

Preface to the third volume

Mathematical fuzzy logic (MFL) is a subdiscipline of mathematical logic. Comprising the mathematical study of a certain family of formal logical systems whose algebraic semantics involve some notion of *truth degree*. The main motivations for MFL, its historical origins, and its later evolution were briefly explained in the preface of the first volume of this handbook series.

Fuzzy logics can be seen as a particular kind of many-valued logical system where the intended semantics is typically based on algebras of linearly ordered truth values. As a subdiscipline of mathematical logic, MFL has acquired the typical core agenda of this field and is studied by many mathematically-minded researchers regardless of its original motivations, as witnessed by the plethora of works on a diversity of topics that have been accumulating over the years.

This handbook series aims to be an up-to-date, systematic presentation of the best developed areas of MFL. The first two volumes were released in 2011, also in *Studies in Logic, Mathematical Logic and Foundations* (vols. 37 and 38), College Publications. The first volume starts with a gentle introduction to MFL assuming only some basic knowledge of classical logic. The second chapter presents and develops a general and uniform framework for MFL based on the notions and methods of abstract algebraic logic. The third chapter is a presentation of the deeply developed proof theory of fuzzy logics. The fourth chapter presents the standard algebraic semantics for fuzzy logics based on classes of semilinear residuated lattices. The fifth chapter closes the first volume with the study of Hájek's BL logic and its algebraic counterpart.

The second volume of the series starts with the sixth chapter and is devoted to another widely studied fuzzy logic, Łukasiewicz logic L , and MV-algebras as its algebraic semantics. The seventh chapter deals with a third distinguished fuzzy logic, Gödel–Dummett logic, and its variants. The eighth chapter studies fuzzy logics in expanded languages providing greater expressive power. The ninth chapter collects results on functional representations of fuzzy logics and their free algebras. The last two chapters are dedicated to complexity issues: Chapter X studies the computational complexity of propositional fuzzy logics, while Chapter XI is devoted to the arithmetical hierarchy of first-order predicate fuzzy logics.

Intense research in MFL has continued in recent years, bringing to a higher level of maturity some of the topics that were omitted from the previous volumes of the handbook. This third volume collects chapters on seven such areas. It starts with Chapter XII, a systematic study of the most prominent part of the algebraic semantics for fuzzy logics: integral residuated chains. The thirteenth chapter presents a different type of semantics for some prominent fuzzy logics obtained by extending Hintikka's and Giles' semantic games. In a similar fashion, the fourteenth chapter is devoted to another game-theoretic interpretation of t-norm-based fuzzy logics, namely, that given by Ulam–Rényi games arising from the theory of error correcting codes. The fifteenth chapter surveys a series of works on fuzzy logics with evaluated syntax in which intermediate truth-values are incorporated as syntactical devices that accompany each formula as a lower bound for its truth-degree. Chapter XVI surveys another area that has recently been intensively developed: fuzzy description logics, understood as tractable fragments of first-order

fuzzy logics amenable to knowledge representation. The seventeenth chapter focuses on another application of MFL by presenting the theory of MV-algebras endowed with functions, called *states*, that allow one to model finitely additive probability measures. The volume is concluded by Chapter XVIII offering a philosophical discussion of the role of MFL in the study of vagueness.

It should be emphasized that this is not a handbook written by a single team of authors, but a collection of chapters prepared by distinguished experts in each area. Nevertheless, the editors have encouraged a reasonable level of homogeneity between the chapters, as regards their structure and notation. The series is conceived as a unit with consecutive page and chapters enumeration. The majority of chapters in this volume contain a purely theoretical (mathematical) presentation, making it mostly a book on mathematical logic, focusing on the study of a particular family of many-valued non-classical logics. However, as briefly outlined before, some chapters feature connections to other mathematical fields, to applications in computer science and, in the case of the last chapter, to developments in analytical philosophy. The intended audience of the book is quite wide, comprising at least the following groups of readers: (1) students of logic looking for a systematic presentation of MFL where they can study the discipline from scratch, (2) experts on MFL that may use it as a reference book for consultation, (3) readers interested in fuzzy set theory and its applications looking for the logical foundations of (some parts) of the area, and (4) readers interested in philosophical and linguistic issues related to reasoning in presence of vagueness, looking for a mathematical apparatus that can be applied to some aspects of those issues.

Acknowledgements The editors wish to express their gratitude to

- all authors for their great effort in writing the chapters,
- the readers assigned to chapters for their invaluable help in improving them (Jesse Alama, Brunella Gerla, Giangiacomo Gerla, Siegfried Gottwald, Gejza Jenca, Ondrej Majer, Vincenzo Marra, George Metcalfe, Franco Montagna, Daniele Mundici, Ulrike Sattler, Stewart Shapiro, and Esko Turunen),
- Petra Ivaničová, Christoph Roschger, and Jane Spurr for technical and administrative support, and
- the joint project of Austrian Science Fund (FWF) I1897-N25 and Czech Science Foundation (GAČR) 15-34650L for supporting the preparation of the book.

Dedication This volume is dedicated to two great researchers who passed away during the preparation of the book: Siegfried Gottwald and Franco Montagna. Siegfried was one of the pioneers in the area of MFL, which he deeply influenced with his monograph *A Treatise on Many-Valued Logics*, and remained a constant contributor to the field and a close collaborator of this handbook series. Franco joined the MFL community later, but he soon became one of the field's major figures, as witnessed by the fact that he is the only author who contributed to each of the three volumes of this series.

Petr Cintula, Christian G. Fermüller, and Carles Noguera
Editors

Contents of Volume 1

I	Introduction to Mathematical Fuzzy Logic Libor Běhounek, Petr Cintula, and Petr Hájek	1
II	A General Framework for Mathematical Fuzzy Logic Petr Cintula and Carles Noguera	103
III	Proof Theory for Mathematical Fuzzy Logic George Metcalfe	209
IV	Algebraic Semantics: Semilinear FL-Algebras Rostislav Horčík	283
V	Hájek’s Logic BL and BL-Algebras Manuela Busaniche and Franco Montagna	355

Contents of Volume 2

VI	Łukasiewicz Logic and MV-Algebras Antonio Di Nola and Ioana Leuştean	469
VII	Gödel–Dummett Logics Matthias Baaz and Norbert Preining	585
VIII	Fuzzy Logics with Enriched Language Francesc Esteva, Lluís Godo, and Enrico Marchioni	627
IX	Free Algebras and Functional Representation for Fuzzy Logics Stefano Aguzzoli, Simone Bova, and Brunella Gerla	713
X	Computational Complexity of Propositional Fuzzy Logics Zuzana Haniková	793
XI	Arithmetical Complexity of First-Order Fuzzy Logics Petr Hájek, Franco Montagna, and Carles Noguera	853

Volume 3

LIST OF AUTHORS

FERNANDO BOBILLO
Department of Computer Science and Systems Engineering
University of Zaragoza
C. Maria de Luna 1
50018 Zaragoza, Spain
Email: fbobillo@unizar.es

MARCO CERAMI
Department of Computer Science
Palacký University Olomouc
17. listopadu 2
771 46 Olomouc, Czech Republic
Email: marco.cerami@upol.cz

FERDINANDO CICALÉSE
Department of Computer Science
University of Verona
Strada Le Grazie 15
37134 Verona, Italy
Email: ferdinando.cicalese@univr.it

FRANCESC ESTEVA
Artificial Intelligence Research Institute (III A)
Spanish National Research Council (CSIC)
Campus de la Universitat Autònoma de Barcelona s/n
08193 Bellaterra, Catalonia, Spain
Email: esteva@iia.csic.es

CHRISTIAN G. FERMÜLLER
Theory and Logic Group 185.2
Vienna University of Technology
Favoritenstraße 9-11
1040 Vienna, Austria
Email: chrisf@logic.at

x

TOMMASO FLAMINIO
Dipartimento di Scienze Teoriche e Applicate
Università dell'Insubria
Via Mazzini 5
21100 Varese, Italy
Email: tommaso.flaminio@uninsubria.it

ÀNGEL GARCÍA-CERDAÑA
Artificial Intelligence Research Institute (IIIA)
Spanish National Research Council (CSIC)
Campus de la Universitat Autònoma de Barcelona s/n
08193 Bellaterra, Catalonia, Spain

and

Information and Communication Technologies Department
Universitat Pompeu Fabra
Tànger 122-140
08018 Barcelona, Catalonia, Spain
Email: angel@iia.csic.es

TOMÁŠ KROUPA
Institute of Information Theory and Automation
Czech Academy of Sciences
Pod Vodárenskou věží 4
182 08 Prague, Czech Republic
Email: kroupa@utia.cas.cz

FRANCO MONTAGNA
Department of Information Engineering and Mathematics
University of Siena
Via Roma 56
53100 Siena, Italy

VILÉM NOVÁK
Institute for Research and Applications of Fuzzy Modeling
University of Ostrava & NSC IT4Innovations
30. dubna 22
701 03 Ostrava 1, Czech Republic
Email: Vilem.Novak@osu.cz

RAFAEL PEÑALOZA
KRDB Research Centre
Free University of Bozen-Bolzano
Piazza Domenicani 3
I-39100 Bozen-Bolzano BZ, Italy
Email: rafael.penaloza@unibz.it

NICHOLAS J.J. SMITH
Department of Philosophy
Main Quadrangle A14
University of Sydney
NSW 2006, Australia
Email: njjsmith@sydney.edu.au

UMBERTO STRACCIA
Istituto di Scienze e Tecnologie dell'Informazione (ISTI)
Consiglio Nazionale delle Ricerche (CNR)
Via G. Moruzzi 1
56124 Pisa, Italy
Email: straccia@isti.cnr.it

THOMAS VETTERLEIN
Department of Knowledge-Based Mathematical Systems
Johannes Kepler University Linz
Altenberger Straße 69
4040 Linz, Austria
Email: Thomas.Vetterlein@jku.at

CONTENTS

Preface to Volume 3	v
Contents of Volumes 1 and 2	vii

Volume 3

The list of authors of Volume 3	ix
---------------------------------	----

XII Algebraic Semantics: The Structure of Residuated Chains

Thomas Vetterlein	929
1 Introduction	929
2 Residuated totally ordered monoids	930
3 The partial-algebra method for the representation of totally ordered monoids	932
3.1 The idea	932
3.2 D.p.r. tomonoids and R-chains	933
3.3 Ordinal sum decomposition of R-chains	936
3.4 Naturally ordered R-chains	937
3.5 The representation of d.p.r. tomonoids	941
4 Coextensions of totally ordered monoids	942
4.1 The idea	942
4.2 Q.n.c. tomonoids and their quotients	943
4.3 The chain of quotients of a q.n.c. tomonoid	946
4.4 The Cayley tomonoid	948
4.5 Real Archimedean coextensions	955
5 Historical remarks and further reading	964
5.1 Partial algebras	964
5.2 Coextensions of tomonoids	965

XIII Semantic Games for Fuzzy Logics

Christian G. Fermüller

969

1	Introduction	969
2	Hintikka's game — from classical to many-valued logics	970
2.1	Hintikka's classical semantic game	970
2.2	Hintikka's game in a fuzzy logic setting	971
2.3	Limits of Hintikka-style games	973
3	An explicit evaluation game for Łukasiewicz logic	975
4	Giles's game for Łukasiewicz logic	978
5	Generalizing Giles's game	984
5.1	General payoff principles	984
5.2	Dialogue principles for logical connectives	986
5.3	Extracting truth functions	989
5.4	Revisiting the game for Łukasiewicz logic	992
5.5	Finitely-valued Łukasiewicz logics	993
5.6	Cancellative hoop logic	993
5.7	Abelian logic	994
5.8	Alternative aggregation functions	994
6	Giles's game and hypersequents — the case of Abelian logic	995
6.1	Abelian logic revisited	996
6.2	A Giles-style game for Abelian logic	996
6.3	Disjunctive states	998
6.4	A hypersequent system for Abelian logic	999
7	Backtracking games	1001
7.1	A backtracking game for Łukasiewicz logic	1001
7.2	A backtracking game for Gödel logic	1003
7.3	An implicit backtracking game for Gödel logic	1005
7.4	An implicit backtracking game for Product logic	1006
8	Propositional random choice games	1007
9	Random choice rules for semi-fuzzy quantifiers	1011
9.1	A note on binary quantifiers	1013
9.2	Blind choice quantifiers	1014
9.3	Deliberate choice quantifiers	1020
10	A brief synopsis of semantic games	1023
11	Historical remarks and further reading	1024

XIV Ulam–Rényi Game Based Semantics for Fuzzy Logics

Ferdinando Cicalese and Franco Montagna

1029

1	Introduction	1029
2	Preliminaries	1030

3	Ulam–Rényi game	1033
3.1	Ulam–Rényi game without lies	1033
3.2	Ulam–Rényi game with a fixed maximum number of lies	1036
3.3	Ulam–Rényi games and many-valued logics	1036
4	The Combinatorics of Ulam–Rényi games	1039
4.1	The fundamental bound on the strategy size	1039
4.2	The basic case — a two batch “optimal” strategy	1040
4.3	Half lies and asymmetric channels	1043
4.4	The combinatorics of the half-lie Ulam–Rényi game	1043
5	Multichannel Ulam–Rényi games and Hájek Basic Logic: preliminaries	1045
6	A game semantics for Gödel logic	1048
6.1	Gödel games	1048
6.2	Some strategies	1049
6.3	Truth functions in Gödel games and completeness	1050
7	A game semantics for product logic	1051
7.1	The Product game	1051
7.2	Some strategies	1053
7.3	Truth functions in Product game and completeness	1054
8	Game semantics for BL and for SBL	1055
8.1	BL games	1055
8.2	Truth functions and completeness for BL-games	1056
8.3	Completeness theorem	1058
9	Historical remarks and further reading	1058

XV Fuzzy Logic with Evaluated Syntax

	Vilém Novák	1063
1	Introduction	1063
2	Language and truth values of $Ev_{\mathcal{L}}$	1064
2.1	The algebra of truth values	1064
2.2	Language and formulas	1066
3	Traditional and evaluated syntax	1067
3.1	Classical syntactic consequence	1068
3.2	Evaluated syntax and graded syntactic consequence	1069
3.3	Semantics	1072
3.4	Formal fuzzy theories	1073
3.5	Soundness and completeness in evaluated syntax	1075
3.6	Restrictions when choosing the algebra of truth values	1075
4	Syntax of $Ev_{\mathcal{L}}$	1078
4.1	Logical axioms	1078
4.2	Inference rules and validity	1079

4.3	Special provable tautologies and properties of the provability degree . . .	1080
4.4	Consistency of fuzzy theories	1081
4.5	Extension of fuzzy theories and deduction theorem	1083
4.6	Complete fuzzy theories	1087
4.7	Henkin fuzzy theories	1091
5	Completeness theorem	1092
6	Model theory	1094
7	Some additional properties	1097
7.1	Enriching the structure of truth values	1097
7.2	Sorites fuzzy theories	1098
7.3	Ev_L and Łukasiewicz logic	1099
8	Historical remarks and further reading	1100

XVI Fuzzy Description Logics

Fernando Bobillo, Marco Cerami, Francesc Esteva,

Àngel García-Cerdaña, Rafael Peñaloza, Umberto Straccia 1105

1	Introduction	1105
2	Algebras of truth values defined by triangular norms	1107
2.1	Triangular norms and conorms	1107
2.2	Divisible finite t-norms and t-conorms	1110
2.3	Expanding the standard chain with an involutive negation	1111
2.4	Expanding the standard chain with other operators	1112
3	Introduction to Fuzzy Description Logics	1112
3.1	Basic concept constructors in FDLs	1114
3.2	Semantics for basic concept constructors	1115
3.3	The hierarchy of fuzzy attributive languages	1116
3.4	Knowledge bases	1121
3.5	Reasoning tasks	1122
4	Fuzzy Description Logics from the point of view of MFL	1123
4.1	A logical framework for FDLs: The propositional logics	1123
4.2	A logical framework for FDLs: The predicate logics	1124
4.3	Basic fuzzy description logics as fragments of fuzzy first-order logics	1126
5	Fuzzy Description Logic languages with higher expressivity	1128
5.1	Fuzzy concept constructors	1128
5.2	Fuzzy role constructors	1131
5.3	Fuzzy datatypes	1132
5.4	New axioms	1133
5.5	Some important languages	1135

6 Reasoning algorithms	1135
6.1 Equivalences between reasoning tasks in finitely-valued FDLs	1135
6.2 Types of algorithms	1137
7 Decidability and complexity issues	1152
7.1 Undecidability of consistency in \mathfrak{JALCE}	1153
7.2 Further undecidability results	1159
7.3 Complexity bounds	1161
7.4 Reasoning with finite chains	1163
8 Some historical remarks and further reading	1164
8.1 Classical Description Logics in short	1164
8.2 Fuzzy Description Logics	1165
8.3 Computational issues in a historical perspective	1169

XVII States of MV-algebras

Tommaso Flaminio and Tomáš Kroupa	1183
1 Introduction	1183
2 Basic notions	1184
2.1 MV-algebras	1184
2.2 Łukasiewicz logic	1187
2.3 Probability measures	1188
2.4 Compact convex sets	1189
3 States	1191
3.1 Basic properties	1192
3.2 Examples of states	1195
3.3 States of finitely presented MV-algebras	1196
3.4 States of ℓ -groups	1200
4 Integral representation	1202
4.1 Characterization of state space	1205
4.2 Existence of invariant states and faithful states	1206
5 De Finetti's coherence criterion for many-valued events	1208
5.1 Betting on formulas of Łukasiewicz logic	1212
5.2 Complexity	1216
6 MV-algebras with internal states	1218
6.1 Subdirectly irreducible SMV-algebras	1220
6.2 States of MV-algebras and internal states	1221
6.3 Dealing with coherent books in SMV-algebraic theory	1224
7 Conditional probability and Dutch Book argument	1225
7.1 Bookmaking on many-valued conditional events	1227
8 Historical remarks and further reading	1232

XVIII Fuzzy Logics in Theories of Vagueness

Nicholas J.J. Smith

1237

1 Introduction	1237
1.1 Characteristics of vague predicates	1237
1.2 Other phenomena	1239
2 Non-fuzzy theories of vagueness	1240
2.1 Theory of vagueness	1240
2.2 Epistemicism	1241
2.3 Gaps and third values	1244
2.4 Plurivaluationism	1247
2.5 Contextualism	1249
3 Fuzzy theories of vagueness	1250
3.1 Degrees of truth	1250
3.2 Truth rules	1253
3.3 The intended model	1256
4 Arguments for	1256
4.1 The nature of vagueness	1256
4.2 Solving the sorites	1261
5 Arguments against	1266
5.1 Artificial precision	1267
5.2 Truth functionality	1272
6 Historical remarks and further reading	1276

INDEX

1283

Chapter XVI: Fuzzy Description Logics

FERNANDO BOBILLO, MARCO CERAMI, FRANCESC ESTEVA,
ÀNGEL GARCÍA-CERDAÑA, RAFAEL PEÑALOZA,
UMBERTO STRACCIA

1 Introduction

Description Logics (DLs) are a family of well established knowledge representation formalisms (see [4] for a complete overview on the subject). The languages for DLs are based on *concepts*, *roles* and *individuals*, that are, according to a well-defined Tarski style semantics, interpreted as sets (or unary predicates), binary relations (or binary predicates), and domain elements, respectively. From a suitable set of atomic concepts and roles, it is possible to build up complex concepts by means of a limited family of *concept constructors*. The different languages of DLs are identified by sets of concept constructors and they are more or less expressive, depending on the concept constructors that are either explicitly present in the language or implicitly definable from other constructors.

From concepts and roles it is possible to store structured knowledge under the form of *knowledge bases* that contain the relevant terminological knowledge about particular knowledge domains. Knowledge bases contain structured knowledge that gives an account of functional dependencies between complex concepts, or relations between individuals and either concepts or roles. In this sense, a knowledge base is usually compounded by a *terminological box*, or TBox, that contains inclusions between complex concepts, an *assertional box*, or ABox, that contains assertions of concepts or roles with respect to individuals or ordered couples of individuals respectively and, sometimes, by a *relational box*, or RBox, that contains inclusions between roles.

Since DLs are mainly oriented to applications, they need to exhibit good computational behavior. The reasoning services provided have to be decidable and, as much as possible, computable in a reasonable time. For this reason, historically the research on DLs has been characterized by the search of a fair trade-off between the expressivity of the languages considered and the computational complexity of the reasoning services.

There are two possible ways to present DL languages and their semantics. One way is to present them directly as representation languages as it is usually done. One example of this kind of presentation can be found in the introductory chapter [8] of the Description Logic Handbook [4]. Another way is to present basic DLs from the logical point of view as fragments of first order classical logic [111].

Fuzzy Description Logics (FDLs) were born in the nineties as a generalization of DLs to the fuzzy framework. The first generalizations consisted in having the same

structure as DLs, but interpreting concepts and roles as fuzzy sets and fuzzy relations respectively (see the section on historical remarks at the end of this chapter). Nevertheless these first approaches had no clear logical counterpart. At the end of nineties, in the framework of Fuzzy Logic, Hájek [78] proposed many-valued residuated logical systems (studied in this handbook) as the kernel of Fuzzy Logic in narrow sense. This has been the starting point of that branch of Fuzzy Logic known as *Mathematical Fuzzy Logic* (MFL). The important development of this subject, of which the results provided in this handbook are a clear account, has established different consolidated logical systems. Within the framework of MFL, Hájek proposes in [80] (see also [82]) to define FDLs from the logical point of view in an analogous way as DLs could be defined from classical first order logic.

This chapter is structured as follows: In Section 2 we introduce the algebras of truth values defined by continuous t-norms (or divisible finite t-norms), their residua and the standard involutive negation. The semantics of our FDLs are defined as interpretations over these algebras. This is a technical section that could be omitted in a first reading. The reader can go back to it when she needs some notion or result concerning the algebras of truth values. In Section 3 we define FDLs taking as constructors the generalization of the classical ones. The semantics is defined by means of interpretations in the algebraic ordered structure defined on the real interval $[0, 1]$ by a continuous t-norm and its residuum, or in the one defined on a finite chain by a divisible finite t-norm and its residuum. We study the hierarchy of basic FDL languages, which is more complex than in the classical case due to the fact that some definabilities between certain constructors are not valid in the fuzzy framework. The axioms in knowledge bases in the fuzzy setting are graded. Reasoning tasks are also graded. We will be interested in stating satisfiability of a concept or subsumption between concepts in a certain degree. Section 4 is devoted to highlight the relationship among FDLs defined in the previous section and fuzzy predicate logics studied in the framework of MFL. First, we define logical systems underlying FDL languages from a semantical point of view as logics arising from particular algebras of truth values. Then we show that FDL languages defined in the previous sections can be seen as fragments of these first-order fuzzy logics, according to Hájek's proposal in [80]. The chapter follows with a section devoted to Fuzzy description languages with higher expressivity and a section dealing with different algorithms to decide the reasoning tasks. Section 7 is devoted to (un)decidability and complexity results. Finally, the last section contains the historical remarks that summarize the reported results from a historical point of view.

