6.8 (valid clause) *Let $\mathcal{P} = (P, U, m)$ be a PGL^+ program. We recursively define the set of valid clauses of P, denoted by P^+, in the following way:*

1. *All clauses of P belong to P^+.*
2. *If $(\varphi \rightarrow q(A), \alpha)$ and $(\psi \wedge q(B) \rightarrow r(C), \beta)$ are two clauses of P^+ such that $\mu_{m(A)} \leq \mu_{m(B)}$, then $(\varphi \wedge \psi \rightarrow r(C), \min(\alpha, \beta))$ is a clause of P^+ as well.*
3. *If $(p(A) \wedge \varphi \rightarrow q(D), \alpha)$ and $(p(B) \wedge \psi \rightarrow q(E), \beta)$ are two clauses of P^+ such that $\mu_{m(A)} \not\leq \mu_{m(B)}$ and $\mu_{m(B)} \not\leq \mu_{m(A)}$, and C and F are two object constants such that $\mu_{m(C)} = \max(\mu_{m(A)}, \mu_{m(B)})$ and $\mu_{m(F)} = \max(\mu_{m(D)}, \mu_{m(E)})$, then $(p(C) \wedge \varphi \wedge \psi \rightarrow q(F), \min(\alpha, \beta))$ is a clause of P^+ as well.*
4. *Only the clauses obtained by 1, 2 or 3 belong to P^+.*

Definition 6.9 (modularity constraint) *Let $\mathcal{P} = (P, U, m)$ be a PGL^+ program. P satisfies the modularity constraint if $P = P^+$.*

Definition 6.10 (basic clause) *Let $\mathcal{P} = (P, U, m)$ be a PGL^+ program. A clause $(\varphi, \alpha) \in P$ is a basic clause of P if $(P^+ \backslash \{(\varphi, \alpha)\})^+ = P^+ \backslash \{(\varphi, \alpha)\}$.*

The following propositions establish the relationship between a PGL^+ program and its set of valid clauses.

Proposition 6.3 *Let $\mathcal{P} = (P, U, m)$ be a PGL^+ program. If $(\varphi, \alpha) \in P^+$, then $P \vdash^{U,m}_{PGL^+} (\varphi, \alpha)$.*

Proof: For each clause $(\varphi, \alpha) \in P^+$, either $(\varphi, \alpha) \in P$ or there exists at least a finite sequence $C_1, C_2, \ldots C_m$ of valid clauses of P such that $C_m = (\varphi, \alpha)$, $C_1 \in P$, $C_2 \in P$ and, for each $i \in \{3, \ldots, m\}$, it holds that either $C_i \in P$ or C_i can be obtained by applying the RE or FU inference rules to previous clauses in the sequence. Hence, for each clause $(\varphi, \alpha) \in P^+$, we have that $P \vdash^{U,m}_{PGL^+} (\varphi, \alpha)$. ■

Proposition 6.4 *Let $\mathcal{P} = (P, U, m)$ be a PGL^+ program. Then,*

$$\|q(C)\|^{U,m}_{P} = \|q(C)\|^{U,m}_{P^+}.$$

Proof: On the one hand, by Proposition 6.3, $P \vdash^{U,m}_{PGL^+} (\varphi, \alpha)$ for each clause $(\varphi, \alpha) \in P^+$. Therefore, by the soundness of the inference rules, $P \models^{U,m}_{PGL^+} (\varphi, \alpha)$ for each clause $(\varphi, \alpha) \in P^+$. On the other hand, we have that $P \subseteq P^+$, and thus, $P^+ \models^{U,m}_{PGL^+} (\psi, \beta)$ for each clause $(\psi, \beta) \in P$. Hence, $\pi \models^{U,m}_{PGL^+} P$ iff $\pi \models^{U,m}_{PGL^+} P^+$ for each possibilistic model $\pi : \mathcal{M}_{U,m} \rightarrow [0, 1]$. Then, by Proposition 6.1, $\|q(C)\|^{U,m}_{P} = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P\} = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P^+\} = \|q(C)\|^{U,m}_{P^+}$. ■

Proposition 6.5 *Let $\mathcal{P} = (P, U, m)$ be a PGL$^+$ program. Then, for each predicate symbol q appearing in P in the head of a formula, there exists at least a clause $(\varphi, \alpha) \in P$ such that the head of φ is q and (φ, α) is a basic clause of P.*

Proof: As the predicate symbol q appears in P in the head of a formula, there exists at least a clause $(\varphi, \alpha) \in P$ such that the head of φ is q. If (φ, α) is a unit PGL$^+$ clause, obviously $(P^+ \backslash \{(\varphi, \alpha)\})^+ = P^+ \backslash \{(\varphi, \alpha)\}$, and thus, the unit PGL$^+$ clause is a basic clause of P. If (φ, α) is a general PGL$^+$ clause and it is not a basic clause of P, by Proposition 6.3, there exists at least a finite sequence $C_1, C_2, \ldots C_m$ of valid clauses of P such that $C_m = (\varphi, \alpha)$, $C_1 \in P \backslash \{(\varphi, \alpha)\}$, $C_2 \in P \backslash \{(\varphi, \alpha)\}$ and, for each $i \in \{3, \ldots, m\}$, it holds that either $C_i \in P \backslash \{(\varphi, \alpha)\}$ or C_i can be obtained by applying the RE or FU inference rules to previous clauses in the sequence. Therefore, as the head of φ is q, there exists at least a clause C_i, with $i \in \{1, \ldots, m-1\}$, such that $C_i = (\psi, \beta) \in P \backslash \{(\varphi, \alpha)\}$ and the head of ψ is q. Moreover, as P does not contain recursive formulas and the FU inference rule stretch the body of formulas, the clause (ψ, β) cannot be obtained by applying the RE or FU inference rules to the clause (φ, α). Now, if (ψ, β) is not a basic clause of P, we repeat the previous process over the clauses of $P \backslash \{(\varphi, \alpha), (\psi, \beta)\}$. Hence, as P is a finite set of PGL$^+$ clauses, we finally find some clause $(\phi, \delta) \in P$ such that the head of ϕ is q and (ϕ, δ) is a basic clause of P. ■

6.4.2 Context constraint

Now let us consider another kind of constraint we want our programs to satisfy. The idea is that in a PGL$^+$ program satisfying the so-called *context* constraint the use of the SU and MP inference is enough to attain a degree of deduction equal to the degree of possibilistic entailment. And for this we need that

(R1) the possibilistic entailment degree of a goal be univocally determined by those clauses in the program having the goal in their head or leading to one of these clauses by resolving them with other clauses, and that

(R2) the SU and MP rules work in a, we can say, locally complete way.

Next we argue the need for these requirements.

R1: As for the first one, the objective is to ensure that, given a PGL$^+$ program $\mathcal{P} = (P, U, m)$ and a PGL$^+$ goal $q(C)$, $\|q(C)\|_P^{U,m}$ can be determined only from the subset P_q of clauses (φ, α) in P for which either q is in the head of φ or q *depends*[4] on the head of φ. Roughly speaking, with this requirement one wants to avoid having formulas of the form $(q(A) \rightarrow t(\overline{B}), \alpha)$ and $(t(B), \beta)$ together in a program since, due to the disjunctive interpretation of fuzzy constants, we would have that a formula of the form $(q(\overline{A}), \delta)$ should be derivable, where $\overline{A}$

[4]We say that q depends on p in P, if P contains a set of clauses $\{(\varphi_1, \alpha_1), \ldots, (\varphi_k, \alpha_k)\}$, with $k \geq 1$, such that p appears in the body of φ_1, the head of φ_k is q, and the head of φ_i appears in the body of φ_{i+1}, with $i \in \{1, \ldots, k-1\}$.

and $\overline{B}$ denote the complement of A and B, respectively, and thus, we should enable a kind of *modus tollens* inference mechanism.

Given a program context $\mathcal{M}_{U,m}$, what happens here is that for any pair of object constants E and A of sort σ_q, and B and F of sort σ_t,

$$\|q(C)\|^{U,m}_{\{(q(E),\alpha),(q(A)\rightarrow t(B),\beta),(t(F),\gamma)\}} = \|q(C)\|^{U,m}_{\{(q(D),1))\}},$$

where D is an object constant of sort σ_q such that, for each $u \in U_{\sigma_q}$,

$$\mu_{m(D)}(u) = \begin{cases} \max(1-\alpha, \mu_{m(E)}(u)), & \text{if there exists } v \in U_{\sigma_t} \text{ such that} \\ & \max(1-\alpha, \mu_{m(E)}(u)) \leq \max(1-\gamma, \mu_{m(F)}(v)) \\ & \text{and either } \mu_{m(A)}(u) \leq \mu_{m(B)}(v) \text{ or} \\ & \max(1-\alpha, \mu_{m(E)}(u)) \leq \max(1-\beta, \mu_{m(B)}(v)) \\ \sup_{v \in U_{\sigma_t}} \min(\max(1-\gamma, \mu_{m(F)}(v)), & \\ \max(1-\beta, \mu_{m(A)}(u) \rightarrow \mu_{m(B)}(v))), & \text{otherwise.} \end{cases}$$

Hence,

$$\|q(C)\|^{U,m}_{\{(q(E),\alpha),(q(A)\rightarrow t(B),\beta),(t(F),\gamma)\}} = \|q(C)\|^{U,m}_{\{(q(E),\alpha)\}}$$

iff for each $u \in U_{\sigma_q}$

$$\begin{aligned} \max(1-\alpha, \mu_{m(E)}(u)) \;\leq\; & \sup_{v \in U_{\sigma_t}} \min(\max(1-\gamma, \mu_{m(F)}(v)), \\ & \max(1-\beta, \mu_{m(A)}(u) \rightarrow \mu_{m(B)}(v))). \end{aligned}$$

Let us see this result by means of an example which can be easily computed.

Example 6.5 Consider the following set of PGL$^+$ clauses:

$$\begin{aligned} P = \{\; & (age\_Mary(between\_18\_20) \rightarrow weight\_Mary(between\_50\_55), 1), \\ & (age\_Mary(between\_17\_20), 1), \\ & (weight\_Mary(49), 1) \;\}, \end{aligned}$$

the PGL$^+$ goal $age\_Mary(17)$, and a particular context $\mathcal{M}_{U,m}$ in which the object constants *between_17_20*, *between_18_20* and *between_50_55* are interpreted as the crisp intervals $[17, 20]$(years), $[18, 20]$(years) and $[50, 55]$(kilograms), respectively. Then, we have that

$$\|age\_Mary(17)\|^{U,m}_{P} = \|age\_Mary(17)\|^{U,m}_{\{(age\_Mary(17),1)\}} = 1.$$

However, in P there is no explicit information expressing that "Mary is 17 years old", and thus, $\|age\_Mary(17)\|^{U,m}_{P}$ should be determined just from $(age\_Mary(between\_17\_20), 1)$. □

Even from a theoretical point of view it would be interesting to extend the PGL$^+$ proof method with "some kind of modus tollens inference mechanism", the logic programming system would have some important computational limitations, besides of probably surprising an unaware user with some results computed by the system.

On the one hand, we would extend our current calculus with at least an inference pattern of the form

$$\frac{\begin{array}{l}(q(A_1) \rightarrow t(B_1), \alpha_1)\\ \cdots\cdots\\ (q(A_n) \rightarrow t(B_n), \alpha_n)\\ (t(E_1), \beta_1)\\ \cdots\cdots\\ (t(E_m), \beta_m)\end{array}}{(q(D), 1)}, \tag{6.1}$$

where, in each context $\mathcal{M}_{U,m}$, D should denote an object constant of sort σ_q such that, for each $u \in U_{\sigma_q}$,

$$\begin{array}{rl}\mu_{m(D)}(u) \geq & \sup_{v \in U_{\sigma_t}} \min(\max(1-\alpha_1, \mu_{m(A_1)}(u) \rightarrow \mu_{m(B_1)}(v)), \ldots,\\ & \max(1-\alpha_n, \mu_{m(A_n)}(u) \rightarrow \mu_{m(B_n)}(v)),\\ & \max(1-\beta_1, \mu_{m(E_1)}(v)), \ldots, \max(1-\beta_m, \mu_{m(E_m)}(v))),\end{array}$$

(σ_t) being the type of t.

On the other hand, the proof algorithm would determine $\|q(C)\|_P^{U,m}$ for each predicate symbol q appearing in P, by combining $\|q(C)\|_{P_q}^{U,m}$ and $\|q(C)\|_{P \setminus P_q}^{U,m}$. However, in some sense, the algorithm for determining $\|q(C)\|_{P \setminus P_q}^{U,m}$ should be based on the previously computed degree $\|q(C)\|_{P_q}^{U,m}$. Therefore, as far as we can see, neither the membership function of the fuzzy constant D of pattern 6.1, nor $\|q(C)\|_P^{U,m}$ should be easily computed remaining as an open question to be solved in the near future.

Hence, in order to define a complete and efficient proof procedure, we have to consider PGL$^+$ programs satisfying that $\|q(C)\|_P^{U,m} = \|q(C)\|_{P_q}^{U,m}$, for each predicate symbol q appearing in P. Thus, we have to define a *computable* constraint over the interpretation of object constants appearing in PGL$^+$ programs that enables us to ensure that, for each $u \in U_{\sigma_q}$,

$$\begin{array}{c}\sup\{\min\{\max(1-\alpha, \|\varphi\|_{\mathbf{M}}) \mid (\varphi, \alpha) \in P_q\} \mid\\ \mathbf{M} = (U, i, m) \in \mathcal{M}_{U,m} \text{ and } i(q) = u\}\\ \leq\\ \sup\{\min\{\max(1-\gamma, \|\phi\|_{\mathbf{M}}) \mid (\phi, \gamma) \in P\} \mid\\ \mathbf{M} = (U, i, m) \in \mathcal{M}_{U,m} \text{ and } i(q) = u\}.\end{array}$$

We say a computable constraint in the sense that it must be checked just from the previously computed information in the system, and thus, it cannot depend on $\|q(C)\|_{P \setminus P_q}^{U,m}$.

R2: As for the second requirement, it is indeed needed to avoid problems of weakening of the deduction power in simple modus ponens inference steps involving a unification process. For instance, given a context $\mathcal{M}_{U,m}$, in Propositions 6.6 and 6.7, we respectively prove that

$$\|p(A)\|_{\{(p(E),\alpha)\}}^{U,m} = \min(\alpha, \delta),$$

where $\delta = N^*(m(A) \mid m(E))$, and

$$\|q(C)\|^{U,m}_{\{(p(A),\min(\alpha,\delta)),(p(A)\to q(B),\beta)\}} = \|q(C)\|^{U,m}_{\{(q(B),\min(\alpha,\delta,\beta))\}}.$$

However, due to the necessity measure used for computing the partial matching between fuzzy constants, the expected equality

$$\|q(C)\|^{U,m}_{\{(p(E),\alpha),(p(A)\to q(B),\beta)\}} = \|q(C)\|_{\{(q(B),\min(\alpha,\delta,\beta))\}}$$

may not hold. Actually, it strongly depends on how m interprets each object constant. Indeed, in Proposition 6.8, we prove that the equality holds iff either $\min(\alpha,\beta) \leq \delta$ or, for some $v \in U_{\sigma_p}$, $\mu_{m(A)}(v) = 0$ and $\mu_{m(E)}(v) = 1-\delta$, (σ_p) being the type of p.

Proposition 6.6 *Let $\mathcal{P} = (\{(q(A),\alpha)\}, U, m)$ be a PGL$^+$ program and let D be an object constant of sort σ_q such that $\mu_{m(D)} = \max(1-\alpha, \mu_{m(A)})$. Then,*

$$\|q(C)\|^{U,m}_{\{(q(A),\alpha)\}} = \|q(C)\|^{U,m}_{\{(q(D),1)\}} = N^*(m(C) \mid m(D)).$$

Proof: On the one hand, a possibilistic model $\pi : \mathcal{M}_{U,m} \to [0,1]$ satisfies $(q(A),\alpha)$ iff $\pi(\mathbf{M}) \leq \max(1-\alpha, \|q(A)\|_{\mathbf{M}})$ for each PGL$^+$ interpretation $\mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m}$. Thus, iff $\pi(\mathbf{M}) \leq \max(1-\alpha, \mu_{m(A)}(i(q)))$, and thus, iff $\pi(\mathbf{M}) \leq \|q(D)\|_{\mathbf{M}}$. Then, $\pi \models^{U,m}_{PGL^+} (q(A),\alpha)$ iff $\pi \models^{U,m}_{PGL^+} (q(D),1)$. Hence, by Proposition 6.1, $\|q(C)\|^{U,m}_{\{(q(A),\alpha)\}} = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} (q(A),\alpha)\} = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} (q(D),1)\} = \|q(C)\|^{U,m}_{\{(q(D),1)\}}$.

On the other hand, by the soundness of the SU inference rule, we have that $\|q(C)\|^{U,m}_{\{(q(D),1)\}} \geq N^*(m(C) \mid m(D))$. Then, we must prove that $\|q(C)\|^{U,m}_{\{(q(D),1)\}} \leq N^*(m(C) \mid m(D))$.

Let $\pi : \mathcal{M}_{U,m} \to [0,1]$ be a possibility distribution with the following definition: $\pi(\mathbf{M}) = \|q(D)\|_{\mathbf{M}}$, for each PGL$^+$ interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$. As $\{(q(A),\alpha)\} \models^{U,m}_{PGL^+} \{(q(D),1)\}$, by Proposition 6.2, π is normalized and, obviously, $\pi \models^{U,m}_{PGL^+} (q(D),1)$. Finally, $N^*([q(C)] \mid \pi) = \inf_{u\in U_{\sigma_q}} \mu_{m(D)}(u) \Rightarrow \mu_{m(C)}(u) = N^*(m(C) \mid m(D))$. Hence, by Proposition 6.1, $\|q(C)\|^{U,m}_{\{(q(D),1)\}} = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} (q(D),1)\} \leq N^*(m(C) \mid m(D))$. ■

Proposition 6.7 *Let $\mathcal{P} = (\{(p(A),\alpha),(p(A) \to q(B),\beta)\}, U, m)$ be a PGL$^+$ program. Then, $\|q(C)\|^{U,m}_{\{(p(A),\alpha),(p(A)\to q(B),\beta)\}} = \|q(C)\|^{U,m}_{\{(q(B),\min(\alpha,\beta))\}}$.*

Proof: We define $P = \{(p(A),\alpha),(p(A) \to q(B),\beta)\}$.

By the soundness of the MP inference rule, $P \models^{U,m}_{PGL^+} \{(q(B),\min(\alpha,\beta))\}$. Therefore, by Proposition 6.1, $\|q(C)\|^{U,m}_P = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P\} \geq \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} (q(B),\min(\alpha,\beta))\} = \|q(C)\|^{U,m}_{\{(q(B),\min(\alpha,\beta))\}}$. Then, we must prove that $\|q(C)\|^{U,m}_P \leq \|q(C)\|^{U,m}_{\{(q(B),\min(\alpha,\beta))\}}$.

Let D be an object constant of sort σ_q such that

$$\mu_{m(D)} = \max(1 - \min(\alpha, \beta), \mu_{m(B)}).$$

Then, by Proposition 6.6, $\|q(C)\|^{U,m}_{\{(q(B),\min(\alpha,\beta))\}} = \|q(C)\|^{U,m}_{\{(q(D),1)\}} = N^*(m(C) \mid m(D))$. As $\|q(C)\|^{U,m}_P = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P\}$, we show that for some possibilistic model $\pi : \mathcal{M}_{U,m} \to [0,1]$ satisfying P, we have that $N^*([q(C)] \mid \pi) = N^*(m(C) \mid m(D))$, and thus, $\|q(C)\|^{U,m}_P \leq \|q(C)\|^{U,m}_{\{(q(D),1)\}}$. We distinguish two cases.

Case $\beta \leq \alpha$. Let $\pi : \mathcal{M}_{U,m} \to [0,1]$ be a possibility distribution with the following definition:

$$\pi(\mathbf{M}) = \begin{cases} \|q(D)\|_{\mathbf{M}}, & \text{if } \|p(A)\|_{\mathbf{M}} = 1 \\ 0, & \text{otherwise.} \end{cases}$$

Case $\alpha \leq \beta$. Let $\pi : \mathcal{M}_{U,m} \to [0,1]$ be a possibility distribution with the following definition:

$$\pi(\mathbf{M}) = \begin{cases} \|q(D)\|_{\mathbf{M}}, & \text{if either } \|p(A)\|_{\mathbf{M}} = 0 \text{ and } \|q(B)\|_{\mathbf{M}} \leq 1 - \alpha, \text{ or} \\ & \|p(A)\|_{\mathbf{M}} = 1 \text{ and } \|q(B)\|_{\mathbf{M}} \geq 1 - \alpha \\ 0, & \text{otherwise.} \end{cases}$$

As $\mathcal{P}$ is a non-recursive PGL$^+$ program, $p \neq q$ and there exists at least two PGL$^+$ interpretations $\mathbf{M}_0$ and $\mathbf{M}_1$ in $\mathcal{M}_{U,m}$ such that $\|p(A)\|_{\mathbf{M}_0} = 0$ and $\|p(A)\|_{\mathbf{M}_1} = 1$, and thus, in both cases we are considering each value $u \in U_{\sigma_q}$. On the other hand, by the soundness of the UN inference rule, $P \models^{U,m}_{PGL^+} (q(D), 1)$. Then, by Proposition 6.2, π normalized in both cases too. Finally, we can easily check that in both cases $\pi \models^{U,m}_{PGL^+} P$ and $N^*([q(C)] \mid \pi) = \inf_{u \in U_{\sigma_q}} \mu_{m(D)}(u) \Rightarrow \mu_{m(C)}(u) = N^*(m(C) \mid m(D))$.

■

Proposition 6.8 *Let* $\mathcal{P} = (P, U, m)$ *be a PGL$^+$ program with* $P = \{(p(E), \alpha), (p(A) \to q(B), \beta)\}$. *Then,*

$$\|q(C)\|^{U,m}_P = \|q(C)\|^{U,m}_{\{(p(A),\min(\alpha,N^*(m(A)|m(E)))),(p(A)\to q(B),\beta)\}}$$

iff either $\min(\alpha, \beta) \leq N^*(m(A) \mid m(E))$ *or, for some* $v \in U_{\sigma_p}$, $\mu_{m(A)}(v) = 0$ *and* $\mu_{m(E)}(v) = 1 - N^*(m(A) \mid m(E))$, (σ_p) *being the type of* p.

Proof: By Proposition 6.7, $\|q(C)\|^{U,m}_{\{(p(A),\min(\alpha,N^*(m(A)|m(E)))),(p(A)\to q(B),\beta)\}} = \|q(C)\|^{U,m}_{\{(q(B),\min(\alpha,N^*(m(A)|m(E)),\beta))\}}$. Then, we must prove that $\|q(C)\|^{U,m}_P = \|q(C)\|^{U,m}_{\{(q(B),\min(\alpha,N^*(m(A)|m(E)),\beta))\}}$ iff either $\min(\alpha, \beta) \leq N^*(m(A) \mid m(E))$ or, for some $v \in U_{\sigma_p}$, $\mu_{m(A)}(v) = 0$ and $\mu_{m(E)}(v) = 1 - N^*(m(A) \mid m(E))$.

By the soundness of the SU and MP inference rules,

$$\{(p(E),\alpha)\} \models^{U,m}_{PGL^+} (p(A), \min(\alpha, N^*(m(A) \mid m(E))))$$

and

$$\{(p(A), \min(\alpha, N^*(m(A) \mid m(E)))), (p(A) \to q(B), \beta)\} \models^{U,m}_{PGL^+} (q(B), \min(\alpha, N^*(m(A) \mid m(E)), \beta)).$$

Then, for each program context $\mathcal{M}_{U,m}$, $P \models^{U,m}_{PGL^+} (q(B), \min(\alpha, N^*(m(A) \mid m(E)), \beta))$. Therefore, by Proposition 6.1, $\|q(C)\|^{U,m}_P = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P\} \geq \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} (q(B), \min(\alpha, N^*(m(A) \mid m(E)), \beta))\} = \|q(C)\|^{U,m}_{\{(q(B),\min(\alpha,N^*(m(A)|m(E)),\beta))\}}$, and thus, for each program context $\mathcal{M}_{U,m}$, $\|q(C)\|^{U,m}_P \geq \|q(C)\|^{U,m}_{\{(q(B),\min(\alpha,N^*(m(A)|m(E)),\beta))\}}$. Hence, we must prove that $\|q(C)\|^{U,m}_P \leq \|q(C)\|^{U,m}_{\{(q(B),\min(\alpha,N^*(m(A)|m(E)),\beta))\}}$ iff either $\min(\alpha,\beta) \leq N^*(m(A) \mid m(E))$ or, for some $v \in U_{\sigma_p}$, $\mu_{m(A)}(v) = 0$ and $\mu_{m(E)}(v) = 1 - N^*(m(A) \mid m(E))$.

Let D be an object constant of sort σ_q such that

$$\mu_{m(D)} = \max(1 - \min(\alpha, N^*(m(A) \mid m(E)), \beta), \mu_{m(B)}).$$

By Proposition 6.6, $\|q(C)\|^{U,m}_{\{(q(B),\min(\alpha,N^*(m(A)|m(E)),\beta))\}} = \|q(C)\|^{U,m}_{\{(q(D),1)\}} = N^*(m(C) \mid m(D))$. Then, $\|q(C)\|^{U,m}_P \leq \|q(C)\|^{U,m}_{\{(q(D),1)\}}$ iff for each $u \in U_{\sigma_q}$ $\sup\{\min(\max(1-\alpha, \|p(E)\|_{\mathbf{M}}), \max(1-\beta, \|p(A) \to q(B)\|_{\mathbf{M}})) \mid \mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m}$ and $i(q) = u\} \geq \max(1 - \min(\alpha, N^*(m(A) \mid m(E)), \beta), \mu_{m(B)}(u))$.

If $\min(\alpha,\beta) \leq N^*(m(A) \mid m(E))$, we distinguish two cases.

Case $\beta \leq \alpha$. For each $u \in U_{\sigma_q}$, there exists at least a PGL$^+$ interpretation $\mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m}$ such that $i(q) = u$ and $\|p(E)\|_{\mathbf{M}} = 1$, and thus, for each $u \in U_{\sigma_q}$, $\min(\max(1-\alpha, \|p(E)\|_{\mathbf{M}}), \max(1-\beta, \|p(A) \to q(B)\|_{\mathbf{M}})) = \max(1-\beta, \|p(A) \to q(B)\|_{\mathbf{M}}) \geq \max(1-\beta, \mu_{m(B)}(u))$. Hence, if $\beta \leq \min(\alpha, N^*(m(A) \mid m(E)))$, then, for each $u \in U_{\sigma_q}$, $\sup\{\min(\max(1-\alpha, \|p(E)\|_{\mathbf{M}}), \max(1-\beta, \|p(A) \to q(B)\|_{\mathbf{M}})) \mid \mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m}$ and $i(q) = u\} \geq \max(1 - \min(\alpha, N^*(m(A) \mid m(E)), \beta), \mu_{m(B)}(u))$.

Case $\alpha \leq \beta$. For each $u \in U_{\sigma_q}$ such that $\mu_{m(B)}(u) \geq 1-\alpha$, there exists at least a PGL$^+$ interpretation $\mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m}$ such that $i(q) = u$ and $\|p(E)\|_{\mathbf{M}} = 1$, and thus, for each $u \in U_{\sigma_q}$ such that $\mu_{m(B)}(u) \geq 1-\alpha$, $\min(\max(1-\alpha, \|p(E)\|_{\mathbf{M}}), \max(1-\beta, \|p(A) \to q(B)\|_{\mathbf{M}})) = \max(1-\beta, \|p(A) \to q(B)\|_{\mathbf{M}}) \geq \max(1-\alpha, \mu_{m(B)}(u))$. And, for each $u \in U_{\sigma_q}$ such that $\mu_{m(B)}(u) \leq 1-\alpha$, there exists at least a PGL$^+$ interpretation $\mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m}$ such that $i(q) = u$ and $\|p(A)\|_{\mathbf{M}} = 0$, and thus, for each $u \in U_{\sigma_q}$ such that $\mu_{m(B)}(u) \leq 1-\alpha$, $\min(\max(1-\alpha, \|p(E)\|_{\mathbf{M}}), \max(1-\beta, \|p(A) \to q(B)\|_{\mathbf{M}})) = \max(1-\alpha, \|p(E)\|_{\mathbf{M}}) \geq \max(1-\alpha, \mu_{m(B)}(u))$. Hence, if $\alpha \leq \min(\beta, N^*(m(A) \mid m(E)))$, then,

for each $u \in U_{\sigma_q}$, $\sup\{\min(\max(1-\alpha, \|p(E)\|_{\mathbf{M}}), \max(1-\beta, \|p(A) \rightarrow q(B)\|_{\mathbf{M}})) \mid \mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m}$ and $i(q) = u\} \geq \max(1 - \min(\alpha, N^*(m(A) \mid m(E)), \beta), \mu_{m(B)}(u))$.

If $N^*(m(A) \mid m(E)) < \min(\alpha, \beta)$, for each $u \in U_{\sigma_q}$ such that $\mu_{m(B)}(u) \geq 1 - N^*(m(A) \mid m(E))$, there exists at least a PGL$^+$ interpretation $\mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m}$ such that $i(q) = u$ and $\|p(E)\|_{\mathbf{M}} = 1$, and thus, for each $u \in U_{\sigma_q}$ such that $\mu_{m(B)}(u) \geq 1 - N^*(m(A) \mid m(E))$, $\min(\max(1-\alpha, \|p(E)\|_{\mathbf{M}}), \max(1-\beta, \|p(A) \rightarrow q(B)\|_{\mathbf{M}})) = \max(1-\beta, \|p(A) \rightarrow q(B)\|_{\mathbf{M}}) \geq \max(1 - N^*(m(A) \mid m(E)), \mu_{m(B)}(u))$. Hence, if $N^*(m(A) \mid m(E)) < \min(\alpha, \beta)$, then, for each $u \in U_{\sigma_q}$ such that $\mu_{m(B)}(u) \geq 1 - N^*(m(A) \mid m(E))$, $\sup\{\min(\max(1-\alpha, \|p(E)\|_{\mathbf{M}}), \max(1-\beta, \|p(A) \rightarrow q(B)\|_{\mathbf{M}})) \mid \mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m}$ and $i(q) = u\} \geq \max(1 - \min(\alpha, N^*(m(A) \mid m(E)), \beta), \mu_{m(B)}(u))$.

Finally, for each $u \in U_{\sigma_q}$ such that $\mu_{m(B)}(u) < 1 - N^*(m(A) \mid m(E))$, we distinguish three cases.

1. For each PGL$^+$ interpretation $\mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m}$ such that $i(q) = u$ and $\|p(E)\|_{\mathbf{M}} > 1 - N^*(m(A) \mid m(E))$, we have that $\min(\max(1-\alpha, \|p(E)\|_{\mathbf{M}}), \max(1-\beta, \|p(A) \rightarrow q(B)\|_{\mathbf{M}})) = \max(1-\beta, \|q(B)\|_{\mathbf{M}})$.

2. For each PGL$^+$ interpretation $\mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m}$ such that $i(q) = u$ and $\|p(E)\|_{\mathbf{M}} \leq 1 - N^*(m(A) \mid m(E))$ and $\|p(A)\|_{\mathbf{M}} > \mu_{m(B)}(i(q))$, we have that $\min(\max(1-\alpha, \|p(E)\|_{\mathbf{M}}), \max(1-\beta, \|p(A) \rightarrow q(B)\|_{\mathbf{M}})) = \min(\max(1-\alpha, \|p(E)\|_{\mathbf{M}}), \max(1-\beta, \|q(B)\|_{\mathbf{M}}))$.

3. For each PGL$^+$ interpretation $\mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m}$ such that $i(q) = u$ and $\|p(E)\|_{\mathbf{M}} \leq 1 - N^*(m(A) \mid m(E))$ and $\|p(A)\|_{\mathbf{M}} \leq \mu_{m(B)}(i(q))$, we have that $\min(\max(1-\alpha, \|p(E)\|_{\mathbf{M}}), \max(1-\beta, \|p(A) \rightarrow q(B)\|_{\mathbf{M}})) = \max(1-\alpha, \|p(E)\|_{\mathbf{M}})$.

Therefore, as $1 - N^*(m(A) \mid m(E)) \geq \max(1-\alpha, 1-\beta)$ and, for some $u \in U_{\sigma_q}$ and $v \in U_{\sigma_p}$, $\mu_{m(B)}(u) = 0$ and $\mu_{m(A)}(v) = 0$, we have that, for each $u \in U_{\sigma_q}$ such that $\mu_{m(B)}(u) < 1 - N^*(m(A) \mid m(E))$, $\sup\{\min(\max(1-\alpha, \|p(E)\|_{\mathbf{M}}), \max(1-\beta, \|p(A) \rightarrow q(B)\|_{\mathbf{M}})) \mid \mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m}$ and $i(q) = u\} \geq \max(1 - \min(\alpha, N^*(m(A) \mid m(E)), \beta), \mu_{m(B)}(u))$, iff there exists at least a PGL$^+$ interpretation $\mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m}$, with $i(q) = u$, such that $\|p(E)\|_{\mathbf{M}} = 1 - N^*(m(A) \mid m(E))$ and $\|p(A)\|_{\mathbf{M}} = 0$.

Hence, we have proved that, for each $u \in U_{\sigma_q}$, $\sup\{\min(\max(1-\alpha, \|p(E)\|_{\mathbf{M}}), \max(1-\beta, \|p(A) \rightarrow q(B)\|_{\mathbf{M}})) \mid \mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m}$ and $i(q) = u\} \geq \max(1 - \min(\alpha, N^*(m(A) \mid m(E)), \beta), \mu_{m(B)}(u))$ iff either $\min(\alpha, \beta) \leq N^*(m(A) \mid m(E))$ or, for some $v \in U_{\sigma_p}$, $\mu_{m(A)}(v) = 0$ and $\mu_{m(E)}(v) = 1 - N^*(m(A) \mid m(E))$. ■

Given a program context $\mathcal{M}_{U,m}$, what happens here is that for any pair of object constants A and E of sort σ_p,

$$\|q(C)\|^{U,m}_{\{(p(E),\alpha),(p(A)\rightarrow q(B),\beta)\}} = \|q(C)\|^{U,m}_{\{(q(D),1)\}},$$

where D is an object constant of sort σ_q such that

- if $\min(\alpha,\beta) \leq N^*(m(A) \mid m(E))$, then, for each $u \in U_{\sigma_q}$,

$$\mu_{m(D)}(u) = \max(1 - \min(\alpha,\beta), \mu_{m(B)}(u)); \text{ and}$$

- if $\min(\alpha,\beta) > N^*(m(A) \mid m(E))$, then, for each $u \in U_{\sigma_q}$,

$$\mu_{m(D)}(u) = \begin{cases} \mu_{m(B)}(u), & \text{if } \mu_{m(B)}(u) \geq 1 - N^*(m(A) \mid m(E)) \\ 1 - N^*(m(A) \mid m(E)), & \text{if } V_{min} < \mu_{m(B)}(u) \text{ and} \\ & \quad \mu_{m(B)}(u) < 1 - N^*(m(A) \mid m(E)) \\ \max(\max(1-\beta, \mu_{m(B)}(u)), & \\ \sup\{\max(1-\alpha, \mu_{m(E)}(v)) \mid v \in U_{\sigma_p} \text{ and} & \\ \mu_{m(E)}(v) \leq 1 - N^*(m(A) \mid m(E)) \text{ and} & \\ \mu_{m(A)}(v) \leq \mu_{m(B)}(u)\}), & \text{if } \mu_{m(B)}(u) \leq V_{min}, \end{cases}$$

where

$$\begin{aligned} V_{min} \;=\; & \inf\{\mu_{m(A)}(v) \;\mid\; v \in U_{\sigma_p} \text{ and} \\ & \mu_{m(A)}(v) \leq 1 - N^*(m(A) \mid m(E)) \text{ and} \\ & \mu_{m(E)}(v) \geq 1 - N^*(m(A) \mid m(E))\}. \end{aligned}$$

Let us see this result by means of an example which can be easily computed.

Example 6.6 Let $\mathcal{P} = (P, U, m)$ be a PGL^+ program such that

- $P = \{ (age\_Mary(between\_19\_21) \rightarrow weight\_Mary(about\_50), 0.4),$
  $(age\_Mary(about\_20), 0.9) \};$
- $m(between\_19\_21) = [18; 19; 21; 22]$(years),
  $m(about\_20) = [15; 20; 20; 25]$(years), and
  $m(about\_50) = [45; 50; 50; 55]$(kilograms).

As $N^*(m(between\_19\_21) \mid m(about\_20)) = 0.25 < \min(0.9, 0.4)$, for any object constant C of sort `kilograms`,

$$\|weight\_Mary(C)\|_P^{U,m} = \|weight\_Mary(C)\|_{\{(weight\_Mary(D),1)\}}^{U,m},$$

where D is an object constant of sort `kilograms` such that, for each $u \in U_{\texttt{kilograms}}$,

$$\mu_{m(D)}(u) = \begin{cases} \mu_{m(about\_50)}(u), & \text{if } u \in [48, 52] \\ 0.6, & \text{otherwise.} \end{cases}$$

Then, by Proposition 6.6,

$$\begin{aligned} \|weight\_Mary(about\_50)\|_{\{(weight\_Mary(D),1)\}}^{U,m} \;&=\; N^*(m(about\_50) \mid m(D)) \\ &=\; 0.4, \end{aligned}$$

and thus, although $\|age\_Mary(between\_19\_21)\|_P^{U,m} < 0.4$, we get that $\|weight\_Mary(about\_50)\|_P^{U,m} = 0.4$. However, our basic intention is to express

that "Mary can be assumed to weigh about 50 kilos (with a necessity ≥ 0.4) whenever we know (with a necessity ≥ 0.4) she is between 19 and 21 years old".

Consider now the following set of PGL$^+$ clauses:

$$P = \{\ (age\_Mary(between\_19\_21) \rightarrow weight\_Mary(between\_49\_51), 0.4),\\ (age\_Mary(about\_20), 0.9)\ \},$$

with $m(between\_49\_51) = [48; 49; 51; 52]$(kilograms). In that case, for any object constant C of sort `kilograms`,

$$\|weight\_Mary(C)\|_P^{U,m} = \|weight\_Mary(C)\|_{\{(weight\_Mary(D),1)\}}^{U,m},$$

where D is an object constant of sort `kilograms` such that, for each $u \in U_{\texttt{kilograms}}$,

$$\mu_{m(D)}(u) = \begin{cases} \mu_{m(between\_49\_51)}(u), & \text{if } u \in [48.75, 51.25] \\ \frac{1}{5}u - 3, & \text{if } u \in [48, 48.75] \\ -\frac{1}{5}u + 5, & \text{if } u \in [51.25, 52] \\ 0.6, & \text{otherwise.} \end{cases}$$

Then, by Proposition 6.6,

$$\begin{aligned} \|weight\_Mary(between\_49\_51)\|_{\{(weight\_Mary(D),1)\}}^{U,m} &= N^*(m(between\_49\_51) \mid m(D)) \\ &= 0.25, \end{aligned}$$

and thus,

$$\begin{aligned} &\|weight\_Mary(between\_49\_51)\|_P^{U,m} \\ &\quad = \min(0.9, N^*(m(between\_19\_21) \mid m(about\_20)), 0.4) \\ &\quad = 0.25. \end{aligned}$$

But, for instance, if $around\_49\_51$ is an object constant of sort `kilograms` such that $m(around\_49\_51) = [45; 49; 51; 55]$, by Proposition 6.6, we have

$$\begin{aligned} \|weight\_Mary(around\_49\_51)\|_{\{(weight\_Mary(D),1)\}}^{U,m} &= N^*(m(around\_49\_51) \mid m(D)) \\ &= 0.4, \end{aligned}$$

and thus, although $\|weight\_Mary(between\_49\_51)\|_P^{U,m} < 0.4$, we get that $\|weight\_Mary(around\_49\_51)\|_P^{U,m} = 0.4$. However, our basic intention was to express that "Mary weighs C kilos (with a necessity ≥ 0.4) whenever we know (with a necessity ≥ 0.4) she weighs between 49 and 51 kilos" for each object constant C of sort `kilograms` such that $N^*(m(C) \mid m(between\_49\_51)) \geq 0.4$.

□

It would be possible to define an inference pattern for transforming PGL$^+$ clauses as we have done in Example 6.6. However, the logic programming system would have some important computational limitations. In fact, after each

resolution step, the membership function of the fuzzy constant in the resolvent clause should be recomputed from the interpretation of all fuzzy constants in the body of the resolved clause. Moreover, at this point, some new valid clauses should be considered since they could not be computed in a pre-processing step (see Section 6.5). Therefore, in order to define a complete and efficient proof procedure, we have to consider PGL$^+$ programs with well-behaved (in the above sense) interpretations of fuzzy constants.

In order to formalize the context constraint, we need the following results.

Proposition 6.9 *Let $\mathcal{P} = (P, U, m)$ be a PGL$^+$ program, let q be a predicate symbol of type (σ_q) appearing in P, and let $FC_P(q) = \{D$ object constant of sort $\sigma_q \mid P \models^{U,m}_{PGL^+} (q(D), 1)\}$. Further, denote by D_q the object constant of sort σ_q such that, for each $u \in U_{\sigma_q}$, $\mu_{m(D_q)}(u) = \inf\{\mu_{m(D)}(u) \mid D \in FC_P(q)\}$. Then, it holds that $P \models^{U,m}_{PGL^+} (q(D_q), 1)$. Therefore, we can safely write*

$$\mu_{m(D_q)} = \bigwedge \{\mu_{m(D)} \mid P \models^{U,m}_{PGL^+} (q(D), 1)\},$$

in the sense that $\bigwedge$ denotes a (point-wise) minimum.