Let us remark that in this chapter we will pay special attention to the case of finite-valued FDLs, which seems to be the most interesting in practice since the basic reasoning tasks for quite expressive languages are decidable and the computational complexity of some of them is the same as the corresponding classical DLs.

Finally, let us stress that there are different issues that are treated in the fuzzy setting that are still not well modeled in the logical setting of MFL. Among the most important there are *fuzzy quantifiers*. Expressions like “the majority of ...”, “a few ...”, etc, that are crucial in the fuzzy approach are not yet considered in FDLs based in MFL. For an excellent report about the state of the art about fuzzy quantifiers in fuzzy logic, see [66], while for fuzzy quantification in FDLs see [109, 110]. Another topic not considered in

this chapter is the treatment of uncertainty in FDLs (see [92] and the references therein for a report about the work done in this topic). Nevertheless, the framework of MFL still offers formal tools that can be used to try to give a mathematical logic treatment to this kind of notions. Undoubtedly, the framework of FDLs, above all in the MFL setting, can be a powerful source of motivation for undertaking this kind of research, as it was predicted by Hájek in the not so known and frequently visited second part of his seminal monograph.

2 Algebras of truth values defined by triangular norms

The fuzzy description logics we deal with in this chapter are based on interpretations on certain linearly ordered algebras of truth values called *canonical chains*. In the infinite case these algebras are those defined on the real interval $[0, 1]$ by a continuous t-norm and its residuum, and by the involutive negation $N(x) = 1 - x$. In the finitely-valued case the canonical chains are the ones defined on the finite chain $\{0, \frac{1}{n-1}, \dots, \frac{n-2}{n-1}, 1\}$ by a divisible finite t-norm, its residuum and the involutive negation N . In this preliminary section we recall the notions of continuous t-norm and divisible finite t-norm, and some notions and results about these canonical chains.

2.1 Triangular norms and conorms

For the notions and results presented in this section, a full reference is the book [91]. A *triangular norm* (or *t-norm*) is a binary operation defined on the real interval $[0, 1]$ which is associative, commutative, non-decreasing in both arguments,¹ and has 1 as unit element. If the t-norm $*$ is continuous, we can define the operation \Rightarrow_* satisfying, for all $x, y, z \in [0, 1]$, the condition

$$x * z \leq y \Leftrightarrow z \leq x \Rightarrow_* y. \quad (1)$$

This operation is called the *residuum* of the t-norm and it can be defined by:

$$x \Rightarrow_* y := \max\{z \mid x * z \leq y\}. \quad (2)$$

PROPOSITION 2.1.1. *The following properties hold for any continuous t-norm $*$ and its residuum \Rightarrow_* :*

- a) $x \Rightarrow_* y = 1$ if and only if $x \leq y$
- b) $1 \Rightarrow_* x = x$
- c) $\min\{x, y\} = x * (x \Rightarrow_* y)$ (divisibility)
- d) $\max\{x, y\} = \min\{(x \Rightarrow_* y) \Rightarrow_* y, (y \Rightarrow_* x) \Rightarrow_* x\}$.

The reason for which property c) in the previous proposition is called *divisibility condition* is the fact that this is equivalent to the following one: for every $x, y \in [0, 1]$ with $x > y$, there exists $z \in [0, 1]$ such that $x * z = y$. Observe that this condition is

¹A function $f: [0, 1]^2 \rightarrow [0, 1]$ is *non-decreasing* in both arguments if for each $x, y, z \in [0, 1]$, $x \leq y$ implies $f(x, z) \leq f(y, z)$ and $f(z, x) \leq f(z, y)$.

*	Minimum (Gödel)	Product	Łukasiewicz
$x * y$	$\min\{x, y\}$	$x \cdot y$	$\max\{0, x + y - 1\}$
$x \Rightarrow_* y$	$\begin{cases} 1 & \text{if } x \leq y \\ y & \text{otherwise} \end{cases}$	$\begin{cases} 1 & \text{if } x \leq y \\ y/x & \text{otherwise} \end{cases}$	$\min\{1, 1 - x + y\}$
\neg_*	$\begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$	$\begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$	$1 - x$

Table 1. The three main continuous t-norms

equivalent to the continuity of the functions defined by $f_y(x) = x * y$ for every $y \in [0, 1]$ and also to the continuity of $*$ due to the fact that $*$ is commutative and non-decreasing in both arguments.

A *negation function* on $[0, 1]$ is a unary operation $n: [0, 1] \rightarrow [0, 1]$ satisfying the following properties:

- $n(0) = 1, n(1) = 0$, and
- for all $x, y \in [0, 1], x \leq y \Rightarrow n(y) \leq n(x)$. (antimonotonicity)

We say that n is *weak* if, for all $x \in [0, 1], x \leq n(n(x))$; and it is said to be *strong* or *involution* if, for all $x \in [0, 1], n(n(x)) = x$. The main example of involutive negation is the so called *standard negation function* defined as $N(x) = 1 - x$. Each continuous t-norm $*$ has an associated negation function defined as $\neg_* x = x \Rightarrow_* 0$, which is always a weak negation function and it is involutive if and only if $*$ is the (finite or infinite) Łukasiewicz t-norm.

Table 1 shows the main continuous t-norms (Minimum, Product and Łukasiewicz) with their residua and the corresponding associated negations. The three continuous t-norms in Table 1 are the basic ones since any continuous t-norm can be expressed as an *ordinal sum* of copies of them [99]. Indeed, there are the following types of continuous t-norms that we will use in the chapter (see for instance [69]):

1. *Idempotent* t-norms: A t-norm is idempotent if for all $x \in [0, 1], x * x = x$.
2. *Strict*² t-norms: A t-norm is *strict* if for every $x, y \in [0, 1]$ such that $x * y = 0$, then $x = 0$ or $y = 0$. This is equivalent to the following two conditions:
 - (a) The associated negation is *Gödel negation*, i.e., $\neg_* 0 = 1$ and $\neg_* x = 0$, otherwise, and
 - (b) for every $x \in [0, 1], \min\{x, \neg_* x\} = 0$.
3. *Nilpotent* t-norms: A t-norm is nilpotent if for each $x \in (0, 1)$ there exists a natural n such that $x^n = 0$. This is equivalent to the following conditions:
 - (a) It is isomorphic to the Łukasiewicz t-norm and
 - (b) the associated negation is involutive.

² Let us remark that the name *strict* is used in a different meaning than in [91], where *strict* means *strictly monotone*.

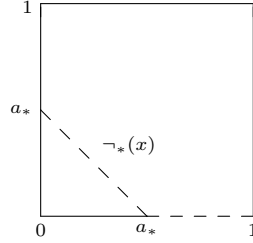


Figure 1. Graph of a residuated negation

The only idempotent continuous t-norm is the minimum and the only nilpotent continuous t-norm up to isomorphisms is Łukasiewicz. Strict continuous t-norms form a very large set containing minimum and product and any ordinal sum that has either no first component or whose first component is not Łukasiewicz t-norm. Finally, we have the set of non-strict continuous t-norms that contains Łukasiewicz t-norm plus any ordinal sum having Łukasiewicz as first component. Therefore, we can associate to each continuous t-norm $*$ an element, denoted a_* , defined as:

- 0 if $*$ is strict or
- the upper bound of the first component of the ordinal sum of $*$ if $*$ is non-strict.

Notice that $a_* = 1$ if and only if $*$ is Łukasiewicz. Let us remark that strict, nilpotent and non-strict continuous t-norms are characterized by their associated negation:

- The t-norm $*$ is strict if and only if \neg_* is Gödel negation.
- The t-norm $*$ is nilpotent if and only if \neg_* is involutive.
- The t-norm $*$ is non-strict if and only if \neg_* is defined as $\neg_*(0) = 1$, $\neg_*(x) = a_* - x$ if $x \in (0, a_*]$ and $\neg_*(x) = 0$ otherwise.

Summarising, all residuated negation functions are definable in the form (see Figure 1):

$$\neg_*(x) = \begin{cases} 1 & \text{if } x = 0 \\ a_* - x & \text{if } x \in (0, a_*] \\ 0 & \text{otherwise.} \end{cases}$$

Observe that if $a_* = 0$, we obtain Gödel negation, and if $a_* = 1$, then $\neg_* = N$. Observe also that if $*$ is strict, then the double negation is given by $\neg_*\neg_*0 = 0$ and $\neg_*\neg_*x = 1$ for $x \neq 0$. Thus $\neg_*\neg_*$ can be seen as a crispification operator that preserves falsity.

The following result is used in Section 3.3 to state the relationship between both quantifier restrictions in our FDL languages.

PROPOSITION 2.1.2. *Let $X \subseteq [0, 1]$. If $*$ is a continuous t-norm, then*

- a) $\neg_* \sup X = \inf\{\neg_*x \mid x \in X\}$.
- b) $\neg_* \inf X \geq \sup\{\neg_*x \mid x \in X\}$; the equality holds only if $*$ is the Łukasiewicz t-norm.

\oplus	Maximum	Probabilistic sum	Łukasiewicz
$x \oplus y$	$\max\{x, y\}$	$(x + y) - (x \cdot y)$	$\min\{1, x + y\}$

Table 2. The dual t-conorms corresponding to the three main continuous t-norms

Proof. By definition, \neg_* is a decreasing continuous function over the interval $(0, 1]$. Therefore, if $\sup X \neq 0$, then a) is obviously true. Analogously, if $\inf X \neq 0$, then $\neg_* \inf X = \sup\{\neg_* x \mid x \in X\}$. Moreover, if $\sup X = 0$, then X can only contain the element 0 and thus a) holds. Consider now the case where $\inf X = 0$. If $0 \in X$, then we have that $\neg_* \inf X = \sup\{\neg_* x \mid x \in X\}$, but if $0 \notin X$, then $\neg_* \inf X = 1$ and $\sup\{\neg_* x \mid x \in X\} = a_*$. In both cases, $\neg_* \inf X \geq \sup\{\neg_* x \mid x \in X\}$, in particular when $a = 1$, which is the Łukasiewicz case. \square

The *standard chain relative to the continuous t-norm $*$* is the structure

$$[0, 1]_* = \langle [0, 1], \max, \min, *, \Rightarrow_*, 0, 1 \rangle.$$

A *triangular co-norm* (or t-conorm) is a binary operation defined on $[0, 1]$ that is associative, commutative, non decreasing in both arguments, and having 0 as neutral element. We say that a t-norm $*$ and a t-conorm \oplus are *dual* with respect to a negation n if, for every $x, y \in [0, 1]$, the De Morgan laws hold:

$$n(x * y) = n(x) \oplus n(y) \quad n(x \oplus y) = n(x) * n(y).$$

In Table 2 we show the t-conorms dual (with respect to the standard involutive negation function $N(x) = 1 - x$) to the three main continuous t-norms.

2.2 Divisible finite t-norms and t-conorms

The notion of t-norm can be extended to bounded finite chains with 0 and 1 as first and last element respectively (see [94, 95]). We will call this kind of operation *finite t-norm*. They fulfill the same properties as t-norms and since they are of finite range all of them have a residuum.

A finite t-norm satisfying the divisibility condition is called a *divisible finite t-norm*. Notice that such a t-norm also satisfies the properties stated for continuous t-norms in Proposition 2.1.1. In [94] it is shown that any divisible finite t-norm is either a finite Łukasiewicz t-norm or a finite minimum t-norm or a finite ordinal sum of copies of both kinds of such finite t-norms. In analogy with the case of t-norms we will consider the following types of divisible finite t-norms:

1. The only divisible finite *idempotent* t-norms are the finite minimum t-norms.
2. The divisible finite *strict* t-norms are the ordinal sums beginning with a finite minimum component.
3. The only divisible finite *nilpotent* t-norms are the Łukasiewicz finite t-norms.
4. The divisible finite *non strict* t-norms are the ordinal sums beginning with a finite Łukasiewicz component.

PROPOSITION 2.2.1. *Let $X \subseteq C_n$. If $*$ is a divisible finite t-norm, then*

$$\neg_* \sup X = \inf \{ \neg_* x \mid x \in X \} \quad \neg_* \inf X = \sup \{ \neg_* x \mid x \in X \}.$$

Proof. It is an easy consequence of Proposition 2.1.2 taking into account that X is finite and thus infima and suprema are in fact minima and maxima. \square

Since all chains with n elements are order-isomorphic, without loss of generality we can take as finite chain of n elements the set:

$$C_n = \left\{ 0, \frac{1}{n-1}, \frac{2}{n-1}, \dots, \frac{n-2}{n-1}, 1 \right\}.$$

Given a divisible finite t-norm $*$ over C_n , the structure

$$C_* = \langle C_n, \max, \min, *, \Rightarrow_*, 0, 1 \rangle$$

is called the *standard finite chain* relative to $*$. The standard chains of n elements corresponding to the finite t-norms of Łukasiewicz and Minimum are restrictions to C_n of the standard chains respectively defined on $[0, 1]$ by Łukasiewicz and minimum t-norms and their residua.

The notion of triangular co-norm can also be extended to finite chains. A t-conorm \oplus defined on a finite chain C is *divisible* if for every $a, b \in C$ such that $b \geq a$, there exists $c \in C$ such that $b = a \oplus c$. We can define the notion of *duality* for finite t-norms and finite t-conorms in an analogous way to the case of t-norms and t-conorm.

2.3 Expanding the standard chain with an involutive negation

In this chapter we will deal with different fuzzy description languages provided with a complementation. In order to interpret complementation we need an involutive negation in our standard algebra of truth values. Consequently, when the negation \neg_* is not involutive, we enrich the standard chain $[0, 1]_*$ or C_* with the standard involutive negation $N(x) = 1 - x$. The structure obtained by enriching a standard chain either $[0, 1]_*$ or C_* with N will be called *canonical algebra relative to $*$* and denoted by T_* . We denote by T the carrier of T_* ; thus T means $[0, 1]$ or C_n .

Having the standard involutive negation N allows us to define in T_* :

- a) The dual t-conorm $x \oplus y := N(N(x) * N(y))$.
- b) The operator known in the literature as *Monteiro–Baaz operator*, which is denoted by δ and defined on any chain as $\delta(1) = 1$ and $\delta(x) = 0$ if $x \neq 1$. This is definable in the following two cases:
 - (a) When $*$ is a strict t-norm, by $\delta x = \neg_*(N(x))$
 - (b) When $*$ is a divisible finite t-norm, by $\delta x = (\neg_*(N(x)))^n$.

Finally let us remark that in the canonical chain T_* defined over the product standard chain $\langle [0, 1], \max, \min, *_{\Pi}, \Rightarrow_{\Pi}, 0, 1 \rangle$, both the Łukasiewicz t-norm $*_{\mathbb{L}}$ and its residuum $\Rightarrow_{\mathbb{L}}$ are definable (see Section 4 of Chapter VIII of this Handbook and the references therein) as:

$$x *_{\mathbb{L}} y = N(x *_{\Pi} N(x \Rightarrow_{\Pi} y)) \quad x \Rightarrow_{\mathbb{L}} y = x *_{\Pi} N(x \Rightarrow_{\Pi} N(y)).$$

Notice that the canonical chain relative to product t-norm embeds the canonical chain relative to Łukasiewicz t-norm.

2.4 Expanding the standard chain with other operators

There are several extra operators that are interesting in order to interpret some FDL constructors. Among them we highlight the following ones:

- The *Monteiro–Baaz delta operator* defined in the previous section. Observe that the operator δ is a *crispification* operator (the image is $\{0, 1\}$) and it is truth preserving ($\delta(x) = 1$ if and only if $x = 1$).
- *Truth stresser* and *truth depresser* functions:
 - A function $s: [0, 1] \rightarrow [0, 1]$ is a *truth stresser* if it is non-decreasing and *sub-diagonal* (for every $x \in [0, 1]$, $s(x) \leq x$) and $s(1) = 1$.
 - A function $d: [0, 1] \rightarrow [0, 1]$ is a *truth depresser* if it is non-decreasing and *super-diagonal* (for every $x \in [0, 1]$, $d(x) \geq x$) and $d(0) = 0$.

These operators corresponds to *linguistic modifiers* (in the sense that composed with an evaluation we obtain a *modified* evaluation). For interested readers, see Chapter VIII of this Handbook or papers [71, 79]. Notice that the function δ is a particular case of truth stresser.

- *Aggregation operators* of arity n defined as functions $g: [0, 1]^n \rightarrow [0, 1]$. The properties that these functions must satisfy depend on the type of aggregation operator we are defining. Typical aggregation operators are *weighted sum*, *ordered weighted averaging* (OWA) or more general integrals (see for instance [134]).

3 Introduction to Fuzzy Description Logics

Knowledge used in real applications is usually imperfect and has to address situations of uncertainty, imprecision and vagueness. From a real world viewpoint, vague concepts like “patient with a high fever” and “person living near Paris” have to be considered. Fuzzy Description Logics are natural generalizations of Description Logics to cope with vague concepts and roles (mainly graded concepts and roles). These generalizations interpret concepts and roles as fuzzy sets and fuzzy relations respectively. In this interpretation each individual is an instance of a concept in a certain degree, which is the membership degree in that such individual belongs to the corresponding fuzzy set. These degrees are values belonging to either the real unit interval $[0, 1]$ or a finite chain. In order to combine fuzzy concepts and roles in the basic FDL languages, we use the same basic constructors as in classical DLs but interpreted by means of the operations that are commonly used to combine fuzzy sets. Accordingly with this choice we will interpret the intersection of concepts as the intersection of fuzzy sets which is commonly point-wise defined by means of a continuous (or finite divisible) t-norm (for t-norms and related notions see Section 2). Given the continuous (or finite divisible) t-norm $*$ used to interpret concept intersection, and given its residuum \Rightarrow_* , the other basic constructors are interpreted as follows:

- Universal and empty concepts are interpreted as the constant functions 1 and 0 respectively.
- The complementation will be interpreted as the standard involutive negation function $N(x) = 1 - x$.
- The union will be interpreted as the dual t-conorm of $*$ with respect to N , denoted \oplus . That is, $x \oplus y = N(N(x) * N(y))$.
- The value restriction $\forall R.C$ is interpreted using the infimum. The degree r in which an individual a is an instance of $\forall R.C$ is computed as follows:
 - For each b in the domain we compute the degree α in which a is related to b and the degree β in which b is an instance of C .
 - Then, r is the infimum of all the values $\alpha \Rightarrow_* \beta$ computed for each b , where \Rightarrow_* is the residuum of the t-norm $*$.
- The existential quantification $\exists R.C$ is interpreted using the supremum. The degree r in which an individual a is an instance of $\exists R.C$ is computed as follows:
 - For each b in the domain we compute the degree α in which a is related to b and the degree β in which b is an instance of C .
 - Then, r is the supremum of all the values $\alpha * \beta$ computed for each b .

Notice that the definitions of quantified concepts are generalizations of the ones in classical DL since if we reduce the set of truth values to $\{0, 1\}$, then we obtain the classical definitions.

Thus, in order to define an interpretation we need to first fix a chain of truth values ($[0, 1]$ or a finite chain) and a continuous or divisible finite t-norm over that chain. In fact, interpretations of the FDL languages can be seen as interpretations of the description language over a canonical chain defined by a t-norm $*$ and the involutive negation N .

If we want to have implication of concepts as a constructor in our basic language it must be included as a primitive constructor. The natural choice is to interpret the implication constructor by using the residuum \Rightarrow_* of the t-norm (see [78, Chapter 2, 2.1.3] for a discussion about this choice in the fuzzy logic setting; see also Section 8.2 for a discussion about this choice in the framework of FDLs), and in general $x \Rightarrow_* y$ coincides neither with $\max\{N(x), y\}$ nor with $N(\min\{x, N(y)\})$ (see Example 3.3.1). Thus, we cannot define the implication constructor from union and complementation (or intersection and complementation). Moreover, having implication in the language, we can define:

- A weak complementation interpreted by the negation function $\neg_* x = x \Rightarrow_* 0$
- A weak intersection interpreted by the minimum which is definable as $\min(x, y) = x * (x \Rightarrow_* y)$
- A weak union interpreted by the maximum which is definable as $\max(x, y) = \min((x \Rightarrow_* y) \Rightarrow_* y, (y \Rightarrow_* x) \Rightarrow_* x)$.

Observe that definability of minimum is possible since the t-norm is continuous or finite divisible.

3.1 Basic concept constructors in FDLs

Let us define a *description signature* as a triple $\mathcal{D} = \langle N_I, N_A, N_R \rangle$, where N_I , N_A and N_R are pairwise disjoint sets of individual names, concept names (the *atomic concepts*), and role names (the *atomic roles*), respectively.

Our basic description languages for \mathcal{D} will be propositional languages generated by the set N_A of atomic concepts taken as set of propositional letters, and using as connectives some specific subset of the following set of *constructors*:

$$\mathfrak{C} = \{\sqcap, \sqcup, \rightarrow, \neg, \sim, \wedge, \vee\} \cup \{\perp, \top\} \cup \{\forall R. \mid R \in N_R\} \cup \{\exists R. \mid R \in N_R\}.$$

The set of *concepts* for each description language is inductively defined by the corresponding subset of the following syntactic rules (we use the symbols C, C_1, C_2 as metavariables for concepts):

$C, C_1, C_2 \rightsquigarrow$	A	(atomic concept)
	$C_1 \sqcap C_2$	(concept strong intersection)
	$C_1 \sqcup C_2$	(concept strong union)
	$C_1 \rightarrow C_2$	(concept implication)
	$\neg C$	(weak complementary concept)
	$\sim C$	(strong complementary concept)
	$C_1 \wedge C_2$	(concept weak intersection)
	$C_1 \vee C_2$	(concept weak union)
	\perp	(empty concept)
	\top	(universal concept)
	$\forall R.C$	(value restriction)
	$\exists R.C$	(existential quantification).

EXAMPLE 3.1.1. Consider the atomic concepts LovesCinema and LovesReading and the atomic role IsFriendOf. Then we can form the concepts:

$$\text{LovesCinema} \sqcap \sim \text{LovesReading}$$

“Person who loves cinema but does not love reading.”

$$\text{LovesReading} \sqcap \exists \text{IsFriendOf}. \sim \text{LovesReading}$$

“Person who loves reading but has a friend who do not love reading.”

$$\exists \text{IsFriendOf}. \text{LovesReading} \rightarrow \text{LovesReading}$$

“Person who loves reading if it has a friend who loves reading.”

$$\forall \text{IsFriendOf}. (\text{LovesCinema} \sqcup \text{LovesReading})$$

“Person whose friends either love cinema or love reading.”

3.2 Semantics for basic concept constructors

An *interpretation* for a description signature $\mathcal{D} = \langle N_I, N_A, N_R \rangle$ is a pair

$$\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$$

consisting of a nonempty (crisp) set $\Delta^{\mathcal{I}}$ (called *domain*) and of a *fuzzy interpretation function* $\cdot^{\mathcal{I}}$ that assigns:

- a) To each concept name $A \in N_A$ a fuzzy set, that is, a function

$$A^{\mathcal{I}}: \Delta^{\mathcal{I}} \rightarrow T,$$

- b) To each role name $R \in N_R$ a fuzzy relation, that is, a function

$$R^{\mathcal{I}}: \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow T,$$

- c) To each individual name $a \in N_I$ an object $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.

Let T be the set of truth values, that is, either the real unit interval $[0, 1]$ or a finite chain $C_n = \{0, \frac{1}{n-1}, \frac{2}{n-1}, \dots, \frac{n-2}{n-1}, 1\}$ and fix a t-norm $*$ over T . Given $*$, and a description signature \mathcal{D} , and an $*$ -interpretation is a pair

$$\mathcal{I}_* = \langle \mathcal{I}, * \rangle.$$

DEFINITION 3.2.1 (Semantics for basic languages). *Given an $*$ -interpretation \mathcal{I}_* , the interpretation $\cdot^{\mathcal{I}}$ is extended to complex concepts by assigning to every complex concept D a fuzzy set $D^{\mathcal{I}}: \Delta^{\mathcal{I}} \rightarrow T$ inductively defined as follows:*

$$D^{\mathcal{I}}(x) := \begin{cases} 0 & \text{if } D = \perp \\ 1 & \text{if } D = \top \\ A^{\mathcal{I}}(x) & \text{if } D = A \in N_A \\ 1 - C^{\mathcal{I}}(x) & \text{if } D = \sim C \\ C^{\mathcal{I}}(x) \Rightarrow_* \perp & \text{if } D = \neg C \\ C_1^{\mathcal{I}}(x) * C_2^{\mathcal{I}}(x) & \text{if } D = C_1 \sqcap C_2 \\ \min\{C_1^{\mathcal{I}}(x), C_2^{\mathcal{I}}(x)\} & \text{if } D = C_1 \wedge C_2 \\ C_1^{\mathcal{I}}(x) \oplus C_2^{\mathcal{I}}(x) & \text{if } D = C_1 \sqcup C_2 \\ \max\{C_1^{\mathcal{I}}(x), C_2^{\mathcal{I}}(x)\} & \text{if } D = C_1 \vee C_2 \\ C_1^{\mathcal{I}}(x) \Rightarrow_* C_2^{\mathcal{I}}(x) & \text{if } D = C_1 \rightarrow C_2 \\ \inf\{R^{\mathcal{I}}(x, y) \Rightarrow_* C^{\mathcal{I}}(y) \mid y \in \Delta^{\mathcal{I}}\} & \text{if } D = \forall R.C \\ \sup\{R^{\mathcal{I}}(x, y) * C^{\mathcal{I}}(y) \mid y \in \Delta^{\mathcal{I}}\} & \text{if } D = \exists R.C. \end{cases}$$

EXAMPLE 3.2.2. Consider the concepts $A_1 = \text{LovesCinema}$ and $A_2 = \text{LovesReading}$ and the atomic role $R = \text{isFriendOf}$ from Example 3.1.1; we can form the concept $\forall R.(A_1 \sqcup A_2)$. Now take $T = \{0, \frac{1}{4}, \frac{2}{4}, \frac{3}{4}, 1\}$ and the minimum t-norm. Thus, now we have that $\oplus = \max$ and \Rightarrow_* is the residuated implication corresponding to minimum (see Table 1). Consider now an interpretation \mathcal{I} where $\Delta^{\mathcal{I}}$ is the a set of students of a postgraduate course in La Sorbone, Paris. Suppose that this set is formed by four students marie (a), jacques (b), ivette (c), pierre (d). The interpretations for the concepts A_1, A_2 and the role R are the following fuzzy sets and fuzzy relation:

	a	b	c	d
$A_1^{\mathcal{I}}$	1/2	1/2	1/4	1
$A_2^{\mathcal{I}}$	3/4	1/4	3/4	0

R	a	b	c	d
a	1	3/4	1	1/4
b	3/4	1	1/2	3/4
c	3/4	1/2	1/2	1
d	1/2	3/4	1	1/2

Let us compute the degree in which student a is an instance of concept $\forall R.(A_1 \sqcup A_2)$:

y	$R^{\mathcal{I}}(a, y)$	$A_1^{\mathcal{I}}(y)$	$A_2^{\mathcal{I}}(y)$	$R^{\mathcal{I}}(a, y) \Rightarrow_{\min} \max\{A_1^{\mathcal{I}}(y), A_2^{\mathcal{I}}(y)\}$
a	1	1/2	3/4	$1 \Rightarrow_{\min} \max\{1/2, 3/4\} = 3/4$
b	3/4	1/2	1/4	$3/4 \Rightarrow_{\min} \max\{1/2, 1/4\} = 1/2$
c	1	1/4	3/4	$1 \Rightarrow_{\min} \max\{1/4, 3/4\} = 3/4$
d	1/4	1	0	$1/4 \Rightarrow_{\min} \max\{1, 0\} = 1$

Thus we have:

$$(\forall R.(A_1 \sqcup A_2))^{\mathcal{I}}(a) = \inf\{R^{\mathcal{I}}(a, y) \Rightarrow_{\min} \max\{A_1^{\mathcal{I}}(y), A_2^{\mathcal{I}}(y)\} \mid y \in \Delta^{\mathcal{I}}\} = \frac{1}{2}.$$

In an analogous way we can compute that the degrees in which students b, c, d are instances of concept $\forall R.(A_1 \sqcup A_2)$ are $\frac{1}{2}$, 1 and $\frac{1}{2}$, respectively.

REMARK 3.2.3. Observe that constructors for classical DLs are the same that the ones previously defined for FDLs, but without the difference between strong and weak union, strong and weak intersection, and strong and weak complementation. Indeed, if in Definition 3.2.1, we substitute \mathbf{T}_* by the Boolean algebra of two elements, we obtain the usual semantics for basic constructors of classical DLs. For the classical definition and semantics of basic languages as \mathcal{ALC} see for instance [8].

3.3 The hierarchy of fuzzy attributive languages

Next we will describe the hierarchy of FDLs languages that are built using the basic constructors introduced in previous section. In order to maintain as much as possible the similarity with classical DL notation, we propose the following fuzzy description languages:

- The language \mathcal{EL} , as in the classical case, contains
 - the universal concept \top ,
 - the strong intersection \sqcap , and
 - the existential quantification $\exists R$,
 as concept constructors.
- The language $\mathfrak{N}\mathcal{EL}$ is built by adding the weak complementation \neg to \mathcal{EL} .
- The language \mathcal{ELC} is built by adding the strong complementation \sim to \mathcal{EL} .
- The language \mathcal{FL}_0 , as in the classical case, contains these concept constructors:
 - the universal concept \top ,
 - the strong intersection \sqcap , and
 - the value restriction $\forall R$.
- The language \mathcal{FL}^- is built by adding to the language \mathcal{FL}_0 the restricted existential quantification $\exists R.\top$.
- The language \mathcal{AL} is built by adding to the language \mathcal{FL}^- the strong complementation \sim applied only to atomic concepts.
- The language \mathcal{ALC} is built by adding the full strong complementation \sim to the language \mathcal{AL} .

We will take as minimal languages of our hierarchy \mathcal{EL} and \mathcal{AL} . Now we will construct the hierarchy of logical based FDL languages beginning with \mathcal{AL} . To this end, we need to introduce explicitly as primitive constructors the following ones: the existential quantifier ($\exists R.$), the complementation (\sim), the implication (\rightarrow), and the strong union (\sqcup). The presence of each one of these constructors in the language will be denoted by adding to the acronym of the language the symbols \mathcal{E} , \mathcal{C} , \mathfrak{I} ,³ and \mathcal{U} , respectively.

The hierarchy of languages that we obtain depends on the semantical choice, i.e., on the truth values set ($[0, 1]$ or C_n), and the t-norm used to interpret strong conjunction. We can distinguish two different cases:

1. *The hierarchy obtained when strong conjunction is interpreted as (finite or infinite) Łukasiewicz t-norm.* This is similar to the classical case. Since in this case both complementations—weak and strong—coincide (see Section 2), other constructors are definable from intersection and complementation like in the classical case: (strong) union and existential restriction are definable by duality as $C \sqcup D \equiv \neg(\neg C \sqcap \neg D)$ and $\exists R.C \equiv \neg\forall R.\neg C$ respectively, and implication is definable using (strong) complementation and union (or intersection) by the equivalence $C \rightarrow D \equiv \neg C \sqcup D \equiv \neg(C \sqcap \neg D)$. Taking into account these definabilities, we have that $\mathcal{ALU}\mathcal{EC} = \mathcal{ALC}$. Thus the graph of FDLs languages is the one given in Figure 2 that is the same as in the classical case.

³ We have chosen \mathfrak{I} because the calligraphic \mathcal{I} is extensively used in the literature on DLs to denote the constructor for inverse roles.

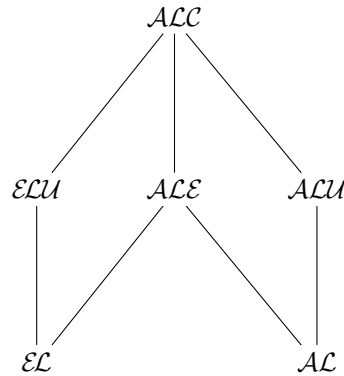


Figure 2. The hierarchy of basic languages under Łukasiewicz semantics

2. *The hierarchy obtained when strong conjunction is interpreted as a (finite or infinite) divisible t-norm different from Łukasiewicz.* In these cases, the hierarchy of basic languages obtained is more cumbersome because we do not have in general the same possibility of reducing languages as in the classical case. This is because in the semantics of our logics:

- Although union is definable from conjunction by duality with respect to strong complementation by $C \sqcup D \equiv \sim(\sim C \sqcap \sim D)$, this is not true for the duality with respect to weak complementation. The reason is that the interpretation of \sim is a continuous function, which is not the case for \neg .
- When the t-norm is different from Łukasiewicz, weak and strong complementation do not coincide, and implication is not definable from intersection and complementation, neither the weak nor the strong one (as an illustration see Example 3.3.1).
- The particular semantics of the universal constructor $\forall R.$ and the existential constructor $\exists R.$ does not allow to define one in terms of the other by duality (see Example 3.3.2 and Proposition 3.3.3).

The new hierarchy of FDL languages over the logic of a continuous t-norm beginning with \mathcal{AL} is represented in Figure 3, that shows the inclusions among the languages obtained by successively adding different basic operators. Observe that, due to the above defined semantics, in our framework the languages \mathcal{ALE} and \mathcal{JAL} are not strictly contained in \mathcal{ALC} .

EXAMPLE 3.3.1. Let $* = \min$, and take $x = 0.2$ and $y = 0.7$. Since $x < y$, we have $x \Rightarrow_{\min} y = 1$. However we have:

$$\begin{aligned} \max\{\neg_{\min}(x), y\} &= \max\{0, 0.7\} = 0.7 \\ \max\{N(x), y\} &= \max\{0.8, 0.7\} = 0.8. \end{aligned}$$

Observe that $N(0.2) = 0.8 \neq 0 = \neg_{\min}(0.2)$.

This example shows that Gödel implication is not definable using the standard operators of the Minimum t-norm. The following example shows that the duality of existential and value restrictions does not hold either.

EXAMPLE 3.3.2. Take a description signature with a unique atomic concept C and a unique atomic role R . Take $T = \{0, \frac{1}{2}, 1\}$, and $*$ = min. Now consider a interpretation \mathcal{I} , where $\Delta^{\mathcal{I}} = \{a, b, c\}$, and R and C are interpreted as:

- $R^{\mathcal{I}}(a, b) = R^{\mathcal{I}}(b, c) = 1$; $R^{\mathcal{I}}(a, a) = R^{\mathcal{I}}(a, c) = \frac{1}{2}$; and $R^{\mathcal{I}}(x, y) = 0$ otherwise.
- $C^{\mathcal{I}}(b) = \frac{1}{2}$; and $C^{\mathcal{I}}(x) = 1$ otherwise.

Then, $(\exists R.C)^{\mathcal{I}}(a) = \sup\{\min\{R^{\mathcal{I}}(a, y), C^{\mathcal{I}}(y)\} \mid y \in \{a, b, c\}\} = \frac{1}{2}$ but

$$\begin{aligned} (\sim \forall R. \sim C)^{\mathcal{I}}(a) &= 1 - \inf\{R^{\mathcal{I}}(a, y) \Rightarrow_{\min} (1 - C^{\mathcal{I}}(y)) \mid y \in \{a, b, c\}\} \\ &= 1 - \inf\{\frac{1}{2} \Rightarrow_{\min} 0, 1 \Rightarrow_{\min} \frac{1}{2}, \frac{1}{2} \Rightarrow_{\min} 0\} \\ &= 1 - \inf\{0, \frac{1}{2}, 0\} = 1. \end{aligned}$$

PROPOSITION 3.3.3. *In any language containing value and existential restrictions and weak negation among its constructors, for each atomic role R and concept C , it holds that $\neg \exists R.C = \forall R. \neg C$. However, in general, $\neg \forall R.C$ is not equal to $\exists R. \neg C$.*

Proof. Let $a \in \Delta^{\mathcal{I}}$. We have:

$$\begin{aligned} (\neg \exists R.C)^{\mathcal{I}}(a) &= (\exists R.C \rightarrow \perp)^{\mathcal{I}}(a) = (\exists R.C)^{\mathcal{I}}(a) \Rightarrow_* 0 \\ &= \sup\{R^{\mathcal{I}}(a, y) * C^{\mathcal{I}}(y) \mid y \in \Delta^{\mathcal{I}}\} \Rightarrow_* 0. \end{aligned}$$

And now, by applying Proposition 2.1.2 and the properties of residuation, we have:

$$\begin{aligned} (\neg \exists R.C)^{\mathcal{I}}(a) &= \sup\{R^{\mathcal{I}}(a, y) * C^{\mathcal{I}}(y) \mid y \in \Delta^{\mathcal{I}}\} \Rightarrow_* 0 \\ &= \inf\{(R^{\mathcal{I}}(a, y) * C^{\mathcal{I}}(y)) \Rightarrow_* 0 \mid y \in \Delta^{\mathcal{I}}\} \\ &= \inf\{R^{\mathcal{I}}(a, y) \Rightarrow_* (C^{\mathcal{I}}(y) \Rightarrow_* 0) \mid y \in \Delta^{\mathcal{I}}\} \\ &= \inf\{R^{\mathcal{I}}(a, y) \Rightarrow_* (\neg C)^{\mathcal{I}}(y) \mid y \in \Delta^{\mathcal{I}}\} = (\forall R. \neg C)^{\mathcal{I}}(a). \end{aligned}$$

Take now the language considered in Example 3.3.2, interpreting R as the same fuzzy relation and C as $C^{\mathcal{I}}(a) = \frac{1}{2}$, $C^{\mathcal{I}}(b) = 1$, $C^{\mathcal{I}}(c) = 0$. Then,

$$(\exists R. \neg C)^{\mathcal{I}}(a) = \sup\{\min\{R^{\mathcal{I}}(a, y), (\neg C)^{\mathcal{I}}(y)\} \mid y \in \{a, b, c\}\} = \frac{1}{2}$$

but

$$(\neg \forall R.C)^{\mathcal{I}}(a) = \neg_{\min} \inf\{R^{\mathcal{I}}(a, y) \Rightarrow_{\min} C^{\mathcal{I}}(y) \mid y \in \{a, b, c\}\} = 1. \quad \square$$

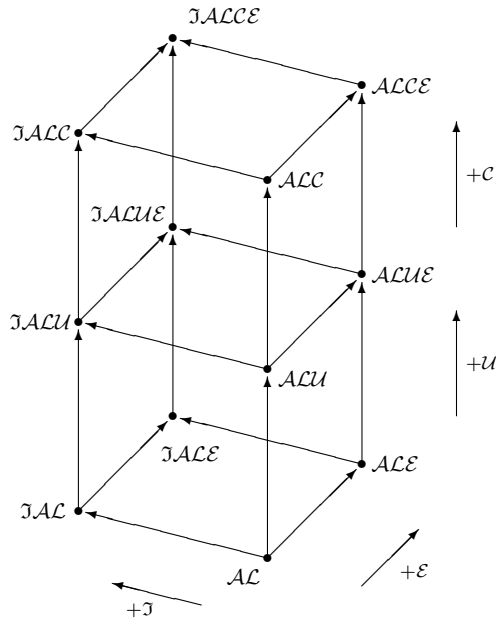


Figure 3. Hierarchy of languages under t-norm based logics different from Łukasiewicz

Since in languages with complementation \sim , the strong union \sqcup is definable from the strong intersection \sqcap and \sim , then the language \mathcal{ALU} is strictly contained in the language $\mathcal{ALC} = \mathcal{ALCU}$. Following our convention to denote languages, the top of the hierarchy in Figure 3 (corresponding to \mathcal{ALC} in the classical case) is called in our framework \mathcal{JALCE} . In papers on FDLs (specially the initial ones) it is common the use of \mathcal{ALC} for the top of the hierarchy which is acceptable in a broad sense, but if we want to make explicit which are the constructors, then we need to be more accurate in the cases having non-Łukasiewicz semantics.

An analogous hierarchy can be built from \mathcal{EL} , the main difference being that we need to change the role played by existential quantification for an analogous role for value restriction (universal quantification). Indeed, using \mathcal{V} for universal quantification,⁴ the language \mathcal{JELV} (obtained by adding \perp , universal quantification and implication to \mathcal{EL}) coincides with the language \mathcal{JALCE} of Figure 3. Notice that \mathcal{NEL} is an intermediate language between \mathcal{EL} and \mathcal{JEL} .

⁴ The ‘ \mathcal{V} ’ in this case stands for ‘value restriction.’

Finally, let us remark that some FDL languages have other constructors like modifiers, crispification, or aggregation (for definitions of these additional constructors see Section 5.1). In these cases, if we want to be accurate, we need to introduce some letter for each one of these constructors, for example \mathcal{M} , \mathcal{D} or \mathcal{G} respectively.

3.4 Knowledge bases

Like in classical DLs, knowledge bases are formed by two finite sets:

- A TBox (Terminological Box), which contains *graded inclusion axioms* stating that certain concept is subsumed by another at least in a given degree.
- An ABox (Assertional Box), which contains *graded assertion axioms* stating that, at least in a given degree, an individual is an instance of some concept, or a pair of individuals is an instance of some role.

A *knowledge base* is a pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, where the first component is a TBox and the second one is an ABox. Let C, D be concepts, R be an atomic role, a, b be individual names and $r \in T$.

A *graded concept inclusion axiom* is an expression of the form:

$$\langle C \sqsubseteq D, r \rangle.$$

A *graded concept assertion axiom* is an expression of the form:

$$\langle C(a), r \rangle.$$

A *graded role assertion axiom* is an expression of the form:

$$\langle R(a, b), r \rangle.$$

Fix a set of truth values T and a t-norm $*$ defined on T , and let \mathcal{I} be an interpretation.

- We say that \mathcal{I} satisfies the *graded concept inclusion axiom* $\langle C \sqsubseteq D, r \rangle$ when

$$\inf\{C^{\mathcal{I}}(x) \Rightarrow_* D^{\mathcal{I}}(x) \mid x \in \Delta^{\mathcal{I}}\} \geq r.$$

- We say that \mathcal{I} satisfies the *graded concept assertion axiom* $\langle C(a), r \rangle$ when

$$C^{\mathcal{I}}(a^{\mathcal{I}}) \geq r.$$

- We say that \mathcal{I} satisfies the *graded role assertion axiom* $\langle R(a, b), r \rangle$ when

$$R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) \geq r.$$

- We say that a graded axiom is **-satisfiable* if there exists an interpretation \mathcal{I} which satisfies that axiom.

When \mathcal{I} satisfies a graded axiom $\langle \alpha, r \rangle$ we say that \mathcal{I} is an **-model* (or simply a *model*) of $\langle \alpha, r \rangle$, and we write $\mathcal{I} \models_* \langle \alpha, r \rangle$ (or simply $\mathcal{I} \models \langle \alpha, r \rangle$ when the t-norm is clear from the context). We write $\mathcal{I} \not\models_* \langle \alpha, r \rangle$ otherwise. We will say that an interpretation \mathcal{I} is a *model* of a knowledge base \mathcal{K} if \mathcal{I} is a model of all the axioms in \mathcal{K} .

For instance, in the setting of Example 3.2.2, the axioms:

$$\langle \text{LovesCinema}(\text{marie}), \frac{3}{4} \rangle, \langle \text{LovesReading}(\text{jacques}), \frac{1}{4} \rangle, \langle \text{isFriendOf}(\text{ivette}, \text{marie}), \frac{1}{4} \rangle$$

are possible graded assertion axioms. Observe that for the interpretation \mathcal{I} given in such example, we have:

$$\mathcal{I} \not\models \langle \text{LovesCinema}(\text{marie}), \frac{3}{4} \rangle \quad \text{and} \quad \mathcal{I} \models \langle \text{LovesReading}(\text{jacques}), \frac{1}{4} \rangle.$$

We also have that $\mathcal{I} \models \langle \text{isFriendOf}(\text{ivette}, \text{marie}), \frac{1}{4} \rangle$. Therefore, \mathcal{I} does not satisfy an ABox containing all these three graded assertion axioms. A possible graded inclusion axiom is $\langle \text{LovesReading} \sqsubseteq \text{LovesCinema}, \frac{1}{4} \rangle$. Observe that the interpretation \mathcal{I} satisfies it since:

$$\inf\{\text{LovesReading}^{\mathcal{I}}(x) \Rightarrow_* \text{LovesCinema}^{\mathcal{I}}(x) \mid x \in \Delta^{\mathcal{I}}\} = \frac{1}{4}.$$

3.5 Reasoning tasks

Reasoning in FDLs involves the same kind of tasks as in the classical case but their results depend on the chosen truth values set T and t-norm $*$. We consider graded notions of reasoning tasks. Given a knowledge base, \mathcal{K} , concepts C and D , and a truth value $r \in T$, we define:

1. \mathcal{K} is **-consistent* if all the axioms in \mathcal{K} are **-satisfiable*, i.e., if there is a model of \mathcal{K} .
2. C is **-satisfiable to a degree greater than or equal to r* with respect to \mathcal{K} if there exists a model \mathcal{I} of \mathcal{K} , and an element $u \in \Delta^{\mathcal{I}}$ such that $C^{\mathcal{I}}(u) \geq r$. In particular, C is *positively *-satisfiable* with respect to \mathcal{K} if it is **-satisfiable* to a degree strictly greater than 0 with respect to \mathcal{K} .
3. C is **-subsumed by D to a degree greater than or equal to r* with respect to \mathcal{K} if, for every model \mathcal{I} of \mathcal{K} , $\inf\{C^{\mathcal{I}}(x) \Rightarrow_* D^{\mathcal{I}}(x) \mid x \in \Delta^{\mathcal{I}}\} \geq r$.⁵
4. An axiom is **-entailed by \mathcal{K}* if it is **-satisfied* in every model of \mathcal{K} .
5. The *Best Entailment Degree* (BED) of $\tau \in \{C(a), R(a, b), C \sqsubseteq D\}$ with respect to \mathcal{K} is the maximum degree r such that $\langle \tau, r \rangle$ is **-entailed by \mathcal{K}* .
6. The *Best Satisfiability Degree* (BSD) of C with respect to \mathcal{K} is the maximum degree r such that C is **-satisfiable to a degree greater than or equal to r* .