Proof: For each object constant $D \in FC_P(q)$ we have $P \models^{U,m}_{PGL^+} (q(D), 1)$ iff $N^*([q(D)] \mid \pi) = 1$, for each possibilistic model $\pi : \mathcal{M}_{U,m} \to [0, 1]$ satisfying P, and thus, iff $\pi(\mathbf{M}) \leq \|q(D)\|_{\mathbf{M}}$ for each PGL$^+$ interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$. Then, we have that $\pi(\mathbf{M}) \leq \inf\{\|q(D)\|_{\mathbf{M}} \mid D \in FC_P(q)\}$ for each possibilistic model π satisfying P and PGL$^+$ interpretation $\mathbf{M}$. Hence, $N^*([q(D_q)] \mid \pi) = 1$ for each possibilistic model π satisfying P, and thus, $P \models^{U,m}_{PGL^+} (q(D_q), 1)$ as well. ■

Proposition 6.10 *Let $\mathcal{P} = (P, U, m)$ be a PGL$^+$ program. Then, using the above notation, $\|q(C)\|^{U,m}_P = \|q(C)\|^{U,m}_{\{(q(D_q),1)\}}$.*

Proof: We define $\alpha_1 = \|q(C)\|^{U,m}_P$ and $\alpha_2 = \|q(C)\|^{U,m}_{\{(q(D_q),1)\}}$.

1. $\alpha_1 \geq \alpha_2$: By Proposition 6.9 $P \models^{U,m}_{PGL^+} (q(D_q), 1)$ then, by Proposition 6.1, $\alpha_1 = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P\} \geq \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} (q(D_q), 1)\} = \alpha_2$.

2. $\alpha_1 \leq \alpha_2$: As $\alpha_1 = \|q(C)\|^{U,m}_P$, by Corollary 6.1, $P \models^{U,m}_{PGL^+} (q(C), \alpha_1)$. Let E be an object constant of sort σ_q such that $\mu_{m(E)} = \max(1 - \alpha_1, \mu_{m(C)})$. Then, by the soundness of the UN inference rule, $P \models^{U,m}_{PGL^+} (q(E), 1)$, and thus, $\mu_{m(D_q)} \leq \mu_{m(E)}$. On the other hand, by Proposition 6.6, $\|q(C)\|^{U,m}_{\{(q(D_q),1)\}} = N^*(m(C) \mid m(D_q))$. Hence, $\alpha_2 = N^*(m(C) \mid m(D_q)) = \inf_{u \in U_{\sigma_q}} \mu_{m(D_q)}(u) \Rightarrow \mu_{m(C)}(u) \geq \inf_{u \in U_{\sigma_q}} \mu_{m(E)}(u) \Rightarrow \mu_{m(C)}(u) = \inf_{u \in U_{\sigma_q}} \max(1 - \alpha_1, \mu_{m(C)}(u)) \Rightarrow \mu_{m(C)}(u) \geq \alpha_1$.

■

At this point we are ready to formalize the context constraint.

Definition 6.11 (context constraint) *Let $\mathcal{P} = (P, U, m)$ be a PGL$^+$ program and let $\mathbf{M}_{sat} = (U, i_{sat}, m)$ be a PGL$^+$ interpretation of $\mathcal{M}_{U,m}$ such that $\|\phi\|_{\mathbf{M}_{sat}} = 1$ for each clause $(\phi, \gamma) \in P$ with $\gamma > 0$. Further, for each predicate symbol q of type (σ_q) appearing in P, denote by P_q the set of clauses $\{(\varphi, \alpha)\} \subseteq P$ such that the head of φ is q or q depends on the head of φ in P; by P_q^+ the set of valid clauses of P_q; and by D_q the object constant of sort σ_q such that $\mu_{m(D_q)} = \bigwedge\{\mu_{m(D)} \mid P_q \models^{U,m}_{PGL^+} (q(D), 1)\}$. If $P_q = \emptyset$ for some predicate symbol q, we can safely define $P_q = \{(q(B), 0)\}$, B being an object constant of sort σ_q, and thus, $P_q^+ = \{(q(B), 0)\}$, and $\mu_{m(D_q)}(u) = 1$ for each $u \in U_{\sigma_q}$. Then, we say that $\mathcal{P}$ satisfies the context constraint if, for each predicate q appearing in P, it holds that*

(C1) *for each clause $(q(A) \rightarrow t(B), \beta) \in P \backslash P_q$, for each $u \in U_{\sigma_q}$, either $\mu_{m(A)}(u) \leq \mu_{m(B)}(i_{sat}(t))$ or $\mu_{m(D_q)}(u) \leq \max(1 - \beta, \mu_{m(B)}(i_{sat}(t)))$; and*

(C2) *for each clause $(p(E) \rightarrow q(F), \delta) \in P_q^+$ either $\delta \leq \|p(E)\|^{U,m}_{P_p}$ or, for some $v \in U_{\sigma_p}$, $\mu_{m(E)}(v) = 0$ and $\mu_{m(D_p)}(v) = 1 - \|p(E)\|^{U,m}_{P_p}$, (σ_p) being the type of p.*

Given a PGL$^+$ program $\mathcal{P} = (P, U, m)$, in the next section we prove that C1 ensures $\|q(C)\|^{U,m}_{P} = \|q(C)\|^{U,m}_{P_q}$ for each predicate symbol q appearing in P, and C2 ensures the degree of deduction obtained from P_q^+ by applying the SU and MP inference rules to be exactly the possibilistic entailment degree.

The most important feature of the context constraint is that, in each deduction step, it can be checked just from the previously computed information. In fact, in Section 6.5, we show that the proof algorithm for PGL$^+$ programs can be divided into four different and sequential steps. A satisfiability and recursivity testing step of PGL$^+$ programs. A pre-processing step, based on the RE and FU inference rules, which ensures the modularity constraint of non-recursive and satisfiable PGL$^+$ programs. A translation step, based on the MP, SU, UN and IN inference rules, which translates a PGL$^+$ program satisfying the modularity constraint into a semantically equivalent set of unit PGL$^+$ clauses, whenever the program satisfied the context constraint. And, finally, a deduction step, based on the SU inference rule, which computes the possibilistic entailment degree of a PGL$^+$ goal from the equivalent set of unit PGL$^+$ clauses. Therefore, from a computational point of view, the context constraint can be easily checked during the translation step.

But, the bad news is that the context constraint C1 is actually stronger than the requirement R1 listed above. Indeed, if a PGL$^+$ program $\mathcal{P} = (P, U, m)$ does not satisfy C1 for some predicate symbol q appearing in P, we cannot know whether $\|q(C)\|_P = \|q(C)\|_{P_q}$, and thus, in that case we should check whether, for each $u \in U_{\sigma_q}$,

$$\mu_{m(D_q)}(u) \leq \sup\{\min\{\max(1 - \gamma, \|\phi\|_{\mathbf{M}}) \mid (\phi, \gamma) \in P\} \mid \mathbf{M} = (U, i, m) \in \mathcal{M}_{U,m} \text{ and } i(q) = u\},$$

which, in turn is equivalent to determine whether

$$\mu_{m(D_q)} \leq \bigwedge \{\mu_{m(D)} \mid P \models^{U,m}_{PGL^+} (q(D), 1)\}.$$

Thus, to strongly ensure $\|q(C)\|^{U,m}_{P} = \|q(C)\|^{U,m}_{P_q}$ is equivalent to extend the PGL$^+$ proof method for determining $\|q(C)\|^{U,m}_{P \setminus P_q}$. Hence, our current context constraint is a useful approach for ensuring that $\|q(C)\|^{U,m}_{P}$ can be computed just from the clauses of P_q, and allowing us to define an efficient proof algorithm.

Finally, as we only consider non-recursive programs, for each PGL$^+$ program $\mathcal{P} = (P, U, m)$ there exists at least a predicate symbol q appearing in P such that q does not depend on no other predicate symbol in P. If $\mathcal{P}$ satisfies the context constraint C1 for q, the proof algorithm determines $\|q(C)\|^{U,m}_{P}$; otherwise we cannot ensure the computed deduction degree to be the possibilistic entailment degree. And, for each other predicate symbol t depending on q in P, if $\mathcal{P}$ satisfies C1 and C2 for q and t, the proof algorithm determines $\|t\|^{U,m}_{P}$. Thus, for each PGL$^+$ goal the proof algorithm can determine if the computed degree of deduction is in fact the maximum degree of possibilistic entailment.

6.4.3 The completeness result

In this section, we prove that the PGL$^+$ proof method is complete for determining the maximum degree of possibilistic entailment of a PGL$^+$ goal from a program satisfying the context constraint. In the following, we assume that PGL$^+$ programs satisfy this requirement.

The following proposition establishes that the degree of deduction obtained by applying the UN and IN inference rules is the greatest lower bound of possibilistic entailment.

Proposition 6.11 *Let $\mathcal{P} = (\{(q(A), \alpha), (q(B), \beta)\}, U, m)$ be a PGL$^+$ program and let D be an object constant of sort σ_q such that*

$$\mu_{m(D)} = \min(\max(1 - \alpha, \mu_{m(A)}), \max(1 - \beta, \mu_{m(B)})).$$

Then, $\|q(C)\|^{U,m}_{\{(q(A),\alpha),(q(B),\beta)\}} = \|q(C)\|^{U,m}_{\{(q(D),1)\}}$.

Proof: We define $P = \{(q(A), \alpha), (q(B), \beta)\}$.

Then, $\pi : \mathcal{M}_{U,m} \to [0, 1]$ is a model of P iff $\pi(\mathbf{M}) \leq \max(1-\alpha, \|q(A)\|_{\mathbf{M}})$ and $\pi(\mathbf{M}) \leq \max(1 - \beta, \|q(B)\|_{\mathbf{M}})$, for each PGL$^+$ interpretation $\mathbf{M} = (U, i, m) \in \mathcal{M}_{U,m}$. Thus, iff $\pi(\mathbf{M}) \leq \min(\max(1-\alpha, \mu_{m(A)}(i(q))), \max(1-\beta, \mu_{m(B)}(i(q))))$, and thus, iff $\pi(\mathbf{M}) \leq \|q(D)\|_{\mathbf{M}}$. Then, $\pi \models^{U,m}_{PGL^+} P$ iff $\pi \models^{U,m}_{PGL^+} (q(D), 1)$. Hence, by Proposition 6.1, $\|q(C)\|^{U,m}_{P} = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P\} = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} (q(D), 1)\} = \|q(C)\|^{U,m}_{\{(q(D),1)\}}$. ■

The following proposition establishes that, when considering a program satisfying the context constraint, the possibilistic entailment degree of a PGL$^+$ goal $q(C)$ can be determined just from the set of program clauses such that their heads are q or q depends on their heads.

Proposition 6.12 *Let $\mathcal{P} = (P, U, m)$ be a PGL$^+$ program, let q be a predicate symbol of type (σ_q), and let $P_q = \{(\varphi, \alpha) \in P \mid head(\varphi) = q$ or q depends on $head(\varphi)$ in $P\}$ or, if there is no clause in P with head q, let $P_q = \{(q(B), 0)\}$, B being some object constant of sort σ_q. Then,*

$$\|q(C)\|_P^{U,m} = \|q(C)\|_{P_q}^{U,m}.$$

Proof: As we consider satisfiable PGL$^+$ programs, if the predicate symbol q does not appear in P, for each $u \in U_{\sigma_q}$ there exists at least a PGL$^+$ interpretation $\mathbf{M} = (U, i, m) \in \mathcal{M}_{U,m}$ such that $i(q) = u$, and $\max(1 - \gamma, \|\phi\|_{\mathbf{M}}) = 1$ for each clause $(\phi, \gamma) \in P$. Hence, if the predicate symbol q does not appear in P, $\|q(C)\|_P^{U,m} = \|q(C)\|_{\{(q(B),0)\}}^{U,m}$ for any object constant B of sort σ_q.

As $P \models_{PGL^+}^{U,m} (q(B), 0)$ for any object constant B, by Proposition 6.1, we have that $\|q(C)\|_P^{U,m} = \|q(C)\|_{P \cup \{(q(B),0)\}}^{U,m}$. Therefore, if the predicate symbol q appears in P and $P_q = \emptyset$, we can safely define P_q as $\{(q(B), 0)\}$ for any object constant B of sort σ_q.

Then, obviously, $\pi \models_{PGL^+}^{U,m} P_q$ for each model $\pi : \mathcal{M}_{U,m} \to [0, 1]$ of P. Hence, by Proposition 6.1, $\|q(C)\|_P^{U,m} = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models_{PGL^+}^{U,m} P\} \geq \inf\{N^*([q(C)] \mid \pi) \mid \pi \models_{PGL^+}^{U,m} P_q\} = \|q(C)\|_{P_q}^{U,m}$. Thus, we must prove that $\|q(C)\|_P^{U,m} \leq \|q(C)\|_{P_q}^{U,m}$ for each predicate symbol q appearing in P.

Let D_q be an object constant of sort σ_q such that

$$\mu_{m(D_q)} = \bigwedge\{\mu_{m(D)} \mid P_q \models_{PGL^+}^{U,m} (q(D), 1)\}.$$

By Proposition 6.10 and 6.6, $\|q(C)\|_{P_q}^{U,m} = \|q(C)\|_{\{(q(D_q),1)\}}^{U,m} = N^*(m(C) \mid m(D_q))$. Then, defining $P_1 = P \backslash P_q$, we must prove that, for each $u \in U_{\sigma_q}$, $\mu_{m(D_q)}(u) \leq \sup\{\min(\min\{\max(1 - \alpha, \|\varphi\|_{\mathbf{M}}) \mid (\varphi, \alpha) \in P_q\}, \min\{\max(1 - \beta, \|\psi\|_{\mathbf{M}}) \mid (\psi, \beta) \in P_1\}) \mid \mathbf{M} = (U, i, m) \in \mathcal{M}_{U,m}$ and $i(q) = u\}$.

By Proposition 6.9, $P_q \models_{PGL^+}^{U,m} (q(D_q), 1)$, and thus, for each $u \in U_{\sigma_q}$ $\mu_{m(D_q)}(u) \geq \sup\{\min\{\max(1 - \alpha, \|\varphi\|_{\mathbf{M}}) \mid (\varphi, \alpha) \in P_q\} \mid \mathbf{M} = (U, i, m) \in \mathcal{M}_{U,m}$ and $i(q) = u\}$. Moreover, as $\mu_{m(D_q)} = \bigwedge\{\mu_{m(D)} \mid P_q \models_{PGL^+}^{U,m} (q(D), 1)\}$, for each $u \in U_{\sigma_q}$, $\mu_{m(D_q)}(u) = \sup\{\min\{\max(1 - \alpha, \|\varphi\|_{\mathbf{M}}) \mid (\varphi, \alpha) \in P_q\} \mid \mathbf{M} = (U, i, m) \in \mathcal{M}_{U,m}$ and $i(q) = u\}$, and thus, for each $u \in U_{\sigma_q}$ there exists at least a PGL$^+$ interpretation $\mathbf{M}_{q_u} = (U, i_{q_u}, m) \in \mathcal{M}_{U,m}$ such that $i_{q_u}(q) = u$ and $\mu_{m(D_q)}(u) = \min\{\max(1 - \alpha, \|\varphi\|_{\mathbf{M}_{q_u}}) \mid (\varphi, \alpha) \in P_q\} = \sup\{\min\{\max(1 - \alpha, \|\varphi\|_{\mathbf{M}}) \mid (\varphi, \alpha) \in P_q\} \mid \mathbf{M} = (U, i, m) \in \mathcal{M}_{U,m}$ and $i(q) = u\}$.

On the other hand, as P is satisfiable in the context determined by U and m, by Proposition 6.2 there exists at least a PGL$^+$ interpretation $\mathbf{M}_{\text{sat}} = (U, i_{\text{sat}}, m) \in \mathcal{M}_{U,m}$ such that $\|\phi\|_{\mathbf{M}_{\text{sat}}} = 1$, for each clause $(\phi, \gamma) \in P$ with $\gamma > 0$. Now, for each $u \in U_{\sigma_q}$, let $\mathbf{M}_u = (U, i_u, m)$ be a PGL$^+$ interpretation of $\mathcal{M}_{U,m}$ such that

$$i_u(p) = \begin{cases} i_{q_u}(p), & \text{if } p \text{ appears in } P_q \\ i_{\text{sat}}(p), & \text{otherwise.} \end{cases}$$

Then, for each unit PGL$^+$ clause $(s(B),\beta) \in P_1$, we have that, for each $u \in U_{\sigma_q}$, $\max(1-\beta, \|s(B)\|_{\mathbf{M}_u}) = 1$. And, for each clause $(r(A) \to s(B),\beta) \in P_1$ such that r does not appear in P_q, we have that, for each $u \in U_{\sigma_q}$, $\max(1-\beta, \|r(A) \to s(B)\|_{\mathbf{M}_u})) = 1$. If r appears in P_q, we distinguish two cases.

Case $r = q$. As $\mathcal{P}$ satisfies the context constraint, for each $u \in U_{\sigma_q}$, either $\mu_{m(A)}(u) \leq \|s(B)\|_{\mathbf{M}_{\mathrm{sat}}}$ or $\mu_{m(D_q)}(u) \leq \max(1-\beta, \|s(B)\|_{\mathbf{M}_{\mathrm{sat}}})$. Hence, for each $u \in U_{\sigma_q}$, $\mu_{m(D_q)}(u) \leq \max(1-\beta, \|q(A) \to s(B)\|_{\mathbf{M}_u})$.

Case $r \neq q$. Let

$$P_r = \{(\phi,\delta) \in P_q \mid head(\phi) = r \text{ or } r \text{ depends on head}(\phi) \text{ in } P_q\}$$

and let D_r be an object constant of sort σ_r such that

$$\mu_{m(D_r)} = \bigwedge\{\mu_{m(D)} \mid P_r \models^{U,m}_{PGL^+} (r(D),1)\}.$$

Again, as $\mathcal{P}$ satisfies the context constraint, for each $v \in U_{\sigma_r}$, either $\mu_{m(A)}(v) \leq \|s(B)\|_{\mathbf{M}_{\mathrm{sat}}}$ or $\mu_{m(D_r)}(v) \leq \max(1-\beta, \|s(B)\|_{\mathbf{M}_{\mathrm{sat}}})$. Therefore, for each $v \in U_{\sigma_r}$, $\mu_{m(D_r)}(v) \leq \max(1-\beta, \mu_{m(A)}(v) \to \|s(B)\|_{\mathbf{M}_{\mathrm{sat}}})$, and thus, for each $u \in U_{\sigma_q}$, $\mu_{m(D_r)}(i_u(r)) \leq \max(1-\beta, \|r(A) \to s(B)\|_{\mathbf{M}_u})$. Finally, as P does not contain recursive formulas and q depends on r in P, for each $u \in U_{\sigma_q}$, $\mu_{m(D_q)}(u) \leq \mu_{m(D_r)}(i_u(r))$. Hence, for each $u \in U_{\sigma_q}$, $\mu_{m(D_q)}(u) \leq \max(1-\beta, \|r(A) \to s(B)\|_{\mathbf{M}_u})$.

Therefore, for each $u \in U_{\sigma_q}$, $\mu_{m(D_q)}(u) = \min\{\max(1-\alpha, \|\varphi\|_{\mathbf{M}_{qu}}) \mid (\varphi,\alpha) \in P_q\} = \min\{\max(1-\alpha, \|\varphi\|_{\mathbf{M}_u}) \mid (\varphi,\alpha) \in P_q\}$ and $\mu_{m(D_q)}(u) \leq \min\{\max(1-\beta, \|\psi\|_{\mathbf{M}_u}) \mid (\psi,\beta) \in P_1\}$. Hence, for each $u \in U_{\sigma_q}$, $\mu_{m(D_q)}(u) \leq \min(\min\{\max(1-\alpha, \|\varphi\|_{\mathbf{M}_u}) \mid (\varphi,\alpha) \in P_q\}, \min\{\max(1-\beta, \|\psi\|_{\mathbf{M}}) \mid (\psi,\beta) \in P_1\})$. Thus, for each $u \in U_{\sigma_q}$, $\mu_{m(D_q)}(u) \leq \sup\{\min(\min\{\max(1-\alpha, \|\varphi\|_{\mathbf{M}}) \mid (\varphi,\alpha) \in P_q\}, \min\{\max(1-\beta, \|\psi\|_{\mathbf{M}}) \mid (\psi,\beta) \in P_1\}) \mid \mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m} \text{ and } i(q) = u\}$. ■

Given a PGL$^+$ program $\mathcal{P} = (P,U,m)$, in the following, for each predicate symbol q of type (σ_q), we denote by

$$P_q = \{(\varphi,\alpha) \in P \mid head(\varphi) = q \text{ or } q \text{ depends on head}(\varphi) \text{ in } P\}$$

or, if there is no clause in P with head q, $P_q = \{(q(B),0)\}$, B being some object constant of sort σ_q; by P_q^+ the set of valid clauses of P_q; and by D_q the object constant of sort σ_q such that

$$\mu_{m(D_q)} = \bigwedge\{\mu_{m(D)} \mid P_q \models^{U,m}_{PGL^+} (q(D),1)\}.$$

In order to prove that the SU and MP rules work in a "locally complete way" for programs satisfying both the modularity and the context constraints, we need the following previous results.

Proposition 6.13 *Let $\mathcal{P} = (P, U, m)$ be a PGL$^+$ program and let $(r(E), \alpha)$ and $(p(A) \rightarrow q(B), \beta)$ be two valid clauses of P_q. Then,* $\|q(C)\|^{U,m}_{\{(r(E),\alpha),(p(D_p),1),(p(A)\rightarrow q(B),\beta)\}} = \|q(C)\|^{U,m}_{\{(r(E),\alpha),(q(B),\min(\beta,\|p(A)\|^{U,m}_{P_p}))\}}$.

Proof: We define $P_1 = \{(r(E), \alpha), (p(D_p), 1), (p(A) \rightarrow q(B), \beta)\}$ and $P_2 = \{(r(E), \alpha), (q(B), \min(\beta, \|p(A)\|^{U,m}_{P_p}))\}$.

By Propositions 6.10 and 6.6, $\|p(A)\|^{U,m}_{P_p} = \|p(A)\|^{U,m}_{\{(p(D_p),1)\}} = N^*(m(A) \mid m(D_p))$. Then, by the soundness of the SU and MP inference rules, $P_1 \models^{U,m}_{PGL^+} P_2$, and thus, $\pi \models^{U,m}_{PGL^+} P_2$ for each model $\pi : \mathcal{M}_{U,m} \rightarrow [0,1]$ of P_1. Therefore, by Proposition 6.1, $\|q(C)\|^{U,m}_{P_1} = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P_1\} \geq \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P_2\} = \|q(C)\|^{U,m}_{P_2}$. Then, we must prove that $\|q(C)\|^{U,m}_{P_1} \leq \|q(C)\|^{U,m}_{P_2}$. We distinguish two cases.

Case $r = q$. Let D be an object constant of sort σ_q such that

$$\mu_{m(D)} = \min(\max(1-\alpha, \mu_{m(E)}), \max(1 - \min(\beta, \|p(A)\|^{U,m}_{P_p}), \mu_{m(B)})).$$

By Proposition 6.11, $\|q(C)\|^{U,m}_{P_2} = \|q(C)\|^{U,m}_{\{(q(D),1)\}}$ and, by Proposition 6.6, $\|q(C)\|^{U,m}_{\{(q(D),1)\}} = N^*(m(C) \mid m(D))$.

On the other hand, as $\mathcal{P}$ is a PGL$^+$ program satisfying the context constraint and $(p(A) \rightarrow q(B), \beta) \in P^+_q$, either $\beta \leq \|p(A)\|^{U,m}_{P_p}$ or, for some $u \in U_{\sigma_p}$, $\mu_{m(A)}(u) = 0$ and $\mu_{m(D_p)}(u) = 1 - \|p(A)\|^{U,m}_{P_p}$. Then, we distinguish two cases.

Case $\beta \leq \|p(A)\|^{U,m}_{P_p}$. Let $\pi : \mathcal{M}_{U,m} \rightarrow [0,1]$ be a possibility distribution with the following definition:

$$\pi(\mathbf{M}) = \begin{cases} \|q(D)\|_{\mathbf{M}}, & \text{if } \|p(D_p)\|_{\mathbf{M}} = 1 \\ 0, & \text{otherwise.} \end{cases}$$

Case $\beta > \|p(A)\|^{U,m}_{P_p}$. Let $\pi : \mathcal{M}_{U,m} \rightarrow [0,1]$ be a possibility distribution with the following definition:

$$\pi(\mathbf{M}) = \begin{cases} \|q(D)\|_{\mathbf{M}}, & \text{if either } \|p(D_p)\|_{\mathbf{M}} = 1 \text{ and} \\ & \|q(B)\|_{\mathbf{M}} \geq 1 - \|p(A)\|^{U,m}_{P_p}, \text{ or} \\ & \|p(D_p)\|_{\mathbf{M}} = 1 - \|p(A)\|^{U,m}_{P_p} \text{ and} \\ & \|p(A)\|_{\mathbf{M}} = 0 \text{ and } \|q(B)\|_{\mathbf{M}} < 1 - \|p(A)\|^{U,m}_{P_p} \\ 0, & \text{otherwise.} \end{cases}$$

By Proposition 6.9, $P_p \models^{U,m}_{PGL^+} (p(D_p), 1)$, and thus, $P \models^{U,m}_{PGL^+} (p(D_p), 1)$. Then, by Proposition 6.2, there exists at least a PGL$^+$ interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$ such that $\|p(D_p)\|_{\mathbf{M}} = 1$. Moreover, as $\mathcal{P}$ is a PGL$^+$ program satisfying the context constraint, if $\beta > \|p(A)\|^{U,m}_{P_p}$, there exists at least a

PGL$^+$ interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$ such that $\|p(D_p)\|_{\mathbf{M}} = 1 - \|p(A)\|^{U,m}_{P_p}$ and $\|p(A)\|_{\mathbf{M}} = 0$. Then, as P does not contain recursive formulas, in both cases we are considering each value $u \in U_{\sigma_q}$.

On the other hand, as $(p(A) \rightarrow q(B), \beta) \in P^+_q$, by the soundness of the RE and FU inference rules, $P \models^{U,m}_{PGL^+} (p(A) \rightarrow q(B), \beta)$ and, as $(q(E), \alpha) \in P_q$, it must be that $(q(E), \alpha) \in P$. Hence, $P \models^{U,m}_{PGL^+} P_1$, and thus, $P \models^{U,m}_{PGL^+} P_2$. And, by the soundness of the UN and IN inference rules, $P \models^{U,m}_{PGL^+} (q(D), 1)$. Then, by Proposition 6.2, π is normalized in both cases. Finally, we can easily check that $\pi \models^{U,m}_{PGL^+} P_1$ and $N^*([q(C)] \mid \pi) = \inf_{u \in U_{\sigma_q}} \mu_{m(D)}(u) \Rightarrow \mu_{m(C)}(u) = N^*(m(C) \mid m(D))$, in both cases too. Hence, by Proposition 6.1, $\|q(C)\|^{U,m}_{P_1} = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P_1\} \leq N^*(m(C) \mid m(D)) = \|q(C)\|^{U,m}_{P_2}$.

Case $r \neq q$. As P_2 only contains unit PGL$^+$ clauses, (P_2, U, m) satisfies the context constraint. Then, by Proposition 6.12, $\|q(C)\|^{U,m}_{P_2} = \|q(C)\|^{U,m}_{\{(q(B), \min(\beta, \|p(A)\|^{U,m}_{P_p}))\}}$. Let D be an object constant of sort σ_q such that

$$\mu_{m(D)} = \max(1 - \min(\beta, \|p(A)\|^{U,m}_{P_p}), \mu_{m(B)}).$$

By Proposition 6.6, $\|q(C)\|^{U,m}_{\{(q(B), \min(\beta, \|p(A)\|^{U,m}_{P_p}))\}} = \|q(C)\|^{U,m}_{\{(q(D),1)\}} = N^*(m(C) \mid m(D))$. As above, taking into account that $\mathcal{P}$ is a PGL$^+$ program satisfying the context constraint and $(p(A) \rightarrow q(B), \beta) \in P^+$, we distinguish two cases.

Case $\beta \leq \|p(A)\|^{U,m}_{P_p}$. Let $\pi : \mathcal{M}_{U,m} \rightarrow [0,1]$ be a possibility distribution with the following definition:

$$\pi(\mathbf{M}) = \begin{cases} \|q(D)\|_{\mathbf{M}}, & \text{if } \|p(D_p)\|_{\mathbf{M}} = 1 \text{ and } \|r(E)\|_{\mathbf{M}} = 1 \\ 0, & \text{otherwise.} \end{cases}$$

Case $\beta > \|p(A)\|^{U,m}_{P_p}$. Let $\pi : \mathcal{M}_{U,m} \rightarrow [0,1]$ be a possibility distribution with the following definition:

$$\pi(\mathbf{M}) = \begin{cases} \|q(D)\|_{\mathbf{M}}, & \text{if either } \|p(D_p)\|_{\mathbf{M}} = 1 \text{ and} \\ & \|r(E)\|_{\mathbf{M}} = 1 \text{ and} \\ & \|q(B)\|_{\mathbf{M}} \geq 1 - \|p(A)\|^{U,m}_{P_p}, \text{ or} \\ & \|p(D_p)\|_{\mathbf{M}} = 1 - \|p(A)\|^{U,m}_{P_p} \text{ and} \\ & \|p(A)\|_{\mathbf{M}} = 0 \text{ and} \\ & \|q(B)\|_{\mathbf{M}} < 1 - \|p(A)\|^{U,m}_{P_p} \text{ and} \\ & \max(1 - \alpha, \|r(E)\|_{\mathbf{M}}) \geq 1 - \|p(A)\|^{U,m}_{P_p} \\ 0, & \text{otherwise.} \end{cases}$$

As $P \models^{U,m}_{PGL^+} (p(D_p), 1)$ and $(r(E), \alpha) \in P$, by Proposition 6.2, there exists at least a PGL$^+$ interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$ such that $\|p(D_p)\|_{\mathbf{M}} = 1$,

and $\|r(E)\|_{\mathbf{M}} = 1$ for each predicate symbol r. Moreover, as $\mathcal{P}$ is a PGL$^+$ program satisfying the context constraint, if $\beta > \|p(A)\|^{U,m}_{P_p}$, there exists at least a PGL$^+$ interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$ such that $\|p(D_p)\|_{\mathbf{M}} = 1 - \|p(A)\|^{U,m}_{P_p}$ and $\|p(A)\|_{\mathbf{M}} = 0$. Then, if $p \neq r$, there exists at least a PGL$^+$ interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$ such that $\|p(D_p)\|_{\mathbf{M}} = 1 - \|p(A)\|^{U,m}_{P_p}$ and $\|p(A)\|_{\mathbf{M}} = 0$ and $\|r(E)\|_{\mathbf{M}} = 1$. If $p = r$, we have that $(p(E), \alpha) \in P_p$. Then, let E' be an object constant of sort σ_p such that $\mu_{m(E')} = \max(1 - \alpha, \mu_{m(E)})$. By the soundness of the UN inference rule, $P_p \models^{U,m}_{PGL^+} (p(E'), 1)$, and thus, $\mu_{m(D_p)} \leq \mu_{m(E')}$. Therefore, if $p = r$, there exists at least a PGL$^+$ interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$ such that $\|p(D_p)\|_{\mathbf{M}} = 1 - \|p(A)\|^{U,m}_{P_p}$ and $\|p(A)\|_{\mathbf{M}} = 0$ and $1 - \|p(A)\|^{U,m}_{P_p} \leq \max(1 - \alpha, \|p(E)\|_{\mathbf{M}})$. Hence, as $r \neq q$ even if $p = r$, in both cases we are considering each value $u \in U_{\sigma_q}$. On the other hand, as $P \models^{U,m}_{PGL^+} P_2$, by the soundness of the UN inference rule $P \models^{U,m}_{PGL^+} (q(D), 1)$. Then, by Proposition 6.2, π is normalized in both cases. Finally, we can easily check that $\pi \models^{U,m}_{PGL^+} P_1$ and $N^*([q(C)] \mid \pi) = \inf_{u \in U_{\sigma_q}} \mu_{m(D)}(u) \Rightarrow \mu_{m(C)}(u) = N^*(m(C) \mid m(D))$, in both cases too. Hence, $\|q(C)\|^{U,m}_{P_1} = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P_1\} \leq N^*(m(C) \mid m(D)) = \|q(C)\|^{U,m}_{P_2}$.

■

Proposition 6.14 *Let $\mathcal{P} = (P, U, m)$ be a PGL$^+$ program, and let $(p(A) \to q(B), \beta)$ and $(r(E) \to q(F), \gamma)$ be two valid clauses of P_q such that $p \neq r$. Then,* $\|q(C)\|^{U,m}_{\{(p(D_p),1),(p(A)\to q(B),\beta),(r(D_r),1),(r(E)\to q(F),\gamma)\}} = \|q(C)\|^{U,m}_{\{(q(B),\min(\beta,\|p(A)\|^{U,m}_{P_p})),(q(F),\min(\gamma,\|r(E)\|^{U,m}_{P_r}))\}}$.

Proof: We define

$$P_1 = \{(p(D_p), 1), (p(A) \to q(B), \beta), (r(D_r), 1), (r(E) \to q(F), \gamma)\}$$

and $P_2 = \{(q(B), \min(\beta, \|p(A)\|^{U,m}_{P_p})), (q(F), \min(\gamma, \|r(E)\|^{U,m}_{P_r}))\}$.

By Propositions 6.10 and 6.6, $\|p(A)\|^{U,m}_{P_p} = \|p(A)\|^{U,m}_{\{(p(D_p),1)\}} = N^*(m(A) \mid m(D_p))$ and $\|r(E)\|^{U,m}_{P_r} = \|r(E)\|^{U,m}_{\{(r(D_r),1)\}} = N^*(m(E) \mid m(D_r))$. Then, by the soundness of the SU and MP inference rules, $P_1 \models^{U,m}_{PGL^+} P_2$, and thus, $\pi \models^{U,m}_{PGL^+} P_2$ for each model $\pi : \mathcal{M}_{U,m} \to [0,1]$ of P_1. Therefore, by Proposition 6.1, $\|q(C)\|^{U,m}_{P_1} = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P_1\} \geq \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P_2\} = \|q(C)\|^{U,m}_{P_2}$. Hence, we must prove that $\|q(C)\|^{U,m}_{P_1} \leq \|q(C)\|^{U,m}_{P_2}$.

Let D be an object constant of sort σ_q such that $\mu_{m(D)} = \min(\max(1 - \min(\beta, \|p(A)\|^{U,m}_{P_p}), \mu_{m(B)}), \max(1 - \min(\gamma, \|r(E)\|^{U,m}_{P_r}), \mu_{m(F)}))$. By Propositions 6.11 and 6.6, $\|q(C)\|^{U,m}_{P_2} = \|q(C)\|^{U,m}_{\{(q(D),1)\}} = N^*(m(C) \mid m(D))$.

On the other hand, as $\mathcal{P}$ is a PGL$^+$ program satisfying the context constraint and $\{(p(A) \to q(B), \beta), (r(E) \to q(F), \gamma)\} \subseteq P_q^+$, either $\beta \leq \|p(A)\|_{P_p}^{U,m}$ or, for some $u \in U_{\sigma_p}$, $\mu_{m(A)}(u) = 0$ and $\mu_{m(D_p)}(u) = 1 - \|p(A)\|_{P_p}^{U,m}$; and either $\gamma \leq \|r(E)\|_{P_r}^{U,m}$ or, for some $v \in U_{\sigma_r}$, $\mu_{m(E)}(v) = 0$ and $\mu_{m(D_r)}(v) = 1 - \|r(E)\|_{P_r}^{U,m}$. Then, we distinguish four cases.

Case $\beta \leq \|p(A)\|_{P_p}^{U,m}$ **and** $\gamma \leq \|r(E)\|_{P_r}^{U,m}$**.** Let $\pi : \mathcal{M}_{U,m} \to [0,1]$ be a possibility distribution with the following definition:

$$\pi(\mathbf{M}) = \begin{cases} \|q(D)\|_{\mathbf{M}}, & \text{if } \|p(D_p)\|_{\mathbf{M}} = 1 \text{ and } \|r(D_r)\|_{\mathbf{M}} = 1 \\ 0, & \text{otherwise.} \end{cases}$$

Case $\beta \leq \|p(A)\|_{P_p}^{U,m}$ **and** $\|r(E)\|_{P_r}^{U,m} < \gamma$**.** Let $\pi : \mathcal{M}_{U,m} \to [0,1]$ be a possibility distribution with the following definition:

$$\pi(\mathbf{M}) = \begin{cases} \|q(D)\|_{\mathbf{M}}, & \text{if } \|p(D_p)\|_{\mathbf{M}} = 1 \text{ and either} \\ & \|r(D_r)\|_{\mathbf{M}} = 1 \text{ and} \\ & \|q(F)\|_{\mathbf{M}} \geq 1 - \|r(E)\|_{P_r}^{U,m}, \text{ or} \\ & \|r(D_r)\|_{\mathbf{M}} = 1 - \|r(E)\|_{P_r}^{U,m} \text{ and} \\ & |r(E)\|_{\mathbf{M}} = 0 \text{ and } \|q(F)\|_{\mathbf{M}} < 1 - \|r(E)\|_{P_r}^{U,m} \\ 0, & \text{otherwise.} \end{cases}$$

Case $\|p(A)\|_{P_p}^{U,m} < \beta$ **and** $\gamma \leq \|r(E)\|_{P_r}^{U,m}$**.** Let $\pi : \mathcal{M}_{U,m} \to [0,1]$ be a possibility distribution with the following definition:

$$\pi(\mathbf{M}) = \begin{cases} \|q(D)\|_{\mathbf{M}}, & \text{if } \|r(D_r)\|_{\mathbf{M}} = 1 \text{ and either} \\ & \|p(D_p)\|_{\mathbf{M}} = 1 \text{ and} \\ & \|q(B)\|_{\mathbf{M}} \geq 1 - \|p(A)\|_{P_p}^{U,m}, \text{ or} \\ & \|p(D_p)\|_{\mathbf{M}} = 1 - \|p(A)\|_{P_p}^{U,m} \text{ and} \\ & \|p(A)\|_{\mathbf{M}} = 0 \text{ and } \|q(B)\|_{\mathbf{M}} < 1 - \|p(A)\|_{P_p}^{U,m} \\ 0, & \text{otherwise.} \end{cases}$$

Case $\|p(A)\|_{P_p}^{U,m} < \beta$ **and** $\|r(E)\|_{P_r}^{U,m} < \gamma$**.** Let $\pi : \mathcal{M}_{U,m} \to [0,1]$ be a possibility distribution with the following definition:

$$\pi(\mathbf{M}) = \begin{cases} \|q(D)\|_{\mathbf{M}}, & \text{if either } \|p(D_p)\|_{\mathbf{M}} = 1 \text{ and} \\ & \|q(B)\|_{\mathbf{M}} \geq 1 - \|p(A)\|_{P_p}^{U,m}, \text{ or} \\ & \|p(D_p)\|_{\mathbf{M}} = 1 - \|p(A)\|_{P_p}^{U,m} \text{ and} \\ & \|p(A)\|_{\mathbf{M}} = 0 \text{ and} \\ & \|q(B)\|_{\mathbf{M}} < 1 - \|p(A)\|_{P_p}^{U,m}; \text{ and either} \\ & \|r(D_r)\|_{\mathbf{M}} = 1 \text{ and} \\ & \|q(F)\|_{\mathbf{M}} \geq 1 - \|r(E)\|_{P_r}^{U,m}, \text{ or} \\ & \|r(D_r)\|_{\mathbf{M}} = 1 - \|r(E)\|_{P_r}^{U,m} \text{ and} \\ & \|r(E)\|_{\mathbf{M}} = 0 \text{ and } \|q(F)\|_{\mathbf{M}} < 1 - \|r(E)\|_{P_r}^{U,m} \\ 0, & \text{otherwise.} \end{cases}$$

By Proposition 6.9, $P_p \models^{U,m}_{PGL^+} (p(D_p),1)$ and $P_r \models^{U,m}_{PGL^+} (r(D_r),1)$, and thus, $P \models^{U,m}_{PGL^+} (p(D_p),1)$ and $P \models^{U,m}_{PGL^+} (r(D_r),1)$. Then, by Proposition 6.2, there exists at least a PGL$^+$ interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$ such that $\|p(D_p)\|_{\mathbf{M}} = 1$ and $\|r(D_r)\|_{\mathbf{M}} = 1$. Moreover, as $\mathcal{P}$ is a PGL$^+$ program satisfying the context constraint and $\{(p(A) \rightarrow q(B),\beta),(r(E) \rightarrow q(F),\gamma)\} \subseteq P_q^+$ and $r \neq p$, in all four cases we are considering each value $u \in U_{\sigma_q}$. On the other hand, by the soundness of the RE and FU inference rules, $P \models^{U,m}_{PGL^+} \{(p(A) \rightarrow q(B),\beta),(r(E) \rightarrow q(F),\gamma)\}$. Hence, $P \models^{U,m}_{PGL^+} P_2$ and, by the soundness of the UN and IN inference rules, $P \models^{U,m}_{PGL^+} \{(q(D),1)\}$. Then, by Proposition 6.2, π is normalized in all four cases. Finally, we can easily check that $\pi \models^{U,m}_{PGL^+} P_1$ and $N^*([q(C)] \mid \pi) = \inf_{u \in U_{\sigma_q}} \mu_{m(D)}(u) \Rightarrow \mu_{m(C)}(u) = N^*(m(C) \mid m(D))$, in all four cases too. Hence, by Proposition 6.1, $\|q(C)\|^{U,m}_{P_1} = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P_1\} \leq N^*(m(C) \mid m(D)) = \|q(C)\|^{U,m}_{P_2}$.

■

Proposition 6.15 *Let $\mathcal{P} = (P,U,m)$ be a PGL$^+$ program, and let $(p(A) \rightarrow q(B),\beta)$ and $(p(E) \rightarrow s(F),\gamma)$ be two valid clauses of P_q such that $q \neq s$. Then, $\|q(C)\|^{U,m}_{\{(p(D_p),1),(p(A)\rightarrow q(B),\beta),(p(E)\rightarrow s(F),\gamma),(s(D_s),1)\}} = \|q(C)\|^{U,m}_{\{(q(B),\min(\beta,\|p(A)\|^{U,m}_{P_p}))\}}$.*

Proof: We define

$$P_1 = (p(D_p),1),(p(A) \rightarrow q(B),\beta),(p(E) \rightarrow s(F),\gamma),(s(D_s),1)\}$$

and $P_2 = \{(q(B),\min(\beta,\|p(A)\|^{U,m}_{P_p}))\}$.

By Propositions 6.10 and 6.6, $\|p(A)\|^{U,m}_{P_p} = \|p(A)\|^{U,m}_{\{(p(D_p),1)\}} = N^*(m(A) \mid m(D_p))$. Then, by the soundness of the SU and MP inference rules, $P_1 \models^{U,m}_{PGL^+} P_2$, and thus, $\pi \models^{U,m}_{PGL^+} P_2$ for each model $\pi : \mathcal{M}_{U,m} \rightarrow [0,1]$ of P_1. Therefore, by Proposition 6.1, $\|q(C)\|^{U,m}_{P_1} = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P_1\} \geq \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P_2\} = \|q(C)\|^{U,m}_{P_2}$. Then, we must prove that $\|q(C)\|^{U,m}_{P_1} \leq \|q(C)\|^{U,m}_{P_2}$.

Let D be an object constant of sort σ_q such that

$$\mu_{m(D)} = \max(1 - \min(\beta,\|p(A)\|^{U,m}_{P_p}),\mu_{m(B)}).$$

By Proposition 6.6, $\|q(C)\|^{U,m}_{P_2} = \|q(C)\|^{U,m}_{\{(q(D),1)\}} = N^*(m(C) \mid m(D))$. As P is satisfiable in the context determined by U and m, let $\mathbf{M}_{\text{sat}} = (U,i_{\text{sat}},m)$ be a PGL$^+$ interpretation of $\mathcal{M}_{U,m}$ such that $\|\phi\|_{\mathbf{M}_{\text{sat}}} = 1$ for each clause $(\phi,\alpha) \in P$ with $\alpha > 0$. Moreover, as $\mathcal{P}$ is a PGL$^+$ program satisfying the context constraint and $(p(A) \rightarrow q(B),\beta) \in P_q^+$, either $\beta \leq \|p(A)\|^{U,m}_{P_p}$ or, for some $u \in U_{\sigma_p}$, $\mu_{m(A)}(u) = 0$ and $\mu_{m(D_p)}(u) = 1 - \|p(A)\|^{U,m}_{P_p}$. Then, we distinguish two cases.

Case $\beta \leq \|p(A)\|_{P_p}^{U,m}$. Let $\pi : \mathcal{M}_{U,m} \to [0,1]$ be a possibility distribution with the following definition:

$$\pi(\mathbf{M}) = \begin{cases} \|q(D)\|_{\mathbf{M}}, & \text{for each } \mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m} \text{ such that} \\ & i(p) = i_{\text{sat}}(p) \text{ and } i(s) = i_{\text{sat}}(s) \\ 0, & \text{otherwise.} \end{cases}$$

Case $\|p(A)\|_{P_p}^{U,m} < \beta$. Let $\pi : \mathcal{M}_{U,m} \to [0,1]$ be a possibility distribution with the following definition:

$$\pi(\mathbf{M}) = \begin{cases} \|q(D)\|_{\mathbf{M}}, & \text{for each } \mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m} \text{ such that} \\ & i(s) = i_{\text{sat}}(s) \text{ and either } i(p) = i_{\text{sat}}(p) \text{ and} \\ & \|q(B)\|_{\mathbf{M}} \geq 1 - \|p(A)\|_{P_p}^{U,m}, \text{ or} \\ & \|p(D_p)\|_{\mathbf{M}} = 1 - \|p(A)\|_{P_p}^{U,m} \text{ and} \\ & \|p(A)\|_{\mathbf{M}} = 0 \text{ and } \|q(B)\|_{\mathbf{M}} < 1 - \|p(A)\|_{P_p}^{U,m} \\ 0, & \text{otherwise.} \end{cases}$$

As $\mathcal{P}$ satisfies the context constraint C2 and $(p(A) \to q(B), \beta) \in P_q^+$ and $s \neq q$, in both cases we are considering each value $u \in U_{\sigma_q}$.

Moreover, by Proposition 6.9, $P_p \models_{PGL^+}^{U,m} (p(D_p), 1)$ and $P_s \models_{PGL^+}^{U,m} (s(D_s), 1)$, and thus, $P \models_{PGL^+}^{U,m} (p(D_p), 1)$ and $P \models_{PGL^+}^{U,m} (s(D_s), 1)$. And, as $\{(p(A) \to q(B), \beta), (p(E) \to s(F), \gamma)\} \subseteq P_q^+$, by the soundness of the RE and FU inference rules, $P \models_{PGL^+}^{U,m} \{(p(A) \to q(B), \beta), (p(E) \to s(F), \gamma)\}$. Hence, $P \models_{PGL^+}^{U,m} P_1$, and thus, $P \models_{PGL^+}^{U,m} P_2$. Finally, by the soundness of the UN inference rule, $P \models_{PGL^+}^{U,m} (q(D), 1)$. Then, by Proposition 6.2, π is normalized in both cases.

On the other hand, as $\mathcal{P}$ satisfies the context constraint C1, for each clause $(p(G) \to t(H), \delta) \in P \backslash P_p$, we have that, for each $u \in U_{\sigma_p}$, either $\mu_{m(G)}(u) \leq \mu_{m(H)}(i_{\text{sat}}(t))$ or $\mu_{m(D_p)}(u) \leq \max(1 - \delta, \mu_{m(H)}(i_{\text{sat}}(t)))$. Now, by Proposition 6.3, for each clause $(p(I) \to l(J), \lambda) \in P^+$ such that $(p(I) \to l(J), \lambda) \notin P$, by Proposition 6.3, we have $(p(I) \to l(J), \lambda)$ can be derived from P by applying the RE or FU inference rules.

Then, for each clause $(p(I) \to l(J), \lambda) \in P^+$ derived from two clauses of the form $\{(p(I) \to r(K_1), \lambda_1), (r(K_2) \to l(J), \lambda_2)\} \subseteq P$ by applying the RE inference rule, we have that if $\lambda_1 = 0$ or $\lambda_2 = 0$, then $\lambda = 0$, and thus, for each $u \in U_{\sigma_p}$, $\mu_{m(D_p)}(u) \leq \max(1 - \lambda, \mu_{m(J)}(i_{\text{sat}}(l)))$. And, if $\lambda_1 \neq 0$ and $\lambda_2 \neq 0$, then $\lambda = \min(\lambda_1, \lambda_2)$ and $\mu_{m(K_1)}(i_{\text{sat}}(r)) \leq \mu_{m(K_2)}(i_{\text{sat}}(r))$ and $\mu_{m(K_2)}(i_{\text{sat}}(r)) \leq \mu_{m(J)}(i_{\text{sat}}(l))$, and thus, for each $u \in U_{\sigma_p}$, either $\mu_{m(I)}(u) \leq \mu_{m(J)}(i_{\text{sat}}(l))$ or $\mu_{m(D_p)}(u) \leq \max(1 - \lambda, \mu_{m(J)}(i_{\text{sat}}(l)))$.

For each clause $(p(I) \to l(J), \lambda) \in P^+$ derived from two clauses of the form $\{(p(I_1) \to l(J_1), \lambda_1), (p(I_2) \to l(J_2), \lambda_2)\} \subseteq P$ by applying the FU inference rule, we have that if $\lambda_1 = 0$ or $\lambda_2 = 0$, then $\lambda = 0$, and thus, for each $u \in U_{\sigma_p}$ $\mu_{m(D_p)}(u) \leq \max(1 - \lambda, \mu_{m(J)}(i_{\text{sat}}(l)))$. And, if $\lambda_1 \neq 0$ and $\lambda_2 \neq 0$, then $\lambda = \min(\lambda_1, \lambda_2)$ and $\mu_{m(I)} = \max(\mu_{m(I_1)}, \mu_{m(I_2)})$

and $\mu_{m(J)} = \max(\mu_{m(J_1)}, \mu_{m(J_2)})$, and thus, for each $u \in U_{\sigma_p}$, either $\mu_{m(I)}(u) \leq \mu_{m(J)}(i_{\text{sat}}(l))$ or $\mu_{m(D_p)}(u) \leq \max(1-\lambda, \mu_{m(J)}(i_{\text{sat}}(l)))$.

Hence, as $(p(E) \rightarrow s(F), \gamma) \in P_q^+$, we have that, for each $u \in U_{\sigma_p}$ either $\mu_{m(E)}(u) \leq \mu_{m(F)}(i_{\text{sat}}(s))$ or $\mu_{m(D_p)}(u) \leq \max(1-\gamma, \mu_{m(F)}(i_{\text{sat}}(s)))$.

Finally, as $P \models^{U,m}_{PGL^+} (p(D_p), 1)$ and $P \models^{U,m}_{PGL^+} (s(D_s), 1)$, by Proposition 6.2 $\mu_{m(D_p)}(i_{\text{sat}}(p)) = 1$ and $\mu_{m(D_s)}(i_{\text{sat}}(s)) = 1$. Therefore, we can easily check that $\pi \models^{U,m}_{PGL^+} P_1$ and $N^*([q(C)] \mid \pi) = \inf_{u \in U_{\sigma_q}} \mu_{m(D)}(u) \Rightarrow \mu_{m(C)}(u) = N^*(m(C) \mid m(D))$, in both cases too.

Thus, by Proposition 6.1, $\|q(C)\|^{U,m}_{P_1} = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P_1\} \leq N^*(m(C) \mid m(D)) = \|q(C)\|^{U,m}_{P_2}$. ■

Corollary 6.2 *Let $\mathcal{P} = (P, U, m)$ be a PGL$^+$ program, and let $(p(A) \rightarrow q(B), \beta)$ and $(r(E) \rightarrow s(F), \gamma)$ be two valid clauses of P_q such that $p \neq r$, $p \neq s$ and $q \neq s$. Then, $\|q(C)\|^{U,m}_{\{(p(D_p),1),(p(A)\rightarrow q(B),\beta),(r(D_r),1),(r(E)\rightarrow s(F),\gamma),(s(D_s),1)\}} = \|q(C)\|^{U,m}_{\{(q(B),\min(\beta,\|p(A)\|^{U,m}_{P_p}))\}}$.*

Proof: We define

$$P_1 = \{(p(D_p), 1), (p(A) \rightarrow q(B), \beta), (r(D_r), 1), (r(E) \rightarrow s(F), \gamma), (s(D_s), 1)\}$$

and

$$P_2 = \{(q(B), \min(\beta, \|p(A)\|^{U,m}_{P_p}))\}.$$

On the one hand, as in Proposition 6.15, $P_1 \models^{U,m}_{PGL^+} P_2$, and thus, $\pi \models^{U,m}_{PGL^+} P_2$ for each model $\pi : \mathcal{M}_{U,m} \rightarrow [0,1]$ of P_1. Therefore, by Proposition 6.1, $\|q(C)\|^{U,m}_{P_1} = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P_1\} \geq \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P_2\} = \|q(C)\|^{U,m}_{P_2}$.

On the other hand, as P is satisfiable in the context determined by U and m, there exists a PGL$^+$ interpretation $\mathbf{M}_{\text{sat}} = (U, i_{\text{sat}}, m) \in \mathcal{M}_{U,m}$ such that $\|\phi\|_{\mathbf{M}_{\text{sat}}} = 1$ for each clause $(\phi, \alpha) \in P$ with $\alpha > 0$. Now, by Proposition 6.9, $P_r \models^{U,m}_{PGL^+} (r(D_r), 1)$ and $P_s \models^{U,m}_{PGL^+} (s(D_s), 1)$, and thus, $P \models^{U,m}_{PGL^+} (r(D_r), 1)$ and $P \models^{U,m}_{PGL^+} (s(D_s), 1)$. And, as $(r(E) \rightarrow s(F), \gamma) \in P_q^+$, by the soundness of the RE and FU inference rules $P \models^{U,m}_{PGL^+} (r(E) \rightarrow s(F), \gamma)$. Therefore, by Proposition 6.2, $\|r(D_r)\|_{\mathbf{M}_{\text{sat}}} = 1$ and $\|s(D_s)\|_{\mathbf{M}_{\text{sat}}} = 1$ and $\max(1-\gamma, \|r(E) \rightarrow s(F)\|_{\mathbf{M}_{\text{sat}}}) = 1$. Finally, as $(r(E) \rightarrow s(F), \gamma) \in P_q^+$, it must be $q \neq r$. Moreover, as $p \neq r$ and $p \neq s$ and $q \neq s$, for each pair of values $u \in U_{\sigma_q}$ and $v \in U_{\sigma_p}$, there exists at least a PGL$^+$ interpretation $\mathbf{M} = (U, i, m) \in \mathcal{M}_{U,m}$ such that $i(q) = u$, $i(p) = v$, $i(r) = i_{\text{sat}}(r)$ and $i(s) = i_{\text{sat}}(s)$.

Hence, following the proof process used in Proposition 6.14 and taking into account that either $\beta \leq \|p(A)\|^{U,m}_{P_p}$ or, for some $v \in U_{\sigma_p}$, $\mu_{m(A)}(v) = 0$ and $\mu_{m(D_p)}(v) = 1 - \|p(A)\|^{U,m}_{P_p}$, we can easily define a normalized possibility distribution $\pi : \mathcal{M}_{U,m} \rightarrow [0,1]$ such that $\pi \models^{U,m}_{PGL^+} P_1$ and $N^*([q(C)] \mid \pi) = \|q(C)\|^{U,m}_{P_2}$, and thus, $\|q(C)\|^{U,m}_{P_1} = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P_1\} \leq \|q(C)\|^{U,m}_{P_2}$. ■

At this point we are ready to prove that the degree of deduction obtained by applying the SU and MP inference rules is the greatest lower bound of possibilistic entailment for a PGL$^+$ program satisfying both the modularity and the context constraints.

Proposition 6.16 *Let $\mathcal{P} = (P, U, m)$ be a PGL^+ program and let $(p(A) \to q(B), \beta)$ be a valid clause of P_q. Then,*

$$\|q(C)\|_P^{U,m} = \|q(C)\|_{P_q^+}^{U,m} = \|q(C)\|_{P_q'}^{U,m},$$

where $P_q' = (P_q^+ \backslash \{(p(A) \to q(B), \beta)\}) \cup \{(q(B), \min(\beta, \|p(A)\|_{P_p}^{U,m}))\}$.

Proof: As $\mathcal{P}$ is a PGL$^+$ program satisfying the context constraint, by Proposition 6.12, $\|q(C)\|_P^{U,m} = \|q(C)\|_{P_q}^{U,m}$ and, by Proposition 6.4, $\|q(C)\|_{P_q}^{U,m} = \|q(C)\|_{P_q^+}^{U,m}$. Hence, we must prove that $\|q(C)\|_{P_q^+}^{U,m} = \|q(C)\|_{P_q'}^{U,m}$.

By Corollary 6.1, $P_p \models_{PGL^+}^{U,m} (p(A), \|p(A)\|_{P_p}^{U,m})$ and, as $P_p \subseteq P_q^+$, we have that $P_q^+ \models_{PGL^+}^{U,m} (p(A), \|p(A)\|_{P_p}^{U,m})$. Therefore, as $(p(A) \to q(B), \beta) \in P_q^+$, by the soundness of the MP inference rule, we have that $P_q^+ \models_{PGL^+}^{U,m} (q(B), \min(\beta, \|p(A)\|_{P_p}^{U,m}))$. Therefore, by Proposition 6.1, $\|q(C)\|_{P_q^+}^{U,m} = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models_{PGL^+}^{U,m} P_q^+\} \geq \inf\{N^*([q(C)] \mid \pi) \mid \pi \models_{PGL^+}^{U,m} P_q'\} = \|q(C)\|_{P_q'}^{U,m}$. Then, we must prove that $\|q(C)\|_{P_q^+}^{U,m} \leq \|q(C)\|_{P_q'}^{U,m}$.

Now, by Proposition 6.4, $\|q(C)\|_{P_q'}^{U,m} = \|q(C)\|_{P_q'^+}^{U,m}$, where

$$P_q'^+ = \begin{cases} P_q', & \text{if } (p(A) \to q(B), \beta) \text{ is a basic clause of } P_q^+ \\ P_q^+ \cup \{(q(B), \min(\beta, \|p(A)\|_{P_p}^{U,m}))\}, & \text{otherwise.} \end{cases}$$

Then, if $(p(A) \to q(B), \beta)$ is not a basic clause of P_q^+, we have that $P_q^+ \subseteq P_q'^+$, and thus, $\|q(C)\|_{P_q^+}^{U,m} \leq \|q(C)\|_{P_q'^+}^{U,m}$. Hence, we must prove that if $(p(A) \to q(B), \beta)$ is a basic clause of P_q^+, then $\|q(C)\|_{P_q^+}^{U,m} \leq \|q(C)\|_{P_q'}^{U,m}$.

Let D_q and D_q' be two object constants of sort σ_q such that

$$\mu_{m(D_q)} = \bigwedge\{\mu_{m(D)} \mid P_q^+ \models_{PGL^+}^{U,m} (q(D), 1)\}$$

and

$$\mu_{m(D_q')} = \bigwedge\{\mu_{m(D)} \mid P_q' \models_{PGL^+}^{U,m} (q(D), 1)\}.$$

By Proposition 6.10 and 6.6, $\|q(C)\|_{P_q^+}^{U,m} = \|q(C)\|_{\{(q(D_q),1)\}}^{U,m} = N^*(m(C) \mid m(D_q))$ and $\|q(C)\|_{P_q'}^{U,m} = \|q(C)\|_{\{(q(D_q'),1)\}}^{U,m} = N^*(m(C) \mid m(D_q'))$. Hence, we must prove that if $(p(A) \to q(B), \beta)$ is a basic clause of P_q^+, then $\mu_{m(D_q')} \leq \mu_{m(D_q)}$. To this end, we prove that $P_q' \models_{PGL^+}^{U,m} (q(D), 1)$ for each object constant D of sort σ_q such that $P_q^+ \models_{PGL^+}^{U,m} (q(D), 1)$.

Suppose that there exists an object constant D of sort σ_q such that $P_q^+ \models^{U,m}_{PGL^+} (q(D),1)$ and $P_q' \not\models^{U,m}_{PGL^+} (q(D),1)$. Now, let D_p be an object constant of sort σ_p such that

$$\mu_{m(D_p)} = \bigwedge\{\mu_{m(D)} \mid P_p \models^{U,m}_{PGL^+} (p(D),1)\}.$$

Then, by Propositions 6.10 and 6.6, $\|p(A)\|^{U,m}_{P_p} = \|p(A)\|^{U,m}_{\{(p(D_p),1)\}} = N^*(m(A) \mid m(D_p))$. As $\mathcal{P}$ satisfies the context constraint C1, by Propositions 6.12 and 6.4, $\|p(A)\|^{U,m}_{P_p} = \|p(A)\|^{U,m}_{P_q^+}$ and $\|p(A)\|^{U,m}_{P_p} = \|p(A)\|^{U,m}_{P_q'}$, and thus, $\|p(A)\|^{U,m}_{P_q^+} = \|p(A)\|^{U,m}_{P_q'}$.

On the other hand, as $(p(A) \to q(B),\beta) \in P_q^+$ and $\mathcal{P}$ satisfies the context constraint C2, either $\beta \leq \|p(A)\|^{U,m}_{P_p}$ or, for some $v \in U_{\sigma_p}$, $\mu_{m(A)}(v) = 0$ and $\mu_{m(D_p)}(v) = 1 - \|p(A)\|^{U,m}_{P_p}$. Then, by Proposition 6.8, $\|q(C)\|^{U,m}_{\{(p(D_p),1),(p(A)\to q(B),\beta)\}} = \|q(C)\|^{U,m}_{\{(q(B),\min(\beta,N^*(m(A)|m(D_p))))\}} = \|q(C)\|^{U,m}_{\{(q(B),\min(\beta,\|p(A)\|^{U,m}_{P_p}))\}}$, and thus, $\|q(C)\|^{U,m}_{\{(p(D_p),1),(p(A)\to q(B),\beta)\}} = \|q(C)\|^{U,m}_{\{(q(B),\min(\beta,\|p(A)\|^{U,m}_{P_q^+})\}} = \|q(C)\|^{U,m}_{\{(q(B),\min(\beta,\|p(A)\|^{U,m}_{P_q'})\}}$. We define

$$P_1 = P_q^+ \backslash \{(p(A) \to q(B),\beta)\}.$$

Then, if

$$P_1 \cup \{(p(A) \to q(B),\beta)\} \models^{U,m}_{PGL^+} (q(D),1)$$

and

$$P_1 \cup \{(q(B),\min(\beta,\|p(A)\|^{U,m}_{P_p}))\} \not\models^{U,m}_{PGL^+} (q(D),1),$$

there should exist at least a clause $(\varphi,\alpha) \in P_1$ which together with $(p(A) \to q(B),\beta)$ should enable P_q^+ to entail $(q(D),1)$. Thus, $(p(A) \to q(B),\beta)$ together with $(\varphi,\alpha) \in P_1$ should entail a clause of the form $(k \to q(E),\gamma) \notin P_q^+$ for some atomic formula k, and such that $\max(1 - \min(\gamma,\|k\|^{U,m}_{P_k}),\mu_{m(E)}(u)) < \max(1 - \min(\beta,\|p(A)\|^{U,m}_{P_p}),\mu_{m(B)}(u))$, for some $u \in U_{\sigma_q}$.

Now, by Proposition 6.13, $\|q(C)\|^{U,m}_{\{(r(F),\alpha),(p(D_p),1),(p(A)\to q(B),\beta)\}} = \|q(C)\|^{U,m}_{\{(r(F),\alpha),(q(B),\min(\beta,\|p(A)\|^{U,m}_{P_p}))\}}$ for each unit clause $(r(F),\alpha) \in P_1$, and thus, $(\varphi,\alpha) \in P_1$ cannot be a unit PGL$^+$ clause. Moreover, by Proposition 6.14, for each clause $(r(F) \to q(G),\alpha) \in P_1$ such that $p \neq r$, $\|q(C)\|^{U,m}_{\{(p(D_p),1),(p(A)\to q(B),\beta),(r(D_r),1),(r(F)\to q(G),\alpha)\}} = \|q(C)\|^{U,m}_{\{(q(B),\min(\beta,\|p(A)\|^{U,m}_{P_p})),(q(G),\min(\alpha,\|r(F)\|^{U,m}_{P_r}))\}}$; and, by Proposition 6.12, neither $\|r(F)\|^{U,m}_{P_r}$ nor $\|p(A)\|^{U,m}_{P_p}$ depend on $(p(A) \to q(B),\beta)$ and $(r(F) \to q(G),\alpha)$. And, by Proposition 6.15, for each clause $(p(F) \to s(G),\alpha) \in P_1$ such that $q \neq s$, $\|q(C)\|^{U,m}_{\{(p(D_p),1),(p(A)\to q(B),\beta),(p(F)\to s(G),\alpha),(s(D_s),1)\}} = \|q(C)\|^{U,m}_{\{(q(B),\min(\beta,\|p(A)\|^{U,m}_{P_p}))\}}$; and, by Proposition 6.12, neither the object

constant D_p nor D_s depend on $(p(A) \to q(B), \beta)$. And, finally, by Corollary 6.2, for each clause $(r(F) \to s(G), \alpha) \in P_1$ such that $p \neq r$ and $p \neq s$ and $q \neq s$, $\|q(C)\|^{U,m}_{\{(p(D_p),1),(p(A)\to q(B),\beta),(r(D_r),1),(r(F)\to s(G),\alpha),(s(D_s),1)\}} = \|q(C)\|^{U,m}_{\{(q(B),\min(\beta,\|p(A)\|^{U,m}_{P_p}))\}}$; and, by Proposition 6.12, the object constants D_p, D_r and D_s do not depend on $(p(A) \to q(B), \beta)$. Hence, $(\varphi, \alpha) \in P_1$ should be either of the form $(p(F) \to q(G), \alpha)$ or $(r \to p(F), \alpha)$, for some object constants F and G and atomic formula r. We distinguish two cases.

Case $(\varphi, \alpha) = (p(F) \to q(G), \alpha)$. As $(p(A) \to q(B), \beta)$ is a basic clause of P_q^+, $(p(A) \to q(B), \beta) \in P_q$. Then, as P_q^+ satisfies the modularity constraint, if $(p(F) \to q(G), \alpha) \in P_1$ with $\mu_{m(A)} \not\leq \mu_{m(F)}$ and $\mu_{m(F)} \not\leq \mu_{m(A)}$, $(p(F') \to q(G'), \min(\alpha, \beta)) \in P_1$ with $\mu_{m(F')} = \max(\mu_{m(F)}, \mu_{m(A)})$ and $\mu_{m(G')} = \max(\mu_{m(G)}, \mu_{m(B)})$.

On the other hand, if $\mu_{m(A)} \not\leq \mu_{m(F)}$ and $\mu_{m(F)} \not\leq \mu_{m(A)}$, we can easily check that $\{(p(F) \to q(G), \alpha), (p(A) \to q(B), \beta)\} \not\models^{U,m}_{PGL^+} (p(F') \to q(H), \delta)$ with $\mu_{m(H)} < \mu_{m(G')}$ or $\delta > \min(\alpha, \beta)$. Hence, if $\{(p(F) \to q(G), \alpha), (p(A) \to q(B), \beta)\} \models^{U,m}_{PGL^+} (p(I) \to q(J), \gamma)$, either $\mu_{m(I)} \leq \mu_{m(A)}$ and $\mu_{m(J)} \geq \mu_{m(B)}$ and $\gamma \leq \beta$, or $\mu_{m(I)} \leq \mu_{m(F)}$ and $\mu_{m(J)} \geq \mu_{m(G)}$ and $\gamma \leq \alpha$, or $\mu_{m(I)} \leq \mu_{m(F')}$ and $\mu_{m(J)} \geq \mu_{m(G')}$ and $\gamma \leq \min(\alpha, \beta)$.

Case $(\varphi, \alpha) = (r \to p(F), \alpha)$. If $(r \to p(F), \alpha) \in P_1$ with $\mu_{m(F)} \leq \mu_{m(A)}$, then $\{(r \to p(F), \alpha), (p(A) \to q(B), \beta)\} \models^{U,m}_{PGL^+} (r \to q(B), \min(\alpha, \beta))$. However, as P_q^+ satisfies the modularity constraint and $(p(A) \to q(B), \beta) \in P_q$, we have that $(r \to q(B), \min(\alpha, \beta)) \in P_1$. On the other hand, by the soundness of the SU and RE inference rules, we have that $\{(r \to p(F), \alpha), (p(A) \to q(B), \beta)\} \models^{U,m}_{PGL^+} (r \to q(B), \min(\alpha, \beta, N^*(m(F) \mid m(D_p)), N^*(m(A) \mid m(F))))$. And, we can easily check that if $\mu_{m(F)}(v) > \mu_{m(A)}(v)$ for some $v \in U_{\sigma_p}$, then $\{(r \to p(F), \alpha), (p(A) \to q(B), \beta)\} \not\models^{U,m}_{PGL^+} (r \to q(B), \min(\alpha, \beta))$ and $\{(r \to p(F), \alpha), (p(A) \to q(B), \beta)\} \not\models^{U,m}_{PGL^+} (r \to q(B), \min(\alpha, \beta, N^*(m(F) \mid m(D_p))))$ and $\{(r \to p(F), \alpha), (p(A) \to q(B), \beta)\} \not\models^{U,m}_{PGL^+} (r \to q(B), \min(\alpha, \beta, N^*(m(A) \mid m(F))))$. Hence, if $\mu_{m(F)}(v) > \mu_{m(A)}(v)$ for some $v \in U_{\sigma_p}$, then $\min(\alpha, \beta, N^*(m(F) \mid m(D_p)), N^*(m(A) \mid m(F))) \geq \lambda$ for each $\lambda \in [0, 1]$ such that $\{(r \to p(F), \alpha), (p(A) \to q(B), \beta)\} \models^{U,m}_{PGL^+} (r \to q(B), \lambda)$. Finally, we have that $\min(N^*(m(F) \mid m(D_p)), N^*(m(A) \mid m(F))) = \min(\inf_{u \in U_{\sigma_p}} \mu_{m(D_p)}(u) \Rightarrow \mu_{m(F)}(u), \inf_{u \in U_{\sigma_p}} \mu_{m(F)}(u) \Rightarrow \mu_{m(A)}(u)) = \inf_{u \in U_{\sigma_p}} \min(\mu_{m(D_p)}(u) \Rightarrow \mu_{m(F)}(u), \mu_{m(F)}(u) \Rightarrow \mu_{m(A)}(u)) \leq \inf_{u \in U_{\sigma_p}} \mu_{m(D_p)}(u) \Rightarrow \mu_{m(A)}(u) = N^*(m(A) \mid m(D_p)) = \|p(A)\|^{U,m}_{P_p}$. Hence, for each $u \in U_{\sigma_q}$, $\max(1 - \min(\beta, \|p(A)\|^{U,m}_{P_p}), \mu_{m(B)}(u)) \leq \max(1 - \min(\alpha, \beta, N^*(m(F) \mid m(D_p)), N^*(m(A) \mid m(F))), \mu_{m(B)}(u))$.

Therefore, if $(\varphi, \alpha) \in P_1$ and $\{(p(A) \to q(B), \beta), (\varphi, \alpha)\} \models^{U,m}_{PGL^+} (k \to q(E), \gamma)$ and, for some $u \in U_{\sigma_q}$, $\max(1 - \min(\gamma, \|k\|^{U,m}_{P_k}), \mu_{m(E)}(u)) < \max(1 -$

$\min(\beta, \|p(A)\|^{U,m}_{P_p}), \mu_{m(B)}(u))$, then $(k \rightarrow q(E), \gamma) \in P_1$. On the other hand, as $(p(A) \rightarrow q(B), \beta)$ is a basic clause of P_q^+, $P_1^+ = P_1$, and thus, P_1 satisfies the modularity constraint too. Then, if $(\psi, \delta) \in P_1$ and, for some atomic formula l, $\{(k \rightarrow q(E), \gamma), (\psi, \delta)\} \models^{U,m}_{PGL^+} (l \rightarrow q(F), \lambda)$ and, for some $u \in U_{\sigma_q}$, $\max(1 - \min(\lambda, \|l\|^{U,m}_{P_l}), \mu_{m(F)}(u)) < \max(1 - \min(\gamma, \|k\|^{U,m}_{P_k}), \mu_{m(E)}(u))$ and $\max(1 - \min(\lambda, \|l\|^{U,m}_{P_l}), \mu_{m(F)}(u)) < \max(1 - \min(\beta, \|p(A)\|^{U,m}_{P_p}), \mu_{m(B)}(u))$, then $(l \rightarrow q(F), \lambda) \in P_1$. Hence, if $P_1 \cup \{(p(A) \rightarrow q(B), \beta)\} \models^{U,m}_{PGL^+} (q(D), 1)$, then $P_1 \cup \{(q(B), \min(\beta, \|p(A)\|^{U,m}_{P_p}))\} \models^{U,m}_{PGL^+} (q(D), 1)$ as well, and thus, $\mu_{m(D'_q)} \leq \mu_{m(D_q)}$. ■

The following propositions establish the basis for defining an efficient algorithm for computing the maximum degree of possibilistic entailment of a goal from a PGL$^+$ program satisfying both the modularity and the context constraints.

Proposition 6.17 *Let $\mathcal{P} = (P, U, m)$ be a PGL$^+$ program. Then,*

$$\|q(C)\|^{U,m}_{P_q^+} = \|q(C)\|^{U,m}_{P_q^{facts}},$$

where $P_q^{facts} = \{(q(E), \gamma) \in P_q\} \cup \{(q(B), \min(\beta, \|p(A)\|^{U,m}_{P_p})) \mid (p(A) \rightarrow q(B), \beta) \in P_q^+\}$.

Proof: By Proposition 6.16, for each clause $(p(A) \rightarrow q(B), \beta) \in P_q^+$, $\|q(C)\|^{U,m}_{P_q^+} = \|q(C)\|^{U,m}_{P'_q}$, where

$$P'_q = (P_q^+ \backslash \{(p(A) \rightarrow q(B), \beta)\}) \cup \{(q(B), \min(\beta, \|p(A)\|^{U,m}_{P_p}))\}.$$

Then, if P'_q satisfies the modularity constraint, we have that, for each clause $(r(F) \rightarrow q(G), \delta) \in P'_q$, $\|q(C)\|^{U,m}_{P'_q} = \|q(C)\|^{U,m}_{P''_q}$, where

$$P''_q = (P'_q \backslash \{(r(F) \rightarrow q(G), \delta)\}) \cup \{(q(G), \min(\delta, \|r(F)\|^{U,m}_{P_r}))\}.$$

On the other hand, in Proposition 6.5 we have proved that, for each finite set of PGL$^+$ clauses Q, if $(\varphi, \alpha) \in Q$ and φ is a rule with head q, then either (φ, α) is a basic clause of Q, or there exists at least a clause $(\psi, \lambda) \in Q$ such that ψ is a rule with head q and (ψ, λ) is a basic clause of Q. Then, if P_q^+ is a finite set of clauses and P_q^+ contains general PGL$^+$ clauses, there exits at least a clause $(p(A) \rightarrow q(B), \beta) \in P_q^+$ such that $(p(A) \rightarrow q(B), \beta)$ is a basic clause of P_q^+. Hence, $\|q(C)\|^{U,m}_{P_q^+} = \|q(C)\|^{U,m}_{P'_q}$ and P'_q satisfies the modularity constraint. Moreover, if P_q^+ is a finite set of clauses, P'_q is a finite set of clauses too. Then, if P'_q contains general PGL$^+$ clauses, there exits at least a clause $(r(F) \rightarrow q(G), \delta) \in P'_q$ such that $(r(F) \rightarrow q(G), \delta)$ is a basic clause of P'_q, and thus, $\|q(C)\|^{U,m}_{P'_q} = \|q(C)\|^{U,m}_{P''_q}$ and P''_q satisfies the modularity constraint. Therefore, if P_q^+ is a finite set of clauses, repeating the above process we can transform P_q^+

into a finite set of PGL$^+$ clauses of the form $P_q^m = \{(q(B), \min(\beta, \|p(A)\|_{P_p}^{U,m})) \mid (p(A) \rightarrow q(B), \beta) \in P_q^+\} \cup \{(q(E), \gamma) \in P_q\} \cup \{(r \rightarrow s, \lambda) \in P_q^+ \mid s \neq q\}$ and $\|q(C)\|_{P_q^+}^{U,m} = \|q(C)\|_{P_q^m}^{U,m}$. Finally, if P_q^m is a finite set of PGL$^+$ clauses, then (P_q^m, U, m) is a PGL$^+$ program satisfying the context constraint. And, by Proposition 6.12, we have that $\|q(C)\|_{P_q^m}^{U,m} = \|q(C)\|_{P_q^{\text{facts}}}^{U,m}$, and thus, $\|q(C)\|_{P_q^+}^{U,m} = \|q(C)\|_{P_q^{\text{facts}}}^{U,m}$. Then, we must prove that P_q^+ is a finite set of clauses.

As P is a finite set of clauses, P_q is a finite set of clauses too. Now, by Proposition 6.3, for each valid clause $(\varphi, \alpha) \in P_q^+$ such that $(\varphi, \alpha) \notin P_q$, (φ, α) can be derived from the clauses of P_q by applying the RE and FU inference rules.

On the one hand, as P_q does not contain recursive formulas, by applying the RE inference rule we derive a finite set of PGL$^+$ clauses. On the other hand, by applying the FU inference rule we can derive an infinite set of PGL$^+$ clauses. However, as only belong to P_q^+ the clauses derived from P_q with the wider body and the narrower head, we just consider a finite subset of them. Therefore, in the worst-case each combination of clauses of P_q derives a new valid clause. Hence, denoting by N the number of clauses of P_q, in the worst-case the number of clauses of P_q^+ is $N + \sum_{i=2}^{N} \binom{N}{i}$, and thus, P_q^+ is a finite set of clauses as well as P_q^m. ■

Proposition 6.18 *Let $\mathcal{P} = (P, U, m)$ be a PGL$^+$ program, let $(q(B), \beta)$ be a unit clause of P_q^+, and let D_q' be an object constant of sort σ_q such that*

$$\mu_{m(D_q')} = \bigwedge \{\mu_{m(D)} \mid P_q^+ \backslash \{(q(B), \beta)\} \models_{PGL^+}^{U,m} (q(D), 1)\}.$$

Then, $\|q(C)\|_{P_q^+}^{U,m} = \|q(C)\|_{\{(q(D_q'),1),(q(B),\beta)\}}^{U,m}$.

Proof: By Proposition 6.9, $P_q^+ \backslash \{(q(B), \beta)\} \models_{PGL^+}^{U,m} (q(D_q'), 1)$, and thus, $P_q^+ \models_{PGL^+}^{U,m} (q(D_q'), 1)$. Therefore, $\pi \models_{PGL^+}^{U,m} \{(q(D_q'), 1), (q(B), \beta)\}$ for each model $\pi : \mathcal{M}_{U,m} \rightarrow [0, 1]$ of P_q^+. And, by Proposition 6.1, $\|q(C)\|_{P_q^+}^{U,m} = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models_{PGL^+}^{U,m} P_q^+\} \geq \inf\{N^*([q(C)] \mid \pi) \mid \pi \models_{PGL^+}^{U,m} \{(q(D_q'), 1), (q(B), \beta)\}\} = \|q(C)\|_{\{(q(D_q),1),(q(B),\beta)\}}^{U,m}$. Then, we must prove that $\|q(C)\|_{P_q^+}^{U,m} \leq \|q(C)\|_{\{(q(D_q'),1),(q(B),\beta)\}}^{U,m}$.

By Proposition 6.17, $\|q(C)\|_{P_q^+}^{U,m} = \|q(C)\|_{P_q^{\text{facts}}}^{U,m}$, where $P_q^{\text{facts}} = \{(q(F), \gamma) \in P_q\} \cup \{(q(E), \min(\alpha, \|p(A)\|_{P_p}^{U,m}) \mid (p(A) \rightarrow q(E), \alpha) \in P_q^+\}$. Then, we must prove that $\|q(C)\|_{P_q^{\text{facts}}}^{U,m} \leq \|q(C)\|_{\{(q(D_q'),1),(q(B),\beta)\}}^{U,m}$.

Let D_1 and D_2 be two object constants of sort σ_q such that $\mu_{m(D_1)} = \min(\min\{\max(1 - \delta, \mu_{m(G)}) \mid (q(G), \delta) \in P_q^{\text{facts}} \backslash \{(q(B), \beta)\}\}, \max(1 - \beta, \mu_{m(B)}))$ and $\mu_{m(D_2)} = \min(\mu_{m(D_q')}, \max(1 - \beta, \mu_{m(B)}))$.

As P_q^{facts} is a finite set of unit PGL$^+$ clauses with head q, by Propositions 6.11 and 6.6, $\|q(C)\|_{P_q^{\text{facts}}}^{U,m} = \|q(C)\|_{\{(q(D_1),1)\}}^{U,m} = N^*(m(C) \mid m(D_1))$ and

$\|q(C)\|^{U,m}_{\{(q(D'_q),1),(q(B),\beta)\}} = \|q(C)\|^{U,m}_{\{(q(D_2),1)\}} = N^*(m(C) \mid m(D_2))$. Then, we must prove that $\mu_{m(D_2)} \leq \mu_{m(D_1)}$.

Now, as $(q(B),\beta)$ is a unit PGL$^+$ clause, $(P^+_q \backslash \{(q(B),\beta)\})^+ = P^+_q \backslash \{(q(B),\beta)\}$, and thus, $(q(B),\beta)$ is a basic clause of P^+_q. Then, by Proposition 6.17, $\|q(D)\|^{U,m}_{P^+_q \backslash \{(q(B),\beta)\}} = \|q(D)\|^{U,m}_{P^{\text{facts}}_q \backslash \{(q(B),\beta)\}}$, for each object constant D of sort σ_q. Then, by the soundness of the UN inference rule, $\mu_{m(D'_q)} \leq \min\{\max(1-\delta, \mu_{m(G)}) \mid (q(G),\delta) \in P^{\text{facts}}_q \backslash \{(q(B),\beta)\}\}$. Hence, $\mu_{m(D_2)} \leq \mu_{m(D_1)}$. ■

Corollary 6.3 *Let $\mathcal{P} = (P,U,m)$ be a PGL$^+$ program, let $(p(A) \rightarrow q(B),\beta)$ be a basic clause of P^+_q, and let D'_q be an object constant of sort σ_q such that*

$$\mu_{m(D'_q)} = \bigwedge\{\mu_{m(D)} \mid P^+_q \backslash \{(p(A) \rightarrow q(B),\beta)\} \models^{U,m}_{PGL^+} (q(D),1)\}.$$

Then, $\|q(C)\|^{U,m}_{P^+_q} = \|q(C)\|^{U,m}_{\{(q(D'_q),1),(q(B),\min(\beta,\|p(A)\|^{U,m}_{P_p}))\}}$.

Proof: As $(p(A) \rightarrow q(B),\beta) \in P^+_q$, by Proposition 6.16, $\|q(C)\|^{U,m}_{P^+_q} = \|q(C)\|^{U,m}_{P'_q}$, where

$$P'_q = (P^+_q \backslash \{(p(A) \rightarrow q(B),\beta)\}) \cup \{(q(B),\min(\beta,\|p(A)\|^{U,m}_{P_p}))\}.$$

Moreover, as $(p(A) \rightarrow q(B),\beta)$ is a basic clause of P^+_q, we have that $(P^+_q \backslash \{(p(A) \rightarrow q(B),\beta)\})^+ = P^+_q \backslash \{(p(A) \rightarrow q(B),\beta)\}$, and thus, P'_q satisfies the modularity constraint and $\mu_{m(D'_q)} = \bigwedge\{\mu_{m(D)} \mid P'_q \backslash \{(q(B),\min(\beta,\|p(A)\|^{U,m}_{P_p}))\} \models^{U,m}_{PGL^+} (q(D),1)\}$. Then, by Proposition 6.18, $\|q(C)\|^{U,m}_{P'_q} = \|q(C)\|^{U,m}_{\{(q(D'_q),1),(q(B),\min(\beta,\|p(A)\|^{U,m}_{P_p}))\}}$, and thus, $\|q(C)\|^{U,m}_{P^+_q} = \|q(C)\|^{U,m}_{\{(q(D'_q),1),(q(B),\min(\beta,\|p(A)\|^{U,m}_{P_p}))\}}$. ■

Finally, before proving the completeness result, we consider two particular cases.

Proposition 6.19 *Let $\mathcal{P} = (P,U,m)$ be a PGL$^+$ program, and let C be an object constant of sort σ_q such that $\mu_{m(C)}(u) = 1$ for each $u \in U_{\sigma_q}$. Then, $\|q(C)\|^{U,m}_P = |q(C)|^{U,m}_P = 1$.*

Proof: As $\mu_{m(C)}(u) = 1$ for each $u \in U_{\sigma_q}$, $\|q(C)\|_{\mathbf{M}} = 1$ for each PGL$^+$ interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$, and thus, $N^*([q(C)] \mid \pi) = 1$ for each possibilistic model $\pi : \mathcal{M}_{U,m} \rightarrow [0,1]$. Therefore, by Proposition 6.1, $\|q(C)\|^{U,m}_P = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models^{U,m}_{PGL^+} P\} = 1$.