In what follows, we will assume that $r > 0$ for **-satisfiability* (consistency, subsumption) since otherwise the result of the reasoning test is trivially true.

⁵ Observe that the logical notion of *validity* applied to a concept C can be expressed in FDL as the fact that C **-subsumes* \top to degree 1.

4 Fuzzy Description Logics from the point of view of MFL

In [80] (see also [82]) Hájek proposes to define FDLs from a logical point of view in an analogous way as DLs can be defined from classical first order logic. It is well known that classical \mathcal{ALC} can be interpreted as a fragment of first order classical logic (see [111]). In a similar way, the fuzzy description logic \mathcal{FALCE} can be interpreted as a fragment of certain fuzzy first order logics. The fuzzy logics used in this interpretation are semantically defined as multi-valued logics that are complete with respect to interpretations on the algebraic structure defined on the chain either $[0, 1]$ or C_n by a divisible t-norm and its residuum and the involutive negation function N . It is worth to notice that the way we define these logical systems is different from the axiomatic approach mainly followed in the overall Handbook. Thus, we introduce them in next sections. We define the logics (propositional and first-order) where the description logic \mathcal{FALCE} is interpreted. In Section 4.3 we introduce the notion of *instance* of a description as a first-order open formula associated with it and we prove that the semantical interpretation of a concept coincides with the semantical interpretation of its instance in first-order logic.

4.1 A logical framework for FDLs: The propositional logics

Let us consider the language \mathcal{L} inductively built from a countable set of propositional variables $\Phi = \{p_j \mid j \in J\}$ using the connectives $\vee, \wedge, \underline{\vee}, \&, \rightarrow, \neg, \sim$, and the constants $\bar{0}$ and $\bar{1}$. Note that we have considered all possible connectives; in fact $\&, \rightarrow, \sim$, and $\bar{0}$ would be enough since the others are definable from them.

A *propositional evaluation* for the \mathcal{L} -formulas over the canonical chain \mathbf{T}_* is a map $e: \Phi \rightarrow T$ which is extended to all formulas by setting:

$$\begin{aligned} e(\varphi \wedge \psi) &= \min\{e(\varphi), e(\psi)\} & e(\neg\varphi) &= e(\varphi) \rightarrow_* 0 \\ e(\varphi \vee \psi) &= \max\{e(\varphi), e(\psi)\} & e(\sim\varphi) &= 1 - e(\varphi) \\ e(\varphi \& \psi) &= e(\varphi) * e(\psi) & e(\bar{0}) &= 0 \\ e(\varphi \underline{\vee} \psi) &= e(\varphi) \oplus e(\psi) & e(\bar{1}) &= 1 \\ e(\varphi \rightarrow \psi) &= e(\varphi) \rightarrow_* e(\psi). \end{aligned}$$

Given the propositional language \mathcal{L} , we define the logic Λ^* semantically as the consequence relation associated with evaluations on the canonical chain \mathbf{T}_* by putting, for all sets $\Gamma \cup \{\varphi\}$ of \mathcal{L} -formulas,

$$\Gamma \models_{\Lambda^*} \varphi \text{ iff for every evaluation } e \text{ over } \mathbf{T}_*, e(\gamma) = 1 \text{ for all } \gamma \in \Gamma \text{ implies } e(\varphi) = 1.$$

REMARK 4.1.1. *The first logic used in FDLs was the so-called Zadeh logic, semantically defined on the chain $\langle [0, 1], \max, \min, \rightarrow, N, 0, 1 \rangle$, where \rightarrow is the so-called S-implication defined as $x \rightarrow y = \max(N(x), y)$, and N is the standard negation $N(x) = 1 - x$. This logical system is not residuated and not considered in MFL (see the historical remarks for a discussion about the reasons for such decision). Nevertheless some earlier results on FDLs based on Zadeh logic are presented in this chapter since they are interesting for further results in the FDLs based on MFL.*

We could introduce some other connectives whose semantics is a function of the corresponding arity on the canonical chain. That is the case of the *Monteiro–Baaz* Δ , that corresponds to crispification and whose semantics is defined by the function δ , or *hedges*, which interpret linguistic modifiers and whose semantics are the unary functions called truth stressers or truth depressers. Another expansions can be obtained by adding *aggregation operators*, which are semantically defined as aggregation functions of the corresponding arity.

4.2 A logical framework for FDLs: The predicate logics

From the propositional logics defined above we can define their first order versions. As defined in Chapter I of this Handbook, a *predicate language* or *first order signature* is a triple $\Sigma = \langle \mathcal{P}, \mathcal{F}, \mathbf{ar} \rangle$, where $\mathcal{P} = \{P, Q, \dots\}$ is a non empty set of *predicate symbols*, $\mathcal{F} = \{f, g, \dots\}$ is a set (disjoint from \mathcal{P}) of *functional symbols*, and \mathbf{ar} is a function assigning to each predicate and function symbol a natural number called the *arity* of the symbol. The functional symbols of arity 0 are called *object constants*. The predicate symbols of arity 0 are called *truth constants*. We will consider that the set of nullary predicates is parametrized by a subalgebra \mathcal{S} of \mathbf{T}_* . Indeed this set will be denoted as $\{\bar{r} \mid r \in S\}$.

Let $V = \{x, y, \dots\}$ be a countable set of objects whose elements are called *object variables*. The set of Σ -terms is the least set X such that $V \subseteq X$, and if f is a k -ary functional symbol and $t_1, \dots, t_k \in X$, then $f(t_1, \dots, t_k) \in X$. An *atomic Σ -formula* is an expression of the form $P(t_1, \dots, t_k)$, where P is a k -ary predicate symbol and t_1, \dots, t_k are Σ -terms. Atomic Σ -formulas and nullary logical connectives of \mathcal{L} are called *atomic $\langle \mathcal{L}, \Sigma \rangle$ -formulas*. The set of $\langle \mathcal{L}, \Sigma \rangle$ -formulas is the least set such that:

- X contains the atomic $\langle \mathcal{L}, \Sigma \rangle$ -formulas,
- X is closed under the logical connectives of \mathcal{L} , and
- if $\alpha \in X$ and x is an object variable, then $(\forall x)\alpha, (\exists x)\alpha \in X$.

An $\langle \mathcal{L}, \Sigma \rangle$ -theory is a set of $\langle \mathcal{L}, \Sigma \rangle$ -formulas. For the sake of simplicity, since the propositional language is fixed, we will speak about Σ -formulas and Σ -theories. An occurrence of a variable x in a formula φ is *bound* if it is in the scope of some quantifier over x ; otherwise it is called a *free* occurrence. A variable is free in a formula φ if it has a free occurrence in φ . A term is *closed* if it contains no variables. A formula is *closed* if it has no free variables; closed formulas are also called *sentences*.

A \mathbf{T}_* -structure for the predicate language Σ is a tuple

$$\mathbf{M} = \langle M, \{P^{\mathbf{M}} \mid P \in \mathcal{P}\}, \{f^{\mathbf{M}} \mid f \in \mathcal{F}\} \rangle, \text{ where}$$

- 1) M is a non-empty set, called the *universe* of the structure.
- 2) For each k -ary predicate symbol $P \in \mathcal{P}$, with $k \geq 1$, $P^{\mathbf{M}}$ is a fuzzy k -ary relation defined on M , that is, a function $P^{\mathbf{M}}: M^k \rightarrow T$, and an element $r \in S \subseteq T$ if P is the truth constant \bar{r} .
- 3) For each k -ary functional symbol $f \in \mathcal{F}$, with $k \geq 1$, $f^{\mathbf{M}}$ is a function $f^{\mathbf{M}}: M^k \rightarrow M$, and an element of M if f is an object constant.

Given a \mathbf{T}_* -structure \mathbf{M} , a map v assigning an element $v(x) \in M$ to each variable x is called an *assignment of the variables in \mathbf{M}* (an \mathbf{M} -assignment). Given \mathbf{M} and v , the *value of a term t in \mathbf{M}* , denoted by $\|t\|_{\mathbf{M},v}$, is defined as $v(x)$ when t is a variable x , and as $a^{\mathbf{M}}$ when t is a constant a , and as $f_{\mathbf{M}}(\|t_1\|_{\mathbf{M},v}, \dots, \|t_k\|_{\mathbf{M},v})$ when t is of the form $f(t_1, \dots, t_k)$. In order to emphasize that a formula α has its free variables in $\{x_1, \dots, x_n\}$, we will denote it by $\alpha(x_1, \dots, x_n)$. Let v be an \mathbf{M} -assignment such that $v(x_1) = b_1, \dots, v(x_n) = b_n$. The *truth value in \mathbf{M} over the canonical chain \mathbf{T}_* of the predicate formula $\varphi(x_1, \dots, x_n)$ for the assignment v* , denoted by $\|\varphi\|_{\mathbf{M},v}^*$ or by $\|\varphi(b_1, \dots, b_n)\|_{\mathbf{M}}^*$, is a value in the carrier T of the canonical chain defined inductively as follows:

$$\begin{aligned}
P^{\mathbf{M}}(\|t_1\|_{\mathbf{M},v}, \dots, \|t_k\|_{\mathbf{M},v}) & \text{ if } \varphi = P(t_1, \dots, t_k) \\
r & \text{ if } \varphi = \bar{r} \\
1 - \|\alpha\|_{\mathbf{M},v}^* & \text{ if } \varphi = \sim \alpha \\
\max\{\|\alpha\|_{\mathbf{M},v}^*, \|\beta\|_{\mathbf{M},v}^*\} & \text{ if } \varphi = \alpha \vee \beta \\
\min\{\|\alpha\|_{\mathbf{M},v}^*, \|\beta\|_{\mathbf{M},v}^*\} & \text{ if } \varphi = \alpha \wedge \beta \\
\|\alpha\|_{\mathbf{M},v}^* * \|\beta\|_{\mathbf{M},v}^* & \text{ if } \varphi = \alpha \& \beta \\
\|\alpha\|_{\mathbf{M},v}^* \Rightarrow_* \|\beta\|_{\mathbf{M},v}^* & \text{ if } \varphi = \alpha \rightarrow \beta \\
\inf\{\|\alpha(a, b_1, \dots, b_n)\|_{\mathbf{M}}^* \mid a \in M\} & \text{ if } \varphi = (\forall x)\alpha(x, x_1, \dots, x_n) \\
\sup\{\|\alpha(a, b_1, \dots, b_n)\|_{\mathbf{M}}^* \mid a \in M\} & \text{ if } \varphi = (\exists x)\alpha(x, x_1, \dots, x_n).
\end{aligned}$$

We will write $\langle \mathbf{T}_*, \mathbf{M} \rangle \models \varphi$ if $\|\varphi\|_{\mathbf{M},v}^* = 1$ for every \mathbf{M} -assignment v . When \mathbf{T}_* is known from the context, then we simply write $\mathbf{M} \models \varphi$. If $\langle \mathbf{T}_*, \mathbf{M} \rangle \models \varphi$ for each \mathbf{T}_* -structure \mathbf{M} , then we say that φ is a \mathbf{T}_* -tautology and we write $\mathbf{T}_* \models \varphi$. A \mathbf{T}_* -structure \mathbf{M} is a \mathbf{T}_* -model, or simply a *model of a first order theory*, of a theory Γ if $\langle \mathbf{T}_*, \mathbf{M} \rangle \models \varphi$ for each $\varphi \in \Gamma$. If $\Gamma = \{\varphi\}$, we say that \mathbf{M} is a model of φ . We will say that a formula φ is *1-satisfiable* if there is a \mathbf{T}_* -model of φ . We will say that a formula φ is *s-satisfiable* if there exists a \mathbf{T}_* -structure \mathbf{M} , a value $s \in T$, and an assignment v such that $\|\varphi\|_{\mathbf{M},v}^* = s$.

We define the logic $\Lambda^*\forall$ in the following way: Given a set $\Gamma \cup \{\varphi\}$ of $\langle \mathcal{L}, \Sigma \rangle$ -formulas, we say that: φ is a *consequence of the theory Γ in $\Lambda^*\forall$* , and we write $\Gamma \models_* \varphi$ whenever every \mathbf{T}_* -model of Γ is also a \mathbf{T}_* -model of φ .

Motivated by his studies on fuzzy description logics, Hájek introduced in [80] the notion of *witnessed models*. In what follows we present this notion because it is used in further sections of this chapter.

An interpretation \mathbf{M} is *witnessed* if, for every formula $\varphi(x, y_1, \dots, y_j)$ and any choice $w_1, \dots, w_j \in M$ of values of y_1, \dots, y_j , there exists $w \in M$ such that

$$\|(\forall x)\varphi(x, w_1, \dots, w_j)\|_{\mathbf{M}}^* = \|\varphi(w, w_1, \dots, w_j)\|_{\mathbf{M}}^*,$$

and there exists $w' \in M$ such that

$$\|(\exists x)\varphi(x, w_1, \dots, w_j)\|_{\mathbf{M}}^* = \|\varphi(w', w_1, \dots, w_j)\|_{\mathbf{M}}^*.$$

REMARK 4.2.1. *For people acquainted with MFL it is interesting to notice that the logic of a t-norm as defined in MFL does not coincide with the logic Λ^* . The logic of a t-norm $*$ defined in the MFL setting (see Section 5 of Chapter V of this Handbook) is the fuzzy logic whose associated variety is the one generated by $[0, 1]_*$ while the logic Λ^* is the fuzzy logic semantically defined by evaluations on $[0, 1]_*$. These two logics only coincide when the logic of a t-norm $*$ is standard complete. This is the case for both Gödel logic (the logic of minimum t-norm) and the logic of a finite divisible t-norm. The fact that, in general, Λ^* does not coincide with the logic of a continuous t-norm is the main motivation to define the logic Λ^* in this chapter. Notice also that the logic of a continuous t-norm (over $[0, 1]$ or finite) is finitely axiomatizable even for its first-order version. This is not the case in general for logics Λ^* and $\Lambda^*\forall$. Indeed, the logic $\Lambda^*\forall$ when $*$ is the Łukasiewicz t-norm is not recursively enumerable and when $*$ is the product t-norm is even not arithmetical (see Section 3 of Chapter XI of this Handbook).*

4.3 Basic fuzzy description logics as fragments of fuzzy first-order logics

In order to interpret our fuzzy description logics as fragments of a predicate logic $\Lambda^*\forall$, we associate any description signature:

$$\mathcal{D} = \langle N_I, N_A, N_R \rangle,$$

to the first-order signature $\Sigma_{\mathcal{D}} = \langle \mathcal{C}_{\mathcal{D}}, \mathcal{P}_{\mathcal{D}} \rangle$, being $\mathcal{C}_{\mathcal{D}} = N_I$ and $\mathcal{P}_{\mathcal{D}} = N_A \cup N_R$, in such a way that we interpret each individual name in N_I as an object constant, each atomic concept in N_A as a unary predicate symbol, and each atomic role in N_R as a binary predicate symbol.

Now we introduce the notion of *instance of a description* as a first-order open formula associated with it. Then, we will prove that the semantical interpretation of a concept coincides with the semantical interpretation of its instance in first-order logic.

DEFINITION 4.3.1 (Instance of a description). Let \mathcal{D} be a description signature. To each concept we assign a first order formula, its *instance*, in the following way: Given a term t and a concept D , the *instance* $D(t)$ of D is defined as follows:

$$\begin{aligned} A(t) & \text{ if } D \text{ is an atomic concept } A \\ C_1(t) \& C_2(t) & \text{ if } D = C_1 \sqcap C_2 \\ C_1(t) \vee C_2(t) & \text{ if } D = C_1 \sqcup C_2 \\ C_1(t) \rightarrow C_2(t) & \text{ if } D = C_1 \rightarrow C_2 \\ \neg C(t) & \text{ if } D = \neg C \\ \sim C(t) & \text{ if } D = \sim C \\ C_1(t) \wedge C_2(t) & \text{ if } D = C_1 \wedge C_2 \\ C_1(t) \vee C_2(t) & \text{ if } D = C_1 \vee C_2 \\ \bar{0} & \text{ if } D = \perp \\ \bar{1} & \text{ if } D = \top \\ (\forall y)(R(t, y) \rightarrow C(y)) & \text{ if } D = \forall R.C \\ (\exists y)(R(t, y) \& C(y)) & \text{ if } D = \exists R.C \end{aligned}$$

Finally, an *instance* of an atomic role R is any atomic first order formula $R(t_1, t_2)$, where t_1 and t_2 are terms.

The next proposition states the relationship between interpretations of concepts in a description logic and interpretations of their instances in the corresponding first-order logic.

PROPOSITION 4.3.2. *Let $\mathcal{D} = \langle N_I, N_A, N_R \rangle$ be a description signature, and $*$ be a continuous or divisible finite t-norm.*

- a) *Given an interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ for the signature \mathcal{D} , extended to complex concepts by using $*$, we define the \mathbf{T}_* -interpretation:*

$$\mathbf{M}_{\mathcal{I}} = \langle \Delta^{\mathcal{I}}, \{A^{\mathcal{I}} \mid A \in N_A\} \cup \{R^{\mathcal{I}} \mid R \in N_R\}, \{a^{\mathcal{I}} \mid a \in N_I\} \rangle.$$

Then, for every $x \in \Delta^{\mathcal{I}}$,

$$\|C(x)\|_{\mathbf{M}_{\mathcal{I}}}^* = C^{\mathcal{I}}(x).$$

- b) *Given a \mathbf{T}_* -interpretation for the signature \mathcal{D} ,*

$$\mathbf{M} = \langle M, \{A^{\mathbf{M}} \mid A \in N_A\} \cup \{R^{\mathbf{M}} \mid R \in N_R\}, \{a^{\mathbf{M}} \mid a \in N_I\} \rangle,$$

we define the interpretation: $\mathcal{I}_{\mathbf{M}} = \langle \Delta^{\mathcal{I}_{\mathbf{M}}}, \cdot^{\mathcal{I}_{\mathbf{M}}} \rangle$, where $\Delta^{\mathcal{I}_{\mathbf{M}}} = M$, $A^{\mathcal{I}_{\mathbf{M}}} = A^{\mathbf{M}}$, $R^{\mathcal{I}_{\mathbf{M}}} = R^{\mathbf{M}}$, and $a^{\mathcal{I}_{\mathbf{M}}} = a^{\mathbf{M}}$. Then, for every $w \in M$,

$$C^{\mathcal{I}_{\mathbf{M}}}(w) = \|C(w)\|_{\mathbf{M}}^*.$$

- c) *For every interpretation \mathcal{I} for the signature \mathcal{D} extended to complex concepts by using $*$, the following holds:*

$$\mathcal{I}_{\mathbf{M}_{\mathcal{I}}} = \mathcal{I}.$$

- d) *For every \mathbf{T}_* -interpretation \mathbf{M} for the signature \mathcal{D} , the following holds:*

$$\mathbf{M}_{\mathcal{I}_{\mathbf{M}}} = \mathbf{M}.$$

Proof. Items a) and b) are proved by an easy induction. Items c) and d) are trivial. \square

Therefore, the notion of instance allows us to understand description languages as fragments of predicate fuzzy logics. This means that we have a translation of our description languages into first-order fuzzy logic.⁶ This translation can be extended to satisfiability of axioms of a knowledge base as stated in the following proposition.

PROPOSITION 4.3.3. *Given an interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ for the signature \mathcal{D} and its corresponding \mathbf{T}_* -interpretation $\mathbf{M}_{\mathcal{I}}$, then*

- a) $\mathcal{I} \models_* \langle C \sqsubseteq D, r \rangle$ if and only if $\langle \mathbf{T}_*, \mathbf{M}_{\mathcal{I}} \rangle \models_* \bar{r} \rightarrow (\forall x)(C(x) \rightarrow D(x)).$
b) $\mathcal{I} \models_* \langle C(a), r \rangle$ if and only if $\langle \mathbf{T}_*, \mathbf{M}_{\mathcal{I}} \rangle \models_* \bar{r} \rightarrow C(a).$
c) $\mathcal{I} \models_* \langle R(a, b), r \rangle$ if and only if $\langle \mathbf{T}_*, \mathbf{M}_{\mathcal{I}} \rangle \models_* \bar{r} \rightarrow R(a, b).$

⁶ In [54] direct translations of the considered description languages to the corresponding first order logics are given in a similar way as it is done in the classical case (see [30]).

Proof. We prove only *a*). The other items are proved analogously.

$$\begin{aligned}
\mathcal{I} \models_* \langle C \sqsubseteq D, r \rangle &\Leftrightarrow \inf\{C^{\mathcal{I}}(x) \Rightarrow_* D^{\mathcal{I}}(x) \mid x \in \Delta^{\mathcal{I}}\} \geq r \\
&\Leftrightarrow \inf\{r \Rightarrow_* (C^{\mathcal{I}}(x) \Rightarrow_* D^{\mathcal{I}}(x)) \mid x \in \mathbf{M}_{\mathcal{I}}\} = 1 \\
&\Leftrightarrow \|\bar{r} \rightarrow (\forall x)(C(x) \rightarrow D(x))\|_{\mathbf{M}_{\mathcal{I}}}^* = 1 \\
&\Leftrightarrow \langle \mathbf{T}_*, \mathbf{M}_{\mathcal{I}} \rangle \models_* \bar{r} \rightarrow (\forall x)(C(x) \rightarrow D(x)). \quad \square
\end{aligned}$$

Since graded axioms are interpreted as first order sentences, we have that a knowledge base is interpreted as a first-order theory. Accordingly, we say that an interpretation \mathbf{M} is a *model* of a knowledge base \mathcal{K} if it satisfies all the first-order sentences corresponding to axioms of \mathcal{K} . As an easy consequence of the previous proposition we have that reasoning tasks can be fairly translated to first-order problems with respect to a theory.

Observe that the first order formulas corresponding to axioms of knowledge bases have the form of the so-called *evaluated formulas* in the setting of Mathematical Fuzzy Logic. Indeed, the truth constants appear only in the antecedent of implications in the formulas corresponding to the axioms.

Finally, it is interesting to remark the relationship between description and modal logics. It is well known that there exists a translation between the classical description logic \mathcal{ALC} and the multi-modal logic \mathbf{K}_m (see [112]). It is worth to say that there exists a similar translation between the FDL language \mathcal{JALCE} and a multi-modal many-valued logic (see [54, 57] for an extensive account on this subject).

5 Fuzzy Description Logic languages with higher expressivity

In Section 3.3 we have defined several FDL languages, the most expressive of them being \mathcal{JALCE} . In classical DLs it is usual to consider even more expressive languages. In fact, the standard language for (crisp) ontology representation, OWL 2 [64], is based on the DL $\mathcal{SROIQ}(\mathbf{D})$, which is an extension of \mathcal{ALC} (recall that in the classical case, \mathcal{JALCE} and \mathcal{ALC} are equivalent). The purpose of this section is to discuss similar extensions of \mathcal{JALCE} in the fuzzy case.

FDLs described so far consider well-known logical connectives. However, the extension of more expressive DLs to the fuzzy case is not always unique. For instance, there could be different ways to extend the notion of cardinality in fuzzy logic. In this section, we will focus on the most common definitions in the FDL literature which often correspond to the most direct extensions [132].