On the other hand, by the triviality axiom, $P \vdash^{U,m}_{PGL^+} (q(C),0)$ and, applying the UN inference rule, we get $(q(C),0) \vdash^{U,m}_{PGL^+} (q(C),1)$, and thus, $|q(C)|^{U,m}_P = \sup\{\alpha \in [0,1] \mid P \vdash^{U,m}_{PGL^+} (q(C),\alpha)\} = 1$. Hence, $\|q(C)\|^{U,m}_P = |q(C)|^{U,m}_P = 1$. ■

Proposition 6.20 *Let $\mathcal{P} = (P, U, m)$ be a PGL$^+$ program, and let C be an object constant of sort σ_q such that $\mu_{m(C)}(u) < 1$ for some $u \in U_{\sigma_q}$. If there is no clause in P with head q, then $\|q(C)\|_P^{U,m} = |q(C)|_P^{U,m} = 0$.*

Proof: As $\mathcal{P}$ is a PGL$^+$ program satisfying the context constraint, by Proposition 6.12, $\|q(C)\|_P^{U,m} = \|q(C)\|_{P_q}^{U,m}$, where, if there is no clause in P with head q, $P_q = \{(q(B), 0)\}$, B being an object constant of sort σ_q.

Now, let $\pi : \mathcal{M}_{U,m} \to [0,1]$ be a possibility distribution such that $\pi(\mathbf{M}) = 1$ for each PGL$^+$ interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$. Obviously, $\pi \models_{PGL^+}^{U,m} (q(B), 0)$ for any object constant B and, as $\mu_{m(C)}(u) < 1$ for some $u \in U_{\sigma_q}$, $N^*([q(C)] \mid \pi) = \inf_{\mathbf{M} \in \mathcal{M}_{U,m}} \pi(\mathbf{M}) \Rightarrow \|q(C)\|_{\mathbf{M}} = 0$. Therefore, by Proposition 6.1, $\|q(C)\|_{P_q}^{U,m} = \inf\{N^*([q(C)] \mid \pi) \mid \pi \models_{PGL^+}^{U,m} P_q\} = 0$, and thus, $\|q(C)\|_P^{U,m} = 0$.

On the other hand, by the triviality axiom, $|q(C)|_P^{U,m} \geq 0$. However, by the soundness of the inference rules, $\|q(C)\|_P^{U,m} \geq |q(C)|_P^{U,m}$, and thus, $\|q(C)\|_P^{U,m} = |q(C)|_P^{U,m} = 0$. ■

Theorem 6.2 (completeness) *Let $\mathcal{P} = (P, U, m)$ be a PGL$^+$ program satisfying the context constraint. Then, $\|q(C)\|_P^{U,m} = |q(C)|_P^{U,m}$.*

Proof: We assume that $\mu_{m(C)}(u) < 1$ for some $u \in U_{\sigma_q}$, (σ_q) being the type of q; otherwise, by Proposition 6.19, $\|q(C)\|_P^{U,m} = |q(C)|_P^{U,m} = 1$.

Moreover, we assume that there is at least clause in P with head q; otherwise, by Proposition 6.20, $\|q(C)\|_P^{U,m} = |q(C)|_P^{U,m} = 0$.

Now, by Proposition 6.12, $\|q(C)\|_P^{U,m} = \|q(C)\|_{P_q}^{U,m}$, where

$$P_q = \{(\varphi, \alpha) \in P \mid \text{head}(\varphi) = q \text{ or } q \text{ depends on head}(\varphi) \text{ in } P\}.$$

Moreover, by Proposition 6.4, $\|q(C)\|_{P_q}^{U,m} = \|q(C)\|_{P_q^+}^{U,m}$, where P_q^+ is the set of valid clauses of P_q. And, by Proposition 6.3, $P_q \vdash_{PGL^+}^{U,m} (\varphi, \alpha)$, for each valid clause (φ, α) of P_q. Hence, we must prove that $\|q(C)\|_{P_q^+}^{U,m} = |q(C)|_{P_q^+}^{U,m}$. However, by the soundness of inference rules, $\|q(C)\|_{P_q^+}^{U,m} \geq |q(C)|_{P_q^+}^{U,m}$, and thus, we must prove that $\|q(C)\|_{P_q^+}^{U,m} \leq |q(C)|_{P_q^+}^{U,m}$. We proceed by induction on n, where n is the number of clauses of P_q^+.

If $n = 1$, it must be that P_q^+ contains either a unit PGL$^+$ clause with predicate symbol q, or a general PGL$^+$ clause with head q.

Suppose that P_q^+ contains only the unit PGL$^+$ clause $(q(B), \beta)$. Let D be an object constant of sort σ_q such that $\mu_{m(D)} = \max(1 - \beta, \mu_{m(B)})$. By Proposition 6.6, $\|q(C)\|_{\{(q(B),\beta)\}}^{U,m} = \|q(C)\|_{\{(q(D),1)\}}^{U,m} = N^*(m(C) \mid m(D))$. On the other hand, applying the UN inference rule, $\{(q(B), \beta)\} \vdash_{PGL^+}^{U,m} (q(D), 1)$ and, applying the SU inference rule, $\{(q(D), 1)\} \vdash_{PGL^+}^{U,m} (q(C), N^*(m(C) \mid m(D)))$. Therefore, $|q(C)|_{\{(q(B),\beta)\}}^{U,m} = \sup\{\alpha \in [0,1] \mid (q(B), \beta) \vdash_{PGL^+}^{U,m} (q(C), \alpha)\} \geq N^*(m(C) \mid m(D))$, and thus, $\|q(C)\|_{\{(q(B),\beta)\}}^{U,m} \leq |q(C)|_{\{(q(B),\beta)\}}^{U,m}$.

Suppose now that P_q^+ contains only the general PGL$^+$ clause $(p(A) \rightarrow q(B), \beta)$. On the one hand, it must be that $(p(A) \rightarrow q(B), \beta) \in P_q$, and thus, $(p(A) \rightarrow q(B), \beta) \in P$. Therefore, $\mu_{m(A)}(v) < 1$ for some $v \in U_{\sigma_p}$, (σ_p) being the type of p. On the other hand, it must be that P contains no clause with head p. Then, by Proposition 6.20, $\|p(A)\|_P^{U,m} = 0$ and, by Proposition 6.12, $\|p(A)\|_P^{U,m} = \|p(A)\|_{\{(p,0)\}}^{U,m}$. As $(p(A) \rightarrow q(B), \beta)$ is a basic clause of P_q^+, by Corollary 6.17, $\|q(C)\|_{\{(p(A)\rightarrow q(B),\beta)\}}^{U,m} = \|q(C)\|_{\{(q(B),\min(\beta,\|p(A)\|_{\{(p,0)\}}^{U,m}))\}}^{U,m}$, and thus, $\|q(C)\|_{\{(p(A)\rightarrow q(B),\beta)\}}^{U,m} = \|q(C)\|_{\{(q(B),0)\}}^{U,m}$. Let D be an object constant of sort σ_q such that $\mu_{m(D)}(u) = 1$ for each $u \in U_{\sigma_q}$. By Proposition 6.6, $\|q(C)\|_{\{(q(B),0)\}}^{U,m} = \|q(C)\|_{\{(q(D),1)\}}^{U,m} = N^*(m(C) \mid m(D))$. As $\mu_{m(C)}(u) < 1$ for some $u \in U_{\sigma_q}$, $N^*(m(C) \mid m(D)) = 0$. Hence, $\|q(C)\|_{\{(p(A)\rightarrow q(B),\beta)\}}^{U,m} = 0$. Finally, by the triviality axiom, $|q(C)|_{\{(p(A)\rightarrow q(B),\beta)\}}^{U,m} \geq 0$, and thus, $|q(C)|_{\{(p(A)\rightarrow q(B),\beta)\}}^{U,m} \geq \|q(C)\|_{\{(p(A)\rightarrow q(B),\beta)\}}^{U,m}$.

Finally, suppose that for each PGL$^+$ program $\mathcal{P}' = (P', U, m)$ satisfying the modularity and the context constrains and containing at most n clauses, it holds that $\|s\|_{P'}^{U,m} \leq |s|_{P'}$ for each PGL$^+$ goal s. And suppose that P_q^+ contains $n+1$ clauses.

As we are assuming that q occurs in P_q^+, by Proposition 6.5, P_q^+ must contain at least a clause (ψ, γ) such that the head of ψ is q and (ψ, γ) is a basic clause of P_q^+. Therefore, as (P_q^+, U, m) is a PGL$^+$ program satisfying the modularity and the context constraints, we have that $\mathcal{P}' = (P', U, m)$ with $P' = P_q^+ \backslash \{(\psi, \gamma)\}$ is a PGL$^+$ program satisfying the modularity and the context constraints and containing n clauses. Now, we distinguish two cases depending on the form of ψ.

Case $(\psi, \gamma) = (q(B), \gamma)$. Let D_q' be an object constant of sort σ_q such that

$$\mu_{m(D_q')} = \bigwedge\{\mu_{m(D)} \mid P' \models_{PGL^+}^{U,m} (q(D), 1)\}.$$

By Proposition 6.18, $\|q(C)\|_{P_q^+}^{U,m} = \|q(C)\|_{\{(q(D_q'),1),(q(B),\gamma)\}}^{U,m}$. Now, let E be an object constant of sort σ_q such that

$$\mu_{m(E)} = \min(\mu_{m(D_q')}, \max(1-\gamma, \mu_{m(B)})).$$

By Propositions 6.11 and 6.6,

$$\|q(C)\|_{\{(q(D_q'),1),(q(B),\gamma)\}}^{U,m} = \|q(C)\|_{\{(q(E),1)\}}^{U,m} = N^*(m(C) \mid m(E)).$$

On the other hand, by Proposition 6.9, $P' \models_{PGL^+}^{U,m} (q(D_q'), 1)$, and thus, $\|q(D_q')\|_{P'}^{U,m} = 1$. Then, by the induction hypothesis, $\|q(D_q')\|_{P'}^{U,m} \leq |q(D_q')|_{P'}^{U,m} \leq |q(D_q')|_{P_q^+}^{U,m}$, and thus, $P_q^+ \vdash_{PGL^+}^{U,m} (q(D_q'), 1)$. Now, as $(q(B), \gamma) \in P_q^+$ and $\mu_{m(E)} = \min(\mu_{m(D_q')}, \max(1-\gamma, \mu_{m(B)}))$, applying the IN inference rule, we get that $P_q^+ \vdash_{PGL^+}^{U,m} (q(E), 1)$. Finally, applying the

SU inference rule, $\{(q(E),1)\} \vdash^{U,m}_{PGL^+} (q(C), N^*(m(C) \mid m(E)))$. Hence, $|q(C)|^{U,m}_{P^+_q} = \sup\{\alpha \in [0,1] \mid P^+_q \vdash^{U,m}_{PGL^+} (q(C),\alpha)\} \geq N^*(m(C) \mid m(E))$, and thus, $\|q(C)\|^{U,m}_{P^+_q} \leq |q(C)|^{U,m}_{P^+_q}$.

Case $(\psi,\gamma) = (p(A) \rightarrow q(B),\gamma)$**.** Let D'_q be an object constant of sort σ_q such that

$$\mu_{m(D'_q)} = \bigwedge\{\mu_{m(D)} \mid P' \models^{U,m}_{PGL^+} (q(D),1)\}.$$

As $(p(A) \rightarrow q(B),\gamma)$ is a basic clause of P^+_q, by Corollary 6.3 and Proposition 6.4, $\|q(C)\|^{U,m}_{P^+_q} = \|q(C)\|^{U,m}_{\{(q(D'_q),1),(q(B),\min(\gamma,\|p(A)\|^{U,m}_{P^+_p}))\}}$, where, as q depends on p in P, $P^+_p = \{(\phi,\delta) \in P^+_q \mid \text{head}(\phi) = p$ or p depends on $\text{head}(\phi)$ in $P^+_q\}$. Now, let E be an object constant of sort σ_q such that

$$\mu_{m(E)} = \min(\mu_{m(D'_q)}, \max(1 - \min(\gamma, \|p(A)\|^{U,m}_{P^+_p}), \mu_{m(B)})).$$

By Propositions 6.11 and 6.6, $\|q(C)\|^{U,m}_{\{(q(D'_q),1),(q(B),\min(\gamma,\|p(A)\|^{U,m}_{P^+_p}))\}} = \|q(C)\|^{U,m}_{\{(q(E),1)\}} = N^*(m(C) \mid m(E))$.

On the other hand, by Proposition 6.9, $P' \models^{U,m}_{PGL^+} (q(D'_q),1)$, and thus, $\|q(D'_q)\|^{U,m}_{P'} = 1$. Then, by the induction hypothesis, $\|q(D'_q)\|^{U,m}_{P'} \leq |q(D'_q)|^{U,m}_{P'} \leq |q(D'_q)|^{U,m}_{P^+_q}$, and thus, $P^+_q \vdash^{U,m}_{PGL^+} (q(D'_q),1)$. Moreover, as (P^+_p,U,m) is a PGL$^+$ program satisfying the modularity and the context constraints and containing at most n clauses, by the induction hypothesis, we get that $\|p(A)\|^{U,m}_{P^+_p} \leq |p(A)|^{U,m}_{P^+_p} \leq |p(A)|^{U,m}_{P^+_q}$. And, by Corollary 6.1, $P^+_p \models^{U,m}_{PGL^+} (p(A), \|p(A)\|^{U,m}_{P^+_p})$, and thus, $P^+_q \vdash^{U,m}_{PGL^+} (p(A),\lambda)$, for some $\lambda \geq \|p(A)\|^{U,m}_{P^+_p}$. Finally, applying the MP inference rule, $\{(p(A),\lambda),(p(A) \rightarrow q(B),\gamma)\} \vdash^{U,m}_{PGL^+} (q(B),\min(\gamma,\lambda))$. As $\mu_{m(E)} = \min(\mu_{m(D'_q)}, \max(1 - \min(\gamma, \|p(A)\|^{U,m}_{P^+_p}), \mu_{m(B)})) \geq \min(\mu_{m(D'_q)}, \max(1 - \min(\gamma,\lambda), \mu_{m(B)}))$, applying the UN inference rule, $\{(q(D'_q),1),(q(B),min(\gamma,\lambda)\} \vdash^{U,m}_{PGL^+} (q(E),1)$. And, applying the SU inference rule, $\{(q(E),1)\} \vdash^{U,m}_{PGL^+} (q(C), N^*(m(C) \mid m(E)))$. Hence, $|q(C)|^{U,m}_{P^+_q} = \sup\{\alpha \in [0,1] \mid P^+_q \vdash^{U,m}_{PGL^+} (q(C),\alpha)\} \geq N^*(m(C) \mid m(E))$, and thus, $\|q(C)\|^{U,m}_{P^+_q} \leq |q(C)|^{U,m}_{P^+_q}$.

■

6.5 Automated deduction

In this section, we define an automated deduction system for non-recursive and satisfiable PGL$^+$ programs satisfying the context constraint. The definition of

an efficient algorithm for checking the satisfiability of PGL$^+$ programs is out the scope of this thesis.

As we have already pointed out, the proof method for this restricted class of PGL$^+$ programs can be divided into three different and sequential steps. A pre-processing step which extends a set of PGL$^+$ clauses with all valid clauses. A translation step which computes a set of unit PGL$^+$ clauses. Since we are interested in the possibilistic entailment degree of a PGL$^+$ goal, the set of unit PGL$^+$ clauses is semantically equivalent to the original one whenever the set of valid clauses satisfies the context constraint. And, finally, a deduction step which computes the possibilistic entailment degree of a PGL$^+$ goal from the set of unit PGL$^+$ clauses.

6.5.1 Completing the knowledge base

Given a non-recursive PGL$^+$ program $\mathcal{P} = (P, U, m)$, in this section we define an algorithm for computing the set of valid clauses of P in the context determined by U and m. The algorithm is based on two basic operations and two auxiliary functions which apply the RE and FU inferences rules over the clauses of P.

The chaining operation:

$\mathit{chaining}_{U,m}$: Set of PGL$^+$ clauses $\times$ PGL$^+$ clause $\rightarrow$ Set of PGL$^+$ clauses

$\mathit{chaining}_{U,m}(S, (q(B), \alpha)) = \emptyset$

$\mathit{chaining}_{U,m}(S, (\varphi \wedge p(A) \rightarrow q(B), \alpha)) = \{(\varphi \wedge \psi \rightarrow q(B), \min(\alpha, \beta)) \mid (\psi \rightarrow p(A'), \beta) \in S \text{ and } \mu_{m(A')} \leq \mu_{m(A)}\} \cup \{(\psi \wedge \varphi \wedge p(A) \rightarrow r(C), \min(\alpha, \beta)) \mid (\psi \wedge q(B') \rightarrow r(C), \beta) \in S \text{ and } \mu_{m(B)} \leq \mu_{m(B')}\}$

The fusion operation:

$\mathit{fusion}_{U,m}$: Set of PGL$^+$ clauses $\times$ PGL$^+$ clause $\rightarrow$ Set of PGL$^+$ clauses

$\mathit{fusion}_{U,m}(S, (q(B), \alpha)) = \emptyset$

$\mathit{fusion}_{U,m}(S, (\varphi \wedge p(A_1) \rightarrow q(B_1), \alpha)) = \{(\varphi \wedge \psi \wedge p(A_3) \rightarrow q(B_3), \min(\alpha, \beta)) \mid (\psi \wedge p(A_2) \rightarrow q(B_2), \beta) \in S \text{ and } \mu_{m(A_1)} \not\leq \mu_{m(A_2)} \text{ and } \mu_{m(A_2)} \not\leq \mu_{m(A_1)} \text{ and } \mu_{m(A_3)} = \max(\mu_{m(A_1)}, \mu_{m(A_2)}) \text{ and } \mu_{m(B_3)} = \max(\mu_{m(B_1)}, \mu_{m(B_2)})\}$

function *chaining*

input
S_1, S_2 : Set of PGL$^+$ clauses
U : Collection of non-empty domains
m : Interpretation of object constants over U
`/* PGL+ program context` $\mathcal{M}_{U,m}$ `*/`

output
S_3 : Set of PGL$^+$ clauses

auxiliary variables

C : PGL$^+$ clause
S : Set of PGL$^+$ clauses

begin
while ($S_2 \neq \emptyset$) **do**
$C := \mathit{select\_clause}(S_2)$;
$S := \mathit{chaining}_{U,m}(S_1, C)$;
$S_1 := S_1 \cup \{C\}$;
$S_2 := (S_2 \backslash \{C\}) \cup (S \backslash S_1)$;
end while
$S_3 := S_1$;
return(S_3)
end function *chaining*

function *fusion*

input
S_1, S_2 : Set of PGL$^+$ clauses
U : Collection of non-empty domains
m : Interpretation of object constants over U
`/* PGL+ program context` $\mathcal{M}_{U,m}$ `*/`

output
S_3 : Set of PGL$^+$ clauses

auxiliary variables
C : PGL$^+$ clause
S : Set of PGL$^+$ clauses

begin
while ($S_2 \neq \emptyset$) **do**
$C := \mathit{select\_clause}(S_2)$;
$S := \mathit{fusion}_{U,m}(S_1, C)$;
$S_1 := S_1 \cup \{C\}$;
$S_2 := (S_2 \backslash \{C\}) \cup (S \backslash S_1)$;
end while
$S_3 := S_1$;
return(S_3)
end function *fusion*

Algorithm 6.1 completing the knowledge base

Input: (P, U, m) : A non-recursive PGL$^+$ program

Output: P^+ : The set of valid clauses of P

Auxiliary variables:

exists_new_valid_clauses : boolean

C, F : Set of PGL$^+$ clauses

Method:

C := *chaining*($\emptyset$, P, U, m)
F := *fusion*($\emptyset$, C, U, m)
if ($C = F$) **then**
  /* C satisfies the modularity constraint */
  $P^+ := C$;
else
  exists_new_valid_clauses := true;
  while (*exists_new_valid_clauses*) **do**
    C:= *chaining*(C, $F \backslash C$, U, m);
    if ($C = F$) **then**
      /* F satisfies the modularity constraint */
      $P^+ := F$;
      exists_new_valid_clauses := false;
    else
      F := *fusion*(F, $C \backslash F$, U, m);
      if ($C = F$) **then**
        /* C satisfies the modularity constraint */
        $P^+ := C$;
        exists_new_valid_clauses := false;
      end if
    end if
  end while
end if

Final treatment: return(P^+ **)**

Given a non-recursive PGL$^+$ program $\mathcal{P} = (P,U,m)$, the algorithm for completing the knowledge base computes, by means of the *chaining* function, the set of valid clauses that can be derived from P by applying the RE inference rule. From this new set of valid clauses the algorithm computes, by means of the *fusion* function, all valid clauses that can be derived by applying the FU inference rule. As the FU inference rule stretches the body of rules, if the fusion step derives some new valid clauses, the algorithm checks if a new set of valid clauses can be derived from them by applying the RE inference rule. In that case, as the RE inference rule modifies the body and the head of rules, the algorithm checks if a new set of valid clauses can be derived by applying the FU

inference rule. This process is performed until either the *chaining* or the *fusion* functions do not compute new valid clauses.

On the other hand, by the proof of Proposition 6.5, for each valid clause (φ, α) of P either (φ, α) is a basic clause of P, or there exists at least a finite sequence $C_1, C_2, \ldots C_m$ of valid clauses of P such that $C_m = (\varphi, \alpha)$, $C_1 \in P \backslash \{(\varphi, \alpha)\}$, $C_2 \in P \backslash \{(\varphi, \alpha)\}$ and, for each $i \in \{3, \ldots, m\}$, either $C_i \in P \backslash \{(\varphi, \alpha)\}$ or C_i can be obtained by applying the RE or FU inference rules to previous clauses in the sequence. And, for each $i \in \{1, \ldots, m-1\}$, C_i cannot be obtained from (φ, α), and thus, the *chaining* and *fusion* functions cannot produce infinite loops. However, as a valid clause can be derived from two or more different sequences, the *chaining* and *fusion* functions check if each valid clause has already been derived. Finally, as each valid clause of P is either a basic clause or can be derived at least from two clauses of P, in the worst-case, each combination of clauses of P derives a different valid clause. Hence, as P is a finite set of PGL$^+$ clauses, denoting by N the number of clauses of P, in the worst-case, the number of valid clauses is $N + \sum_{i=2}^{N} \binom{N}{i} \in \Theta(\frac{N^{N/2}}{N/2})$. However, only a reduced set of PGL$^+$ clauses of P can combined to derive new valid clauses. Then, the algorithm does not systematically check all possible combinations, but only extends valid clauses which have been previously computed. Thus, the algorithm checks if three different clauses C_1, C_2 and C_3 of P derive a new valid clause whenever either C_1 and C_2, or C_1 and C_3, or C_2 and C_3 have already derived a valid clause different to C_1, C_2 and C_3.

6.5.2 Transforming the knowledge base

Given a non-recursive and satisfiable PGL$^+$ program $\mathcal{P} = (P, U, m)$ and its set of valid clauses P^+, in this section we define an algorithm for computing a set of unit PGL$^+$ clauses which is equivalent to P^+, for determining the maximum degree of possibilistic entailment of a goal, whenever $\mathcal{P}$ satisfies the context constraint. Thus, if $\mathcal{P}$ satisfies the context constraint, the algorithm determines, for each predicate symbol q of type (σ_q) appearing in P, the object constant D_q of sort σ_q such that

$$\mu_{m(D_q)} = \bigwedge \{\mu_{m(D)} \mid P^+ \models^{U,m}_{PGL^+} (q(D), 1)\}.$$

Moreover, the algorithm computes the set of predicate symbols appearing in P for which one can ensure the proof method is complete. Thus, the algorithm determines, for each predicate symbol q appearing in P, if the object constant D_q can be computed, by applying the MP and SU inference rules, just from the set of clauses of P^+ such that their heads are q or q depends on their heads in P.

Algorithm 6.2 transforming the knowledge base

Input:

(P, U, m) : A non-recursive and satisfiable PGL$^+$ program

P^+ : The set of valid clauses of P

$I_{\text{sat}} = (U, i_{\text{sat}}, m)$: $I_{\text{sat}} \in \mathcal{I}_{U,m}$ such that, for each clause $(\phi, \alpha) \in P$ with $\alpha > 0$, $I_{\text{sat}}(\phi) = 1$

Output:

$Pred$: The set of predicate symbols appearing in P for which the proof method is complete

$\mathcal{D}$: $\{D_q$ object constant $\mid \mu_{m(D_q)} = 1$, if $q \notin Pred$, and $\mu_{m(D_q)} = \bigwedge\{\mu_{m(D)} \mid P \models^{U,m}_{PGL^+} (q(D), 1)\}$, otherwise$\}$

Auxiliary variables:

q : predicate symbol

D_q : object constant

Method:

$Pred := \{q$ predicate symbol appearing in $P\}$;
$\mathcal{D} := \emptyset$;
while ($P^+ \neq \emptyset$) **do**
  $q :=$ *select_independent_predicate_symbol*(P^+);
  if (P^+ contains no PGL$^+$ clause with head q) **then**
    $D_q :=$ object constant such that $\mu_{m(D_q)} = 1$;
  else
    $D_q :=$ object constant such that
      $\mu_{m(D_q)} = \min\{\max(1 - \beta, \mu_{m(B)}) \mid (q(B), \beta) \in P^+\}$;
  end if
  $\mathcal{D} := \mathcal{D} \cup \{D_q\}$;
  $P^+ := P^+ \backslash \{(q(B), \beta) \in P^+\}$;
  if ($q \in Pred$ and, for some clause $(q(F) \wedge \varphi \rightarrow t(G), \gamma) \in P$ and value $u \in U_{\sigma_q}$, $\mu_{m(F)}(u) > \mu_{m(G)}(i_{\text{sat}}(t))$ and $\mu_{m(D_q)}(u) > \max(1 - \gamma, \mu_{m(G)}(i_{\text{sat}}(t)))$)
  then `/* checking the context constraint C1 for` q `*/`
    $Pred := Pred \backslash \{q\}$;
  end if
  for (each clause $(q(H) \wedge \psi \rightarrow r(I), \delta) \in P^+$ such that $r \in Pred$) **do**
    if ($q \notin Pred$) **then**
      $Pred := Pred \backslash \{r\}$;

else
if ($\delta > N^*(m(H) \mid m(D_q))$ and there exists no value $u \in U_{\sigma_q}$
such that $\mu_{m(H)}(u) = 0$ and $\mu_{m(D_q)}(u) = N^*(m(H) \mid m(D_q))$)
then `/* checking the context constraint C2 for` r `*/`
$Pred := Pred \backslash \{r\}$;
end if
end if
end for
$P^+ := P^+ \cup \{(\psi \rightarrow r(I), \min(\delta, N^*(m(H) \mid m(D_q)))) \mid$
$(q(H) \wedge \psi \rightarrow r(I), \delta) \in P^+\}$;
$P^+ := P^+ \backslash \{(q(H) \wedge \psi \rightarrow r(I), \delta) \in P^+\}$;
end while

Final treatment: return($Pred$, $\mathcal{D}$ **)**

The algorithm for transforming a non-recursive and satisfiable PGL$^+$ program $\mathcal{P} = (P, U, m)$ into a set of unit PGL$^+$ clauses is based on Proposition 6.10 which establishes that, given a PGL$^+$ goal $q(C)$,

$$\|q(C)\|_P^{U,m} = \|q(C)\|_{\{(q(D_q),1)\}}^{U,m},$$

where D_q is an object constant such that

$$\mu_{m(D_q)} = \bigwedge \{\mu_{m(D)} \mid P \models_{PGL^+}^{U,m} (q(D), 1)\}.$$

Moreover, by Proposition 6.4,

$$\|q(C)\|_P^{U,m} = \|q(C)\|_{P^+}^{U,m},$$

where P^+ is the set of valid clauses of P in the context determined by U and m. P^+ is computed by means of the completion algorithm. Then, if a PGL$^+$ program satisfies the context constraint C1, by Proposition 6.12, for each predicate symbol q appearing in P^+ the object constant D_q can be computed from the set of clauses of P^+ such that their heads are q or q depends on their heads in P^+. And, if a PGL$^+$ program satisfies the context constraint C2, by Proposition 6.17, each general PGL$^+$ clause of P^+ such that it head is q can be transformed into a unit PGL$^+$ clause by applying the SU and MP inference rules. Finally, by Proposition 6.11, D_q can be computed from a finite set of unit PGL$^+$ clauses by applying the UN and IN inference rules.

On the other hand, as a non-recursive PGL$^+$ program does not contain recursive formulas, there exists at least a predicate symbol q which depends on no other predicate symbol in P^+. And, after applying the replacement step, based on Proposition 6.17, for each predicate symbol depending on q in P^+,

there exists at least a predicate symbol $s \neq q$ which depends on no other predicate symbol in P^+. The *select_independent_predicate_symbol* function selects this predicate symbol for each algorithm iteration. Hence, if a PGL$^+$ program satisfies the context constraint C1 for q and constraints C1 and C2 for s, the object constants D_q and D_s computed by the algorithm are

$$\bigwedge\{\mu_{m(D)} \mid P \models^{U,m}_{PGL^+} (q(D), 1)\}$$

and

$$\bigwedge\{\mu_{m(D)} \mid P \models^{U,m}_{PGL^+} (s(D), 1)\},$$

respectively. Thus, repeating the above process, the algorithm determines, for each predicate symbol of P^+, if the proof method is complete and computes the associated unit PGL$^+$ clause. Finally, the time complexity of the algorithm for computing the set of unit PGL$^+$ clauses is linear in the total number of occurrences of predicates symbols in P^+. And, when extending a PGL$^+$ program with unit PGL$^+$ clauses, only the equivalent set of unit clauses must be computed again, and thus, once we have computed the set of valid clauses of a PGL$^+$ program we must perform the completion algorithm iff new general PGL$^+$ clauses are added to the knowledge base.

6.5.3 Computing the maximum degree of deduction

Given a non-recursive and satisfiable PGL$^+$ program $\mathcal{P} = (P, U, m)$, in Sections 6.5.1 and 6.5.2, we have defined algorithms for computing the set of valid clauses of P and transforming this set into a set of unit PGL$^+$ clauses, respectively. Moreover, if $\mathcal{P}$ satisfies the context constraint for each predicate symbol of P, the computed set of unit PGL$^+$ clauses is semantically equivalent to P for determining the maximum degree of possibilistic entailment of a PGL$^+$ goal. And, in that case, if $(q(D_q), 1)$ is the unit PGL$^+$ clause associated with the predicate symbol q of P, by Propositions 6.10 and 6.6,

$$\|q(C)\|^{U,m}_P = \|q(C)\|^{U,m}_{\{(q(D_q),1)\}} = N^*(m(C) \mid m(D_q)),$$

for any object constant C. On the other hand, if $\mathcal{P}$ does not satisfy the context constraint for some predicate symbol q, obviously, the object constant D_q computed by Algorithm 6.2 is

$$\bigwedge\{\mu_{m(D)} \mid P \vdash^{U,m}_{PGL^+} (q(D), 1)\}.$$

Hence,

$$|q(C)|^{U,m}_P = N^*(m(C) \mid m(D_q))$$

and, if $\mathcal{P}$ satisfies the context constraint for the predicate symbol q, we can ensure that

$$|q(C)|^{U,m}_P = \|q(C)\|^{U,m}_P.$$

Therefore, after applying Algorithms 6.1 and 6.2 to a non-recursive and satisfiable PGL$^+$ program, the maximum degree of deduction of a PGL$^+$ goal can

be computed in a constant time complexity in the sense that it is equivalent to compute the partial matching between two fuzzy constants.

Algorithm 6.3 computing the maximum degree of deduction

Input:

(P, U, m) : A non-recursive and satisfiable PGL$^+$ program

$Pred$: The set of predicate symbols appearing in P for which the maximum degree of deduction is the maximum degree of possibilistic entailment

$\mathcal{D}$: $\{D_q$ object constant $\mid$
$\mu_{m(D_q)} = \bigwedge\{\mu_{m(D)} \mid P \models^{U,m}_{PGL^+} (q(D), 1)\}$ if $q \in Pred$,
and $\mu_{m(D_q)} = \bigwedge\{\mu_{m(D)} \mid P \vdash^{U,m}_{PGL^+} (q(D), 1)\}$, otherwise$\}$

$q(C)$: PGL$^+$ goal

Output:

$|q(C)|^{U,m}_P$: $\sup\{\alpha \in [0,1] \mid P \vdash^{U,m}_{PGL^+} (q(C), \alpha)\}$

maximum_degree_of_possibilistic_entailment : boolean

`/* true if we can ensure that` $|q(C)|^{U,m}_P = \|q(C)\|^{U,m}_P$ `*/`

Method:

if (for each $u \in U_{\sigma_q}$ $\mu_{m(C)}(u) = 1$) **then**

`/* checking the first particular case */`

$|q(C)|^{U,m}_P := 1$;

maximum_degree_of_possibilistic_entailment := true;

else

if (q does not appear in P) **then**

`/* checking the second particular case */`

$|q(C)|^{U,m}_P := 0$;

maximum_degree_of_possibilistic_entailment := true;

else

$|q(C)|^{U,m}_P := N^*(m(C) \mid m(D_q))$;

maximum_degree_of_possibilistic_entailment := $q \in Pred$;

end if

end if

Final treatment:

return($|q(C)|^{U,m}_P$, *maximum_degree_of_possibilistic_entailment* **)**

Chapter 7

An automated deduction system for a first-order extension of PGL$^+$

7.1 Introduction

In Chapter 5, we defined PGL, a general propositional fuzzy possibilistic logic based on Gödel infinitely-valued logic. Then, in Chapter 6 we defined PGL$^+$, a fuzzy possibilistic logic programming language based on the Horn-rule fragment of PGL extended with fuzzy constants and a semantical unification mechanism. Now in this chapter we define a first-order extension of PGL$^+$.

To achieve our objective, analogously to what we did with the propositional case, we first define a general first-order possibilistic logic with fuzzy constants based on Gödel predicate logic (called PGL$^+\forall$). Then, for logic programming purposes, we focus our attention on a first-order Horn-rule fragment of PGL$^+\forall$. Finally, we provide a translator of PGL$^+\forall$ programs into machine code by extending the Warren Abstract Machine to our fuzzy and possibilistic context.

The chapter is organized in four parts. In the first part, Section 7.2, we present the syntax and the semantics of PGL$^+\forall$. In the second part, Section 7.3, we provide the Horn-rule fragment of PGL$^+\forall$ with a sound modus ponens-style calculus including several unification rules, a merging and a weight weakening rules. In the third part, Section 7.4, we define an automated deduction system based on the above calculus. In Section 7.4.1, we develop a directional algorithm for computing a most general fuzzy unifier of a pair of atomic formulas; in Section 7.4.2, we describe a proof procedure oriented to program queries that applies the generalized modus ponens inference rule in a reverse way by using a depth-first strategy; and, in Section 7.4.3, we apply these algorithms to the construction of a proof tree when general and specific fuzzy constants have to be unified. In the last part, Section 7.5, we provide a compiler for PGL$^+\forall$ programs,

and a user-friendly environment which has been implemented to facilitate the graphical representation of fuzzy constants and compilation tasks.

7.2 A first-order possibilistic logic with fuzzy constants based on Gödel predicate logic

In this section, we collect together the main definitions of the syntax and the semantics of a general first-order fuzzy possibilistic logic based on Gödel predicate logic, denoted hereafter as PGL$^+\forall$.

7.2.1 Gödel predicate logic extended with fuzzy constants

Throughout this section, we describe the syntax and the many-valued semantics of Gödel predicate logic extended with fuzzy constants, denoted hereafter as G$^+\forall$.

The *basic components* of G$^+\forall$ are:

- *Sorts* of variables and constants. We distinguish a *basic sort* σ from its corresponding (imprecise and fuzzy) *extended sort* $f\sigma$. A *type* is a tuple of sorts.
- A set $\mathcal{X}$ of object *variables* and a set $\mathcal{C}$ of object *constants* (crisp and fuzzy constants), each having its sort.
- A set *Var* of *primitive propositions*.
- A set *Pred* of *regular predicates*, each one having a type.
- *Connectives* $\wedge$, $\rightarrow$.
- *Quantifiers* $\forall$, $\exists$.
- *Truth* constants $\top$, $\bot$.

Definition 7.1 (G$^+\forall$ term) *A G$^+\forall$ term is either an object variable from $\mathcal{X}$ or an object constant from $\mathcal{C}$.*

Definition 7.2 (G$^+\forall$ atomic formula) *A G$^+\forall$ atomic formula is either a truth constant, or a primitive proposition from Var, or of the form $p(t_1, ..., t_n)$, where p is a predicate symbol of arity n from Pred and $t_1, ..., t_n$ are terms such that the sorts of $t_1, ..., t_n$ correspond to the type of p.*

G$^+\forall$ *formulas* are built from atomic formulas using connectives and quantifiers in the usual way.

Next, we define the semantics of G$^+\forall$ formulas, which, due to the presence of fuzzy constants, is many-valued, instead of Boolean.

Definition 7.3 (disjunctive many-valued interpretation) *A disjunctive many-valued interpretation $\mathbf{M} = (U, i, m)$ maps:*

1. *each basic sort σ into a non-empty domain U_σ and each extended sort $f\sigma$ into the set $F(U_\sigma)$ of imprecise and fuzzy sets of U_σ;*

2. *a primitive proposition into a truth value of the unit interval* $[0,1]$*;*

3. *a predicate p of type $(\sigma_1,\ldots,\sigma_k,f\sigma_{k+1},\ldots,f\sigma_n)$ into a* crisp *relation*

$$i(p) \subseteq U_{\sigma_1} \times \cdots \times U_{\sigma_n}$$

such that, for each $(u_1,\ldots,u_k) \in U_{\sigma_1} \times \cdots \times U_{\sigma_k}$, there exist unique values $(u_{k+1},\ldots,u_n) \in U_{\sigma_{k+1}} \times \cdots \times U_{\sigma_n}$ and $(u_1,\ldots,u_n) \in i(p)$; and

4. *a (precise) object constant c of sort σ into a value $m(c) \in U_\sigma$, and an (imprecise or fuzzy) object constant A of sort $f\sigma$ into a (normalized fuzzy) set $m(A) \in F(U_\sigma)$. We denote the membership function of $m(\cdot)$ by $\mu_{m(\cdot)}$. The value $m(c) \in U_\sigma$ is also represented by a fuzzy set, given by $\mu_{m(c)}(m(c)) = 1$, and $\mu_{m(c)}(u) = 0$ for each $u \in U_\sigma$ such that $u \neq m(c)$.*

We denote the set of all possible disjunctive many-valued interpretations by $\mathcal{M}$.

Remark that a many-valued interpretation $\mathbf{M} = (U,i,m)$ is a disjunctive interpretation in the sense that if p is a predicate of type $(f\sigma_1,\ldots,f\sigma_n)$, then there exist unique values $(u_1,\ldots,u_n) \in U_{\sigma_1} \times \cdots \times U_{\sigma_n}$ such that $i(p) = \{(u_1,\ldots,u_n)\}$, and if p is a predicate of type $(\sigma_1,f\sigma_2\ldots,f\sigma_n)$, then for each value $u_1 \in U_{\sigma_1}$ there exist unique values $(u_2,\ldots,u_n) \in U_{\sigma_2} \times \cdots \times U_{\sigma_n}$ such that $(u_1,u_2,\ldots,u_n) \in i(p)$.

Definition 7.4 (fuzzy evaluation of variables) *An evaluation of variables is a mapping v assigning to each variable x of sort σ an element $v(x) \in U_\sigma$. If x is of sort $f\sigma$, then $v(x)$ is a (normalized fuzzy) set of $F(U_\sigma)$. As above, we denote the membership function of $v(x)$ by $\mu_{v(x)}$, where $v(x)$ is either an element of U_σ or a (normalized fuzzy) set of $F(U_\sigma)$.*

In contrast to PLFC, in PGL$^+\forall$ the unification between fuzzy constants is allowed, and thus, variables have to be evaluated to fuzzy constants.

Definition 7.5 (truth value of a G$^+\forall$ atomic formula) *The truth value of a $G^+\forall$ atomic formula φ under an interpretation $\mathbf{M} = (U,i,m)$ and an evaluation v of variables, denoted by $\|\varphi\|_{\mathbf{M},v}$, is the value 1 if φ is the truth constant $\top$, is the value 0 if φ is the truth constant $\bot$, is just the truth value $i(q)$ if φ is a primitive proposition q, and it is computed as*

$$\sup_{(\ldots,u,\ldots,w,\ldots)\in i(p)} \min(\ldots,\mu_{v(x)}(u),\ldots,\mu_{m(c)}(w),\ldots)$$

if φ is of the form $p(\ldots,x,\ldots,c,\ldots)$, where x is an object variable and c is an object constant.

Notice that $\|p(\ldots,x,\ldots,c,\ldots)\|_{\mathbf{M},v}$ may lie somewhere in the unit interval $[0,1]$ as soon as p contains some fuzzy constant or v evaluates some variable to a fuzzy set. Moreover, if p is a predicate of type $(f\sigma_1,\ldots,f\sigma_n)$, then $i(p)=\{(\ldots,u,\ldots,w,\ldots)\}$ and

$$\|p(\ldots,x,\ldots,c,\ldots)\|_{\mathbf{M},v}=\min(\ldots,\mu_{v(x)}(u),\ldots,\mu_{m(c)}(w),\ldots).$$

The above truth value extends to the value $\|\varphi\|_{\mathbf{M},v}$, for each G$^+\forall$ formula φ, in the usual way by means of the min-conjunction and the Gödel's many-valued implication. Thus,

$$\begin{array}{rcl}
\|\varphi\wedge\psi\|_{\mathbf{M},v} & = & \min(\|\varphi\|_{\mathbf{M},v},\|\psi\|_{\mathbf{M},v}),\\
\|\varphi\rightarrow\psi\|_{\mathbf{M},v} & = & \begin{cases} 1, & \text{if } \|\varphi\|_{\mathbf{M},v}\leq\|\psi\|_{\mathbf{M},v}\\ \|\psi\|_{\mathbf{M},v}, & \text{otherwise,}\end{cases}\\
\|(\forall x)\varphi\|_{\mathbf{M},v} & = & \inf\{\|\varphi\|_{\mathbf{M},v'}\mid v'(y)=v(y)\text{ for each variable } y\text{, except } x\},\\
\|(\exists x)\varphi\|_{\mathbf{M},v} & = & \sup\{\|\varphi\|_{\mathbf{M},v'}\mid v'(y)=v(y)\text{ for each variable } y\text{, except } x\}.
\end{array}$$

The *truth value* of a G$^+\forall$ formula φ in a disjunctive many-valued interpretation $\mathbf{M}$ is $\|\varphi\|_{\mathbf{M}}=\inf\{\|\varphi\|_{\mathbf{M},v}\mid v \text{ is an evaluation of variables}\}$; and φ is a *1-tautology* if $\|\varphi\|_{\mathbf{M}}=1$ for each interpretation $\mathbf{M}$. A disjunctive many-valued interpretation $\mathbf{M}$ is a *model* of a set of G$^+\forall$ formulas T if $\|\varphi\|_{\mathbf{M}}=1$ for each $\varphi\in T$. Finally, T entails another G$^+\forall$ formula ϕ if $\|\phi\|_{\mathbf{M}}=1$ for each model $\mathbf{M}$ of T.