The section is divided into five parts. The first four parts describe new concept constructors, role constructors, datatypes, and axioms, respectively. The fifth part summarizes some important expressive DL languages.

5.1 Fuzzy concept constructors

In this subsection we include fuzzy versions of the concept constructors corresponding to the DL $\mathcal{SROIQ}(\mathbf{D})$ together with some fuzzy concept constructors that do not have an equivalent in the classical case. More examples of new fuzzy concepts that do not have a crisp counterpart can be found in [23].

5.1.1 Nominals

The first extension that we will consider are *nominals*, which make it possible to give concept definitions by extension, using an enumeration of their instances. This constructor is denoted with the letter \mathcal{O} . Let a be an individual name and let $r \in T$. The syntax of concepts is extended with a new case:

$$C \rightsquigarrow \{r/a\} \quad (\text{nominal})$$

and the semantics is the following:

$$\{r/a\}^{\mathcal{I}}(x) = \begin{cases} r & \text{if } x = a^{\mathcal{I}} \\ 0 & \text{otherwise.} \end{cases}$$

The idea is that if x is interpreted as the same element of the domain than a , then the concept is evaluated as r . Otherwise, the concept is evaluated to 0. Although nominals can take an arbitrary truth degree r , in the literature they are usually restricted to the case where $r = 1$.

For example, the $\{1/\text{germany}\} \sqcup \{1/\text{austria}\} \sqcup \{0.67/\text{switzerland}\}$ denotes the fuzzy concept of German-speaking country, to which germany and austria belong with degree 1 and switzerland, with 4 official languages, belongs with degree 0.67.

5.1.2 Unqualified cardinality restrictions

Cardinality restrictions (also called number restrictions) describe the number of R -successors that an element of the domain can have, for a given role R . Using these constructors, we can refer to those elements that have *at most* or *at least* a given number of successors through the role R . Logics allowing these concept constructors are denoted with the letter \mathcal{N} .

To allow for unqualified cardinality restrictions, the syntax of the language is extended to include the constructors

$$\begin{aligned} C \rightsquigarrow \geq n R & \quad (\text{minimal cardinality restriction}) \\ C \rightsquigarrow \leq n R & \quad (\text{maximal cardinality restriction}), \end{aligned}$$

where $R \in N_R$ and $n \in \mathbb{N}$. These restrictions are called unqualified because we do not impose any other restriction on the objects of the relation. Following Example 3.1.1, ≥ 2 isFriendOf denotes the fuzzy set of people having at least 2 friends.

It is worth to note that the semantics of this concept constructor is particularly debatable, since there exist a lot of definitions of cardinality of fuzzy sets. We will consider here the following translations to first order logic (although alternative definitions are possible):

$$\begin{aligned} (\geq n R) & \mapsto (\exists y_1) \dots (\exists y_n) \left(R(x, y_1) \wedge \dots \wedge R(x, y_n) \wedge \bigwedge_{1 \leq i < j \leq n} y_i \neq y_j \right) \\ (\leq n R) & \mapsto (\forall y_1) \dots (\forall y_{n+1}) \left(R(x, y_1) \wedge \dots \wedge R(x, y_{n+1}) \rightarrow \bigvee_{1 \leq j < k \leq n+1} y_j = y_k \right). \end{aligned}$$

In fact, it is more common in the FDLs literature to interpret some of the conjunctions as strong conjunctions and not as weak ones. Recall that we are assuming that individual equality is a yes-no question, so the expressions $y_i \neq y_j$ and $y_j = y_k$ are evaluated in $\{0, 1\}$.

It is also possible to define *exact cardinality restrictions* ($= n R$) combining a maximal and a minimal cardinality restriction: $(\geq n R) \sqcap (\leq n R)$. Moreover, functional roles can also be expressed in any logic that allows for unqualified number restrictions. Indeed, the axiom $\langle \top \sqsubseteq (\leq 1 R), 1 \rangle$ restricts the role R to be interpreted as a functional role (see Section 5.4.2 for a definition of functional roles).

5.1.3 Qualified cardinality restrictions

Qualified cardinality restrictions generalized the restrictions from Section 5.1.2 to consider only those R -successors that are elements of a specific concept C . The use of these constructors is denoted with the letter \mathcal{Q} . Languages with qualified number restrictions allow in their syntax the constructors

$$C \rightsquigarrow \begin{array}{l} \geq n R.C \mid \quad (\text{minimal qualified cardinality restriction}) \\ \leq n R.C \quad (\text{maximal qualified cardinality restriction}). \end{array}$$

where $R \in N_R$, $n \in \mathbb{N}$, and C is a concept from this language. For example, the concept $\geq 2 R.Blond$ denotes the fuzzy set of people having at least 2 blond friends.

As in the previous case, an *exact qualified cardinality restriction* $= n R.C$ can be defined as $(\geq n R.C) \sqcap (\leq n R.C)$, and a possible semantics is obtained by translating these concepts to the following first order formulas:

$$\begin{aligned} (\geq n R.C) \mapsto & (\exists y_1) \dots (\exists y_n) (R(x, y_1) \wedge C(y_1) \wedge \dots \wedge R(x, y_n) \wedge C(y_n) \wedge \\ & \bigwedge_{1 \leq i < j \leq n} y_i \neq y_j) \\ (\leq n R.C) \mapsto & (\forall y_1) \dots (\forall y_{n+1}) (R(x, y_1) \wedge C(y_1) \wedge \dots \wedge R(x, y_{n+1}) \wedge C(y_{n+1}) \rightarrow \\ & \bigvee_{1 \leq j < k \leq n+1} y_j = y_k). \end{aligned}$$

Clearly, unqualified number restrictions are just special cases of qualified number restrictions; for instance, the concept $(\leq R)$ is equivalent to $(\leq R.\top)$.

Whenever we consider number restrictions, it is useful to allow for additional axioms for stating explicitly whether two individuals are equivalent (i.e., interpreted as the same element of the domain), or different. These axioms are introduced in Section 5.4.6.

5.1.4 Local reflexivity concepts

Local reflexivity concepts allow expressing that an individual is related to itself via some role R , without implying that the role is globally reflexive. To have them in the language, the syntax of concepts is extended as follows:

$$C \rightsquigarrow \exists R.Self \quad (\text{local reflexivity concept})$$

and the semantics is $(\exists R.Self)^{\mathcal{I}}(x) = R^{\mathcal{I}}(x, x)$. For example, $\exists likes.Self$ denotes narcissists, i.e., people who like themselves.

5.1.5 Fuzzy modified concepts

In fuzzy logic it is usually desirable to consider language hedges (or fuzzy modifiers) interpreting linguistic modifiers. Given a hedge $h: T \rightarrow T$, it is possible to apply it to change the membership of a fuzzy concept, thus obtaining a *fuzzy modified concept* [128]. Given an hedge h , the syntax of concepts is extended as follows:

$$C \rightsquigarrow h(C) \quad (\text{fuzzy modified concept})$$

and the semantics is $(h(C))^{\mathcal{I}}(x) = h(C^{\mathcal{I}}(x))$. For example, we can apply the hedge very to the concept Tall to define very(Tall), denoting the fuzzy set of very tall people.

These concepts do not have an equivalent in classical DLs, since in classical logics no grading is possible. A particular case of fuzzy modifier is the constructor Δ . Observe that the semantics is given by $(\Delta(C))^{\mathcal{I}}(x) = \delta(C^{\mathcal{I}}(x))$ (see Section 2.4).

5.1.6 Aggregated concepts

Fuzzy aggregation operators $@: T^n \rightarrow T$ are used to aggregate n values into a single one. In particular, they could be used to aggregate the evaluation of n fuzzy concepts. The application of an aggregation operator to several concepts produce an *aggregated concept* [77]. Examples of these aggregation operators are the weighted sum, the Ordered Weighted Averaging (OWA) operator or more general fuzzy integrals. Given an aggregation operator $@$, the syntax of concepts is extended as follows:

$$C_1, \dots, C_n \rightsquigarrow @(C_1, \dots, C_n) \quad (\text{aggregated concept})$$

and the semantics is $(@(C_1, \dots, C_n))^{\mathcal{I}}(x) = @(C_1^{\mathcal{I}}(x), \dots, C_n^{\mathcal{I}}(x))$. This concept does not have an equivalent in classical DLs.

For example, given the fuzzy concepts Tall and Blonde, we can define the fuzzy concept $WS_{[0.67, 0.33]}(\text{Tall}, \text{Blonde})$ which denotes the fuzzy set of tall blonde people, where the degree of being tall has double the importance than the degree of being blonde.

5.2 Fuzzy role constructors

In this subsection we include fuzzy versions of the role constructors in the DL $\mathcal{SROIQ}(\mathbf{D})$ together with some fuzzy roles without an equivalent in the classical case.

5.2.1 Inverse roles

The inverse relations of the roles in the language are called *inverse roles*. This constructor is denoted by appending \mathcal{I} to the language name. Up to now, we assumed that every role is atomic, that is, a member of N_R . To support inverse roles, the syntax need to be extended to allow roles of the form

$$R \rightsquigarrow R^- \quad (\text{inverse role}),$$

where R is an atomic role. For example, the roles hasParent and hasChild are inverse. To interpret these complex roles, we extend the interpretation function to define $(R^-)^{\mathcal{I}}(x, y) = R^{\mathcal{I}}(y, x)$. Note that $(R^-)^-$ is equivalent to R , and thus we can restrict without loss of generality to inverse of atomic roles.

5.2.2 Universal role

The *universal role* is similar to the top concept. It is denoted with the letter U and is interpreted as $U^{\mathcal{I}}(x, y) = 1$ for all elements x, y in the domain. Its use is often restricted, depending on the particular DL language (see [88] for details).

5.2.3 Fuzzy modified roles

Similarly to the case of fuzzy modified concepts, given an hedge h , the syntax of roles can be extended as follows:

$$R \rightsquigarrow h(R) \quad (\text{fuzzy modified role}).$$

The semantics is $(h(R))^{\mathcal{I}}(x, y) = h(R^{\mathcal{I}}(x, y))$. For example, we can apply the hedge *very* to the role *isFriendOf* to define *very(isFriendOf)*, denoting the fuzzy relation of very good friends. This role constructor does not have an equivalent in classical DLs.

5.3 Fuzzy datatypes

Up to now, the object of every relation is an individual of the world. The idea behind datatypes, also known as concrete domains, is to allow individuals having relations (e.g. “to have age”) with certain kinds of data (e.g. the number “47”) that do not belong to the represented domain, rather to a different domain that is already structured and whose structure is already known to the machine (for this reason it is “concrete”). These kinds of individuals, called *data values* can be numerical or textual, among many other possibilities. Clearly data values could be introduced as usual individuals with their attributes, but since their structure is already known for the machine (e.g. the machine already knows that $3 < 5$), in this way there is no need of introducing an ontology to fix the structure of datatypes. Possible datatypes include numbers (real, rational, integer, non-negative, etc.), strings, booleans, dates, times, or XML literals.

In the fuzzy case, in addition to relating an individual with some data value dv , it is possible to relate it with a fuzzy membership function d . Typical examples of fuzzy datatypes, when the datatype domain is a dense total order, are the trapezoidal, triangular, left-shoulder or right-shoulder membership functions. For example, the datatype *hasAgeAround*(n) can be defined using a triangular membership function with parameters $(\max\{0, n - 3\}, n, n + 3)$. This constructor is denoted by appending **(D)** to the name of the language, e.g. $\mathcal{ALC}(\mathbf{D})$.

Datatypes can be used in complex concepts that use a role, namely in existential quantification, value restriction and, if allowed in the language, qualified cardinality restrictions. Hence, the syntax of the concept is extended as follows:

$$\begin{aligned} C \rightsquigarrow \quad & \forall R.d \mid && (\text{data value restriction}) \\ & \exists R.d \mid && (\text{data existential quantification}) \\ & \geq n R.d \mid && (\text{minimal qualified data cardinality restriction}) \\ & \leq n R.d && (\text{maximal qualified data cardinality restriction}). \end{aligned}$$

The semantics are similar to those of the respective cases $\forall R.C$, $\exists R.C$, $\geq n R.C$ and $\leq n R.C$ but replacing the interpretation (instantiation) of C on an element $y \in M$ with the interpretation (instantiation) of d on a data value dv .

5.4 New axioms

We will consider in this subsection fuzzy versions of the axioms in the description logic $\mathcal{SROIQ}(\mathbf{D})$.

5.4.1 Transitive roles

The first extension regarding axioms that we are going to explore is the introduction of *transitive roles* in the language. For example, the role `isSimilarTo` can be considered as transitive. This element can be denoted by adding \mathcal{R}_+ to the language name. For example, the extension of \mathcal{ALC} with transitive roles is denoted as \mathcal{ALCR}_+ . In classical DLs, \mathcal{ALCR}_+ is usually shortened to \mathcal{S} due to its equivalence with the modal logic $\mathbf{S4}$. In FDLs, we prefer to use \mathcal{S} to denote the extension of \mathcal{JALCE} with transitive roles.

In order to have transitive roles in the language, one must extend the syntax of the language to have another type of axioms of the form $\text{trans}(R)$, where R is a role. Its corresponding first order sentence is

$$(\forall x)(\forall y)(\forall z)(R(x, z) \& R(z, y) \rightarrow R(x, y)).$$

Since the first order sentence should take value 1, an interpretation \mathcal{I} satisfies an axiom $\text{trans}(R)$ if and only if it verifies that for every $x, y \in M$,

$$R^{\mathcal{I}}(x, y) \geq \sup_{z \in M} R^{\mathcal{I}}(x, z) * R^{\mathcal{I}}(z, y).$$

5.4.2 Functional roles

In classical DLs, a role R is called *functional* if any object of the domain can have at most one successor via the relation R . The use of this constructor in a DL is denoted with the letter \mathcal{F} . In the fuzzy case, we can assert that the concrete role `livesInMainResidence` is functional, because everybody has a unique main residence place, but this relation is graded to differentiate people who live always in their main residence from people who spend a lot of time out of their home.

Syntactically, functional roles are just a special kind of role. When defining an FDL that allows for functional roles, we simply assume that the set N_R of role names is partitioned into two sets N_{FR} and N_{GR} of *functional roles* and *general roles*, respectively. Semantically, interpretations are required to treat the roles $R \in N_{FR}$ as fuzzy partial functions. While there are several possible ways in which partial functionality can be defined in the fuzzy setting, the most usual in the FDL literature is that an interpretation must satisfy the first order formula

$$(\forall y_1)(\forall y_2)((R(x, y_1) \wedge R(x, y_2)) \rightarrow (y_1 = y_2)),$$

for all functional roles $R \in N_{FR}$. Recall that $y_1 = y_2$ is evaluated in $\{0, 1\}$.

5.4.3 Role inclusions

It is also natural to consider role inclusions (or role hierarchies) as part of the language in a similar way as it contains concept inclusions. For instance, the role `hasSon` is more specific than the role `hasChild`. This is denoted by appending \mathcal{H} (as the initial letter of hierarchies) to the language name. In the fuzzy case, these inclusions can be graded, of course.

Let R, S be two roles and let $r \in T$. A *graded role inclusion axiom* is an expression of one of the form

$$\langle R \sqsubseteq S, r \rangle,$$

whose corresponding first order sentence is

$$\bar{r} \rightarrow (\forall x)(\forall y)(R(x, y) \rightarrow S(x, y)).$$

5.4.4 More axioms involving roles

Traditionally, DLs have paid much more attention to the expressivity of concepts than to that of roles. Recently, several new axioms involving roles (and 5 other types of axioms) have been proposed [88]. These new axioms can be extended to the fuzzy case. Let R, S be roles and a, b be individuals. The syntax of axioms is extended as follows:

$\langle \neg R(a, b), r \rangle$		(weakly complemented role assertion)
$\langle \sim R(a, b), r \rangle$		(strongly complemented role assertion)
$\langle R_1 \dots R_m \sqsubseteq S, r \rangle$		(role inclusion axiom)
$\text{dis}(R, S)$		(disjoint roles)
$\text{ref}(R)$		(reflexive role).

Let us consider some examples. The assertion $\langle \neg \text{isFriendOf}(\text{alice}, \text{bob}), 0.5 \rangle$ states that alice and bob cannot be considered friends with degree greater or equal than 0.5. The axiom $\langle \text{hasMotherhasSister} \sqsubseteq \text{hasAunt}, 1 \rangle$ states the sister of one's mother is his/her aunt. The roles hasMother and hasFather are disjoint. Finally, the fuzzy role isCloseTo is reflexive.

The corresponding first order sentences are the following ones:

$$\begin{aligned} \langle \neg R(a, b), r \rangle &\rightsquigarrow r \rightarrow \neg R(a, b) \\ \langle \sim R(a, b), r \rangle &\rightsquigarrow r \rightarrow \sim R(a, b) \\ \langle R_1 \dots R_m \sqsubseteq S, r \rangle &\rightsquigarrow r \rightarrow (\forall x_1) \dots (\forall x_{m+1}) ((\exists x_2) \dots (\exists x_m) \\ &\quad (R_1(x_1, x_2) \& \dots \& R_m(x_m, x_{m+1})) \rightarrow S(x_1, x_{m+1})) \\ \text{dis}(R, S) &\rightsquigarrow (\forall x)(\forall y)(R(x, y) \wedge S(x, y) \rightarrow 0) \\ \text{ref}(R) &\rightsquigarrow \bar{1} \rightarrow (\forall x)R(x, x). \end{aligned}$$

It is worth noticing that role inclusion axioms are strictly more expressive than role hierarchies. Moreover, irreflexive, symmetric, asymmetric and transitive roles can be simulated using the constructors and axioms introduced in this section. Let R be a role, then R is

<i>irreflexive</i>	if	$\langle \top \sqsubseteq \neg \exists R.\text{Self}, 1 \rangle$
<i>symmetric</i>	if	$\langle R \sqsubseteq R^-, 1 \rangle$
<i>asymmetric</i>	if	$\text{dis}(R, R^-)$
<i>transitive</i>	if	$\langle RR \sqsubseteq R, 1 \rangle$.

There are some additional syntactic restrictions to guarantee decidability. The interested reader is referred to [26, 132].

5.4.5 Crispification axioms

In some applications, it is important to restrict some concepts and roles to be *crisp*; that is, interpreted in such a way that all elements belong to them to a degree in $\{0, 1\}$. We can achieve this by including axioms of the form $\text{crisp}(C)$ and $\text{crisp}(R)$ where C and R are (possibly complex) concepts and roles. An interpretation \mathcal{I} satisfies such axioms if $(\text{crisp}(C))^{\mathcal{I}}(x) \in \{0, 1\}$ holds for all x , and $(\text{crisp}(R))^{\mathcal{I}}(x, y) \in \{0, 1\}$ holds for all x, y , respectively.

Notice that in FDL languages containing Δ these axioms can be formulated as $\Delta(C) = C$ and $\Delta(R) = R$ respectively.

5.4.6 Individual restrictions

It is sometimes convenient to express that two individual names correspond to the same element of the domain, or that they must be interpreted as different elements. This can be achieved by allowing *individual restrictions* of the form $a = b$ and $a \neq b$, where a, b are two individual names. The semantics of these axioms is the obvious one: the interpretation \mathcal{I} satisfies $a = b$ if $a^{\mathcal{I}} = b^{\mathcal{I}}$; it satisfies $a \neq b$ if $a^{\mathcal{I}} \neq b^{\mathcal{I}}$.

5.5 Some important languages

To conclude this section, we will highlight some important languages whose classical versions are important from a practical point of view.

- The classical language $\mathcal{SHOIN}(\mathbf{D})$ is the logical counterpart behind the *Web Ontology Language* (OWL). $\mathcal{SHOIN}(\mathbf{D})$ is the extension of \mathcal{ALC} with transitive roles, role hierarchies, nominals, inverse roles, cardinality restrictions, and datatypes. OWL was the standard ontology language until December 2012, when it was replaced by its successor OWL 2.
- Crisp $\mathcal{SROIQ}(\mathbf{D})$ is equivalent to the language OWL 2, the current standard language for (crisp) ontology representation. $\mathcal{SROIQ}(\mathbf{D})$ is the extension of $\mathcal{SHOIN}(\mathbf{D})$ with local reflexivity concepts, qualified cardinality restrictions, the universal role, complemented role assertions, role inclusion axioms, disjoint roles and reflexive roles.

The standardisation of OWL and OWL 2 has given rise to a massive increase of the development of crisp ontologies and tools,⁷ such as reasoners, editors or APIs. In the fuzzy case, no standard language has been defined yet.

6 Reasoning algorithms

We start this section by discussing the equivalences between different reasoning tasks. This way, it is sufficient to provide a reasoning algorithm for only one of them. Then we provide an overview of the most important existing reasoning algorithms.

6.1 Equivalences between reasoning tasks in finitely-valued FDLs

In classical DLs, it is possible to show the inter-definability of the different reasoning tasks. In FDLs, this is not always the case, as depending on the fuzzy operators

⁷ See http://wiki.opensemanticframework.org/index.php/Ontology_Tools for examples.

some of the reductions may not hold. In this subsection, we will restrict our attention to FDLs with the finitely-valued property and show the reduction of every reasoning task to $*$ -consistency. To achieve this, we will assume that the language contains $*$, \sim , and \rightarrow .

Assuming a finite number of degrees makes it possible to reduce some tasks to a binary search over the degrees and consider the predecessor of every degree of truth. Furthermore, infinitely-valued DLs with GCIs are undecidable if $*$ is different from Gödel t-norm, as we will see in Section 7.

For every $r \in T, r > 0$, we use $\text{pred}(r)$ to denote its *predecessor*, and then we define $r^- = \text{pred}(r)$. This way, $\langle \sim C(a), r^- \rangle$ requires that the value of $C(a)$ is strictly smaller than r . Now, the following result can easily be shown:

THEOREM 6.1.1. *Let \mathcal{K} be a knowledge base and $r \neq 0$. In \mathfrak{JALCE} , the following equivalences between reasoning tasks hold:*

- C is $*$ -satisfiable to a degree greater than or equal to r with respect to \mathcal{K} if and only if $\mathcal{K} \cup \{\langle C(a), r \rangle\}$ is $*$ -consistent, where a is an individual not appearing in \mathcal{K} .
- C is $*$ -subsumed by D to a degree greater than or equal to r with respect to \mathcal{K} if and only if $\mathcal{K} \cup \{\langle (\sim(C \rightarrow D))(a), r^- \rangle\}$ is not $*$ -consistent, where a is an individual not appearing in \mathcal{K} .
- \mathcal{K} entails:
 - a) A graded concept assertion $\langle C(a), r \rangle$ if and only if $\mathcal{K} \cup \{\langle (\sim C)(a), r^- \rangle\}$ is not $*$ -consistent.
 - b) A graded role assertion $\langle R(a, b), r \rangle$ if and only if

$$\mathcal{K} \cup \{\langle (\sim(\exists R.B))(a), r^- \rangle, \langle B(b), 1 \rangle\}$$
 is not $*$ -consistent, where B is an atomic concept not appearing in \mathcal{K} .
 - c) A graded concept inclusion $\langle C \sqsubseteq D, r \rangle$ if and only if

$$\mathcal{K} \cup \{\langle (\sim(C \rightarrow D))(a), r^- \rangle\}$$
 is not $*$ -consistent, where a is an individual not appearing in \mathcal{K} .
- The BED of an axiom τ can be computed as a binary search over the degrees in the finite chain of the maximum value $r \in T$ such that \mathcal{K} entails $\langle \tau, r \rangle$, and entailment can be reduced to $*$ -consistency.
- The BSD of a concept C can be computed as a binary search over the degrees in the finite chain of the maximum value $r \in T$ such that $\mathcal{K} \cup \{\langle C(a), r \rangle\}$ is $*$ -consistent, where a is an individual not appearing in \mathcal{K} .