Example 7.1 Let $age(\cdot,\cdot)$ be a binary predicate of type $(\texttt{person\_name}, f\texttt{years\_old})$, let *Peter* and *Ruth* be constants of sort `person_name` and let *about_35* be a constant of sort f`years_old`. Let $\mathbf{M}=(U,i,m)$ be a disjunctive many-valued interpretation such that:

1. $U = \{\ U_{\texttt{person\_name}}=\{Peter,Ruth\},$
   $U_{\texttt{years\_old}}=[0,120](\text{years})\ \}$;
2. $i(age)=\{(Peter,37),(Ruth,34)\}$;
3. $m(Peter)=Peter$,
   $m(Ruth)=Ruth$,
   $m(x)=x$, for each real $x\in[0,120](\text{years})$, and
   $m(about\_35)=[30;35;35;40](\text{years})$.

Now consider the G$^+\forall$ atomic formula $age(Peter,x)$ and the following four evaluations of variable x: $v_1(x)=35$, $v_2(x)=37$, $v_3(x)=[30;40;50;60]$ and $v_4(x)=[30;35;40;45]$. Then, we have the following truth values of the atomic formula $age(Peter,x)$ in the interpretation $\mathbf{M}$ under the corresponding variable evaluations:

$$\|age(Peter,x)\|_{\mathbf{M},v_1}=\mu_{\{35\}}(37)=0$$

$$\|age(Peter,x)\|_{\mathbf{M},v_2}=\mu_{\{37\}}(37)=1$$

$$\|age(Peter, x)\|_{\mathbf{M},v_3} = \mu_{[30;40;50;60]}(37) = 0.7$$

$$\|age(Peter, x)\|_{\mathbf{M},v_4} = \mu_{[30;35;40;45]}(37) = 1$$

Now, consider the atomic formula $age(Peter, about\_35)$. In this case, we have no free variable, and the truth value in the interpretation $\mathbf{M}$ itself is:

$$\|age(Peter, about\_35)\|_{\mathbf{M}} = \mu_{m(about\_35)}(37) = \mu_{[30;35;35;40]}(37) = 0.6.$$

Finally, the truth value in the interpretation $\mathbf{M}$ of the G$^+\forall$ formula $((\exists x)age(x, about\_35))$ is computed as:

$$\begin{aligned}
&\|(\exists x)age(x, about_35)\|_{\mathbf{M}} \\
&\quad = \max(\|age(x, about_35)\|_{\mathbf{M},v(x)=Peter}, \|age(x, about_35)\|_{\mathbf{M},v(x)=Ruth}) \\
&\quad = \max(\mu_{m(about_35)}(37), \mu_{m(about_35)}(34)) \\
&\quad = \max(\mu_{[30;35;35;40]}(37), \mu_{[30;35;35;40]}(34)) \\
&\quad = \max(0.6, 0.8) \\
&\quad = 0.8.
\end{aligned}$$

□

7.2.2 Possibilistic reasoning over Gödel predicate logic extended with fuzzy constants

In this section, we extend G$^+\forall$ to enable fuzzy reasoning under possibilistic uncertainty, and thus, we extend G$^+\forall$ formulas with certainty-weights and we define a well-behaved possibilistic semantics on top of G$^+\forall$.

A PGL$^+\forall$ *formula* is a pair of the form (φ, α), where φ is a G$^+\forall$ formula and $\alpha \in [0, 1]$ is a certainty value formalizing that "φ is certain with a necessity of at least α".

In order to define a well-behaved possibilistic semantics on top of G$^+\forall$, as we did for PLFC and PGL$^+$, we need to introduce the notion of context. Let U be a collection of non-empty domains and let m be an interpretation of object constants over U (or over $[0, 1]^U$ in the case of fuzzy constants). We define the PGL$^+\forall$ *context* determined by U and m, denoted by $\mathcal{M}_{U,m}$, as the set of all disjunctive many-valued interpretations $\mathbf{M} \in \mathcal{M}$ having U as a domain and m as an interpretation of object constants.

Then, given a PGL$^+\forall$ context $\mathcal{M}_{U,m}$, a *possibilistic model* is a normalized possibility distribution π on the set of disjunctive many-valued interpretations $\mathcal{M}_{U,m}$.

Definition 7.6 (necessity evaluation) *Let $\mathcal{M}_{U,m}$ be a PGL$^+\forall$ context, let $\pi : \mathcal{M}_{U,m} \to [0, 1]$ be a possibilistic model, and let v be an evaluation of variables over the domain U (in the sense of Definition 7.4). The necessity evaluation of a G$^+\forall$ formula φ given by π under the evaluation of variables v, written $N^*_v([\varphi] \mid \pi)$, is defined as*

$$N^*_v([\varphi] \mid \pi) = \inf_{\mathbf{M} \in \mathcal{M}_{U,m}} \pi(\mathbf{M}) \Rightarrow \|\varphi\|_{\mathbf{M},v},$$

where $\Rightarrow$ is the reciprocal of Gödel's many-valued implication[1].

For the sake of simplicity, when a G$^+\forall$ formula φ contains no free variables we simply define the necessity evaluation of φ given by π as

$$N^*([\varphi] \mid \pi) = \inf_{\mathbf{M} \in \mathcal{M}_{U,m}} \pi(\mathbf{M}) \Rightarrow \|\varphi\|_{\mathbf{M}}.$$

Definition 7.7 (possibilistic satisfiability) *Given a PGL$^+\forall$ context $\mathcal{M}_{U,m}$ and an evaluation of variables v, a possibilistic model $\pi : \mathcal{M}_{U,m} \to [0,1]$ satisfies a PGL$^+\forall$ formula (φ, α) in the context $\mathcal{M}_{U,m}$ under the evaluation v, written $\pi \models^{U,m,v}_{PGL^+\forall} (\varphi, \alpha)$, iff $N^*_v([\varphi] \mid \pi) \geq \alpha$. Furthermore, π satisfies (φ, α) in the context $\mathcal{M}_{U,m}$, written $\pi \models^{U,m}_{PGL^+\forall} (\varphi, \alpha)$, iff $\pi \models^{U,m,v}_{PGL^+\forall} (\varphi, \alpha)$ for each evaluation v of variables.*

When (φ, α) is a PGL$^+\forall$ formula of the form

$$((\forall \bar{x})\varphi(\bar{x}), \alpha),$$

the above definition of possibilistic satisfiability is equivalent to the following one:

$$\pi \models^{U,m}_{PGL^+\forall} ((\forall \bar{x})\varphi(\bar{x}), \alpha) \text{ iff } N^*_v([\varphi(\bar{x})] \mid \pi) \geq \alpha$$

for each evaluation v of variables $\bar{x}$. However, the possibilistic satisfiability of a PGL$^+\forall$ formula of the form

$$((\exists \bar{x})\varphi(\bar{x}), \alpha)$$

cannot be expressed in the same way, i.e. if $N^*_v([\varphi(\bar{x})] \mid \pi) \geq \alpha$ for some evaluation v of variables $\bar{x}$, then $\pi \models^{U,m}_{PGL^+\forall} ((\exists \bar{x})\varphi(\bar{x}), \alpha)$, but the reverse does not hold.

Example 7.2 Consider the context $\mathcal{M}_{U,m}$, where U and m are as in Example 7.1, and the following predicate interpretation mappings:

$$\begin{aligned} i_0(age) &= \{(Peter, 37), (Ruth, 34)\} \\ i_1(age) &= \{(Peter, 35), (Ruth, 40)\} \\ i_2(age) &= \{(Peter, 32), (Ruth, 35)\} \\ i_3(age) &= \{(Peter, 40), (Ruth, 36)\} \end{aligned}$$

and denote $\mathbf{M}_0 = (U, i_0, m)$, $\mathbf{M}_1 = (U, i_1, m)$, $\mathbf{M}_2 = (U, i_2, m)$ and $\mathbf{M}_3 = (U, i_3, m)$. Finally, consider the following possibility distribution π:

$$\pi(\mathbf{M}) = \begin{cases} 0.4, & \text{if } \mathbf{M} = \mathbf{M}_0 \\ 1, & \text{if } \mathbf{M} = \mathbf{M}_1 \\ 0.6, & \text{if } \mathbf{M} = \mathbf{M}_2 \\ 0.2, & \text{if } \mathbf{M} = \mathbf{M}_3 \\ 0, & \text{otherwise} \end{cases}$$

Let us now compute how much the possibilistic model π makes the G$^+\forall$ atomic formula $age(Peter, about\_35)$ certain. Remember that $m(about\_35) = [30; 35; 35; 40]$, and thus, we have:

[1] Remember that $x \Rightarrow y = 1$, if $x \leq y$, and $x \Rightarrow y = 1 - x$, otherwise.

$$\begin{array}{l}\|age(Peter, about\_35)\|_{\mathbf{M}_0} = \mu_{m(about\_35)}(37) = 0.6,\\ \|age(Peter, about\_35)\|_{\mathbf{M}_1} = \mu_{m(about\_35)}(35) = 1,\\ \|age(Peter, about\_35)\|_{\mathbf{M}_2} = \mu_{m(about\_35)}(32) = 0.2, \text{ and}\\ \|age(Peter, about\_35)\|_{\mathbf{M}_3} = \mu_{m(about\_35)}(40) = 0.\end{array}$$

Then,

$$\begin{array}{l}N^*([age(Peter, about\_35)] \mid \pi)\\ \quad = \min(0.4 \Rightarrow 0.6, 1 \Rightarrow 1, 0.6 \Rightarrow 0.2, 0.2 \Rightarrow 0, \{0 \Rightarrow y \mid y \in [0,1]\})\\ \quad = \min(1, 1, 0.4, 0.8, 1) \;=\; 0.4.\end{array}$$

Therefore, for instance, in the current context,

$$\pi \models^{U,m}_{PGL^+\forall} (age(Peter, about\_35), 0.4),$$

but

$$\pi \not\models^{U,m}_{PGL^+\forall} (age(Peter, about\_35), 0.9).$$

Finally, consider the PGL$^+\forall$ formula $((\exists x) age(x, about\_35), 0.9)$. To compute how much the possibilistic model π makes this formula certain we must consider, for each disjunctive many-valued interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$, all possible evaluation of variable x of sort `person_name`, and thus we have:

$$\begin{array}{l}\|(\exists x) age(x, about\_35)\|_{\mathbf{M}_0}\\ \quad = \max(\|(age(x, about\_35)\|_{\mathbf{M}_0, v(x)=Peter}, \|(age(x, about\_35)\|_{\mathbf{M}_0, v(x)=Ruth})\\ \quad = \max(\mu_{m(about\_35)}(37), \mu_{m(about\_35)}(34))\\ \quad = \max(0.6, 0.8)\\ \quad = 0.8.\\ \|(\exists x) age(x, about\_35)\|_{\mathbf{M}_1}\\ \quad = \max(\|(age(x, about\_35)\|_{\mathbf{M}_1, v(x)=Peter}, \|(age(x, about\_35)\|_{\mathbf{M}_1, v(x)=Ruth}\\ \quad = \max(\mu_{m(about\_35)}(35), \mu_{m(about\_35)}(40))\\ \quad = \max(1, 0)\\ \quad = 1.\\ \|(\exists x) age(x, about\_35)\|_{\mathbf{M}_2}\\ \quad = \max(\|(age(x, about\_35)\|_{\mathbf{M}_2, v(x)=Peter}, \|(age(x, about\_35)\|_{\mathbf{M}_2, v(x)=Ruth})\\ \quad = \max(\mu_{m(about\_35)}(32), \mu_{m(about\_35)}(35))\\ \quad = \max(0.2, 1)\\ \quad = 1.\\ \|(\exists x) age(x, about\_35)\|_{\mathbf{M}_3}\\ \quad = \max(\|(age(x, about\_35)\|_{\mathbf{M}_3, v(x)=Peter}, \|(age(x, about\_35)\|_{\mathbf{M}_3, v(x)=Ruth})\\ \quad = \max(\mu_{m(about\_35)}(40), \mu_{m(about\_35)}(36))\\ \quad = \max(0, 0.8)\\ \quad = 0.8.\end{array}$$

Then,

$$\begin{array}{l}N^*([(\exists x) age(x, about\_35)] \mid \pi)\\ \quad = \min(0.4 \Rightarrow 0.8, 1 \Rightarrow 1, 0.6 \Rightarrow 1, 0.2 \Rightarrow 0.8, \{0 \Rightarrow y \mid y \in [0,1]\})\\ \quad = \;1.\end{array}$$

So, in the current context, $N^*([(\exists x)age(x, about\_35)] \mid \pi) \geq 0.9$, and thus,

$$\pi \models^{U,m}_{PGL^+\forall} ((\exists x)age(x, about\_35), 0.9).$$

However, we have that

$$N^*_{v(x)=Peter}([age(x, about\_35)] \mid \pi) = 0.4,$$

and

$$N^*_{v(x)=Ruth}([age(x, about\_35)] \mid \pi) = 0.$$

□

Definition 7.8 (possibilistic entailment) *Let K be a set of $PGL^+\forall$ formulas and let (φ, α) be a $PGL^+\forall$ formula. K entails (φ, α) in a given $PGL^+\forall$ context $\mathcal{M}_{U,m}$, written $K \models^{U,m}_{PGL^+\forall} (\varphi, \alpha)$, iff each possibilistic model $\pi : \mathcal{M}_{U,m} \to [0,1]$ satisfying all the $PGL^+\forall$ formulas in K also satisfies (φ, α).*

We refer to a possibilistic model $\pi : \mathcal{M}_{U,m} \to [0,1]$ satisfying all the PGL$^+\forall$ formulas in a set K as a model of K.

7.3 A first-order possibilistic logic programming language with fuzzy constants

In order to define a first-order automated deduction system, we restrict ourselves to the Horn-rule fragment of PGL$^+\forall$, and thus, the logic programming system is based on two special types of PGL$^+\forall$ formulas: *clauses* and *queries*.

Definition 7.9 (PGL$^+\forall$ clause) *A clause is a $PGL^+\forall$ formula of the type $((\forall \bar{x})(p_1 \wedge \cdots \wedge p_k \to q), \alpha)$, where $p_1, \ldots, p_k, q$ are $G^+\forall$ atomic formulas such that all their variables, denoted as $\bar{x}$, are universally quantified.*

When the body of a clause is the truth constant $\top$, we talk of a PGL$^+\forall$ *unit* clause. They have the form $((\forall \bar{x})(\top \to q), \alpha)$ and are simply written $((\forall \bar{x})(\to q), \alpha)$, where $\bar{x}$ denotes the set of variables of the G$^+\forall$ atomic formula q.

Definition 7.10 (PGL$^+\forall$ query) *A query is a $PGL^+\forall$ formula of the type $((\exists \bar{x})(p_1 \wedge \cdots \wedge p_n), \alpha)$, where $p_1, \ldots, p_n$ are $G^+\forall$ atomic formulas such that all their variables, denoted as $\bar{x}$, are existentially quantified.*

When the G$^+\forall$ formula of a query is the truth constant $\top$, we talk of the *truth* query.

PGL$^+\forall$ clauses allow us to represent fuzzy information of the real world under possibilistic uncertainty. For instance, the statements

"it is almost certain that the book is cheap"

and

"it is more or less certain that when a book is cheap, people buy it",

are represented in this framework respectively as

$$(\rightarrow price(book, cheap), 0.9) \quad \text{and} \quad (price(book, cheap) \rightarrow buy(book), 0.7),$$

where $price(\cdot,\cdot)$ and $buy(\cdot)$ are regular predicates of type $(\texttt{product}, f\texttt{product\_price})$ and $(\texttt{product})$, respectively; *book* is an object constant of sort `product`; and *cheap* is a fuzzy constant of sort f`product_price`. On the other hand, PGL$^+\forall$ queries allow us to ask about the possibilistic uncertainty of fuzzy information from a knowledge base. For instance, the queries

"can we be almost certain that there is something to buy?"

and

"can we be more or less certain that there is something cheap?",

are represented in this framework respectively as

$$((\exists x) buy(x), 0.9) \quad \text{and} \quad ((\exists x) price(x, cheap), 0.7).$$

From now on, for the sake of simplicity, PGL$^+\forall$ formulas refer to PGL$^+\forall$ clauses and queries.

At this point we are interested in defining a sound proof method for PGL$^+\forall$ queries based on a fuzzy unification mechanism between fuzzy constants. To this end, we first tried to define a refutation proof method by resolution, but we failed because refutation itself is not sound with respect to the possibilistic entailment of PGL$^+\forall$ formulas. Let us consider the following grounded G$^+\forall$ atomic formulas:

A1: $weight(between\_54\_56)$

A2: $weight(about\_55)$

where *between_54_56* and *about_55* are two fuzzy constants of sort f`kilograms`, and the following PGL$^+\forall$ context:

1. $U = \{U_{\texttt{kilograms}} = [0, 200](\text{kilograms})\}$;
2. $m(between\_54\_56) = [52; 54; 56; 58]$, and
   $m(about\_55) = [50; 55; 55; 60]$.

As $\mu_{m(between\_54\_56)}(54) = 1 > \mu_{m(about\_55)}(54)$,

$$\inf_{u \in [0,200]} \mu_{m(between\_54\_56)}(u) \Rightarrow \mu_{m(about\_55)}(u) = 0,$$

and thus, the PGL$^+\forall$ query $(weight(about\_55), \alpha)$ cannot be a logical consequence of the PGL$^+\forall$ unit clause $(\rightarrow weight(between\_54\_56), 1)$ if $\alpha > 0$. On the other hand, the negation of the query should be of the form

$$(weight(about\_55) \rightarrow \bot, 1).$$

Then, it is easy to check that

$$\{(\rightarrow weight(between\_54\_56), 1), (weight(about\_55) \rightarrow \bot, 1)\} \models^{U,m}_{PGL^+\forall} (\bot, 1),$$

however, we have seen that

$$(\rightarrow weight(between\_54\_56), 1) \models^{U,m}_{PGL^+\forall} (weight(about\_55), \alpha) \text{ iff } \alpha = 0.$$

Hence, in PGL$^+\forall$

$$K \cup \{(\varphi \rightarrow \bot, 1)\} \models^{U,m}_{PGL^+\forall} (\bot, \alpha) \not\Rightarrow K \models^{U,m}_{PGL^+\forall} (\varphi, \alpha),$$

where $(\varphi \rightarrow \bot, 1)$ stands for the negation of a PGL$^+\forall$ query $(\varphi, 1)$. Hence, we have turned our attention to a sound deductive proof method based on the PGL$^+$ semantical unification pattern (see Section 6.2.2).

For the sake of simplicity, when clear from the connectives of PGL$^+\forall$ formulas, the quantifiers are dropped, i.e. PGL$^+\forall$ formulas of the form $((\forall \bar{x})(p_1 \wedge \cdots \wedge p_k \rightarrow q), \alpha)$ and $((\exists \bar{y})(p_1 \wedge \cdots \wedge p_n), \alpha)$, where $\bar{x}$ and $\bar{y}$ denote the set of variables involved in the atomic formulas, are simply written as $(p_1 \wedge \cdots \wedge p_k \rightarrow q, \alpha)$ and $(p_1 \wedge \cdots \wedge p_n, \alpha)$, respectively. Furthermore, we assume that $((\forall \bar{x})\varphi(\bar{x}), \alpha)$, $((\exists \bar{x})\varphi(\bar{x}), \alpha)$, and (φ, α) denote a PGL$^+\forall$ clause, a query, and a formula (clause or query), respectively.

Given a context $\mathcal{M}_{U,m}$, the calculus for PGL$^+\forall$ has the following triviality axiom and inference rules:

Axiom: $(\varphi, 0)$.

Resolution on PGL$^+\forall$ clauses:

$$\frac{(p \wedge q \rightarrow r, \alpha), (s \wedge t \rightarrow p, \beta)}{(s \wedge t \wedge q \rightarrow r, \min(\alpha, \beta))} \text{ [RR]}.$$

***Remark* 7.1** The following weighted *modus ponens* rule can be seen as a particular case of the RR rule:

$$\frac{\begin{array}{l}(p_1 \wedge \cdots \wedge p_n \rightarrow q, \alpha) \\ (\rightarrow p_1, \beta_1), \ldots, (\rightarrow p_n, \beta_n)\end{array}}{(\rightarrow q, \min(\alpha, \beta_1, \ldots, \beta_n))} \text{ [MP]}.$$

Query conjunction:

$$\frac{(\rightarrow p_1, \alpha_1), \ldots, (\rightarrow p_k, \alpha_k)}{(p_1 \wedge \cdots \wedge p_k, \min(\alpha_1, \ldots, \alpha_k))} \text{ [QC]}.$$

Semantical unification on PGL$^+\forall$ unit clauses:

$$\frac{(\rightarrow p(\ldots, A), \alpha)}{(\rightarrow p(\ldots, B), \min(\alpha, N^*(m(B) \mid m(A))))} \text{ [SU]}$$

if p is a predicate symbol of type $(\sigma_1, \ldots, \sigma_k, f\sigma_{k+1}, \ldots, f\sigma_n)$, and A and B are object constants of sort $f\sigma_n$; and where

$$N^*(m(B) \mid m(A)) = \inf_{u \in U_{\sigma_n}} \mu_{m(A)}(u) \Rightarrow \mu_{m(B)}(u),$$

$\Rightarrow$ being the reciprocal of Gödel's many-valued implication.

***Remark* 7.2** If $\mu_{m(A)} \leq \mu_{m(B)}$, we get the following rule instance:

$$\frac{(\rightarrow p(\ldots, A), \alpha)}{(\rightarrow p(\ldots, B), \alpha)}.$$

***Remark* 7.3** The rule admits the following straightforward generalization:

$$\frac{(\rightarrow p(\ldots, A_1, \ldots, A_p), \alpha)}{(\rightarrow p(\ldots, B_1, \ldots, B_p), \min(\alpha, \beta_1, \ldots, \beta_p))},$$

where $\beta_i = N^*(m(B_i) \mid m(A_i))$, for $i = 1, \ldots, p$.

Particularization on $PGL^+\forall$ clauses:

$$\frac{((\forall \bar{x}, y)\varphi(\bar{x}, y), \alpha)}{((\forall \bar{x})\varphi(\bar{x}, A), \alpha)} \text{ [PR]}$$

if A is an object constant of sort of variable y.

***Remark* 7.4** The rule admits the following straightforward generalization:

$$\frac{((\forall \bar{x}, y_1, \ldots, y_n)\varphi(\bar{x}, y_1, \ldots, y_n), \alpha)}{((\forall \bar{x})\varphi(\bar{x}, A_1, \ldots, A_n), \alpha)}.$$

Query instance:

$$\frac{((\exists \bar{x})\varphi(\bar{x}, A), \alpha)}{((\exists \bar{x}, y)\varphi(\bar{x}, y), \alpha)} \text{ [QI]}$$

if y is a variable of sort of object constant A.

***Remark* 7.5** The rule admits the following straightforward generalization:

$$\frac{((\exists \bar{x})\varphi(\bar{x}, A_1, \ldots, A_n), \alpha)}{((\exists \bar{x}, y_1, \ldots, y_n)\varphi(\bar{x}, y_1, \ldots, y_n), \alpha)}.$$

Renaming variables:

$$\frac{(\varphi(x_1, \ldots, x_k, x_{k+1}, \ldots, x_p), \alpha)}{(\varphi(y_1, \ldots, y_k, x_{k+1}, \ldots, x_p), \alpha)} \text{ [RV]}$$

if $y_1, \ldots, y_k$ are different variables and $y_i \notin \{x_{k+1}, \ldots x_p\}$, for $i = 1, \ldots, k$.

Merging:

$$\frac{(\varphi,\alpha),(\varphi,\beta)}{(\varphi,\max(\alpha,\beta))}\ [\mathrm{MR}].$$

Weakening:

$$\frac{(\varphi,\alpha)}{(\varphi,\beta)}\ [\mathrm{WR}]$$

if $\beta \leq \alpha$.

Obviously, the axiom is a valid PGL$^+\forall$ formula and inference rules are proved to be sound with respect to the possibilistic entailment of PGL$^+\forall$ formulas. We assume a particular PGL$^+\forall$ context $\mathcal{M}_{U,m}$ to be given, and thus, the notion of soundness is relative to the context.

Theorem 7.1 *soundness of the PGL$^+\forall$ inference rules For each PGL$^+\forall$ context $\mathcal{M}_{U,m}$, the RR, QC, SU, PR, QI, RV, MR and WR inference rules are sound with respect to the possibilistic entailment of PGL$^+\forall$ formulas.*

Proof:

RR: Given a PGL$^+\forall$ context $\mathcal{M}_{U,m}$, we must prove, for each possibilistic model $\pi : \mathcal{M}_{U,m} \to [0,1]$, that $\pi \models^{U,m}_{PGL^+\forall} (p \wedge q \to r, \alpha)$ and $\pi \models^{U,m}_{PGL^+\forall} (s \wedge t \to p, \beta)$ imply $\pi \models^{U,m}_{PGL^+\forall} (s \wedge t \wedge q \to r, \min(\alpha,\beta))$.

Assume that $\pi \models^{U,m}_{PGL^+\forall} (p \wedge q \to r, \alpha)$ and $\pi \models^{U,m}_{PGL^+\forall} (s \wedge t \to p, \beta)$. This means that $N^*_v([p \wedge q \to r] \mid \pi) \geq \alpha$ and $N^*_v([s \wedge t \to p] \mid \pi) \geq \beta$, for each evaluation v of variables. The two conditions amount to, for each interpretation $\mathbf{M} \in \mathcal{M}_{U,m}$, $\pi(\mathbf{M}) \Rightarrow \|p \wedge q \to r\|_{\mathbf{M},v} \geq \alpha$ and $\pi(\mathbf{M}) \Rightarrow \|s \wedge t \to p\|_{\mathbf{M},v} \geq \beta$. Thus, $\pi(\mathbf{M}) \Rightarrow \min(\|p \wedge q \to r\|_{\mathbf{M},v}, \|s \wedge t \to p\|_{\mathbf{M},v}) \geq \min(\alpha,\beta)$. But, by residuation, $\min(\|p \wedge q \to r\|_{\mathbf{M},v}, \|s \wedge t \to p\|_{\mathbf{M},v}) \leq \|s \wedge t \wedge q \to r\|_{\mathbf{M},v}$. Then, for each interpretation $\mathbf{M}$, $\pi(\mathbf{M}) \Rightarrow \|s \wedge t \wedge q \to r\|_{\mathbf{M},v} \geq \min(\alpha,\beta)$. Hence, $N^*_v([s \wedge t \wedge q \to r] \mid \pi) \geq \min(\alpha,\beta)$ for each evaluation v of variables, and thus, $\pi \models^{U,m}_{PGL^+\forall} (s \wedge t \wedge q \to r, \min(\alpha,\beta))$ as well.

QC: Given a PGL$^+\forall$ context $\mathcal{M}_{U,m}$, for each possibilistic model $\pi : \mathcal{M}_{U,m} \to [0,1]$, $\pi \models^{U,m}_{PGL^+\forall} (\to p_i, \alpha_i)$ iff, for each $\mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m}$ and evaluation of variables v, $\pi(\mathbf{M}) \Rightarrow \| \to p_i\|_{\mathbf{M},v} \geq \alpha_i$ for $i = 1,\ldots,k$. But, $\| \to p_i\|_{\mathbf{M},v} = \|p_i\|_{\mathbf{M},v}$, then $\pi(\mathbf{M}) \Rightarrow \min(\|p_1\|_{\mathbf{M},v},\ldots,\|p_k\|_{\mathbf{M},v}) \geq \min(\alpha_1,\ldots,\alpha_k)$ for each evaluation of variables v. Therefore, $\pi(\mathbf{M}) \Rightarrow \min(\|p_1\|_{\mathbf{M},v},\ldots,\|p_k\|_{\mathbf{M},v}) \geq \min(\alpha_1,\ldots,\alpha_k)$ for some evaluation of variables v. Thus, $\pi \models^{U,m}_{PGL^+\forall} (p_1 \wedge \cdots \wedge p_k, \min(\alpha_1,\ldots,\alpha_k))$ as well.

SU: Given a PGL$^+\forall$ context $\mathcal{M}_{U,m}$, we must prove, for each possibilistic model $\pi : \mathcal{M}_{U,m} \to [0,1]$, that if $\pi \models^{U,m}_{PGL^+\forall} (\to p(\ldots,A),\alpha)$

then $\pi \models^{U,m}_{PGL^+\forall} (\rightarrow p(\ldots,B), \min(\alpha, N^*(m(B) \mid m(A))))$. Assume that $\pi \models^{U,m}_{PGL^+\forall} (\rightarrow p(\ldots,A),\alpha)$. This means that $N^*_v([\rightarrow p(\ldots,A)] \mid \pi) \geq \alpha$ for each evaluation v of variables, and thus, $\pi(\mathbf{M}) \Rightarrow \| \rightarrow p(\ldots,A)\|_{\mathbf{M},v} \geq \alpha$ for each $\mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m}$.

Now, since p is a predicate of type $(\sigma_1,\ldots,\sigma_k,f\sigma_{k+1},\ldots,f\sigma_n)$, we have that for each $(u_1,\ldots,u_k) \in U_{\sigma_1} \times \cdots \times U_{\sigma_k}$ there exist unique values $(u_{k+1},\ldots,u_n) \in U_{\sigma_{k+1}} \times \cdots \times U_{\sigma_n}$ and $(u_1,\ldots,u_n) \in i(p)$. Hence, $\| \rightarrow p(\ldots,A)\|_{\mathbf{M},v} = \|p(\ldots,A)\|_{\mathbf{M},v} = \min(\gamma_{v,m}(\mathbf{o_M}), \delta_{v,m}(\mathbf{u_M}), \mu_{m(A)}(w_\mathbf{M}))$, where $\mathbf{o_M}$ and $\mathbf{u_M}$ denote vectors of values from the domain such that $(\mathbf{o_M},\mathbf{u_M},w_\mathbf{M}) \in i(p)$, $\gamma_{v,m}(\mathbf{o_M}) = 1$ and $\delta_{v,m}(\mathbf{u_M}) \in [0,1]$. Then, we have the following consecutive inequalities:

1. $\pi(\mathbf{M}) \Rightarrow \min(\delta_{v,m}(\mathbf{u_M}), \mu_{m(A)}(w_\mathbf{M})) \geq \alpha$
2. $\pi(\mathbf{M}) \Rightarrow \delta_{v,m}(\mathbf{u_M}) \geq \alpha$
   $\pi(\mathbf{M}) \Rightarrow \mu_{m(A)}(w_\mathbf{M}) \geq \alpha$
3. $\min(\pi(\mathbf{M}) \Rightarrow \mu_{m(A)}(w_\mathbf{M}), \mu_{m(A)}(w_\mathbf{M}) \Rightarrow \mu_{m(B)}(w_\mathbf{M})) \geq$
   $\min(\mu_{m(A)}(w_\mathbf{M}) \Rightarrow \mu_{m(B)}(w_\mathbf{M}), \alpha)$
4. $\min(\pi(\mathbf{M}) \Rightarrow \mu_{m(A)}(w_\mathbf{M}), \mu_{m(A)}(w_\mathbf{M}) \Rightarrow \mu_{m(B)}(w_\mathbf{M})) \leq$
   $\pi(\mathbf{M}) \Rightarrow \mu_{m(B)}(w_\mathbf{M})$
5. $\pi(\mathbf{M}) \Rightarrow \mu_{m(B)}(w_\mathbf{M}) \geq \min(\mu_{m(A)}(w_\mathbf{M}) \Rightarrow \mu_{m(B)}(w_\mathbf{M}), \alpha)$
6. $\pi(\mathbf{M}) \Rightarrow \min(\delta_{v,m}(\mathbf{u_M}), \mu_{m(B)}(w_\mathbf{M}))$
   $= \min(\pi(\mathbf{M}) \Rightarrow \delta_{v,m}(\mathbf{u_M}), \pi(w) \Rightarrow \mu_{m(B)}(w_\mathbf{M}))$
   $\geq \min(\mu_{m(A)}(w_\mathbf{M}) \Rightarrow \mu_{m(B)}(w_\mathbf{M}), \alpha)$
7. $\inf_{\mathbf{M}\in\mathcal{M}_{U,m}} \pi(\mathbf{M}) \Rightarrow \min(\delta_{v,m}(\mathbf{u_M}), \mu_{m(B)}(w_\mathbf{M})) \geq$
   $\inf_{\mathbf{M}\in\mathcal{M}_{U,m}} \min(\mu_{m(A)}(w_\mathbf{M}) \Rightarrow \mu_{m(B)}(w_\mathbf{M}), \alpha)$
8. $\inf_{\mathbf{M}\in\mathcal{M}_{U,m}} \pi(\mathbf{M}) \Rightarrow \| \rightarrow p(\ldots,B)\|_{\mathbf{M},v} \geq$
   $\inf_{\mathbf{M}\in\mathcal{M}_{U,m}} \min(\mu_{m(A)}(w_\mathbf{M}) \Rightarrow \mu_{m(B)}(w_\mathbf{M}), \alpha)$
9. $N^*_v([\rightarrow p(\ldots,B)] \mid \pi) \geq \min(\alpha, \inf_{u\in U_\sigma} \mu_{m(A)}(u) \Rightarrow \mu_{m(B)}(u))$
10. $N^*_v([\rightarrow p(\ldots,B)] \mid \pi) \geq \min(\alpha, N^*(m(B) \mid m(A)))$ for each evaluation v of variables.

Thus, $\pi \models^{U,m}_{PGL^+\forall} (\rightarrow p(\ldots,B), \min(\alpha, N^*(m(B) \mid m(A))))$ as well.

PR: Given a PGL$^+\forall$ context $\mathcal{M}_{U,m}$, we must prove, for each possibilistic model $\pi : \mathcal{M}_{U,m} \rightarrow [0,1]$, that if $\pi \models^{U,m}_{PGL^+\forall} ((\forall\bar{x},y)\varphi(\bar{x},y),\alpha)$ then $\pi \models^{U,m}_{PGL^+\forall} ((\forall\bar{x})\varphi(\bar{x},A),\alpha)$. Assume that $\pi \models^{U,m}_{PGL^+\forall} ((\forall\bar{x},y)\varphi(\bar{x},y),\alpha)$. This means that $N^*_v([\varphi(\bar{x},y)] \mid \pi) \geq \alpha$ for each evaluation v of variables $\bar{x}$ and y. By Definitions 7.3 and 7.4 some evaluations, namely v', evaluate variable y of sort σ (or $f\sigma$) to the element $m(A) \in U_\sigma$ (respectively, $m(A) \in F(U_\sigma)$), and thus, $\|\varphi(\bar{x},y)\|_{\mathbf{M},v'} = \|\varphi(\bar{x},A)\|_{\mathbf{M},v'}$ for each $\mathbf{M} = (U,i,m) \in \mathcal{M}_{U,m}$, which in turn implies $N^*_{v'}(\varphi(\bar{x},A) \mid \pi) \geq \alpha$ for each evaluation v' of variables $\bar{x}$. Hence, $\pi \models^{U,m}_{PGL^+\forall} ((\forall\bar{x})\varphi(\bar{x},A),\alpha)$.

QI: Given a PGL$^+\forall$ context $\mathcal{M}_{U,m}$, for each possibilistic model $\pi : \mathcal{M}_{U,m} \to [0,1]$, $\pi \models^{U,m}_{PGL^+\forall} ((\exists \bar{x})\varphi(\bar{x}, A), \alpha)$ iff $\pi(\mathbf{M}) \Rightarrow \|(\exists \bar{x})\varphi(\bar{x}, A)\|_{\mathbf{M}} \geq \alpha$ for each interpretation $\mathbf{M} = (U, i, m) \in \mathcal{M}_{U,m}$. But, $\|(\exists \bar{x})\varphi(\bar{x}, A)\|_{\mathbf{M}} = \sup_{v(\bar{x})} \|\varphi(\bar{x}, A)\|_{\mathbf{M},v}$. Now, for each evaluation v of variables $\bar{x}$ there exists an evaluation, namely v', of the form $v'(y) = m(A)$, and $v'(x_i) = v(x_i)$ for each variable $x_i \in \bar{x}$ such that $\|\varphi(\bar{x}, A)\|_{\mathbf{M},v} = \|\varphi(\bar{x}, y)\|_{\mathbf{M},v'}$. Thus, $\pi \models^{U,m}_{PGL^+\forall} ((\exists \bar{x}, y)\varphi(\bar{x}, y), \alpha)$ as well.

RV: Given a PGL$^+\forall$ context $\mathcal{M}_{U,m}$ and a possibilistic model $\pi : \mathcal{M}_{U,m} \to [0,1]$, assume that $\pi \models^{U,m}_{PGL^+\forall} (\varphi(x_1, \ldots, x_p), \alpha)$. Because $\varphi(x_1, \ldots, x_p)$ contains no free variables, this means that $\pi(\mathbf{M}) \Rightarrow \|\varphi(x_1, \ldots, x_p)\|_{\mathbf{M}} \geq \alpha$ for each interpretation $\mathbf{M} = (U, i, m) \in \mathcal{M}_{U,m}$, and the truth value $\|\varphi(x_1, \ldots, x_p)\|_{\mathbf{M}}$ is defined over all possible evaluation v of variables $x_1, \ldots, x_p$. Now, as variables $y_1, \ldots, y_k$ are different and $y_i \notin \{x_{k+1}, \ldots x_p\}$, for $i = 1, \ldots, k$, we have that for each evaluation v of variables $x_1, \ldots, x_p$ there exists an equivalent evaluation, namely v', of the form $v'(y_1) = v(x_1), \ldots, v'(y_k) = v(x_k), v'(x_{k+1}) = v(x_{k+1}), \ldots, v'(x_p) = v(x_p)$ such that $\|\varphi(x_1, \ldots, x_p)\|_{\mathbf{M},v} = \|\varphi(y_1, \ldots, y_k, x_{k+1}, \ldots, x_p)\|_{\mathbf{M},v'}$ for each interpretation $\mathbf{M}$. Hence, $\pi \models^{U,m}_{PGL^+\forall} (\varphi(y_1, \ldots, y_k, x_{k+1}, \ldots, x_p), \alpha)$ as well.

MR: Given a PGL$^+\forall$ context $\mathcal{M}_{U,m}$ and a possibilistic model $\pi : \mathcal{M}_{U,m} \to [0,1]$, assume that $\pi \models^{U,m}_{PGL^+\forall} (\varphi, \alpha)$ and $\pi \models^{U,m}_{PGL^+\forall} (\varphi, \beta)$. Because φ contains no free variable, this means that $N^*([\varphi] \mid \pi) \geq \alpha$ and $N^*([\varphi] \mid \pi) \geq \beta$. Thus, $N^*([\varphi] \mid \pi) \geq \max(\alpha, \beta)$.

WR: Given a PGL$^+\forall$ context $\mathcal{M}_{U,m}$ and a possibilistic model $\pi : \mathcal{M}_{U,m} \to [0,1]$, assume that $\pi \models^{U,m}_{PGL^+\forall} (\varphi, \alpha)$. Because φ contains no free variable, this means that $N^*([\varphi] \mid \pi) \geq \alpha$. Then, $N^*([\varphi] \mid \pi) \geq \beta$ for each $\beta \leq \alpha$.

■

Definition 7.11 (proof in PGL$^+\forall$) *Let K be a set of PGL$^+\forall$ formulas and let (φ, α) be a PGL$^+\forall$ formula. Given a context $\mathcal{M}_{U,m}$, K proves (φ, α), written $K \vdash^{U,m}_{PGL^+\forall} (\varphi, \alpha)$, iff there exists a finite sequence of PGL$^+\forall$ formulas $C_1, \ldots, C_m$ such that $C_m = (\varphi, \alpha)$ and, for each $i \in \{1, \ldots, m\}$, it holds that either $C_i \in K$, C_i is an instance of the triviality axiom or C_i is obtained by applying the RR, QC, SU, PR, QI, RV, MR and WR inference rules to previous PGL$^+\forall$ formulas in the sequence.*

Since Theorem 7.1 proves the soundness of PGL$^+\forall$ inference rules, we straightforwardly get the soundness of the deduction $\vdash^{U,m}_{PGL^+\forall}$.

Corollary 7.1 *Let K be a set of PGL$^+\forall$ formulas and let (φ, α) be a PGL$^+\forall$ formula. For each PGL$^+\forall$ context $\mathcal{M}_{U,m}$, if $K \vdash^{U,m}_{PGL^+\forall} (\varphi, \alpha)$ then $K \models^{U,m}_{PGL^+\forall} (\varphi, \alpha)$.*

7.4 Automated deduction

In the last section, we defined a sound deductive proof method for PGL$^+\forall$ clauses and queries based on the triviality axiom and the PGL$^+\forall$ inference rules. Now we are interested in automating this deductive process. For this purpose, taking into account that PGL$^+\forall$ queries are existentially quantified and unification for fuzzy constants is available in PGL$^+\forall$ unit clauses (by the SU inference rule), we define a deductive proof method oriented to PGL$^+\forall$ queries that is based on the MP, QC and WR inference rules, and a directional unification algorithm of fuzzy constants. However, because the soundness of the PGL$^+\forall$ proof method is relative to a context determined by a domain U and an interpretation m of object constants, before defining the unification algorithm and the proof procedure for PGL$^+\forall$ queries, we have to fix a particular PGL$^+\forall$ context, and thus, we are led first to formalize the notion of PGL$^+\forall$ program.

As in classical logic programming systems, a PGL$^+\forall$ program clause is either a fact or a rule that is defined as follows:

fact: it is a PGL$^+\forall$ unit clause such that the head of the G$^+\forall$ formula is not a truth constant; and

rule: it is a PGL$^+\forall$ clause such that neither the head nor the body of the G$^+\forall$ formula are a truth constant.

From now on, for the sake of a simpler and more standard notation, we write a fact $((\forall\bar{x})(\rightarrow q), \alpha)$, a rule $((\forall\bar{x})(p_1 \wedge \cdots \wedge p_k \rightarrow q), \alpha)$ and a query $((\exists\bar{x})(p_1 \wedge \cdots \wedge p_n), \alpha)$ as $(\rightarrow q, \alpha)$, $(p_1 \wedge \cdots \wedge p_k \rightarrow q, \alpha)$ and $(p_1 \wedge \cdots \wedge p_n, \alpha)$, respectively.

Definition 7.12 (PGL$^+\forall$ program) *A PGL$^+\forall$ program is a structure $\mathcal{P} = (P, U, m)$, where P is a finite set of facts and rules, U is a collection of non-empty domains, and m is an interpretation of object constants over U (or over $[0,1]^U$ in the case of imprecise and fuzzy constants).*

Definition 7.13 (proof in PGL$^+\forall$ programs) *Let $\mathcal{P} = (P, U, m)$ be a PGL$^+\forall$ program and let (φ, α) be a PGL$^+\forall$ query. $\mathcal{P}$ proves (φ, α), written $\mathcal{P} \vdash (\varphi, \alpha)$, iff P proves (φ, α) in the context $\mathcal{M}_{U,m}$ using the triviality axiom and the MP, QC, SU, PR, QI, RV, MR, and WR inference rules.*

Corollary 7.2 *Let $\mathcal{P} = (P, U, m)$ be a PGL$^+\forall$ program and let (φ, α) be a PGL$^+\forall$ query. If $\mathcal{P} \vdash (\varphi, \alpha)$ then $P \models^{U,m}_{PGL^+\forall} (\varphi, \alpha)$.*

Example 7.3 Let $price(\cdot,\cdot)$ be a binary predicate of type (`product`, f`product_price`), let $buy(\cdot)$ be a unary predicate of type (`product`), and let $\mathcal{P} = (P, U, m)$ be a PGL$^+\forall$ program with

1. $P = \{\ (\rightarrow price(book, about\_35), 0.9),$
   $(price(x, between\_30\_40) \rightarrow buy(x), 0.7)\ \};$

2. $U = \{\ U_{\texttt{product}} = \{book\},$
   $U_{\texttt{product\_price}} = [0, 100](\text{euros})\ \};$
3. $m(book) = book,$
   $m(about\_35) = [30; 35; 35; 40]$, and
   $m(between\_30\_40) = [25; 30; 40; 45]$.

Let us now compute a proof for the PGL$^+\forall$ query $(buy(book), 0.5)$ from P in the context determined by U and m. Applying the SU inference rule to the fact

$$(\rightarrow price(book, about\_35), 0.9),$$

we infer

$$(\rightarrow price(book, between\_30\_40), \min(0.9, \beta)),$$

where

$$\beta = N^*(m(between\_30\_40) \mid m(about\_35)) = 1.$$

Applying the PR rule to the program rule

$$(price(x, between\_30\_40) \rightarrow buy(x), 0.7),$$

we infer

$$(price(book, between\_30\_40) \rightarrow buy(book), 0.7).$$

Now, applying the MP inference rule to the derived fact and rule, we get

$$(\rightarrow buy(book), 0.7)$$

, but by the WR inference rule, we infer $(\rightarrow buy(book), 0.5)$. Finally, by the QC inference rule, we obtain the PGL$^+\forall$ query $(buy(book), 0.5)$. Hence,

$$\mathcal{P} \vdash (buy(book), 0.5),$$

and thus, $P \models^{U,m}_{PGL^+\forall} (buy(book), 0.5)$. □

In what follows, we are concerned with automating the proof method of PGL$^+\forall$ queries from PGL$^+\forall$ programs. To this aim, we define a backward first-order proof procedure based on the MP, QC and WR inference rules, and a directional fuzzy unification algorithm. That is, the proof procedure attempts to construct a proof tree for an input query beginning at the leaves (the atomic formulas of the query) and working up towards the root (the truth query $\top$). We can think of this process as one of "reducing" a PGL$^+\forall$ query (φ, α) to the truth query $\top$ with a necessity of at least α. At each reduction step, an atomic formula of the query matching the head of a PGL$^+\forall$ program clause is replaced by the body of it. If the PGL$^+\forall$ program clause is chosen correctly at each reduction step, a derivation through the PGL$^+\forall$ inference rules is traced out in reverse. We then need an algorithm that lets us know when two atomic formulas match. Moreover, we need an algorithm that automatically computes the set of substitutions or transformations that have to be applied in each reduction step. This algorithm is the *directional fuzzy unification algorithm* which is based on the SU, PR, QI, RV and MR inference rules.

7.4.1 Fuzzy unification

Because of the semantical unification inference rule, the unification between fuzzy constants is performed on PGL$^+\forall$ unit clauses and the unification degree is computed by means of a *directional* necessity measure on fuzzy sets, directional in the sense that, if A and B are two different fuzzy sets, in general,

$$N^*(A \mid B) \neq N^*(B \mid A).$$

So, we address the problem of unification involving fuzzy constants in systems in which a separation between general and specific patterns can be made (Rios-Filho and Sandri, 1995). The patterns classified in the first class are those expressing general domain information, such as, for instance, rules in knowledge-based systems, or ungrounded clauses in logic programming languages. The ones classified in the second class come from specific knowledge about a particular problem, such as facts in knowledge-based systems, or grounded clauses in logic programming languages.

In our framework, the proof method for PGL$^+\forall$ queries is performed in a bottom-up manner through the MP, QC and WR inference rules. Therefore, during the construction of a proof tree for a PGL$^+\forall$ query, the set of object constants from the head of PGL$^+\forall$ program clauses (initial facts and possibly derived facts) make up the *specific object constants*, whereas the set of object constants in PGL$^+\forall$ queries (the initial query and all possible subsequent queries generated by the proof method) make up the *general object constants*. The idea is to define a directional unification between specific and general object constants in the sense of computing the unification degree that, by the SU inference rule, is computed as

$$N^*(\text{general\_object\_constant} \mid \text{specific\_object\_constant}).$$

So, the unification degree for a PGL$^+\forall$ query is higher as soon as the information about the real world, expressed through a set of PGL$^+\forall$ program clauses, is more specific than the information requested by the PGL$^+\forall$ query. The following examples describe some cases that can occur during the unification process between general and specific object constants.

The most simple case appears when we have to unify a specific object constant with a general object constant.

Example 7.4 Let us consider the PGL$^+\forall$ program of Example 7.3 in which

$$\begin{aligned} P = \ & \{\ (\rightarrow price(book, about\_35), 0.9), \\ & (price(x, between\_30\_40) \rightarrow buy(x), 0.7)\ \}, \end{aligned}$$

where *between_30_40* and *about_35* are two fuzzy constants defined over the reference set $[0, 100](euros)$, $m(between\_30\_40) = [25; 30; 40; 45]$ and $m(about\_35) = [30; 35; 35; 40]$. To compute a proof for the PGL$^+\forall$ query

$$(buy(book), 0.5)$$

we have to derive a fact of the form $(\rightarrow buy(book), \alpha)$, with $\alpha \geq 0.5$. Thus, we have to compute a proof for the query

$$(price(book, between\_30\_40), 0.5).$$

Now, because the fuzzy constant *between_30_40* appears in the query and the fuzzy constant *about_35* in the program fact $(\rightarrow price(book, about\_35), 0.9)$, they are considered general and specific object constants, respectively. Then, applying the SU inference rule to the fact

$$(\rightarrow price(book, about\_35), 0.9),$$

we infer

$$(\rightarrow price(book, between\_30\_40), \min(0.9, \beta))$$

with

$$\beta = N^*(m(between\_30\_40) \mid m(about\_35)) = 1,$$

which, in turn, corresponds to computing the unification degree of the general object constant *between_30_40* and the specific object constant *about_35*. □

The second case that we have to analyze is the unification of a variable with multiple specific object constants.

Example 7.5 Let $age(\cdot,\cdot)$ and $friend(\cdot,\cdot)$ be two binary predicates of type (`person_name`, `fyears_old`) and (`person_name`, `person_name`), respectively; and let $\mathcal{P} = (P, U, m)$ be a PGL$^+\forall$ program with

1. $P = \{\ (\rightarrow age(Mary, young), 0.8),$
   $(\rightarrow age(John, about\_18), 0.9),$
   $(age(x,z) \wedge age(y,z) \rightarrow friend(x,y), 0.6)\ \};$
2. $U\ =\ \{\ U_{\texttt{person\_name}} = \{Mary, John\},$
   $U_{\texttt{years\_old}} = [0, 120](\text{years})\ \};$
3. $m(Mary) = Mary,$
   $m(John) = John,$
   $m(young) = [10; 15; 35; 40]$, and
   $m(about\_18) = [16; 18; 18; 20].$

To compute a proof for the PGL$^+\forall$ query

$$(friend(Mary, John), 0.5)$$

we have to derive a fact of the form $(\rightarrow friend(Mary, John), \alpha)$, with $\alpha \geq 0.5$. Thus, we have to compute a proof for the queries

$$(age(Mary, z), 0.5) \quad \text{and} \quad (age(John, z), 0.5).$$

Now, because the fuzzy constants *young* and *about_18* appear in the program facts

$$(\rightarrow age(Mary, young), 0.8) \quad \text{and} \quad (\rightarrow age(John, about\_18), 0.9),$$

they are considered specific object constants in the base system. Then, in order to compute a proof for the queries $(age(Mary, z), 0.5)$ and $(age(John, z), 0.5)$, variable z has to be instantiated either to the fuzzy constant *young* or *about_18*. If variable z is instantiated to the fuzzy constant *young*, by applying the SU inference rule to the fact $(\rightarrow age(John, about\_18), 0.9)$, we infer

$$(\rightarrow age(John, young), \min(0.9, \beta_1)),$$

where

$$\beta_1 = N^*(m(young) \mid m(about\_18)) = 1.$$

On the other hand, if variable z is instantiated to the fuzzy constant *about_18*, by applying the SU inference rule to the fact $(\rightarrow age(Mary, young), 0.8)$, we infer

$$(\rightarrow age(Mary, about\_18), \min(0.9, \beta_2)),$$

where

$$\beta_2 = N^*(m(about\_18) \mid m(young)) = 0.$$

Therefore, if variable z is instantiated to the fuzzy constant *young*, the query $friend(Mary, John)$ can be proved to be true from $\mathcal{P}$ with a necessity of at least 0.6, which, in turn, corresponds to computing the unification degree of a variable and multiple specific object constants. □

Finally, the following example analyzes the unification of a variable and multiple general and specific object constants.

Example 7.6 Let $price(\cdot,\cdot)$, $value(\cdot,\cdot)$ and $buy\_price\_value(\cdot,\cdot,\cdot)$ be three predicates of type $(\texttt{product}, f\texttt{product\_price})$, $(\texttt{product}, f\texttt{product\_value})$ and $(\texttt{product}, f\texttt{product\_price}, f\texttt{product\_value})$, respectively; and let $\mathcal{P} = (P, U, m)$ be a PGL$^+\forall$ program with

1. $P = \{\ (\rightarrow price(book, 34), 1),$
   $(\rightarrow value(book, about\_35), 0.8),$
   $(price(x, y) \wedge value(x, y) \rightarrow buy\_price\_value(x, y, y), 0.7)\ \};$

2. $U = \{\ U_{\texttt{product}} = \{book\},$
   $U_{\texttt{product\_price}} = [0, 100](\text{euros}),$
   $U_{\texttt{product\_value}} = [0, 100](\text{euros})\ \};$

3. $m(book) = book,$
   $m(34) = 34,$
   $m(about\_35) = [30; 35; 35; 40],$
   $m(around\_35) = [32; 34; 36; 38]$, and
   $m(between\_30\_40) = [25; 30; 40; 45].$

To compute a proof for the PGL$^+\forall$ query

$$(buy\_price\_value(book, around\_35, between\_30\_40), 0.5)$$

we have to derive a fact of the form

$$(\rightarrow buy\_price\_value(book, around\_35, between\_30\_40), \alpha),$$

with $\alpha \geq 0.5$. Now, since the fuzzy constants $around\_35$ and $between\_30\_40$ appear in the query they are considered general object constants in the base system. Then, variable y of the program rule

$$(price(x, y) \wedge value(x, y) \rightarrow buy\_price\_value(x, y, y), 0.7)$$

has to be instantiated either to the fuzzy constant $around\_35$ or $between\_30\_40$. If variable y is instantiated to the fuzzy constant $around\_35$, we have to compute a proof for the queries

$$(price(book, around\_35), 0.5) \quad \text{and} \quad (value(book, around\_35), 0.5),$$

which, by Example 7.4, corresponds with computing the unification degree between specific and general object constants, and thus, from the facts

$$(\rightarrow price(book, 34), 1) \quad \text{and} \quad (\rightarrow value(book, about\_35), 0.8),$$

by applying the SU inference rule, we infer

$$(\rightarrow price(book, around\_35), \min(1, \beta_1))$$

and

$$(\rightarrow value(book, around\_35), \min(0.8, \beta_2)),$$

respectively, where

$$\beta_1 = N^*(m(around\_35) \mid m(34)) = 1$$

and

$$\beta_2 = N^*(m(around\_35) \mid m(about\_35)) = 0.67.$$

Now, applying the MP inference rule to the instantiated rule

$$(price(book, around\_35) \wedge value(book, around\_35) \rightarrow buy\_price\_value(book, around\_35, around\_35), 0.7),$$

we infer

$$(\rightarrow buy\_price\_value(book, around\_35, around\_35), \min(1, 0.67, 0.7)).$$

Finally, applying the SU inference rule to the derived fact, we infer

$$(\rightarrow buy\_price\_value(book, around\_35, between\_30\_40), \min(0.67, \beta_3)),$$

where

$$\beta_3 = N^*(m(between\_30\_40) \mid m(around\_35)) = 1.$$

On the other hand, if variable y is instantiated to the fuzzy constant *between_30_40*, following the above process, we infer

$$(\rightarrow buy\_price\_value(book, between\_30\_40, between\_30\_40), \min(1, 0.8, 0.7)).$$

However, applying the SU inference rule to the derived fact, we infer

$$(\rightarrow buy\_price\_value(book, around\_35, between\_30\_40), \min(0.7, \beta_4)),$$

where

$$\beta_4 = N^*(m(around\_35) \mid m(between\_30\_40)) = 0.$$

Therefore, if variable y is instantiated to the fuzzy constant *around_35*, the query $buy\_price\_value(book, around\_35, between\_30\_40)$ can be proved to be true from $\mathcal{P}$ with a necessity of at least 0.67 which, in turn, corresponds with computing the unification degree of a variable and multiple general and specific object constants. □

In the rest of this section, we specify the data structures used in the unification algorithm, we formalize the definitions of fuzzy substitution, directional fuzzy unifier and most general directional fuzzy unifier, we develop a directional algorithm for determining a most general directional fuzzy unifier of two atomic formulas based on the distinction between general and specific object constants and, finally, we describe the computation of the unification degree of a fuzzy substitution by means of the SU and MR inference rules.

We represent the *semantic component* of a variable as two lists of object constants. The first one, called the *general list*, contains the general object constants to which the variable has been instantiated during the construction of the proof tree for a PGL$^+\forall$ query and it is concerned with the set of object constants from the query to which the variable has been unified. The second one, called the *specific list*, shows the set of object constants from heads of PGL$^+\forall$ program clauses to which the variable has been unified. In what follows, we refer to these two lists as the *semantic structure* of a variable. All constants of the semantic structure of a variable are defined over the same universal set. Before starting the proof procedure, the semantic structure of variables appearing in a program clause or query is empty.

Definition 7.14 (fuzzy substitution) *A fuzzy substitution σ consists of the following two components.*

Syntactic component: It is a mapping from variables to variables, and is written as $\theta = \{x_1/t_1, ..., x_n/t_n\}$, where variables $x_1, ..., x_n$ are different and $x_i \neq t_i$, for $i = 1, \ldots, n$.

Semantic component: It is a mapping from variables and dummy variables to semantic structures. A dummy variable is a variable associated with a pair of object constants. We represent a semantic component as $SC = \{x_1 \rightarrow (gl_1, sl_1), ..., x_m \rightarrow (gl_n, sl_n)\}$, where

- $x_1, ..., x_m$ *are different variables or dummy variables;*
- $(gl_1, sl_1), ..., (gl_n, sl_n)$ *are semantic structures;*
- *each variable* t_i*, for* $i = 1, \ldots, n$*, has a semantic structure, i.e.* $t_i \in \{x_1, ..., x_m\}$*; and*
- *each variable* x_i *such that* $x_i \notin \{t_1, ..., t_m\}$*, for* $i = 1, \ldots, n$*, has an empty semantic structure, i.e.* $x_i \rightarrow ([\,|\,], [\,|\,]) \in SC$.

From now on, we write a fuzzy substitution σ with syntactic component θ and semantic component SC as $\sigma(\theta, SC)$. Then, we say that two fuzzy substitutions $\sigma_1(\theta_1, SC_1)$ and $\sigma_2(\theta_2, SC_2)$ are equal iff $\theta_1 = \theta_2$ and $SC_1 = SC_2$.

Substitutions operate on expressions. By an expression, we mean a PGL$^+\forall$ term or a sequence of PGL$^+\forall$ atomic formulas.

Definition 7.15 (fuzzy instance of an expression) *Let* E *be an expression and let* $\sigma(\theta, SC)$ *be a fuzzy substitution. The* fuzzy instance *of* E*, written as* $E\sigma$*, stands for the result of applying the two following steps sequentially:*

1. Syntactic substitution: *Applying the syntactic component* θ *of the fuzzy substitution* σ *to the expression* E *and it is denoted by* $E\theta$*. The result is obtained from simultaneously replacing each occurrence in* E *of a variable from the domain of* θ *by the corresponding substitution term.*

2. Semantic substitution: *Applying the semantic component* SC *of the fuzzy substitution* σ *to the expression* $E\theta$*. The result is obtained from linking each variable of* $E\theta$ *with semantic structure in* SC *with the respective structure.*

Definition 7.16 (composition of fuzzy substitutions) *Fuzzy substitutions can be composed. Let* $\sigma_1(\theta_1, SC_1)$ *and* $\sigma_2(\theta_2, SC_2)$ *be two fuzzy substitutions with* $\theta_1 = \{x_1/t_1, ..., x_n/t_n\}$, $\theta_2 = \{y_1/s_1, ..., y_m/s_m\}$, $SC_1 = \{x_1 \rightarrow l_1, ..., x_k \rightarrow l_k\}$ *and* $SC_2 = \{y_1 \rightarrow m_1, ..., y_p \rightarrow m_p\}$*. The* composition *of* σ_1 *and* σ_2*, written* $\sigma_1\sigma_2$*, is a fuzzy substitution* $\sigma(\theta, SC)$ *in which*

1. θ *is defined by removing from the set*

$$\{x_1/t_1\theta_2, ..., x_n/t_n\theta_2, y_1/s_1, ..., y_m/s_m\}$$

those pairs $x_i/t_i\theta_2$ *for which* $t_i\theta_2$ *is* x_i *and those pairs* y_i/s_i *such that* $y_i \in \{x_1, ..., x_n\}$.

2. SC *is defined by removing from the set*

$$\{x_1 \rightarrow l_1, ..., x_k \rightarrow l_k, y_1 \rightarrow m_1, ..., y_p \rightarrow m_p\}$$

those pairs $x_i \rightarrow l_i$ *such that* $x_i \in \{y_1, ..., y_p\}$.

Definition 7.17 (more general fuzzy substitution) *Two fuzzy substitutions* $\sigma_1(\theta_1, SC_1)$ *and* $\sigma_2(\theta_2, SC_2)$ *are equal if* $\theta_1 = \theta_2$ *and* $SC_1 = SC_2$. *A semantic structure* (gl, sl) *is more general than a semantic structure* (gl', sl') *if* gl *and* sl *are sublists of* gl' *and* sl', *respectively. Then, we say that a fuzzy substitution* $\sigma_1(\theta_1, \{x_1 \rightarrow l_1, ..., x_k \rightarrow l_k\})$ *is more general than a fuzzy substitution* σ_2 *if there exists some fuzzy substitution* $\sigma_3(\{y_1/s_1, ..., y_m/s_m\}, \{y_1 \rightarrow m_1, ..., y_p \rightarrow m_p\})$ *such that*

1. $\sigma_2 = \sigma_1\sigma_3$; *and*

2. *each variable* x_i, *for* $i = 1, \ldots, k$, *either* $x_i \notin \{y_1, ..., y_p\}$, *or* $x_i \rightarrow m_{x_i} \in \{y_1 \rightarrow m_1, ..., y_p \rightarrow m_p\}$ *and* l_i *is more general than* m_{x_i}, *or* $x_i/s_{x_i} \in \{y_1/s_1, ..., y_m/s_m\}$ *and* $s_{x_i} \rightarrow m_{s_{x_i}} \in \{y_1 \rightarrow m_1, ..., y_p \rightarrow m_p\}$ *and* l_i *is more general than* $m_{s_{x_i}}$.

Definition 7.18 (directional fuzzy unifier) *Let* φ *and* ϕ *be two* $PGL^+\forall$ *atomic formulas with predicate symbol* p *of arity* n *and let* $\sigma(\theta, SC)$ *be a fuzzy substitution. If* $\varphi\sigma$ *is of the form* $p(s_1, ..., s_n)$ *and* $\phi\sigma$ *is of the form* $p(t_1, ..., t_n)$, *we say that* σ *is a directional fuzzy unifier from* ϕ *to* φ *if each pair* (s_i, t_i), *for* $i = 1, \ldots, n$, *satisfies one of the following conditions:*

1. s_i *and* t_i *are two object constants and either* $s_i = t_i$, *or there exists in SC a dummy variable with semantic structure* $([s_i|], [t_i|])$.

2. t_i *is a variable,* s_i *is an object constant which belongs to the general list of the semantic structure of* t_i, *and the* $i-$*term of* ϕ *is a variable liked with a semantic structure which is more general than the semantic structure of* t_i.

3. s_i *is a variable,* t_i *is an object constant which belongs to the specific list of the semantic structure of* s_i, *and the* $i-$*term of* φ *is a variable liked with a semantic structure which is more general than the semantic structure of* s_i.

4. s_i *and* t_i *are a same variable and the* $i-$*terms of both* φ *and* ϕ *are variables liked with semantic structures which are more general than the semantic of* s_i.

Definition 7.19 (most general directional fuzzy unifier) *A directional fuzzy unifier* σ *from an atomic formula* ϕ *to an atomic formula* φ *is a most general directional fuzzy unifier (or* mgdfu *for short), if it is more general than each other directional fuzzy unifier from* ϕ *to* φ. *In fact, mgdfu's are unique modulo renaming of variables.*

In order to develop the unification algorithm, we follow the presentation of Apt (1990) and Lassez et al. (1988), based upon Herbrand's original algorithm which deals with solutions of finite sets of term equations. This algorithm was first presented by Martelli and Montanari (1982).

Algorithm 7.1 directional fuzzy unification algorithm

Input: Two PGL$^+\forall$ atomic formulas with predicate symbol p of arity n of the form $p(s_1, ..., s_n)$ and $p(t_1, ..., t_n)$ such that they do not have any variable in common and each variable keeps the link to the semantic structure associated with the respective formula. The algorithm unifies general object constants with specific object constants and gives rise to a *directional unification* process. Therefore, we assume that $p(s_1, ..., s_n)$ is an atomic formula from a query, and thus, $p(t_1, ..., t_n)$ is the head of a PGL$^+\forall$ program clause, i.e. an object constant $s_i \in \{s_1, ..., s_n\}$ (respectively, $t_i \in \{t_1, ..., t_n\}$) denotes a general object constant (respectively, a specific object constant).

Output: An mgdfu $\sigma(\theta, SC)$ from $p(t_1, ..., t_n)$ to $p(s_1, ..., s_n)$.

Initialization:

- From the pair of PGL$^+\forall$ atomic formulas $p(s_1, ..., s_n)$ and $p(t_1, ..., t_n)$ we construct a set of pairs of terms of the form

$$ST = \{(s_1, t_1), ..., (s_n, t_n)\}.$$

- $SC := \{x \rightarrow (gl_x, sl_x) \mid x$ is a variable of one of the atomic formulas and (gl_x, sl_x) is the semantic structure linked with x at the atomic formula$\}$

Method: Choose a pair (s_i, t_i) from ST and perform the associated action until ST and SC do not change.

1. s_i and t_i are object constants:
   if ($s_i \neq t_i$) **then**
   if (the dummy variable $(s_i, t_i) \notin SC$) **then**
   $SC := SC \cup \{(s_i, t_i) \rightarrow ([s_i|], [t_i|])\}$, where (s_i, t_i) denotes a dummy variable and $([s_i|], [t_i|])$ its semantic structure
   end if
   end if
   Delete the pair (s_i, t_i) from ST
2. s_i is an object constant and t_i is a variable:
   if ($t_i \rightarrow (gl_{t_i}, sl_{t_i}) \in SC$ and $s_i \notin gl_{t_i}$) **then**
   $SC := (SC \backslash \{t_i \rightarrow (gl_{t_i}, sl_{t_i})\}) \cup \{t_i \rightarrow (\texttt{insert}(gl_{t_i}, s_i), sl_{t_i})\}$
   end if
   Delete the pair (s_i, t_i) from ST
3. s_i is a variable and t_i is an object constant:
   if ($s_i \rightarrow (gl_{s_i}, sl_{s_i}) \in SC$ and $t_i \notin sl_{s_i}$) **then**
   $SC := (SC \backslash \{s_i \rightarrow (gl_{s_i}, sl_{s_i})\}) \cup \{s_i \rightarrow (gl_{s_i}, \texttt{insert}(sl_{s_i}, t_i))\}$

end if

Delete the pair (s_i, t_i) from ST

4. s_i and t_i are variables:

if ($s_i = t_i$) **then**

Delete the pair (s_i, t_i) from ST

else

if ($s_i \to ([\,|\,], [\,|\,]) \in SC$) **then**

if (s_i has another occurrence in ST) **then**

Perform the syntactic substitution $\{s_i/t_i\}$ on each other pair of terms of ST

end if

else

if ($t_i \to ([\,|\,], [\,|\,]) \in SC$) **then**

Replace (s_i, t_i) by (t_i, s_i) in ST

else

$SC := (SC \setminus \{s_i \to (gl_{s_i}, sl_{s_i}), t_i \to (gl_{t_i}, sl_{t_i})\}) \cup$
$\cup \{s_i \to ([\,|\,], [\,|\,]), t_i \to (gl_{t_i} \circ gl_{s_i}, sl_{t_i} \circ sl_{s_i})\}$,
where $\circ$ denotes the concatenation operation of lists.

Perform the syntactic substitution $(\{s_i/t_i\}, SC)$ on each other pair of terms of ST

end if

end if

end if

Final treatment:

If $\{(x_1, u_1), ..., (x_k, u_k)\}$ is the resulting set of pairs of terms, then $\theta := \{x_1/u_1, ..., x_k/u_k\}$ and $\sigma = (\theta, SC)$.

Proposition 7.1 *The directional fuzzy unification algorithm computes an mgdfu from an atomic formula of the form $p(t_1, ..., t_n)$ to an atomic formula of the form $p(s_1, ..., s_n)$.*

Proof:

- *The directional fuzzy unification algorithm terminates.*

  The algorithm terminates when, for each pair of terms, $(s_i, t_i) \in ST$, the execution of the associated action does not modify ST and SC:

  (i) We can easily check that, for each pair of terms $(s_i, t_i) \in ST$, there exists exactly one associated algorithmic action.

  (ii) We prove that for each pair of terms (s_i, t_i) obtained from $p(s_1, ..., s_n)$ and $p(t_1, ..., t_n)$, the algorithm modifies ST and SC a finite number of times.

Therefore, (i) and (ii) imply termination.

To prove (ii) we must analyze the behavior of the algorithm for each action. We show that each algorithmic action either can be performed a finite number of times, reduces the number of pairs of ST, or replaces a pair of terms by an equivalent one that satisfies a different action:

- Actions (1), (2) and (3) reduce the number of pairs of ST.
- Action (4) either reduces the number of pairs of ST, replaces a pair of terms by an equivalent one that satisfies the action (4), but not the replacing condition, or performs a substitution. However, for each variable s_i the substitution $\{s_i/t_i\}$ can be performed at most once, so the action can be performed only a finite number of times.

- *The directional fuzzy unification algorithm computes an mgdfu from* $p(t_1, ..., t_n)$ *to* $p(s_1, ..., s_n)$.

  The invariant of the iteration is "the composition of the fuzzy substitutions computed by the algorithm make up an mgdfu of the resolved pairs of terms from $p(t_1, ..., t_n)$ to $p(s_1, ..., s_n)$".

  To establish the correctness of the algorithm, we must prove that the algorithm satisfies the invariant of the iteration before the first iteration and after each repetition including the final one:

  (i) Before the first iteration, the set of resolved pairs of terms from $p(t_1, ..., t_n)$ to $p(s_1, ..., s_n)$ is $\emptyset$, and the composition of the fuzzy substitutions computed by the algorithm is the fuzzy substitution $(\emptyset, SC)$, where for each variable of the atomic formulas SC keeps the link to the semantic structure associated in the respective formula.

  (ii) We know that for each pair of terms (s_i, t_i), there exists one associated algorithmic action. Then,

  - Action (1) computes a substitution with an empty syntactic structure and a dummy variable as semantic structure, which corresponds with an mgdfu from t_i to s_i. Therefore, the composition of the new substitution is an mgdfu of the resolved set of pairs of terms from $p(t_1, ..., t_n)$ to $p(s_1, ..., s_n)$.
  - Actions (2) and (3) modify the semantic structure of a variable according to the definition of a directional fuzzy unifier and its construction ensures that the new structure is most general.
  - Action (4) is performed when two variables have to be unified. If the two variables are equal, the action generates an empty substitution and the invariant remains true. Otherwise, for each variable s_i and t_i, if action (4) has to be performed, it is because no syntactic substitution has been applied before. Therefore, the semantic structure of either s_i or t_i is empty, or both semantic structures contain a list of object constants. If the semantic

structure of t_i is empty, action (4) replaces the pair by an equivalent one and does not create a new substitution. In that case, the elements of the invariant have not changed. If s_i has an empty semantic structure, it must be the first time that the variable is considered and the composition of the new substitution is an mgdfu of the new set of resolved pairs of terms from $p(t_1, ..., t_n)$ to $p(s_1, ..., s_n)$. Finally, if actions (2) or (3) have been applied for s_i and t_i, the unification algorithm computes a substitution that replaces s_i by t_i and maintains the fuzzy instances of s_i. Then, after this step the invariant is already true.

(iii) Finally, we must prove that the composition of the fuzzy substitutions computed by the algorithm make up an mgdfu from $p(t_1, ..., t_n)$ to $p(s_1, ..., s_n)$. For this purpose, we have to prove that after the last iteration the set of resolved pairs of terms of $p(s_1, ..., s_n)$ and $p(t_1, ..., t_n)$ is $\{(s_1, t_1), ..., (s_n, t_n)\}$. We have seen that for each pair of terms (s_i, t_i) there exists exactly one associated action in the algorithm. Every action computes a fuzzy substitution except those that replace a pair of terms by an equivalent one. The algorithm ensures that for each pair at most one replacement is performed. Then, at the end of the iteration each pair has been resolved.

■

According to Proposition 7.1, each pair of PGL$^+\forall$ atomic formulas of the form $p(s_1, ..., s_n)$ and $p(t_1, ..., t_n)$ can be unified. We then need to measure, through the SU and MR inference rules, the unification degree of a computed mgdfu.

Example 7.7 Let us consider the PGL$^+\forall$ program of Example 7.5, in which

$$\begin{aligned} P \;=\; \{\; & (\rightarrow age(Mary, young), 0.8), \\ & (\rightarrow age(John, about\_18), 0.9), \\ & (age(x, z) \wedge age(y, z) \rightarrow friend(x, y), 0.6)\;\}, \end{aligned}$$

where *young* and *about_18* are two fuzzy constants defined over the reference set $[10; 15; 35; 40]$(years), $m(young) = [10; 15; 35; 40]$ and $m(about\_18) = [16; 18; 18; 20]$. As the proof method for PGL$^+\forall$ queries is performed in a bottom-up manner through the MP, QC and WR inference rules, to compute a proof for the PGL$^+\forall$ query

$$(friend(Mary, John), 0.5),$$

the atomic formula $friend(Mary, John)$ of the query is unified with the head of the program rule

$$(age(x, z) \wedge age(y, z) \rightarrow friend(x, y), 0.6)$$

and an mgdfu is

$$\sigma_1 = (\emptyset, \{x \rightarrow ([Mary|\,], [\,|\,]), y \rightarrow ([John|\,], [\,|\,])\}).$$

Then, the new PGL$^+\forall$ query is

$$(age(x,z) \wedge age(y,z), 0.5),$$

where the semantic structure of variables x and y is

$$x \rightarrow ([Mary|\,],[\,|\,]) \quad \text{and} \quad y \rightarrow ([John|\,],[\,|\,]),$$

respectively. Now, the atomic formula $age(x,z)$ is unified with the atomic formula of the fact

$$(\rightarrow age(Mary, young), 0.8)$$

and an mgdfu is

$$\sigma_2 = (\emptyset, \{x \rightarrow ([Mary|\,],[Mary|\,]), z \rightarrow ([\,|\,],[young|\,])\}).$$

Therefore, the new PGL$^+\forall$ query is

$$(age(y,z), 0.5),$$

where the semantic structure of variables y and z is

$$y \rightarrow ([John|\,],[\,|\,]) \quad \text{and} \quad z \rightarrow ([\,|\,],[young|\,]),$$

respectively. Then, the atomic formula $age(y,z)$ is unified with the atomic formula of the fact

$$(\rightarrow age(Mary, young), 0.8)$$

and an mgdfu is

$$\sigma_3 = (\emptyset, \{y \rightarrow ([John|\,],[Mary|\,]), z \rightarrow ([\,|\,],[young|young])\}).$$

However, as $N^*(m(John) \mid m(Mary)) = 0$, the proof procedure fails and a new program clause is considered. Finally, the atomic formula $age(y,z)$ is unified with the atomic formula of the fact

$$(\rightarrow age(John, about\_18), 0.9)$$

and an mgdfu is

$$\sigma_3' = (\emptyset, \{y \rightarrow ([John|\,],[John|\,]), z \rightarrow ([\,|\,],[young|about\_18])\}).$$

Now, composing σ_1, σ_2 and σ_3' we get

$$\sigma_4 = (\emptyset,\ \{x \rightarrow ([Mary|\,],[Mary|\,]), y \rightarrow ([John|\,],[John|\,]),\\ z \rightarrow ([\,|\,],[young|about\_18])\}\ ),$$

which, in turn, shows that the PGL$^+\forall$ query $(friend(Mary, John), 0.5)$ can be proved from two different PGL$^+\forall$ clauses:

1. $(age(Mary, young) \wedge age(John, young) \rightarrow friend(Mary, John), 0.6)$

2. $(age(Mary, about\_18) \wedge age(John, about\_18) \rightarrow friend(Mary, John), 0.6)$

The first clause can be resolved through the MP inference rule by applying the SU inference rule to the fact

$$(\rightarrow age(John, about\_18), 0.9)$$

and the unification degree is

$$N^*(m(young) \mid m(about\_18)) = 1,$$

then the PGL$^+\forall$ query can be proved to be true with a necessity degree of at least $\min(1, 0.8, 0.9, 0.6) = 0.6$. The second clause can be resolved through the MP inference rule by applying the SU inference rule to the fact

$$(\rightarrow age(Mary, young), 0.8)$$

and the unification degree is

$$N^*(m(about\_18) \mid m(young)) = 0,$$

then the PGL$^+\forall$ query can be proved to be true with a necessity degree of at least $\min(0, 0.8, 0.9, 0.6) = 0$. Finally, by the MR inference rule,

$$\mathcal{P} \vdash (friend(Mary, John), 0.6)$$

and, by the WR inference rule,

$$\mathcal{P} \vdash (friend(Mary, John), 0.5).$$

So, the unification degree of the fuzzy substitution σ_4 has to be computed as the minimum of the unification degree of variables x, y and z, where the unification degree of variables x and y has to be computed through the SU inference rule as

$$N^*(m(Mary) \mid m(Mary)) = 1 \quad \text{and} \quad N^*(m(John) \mid m(John)) = 1,$$

respectively; and the unification degree of variable z has to be computed through the SU and MR inference rules as the maximum of

$$N^*(m(young) \mid m(about\_18)) = 1$$

and

$$N^*(m(about\_18) \mid m(young)) = 0.$$

□

Definition 7.20 (unification degree) *Let $\mathcal{M}_{U,m}$ be a PGL$^+\forall$ context and let $\sigma(\theta, SC)$ be a fuzzy substitution with $SC = \{v_1 \rightarrow (gl_1, sl_1), ..., v_n \rightarrow (gl_n, sl_n)\}$, where $v_1, ..., v_n$ are variables or dummy variables and $(gl_1, sl_1), ..., (gl_n, sl_n)$ are*

their semantic structures. We define the unification degree of the fuzzy substitution σ in the context $\mathcal{M}_{U,m}$, denoted by $UD_{U,m}(\sigma)$, in the following way:

$$UD_{U,m}(\sigma) = \min(UDV_{U,m}(v_1, SC), \ldots, UDV_{U,m}(v_n, SC)),$$

where $UDV_{U,m}(v_i, SC)$ denotes the unification degree of variable v_i, for $i = 1, \ldots, n$, determined by the semantic component of the fuzzy substitution in the context $\mathcal{M}_{U,m}$. The unification degree of a variable v is computed, through the SU and MR inference rules, depending on its type and its semantic structure in the following way:

- *If v is a dummy variable and $v \to ([gc|], [sc|]) \in SC$, then*

$$UDV_{U,m}(v, SC) \;=\; N^*(m(gc) \mid m(sc));$$

- *If v is a variable and $v \to ([\,|\,], [sc_1|sc_2 ... sc_k]) \in SC$, then*

$$\begin{aligned} UDV_{U,m}(v, SC) \;=\; & \max\{\min(N^*(m(sc_j) \mid m(sc_1)), \ldots, \\ & N^*(m(sc_j) \mid m(sc_k))) \mid sc_j \in \{sc_1, \ldots, sc_k\}\}, \end{aligned}$$

  and the specific object constant sc_j that provides the maximum make up the assignment of value to the variable;

- *If v is a variable and $v \to ([gc_1|gc_2 ... gc_l], [sc_1|sc_2 ... sc_k]) \in SC$, then*

$$\begin{aligned} UDV_{U,m}(v, SC) \;=\; & \max\{\min(N^*(m(gc_1) \mid m(gc_j)), \ldots, \\ & N^*(m(gc_l) \mid m(gc_j)), N^*(m(gc_j) \mid m(sc_1)), \ldots, \\ & N^*(m(gc_j) \mid m(sc_k))) \mid gc_j \in \{gc_1, \ldots, gc_l\}\}, \end{aligned}$$

  and the general object constant gc_j that provides the maximum make up the assignment of value to the variable.

7.4.2 Proof procedure

As we have already pointed out, the proof of a PGL$^+\forall$ query from a PGL$^+\forall$ program is defined by derivation using the MP, QC, SU, PR, QI, RV, MR, and WR inference rules, where the SU, PR, QI, RV and MR inference rules are automatically performed by the fuzzy unification algorithm. Therefore, the proof procedure is based on the MP, QC and WR inference rules.

The proof procedure we define applies the generalized modus ponens and the query conjunction inference rules in a reverse way as it is done in classical backward inference systems, i.e. rather than applying the MP inference rule to $p_1 \wedge \cdots \wedge p_n \to q$ once we have deduced $p_1, \ldots, p_n$, it checks first whether q can be deduced, which in turn leads to checking whether $p_1, \ldots, p_n$ can be deduced. Therefore, at each step, an atomic formula of a PGL$^+\forall$ query that can be unified, through an mgdfu σ, with the head of a PGL$^+\forall$ program clause, is replaced by the body of it. The new PGL$^+\forall$ query is obtained from applying the mgdfu σ to the resulting query. The proof procedure finishes when the

truth query $\top$ has to be checked. Moreover, the proof procedure is based on chronological backtracking and the search strategy is depth-first. Finally, the assignment of value for variables is computed by means of a repeated use of the fuzzy unification algorithm.