Note the implicit assumptions of having existential restriction, concept implication, and involutive negation in the language. It is also worth to note that the BED and the BSD problems can be computed in a more efficient way by developing a specific reasoning algorithm, such as those described in Section 6.2.2, that requires to solve one single test instead of several ones.

6.2 Types of algorithms

The literature includes a substantial amount of work on proof methods for first order fuzzy logics. The interested reader is referred to [97] and, for the particularly interesting case of finite fuzzy logics, to [13]. Many specific algorithms to reason with FDLs have been proposed in the literature. They can be classified in the following categories:

- *Tableaux algorithms*, extending the tableaux algorithms for classical DLs to the fuzzy case.
- *Tableaux algorithms and optimization problems*, using a tableaux algorithm to reduce the reasoning to an optimization problem.
- *Automata-based algorithms*, adopting ideas similar to those used to prove complexity results in the classical case.
- *Reduction to classical DLs*, for which existing reasoning algorithms are known.
- *Reduction to propositional fuzzy logics*, for which reasoning has been studied.
- *Structural subsumption algorithms*, extending the structural subsumption algorithms for classical DLs with low expressivity to the fuzzy case.

In the rest of this section, we give a deeper look into each of these families of algorithms, discussing their advantages and limitations, and one salient example of algorithm of the family. For simplicity, we will consider algorithms for relatively simple logics. The most important results, involving more expressive logics, can be found in Section 8.

6.2.1 Tableau-based algorithms

Tableaux algorithms are the most popular solution to decide satisfiability or consistency in classical DLs [12]. Although these algorithms usually have a sub-optimal worst-case complexity, they typically behave well in practice.

Given a KB, tableau-based algorithms try to build a data structure (called a *tableau*) representing a finite description of a model of the KB and from which it is immediate to build a complete model. This structure is typically a graph where the nodes represent individuals of the domain (annotated with a list of concepts that the individual belongs to) and the arcs represent roles establishing relations between two individuals.

The graph is initialized using the assertions that appear in the KB. Then, tableau-expansion rules are used to decompose complex concepts into simpler ones. For instance, the conjunction rule decomposes a concept $C \sqcap D$ into both C and D . Some of the rules are non-deterministic, making it sometimes necessary to try several possibilities. For instance, the disjunction rule decompose a concept $C \sqcup D$ by producing two possible tableaux, one where C is added to the node and another one where D is added. In most DLs, due to the presence of cycles in the axioms, it is also necessary to use a *blocking* mechanism to guarantee termination of the approach.

After all the possible rules have fired, the tableau is completely expanded and it only remains to check for obvious contradictions. For instance, if a node belongs to both a concept and its complement, then the result of the algorithm outputs as an answer that the input KB is inconsistent. Otherwise, the algorithm yields a positive answer to the consistency of the input KB.

Algorithm 1 Consistency checking in $Z\text{-}\mathcal{ALC}$ without a TBox**Require:** A fuzzy KB \mathcal{K} **Ensure:** true if \mathcal{K} is *-consistent; false otherwise

```

1: for each concept assertion  $\langle C(a), r \rangle \in \mathcal{K}$  do
2:    $\mathcal{K} \leftarrow (\mathcal{K} \setminus \{\langle C(a), r \rangle\}) \cup \{\langle \text{nnf}(C)(a), r \rangle\}$  {Negation Normal Form}
3: end for
   {Initialization of the forest}
4: create an empty forest  $\mathcal{F}$ 
5: for each concept assertion  $\langle C(a), r \rangle \in \mathcal{K}$  do
6:   create a node  $v_a$  in  $\mathcal{F}$  if it does not exist
7:    $\mathcal{L}(v_a) \leftarrow \mathcal{L}(v_a) \cup \{\langle C \geq r \rangle\}$ 
8: end for
9: for each role assertion  $\langle R(a, b), r \rangle \in \mathcal{K}$  do
10:  create nodes  $v_a, v_b$  in  $\mathcal{F}$  if they do not exist
11:  create an edge  $\langle v_a, v_b \rangle$  in  $\mathcal{F}$  if it does not exist
12:   $\mathcal{L}(\langle v_a, v_b \rangle) = \mathcal{L}(\langle v_a, v_b \rangle) \cup \{\langle R \geq r \rangle\}$ 
13: end for
   {Tableaux rules}
14: for each rule application to  $\mathcal{F}$  do
15:    $\mathcal{F}' \leftarrow \mathcal{F}$ 
   {build a completion graph of  $\mathcal{F}'$ }
16:   while some tableaux rule is applicable to some node  $v$  in  $\mathcal{F}'$  do
17:     apply one of the rules in Table 3 to  $v$ 
18:     mark the applied rule as not applicable to node  $v$ 
   {Stop when a clash is found }
19:     if there is not some node of  $\mathcal{F}'$  with a clash then
20:       continue with the next iteration of the for loop
21:     end if
22:   end while
   {There is a clash-free completion graph }
23:   if there is not any node in  $\mathcal{F}'$  with a clash then
24:     return true
25:   end if
26: end for
   {There are no clash-free completion graphs }
27: return false

```

It is sometimes possible to extend these algorithms to work with FDLs. The main idea is that every node is annotated with restrictions of the form $\langle C \geq r \rangle$ or $\langle C \leq r \rangle$, where C is a concept and $r \in T$, stating that the individual nodes belong to the concept C at least with degree r . The rules are updated to take into account the semantics of the concepts in the fuzzy case. For instance, $\langle C \wedge D \geq r \rangle$ is decomposed into both $\langle C \geq r \rangle$ and $\langle D \geq r \rangle$. The notion of contradiction is updated as well since now it is usually possible that a node belongs to some degree to both a concept and its complement.

	preconditions	actions
(\neg_+)	$\langle \neg C \geq t_i \rangle \in \mathcal{L}(v)$	$\mathcal{L}(v) = \mathcal{L}(v) \cup \{\langle C \leq t_{n-i} \rangle\}$
(\neg_-)	$\langle \neg C \leq t_i \rangle \in \mathcal{L}(v)$	$\mathcal{L}(v) = \mathcal{L}(v) \cup \{\langle C \geq t_{n-i} \rangle\}$
(\sqcap_+)	$\langle C \sqcap D \geq t_i \rangle \in \mathcal{L}(v)$	$\mathcal{L}(v) = \mathcal{L}(v) \cup \{\langle C \geq t_i \rangle, \langle D \geq t_i \rangle\}$
(\sqcap_-)	$\langle C \sqcap D \leq t_i \rangle \in \mathcal{L}(v)$	$\mathcal{L}(v) = \mathcal{L}(v) \cup \{X\}$ for some $X \in \{\langle C \leq t_i \rangle, \langle D \leq t_i \rangle\}$
(\sqcup_+)	$\langle C \sqcup D \geq t_i \rangle \in \mathcal{L}(v)$	$\mathcal{L}(v) = \mathcal{L}(v) \cup \{X\}$ for some $X \in \{\langle C \geq t_i \rangle, \langle D \geq t_i \rangle\}$
(\sqcup_-)	$\langle C \sqcup D \leq t_i \rangle \in \mathcal{L}(v)$	$\mathcal{L}(v) = \mathcal{L}(v) \cup \{\langle C \leq t_i \rangle, \langle D \leq t_i \rangle\}$
(\exists_+)	$\langle \exists R.C \geq t_i \rangle \in \mathcal{L}(v)$ $\exists y$ with $\langle R \geq t_j \rangle \in \mathcal{L}(\langle v, y \rangle)$ $\langle C \geq t_k \rangle \in \mathcal{L}(y), j, k \geq i$	create a new node w $\mathcal{L}(\langle v, w \rangle) = \{\langle R \geq t_j \rangle\}$ $\mathcal{L}(w) = \{\langle C \geq t_i \rangle\}$
(\exists_-)	$\langle \exists R.C \leq t_i \rangle \in \mathcal{L}(v)$ $\langle R \geq t_j \rangle \in \mathcal{L}(\langle v, w \rangle), j > i$	$\mathcal{L}(w) = \mathcal{L}(w) \cup \{\langle C \leq t_i \rangle\}$
(\forall_+)	$\langle \forall R.C \geq t_i \rangle \in \mathcal{L}(v)$ $\langle R \geq t_j \rangle \in \mathcal{L}(\langle v, w \rangle), j > n - i$	$\mathcal{L}(w) = \mathcal{L}(w) \cup \{\langle C \geq t_i \rangle\}$
(\forall_-)	$\langle \forall R.C \leq t_i \rangle \in \mathcal{L}(v)$ $\exists y$ with $\langle R \geq t_{n-j} \rangle \in \mathcal{L}(\langle v, y \rangle)$ $\langle C \leq t_k \rangle \in \mathcal{L}(y), j, k \leq i$	create a new node w $\mathcal{L}(\langle v, w \rangle) = \{\langle R \geq t_{n-i} \rangle\}$ $\mathcal{L}(w) = \{\langle C \geq t_i \rangle\}$

Table 3. Rules of the tableaux algorithm

In the case of FDLs the obtained tableau has a contradiction if a node has two pairs $\langle C \geq r_1 \rangle$ and $\langle C \leq r_2 \rangle$ with $r_1 < r_2$.

This family of algorithms can be used to decide *-consistency in Zadeh FDLs or if the degrees of truth is taken from a residuated De Morgan lattice, as we will describe below. Indeed, in all these FDLs the weak and the strong conjunction coincide, and then its semantics is the minimum and thus $\langle C \wedge D \geq r \rangle$ can be decomposed into both $\langle C \geq r \rangle$ and $\langle D \geq r \rangle$. In more general cases, tableaux algorithms are combined with an optimization problem, as we will see in the next subsection.

Let us see in detail Algorithm 1 to decide the consistency of the fuzzy knowledge base \mathcal{K} in $Z\text{-}\mathcal{ALC}$ with an empty TBox. For simplicity, we will restrict to a chain $T = \{t_0, t_1, \dots, t_n\}$ of degrees of truth, but the extension to $[0, 1]$ is easy [125].

Like most of the tableaux algorithms, the algorithm works on *completion-forests* since an ABox might contain several individuals with arbitrary roles connecting them. A completion-forest \mathcal{F} for \mathcal{K} is a collection of nodes arbitrarily connected by edges.

- Each node v is labeled with a set $\mathcal{L}(v)$ of expressions of the form $\langle C \geq t_i \rangle$ and $\langle D \leq t_j \rangle$, meaning that v is an instance of C to degree greater than or equal to t_i and an instance of D to degree less or equal than t_j .
- Each edge $\langle v, w \rangle$ is labeled with a set $\mathcal{L}(\langle v, w \rangle)$ of expressions of the form $\langle R \geq t_i \rangle$ and $\langle R \leq t_j \rangle$.

\mathcal{F} is then expanded by repeatedly applying the tableaux rules in Table 3. Note that there are two non-deterministic rules, namely (\sqcup_+) and (\sqcap_-) . In classical DLs, the conjunction does not generate a non-deterministic rule. Similarly, the rules (\exists_+) and (\forall_-) generate new individuals; while in classical DLs only existential restrictions do.

If the rules can be applied in such a way that they yield a complete and clash-free completion-forest, \mathcal{K} is $*$ -consistent. Otherwise, \mathcal{K} is not $*$ -consistent. Since there are non-deterministic rules, there are many possible ways to apply the rules, so the algorithm must consider each possible rule application until a clash-free completion-forest is found. This can be computed by using a search with backtracking. A node v contains a clash in any of the following situations:

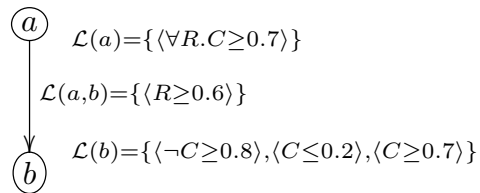
- $\langle \perp \geq t_i \rangle \in \mathcal{L}(v), i > 0,$
- $\langle \top \leq t_i \rangle \in \mathcal{L}(v), i < n,$ or
- $\langle C \geq t_i \rangle \in \mathcal{L}(v), \langle C \leq t_j \rangle \in \mathcal{L}(v), j < i.$

Under Zadeh fuzzy logic, we can assume w.l.o.g. that concept are in *Negation Normal Form* (NNF) where the negation only appears before an atomic concept. In fact, a fuzzy concept $\neg C$ can be transformed into NNF by recursively applying the function nnf defined as follows: $\text{nnf}(\neg \top) = \perp$, $\text{nnf}(\neg \perp) = \top$, $\text{nnf}(\neg A) = \neg A$, $\text{nnf}(\neg \neg C) = C$, $\text{nnf}(\neg(C \sqcap D)) = \neg C \sqcup \neg D$, $\text{nnf}(\neg(C \sqcup D)) = \neg C \sqcap \neg D$, $\text{nnf}(\neg \exists R.C) = \forall R.\neg C$, and $\text{nnf}(\neg \forall R.C) = \exists R.\neg C$.

EXAMPLE 6.2.1. Consider the chain of 11 degrees of truth and the fuzzy KB

$$\{ \langle (\forall R.C)(a), 0.7 \rangle, \langle R(a, b), 0.6 \rangle, \langle (\neg C)(b), 0.8 \rangle \}.$$

The completion graph after the initialization and the application of the rules (\neg) and (\forall) can be depicted as:



It can be seen that the KB is $*$ -inconsistent because there is a clash: node b contains both $\langle C \leq 0.2 \rangle$ and $\langle C \geq 0.7 \rangle$.

6.2.2 Tableaux algorithms and optimization problems

In general, the tableaux algorithms already described are not appropriate for other t -norms different from Gödel or to manage fuzzy datatypes or fuzzy concepts without a counterpart in crisp DLs, such as fuzzy modified concepts or aggregated concepts. Intuitively, given $\langle C \sqcap D \geq r \rangle$ we cannot infer both $\langle C \geq r \rangle$ and $\langle D \geq r \rangle$, but only $\langle C \geq r_1 \rangle$ and $\langle D \geq r_2 \rangle$ for $r_1, r_2 \in T$ such that $r_1 * r_2 \geq r$. Hence, the tableau rules not only decompose complex concept expressions into simpler ones but

also generate a system of constraints. For example, in the case of the Łukasiewicz t-norm, the restriction $r_1 *_{\mathbb{L}} r_2 \geq r$ can be encoded using the set of linear constraints $\{y \leq 1 - r, r_1 + r_2 - 1 \geq r - y, y \in \{0, 1\}\}$. In this case, the tableau rules are deterministic and only one optimization problem is obtained. Indeed, in the previous example the two possibilities $y = 0$ and $y = 1$ encode the non-deterministic choice.

After the tableau is completely expanded, an optimization problem must be solved before obtaining the final solution. The optimization problem has a solution iff the fuzzy KB is consistent. Note that we are using the reduction from entailment to inconsistency provided by Theorem 6.1.1. In general, the algorithm produces a bounded Mixed Integer Non Linear Programming (bMINLP) optimization problem. In some particular cases, an easier problem is obtained, such as a bounded Mixed Integer Linear Programming (bMILP) problem in Łukasiewicz DLs, or a bounded Mixed Integer Quadratically Constrained Programming (MIQCP) problem in Product DLs.

Now we discuss in detail Algorithm 2 to compute the BED of $C(a)$ with respect to \mathcal{K} in $\mathbb{L}\text{-ALC}$ with an empty TBox [60]. In order to determine the BED problem of $C(a)$ with respect to \mathcal{K} , we consider an expression of the form $\langle(-C)(a), 1 - x\rangle$ (informally, $\langle C(a) \leq x\rangle$), where x is a $[0, 1]$ -valued variable. Then we construct a tableaux for $\mathcal{K}' = \mathcal{K} \cup \{\langle(-C)(a), 1 - x\rangle\}$ in which the application of satisfiability preserving rules generates new fuzzy assertion axioms together with *inequations* over $[0, 1]$ -valued variables. These inequations have to hold in order to respect the semantics of the DL constructors. Hence, we *minimize* the original variable x such that all constraints are satisfied.

Like most of the tableaux algorithms, our algorithm works on *completion-forests* since an ABox might contain several individuals with arbitrary roles connecting them.

A completion-forest \mathcal{F} for \mathcal{K} is a collection of trees whose distinguished roots are arbitrarily connected by edges.

- Each node v is labeled with a set $\mathcal{L}(v)$ of concepts C . If $C \in \mathcal{L}(v)$ then we consider a variable $x_{C(v)}$. The intuition is that v is an instance of C to degree equal or greater than the value of the variable $x_{C(v)}$.
- Each edge $\langle v, w \rangle$ is labeled with a set $\mathcal{L}(\langle v, w \rangle)$ of roles R and if $R \in \mathcal{L}(\langle v, w \rangle)$ then we consider a variable $x_{R(v,w)}$ representing the degree of being $\langle v, w \rangle$ and instance of R .

The forest has associated a set $\mathcal{C}_{\mathcal{F}}$ of constraints of the form $l \leq l'$, $l = l'$, $x_i \in [0, 1]$, $y_i \in \{0, 1\}$, where l, l' are linear expressions using the variables occurring in the forest. \mathcal{F} is then expanded by repeatedly applying the tableaux rules in Table 4. We note that $x_1 \rightarrow x_2 \geq z$ can be encoded as $1 - x_1 \oplus x_2 \geq z$ and that $x_1 \otimes x_2 \geq z$, $x_1 \otimes x_2 \leq z$ and $x_1 \oplus x_2 \geq z$, $(x_1, x_2, z \in [0, 1])$ can be encoded in MILP as follows:

- $x_1 \otimes x_2 \geq z \mapsto \{y \leq 1 - z, x_1 + x_2 - 1 \geq z - y, y \in \{0, 1\}\}$, where y is a new variable.
- $x_1 \otimes x_2 \leq z \mapsto \{x_1 + x_2 - 1 \leq z\}$.
- $x_1 \oplus x_2 \geq z \mapsto \{x_1 + x_2 \geq z\}$.

Algorithm 2 Computation of the BED of $C(a)$ in \mathcal{LALC} without a TBox**Require:** A concept assertion $C(a)$, a fuzzy KB \mathcal{K} **Ensure:** BED of $C(a)$ with respect to \mathcal{K}

```

1:  $\mathcal{K} \leftarrow \mathcal{K} \cup \{\langle \text{nnf}(\neg C)(a), 1 - x \rangle\}$ 
   {Negation Normal Form}
2: for each concept assertion  $\langle C(a), r \rangle \in \mathcal{K}$  do
3:    $\mathcal{K} \leftarrow (\mathcal{K} \setminus \{\langle C(a), r \rangle\}) \cup \{\langle \text{nnf}(C)(a), r \rangle\}$ 
4: end for
   {Initialization of the forest}
5: create an empty forest  $\mathcal{F}$ 
6:  $\mathcal{C}_{\mathcal{F}} \leftarrow \emptyset$ 
7: for each concept assertion  $\langle C(a), r \rangle \in \mathcal{K}$  do
8:   create a node  $v_a$  in  $\mathcal{F}$  if it does not exist
9:    $\mathcal{L}(v_a) \leftarrow \mathcal{L}(v_a) \cup \{C\}$ 
10:   $\mathcal{C}_{\mathcal{F}} \leftarrow \mathcal{C}_{\mathcal{F}} \cup \{x_{C(v_a)} \geq r\}$ 
11: end for
12: for each role assertion  $\langle R(a, b), r \rangle \in \mathcal{K}$  do
13:   create nodes  $v_a, v_b$  in  $\mathcal{F}$  if they do not exist
14:   create an edge  $\langle v_a, v_b \rangle$  in  $\mathcal{F}$  if it does not exist
15:    $\mathcal{L}(\langle v_a, v_b \rangle) \leftarrow \mathcal{L}(\langle v_a, v_b \rangle) \cup \{R\}$ 
16:    $\mathcal{C}_{\mathcal{F}} \leftarrow \mathcal{C}_{\mathcal{F}} \cup \{x_{R(v_a, v_b)} \geq r\}$ 
17: end for
   {Tableaux rules}
18: while some tableaux rule is applicable to some node in  $\mathcal{F}$  do
19:   apply one of the rules in Table 4 to a node  $v$ 
20:   mark the applied rule as not applicable to node  $v$ 
21: end while
   {Solve the optimization problem}
22: for each variable  $x$  of the forms  $x_{C(v)}$  or  $x_{R(v, w)} \in \mathcal{K}$  do
23:    $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x \in [0, 1]\}$ 
24: end for
25: if  $\mathcal{C}_{\mathcal{F}}$  has a solution then
26:   solve the optimization problem
27:   return  $x$ 
28: else
29:   return 1 { $\mathcal{K}$  is inconsistent}
30: end if

```

Taking into account that the residuated negation in Łukasiewicz is involutive, we can use the recursive definition of nnf to transform any concept into Negation Normal Form, where the negation only appears before an atomic concept (see pag. 1140).

Another important observation is that since there is not a TBox, there is a model of \mathcal{K} iff there is a witnessed model of \mathcal{K} [80].

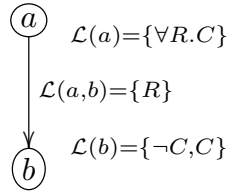
	preconditions	actions
(\perp)	$\perp \in \mathcal{L}(v)$	$\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:\perp} = 0\}$
(\top)	$\top \in \mathcal{L}(v)$	$\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:\top} = 1\}$
(\neg)	$\neg A \in \mathcal{L}(v)$	$\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:\neg C} = 1 - x_{v:C}\}$
(\sqcap)	$C \sqcap D \in \mathcal{L}(v)$	$\mathcal{L}(v) = \mathcal{L}(v) \cup \{C, D\}$ $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:C} \otimes x_{v:D} \geq x_{v:C \sqcap D}\}$
(\sqcup)	$C \sqcup D \in \mathcal{L}(v)$	$\mathcal{L}(v) = \mathcal{L}(v) \cup \{C, D\}$ $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:C} \oplus x_{v:D} \geq x_{v:C \sqcup D}\}$
(\exists)	$\exists R.C \in \mathcal{L}(v)$	create a new node w $\mathcal{L}(\langle v, w \rangle) = \mathcal{L}(\langle v, w \rangle) \cup \{R\}$, and $\mathcal{L}(w) = \mathcal{L}(w) \cup \{C\}$, and $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{(v,w):R} \otimes x_{w:C} \geq x_{v:\exists R.C}\}$
(\forall)	$\forall R.C \in \mathcal{L}(v)$ $R \in \mathcal{L}(\langle v, w \rangle)$	$\mathcal{L}(w) = \mathcal{L}(w) \cup \{C\}$ $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{w:C} \geq x_{v:\forall R.C} \otimes x_{(v,w):R}\}$

Table 4. Rules of the tableaux algorithm combined with an optimization problem

EXAMPLE 6.2.2. Consider the fuzzy KB

$$\{\langle (\forall R.C)(a), 0.7 \rangle, \langle R(a, b), 0.6 \rangle, \langle \neg C(b), 0.8 \rangle\}.$$

The completion graph after the initialization and the application of the rules (\neg) and (\forall) can be depicted as:



where $\mathcal{C}_{\mathcal{F}} = \{x_{(\forall R.C)(a)} \in [0, 1], x_{R(a,b)} \in [0, 1], x_{(\neg C)(b)} \in [0, 1], x_{C(b)} \in [0, 1], x_{(\forall R.C)(a)} \geq 0.7, x_{R(a,b)} \geq 0.6, x_{(\neg C)(b)} \geq 0.8, x_{(\neg C)(b)} = 1 - x_{C(b)}, x_{C(b)} \geq x_{(\forall R.C)(a)} \otimes x_{R(a,b)}\}$.