Let $\mathcal{P} = (P, U, m)$ be a PGL$^+\forall$ program. In order to verify if a PGL$^+\forall$ query (φ, α) can be proved from $\mathcal{P}$, we apply the proof procedure described below with the function call

$$Deductive\_Proof\_Procedure(P, U, m, \varphi, \alpha, (\emptyset, \emptyset), 1),$$

where $(\emptyset, \emptyset)$ and 1 mean, respectively, that neither a fuzzy substitution nor a necessity degree have been computed before starting the proof procedure.

function *Deductive_Proof_Procedure*

input
P : Set of PGL$^+\forall$ program clauses
U : Collection of non-empty domains
m : Interpretation of object constants over U `/* context` $\mathcal{M}_{U,m}$ `*/`
query : Set of PGL$^+\forall$ atomic formulas
threshold : Necessity degree `/* proof threshold */`
σ : Fuzzy substitution
β: Necessity degree `/* computed degree of proof */`

output
entailment : boolean
$\sigma_{entailment}$: Fuzzy substitution
$\beta_{entailment}$: Necessity degree

auxiliary variables
P' : Set of PGL$^+\forall$ program clauses
query' : Set of PGL$^+\forall$ atomic formulas
q : PGL$^+\forall$ atomic formula
σ_1, σ_2, : Fuzzy substitution /* $\sigma_i = (\theta_i, SC_i)$ */
β': Necessity degree
C: PGL$^+\forall$ program clause

begin
if (*Empty_Set*(*query*)) **then**
entailment := true;
$\sigma_{entailment} := \sigma$;
$\beta_{entailment} := \min(\beta, UD_{U,m}(\sigma))$;
return(*entailment*, $\sigma_{entailment}$, $\beta_{entailment}$)
else
$P' :=$ *Rename_Variables*(P);
$q :=$ *First*(*query*);
for (each PGL$^+\forall$ program clause $C \in P'$)**do**
if (*Weight*(C) $\geq$ *threshold* and *Predicate*(q) = *Predicate*(*Head*(C)))
then

```
        σ1 := mgdfu(q,Head(C));
        query' := (Body(C) ∪ (query \ {q}))σ1;
        σ2 := σσ1;
        if ( min{UDV_{U,m}(v, SC2) | v ∉ query' and v ∈ SC2} ≥ threshold )
        then
          β' := min(β, Weight(C));
          (entailment,σ_entailment,β_entailment) :=
             Deductive_Proof_Procedure(P', U, m, query', threshold, σ2, β');
          if ( entailment ) then
            return( entailment, σ_entailment, β_entailment )
          end if
        end if
      end if
    end for
    entailment := false;
    σ_entailment := (∅, ∅);
    β_entailment := 0;
    return( entailment, σ_entailment, β_entailment )
  end if
end function  Deductive_Proof_Procedure
```

The function *Rename_Variables* applies the RV inference rule to ensure that the name of variables is different in each level of the proof tree and in each PGL$^+\forall$ program clause. Moreover, the semantic structure of all renamed variables is initialized to the empty set. The function *Head* returns the conclusion of a PGL$^+\forall$ program clause; the function *Body* produces the set of premises of a program rule or the empty set for facts; and the function *Weight* returns the certainty-weight attached to a PGL$^+\forall$ program clause. Finally, functions $UD_{U,m}$ and $UDV_{U,m}$ use Definition 7.20 for computing the unification degree, in the PGL$^+\forall$ context determined by U and m, of a fuzzy substitution and a variable, respectively.

The function *Deductive_Proof_Procedure* runs in a depth-first manner applying the generalized modus ponens rule in a reverse order, i.e. an atomic formula q is deduced, with a necessity of at least α, if all the atomic formulas of the body of a PGL$^+\forall$ program clause, whose head unifies with q, are deduced with a necessity of at least α and the certainty-weight of the PGL$^+\forall$ program clause is higher than or equal to α. Thus, the function call

$$Deductive\_Proof\_Procedure(P, U, m, \{q_1, \ldots, q_n\}, \alpha, \sigma, \beta),$$

selects the leftmost atom q_1 of the query and a PGL$^+\forall$ program clause C of the form

$$(Body(C) \rightarrow Head(C),\ Weight(C))$$

such that $Weight(C) \geq \alpha$ and $Head(C)$ unifies with q_1. Let σ_1 be an mgdfu from $Head(C)$ to q_1, if the unification degree of each completely instantiated variable

through the fuzzy substitution $\sigma\sigma_1$ is higher than or equal to the threshold α, a new function call of the form

$$\mathit{Deductive\_Proof\_Procedure}(P, U, m, (\mathit{Body}(C) \cup \{q_2, \ldots, q_n\})\sigma_1, \alpha, \sigma\sigma_1, \beta'),$$

with $\beta' = \min(\beta, \mathit{Weight}(C))$ is produced. A variable or a dummy variable v is said to be completely instantiated through the fuzzy substitution

$$\sigma_2(\theta_2, SC_2) = \sigma\sigma_1,$$

if SC_2 contains a non-empty semantic structure for v and variable v has no occurrence in the new query

$$(\mathit{Body}(C) \cup \{q_2, \ldots, q_n\})\sigma_1.$$

If no PGL$^+\forall$ program clause satisfies these constraints for the leftmost atom of the query, the function *Deductive_Proof_Procedure* backtracks. The function finishes when the truth query $\top$ has to be checked or when the input PGL$^+\forall$ query cannot be proved with a necessity degree equal or higher than the threshold α. In the first case, the computed fuzzy substitution σ corresponds with the assignment of value to variables and

$$\beta_{\mathit{entailment}} = \min(\beta, UD_{U,m}(\sigma)) \geq \alpha$$

gives the necessity degree with which the initial query has been proved from the PGL$^+\forall$ program, i.e. given a PGL$^+\forall$ program $\mathcal{P} = (P, U, m)$ and a PGL$^+\forall$ query (φ, α), if the function call

$$\mathit{Deductive\_Proof\_Procedure}(P, U, m, \varphi, \alpha, (\emptyset, \emptyset), 1),$$

returns $(\mathit{true}, \sigma, \beta_{\mathit{entailment}})$, then $\mathcal{P} \vdash (\varphi, \beta_{\mathit{entailment}})$ with $\beta_{\mathit{entailment}} \geq \alpha$ and, by the WR inference rule, $\mathcal{P} \vdash (\varphi, \alpha)$.

7.4.3 Examples

In this section, we apply the unification algorithm and the deductive proof procedure to examples of Section 7.4.1. For this purpose, we describe the construction of a proof tree corresponding to a branch of the search tree.

The first example of Section 7.4.1 shows the unification of two fuzzy constants. Let

$$(\mathit{buy}(\mathit{book}), 0.5)$$

be a PGL$^+\forall$ query and let $\mathcal{P} = (P, U, m)$ be a PGL$^+\forall$ program with

1. $P = \{\ (\rightarrow \mathit{price}(\mathit{book}, \mathit{about\_35}), 0.9),$
   $(\mathit{price}(x, \mathit{between\_30\_40}) \rightarrow \mathit{buy}(x), 0.7)\ \};$
2. $U = \{\ U_{\texttt{product}} = \{\mathit{book}\},$
   $U_{\texttt{product\_price}} = [0, 100](\text{euros})\ \};$

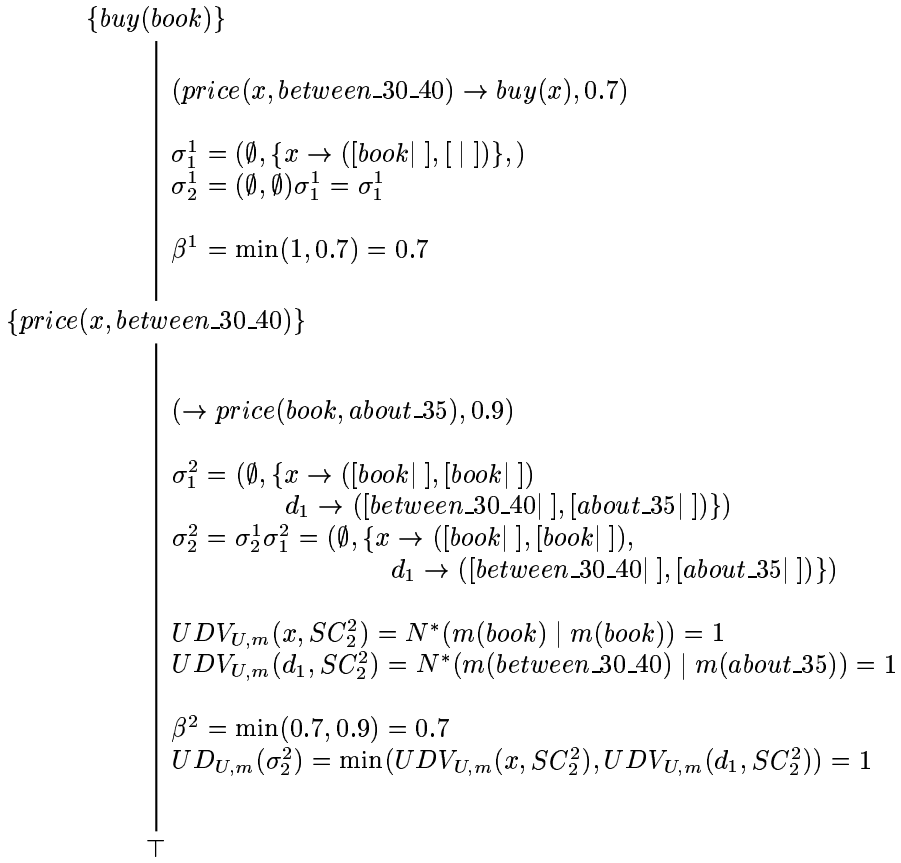

Figure 7.1: Unification of two fuzzy constants.

3. $m(book) = book$,
   $m(about\_35) = [30; 35; 35; 40]$, and
   $m(between\_30\_40) = [25; 30; 40; 45]$.

The proof tree generated by the function call

$$Deductive\_Proof\_Procedure(P, U, m, \{buy(book)\}, 0.5, (\emptyset, \emptyset), 1),$$

is described in Figure 7.1, where d_1 denotes the dummy variable associated with the fuzzy constants *between_25_30* and *about_35*. When the truth query $\top$ is obtained, the function returns

$$(\text{true}, \sigma_2^2, \min(\beta^2, UD_{U,m}(\sigma_2^2))),$$

and thus,

$$\mathcal{P} \vdash (buy(book), 0.7).$$

Now, applying the WR inference rule, we get

$$\mathcal{P} \vdash (buy(book), 0.5).$$

The second example of Section 7.4.1 analyzes the unification of a variable and multiple specific fuzzy constants. Let

$$(friend(Mary, John), 0.5)$$

be a PGL$^+\forall$ query and let $\mathcal{P} = (P, U, m)$ be a PGL$^+\forall$ program with

1. $P = \{\ (\rightarrow age(Mary, young), 0.8),$
   $(\rightarrow age(John, about\_18), 0.9),$
   $(age(x, z) \wedge age(y, z) \rightarrow friend(x, y), 0.6)\ \}$;
2. $U = \{\ U_{\texttt{person\_name}} = \{Mary, John\},$
   $U_{\texttt{years\_old}} = [0, 120](\text{years})\ \}$;
3. $m(Mary) = Mary$,
   $m(John) = John$,
   $m(young) = [10; 15; 35; 40]$, and
   $m(about\_18) = [16; 18; 18; 20]$.

The proof tree associated with the function call

$$Deductive\_Proof\_Procedure(P, U, m, \{friend(Mary, John)\}, 0.5, (\emptyset, \emptyset), 1),$$

is described in Figure 7.2. When the truth query $\top$ has to be checked, the function returns

$$(true, \sigma_2^3, \min(\beta^3, UD_{U,m}(\sigma_2^3))),$$

and thus,

$$\mathcal{P} \vdash (friend(Mary, John), 0.6).$$

Therefore, applying the WR inference rule, we get

$$\mathcal{P} \vdash (friend(Mary, John), 0.5).$$

Remark that in the third level of the proof tree, before using the fact

$$(\rightarrow age(John, about\_18), 0.9),$$

the fact

$$(\rightarrow age(Mary, young), 0.8)$$

is chosen by the proof algorithm and the semantic structure generated by the fuzzy unification algorithm for variable y is

$$y \rightarrow ([John|\,], [Mary|\,]).$$

Then, the unification degree computed for variable y is

$$N^*(m(John) \mid m(Mary)) = 0,$$

which is less than the threshold 0.5, and the function *Deductive_Proof_Procedure* backtracks.

The last example of Section 7.4.1 analyzes the unification of a variable and multiple specific and general object constants. Let

$$(buy\_price\_value(book, around\_35, between\_30\_40), 0.5)$$

be a PGL$^+\forall$ query and let $\mathcal{P} = (P, U, m)$ be a PGL$^+\forall$ program with

1. $P = \{\ (\rightarrow price(book, 34), 1),$
   $(\rightarrow value(book, about\_35), 0.8),$
   $(price(x, y) \wedge value(x, y) \rightarrow buy\_price\_value(x, y, y), 0.7)\ \};$
2. $U = \{\ U_{\texttt{product}} = \{book\},$
   $U_{\texttt{product\_price}} = [0, 100](\text{euros}),$
   $U_{\texttt{product\_value}} = [0, 100](\text{euros})\ \};$
3. $m(book) = book,$
   $m(34) = 34,$
   $m(about\_35) = [30; 35; 35; 40],$
   $m(around\_35) = [32; 34; 36; 38],$ and
   $m(between\_30\_40) = [25; 30; 40; 45].$

The proof tree generated by the function call

$$Deductive\_Proof\_Procedure(P, U, m, query, 0.5, (\emptyset, \emptyset), 1),$$

with $query = \{buy\_price\_value(book, around\_35, between\_30\_40)\}$ is described in Figure 7.3. When the truth query $\top$ is obtained, the function returns

$$(true, \sigma_2^3, \min(\beta^3, UD_{U,m}(\sigma_2^3))),$$

$\{friend(Mary, John)\}$

$(age(x,z) \wedge age(y,z) \rightarrow friend(x,y), 0.6)$

$\sigma_1^1 = (\emptyset, \{x \rightarrow ([Mary|\,], [\,|\,]), y \rightarrow ([John|\,], [\,|\,])\})$
$\sigma_2^1 = (\emptyset, \emptyset)\sigma_1^1 = \sigma_1^1$
$\beta^1 = \min(1, 0.6) = 0.6$

$\{age(x,z), age(y,z)\}$

$(\rightarrow age(Mary, young), 0.8)$

$\sigma_1^2 = (\emptyset, \{x \rightarrow ([Mary|\,], [Mary|\,]), z \rightarrow ([\,|\,], [young|\,])\})$
$\sigma_2^2 = \sigma_2^1\sigma_1^2 = (\emptyset, \{x \rightarrow ([Mary|\,], [Mary|\,]),$
$y \rightarrow ([John|\,], [\,|\,]),$
$z \rightarrow ([\,|\,], [young|\,])\})$
$UDV_{U,m}(x, SC_2^2) = N^*(m(Mary) \mid m(Mary)) = 1$
$\beta^2 = \min(0.6, 0.8) = 0.6$

$\{age(y,z)\}$

$(\rightarrow age(John, about\_18), 0.9)$

$\sigma_1^3 = (\emptyset, \{y \rightarrow ([John|\,], [John|\,]),$
$z \rightarrow ([\,|\,], [young|about\_18])\})$
$\sigma_2^3 = \sigma_2^2\sigma_1^3 = (\emptyset, \{x \rightarrow ([Mary|\,], [Mary|\,]),$
$y \rightarrow ([John|\,], [John|\,]),$
$z \rightarrow ([\,|\,], [young|about\_18\,])\})$
$UDV_{U,m}(x, SC_2^3) = UDV_{U,m}(x, SC_2^2) = 1$
$UDV_{U,m}(y, SC_2^3) = N^*(m(John) \mid m(John)) = 1$
$UDV_{U,m}(z, SC_2^3) = \max(N^*(m(young) \mid m(about\_18)),$
$N^*(m(about\_18) \mid m(young))) =$
$= \max(1, 0) = 1$
$\beta^3 = \min(0.6, 0.9) = 0.6$
$UD_{U,m}(\sigma_2^3) = \min(UDV_{U,m}(x, SC_2^3),$
$UDV_{U,m}(y, SC_2^3),$
$UDV_{U,m}(z, SC_2^3)) = 1$

$\top$

Figure 7.2: Unification of a variable and multiple specific fuzzy constants.

and thus,

$$\mathcal{P} \vdash (buy\_price\_value(book, around\_35, between\_30\_40), 0.67).$$

Finally, applying the WR inference rule, we get

$$\mathcal{P} \vdash (buy\_price\_value(book, around\_35, between\_30\_40), 0.5).$$

7.5 A translator system

In this section, we first define a system for translating PGL$^+\forall$ programs into code machine. Then, we describe a user-friendly environment that has been developed to facilitate the graphical representation of fuzzy constants and compilation tasks.

The Warren Abstract Machine (WAM), designed in 1983 by Warren (1983), is an abstract machine consisting of a memory architecture and an instruction set. In (Alsinet et al., 1995; Alsinet and Manyà, 1996) we defined a logic programming environment for a family of first-order infinitely-valued logics. This system was implemented by extending the WAM to the infinitely-valued context. From a syntactic point of view, there are only two differences between PGL$^+\forall$ and this family of infinitely-valued logics. On the one hand, terms are not only variables and precise object constants, but also imprecise and fuzzy object constants. On the other hand, unification between terms is not only syntactic, but also semantic based on a necessity-like measure. Therefore, we can easily define a translator system for PGL$^+\forall$ programs by extending the environment defined in (Alsinet et al., 1995; Alsinet and Manyà, 1996) with both fuzzy constants and the directional fuzzy unification algorithm.

In this manner, we divide the translation process in two phases. The first one translates a PGL$^+\forall$ program into an equivalent intermediate code representation which we call *possibilistic and fuzzy WAM code*. The second phase generates the target code consisting of relocatable machine code from the intermediate representation and the possibilistic and fuzzy WAM implementation.

The first phase extends the WAM target language and architecture for compiling pure Prolog to handle the semantical definition of fuzzy constants and to acomodate the implementation of certainty-weighted facts and rules. To establish our WAM extension on a von Neumann architecture, corresponding to the target code generation phase, we have chosen the high level language C++. Therefore, in (Morlán, 2000), we have developed a collection of library functions that simulates the memory organization of the WAM architecture, the sequence of low level instructions associated to each WAM instruction and the PGL$^+\forall$ directional fuzzy unification algorithm. Figure 7.4 shows the two phases that take part in the translation process.

A translator of PGL$^+\forall$ programs into possibilistic and fuzzy WAM code can be syntax-directed. That is, the translation process is driven by the syntactic structure of a program as recognized by the parser. The semantic routines,

$\{buy\_price\_value(book, around\_35, between\_30\_40)\}$

$(price(x,y) \wedge value(x,y) \rightarrow buy\_price\_value(x,y,y), 0.7)$

$\sigma_1^1 = (\emptyset, \{x \rightarrow ([book|\,], [\,|\,]),$
$\quad y \rightarrow ([around\_35|between\_30\_40], [\,|\,])\})$
$\sigma_2^1 = (\emptyset, \emptyset)\sigma_1^1 = \sigma_1^1$
$\beta^1 = \min(1, 0.7) = 0.7$

$\{price(x,y), value(x,y)\}$

$(\rightarrow price(book, 34), 1)$

$\sigma_1^2 = (\emptyset, \{x \rightarrow ([book|\,], [book|\,]),$
$\quad y \rightarrow ([around\_35|between\_30\_40], [34|\,])\})$
$\sigma_2^2 = \sigma_2^1\sigma_1^2 = \sigma_1^2$
$\beta^2 = \min(0.7, 1) = 0.7$

$\{value(x,y)\}$

$(\rightarrow value(book, about\_35), 0.8)$

$\sigma_1^3 = (\emptyset, \{y \rightarrow ([around\_35|between\_30\_40], [34|about\_35])\})$
$\sigma_2^3 = \sigma_2^2\sigma_1^3 = (\emptyset, \{x \rightarrow ([book|\,], [book|\,]),$
$\quad y \rightarrow ([around\_35|between\_30\_40], [34|about\_35])\})$
$UDV_{U,m}(x, SC_2^3) = N^*(m(book) \mid m(book)) = 1$
$UDV_{U,m}(y, SC_2^3) = \max(\min(N^*(m(between\_30\_40) \mid m(around\_35)),$
$\quad N^*(m(around\_35) \mid m(34)),$
$\quad N^*(m(around\_35) \mid m(about\_35))),$
$\quad \min(N^*(m(around\_35) \mid m(between\_30\_40)),$
$\quad N^*(m(between\_30\_40) \mid m(34)),$
$\quad N^*(m(between\_30\_40) \mid m(about\_35)))) =$
$= \max(\min(1, 1, 0.67), \min(0, 1, 1)) = 0.67$
$\beta^3 = \min(0.7, 0.8) = 0.7$
$UD_{U,m}(\sigma_2^3) = \min(UDV_{U,m}(x, SC_2^3), UDV_{U,m}(y, SC_2^3)) = 0.67$

$\top$

Figure 7.3: Unification of a variable and multiple general and specific fuzzy constants.

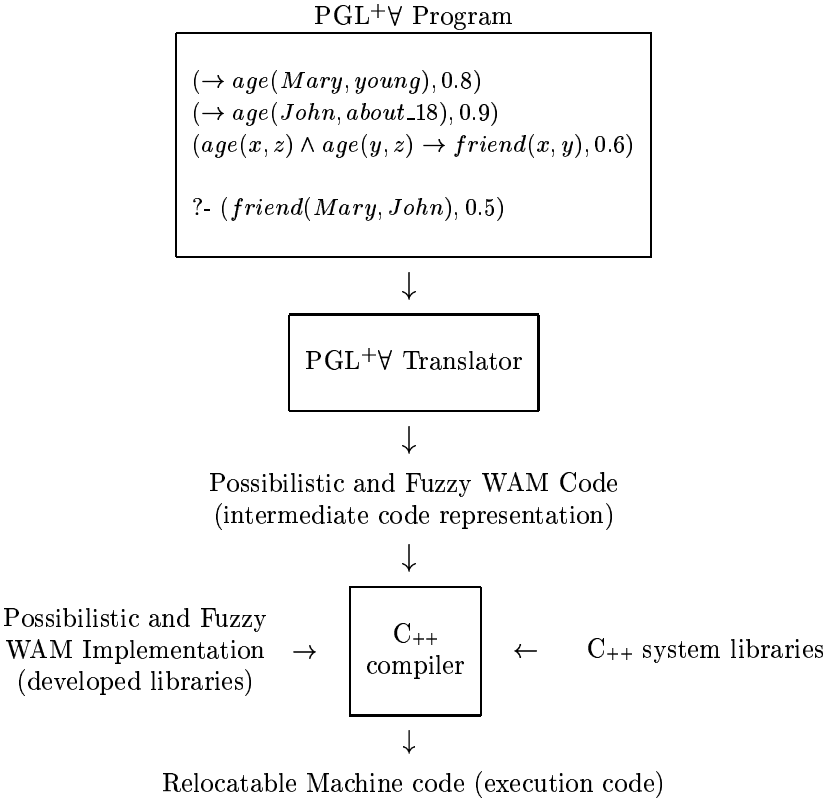

Figure 7.4: Translation process

which are the part of the translator that interprets the meaning (semantics) of a program, perform this interpretation based on its syntactic structure. On account of the possibilistic and fuzzy WAM implementation, it is necessary that permanent variables be saved in an environment associated with each activation of the procedure they appear in. Permanent variables are those that occur in more than one body atom. Then, it is necessary to know the permanent variables for a program clause before generating the suitable possibilistic and fuzzy WAM code. The solution consists of organizing the translation process around the syntactic tree structure and multiple tree traversals.

On the one hand, a bottom-up parser builds the abstract syntactic tree, which is the intermediate representation that allows translation to be decoupled from parsing, and the two semantic processing tasks –static semantic checking, and possibilistic and fuzzy WAM code generation– can be accomplished by using semantic attributes attached to the nodes of the syntactic tree and evaluated by more than one traversal of it. On the other hand, semantic routines are in charge of translating the abstract syntactic tree into a set of intermediate instructions, i.e. an abstract representation of a PGL$^+\forall$ program clause into a set of possibilistic and fuzzy WAM instructions.

The pseudocode of the most important functions involved in the possibilistic and fuzzy WAM code generation process are described bellow.

action *Code_Generation*

input
clause_list : Pointer to clause

auxiliary variables
pointer_clause : Pointer to clause
m, n : integer

begin
Sort_Clauses(*clause_list*);
pointer_clause := *First_Clause*(*clause_list*);
$n := 1$;
while (*pointer_clause* $\neq$ NILL) **do**
m := *Number_Equal_Clauses*(*pointer_clause*);
Label_Generation(*pointer_clause*);
if ($m > 1$) **then**
case n **of**
1:
Generate(`try_me_else`);
$n := n+1$;
m:
Generate(`trust_me`);
$n := 1$;
otherwise:
Generate(`retry_me_else`);

```
            n := n+1;
          end case
        end if
        Clause_Code_Generation(pointer_clause);
        pointer_clause := NextClause(pointer_clause);
      end while
    end action Code_Generation
```

The action *Sort_Clauses* groups the program clauses with equal head predicate preserving the appearing order; the function *Number_Equal_Clauses* calculates the number of clauses with the same head predicate as the actual *pointer_clause*; and the action *Label_Generation* generates the label for the set of possibilistic and fuzzy WAM instructions. `try_me_else`, `trust_me` and `retry_me_else` are WAM instructions.

```
action Clause_Code_Generation
  input
    pointer_clause: : Pointer to clause
  auxiliary variables
    pointer_body_clause : Pointer to clause
    n : integer
  begin
    n := Number_Permanent_Variables(pointer_clause);
    Generate(allocate(n));
    Get_Arguments(pointer_clause→head);
    pointer_body_clause := pointer_clause→body;
    while ( pointer_body_clause ≠ NILL ) do
      Put_Arguments(pointer_body_clause);
      pointer_body_clause := pointer_clause→body;
    end while
    Code_Generation_Weight(pointer_clause);
    Generate(deallocate)
  end action Clause_Code_Generation
```

The function *Number_Permanent_Variables* traverses the abstract syntactic tree in a left-to-right pass carrying the permanent variables for each clause. Actions *Get_Arguments* and *Put_Arguments* generate code for a head atomic formula and a body atomic formula, respectively. The action *Code_Generation_Weight* determines the computed necessity degree, the result is assigned to a global register. To preserve it, the value is saved in the activation environment of the procedure as the permanent variables are and in the `Choice_Point` as the argument registers are. The `Choice_Point` is a memory structure that stores the computation state of a resolution tree node with more

than one child and its aim is to preserve the information between the different alternatives. Finally, `allocate(n)`, `getlevel` and `deallocate` are WAM instructions.

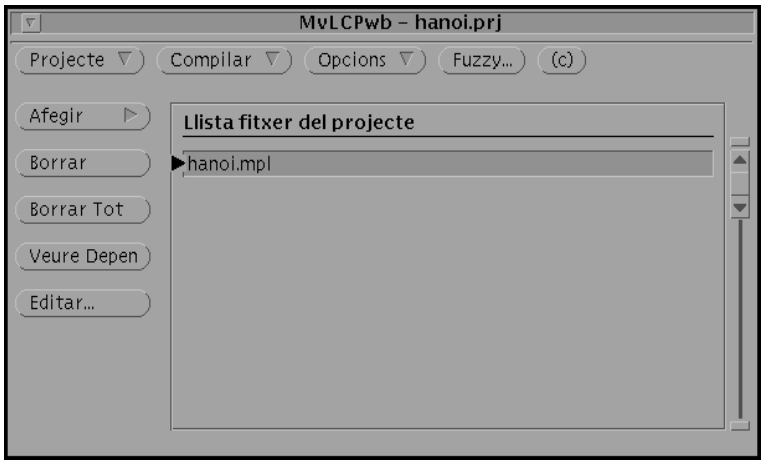

Figure 7.5: Development environment

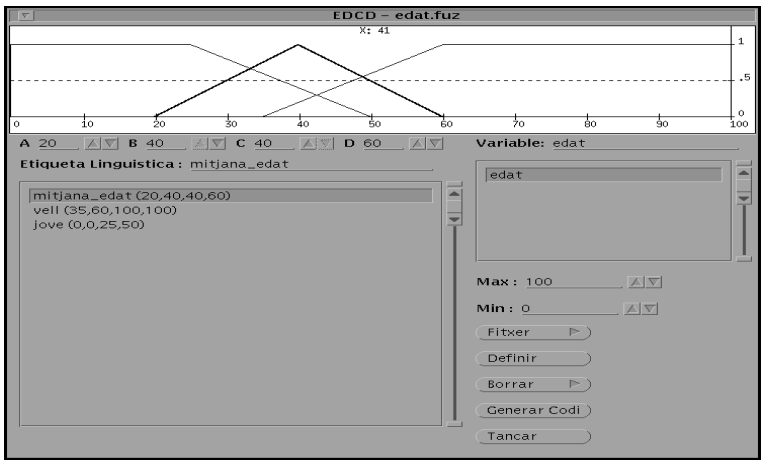

Figure 7.6: Definition of fuzzy constants

Morlán (2000) has implemented the possibilistic and fuzzy WAM library by extending the system defined in (Alsinet et al., 1995; Alsinet and Manyà, 1996) with the directional fuzzy unification algorithm. However, the implementation of the translation phase of PGL$^+\forall$ programs into possibilistic and fuzzy WAM

code, based on actions *Code_Generation* and *Clause_Code_Generation*, is not yet ready.

Finally, to facilitate the implementation and compilation tasks, we have developed a user-friendly environment under X Window. Figures 7.5 and 7.6 display the development environment and the definition of fuzzy constants windows, respectively. Currently, this environment only considers trapezoidal fuzzy set.

Chapter 8

Conclusions and future work

In the preceding chapters, we have been concerned with the formalization of logical systems for reasoning with possibilistic uncertainty and vague knowledge. As one of our ultimate objectives was to obtain competitive (from a computational point of view) algorithms for automated deduction in the frame of possibilistic logic programming with fuzzy constants, our approach has not only been confined to define formal possibilistic semantics and sound (and complete when possible) calculi, but also sketch pseudo-code of algorithms that automate their application.

Our next step should be to design suitable data structures for representing fuzzy constants and incorporate them into efficient implementations of the proof procedures, as well as to examine their behavior in real-world applications. We would like to point out that the logical systems we have defined in this thesis might be particularly useful within the framework of fuzzy deductive data bases and diagnostic systems in which symptoms are not always described accurately. They might also be valuable to implement dialog agents in multi-agent systems when vague, incomplete or imprecise information about the real world and approximately specified rules are needed.

In Chapter 1, we have already given in detail the main contributions of this thesis. Here, we summarize them briefly:

First, we have defined a formal semantics and a sound resolution-style refutation proof procedure for the necessity-valued fragment of first-order possibilistic logic extended with fuzzy constants and fuzzily restricted quantifiers (called PLFC). The PLFC resolution rule includes an implicit fuzzy unification mechanism of fuzzy events which is based on a necessity-like measure.

Second, we have formalized a semantics for a general possibilistic logic with fuzzy propositional variables (called PGL) which is based on Gödel infinitely-valued logic and, based on this, we have defined a sound deductive proof method which is based on a Hilbert-style axiomatization of the logic and

a generalized modus ponens inference rule.

Third, we have considered the Horn-rule fragment of PGL and, we have proved, for this restricted class of formulas, that the modus ponens-style calculus is complete for determining the maximum degree of possibilistic belief with which a fuzzy propositional variable can be entailed from a set of formulas.

Fourth, we have defined a fuzzy possibilistic logic programming language (called PGL$^+$) by extending the Horn-rule fragment of PGL with fuzzy constants and a semantical unification mechanism. The PGL$^+$ unification system is essentially based on a necessity-like measure, a fusion and an intersection mechanisms.

Fifth, we have formalized the notion of non-recursive and satisfiable PGL$^+$ program, we have proved that the PGL$^+$ unification mechanism preserves completeness for this restricted class of programs further satisfying a context constraint, and we have defined an efficient as possible proof procedure for computing the maximum degree of deduction of an atomic formula with fuzzy constants from a program. The PGL$^+$ proof procedure clearly departs from classical propositional interpreters and it has been developed in four sequential steps. A satisfiability testing step. A pre-processing step which ensures that all (explicit and hidden) program clauses are considered. A translation step which translates a program into a set of 1-weighted facts. And finally, a deduction step which just computes the unification degree between two fuzzy constants.

Sixth, we have defined a general first-order fuzzy possibilistic logic (called PGL$^+\forall$) which is based on Gödel predicate logic extended with fuzzy constants. Then, we have tackled logic programming aspects by considering the sublanguage of definite clauses of PGL$^+\forall$; defining, for this class of clauses, a sound SLD-style resolution procedure which is based on a directional fuzzy unification algorithm; and describing how a translator of PGL$^+\forall$ programs into machine code can be designed extending the Warren Abstract Machine to our fuzzy and possibilistic context.

The main differences between PLFC and PGL$^+\forall$ are (i) at the level of the syntax and semantics of the language; (ii) at the level of providing the language with a sound calculus; and (iii) at the level of defining an automated deduction method based on (ii).

Regarding the syntax, in PLFC, formulas are pairs of the form $(\varphi(\bar{x}), f(\bar{y}))$, where $\bar{x}$ and $\bar{y}$ denote sets of free and implicitly universally quantified variables and $\bar{y} \supseteq \bar{x}$, $\varphi(\bar{x})$ is a disjunction of literals with fuzzy constants, and $f(\bar{y})$ is a valid valuation function which expresses the certainty of $\varphi(\bar{x})$ in terms of necessity measures. Basically, valuation functions $f(\bar{y})$ are constant values and variable weights which are not considered in PGL$^+\forall$. Variable weights are suitable for modeling statements of the form "the more x is A (or x belongs to A), the more certain is $p(x)$", where A is a fuzzy set.

In PGL$^+\forall$, formulas are pairs of the form (φ, α), where φ is a first-order definite clause or a query with fuzzy constants and regular predicates and $\alpha \in [0, 1]$ is a lower bound on the belief on φ in terms of necessity measures.

On the other hand, in PLFC, fuzzy constants can be seen as (flexible) restric-

tions on an existential quantifier. In general, "$L(B)$ is true at least to degree α", is represented in PLFC as $(L(B), \alpha)$, where L is either a positive or a negative literal and B is a fuzzy constant. Therefore, as the proof method for PLFC is defined by refutation through a generalized resolution rule between positive and negative literals, the unification (in the classical sense) between fuzzy constants is not allowed. However, as variable weights are interpreted as conjunctive (fuzzy) knowledge, an implicit semantical unification between fuzzy events is performed between variable weights and fuzzy constants.

In PGL$^+\forall$, fuzzy constants are interpreted as disjunctive fuzzy knowledge too, but in contrast to PLFC, we do not have negative literals in the language; we have provided the language with a sound modus ponens-style calculus by derivation based on an explicit unification mechanism of fuzzy constants; and we have proved, for clauses with only fuzzy constants (i.e. for the PGL$^+$ sublanguage of PGL$^+\forall$), that completeness can be gained extending the system with a fusion and an intersection mechanisms between fuzzy constants.

Regarding the semantics, because of the fuzzy information, the truth evaluation of formulas is many-valued in both systems, and belief states are modeled by normalized possibility distributions on a set of many-valued interpretations, also in both systems. However, the basic connectives of PLFC are negation $\neg$ and disjunction $\vee$ while in PGL$^+\forall$, they are conjunction $\wedge$ and implication $\rightarrow$, and the semantics for the two sets of connectives are not equivalent, i.e. the two sets of connectives are not inter-definable. Moreover, the extended necessity measure for fuzzy sets suggested by Dubois et al. (1994c), which is used in PGL$^+\forall$ for setting the possibilistic semantics of formulas, is different from the one used in PLFC, although both are extensions of the standard necessity measure for crisp sets.

Regarding automated deduction we have developed a sound resolution-style refutation procedure for PLFC which is based on the computation of the necessity evaluation of fuzzy events. In order to get PLFC clauses with the greatest possible weights (i.e. to get higher unification degrees), a fusion mechanism must be applied after each resolution step. Therefore, the refutation procedure cannot be oriented to a resolvent clause and the search space consists of all possible orderings of the literals in the knowledge base. As already pointed out, in order to gain completeness, a fusion mechanism is also needed for PGL$^+\forall$. Therefore, from a computational point of view, the extension of a knowledge base with all (explicit and hidden) clauses through a fusion mechanism is a drawback in both systems. However, in contrast to PLFC, we have shown that the fusion mechanism for PGL$^+\forall$ clauses with only fuzzy constants (i.e. the PGL$^+$ sublanguage) can be performed in a pre-processing step of the knowledge base. Hence, for PGL$^+\forall$, we have developed an SLD-style resolution procedure oriented to queries which is based on a generalized modus ponens inference rule and a directional fuzzy unification algorithm.

In table 8.1, we summarize the main differences between the necessity-valued fragment of possibilistic logic (PL) and the formalization provided in this thesis for its extension with fuzzy constants and fuzzily restricted quantifiers (PLFC).

	PL	**PLFC**
Syntax	$(\neg p(x) \vee q(a), \alpha)$ a: crisp constant Regular predicates	$(\neg p(x) \vee q(A), \min(\alpha, B(x)))$ A and B: fuzzy sets Typed regular predicates
Interpr.	$\mathbf{M} = (U, i, m)$, $m(a) \in U$ $\|\varphi\|_{\mathbf{M}} \in \{0, 1\}$ (Boolean) $[\varphi]$: crisp set of models	$\mathbf{M} = (U, i, m)$, $m(A) : U \to [0, 1]$ $\|\varphi\|_{\mathbf{M}} \in [0, 1]$ (many-valued) $[\varphi]$: fuzzy set of models
Poss. Models	$\mathcal{M} = \{\mathbf{M}\}$ $\pi : \mathcal{M} \to [0, 1]$ $\pi \models (\varphi, \alpha)$ iff $N([\varphi] \mid \pi) \geq \alpha$ iff $\pi(\mathbf{M}) \leq 1 - \alpha$ $\forall\, \mathbf{M} \in \mathcal{M}$ with $\|\varphi\|_{\mathbf{M}} = 0$	$\mathcal{M}_{U,m} = \{\mathbf{M} = (U, i, m)\}$ $\pi : \mathcal{M}_{U,m} \to [0, 1]$ $\pi \models (\varphi, \alpha)$ iff $N([\varphi] \mid \pi) \geq \alpha$ iff $\pi(\mathbf{M}) \leq 1 - \alpha$ $\forall\, \mathbf{M} \in \mathcal{M}_{U,m}$ with $\|\varphi\|_{\mathbf{M}} < \alpha$
Resolution	$\frac{(\neg p(x) \vee q(x), \alpha), (p(b), \beta)}{(q(b), \min(\alpha, \beta))}$	$\frac{(\neg p(x) \vee q(x), \min(\alpha, A(x))), (p(B), \beta)}{(q(B), \min(\alpha, \beta, N(A \mid [B]_\beta)))}$
Proof method	Refutation by resolution	Refutation by resolution
Unification	Syntactic (classic)	Semantic: $N(A \mid [B]_\beta)$
Sound./Compl.	Soundness and Completeness	Soundness

Table 8.1: Possibilistic logic and PLFC

And in table 8.2 we summarize the main differences between PLFC and PGL$^+\forall$.

Apart from the implementation of the proof procedures we have introduced in this thesis, our ongoing research is addressed to extend semantical unification to allow a similarity-based unification of object constants.

In fact, in both PLFC and PGL$^+\forall$, the matching degree between two object (fuzzy) constants is computed in terms of a necessity measure for fuzzy sets. As a consequence, the unification degree between two different and precise object constants is null in both systems. Sometimes this is a rather unpleasant behavior, especially if we are trying to model approximate knowledge. To remedy this situation, in (Alsinet and Godo, 2001a) we have addressed the issue of extending the (graded) unification of fuzzy constants to cope with a similarity-based unification of precise object constants. For simplicity and practical reasons, we have focused on a sublogic of Horn-like PLFC clauses expressing disjunctive informa-

	PLFC	**PGL$^+\forall$**
Syntax	$(\neg age(x,y) \vee salary(x,low), young(y))$ Disjunctive + Conjunctive fuzzy constants	$(age(x,young) \rightarrow salary(x,low), 1)$ Disjunctive fuzzy constants
Interpr.	$\mathbf{M} = (U, i, m)$ $m(A) : U \rightarrow [0,1]$ $i(p) \subseteq U$ (conjunctive inter.) $v(x) \in U$ (crisp evaluations) $\|p(A)\|_{\mathbf{M}} = \sup_{u \in i(p)} \mu_{m(A)}(u)$ $\|\neg q(B)\|_{\mathbf{M}} = \sup_{u \notin i(q)} \mu_{m(B)}(u)$ $\|\neg q(B) \vee p(A)\|_{\mathbf{M}} =$ $\max(\|\neg q(B)\|_{\mathbf{M}}, \|p(A)\|_{\mathbf{M}})$	$\mathbf{M} = (U, i, m)$ $m(A) : U \rightarrow [0,1]$ $i(p) \in U$ (disjunctive inter.) $v(x) : U \rightarrow [0,1]$ (fuzzy evaluations) $\|p(A)\|_{\mathbf{M}} = \mu_{m(A)}(i(p))$ $\|q(B) \rightarrow p(A)\|_{\mathbf{M}}$ Gödel truth functions
Poss. Models	$\pi : \mathcal{M}_{U,m} \rightarrow [0,1]$ $\pi \models (\varphi, \alpha)$ iff $N([\varphi] \mid \pi) \geq \alpha$ $N([\varphi] \mid \pi) = \inf_{\mathbf{M}} \max(1 - \pi(\mathbf{M}), \|\varphi\|_{\mathbf{M}})$	$\pi : \mathcal{M}_{U,m} \rightarrow [0,1]$ $\pi \models (\varphi, \alpha)$ iff $N^*([\varphi] \mid \pi) \geq \alpha$ $N^*([\varphi] \mid \pi) = \inf_{\mathbf{M}} \pi(\mathbf{M}) \Rightarrow \|\varphi\|_{\mathbf{M}}$
Unifi.	Variable weights/ Imprecise constants	Fuzzy constants/ Fuzzy constants
Proof method	Refutation by resolution (implicit unification) Fusion over each resolvent clause	Modus ponens + Unification rule PGL$^+$: Fusion in a pre-processing step + intersection mechanism
Sound./ Compl.	Soundness	PGL$^+$: completeness for goals (context constraint)

Table 8.2: PLFC and PGL$^+\forall$

tion, and thus, fuzzy constants are only allowed in the head of a clause. Then, each precise object constant appearing in the body of a clause is fuzzified by means of a similarity relation, and the fuzzified constant is placed as a variable weight. With this we enlarge the applicability of the clause to constants *close* to the original ones. For instance, the set of Horn PLFC clauses

$$\{\ (price\_book(34), 1),\\ (price\_book(35) \rightarrow buy\_book, 1)\ \},$$

are transformed into

$$\{\ (price\_book(34), 1),\\ (price\_book(x) \rightarrow buy\_book, around\_35(x))\ \},$$

where we take $\mu_{around\_35}(x) = S(35, x)$ with

$$S : D_{\texttt{product\_price}} \times D_{\texttt{product\_price}} \rightarrow [0, 1]$$

being a fuzzy proximity (i.e. reflexive) relation on the domain $D_{\texttt{product\_price}}$ of the sort `product_price`. Now, applying the PLFC resolution rule we get

$$(buy\_book, N(around\_35 \mid 34)),$$

and in this case we have

$$N(around\_35 \mid 34) = \mu_{around\_35}(34) = S(35, 34).$$

Hence,

$$(buy\_book, S(35, 34))$$

is the derived clause after using the proposed similarity-based unification mechanism.