It can be seen that this system of constraints has no solution since

$$x_{C(b)} \geq x_{(\forall R.C)(a)} \otimes x_{R(a,b)} \geq 0.7 \otimes 0.6 = 0.3 \text{ and}$$

$$x_{C(b)} = 1 - x_{(\neg C)(b)} \leq 1 - 0.8 = 0.2.$$

Hence, the KB is inconsistent.

Although the complexity of these algorithms has not been deeply studied, bMINLP and bMILP problems are NP-HARD problems, so these families have a high worst-case computational complexity.

6.2.3 Automata-based algorithms

Another important approach for reasoning in classical DLs is to reduce the reasoning problem to decide whether the language accepted by an automaton is empty or not. An advantage of this approach is that automata can handle non-determinism and infinite structures in a simple and elegant way. Thus, automata-based methods do not need complex blocking conditions or backtracking to ensure that a model exists. The reduction to automata has been successfully used to prove tight complexity bounds for a large family of classical DLs [1, 5]. Recently, it has been also explored for reasoning in FDLs. The main idea is to exploit the fact that many of these logics enjoy the *tree-model property*: a KB has a model iff it has a well-structured tree-shaped model. In order to decide $*$ -consistency, one only needs to decide the existence of one such model.

The typical approach is to abstract even further and construct an automaton whose runs describe so-called *Hintikka trees*. Hintikka trees can be seen as an abstract representation of the tree-shaped models, in which the membership degree of all relevant concepts is explicitly stated. More precisely, a Hintikka tree is a labeled infinite tree in which every node is labeled with a partial function, called the Hintikka function, from the set of all concepts appearing in the KB to the set of membership degrees, and every edge is labeled with a membership degree. The function at each node needs to be locally consistent; that is, the membership degree of e.g. the concept $C \sqcap D$ needs to be consistent with the membership degrees of C and D at each node of the tree. Likewise, it must be consistent with every GCI in the KB. This ensures that the semantics are locally preserved at every node. To handle existential and value restrictions, the functions associated at each node and its successors are also restricted in an appropriate manner; this restriction is called the Hintikka condition. Each such Hintikka tree represents a set of infinite models, that have as domain all the nodes in the tree, and the concept names and role names are interpreted according to their respective labels in the tree.

Since all the conditions are local, it is easy to build an automaton that receives as input an unlabeled infinite tree, and labels it through its runs in a way that successful runs correspond exactly to the set of Hintikka trees: the states are the locally consistent Hintikka functions, and the transitions are those tuples satisfying the Hintikka condition. Thus, the input KB is consistent iff this automaton has a successful run. The latter problem is known to be decidable in polynomial time on the size of the automaton.

Automata-based algorithms to decide witnessed $*$ -consistency for the logics between fuzzy \mathcal{ALC} and \mathcal{SHI} with GCIs for any finite residuated De Morgan lattice with a t-norm $*$ are summarized in [42]. Using more advanced automata-based techniques, it can be shown that the complexity in these logics is preserved.

Using an automata-based method to decide witnessed G-consistency for \mathcal{JALCE} in exponential time has been discussed in [35]. In this case, a model cannot be built directly, since that would require an automaton with infinitely many states. Instead, a novel idea is used, based on the defining properties of the minimum t-norm. The main insight is that the operators of this logic are invariant w.r.t. the specific membership degrees used, but depend only on the order between them. Thus, in this case it is possible to use preorders between the concepts, rather than explicit membership degrees, to label the nodes of the Hintikka tree. Each such Hintikka tree potentially represents uncountably many models, namely all those that share the exact same ordering among the relevant concepts.

While automata-based methods are useful for proving tight complexity bounds, they are not used in practical applications. The reason for this is that their worst-case behavior, used for providing complexity upper bounds, matches its best-case behavior. Before any reasoning step can be done, an automaton that is typically of exponential size needs to be constructed. Thus, even for simple input KBs, an exponential runtime cannot be avoided.

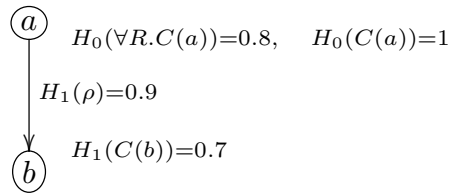
EXAMPLE 6.2.3. Consider the fuzzy KB

$$\{\langle \forall R.C(a), 0.8 \rangle, \langle \top \sqsubseteq C(b), 0.7 \rangle\}.$$

Three possible Hintikka functions are

$$\begin{aligned} H_0 &:= \forall r.C \mapsto 0.8, C \mapsto 1, \rho \mapsto 1, \\ H_1 &:= C \mapsto 0.7, \rho \mapsto 0.9, \\ H_2 &:= \rho \mapsto 0. \end{aligned}$$

Hintikka trees for this KB have arity 1, since only one quantified concept exists. One possible such tree can be depicted as:



Recall that Hintikka trees are infinite by definition; in this case, the node b has a successor labeled with the Hintikka function H_2 , which means that it is a successor to degree 0. Thus, it is not depicted in the figure. The Hintikka automaton for this very small KB contains over 16 states.

6.2.4 Reduction to classical DLs

The idea of this family of algorithms is to reduce fuzzy reasoning to finitely many crisp DL reasoning tasks. If the number of degrees is finite, it is possible to simulate fuzzy concepts and fuzzy roles using several α -cuts of the fuzzy sets and the fuzzy properties. The α -cut of a fuzzy set X is the crisp set that contains all the elements which belong to X with a degree greater than or equal to α . In this way, every axiom in the fuzzy ontology can be represented using these α -cuts. It can be shown that a fuzzy KB is $*$ -consistent if and only if its equivalent crisp KB is; other reasoning tasks can be reduced to $*$ -consistency.

In general, the translation of the equivalent crisp KBs grows exponentially with the size of the fuzzy KB. This approach is theoretically interesting to prove the decidability of FDLs. From an implementation point of view, it makes it possible to reuse existing crisp reasoners but since the translation to the crisp case is exponential it does not provide a tight complexity bound.

Algorithm 3 Reduction of a finitely-valued fuzzy \mathcal{ALCH} KB into a crisp KB**Require:** A fuzzy KB \mathcal{K} **Ensure:** true if \mathcal{K} is *-consistent; false otherwise

```

1:  $\text{crisp}(\mathcal{K}) \leftarrow \emptyset$ 
   {Axioms to keep the semantics of the new crisp concepts and roles}
2: for  $i := 1$  to  $n - 1$  do
3:   for each  $A \in N_A$  do
4:      $\text{crisp}(\mathcal{K}) \leftarrow \text{crisp}(\mathcal{K}) \cup \{A_{\geq t_{i+1}} \sqsubseteq A_{\geq t_i}\}$ 
5:   end for
6:   for each  $R \in N_R$  do
7:      $\text{crisp}(\mathcal{K}) \leftarrow \text{crisp}(\mathcal{K}) \cup \{R_{\geq t_{i+1}} \sqsubseteq R_{\geq t_i}\}$ 
8:   end for
9: end for
   {Crisp reduction of the axioms in the fuzzy KB}
10: for each axiom of the form  $\langle \tau, r \rangle \in \mathcal{K}$  do
11:    $\text{crisp}(\mathcal{K}) \leftarrow \text{crisp}(\mathcal{K}) \cup \kappa(\langle \tau, r \rangle)$ 
12: end for
13: if  $\text{crisp}(\mathcal{K})$  is consistent then
14:   return true
15: else
16:   return false
17: end if

```

To illustrate this family of algorithms, Algorithm 3 shows the process of reducing a \mathcal{ALCH} KB in finite Gödel extended with an involutive negation to an equivalent crisp KB. It is an adaptation of the more general algorithm proposed in [18].

It is convenient to recall that N_A is the set of atomic concepts and N_R the set of atomic roles in a fuzzy knowledge base \mathcal{K} , that we assume a finite chain of degrees of truth $T = \{t_0, t_1, \dots, t_n\}$ with $t_0 = 0$ and $t_n = 1$, and that axioms of the form $\langle \tau, t_0 \rangle$ are not allowed as they are tautologies. We will use T^+ to denote $T \setminus \{t_0\}$.

First, some new crisp concepts and roles representing α -cuts of the fuzzy concepts and roles are introduced. Having assumed a finite number of degrees of truth makes it possible to add α -cuts for every possible degree. The semantics of these newly introduced atomic concepts and roles is preserved by some new axioms (Lines 2–9). The idea is that an α -cut is subsumed by a β -cut if $\beta \leq \alpha$. Then, a mapping κ represents every axiom in the fuzzy KB using a crisp axiom that relies on these new crisp elements. Formally, the definition of κ is the following:

$$\begin{aligned}
\kappa(\langle C(a), r \rangle) &= \{\rho(C \geq r)(a)\} \\
\kappa(\langle R(a, b), r \rangle) &= \{\rho(R \geq r)(a, b)\} \\
\kappa(\langle C \sqsubseteq D, r \rangle) &= \bigcup_{t_j \in T^+, t_j \leq r} \{\rho(C \geq t_j) \sqsubseteq \rho(D \geq t_j)\} \\
\kappa(\langle R \sqsubseteq S, r \rangle) &= \bigcup_{t_j \in T^+, t_j \leq r} \{\rho(R \geq t_j) \sqsubseteq \rho(S \geq t_j)\}.
\end{aligned}$$

$$\begin{aligned}
\rho(\top \geq t_i) &= \top \\
\rho(\top \leq t_i) &= \perp \\
\rho(\perp \geq t_i) &= \perp \\
\rho(\perp \leq t_i) &= \top \\
\rho(A \geq t_i) &= A_{\geq t_i} \\
\rho(A \leq t_i) &= \neg A_{\geq t_{i+1}} \\
\rho(\neg C \geq t_i) &= \rho(C \leq t_{n-i}) \\
\rho(\neg C \leq t_i) &= \rho(C \geq t_{n-i}) \\
\rho(\sim C \geq t_i) &= \rho(C \leq t_0) \\
\rho(\sim C \leq t_i) &= \rho(C \geq t_1) \\
\rho(C \sqcap D \geq t_i) &= \rho(C \geq t_i) \sqcap \rho(D \geq t_i) \\
\rho(C \sqcap D \leq t_i) &= \rho(C \leq t_i) \sqcup \rho(D \leq t_i) \\
\rho(C \sqcup D \geq t_i) &= \rho(C \geq t_i) \sqcup \rho(D \geq t_i) \\
\rho(C \sqcup D \leq t_i) &= \rho(C \leq t_i) \sqcap \rho(D \leq t_i) \\
\rho(C \rightarrow D \geq t_i) &= \sqcap_{t_j \in T^+, t_j \leq t_i} (\neg \rho(C \geq t_j) \sqcup \rho(D \geq t_j)) \\
\rho(C \rightarrow D \leq t_i) &= \sqcup_{t_j \in T, t_j \leq t_i} (\rho(C \geq t_{j+1}) \sqcap \rho(D \leq t_j)) \\
\rho(\forall R.C \geq t_i) &= \sqcap_{t_j \in T^+, t_j \leq t_i} (\forall \rho(R \geq t_j). \rho(C \geq t_j)) \\
\rho(\forall R.C \leq t_i) &= \sqcup_{t_j \in T, t_j \leq t_i} (\exists \rho(R \geq t_{j+1}). \rho(C \leq t_j)) \\
\rho(\exists R.C \geq t_i) &= \exists \rho(R \geq t_i). \rho(C \geq t_i) \\
\rho(\exists R.C \leq t_i) &= \forall \rho(R \geq t_{i+1}). \rho(C \leq t_i) \\
\rho(R \geq t_i) &= R_{\geq t_i}
\end{aligned}$$

Table 5. Mapping of concept and role expressions

κ needs an auxiliary mapping ρ . Given a fuzzy concept C , $\rho(C \geq t_i)$ is a crisp set containing all the elements which belong to C with a degree greater than or equal to t_i . $\rho(R \geq t_i)$ is defined in a similar way for fuzzy roles, and $\rho(C \leq t_i)$ contains individuals belonging to C with a degree less or equal than t_i . Formally, the definition of ρ can be found in Table 5.

EXAMPLE 6.2.4. Assume a set of degrees of truth $\{0, 1/3, 2/3, 1\}$. In Gödel fuzzy logic, the fuzzy knowledge base $\mathcal{F} = \{\langle C(a), 1/3 \rangle, \langle C(b), 2/3 \rangle\}$ is equivalent to the crisp knowledge base

$$\mathcal{K} = \{C_{1/3}(a), C_{2/3}(b), C_{2/3} \sqsubseteq C_{1/3}, C_1 \sqsubseteq C_{2/3}\},$$

where $C_{1/3}, C_{2/3}, C_1$ are crisp concepts.

Algorithm 4 Reduction into a crisp KB for t-norms without zero divisors**Require:** A fuzzy KB \mathcal{K} **Ensure:** true if \mathcal{K} is *-consistent; false otherwise

```

1:  $\text{crisp}(\mathcal{K}) \leftarrow \emptyset$ 
2: for each axiom of the form  $\langle \tau, r \rangle \in \mathcal{K}$  do
3:    $\text{crisp}(\mathcal{K}) \leftarrow \text{crisp}(\mathcal{K}) \cup \{\tau\}$ 
4: end for
5: if  $\text{crisp}(\mathcal{K})$  is consistent then
6:   return true
7: else
8:   return false
9: end if

```

Another alternative is to ignore fuzziness by just discarding the degree of truth in the fuzzy axioms (the details of this process are provided in Lemma 7.3.1). In some cases, it can be shown that KB *-consistency is trivially reducible to crisp reasoning. Hence, KB *-consistency and crisp consistency have the same computational complexity. It is not possible if the language contains an involutive negation or has upper bounds in the axioms, i.e., axioms of the form $\langle \tau \leq r \rangle$. If this is not the case and we consider a strict t-norm $*$ (that is, if $*$ is an ordinal sum that does not start with a Łukasiewicz component), a fuzzy KB in \mathcal{SHOI} is *-consistent if and only if it has a crisp model [34]. Algorithm 4 illustrates this result. The same idea can be applied to the concept *-satisfiability but not to *-subsumption. In the less expressive FDL \mathcal{EL} , the same result does hold for *-subsumption [43].

EXAMPLE 6.2.5. In Gödel fuzzy logic, the fuzzy knowledge base

$$\mathcal{F} = \{\langle C(a), 1/3 \rangle, \langle C(b), 2/3 \rangle\}$$

is equivalent to the crisp base $\mathcal{K} = \{C(a), C(b)\}$, where C is a crisp concept.

6.2.5 Reduction to propositional fuzzy logics

P. Hájek proposed a reduction of FDL concepts into propositional fuzzy formulas [80]. In particular, he considered the FDL \mathcal{JALUE} under arbitrary continuous t-norms and defined a reasoning algorithm for witnessed *-satisfiability and *-validity of concepts based on a reduction to BL. Please note that the involutive negation is not part the language when the semantics is not based on Łukasiewicz t-norm. The strong disjunction was also not considered in the original formulation, but, since the algorithm presented in [80] is based on a reduction to the respective propositional calculus, it can be easily expanded in order to cope with the strong disjunction too. The same author extended the results by introducing rational truth degrees, and thus making it possible to have ABoxes [81].

This approach also has some limitations: the translation is more than exponential in the size of the fuzzy concept (see [54] for a detailed proof), it only works for witnessed concept satisfiability, and it does not consider TBoxes in the language.

Let us now describe in detail Hájek's algorithm [80]. To decide the satisfiability of a concept C , we start with an instance of the form $C(a)$ for a new individual a . Next, a finite *witnessing theory* $T(C(a))$ is computed from $C(a)$. $C(a)$ can be seen as a propositional combination of instances of non-quantified concepts and instances of quantified concepts. The idea is that for every instance of a quantified concept, a new witness individual and some instances keeping the semantics of the quantified concept are created. The new instances are quantifier-free formulas. This is similar to what tableaux algorithms do. However, instead of looking for a clash-free completion forest, all the instances are reduced to a propositional fuzzy logic formula.

Firstly, we will show how to compute a finite theory $T(C(a))$ from an instance $C(a)$. Then, we will show how to compute a propositional formula $\text{prop}(C(a))$ for every instance $C(a)$. $T(C(a))$ can be computed as shown in Algorithm 5. It uses an iterative process that is repeated n times, where n is the degree of nesting of quantifiers in C , denoted $\text{nest}(C)$ and defined as follows:

$$\begin{aligned}\text{nest}(A) &= 0 \\ \text{nest}(\sim C) &= \text{nest}(C) \\ \text{nest}(C \sqcap D) = \text{nest}(C \vee D) = \text{nest}(C \wedge D) &= \max\{\text{nest}(C), \text{nest}(D)\} \\ \text{nest}(\forall R.C) = \text{nest}(\exists R.C) &= 1 + \text{nest}(C).\end{aligned}$$

Every step i creates a set of new individuals (denoted I_i), adds new axioms to the theory (denoted $T(C(a))$), and introduces new instances to be processed by the next step (denoted S_i). The i -th step considers the set of instances S_i introduced in the previous step and produces a new set with a degree of nesting $n - i$. The algorithm starts with the initial instance $C(a)$. Instances α of the forms $(\exists R.C)(b)$ and $(\forall R.C)(b)$ can generate new individuals d_α which are witnesses of the existential and universal restrictions. All the individuals created in the i -th step are collected in a set I_i and will be revisited to add some new axioms, keeping the semantics of existential and universal restrictions. The axiom $x \equiv y$ is used as a shorthand for the pair of formulas $x \rightarrow y$ and $y \rightarrow x$.

Now, it remains to show how to compute $\text{prop}(C(a))$. These formulas are defined by induction, where the base case are atomic concepts, existential quantifications and value restrictions, which are assigned a propositional variable $p_{C,a}$. In particular, $\text{prop}(C(a))$ is defined as follows:

$$\begin{aligned}\text{prop}(A(a)) &= p_{A,a} \\ \text{prop}(\exists R.C(a)) &= p_{\exists R.C,a} \\ \text{prop}(\forall R.C(a)) &= p_{\forall R.C,a} \\ \text{prop}(\perp)(a) &= \bar{0} \\ \text{prop}(C \sqcap D(a)) &= \text{prop}(C(a)) \& \text{prop}(D(a)) \\ \text{prop}(C \wedge D(a)) &= \text{prop}(C(a)) \& (\text{prop}(C(a)) \rightarrow \text{prop}(D(a))) \\ \text{prop}(C \vee D(a)) &= \text{prop}((C \rightarrow D) \rightarrow D(a)) \& \\ &\quad (\text{prop}((C \rightarrow D) \rightarrow D(a)) \rightarrow \text{prop}((D \rightarrow C) \rightarrow C(a))) \\ \text{prop}(C \rightarrow D(a)) &= \text{prop}(C(a)) \rightarrow \text{prop}(D(a)) \\ \text{prop}(\neg C(a)) &= \text{prop}(C(a)) \rightarrow \bar{0}.\end{aligned}$$

Algorithm 5 Hájek's reduction to propositional fuzzy logics**Require:** C **Ensure:** true if C is *-satisfiable; false otherwise

```

1:  $a \leftarrow$  some constant
2:  $S_0 \leftarrow \{C(a)\}$ 
   {Compute  $T(C(a))$ }
3: for  $i := 1$  to  $\text{nest}(C)$  do
4:    $I_i \leftarrow \emptyset$ 
5:    $T(C(a)) \leftarrow \emptyset$ 
6:    $S_i \leftarrow \emptyset$ 
   {Create witnesses and add witnessing axioms}
7:   for each  $\alpha \in S_{i-1}$  do
8:     if  $\alpha$  of the form  $\forall R.D(b)$  then
9:       create a new individual  $d_\alpha$ 
10:       $I_i \leftarrow I_i \cup \{d_\alpha\}$ 
11:       $T(C(a)) \leftarrow T(C(a)) \cup \{\forall R.D(b) \equiv (R(b, d_\alpha) \rightarrow D(d_\alpha))\}$ 
12:       $S_i \leftarrow S_i \cup \{D(d_\alpha)\}$ 
13:     else if  $\alpha$  is of the form  $\exists R.D(b)$  then
14:       create a new individual  $d_\alpha$ 
15:       $I_i \leftarrow I_i \cup \{d_\alpha\}$ 
16:       $T(C(a)) \leftarrow T(C(a)) \cup \{\exists R.D(b) \equiv (R(b, d_\alpha) \& D(d_\alpha))\}$ 
17:       $S_i \leftarrow S_i \cup \{D(d_\alpha)\}$ 
18:     end if
19:   end for
   {Apply restrictions to each individual generated in this step using the same role}
20:   for each  $\alpha \in S_{i-1}$  of the form  $(\forall R.D)(b)$  and each  $d_\beta \in I_i$  with  $\alpha \neq \beta$  do
21:     if  $\beta$  has the form  $\exists R.X(b)$  or  $\forall R.X(b)$  and  $\alpha \neq \beta$  then
22:        $T(C(a)) \leftarrow T(C(a)) \cup \{\forall R.D(b) \rightarrow (R(b, d_\beta) \rightarrow D(d_\beta))\}$ 
23:        $S_i \leftarrow S_i \cup \{C(d_\beta)\}$ 
24:     end if
25:   end for
26:   for each  $\alpha \in S_{i-1}$  of the form  $\exists R.D(b)$  and each  $d_\beta \in I_i$  with  $\alpha \neq \beta$  do
27:     if  $\beta$  has the form  $\exists R.X(b)$  or  $\forall R.X(b)$  and  $\alpha \neq \beta$  then
28:        $T(C(a)) \leftarrow T(C(a)) \cup \{(R(b, d_\beta) \& D(d_\beta)) \rightarrow \exists R.D(b)\}$ 
29:        $S_i \leftarrow S_i \cup \{C(d_\beta)\}$ 
30:     end if
31:   end for
32: end for
33: return  $T(C(a))$ 
   {Solve the propositionally equivalent formula}
34: if  $\text{prop}(C(a)) \cup \text{prop}(T(C(a)))$  is *-satisfiable then
35:   return true
36: else
37:   return false
38: end if

```

EXAMPLE 6.2.6. Consider the concept

$$\forall R.C \sqcap \exists R.\neg C.$$

According to Algorithm 5, the list of propositional formulas produced from the above KB at the end of the process is the following:

1. $p_{\forall R.C} \& p_{\exists R.\neg C}$
2. $p_{\forall R.C(d)} \equiv p_{R(d, d_{\forall R.C(b)})} \rightarrow p_{C(d_{\forall R.C(b)})}$
3. $p_{\exists R.\neg C(d)} \equiv p_{R(d, d_{\exists R.\neg C(b)})} \& \neg p_{C(d_{\exists R.\neg C(b)})}$
4. $p_{\forall R.C(d)} \rightarrow (p_{R(d, d_{\exists R.\neg C(b)})} \rightarrow p_{C(d_{\exists R.\neg C(b)})})$
5. $(p_{R(d, d_{\forall R.C(b)})} \& \neg p_{C(d_{\forall R.C(b)})}) \rightarrow p_{\exists R.\neg C(d)}$.