The similarity-based fuzzification mechanism of precise constants can be easily extended to fuzzy constants themselves. On the other hand, the proposed methodology can be used solely as a pure similarity-based reasoning with classical (precise) constants (that is, with no fuzzy constants at all). The comparison of the resulting system with the ones proposed by (Arcelli et al., 1998; Gerla and Sessa, 1999; Formato et al., 2000) on the one hand, and the one proposed by (Vinař and Vojtáš, 2000; Vojtáš et al., 2001b), in different frameworks, will be a matter of high interest.

Finally, we would like to point out two other future research perspectives.

First, we aim to extend the current first-order inference systems (which are only sound), which would allow us to have a complete possibilistic entailment. Actually, as we have already mentioned, the PGL$^+$ proof procedure is complete for determining the possibilistic entailment degree of an atomic formula with fuzzy constants from a restricted class of programs. Therefore, in order to gain completeness in both PLFC and PGL$^+\forall$, it will be necessary at least to extend the PGL$^+$ fusion and intersection mechanisms to the first-order case. On the other hand, when restricting ourselves to definite clauses, the fusion mechanisms

can be applied as a pre-processing step of the knowledge base. Hence, in order to define an SLD-style refutation procedure for PLFC it seems necessary to consider a suitable Horn-like sublanguage of PLFC.

Second, we aim at extending the base languages to allow *fuzzy computable functions*. A fuzzy computable function is an operation on fuzzy sets which can be computed as soon as all its arguments are instantiated. Some useful fuzzy computable functions are *standard fuzzy set theoretic operations* (i.e. fuzzy complement, intersection and union) and *arithmetic operations $*$ on fuzzy numbers* by means of the extension principle (i.e. $\mu_{A*B}(w) = \sup_{w=u*v} \min(\mu_A(u), \mu_B(v))$).

For instance, when extending PGL$^+\forall$ with fuzzy computable functions we obtain a unified framework for the treatment of fuzzy constants and variable weights (i.e. fuzzily restricted quantifiers). For instance, the PLFC clause

$$(\neg age(x,z) \vee \neg age(y,t) \vee friend(x,y), approx\_equal(z,t)),$$

where $approx\_equal(\cdot,\cdot)$ is a fuzzy relation with fuzzy arguments, could be represented in PGL$^+\forall$ as

$$(age(x,z) \wedge age(y,t) \wedge distance(z,t,approx\_zero) \rightarrow friend(x,y), 1),$$

where $approx\_zero$ is a fuzzy constant and $distance(\cdot,\cdot,\cdot)$ is a classical predicate with fuzzy arguments that would be defined through the fuzzy set subtraction operation $\ominus$ in the following way:

$$(\rightarrow distance(x,y,x \ominus y), 1).$$

In possibilistic terms, the PLFC clause

$$(\neg age(x,z) \vee \neg age(y,t) \vee friend(x,y), approx\_equal(z,t))$$

is interpreted as

"the more two persons are about a same age,
the more certain they are friends",

and the PGL$^+\forall$ clause

$$(age(x,z) \wedge age(y,t) \wedge distance(z,t,approx\_zero) \rightarrow friend(x,y), 1)$$

as

"the more certain the (fuzzy) distance between the age
of two persons is close to zero, the more certain they are friends".

Thus, from a semantic point of view, fuzzily restricted quantifiers can be handled by means of fuzzy constants and computable functions. Furthermore, when resolving the PLFC clause

$$(\neg age(x,z) \vee \neg age(y,t) \vee friend(x,y), approx\_equal(z,t))$$

with the set of PLFC clauses

$$\{(age(Mary, around\_19), 0.8), (age(John, about\_18), 0.7)\},$$

where $around\_19$ and $about\_18$ are two fuzzy constants, we get

$$(friend(Mary, John), \min(0.8, 0.7, approx\_equal([around\_19]_{0.8}, [about\_18]_{0.7}))),$$

where $approx\_equal([around\_19]_{0.8}, [about\_18]_{0.7})$ is computed as

$$N(approx\_equal \mid \ [around\_19]_{0.8} \times [about\_18]_{0.7}),$$

where $\times$ stands for the Cartesian product, and hence, extending the necessity measure $N(\cdot \mid \cdot)$ to general fuzzy relations. On the other hand, from the following set of PGL$^+\forall$ clauses:

$$\begin{array}{l}\{\ (\rightarrow age(Mary, around\_19), 0.8),\\ \quad (\rightarrow age(John, about\_18), 0.7),\\ \quad (age(x, z) \wedge age(y, t) \wedge distance(z, t, approx\_zero) \rightarrow friend(x, y), 1),\\ \quad (\rightarrow distance(x, y, x \ominus y), 1)\ \}\end{array}$$

we infer that

$$(friend(Mary, John), \min(0.8, 0.7, N^*(approx\_zero \mid around\_19 \ominus about\_18))),$$

and thus, from a syntactic point of view, fuzzily restricted quantifiers can be handled by means of fuzzy constants and computable functions as well.

Finally, other benefits of considering computable functions are to handle imprecise temporal knowledge in the possibilistic logic framework as proposed by Sandri and Godo (1999), as well as recursively define concepts. For instance, the Sorites paradox

> "a heap of sand with $X - 1$ grains is large,
> if a heap of sand with X grains is large"

could be represented in PGL$^+\forall$ extended with fuzzy computable functions as

$$(large\_heap(x) \wedge weighted\_dis(x, x \ominus 1, close\_zero) \rightarrow large\_heap(x \ominus 1), 0.9),$$

where $close\_zero$ is a fuzzy constant and $weighted\_dis(\cdot, \cdot, \cdot)$ is a classical predicate with fuzzy arguments which would be defined through the fuzzy set division operation $\oslash$ in the following way:

$$(\rightarrow weighted\_dis(x, y, (x \ominus y) \oslash \max(x, y)), 1),$$

where max is the extension to fuzzy sets of the usual max-operation by means of the extension principle.

References

Adlassnig, K., Scheithauer, W. and Kolarz, G. (1986). Fuzzy medical diagnosis in a hospital. In Prade, H. and Negoita, C., editors, *Fuzzy Logic in Knowledge Engineering*, volume 86 of *Interdisciplinary Systems Research Series*, pages 275–294. Verlag TÜV Rheinland, Köln.

Aït-Kaci, H. (1991). *Warren's Abstract Machine. A Tutorial Reconstruction.* MIT Press.

Alsina, C., Trillas, E. and Valverde, L. (1983). On some logical connectives for fuzzy set theory. *Journal Math. Anal. Appl.*, 93:15–26.

Alsinet, T., Manyà, F., Jové, L. and Morillo, J. (1995). A multiple-valued logic programming system: Definition and implementation. In *Actas de la VI Conferencia de la Asociación Española para la Inteligencia Artificial, CAEPIA-95*, pages 37–45, Alicante, Spain.

Alsinet, T. and Manyà, F. (1996). A declarative programming environment for infinitely-valued logics. In *Proceedings of the Sixth International Conference on Information Procesing and Management of Uncertainty in Knowledge-Based Systems, IPMU-96*, volume 3, pages 1205–1210, Granada, Spain.

Alsinet, T. (1997a). Algorisme d'unificació fuzzy direccional. *Butlletí de l'Associació Catalana d'Intel·ligència Artificial*, 12:109–117. Actes de les Jornades d'Intel·ligència Artificial: Noves Tendències, Lleida, Spain.

Alsinet, T. (1997b). Una primera aproximación a la unificación fuzzy vía relaciones de compatibilidad. In *Actas del VII Congreso Español sobre Tecnologías y Lógica Fuzzy, ESTYLF-97*, pages 159–164, Tarragona, Spain.

Alsinet, T. and Godo, L. (1998). Fuzzy unification degree. In *Proceedings of the Second International Workshop on Logic Programming and Soft Computing*, pages 23–43, Manchester, UK.

Alsinet, T., Godo, L. and Sandri, S. (1999). On the semantics and automated deduction for PLFC, a logic of possibilistic uncertainty and fuzziness. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, UAI-99*, pages 3–12, Stockholm, Sweden. Morgan Kaufmann.

Alsinet, T. and Godo, L. (2000a). A complete calculus for possibilistic logic programming with fuzzy propositional variables. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, UAI-2000*, pages 1–10, Stanford, CA. Morgan Kaufmann.

Alsinet, T. and Godo, L. (2000b). A complete proof method for possibilistic logic programming with semantical unification of fuzzy constants. In *Actas del X Congreso Español sobre Tecnologías y Lógica Fuzzy, ESTYLF-2000*, pages 279–284, Sevilla, Spain.

Alsinet, T. and Godo, L. (2001a). Adding similarity to possibilistic logic with fuzzy constants. In *Proceedings of the Ninth International Fuzzy Systems Association World Congress, IFSA-2001*, pages 1535–1540, Vancouver, British Columbia, Canada.

Alsinet, T. and Godo, L. (2001b). A proof procedure for possibilistic logic programming with fuzzy constants. In *Proceedings of the Sixth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU-2001*, pages 760–771, Toulouse, France. Springer, LNAI 2143.

Alsinet, T. and Godo, L. (2002). Towards an automated deduction system for first-order possibilistic logic programming with fuzzy constants. *International Journal of Intelligent Systems*, 17:887–924.

Appelbaum, L. and Ruspini, E. (1985). ARIES: An approximate reasoning inference engine. In Gupta, M., Kandel, A., Bandler, W. and Kiszka, J., editors, *Approximate Reasoning in Expert Systems*, pages 745–765. Noth-Holland.

Apt, K. (1990). Logic programming. In Leeuwen, J. V., editor, *Handbook of Theoretical Computer Science*, volume B, chapter 10. Elsevier.

Arcelli, F., Formato, F. and Gerla, G. (1998). Fuzzy unification as a foundation of fuzzy logic programming. In Arcelli, F. and Martin, T., editors, *Logic Programming and Soft Computing*, chapter 3, pages 51–68. Research Studies Press.

Baaz, M., Fermüller, C., Ovrutcki, A. and Zach, R. (1993). MULTLOG: A system for axiomatising many-valued logics. In Voronkov, A., editor, *Proceedings of the Fourth International Conference on Logic Programming and Automated Theorem Proving, LPAR-93*, pages 345–347. Springer-Verlag, LNAI 698.

Baaz, M. and Fermüller, C. (1995). Resolution-based theorem proving for many-valued logics. *Journal of Symbolic Computation*, 19:353–391.

Baaz, M., Fermüller, C., Salzer, G. and Zach, R. (1996). MUltlog 1.0: Towards an expert system for many-valued logics. In *Proceedings of the Thirteenth International Conference on Automated Deduction, CADE-96*, pages 226–230. Springer-Verlag, LNAI 1104.

Baaz, M., Hájek, P., Svejda, D. and Krajíćek, J. (1998). Embedding logics into product logic. *Studia Logica*, 61(1):35–47.

Bacchus, F. (1990). *Representing and Reasoning with Probabilistic Knowledge*. MIT Press, Cambridge, Mass.

Baldwin, J. and Pilsworth, B. (1979). Fuzzy truth definition of possibility measure for decision classification. *International Journal Man-Machine Studies*, 11:351–380.

Baldwin, J. (1981). Fuzzy logic and fuzzy reasoning. In Mandani, E. and Gaines, B., editors, *Fuzzy Reasoning and its Applications*, pages 133–148. Academic Press.

Baldwin, J. (1986). Support logic programming. *International Journal of Intelligent Systems*, 1:73–104.

Baldwin, J. (1987a). Evidential support logic programming. *Fuzzy Sets and Systems*, 24:1–26.

Baldwin, J., Martin, T. and Pilsworth, B. (1987b). The implementation of FPPROLOG: A fuzzy Prolog interpreter. *Fuzzy Sets and Systems*, 23:119–129.

Baldwin, J., Martin, T. and Pilsworth, B. (1993). Fril: A support logic programming system. In Hand, D., editor, *AI and Computer Power: The Impact on Statistics*, pages 129–149. Chapman and Hall.

Baldwin, J., Martin, T. and Pilsworth, B. (1995). *Fril-Fuzzy and Evidential Reasoning in Artificial Intelligence*. Research Studies Press.

Baldwin, J. and Martin, T. (1996). Fuzzy classes in object-oriented logic programming. In *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, Fuzz-IEEE-96*, pages 1358–1364, New Orleans, Louisiana. IEEE Press.

Beckert, B., Hähnle, R., Oel, P. and Sulzmann, M. (1996). The many-valued tableau-based theorem prover $_3T^AP$: Version 4.0. In *Proceedings of the Thirteenth International Conference on Automated Deduction, CADE-96*, pages 303–307. Springer-Verlag, LNAI 1104.

Beckert, B., Hähnle, R. and Manyà, F. (2000). The SAT problem of signed CNF formulas. In Basin, D., D'Agostino, M., Gabbay, D., Matthews, S. and Viganò, L., editors, *Labelled Deduction*, volume 17 of *Applied Logic Series*, pages 61–82. Kluwer.

Bel, G., Farreny, D. and Prade, H. (1986). Towards the use of fuzzy rule-based systems in the monitoring of manufacturing systems. In Crestin, J. and Waters, J. M., editors, *Software for Discrete Manufacturing*, pages 525–535. North-Holland.

Bénéjam, J. (1986). La méthode de beth pour la construction de modèles en logique à valeurs réelles. In *Proceedings of the First International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU-86*, pages 393–396, Paris, France.

Benferhat, S., Dubois, D. and Prade, H. (1997). Nonmonotonic reasoning, conditional objects and possibility theory. *Artificial Intelligence*, 92:259–276.

Bonissone, P., Gans, S. and Decker, K. (1987). RUM: A layered architecture for reasoning with uncertainty. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence, IJCAI-87*, pages 891–898, Milano, Italy. Morgan Kaufmann.

Bouchon-Meunier, B., Dubois, D., Godo, L. and Prade, H. (1999). Fuzzy sets and possibility theory in approximate and plausible reasoning. In Bezdek, J., Dubois, D. and Prade, H., editors, *Fuzzy Sets in Approximate Reasoning and Information Systems*, Fuzzy Sets Series, pages 15–190. Kluwer.

Buchanan, B. and Shortliffe, E. (1984). *Rule-Based Expert Systems – The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley.

Cao, T. and Creasy, P. (2000). Fuzzy types: A framework for handling uncertainty about types of objects. *International Journal of Approximate Reasoning*, 25(3):217–253.

Cayrol, M., Farreny, H. and Prade, H. (1982). Fuzzy pattern matching. *Kybernetes*, 11:103–116.

Cignoli, R., Esteva, F., Godo, L. and Torrens, A. (2000). Basic fuzzy logic is the logic of continuous t-norms and their residua. *Soft Computing*, 4:106–112.

Comerauer, A., Kanoui, H., Roussel, P. and Pasero, R. (1973). Un système de communication homme-machine en français. Technical report, Groupe de Recherche en Intelligence Artificielle, Université d'Aix-Marseille.

da Costa, N., Henschen, L., Lu, J. and Subrahmanian, V. (1990). Automatic theorem proving in paraconsistent logics: Theory and implementations. In Stickel, M., editor, *Proceedings of the Tenth International Conference on Automated Deduction, CADE-90*, pages 72–86. Springer-Verlag, LNCS 449.

Davis, M. and Putnam, H. (1960). A computing procedure for quantification theory. *Journal of the ACM*, 7:201–215.

De Baets, B., Esteva, F., Fodor, J. and Godo, L. (2001). Systems of ordinal fuzzy logic with application to preference modeling. *Fuzzy Sets and Systems*, 124(3):353–359.

Di Zenzo, S. (1988). A many-valued logic for approximate reasoning. *IBM Journal of Research and Development*, 32(4):552–565.

Dubois, D. and Prade, H. (1986). Truth, vagueness and uncertainty - On a frequent misunderstanding in approximate reasoning. In *Proceedings of the North-American Fuzzy Information Processing Society Conference*, pages 52–56, San Francisco, CA.

Dubois, D., Lang, J. and Prade, H. (1987a). Theorem-proving under uncertainty - A possibility theory-based approach. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence, IJCAI-87*, pages 984–986, Milano, Italy. Morgan Kaufmann.

Dubois, D. and Prade, H. (1987b). Necessity measures and the resolution principle. *IEEE Transactions on Systems, Man and Cybernetics*, 17:474–478.

Dubois, D., Prade, H. and Testemale, C. (1988a). In search of a modal system for possibility theory. In *Proceedings of the Eighth European Conference on Artificial Intelligence, ECAI-88*, pages 501–506, Munich, Germany. Pitman Publishing.

Dubois, D., Prade, H. and Testemale, C. (1988b). Weighted fuzzy pattern matching. *Fuzzy Sets and Systems*, 28:313–331.

Dubois, D. (1989a). Fuzzy knowledge in an artificial intelligence system for job-shop scheduling. In Evans, G., Karwowski, W. and Wilhelm, M., editors, *Applications of Fuzzy Set Methodologies in Industrial Engineering Advances*, pages 73–89. Elsevier.

Dubois, D. and Prade, H. (1989b). Handling uncertainty in expert systems - Pitfalls, difficulties, remedies. In Hollnagel, E., editor, *The Reliability of Expert Systems*, pages 65–118. Ellis Horwood.

Dubois, D. and Prade, H. (1990). Resolution principles in possibilistic logic. *International Journal of Approximate Reasoning*, 4:1–21.

Dubois, D., Lang, J. and Prade, H. (1991a). Fuzzy sets in approximate reasoning - Part 2: Logical approaches. *Fuzzy Sets and Systems*, 40(1):203–244.

Dubois, D., Lang, J. and Prade, H. (1991b). Towards possibilistic logic programming. In *Proceedings of the Joint International Conference and Symposium on Logic Programming*, pages 581–595, Paris, France.

Dubois, D. and Prade, H. (1991c). Fuzzy sets in approximate reasoning - Part 1: Inference with possibility distributions. *Fuzzy Sets and Systems*, 40(1):143–202.

Dubois, D., Lang, J. and Prade, H. (1994a). Automated reasoning using possibilistic logic: Semantics, belief revision and variable certainty weights. *IEEE Transactions on Data and Knowledge Engineering*, 1(6):64–71.

Dubois, D., Lang, J. and Prade, H. (1994b). Handling uncertainty, context vague predicates and partial inconsistency in possibilistic logic. In Driankov, D., Eklund, P. and Ralescu, A., editors, *Fuzzy Logic and Fuzzy Control*, pages 45–55. Springer-Verlag, LNAI 833.

Dubois, D., Lang, J. and Prade, H. (1994c). Possibilistic logic. In Gabbay, D., Hogger, C. and Robinson, J., editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, Nonmonotonic Reasoning and Uncertain Reasoning, pages 439–513. Claredon Press.

Dubois, D., Prade, H. and Sandri, S. (1996). Possibilistic logic augmented with fuzzy unification. In *Proceedings of the Sixth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU-96*, pages 1009–1014, Granada, Spain.

Dubois, D. and Prade, H. (1998a). *Possibility Theory: An Approach to Computerized Processing of Uncertainty*. Plenum Press.

Dubois, D., Prade, H. and Sandri, S. (1998b). Possibilistic logic with fuzzy constants and fuzzily restricted quantifiers. In Arcelli, F. and Martin, T., editors, *Logic Programming and Soft Computing*, chapter 4, pages 69–90. Research Studies Press.

Duda, R., Gaschnig, J. and Hart, P. (1981). Model design in the prospector consultant system for mineral exploration. In Michie, D., editor, *Expert Systems in the Microelectronic Age*, pages 153–167. Edinburgh University Press.

Escalada-Imaz, G. and Manyà, F. (1994). An approach to first-order multiple-valued logic programming. In *Actas del IV Congreso Español sobre Tecnologías y Lógica Fuzzy, ESTYLF-94*, pages 151–156, Blanes, Spain.

Escalada-Imaz, G. and Manyà, F. (1995). Efficient interpretation of propositional multiple-valued logic programs. In Bouchon-Meunier, B., Yager, R. and Zadeh, L., editors, *Advances in Intelligent Computing*, pages 428–439. Springer-Verlag, LNCS 945.

Escalada-Imaz, G., Manyà, F. and Sobrino, A. (September 1996). Principles of logic programming with uncertain information. description of some of the most relevant systems. *Theoria*, 11(27):123–148.

Esteva, F., Godo, L., Hájek, P. and Navara, M. (2000). Residuated fuzzy logics with an involutive negation. *Archive for Mathematical Logic*, 39:103–124.

Feingenbaum, E. (1987). Knowledge engineering in the 1980's. Technical report, Department of Computer Science, Stanford University, Stanford, CA.

Formato, F. (1998). *On Similarity and its Application to Logic Programming*. PhD thesis, Universitá di Napoli Federico II.

Formato, F., Gerla, G. and Sessa, M. (2000). Similarity-based unification. *Fundamenta Informaticae*, 40:1–22.

Freksa, C. (1981). *Linguistic Pattern Characterization and Analysis*. PhD thesis, University of California, Berkeley.

Gerla, G. and Sessa, M. (1999). Similarity in logic programming. In Chen, G., Ying, M. and Cai, K., editors, *Fuzzy Logic and Soft Computing*, chapter 2, pages 19–31. Kluwer.

Gilmore, P. C. (1960). A proof method for quantification theory. *IBM Journal Res. Develop*, 4:28–35.

Gödel, K. (1932). Zum intuitionistischen aussagenkalkül. *Anzeiger Akademie der Wissenschaften Wien, mathematisch-naturwiss. Klasse*, 32:65–66. Reprinted and translated in (Gödel, 1986).

Gödel, K. (1986). *Collected Works: Publications 1929–1936*, volume 1. Oxford University Press. Edited by S. Feferman, J. Dawson and S. Kleene.

Godo, L., López de Mántaras, R., Sierra, C. and Verdaguer, A. (1987). Managing linguistically expressed uncertainty in MILORD - Application to medical diagnosis. In *Proceedings of the Seventh International Workshop on Expert Systems and their Applications*, pages 571–596, Avignon, France.

Godo, L., López de Mántaras, R., Sierra, C. and Verdaguer, A. (1989). MILORD: The architecture and the management of linguistically expressed uncertainty. *International Journal of Intelligent Systems*, 4:471–501.

Godo, L. (1990). *Contribució a l'Estudi de Models d'inferència en els Sistemes Possibilístics*. PhD thesis, Facultat d'Informàtica de Barcelona, Universitat Politècnica de Catalunya.

Godo, L. and Vila, L. (1995). Possibilistic temporal reasoning based on fuzzy temporal constraint. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI-95*, pages 1916–1922, Montreal, Canada. Morgan Kaufmann.

Godo, L., Esteva, F. and Hájek, P. (2000). Reasoning about probability using fuzzy logic. *Neural Network World*, 10(5):811–824.

Godo, L., Hájek, P. and Esteva, F. (2001). A fuzzy modal logic for belief functions. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI-2001*, volume 1, pages 723–729, Seattle, WA. Morgan Kaufmann.

Gottwald, S. (1999). Many-valued logic and fuzzy set theory. In Höhle, U. and Rodabaugh, S., editors, *Mathematics of Fuzzy Sets: Logic, Topology, and Measure Theory*, Fuzzy Sets Series, pages 5–90. Kluwer.

Gottwald, S. (2001). *A Treatise on Many-Valued Logics*. Research Studies Press.

Hähnle, R. (1994a). *Automated Deduction in Multiple-Valued Logics*, volume 10 of *International Series of Monographs in Computer Science*. Oxford University Press.

Hähnle, R. (1994b). Short conjunctive normal forms in finitely-valued logics. *Journal of Logic and Computation*, 4(6):905–927.

Hähnle, R. (1996). Exploiting data dependencies in many-valued logics. *Journal of Applied Non-Classical Logics*, 6:49–69.

Hähnle, R. and Escalada-Imaz, G. (1997). Deduction in many-valued logics: A survey. *Mathware and Soft Computing*, 4(2):69–97.

Hähnle, R. (2001). Advanced many-valued logics. In Gabbay, D., editor, *Handbook of Philosophical Logic*, volume D2: Classical Logics 2, chapter 5. Kluwer, Dordrecht, second edition.

Hájek, P. (1995a). Fuzzy logic and arithmetical hierarchy. *Fuzzy Sets and Systems*, 73:359–363.

Hájek, P., Godo, L. and Esteva, F. (1995b). Fuzzy logic and probability. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, UAI-95*, pages 237–244, Montreal, Canada. Morgan Kaufmann.

Hájek, P., Godo, L. and Esteva, F. (1996). A complete many-valued logic with product-conjunction. *Archive for Mathematical Logic*, 35:191–208.

Hájek, P. and Godo, L. (1997). Deductive systems of fuzzy logic. *Tatra Mountains Math. Publ.*, 13:35–66.

Hájek, P. (1998a). Basic fuzzy logic and BL-algebras. *Soft Computing*, 2:124–128.

Hájek, P. (1998b). *Metamathematics of Fuzzy Logic*. Kluwer.

Heinsohn, J. (1994). Probabilistic description logics. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, UAI-94*, pages 311–326, Seattle, WA. Morgan Kaufmann.

Herbrand, J. (1967). Investigations in proof theory. In van Heijenoort, J., , editor, *From Frege to Gödel: A Source Book in Mathematical Logic, 1879-1931*, pages 525–581. Harvard University Press, Cambridge, Mass.

Hinde, C. (1986). Fuzzy Prolog. *International Journal Man-Machine Studies*, 24:569–595.

Ishizuka, M. and Kanai, N. (1985). Prolog-ELF incorporating fuzzy logic. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence, IJCAI-85*, pages 701–703, Los Angeles, CA. Morgan Kaufmann.

Ivánek, J., Švenda, J. and Ferjenčik, J. (1988). Inference in expert systems based on complete multivalued logic. In *Kybernetica, Proceedings of the Workshop on Uncertainty Processing in Expert Systems*, pages 25–32, Alšovice.

Iványek, J. (1991). Representation of expert knowledge as a fuzzy axiomatic theory. *International Journal of General Systems*, 20:55–58.

Kikuchi, H. and Mukaidono, M. (1988). PROFIL: fuzzy interval logic Prolog. In *Proceedings of the International Workshop of Fuzzy Systems Applications*, pages 205–206, Iizuka, Japan.

Klawonn, F. and Kruse, R. (1994). A Łukasiewicz logic based Prolog. *Mathware and Soft Computing*, 1:5–29.

Klawonn, F. (1995). Prolog extensions to many-valued logics. In Höhle, U. and Klement, E.P., , editor, *Non-Classical Logics and Their Applications to Fuzzy Subsets. A Handbook of the Mathematical Foundations of Fussy Sets Theory*, pages 271–289. Kluwer.

Klement, E. and Navara, M. (1999). Propositional fuzzy logics based on frank t-norms: A comparison. In Dubois, D., Prade, H. and Klement, E., editors, *Fuzzy Sets, Logics and Reasoning about knowledge*, pages 25–47. Kluwer.

Klement, E., Mesiar, R. and Pap, E. (2000). *Triangular Norms*. Kluwer.

Kowalski, R. (1974). Predicate logic as a programming language. *Information Processing*, 74:569–574.

Kowalski, R. (1979a). Algorithm = logic + control. *Communications of the ACM*, 22:424–436.

Kowalski, R. (1979b). *Logic for Problem Solving*. North-Holland.

Kullman, P. and Sandri, S. (1999). Possibilistic logic as an annotated logic. In *Proceedings of the Eighth IEEE International Conference on Fuzzy Systems, Fuzz-IEEE-99*, pages 210–215, Seul, South Korea. IEEE Press.

Lassez, J., Maher, M. and Marriott, K. (1988). Unification revisited. In Minker, J., editor, *Foundations of Deductive Databases and Logic Programming*. Morgan Kaufmann.

Léa Sombé, (Besnard, P., Cordier, M., Dubois, D., Fariñas del Cerro, L., Froidevaux, C., Moinard, Y., Prade, H., Schwind, C. and Siegel, P.), (1990). *Reasoning under Incomplete Information in Artificial Intelligence*. Wiley.

Lee, R. and Chang, C. (1971). Some properties of fuzzy logic. *Information and Control*, 19(5):417–431.

Lee, R. (1972). Fuzzy logic and the resolution principle. *Journal of the ACM*, 19(1):109–119.

LeFaivre, R. (1974a). *Fuzzy Problem-Solving*. PhD thesis, University of Wisconsin.

LeFaivre, R. (1974b). The representation of fuzzy knowledge. *Journal of Cybernetics*, 4(2):57–66.

Lehmke, S. (1995). *On resolution-based theorem proving in propositional fuzzy logic with 'bold' connectives*. Universität Dortmund, Fachbereich Informatik. Master's Thesis.

Lesmo, L., Saitta, L. and Torasso, P. (1983). Fuzzy production rules: A learning methodology. In Wang, P., editor, *Advances in Fuzzy Sets, Possibility Theory, and Applications*, pages 181–198. Plenum Press.

Li, D. and Liu, G. (1990). *A Fuzzy Prolog Database System*. Research Studies Press and John Wiley and Sons.

Liau, C. and Lin, B. (1988). Fuzzy logic with equality. *International Journal Pattern Recognition and Artificial Intelligence*, 2(2):351–365.

Liu, X. and Xiao, H. (1985). Operator fuzzy logic and resolution. In *Proceedings of the Fifteenth International Symposium on Multiple-Valued Logic, ISMVL-85*, pages 68–75, Kingston, Canada. IEEE Press.

Liu, X. (1989). Linear λ-paramodulation in operator fuzzy logic. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, IJCAI-89*, pages 435–440, Detroit, MI. Morgan Kaufmann.

Lu, J., Henschen, L., , Subrahmanian, V. and da Costa, N. (1991). Reasoning in paraconsistent logics. In Boyer, R., editor, *Automated Reasoning: Essays in Honor of Woody Bledsoe*, pages 181–210. Kluwer.

Łukasiewicz, J. (1970). *Selected works*. North-Holland.

Lukasiewicz, T. (1998). Probabilistic logic programming. In *Proceedings of the Thirteenth European Conference on Artificial Intelligence, ECAI-98*, pages 388–392, Brighton, UK. John Wiley and Sons.

Magrez, P. and Smets, P. (1989). Fuzzy modus ponens: A new model suitable for applications in knowledge-based systems. *International Journal of Intelligent Systems*, 4:181–200.

Manyà, F. (1996). *Proof Procedures for Multiple-Valued Propositional Logics*. PhD thesis, Facultat de Ciències, Universitat Autònoma de Barcelona. Published as (Manyà, 1999).

Manyà, F. (1999). *Proof Procedures for Multiple-Valued Propositional Logics*. Number 9 in Monografies de l'Institut d'Investigació en Intel·ligència Artificial. IIIA-CSIC.

Martelli, A. and Montanari, U. (1982). An efficient unification algorithm. *ACM Transactions on Programming Language Systems*, 4(2):258–282.

Martin-Clouaire, R. and Prade, H. (1985). On the problems of representation and propagation of uncertainty in expert systems. *International Journal Man-Machine Studies*, 22:251–264.

Martin, T. and Arcelli, F. (1998). Logic programming and soft computing - An introduction. In Arcelli, F. and Martin, T., editors, *Logic Programming and Soft Computing*, chapter 1, pages 1–18. Research Studies Press.

McNaughton, R. (1951). A theorem about infinite-valued sentencial logic. *Journal of Symbolic Logic*, 16:1–13.

Morgan, C. (1976). A resolution principle for a class of many-valued logics. *Logique et Analyse*, 19(74–75–76):311–339.

Morlán, J. J. (2000). *Implementación de un Compilador de Programación Lógica Posibilística con Constantes Difusas*. EUP-Universitat de Lleida. Master's Thesis. Available at http://alumnes.eup.udl.es/~e1803912/.

Mukaidono, M., Shen, Z. and Ding, L. (1989). Fundamentals of fuzzy Prolog. *International Journal of Approximate Reasoning*, 3:179–193.

Mukaidono, M. and Yasui, H. (1994). Postulates and proposals of fuzzy Prolog. In *Proceedings of the Second European Congress on Intelligent Techniques and Soft Computing, EUFIT-94*, pages 1080–1086, Aachen, Germany.

Mundici, D. (1994). A constructive proof of McNaughton's theorem in infinite-valued logic. *Journal of Symbolic Logic*, 59(2):596–602.

Murray, N. and Rosenthal, E. (1993). Signed formulas: A liftable meta logic for multiple-valued logics. In *Proceedings of the International Symposium on Methodologies for Intelligent Systems, ISMIS-93*, pages 275–284. Springer LNAI 689.

Novák, V. (1990). On the syntactico-semantical completeness of first-order fuzzy logic. Part I: Syntax and semantics. Part II: Main results. *Kibernetika*, 26(1):47–66 and 134–154.

Novák, V. (1992). *The Alternative Mathematical Model of Linguistic Semantics and Pragmatics*. Plenum.

Novák, V. (1995a). A new proof of completeness of fuzzy logic and some conclusions for approximate reasoning. In *Proceedings of the Fourth IEEE International Conference on Fuzzy Systems, Fuzz-IEEE-95*, pages 1461–1468, Yokohama, Japan. IEEE Press.

Novák, V. and Iványek, J. (1995b). The position of fuzzy logic in rule-based expert systems. In *Proceedings of the Fifth International Fuzzy Systems Association World Congress, IFSA-95*, pages 33–35, São Paulo, Brazil.

Novák, V. (1999). Weighted inference systems. In Bezdek, J., Dubois, D. and Prade, H., editors, *Fuzzy Sets in Approximate Reasoning and Information Systems*, Fuzzy Sets Series, pages 191–241. Kluwer.

Novák, V. and Perfilieva, I. (2000). Some consequences of Herbrand and McNaughton theorems in fuzzy logic. In Novák, V. and Perfilieva, I., editors, *Discovering the World with Fuzzy Logic*, Studies in Fuzziness and Soft Computing, pages 271–295. Physica-Verlag.

Orci, I. (1989). Programming in possibilistic logic. *International Journal of Expert Systems*, 2(1):79–94.

Orłowska, E. (1978). The resolution principle for ω^+-valued logic. *Fundamenta Informaticae*, II(1):1–15.

Orlowska, E. and Wierzchon, S. (1985). Mechanical reasoning in fuzzy logics. *Logique et Analyse*, 110-11:193–207.

Pavelka, J. (1979). On fuzzy logic I - Many-valued rules of inference. II - Enriched residuated lattices and semantics of propositional calculi. III - Semantical completeness of some many-valued propositional calculi. *Zeitschr. f. Math. Logik und Grundlagen d. Math.*, 25:45–52, 119–134, 447–464.

Prade, H. (1982). Modal semantics and fuzzy set theory. In Yager, R., editor, *Fuzzy Sets and Possibility Theory - Recent Developments*, pages 232–246. Pergamon Press.

Prawitz, D. (1960). An improved proof procedure. *Theoria*, 26:102–139.

Puyol, J. (1994). *Modularization, Uncertainty, Reflective Control and Deduction by Specialization in MILORD II, a Language for Knowledge-Based Systems.* PhD thesis, Facultat de Ciències, Universitat Autònoma de Barcelona. Published as (Puyol, 1996).

Puyol, J. (1996). *MILORD II: A Language for Knowledge-Based Systems.* Number 1 in Monografies de l'Institut d'Investigació en Intel·ligència Artificial. IIIA-CSIC.

Ray, K. (1990). Bottom-up inferences using fuzzy reasoning. *BUSEFAL*, 42:81–90.

Rios-Filho, L. and Sandri, S. (1995). Contextual fuzzy unification. In *Proceedings of the Fifth International Fuzzy Systems Association World Congress, IFSA-95*, pages 81–84, São Paulo, Brazil.

Robinson, J. A. (1965). A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41.

Rose, A. and Rosser, J. (1958). Fragments of many-valued statement calculi. *Trans. A.M.S.*, 87:1–53.

Sanchez, E. (1989). Importance in knowledge systems. *Information Systems*, 14(6):455–464.

Sandri, S. and Godo, L. (1999). Treatment of temporal information in possibilistic logic with fuzzy constants. In *Proceedings of the Eighth International Fuzzy Systems Association World Congress, IFSA-99*, pages 561–565. Taipei, Taiwan.

Schmitt, P. (1986). Computational aspects of three-valued logic. In Siekmann, J., editor, *Proceedings of the Eighth International Conference on Automated Deduction, CADE-86*, pages 190–198. Springer-Verlag, LNCS 230.

Schwartz, D. (1989). Outline of a naive semantics for reasoning with qualitative linguistic information. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, IJCAI-89*, pages 1068–1073, Detroit, MI. Morgan Kaufmann.

Schweizer, B. and Sklar, A. (1963). Associative functions and abstract semigroups. *Publ. Math.*, 10:69–81.

Shen, Z., Ding, L. and Mukaidono, M. (1988). Fuzzy resolution principle. In *Proceedings of the Eighteenth International Symposium on Multiple-Valued Logic, ISMVL-88*, pages 210–214, Palma de Mallorca, Spain. IEEE Press.

Shortliffe, E. and Buchanan, B. (1975). A model of inexact reasoning in medicine. *Math. Biosci*, 23:351–379.

Sierra, C. (1989). *MILORD: Arquitectura multi-nivell per a sistemes experts en classificació*. PhD thesis, Facultat d'Informàtica de Barcelona, Universitat Politècnica de Catalunya.

Soula, G., Vialettes, B., San marco, J., Thirion, X. and Roux, M. (1986). PROTIS: A fuzzy expert system with medical applications. In Prade, H. and Negoita, C., editors, *Fuzzy Logic in Knowledge Engineering*, volume 86 of *Interdisciplinary Systems Research Series*, pages 295–310. Verlag TÜV Rheinland, Köln.

Stachniak, Z. and O'Hearn, P. (1990). Resolution in the domain of strongly finite logics. *Fundamenta Informaticae*, XIII:333–351.

Stachniak, Z. (1996). *Resolution Proof Systems: an Algebraic Theory*. Kluwer.

Thiele, H. and Lehmke, S. (1994). On 'bold' resolution theory. In *Proceedings of the Third IEEE International Conference on Fuzzy Systems, Fuzz-IEEE-94*, pages 1945–1950, Orlando, Florida. IEEE Press.

Tong, R. and Shapiro, D. (1985). Experimentals investigations of uncertainty in a rule-based system for information retrieval. *International Journal Man-Machine Studies*, 22:265–282.

Tong, R. and Appelbaum, L. (1988). Experiments with interval-valued uncertainty. In Lemmer, J. and Kanal, L., editors, *Uncertainty in Artificial Intelligence 2*, pages 63–75. Noth-Holland.

Tonis, A. and Perfilieva, I. (2000). Functional system of infinite-valued propositional calculus. *Diskretnij analiz i issledovanie operatsii (Discrete Analysis and Operation Research)*, 7(2):75–85.

Umano, M. (1986). A fuzzy production system. In Prade, H. and Negoita, C., editors, *Fuzzy Logic in Knowledge Engineering*, volume 86 of *Interdisciplinary Systems Research Series*, pages 194–208. Verlag TÜV Rheinland, Köln.

Umano, M. (1987). Fuzzy set Prolog. In *Proceedings of the Second International Fuzzy Systems Association World Congress, IFSA-87*, pages 750–753, Tokyo, Japan.

Umano, M. (1989). Implementation of fuzzy production system. In *Proceedings of the Third International Fuzzy Systems Association World Congress, IFSA-89*, pages 450–453, Seattle, WA.

Valverde, L. and Trillas, E. (1985). On modus ponens in fuzzy logic. In *Proceedings of the Fifteenth International Symposium on Multiple-Valued Logic, ISMVL-85*, pages 294–301. IEEE Press.

Vinař, J. and Vojtáš, P. (2000). A formal model for fuzzy knowledge based systems with similarities. *Neural Network World*, 10(5):891–905.

Virtanen, H. (1994). Fuzzy unification. In *Proceedings of the Fifth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU-94*, pages 1147–1152, Paris, France.

Virtanen, H. (1998). Linguistic logic programming. In Arcelli, F. and Martin, T., editors, *Logic Programming and Soft Computing*, chapter 5, pages 91–128. Research Studies Press.

Vojtáš, P. and Paulik, L. (1996). Extensions of logic programming. In Dyckhoff, R., Herre, H. and Schroeder-Heister, P., editors, *Soundness and Completeness of Non-Classical Extended SLD-Resolution*, pages 289–301. Springer-Verlag, LNCS 1050.

Vojtáš, P. (1998). Fuzzy reasoning with tunable t-operators. *Journal for Advanced Computer Intelligence*, 2:121–127.

Vojtáš, P. (2001a). Fuzzy logic programming. *Fuzzy Sets and Systems*, 124(3):361–370.

Vojtáš, P., Alsinet, T. and Godo, L. (2001b). Different models of fuzzy logic programming with fuzzy unification (towards a revision of fuzzy databases). In *Proceedings of the Ninth International Fuzzy Systems Association World Congress, IFSA-2001*, pages 1541–1546, Vancouver, British Columbia, Canada.

Wagner, G. (1998). *Foundations of Knowledge Systems with Applications to Databases and Agents.* Kluwer.

Warren, D. (1983). An abstract Prolog instruction set. Technical Note 309, SRI International, Menlo Park, CA.

Weigert, T., Tsai, J. and Liu, X. (1993). Fuzzy operator logic and fuzzy resolution. *Journal of Automated Reasoning*, 10:59–78.

Yager, R. (1980). An approach to inference in approximate reasoning. *International Journal Man-Machine Studies*, 13:323–338.

Yager, R. (1985). Inference in a multivalued logic system. *International Journal Man-Machine Studies*, 23:27–44.

Yager, R. (1989). Approximate reasoning as a basis for rule-based expert systems. *IEEE Transactions on Systems, Man and Cybernetics*, 14(6):455–464.

Yasui, H., Hamada, Y. and Mukaidono, M. (1995). Fuzzy Prolog based on Łukasiewicz implication and bounded product. In *Proceedings of the Fourth IEEE International Conference on Fuzzy Systems, Fuzz-IEEE-95*, pages 949–954, Yokohama, Japan. IEEE Press.

Yasui, H. and Mukaidono, M. (1996). A consideration on fuzzy logic programming based on Łukasiewicz implication. *Journal of Japan Society for Fuzzy Theory and Systems*, 8(5):937–946.

Yasui, H. and Mukaidono, M. (1998). Fuzzy Prolog based on Łukasiewicz implication. In Arcelli, F. and Martin, T., editors, *Logic Programming and Soft Computing*, chapter 8, pages 147–162. Research Studies Press.

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8:338–353.

Zadeh, L. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1(1):3–28.

Zadeh, L. (1979). A theory of approximate reasoning. In Hayes, J., Michie, D. and Mikulich, L., editors, *Machine Intelligence*, volume 9, pages 149–194. Elsevier.