We show that the propositional theory above is not 1-satisfiable; suppose that there exists a propositional evaluation e such that $e(p_{\forall R.C} \& p_{\exists R.\neg C}) = 1$, then

- by 1. both $e(p_{\forall R.C}) = 1$ and $e(p_{\exists R.\neg C}) = 1$, hence
- by 3. both $e(p_{R(d, d_{\exists R.\neg C(b)})}) = 1$ and $e(p_{C(d_{\exists R.\neg C(b)})}) = 0$, therefore
- by 4. $e(p_{\forall R.C(d)}) = 0$, a contradiction.

6.2.6 Structural subsumption algorithms

A structural subsumption algorithm is a relatively simple algorithm suited for solving subsumption with respect to empty knowledge bases in very basic FDLs. Both in the case of classical DLs and of FDLs it has been historically the first algorithm to be designed (see Section 8 for more historical details). Its problem is that it turns out to be incomplete for more expressive languages.

Let us now describe in detail the structural subsumption algorithm for subsumption with respect to empty knowledge bases in language \mathcal{FL}^- , from [61]. To decide if a concept C is subsumed by a concept D , we recursively build a set of matrices $\mathbf{E}_{C,D}$. These matrices contain a row for each value restriction $\forall R.F$ which is a conjunct of D and a column for each value restriction $\forall R.E$ which is a conjunct of C . In particular, if there are multiple occurrences of the same expression then there are multiple rows for them. The same observation applies for the columns in $\mathbf{E}_{C,D}$. The only non-deterministic problem is to decide whether every value restriction $\forall R.F$ which is a conjunct of a given subconcept D' of D is subsumed by a different value restriction $\forall R.E$ which is a conjunct of a given subconcept C' of C . But, as in [108], instead of checking out this fact for different non-deterministic guesses, a suitable procedure for the bipartite matching problem (see [86] for example) can give an answer in polynomial time.

Algorithm 6 Computing structural subsumption for $\mathbf{L}_n\text{-}\mathcal{FL}^-$ **Require:** C, D **Ensure:** true if C is 1-subsumed by D ; false otherwise

```

if there is an occurrence of an atomic or existential conjunct  $A$  of  $D$  that is not in  $C$ 
  where concept  $A$  appears in  $C$  strictly less  $n - 1$  times then
    return false
  else
     $\mathbf{E}_{C,D} := \emptyset$ 
    for value restriction  $\forall R.F$  which is a conjunct of  $D$  do
      for value restriction  $\forall R.E$  which is a conjunct of  $C$  do
         $\mathbf{E}_{C,D}(\forall R.F, \forall R.E) := \mathbf{L}_n\text{-SUBS}(1, F, E)$ 
      end for
    end for
    if there is a maximal bipartite matching for  $\mathbf{E}_{C,D}$  then
      return true
    else
      return false
    end if
  end if

```

EXAMPLE 6.2.7. Consider the concepts

$$\forall R.(\forall P.(A \sqcap C) \sqcap \forall P.C) \sqcap \forall R.(C \sqcap D) \quad \text{and} \quad \forall R.(\forall P.A \sqcap \forall P.B) \sqcap \forall R.C.$$

According to Algorithm 6, if we want to see whether the first concept is subsumed by the second, the following matrices will be created:

	$\forall P.A$	$\forall P.B$		$\forall R.(\forall P.A \sqcap \forall P.B)$	$\forall R.C$
$\forall P.(A \sqcap C)$	×		$\forall R.(\forall P.(A \sqcap C) \sqcap \forall P.C)$		
$\forall P.C$			$\forall R.(C \sqcap D)$		×

It can be seen that in the first matrix the subconcept $\forall P.C$ is not subsumed by any subconcept of $\forall R.(\forall P.A \sqcap \forall P.B) \sqcap \forall R.C$. Therefore, recursively, the concept $\forall R.(\forall P.(A \sqcap C) \sqcap \forall P.C) \sqcap \forall R.(C \sqcap D)$ is not subsumed by $\forall R.(\forall P.A \sqcap \forall P.B) \sqcap \forall R.C$.

7 Decidability and complexity issues

In this section, we present the known results on the decidability and complexity of reasoning in FDLs. We first prove that KB consistency is undecidable for the DL $\mathcal{JALCC\mathcal{E}}$ if the semantics is based on the product t-norm. The proof presented is intended only as a prototype for a general method of proving undecidability that has been applied to a large class of description logic languages. These further results are summarized next, with a general idea on how the framework can be applied to obtain them. Afterwards, we provide an overview of the complexity of reasoning in the known decidable cases.

7.1 Undecidability of consistency in \mathcal{JALCE}

The consistency problem has been recently shown to be undecidable for several FDLs, where general concept inclusion axioms are allowed [36] (see Section 8 for more historical details). While these proofs of undecidability differ in their details, they are all based on the same general idea of modeling instances of the *Post correspondence problem* through structured tree-shaped models of a KB. In this section, we provide a prototypical proof of undecidability by showing that Π -consistency of \mathcal{JALCE} knowledge bases (w.r.t. witnessed models) is undecidable. This proof is intended to highlight the core steps followed by most other undecidability proofs for FDLs, and can thus be used as a basic model for understanding these other proofs.

For this whole section, we fix the semantics of \mathcal{JALCE} to consider only the product t -norm and its associated operators plus the standard involutive negation for interpreting the different constructors. This is an FDL language defined on the first-order version of the logic $\mathbb{L}\Pi_{\frac{1}{2}}$ (see Section 5.4 of Chapter VIII of this Handbook). The proof of undecidability is based on a reduction from a slight variant of the Post correspondence problem, which is well-known to be undecidable [106].

DEFINITION 7.1.1. *Let $\langle v_1, w_1 \rangle, \dots, \langle v_m, w_m \rangle$ be a finite list of pairs of words over a finite alphabet $\Sigma = \{1, \dots, s\}$, $s > 1$. The Post correspondence problem (PCP) consists of deciding whether there is a sequence i_1, i_2, \dots, i_k , $1 \leq i_j \leq m$, such that $v_1 v_{i_1} v_{i_2} \dots v_{i_k} = w_1 w_{i_1} w_{i_2} \dots w_{i_k}$. If such a sequence exists, the word $i_1 i_2 \dots i_k$ is called a solution of the problem.*

Given a word $\mu = i_1 i_2 \dots i_k \in \{1, \dots, m\}^*$, we will use v_μ and w_μ to denote the words $v_1 v_{i_1} v_{i_2} \dots v_{i_k}$ and $w_1 w_{i_1} w_{i_2} \dots w_{i_k}$, respectively.

Intuitively, we can see every instance of the PCP as an m -ary infinite tree such that (i) the root node is labeled with $\langle v_\varepsilon, w_\varepsilon \rangle := \langle v_1, w_1 \rangle$, and (ii) if a node is labeled with the pair $\langle v, w \rangle$, then its i -th successor is labeled with the pair $\langle v v_i, w w_i \rangle$ (see Figure 4).⁸ The PCP then consists in deciding whether there is a node in this tree whose label $\langle v, w \rangle$ is such that $v = w$.

Recall from Definition 7.1.1 that the alphabet Σ over which an instance of the PCP is described consists of the first s positive integers. We can thus view every word in Σ^* as a natural number represented in base $s + 1$ in which 0 never occurs. Slightly abusing this intuition, we will express the empty word as the number 0. To reduce the PCP, we need to be able to express words not as integers, but rather as real numbers in the interval $[0, 1]$. To achieve this, we encode each word u in Σ^* through the number $2^{-u} \in [0, 1]$.

The main idea of the reduction is to construct a knowledge base $\mathcal{K}_{\mathcal{P}} = \langle \mathcal{T}_{\mathcal{P}}, \mathcal{A}_{\mathcal{P}} \rangle$ whose models encode an instance of the PCP, viewed as a tree as in Figure 4. In other words, every model contains, for each possible solution $\mu \in \{1, \dots, m\}^*$, a domain element at which the interpretation of two designated concept names A and B will correspond to the words v_μ, w_μ , respectively. More formally, for a given instance $\mathcal{P} = \langle \langle v_1, w_1 \rangle, \dots, \langle v_m, w_m \rangle \rangle$ of the PCP, we define an ABox $\mathcal{A}_{\mathcal{P}}$ and a TBox $\mathcal{T}_{\mathcal{P}}$ such that for every witnessed model \mathcal{I} of $\mathcal{A}_{\mathcal{P}}$ and $\mathcal{T}_{\mathcal{P}}$ and every $\mu \in \{1, \dots, m\}^*$

⁸ Notice that the first word concatenated is always v_1 and w_1 .

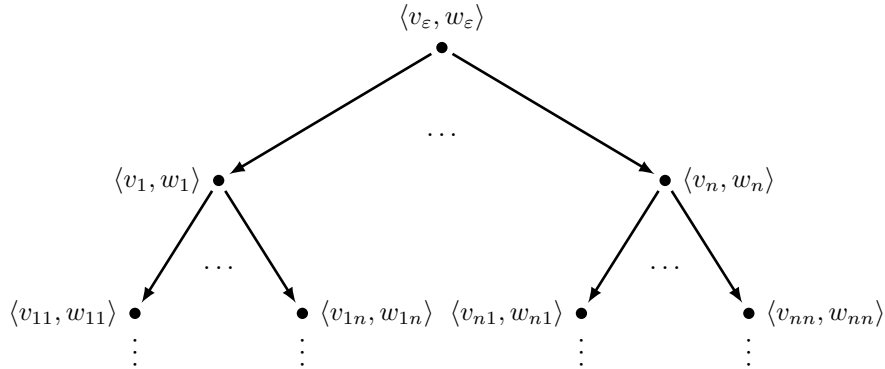


Figure 4. An instance of the PCP

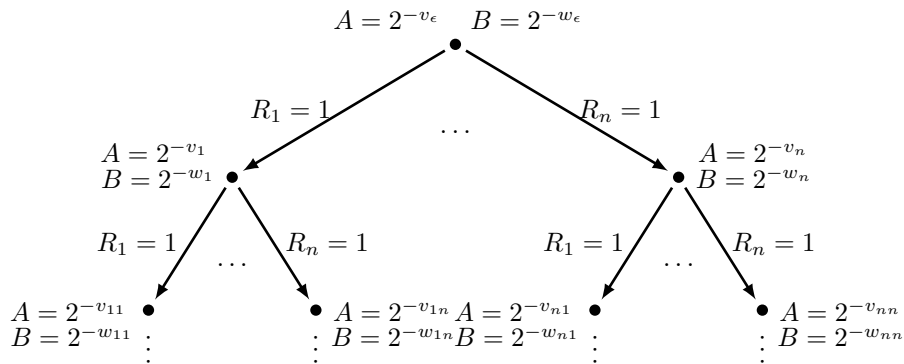


Figure 5. The interpretation $\mathcal{I}_{\mathcal{P}}$

there is $\delta_\mu \in \Delta^{\mathcal{I}}$ with $A^{\mathcal{I}}(\delta_\mu) = 2^{-v_\mu}$ and $B^{\mathcal{I}}(\delta_\mu) = 2^{-w_\mu}$. Additionally, we will show that there is a witnessed model $\mathcal{I}_{\mathcal{P}}$ of $\mathcal{A}_{\mathcal{P}}$ and $\mathcal{T}_{\mathcal{P}}$ whose domain has only these elements (see Figure 5). Then, the instance \mathcal{P} has a solution iff for every witnessed model \mathcal{I} of the constructed knowledge base $\mathcal{K}_{\mathcal{P}}$ there exists a $\delta \in \Delta^{\mathcal{I}}$ such that $A^{\mathcal{I}}(\delta) = B^{\mathcal{I}}(\delta)$.

We first introduce some abbreviations that will be helpful in simplifying the description of the construction. We use the expression C^n to denote the n -ary conjunction of a concept description C with itself; formally, $C^0 := \top$ and $C^{m+1} := C \sqcap C^m$ for every $n \geq 0$. Given two concept descriptions C, D and a role name R , we use the expression $\langle C \overset{R}{\rightsquigarrow} D \rangle$ to denote the two axioms $\langle C \sqsubseteq \forall R.D, 1 \rangle$ and $\langle \exists R.D \sqsubseteq C, 1 \rangle$. These axioms are used to transfer specific values along role connections of degree 1, as described in the following lemma.

$$\begin{array}{l}
\mathcal{A}_{\mathcal{P}} := \{ \langle A(a), 2^{-v_1} \rangle, \langle B(a), 2^{-w_1} \rangle, \langle \sim A(a), 1 - 2^{-v_1} \rangle, \langle \sim B(a), 1 - 2^{-w_1} \rangle \} \cup \\
\quad \{ \langle V_i(a), 2^{-v_i} \rangle, \langle W_i(a), 2^{-w_i} \rangle, \langle \sim V_i(a), 1 - 2^{-v_i} \rangle, \langle \sim W_i(a), 1 - 2^{-w_i} \rangle \mid 1 \leq i \leq m \} \\
\mathcal{T}_{\mathcal{P}}^0 := \{ \langle V_j \overset{R_i}{\rightsquigarrow} V_j \rangle, \langle W_j \overset{R_i}{\rightsquigarrow} W_j \rangle \mid 1 \leq i, j \leq m \} \\
\mathcal{T}_{\mathcal{P}}^i := \{ \langle \top \sqsubseteq \exists R_i . \top, 1 \rangle, \langle (V_i \sqcap A^{(s+1)^{|v_i|}}) \overset{R_i}{\rightsquigarrow} A \rangle, \langle (W_i \sqcap B^{(s+1)^{|w_i|}}) \overset{R_i}{\rightsquigarrow} B \rangle \} \\
\mathcal{T}_{\mathcal{P}} := \bigcup_{i=0}^m \mathcal{T}_{\mathcal{P}}^i \\
\mathcal{T}'_{\mathcal{P}} := \mathcal{T}_{\mathcal{P}} \cup \{ \langle \top \sqsubseteq \sim((A \rightarrow B) \sqcap (B \rightarrow A)), 0.5 \rangle \} \\
\mathcal{K}_{\mathcal{P}} := \langle \mathcal{T}_{\mathcal{P}}, \mathcal{A}_{\mathcal{P}} \rangle \\
\mathcal{K}'_{\mathcal{P}} := \langle \mathcal{T}'_{\mathcal{P}}, \mathcal{A}_{\mathcal{P}} \rangle
\end{array}$$

Figure 6. The knowledge bases $\mathcal{K}_{\mathcal{P}}$ and $\mathcal{K}'_{\mathcal{P}}$

LEMMA 7.1.2. *For every interpretation \mathcal{I} and all $x, y \in \Delta^{\mathcal{I}}$, we have*

- $(C^n)^{\mathcal{I}}(x) = (C^{\mathcal{I}}(x))^n$.
- If \mathcal{I} satisfies $\langle C \overset{R}{\rightsquigarrow} D \rangle$ and $R^{\mathcal{I}}(x, y) = 1$, then $C^{\mathcal{I}}(x) = D^{\mathcal{I}}(y)$.

Proof. The first equality obviously follows from the definition of C^n and the fact that we are using the product t-norm. For the second equality, since \mathcal{I} satisfies the GCI $\langle C \sqsubseteq \forall R.D, 1 \rangle$, we have that $C^{\mathcal{I}}(x) \leq (\forall R.D)^{\mathcal{I}}(x)$, and thus

$$C^{\mathcal{I}}(x) \leq \inf_{\delta \in \Delta^{\mathcal{I}}} (R^{\mathcal{I}}(x, \delta) \Rightarrow_{\sqcap} D^{\mathcal{I}}(\delta)) \leq R^{\mathcal{I}}(x, y) \Rightarrow_{\sqcap} D^{\mathcal{I}}(y) = D^{\mathcal{I}}(y).$$

Dually, since \mathcal{I} satisfies $\langle \exists R.D \sqsubseteq C, 1 \rangle$, we obtain $(\exists R.D)^{\mathcal{I}}(x) \leq C^{\mathcal{I}}(x)$, and hence

$$C^{\mathcal{I}}(x) \geq \sup_{\delta \in \Delta^{\mathcal{I}}} (R^{\mathcal{I}}(x, \delta) * D^{\mathcal{I}}(\delta)) \geq R^{\mathcal{I}}(x, y) * D^{\mathcal{I}}(y) = D^{\mathcal{I}}(y).$$

These two facts together imply that $C^{\mathcal{I}}(x) = D^{\mathcal{I}}(y)$, as claimed. \square

We are now ready to construct a knowledge base that encodes the search space for a solution of \mathcal{P} as depicted in Figure 5. Given an instance \mathcal{P} , we construct the knowledge base $\mathcal{K}_{\mathcal{P}}$ depicted in Figure 6. To aid in the understanding of this construction, we now develop it step-wise, explaining the use of each axiom in the reduction. We first show how to enforce the tree-like structure into every model. Afterwards, we add new axioms that allow us to detect whether a solution exists or not.

7.1.1 Constructing the successor nodes

We start by assuming that we have already constructed a node $\delta \in \Delta^{\mathcal{I}}$ in some interpretation that encodes $v, w \in \Sigma^*$; that is, $A^{\mathcal{I}}(\delta) = 2^{-v}$ and $B^{\mathcal{I}}(\delta) = 2^{-w}$. We show how to generate the first such node in Section 7.1.2; our goal now is to ensure, for each $i, 1 \leq i \leq m$, the existence of an R_i -successor of δ that encodes the concatenation of v, w with the i -th pair from \mathcal{P} , i.e., vv_i and ww_i . We assume for now that there are concept names V_i, W_i encoding v_i and w_i ; more precisely, we know that $V_i^{\mathcal{I}}(\delta) = 2^{-v_i}$ and $W_i^{\mathcal{I}}(\delta) = 2^{-w_i}$ for each $i, 1 \leq i \leq m$. We construct the TBoxes:

$$\mathcal{T}_{\mathcal{P}}^i := \{ \langle \top \sqsubseteq \exists R_i . \top, 1 \rangle, \langle (V_i \sqcap A^{(s+1)^{|v_i|}}) \overset{R_i}{\rightsquigarrow} A \rangle, \langle (W_i \sqcap B^{(s+1)^{|w_i|}}) \overset{R_i}{\rightsquigarrow} B \rangle \}.$$

Recall that we are viewing each word in Σ^* as a natural number in base $s + 1$. Therefore, under this view, the concatenation of two words u, u' corresponds to the operation $u \cdot (s + 1)^{|u'|} + u'$ on natural numbers. Consider the concept $V_i \sqcap A^{(s+1)^{|v_i|}}$, the interpretation \mathcal{I} satisfies

$$(V_i \sqcap A^{(s+1)^{|v_i|}})^{\mathcal{I}}(\delta) = V_i^{\mathcal{I}}(\delta) \cdot (A^{\mathcal{I}}(\delta))^{(s+1)^{|v_i|}} = 2^{-(v(s+1)^{|v_i|} + v_i)} = 2^{-vv_i}.$$

In other words, this concept encodes the word vv_i , which is the word that should appear in the i -th successor of δ . If \mathcal{I} is a witnessed model of $\mathcal{T}_{\mathcal{P}}^i$, then from the first axiom it follows that there must exist an element $\gamma \in \Delta^{\mathcal{I}}$ with $R_i^{\mathcal{I}}(\delta, \gamma) = 1$. As argued above, the last two axioms then ensure that $A^{\mathcal{I}}(\gamma) = 2^{-vv_i}$ and $B^{\mathcal{I}}(\gamma) = 2^{-ww_i}$; thus, the concept names A and B encode, at node γ , the words vv_i and ww_i , as desired.

The idea is to use this construction recursively so that, starting from an appropriate root node, we can guarantee the existence of the full infinite tree that describes the instance \mathcal{P} . To achieve this, we need to guarantee that at this new node γ we also have that $V_j^{\mathcal{I}}(\gamma) = 2^{-v_j}$ and $W_j^{\mathcal{I}}(\gamma) = 2^{-w_j}$ hold for every $j, 1 \leq j \leq m$, since this was a necessary condition for the correctness of the construction. This can be done by including the axioms

$$\mathcal{T}_{\mathcal{P}}^0 := \{ \langle V_j \overset{R_i}{\rightsquigarrow} V_j \rangle, \langle W_j \overset{R_i}{\rightsquigarrow} W_j \rangle \mid 1 \leq i, j \leq m \}$$

to the TBox.

The union of the TBoxes $\mathcal{T}_{\mathcal{P}}^i$ for $0 \leq i \leq m$ guarantees the construction of the search tree for \mathcal{P} , provided that we can build a node that encodes the appropriate restrictions at the root of this tree.

7.1.2 Constructing the root node

We now need to ensure that there exists a node $\delta_\varepsilon \in \Delta^{\mathcal{I}}$ such that $A^{\mathcal{I}}(\delta_\varepsilon) = 2^{-v_1}$ and $B^{\mathcal{I}}(\delta_\varepsilon) = 2^{-w_1}$; that is, where A and B encode v_ε and w_ε , respectively. Additionally, at this node δ_ε , $V_j^{\mathcal{I}}(\delta_\varepsilon) = 2^{-v_j}$, $W_j^{\mathcal{I}}(\delta_\varepsilon) = 2^{-w_j}$ must hold for every $j, 1 \leq j \leq m$. To enforce these conditions, we use the ABox:

$$\begin{aligned} \mathcal{A}_{\mathcal{P}} := & \{ \langle A(a), 2^{-v_1} \rangle, \langle B(a), 2^{-w_1} \rangle, \langle \sim A(a), 1 - 2^{-v_1} \rangle, \langle \sim B(a), 1 - 2^{-w_1} \rangle \} \cup \\ & \cup \{ \langle V_i(a), 2^{-v_i} \rangle, \langle W_i(a), 2^{-w_i} \rangle, \langle \sim V_i(a), 1 - 2^{-v_i} \rangle, \langle \sim W_i(a), 1 - 2^{-w_i} \rangle \mid 1 \leq i \leq m \}. \end{aligned}$$

Recall that \sim stands for the strong complementation constructor. Any interpretation satisfying the two axioms $\langle A(a), 2^{-v_1} \rangle$ and $\langle \sim A(a), 1 - 2^{-v_1} \rangle$ contains an element $\delta_\varepsilon := a^{\mathcal{I}}$ such that $A^{\mathcal{I}}(\delta_\varepsilon) \geq 2^{-v_1}$ and $1 - A^{\mathcal{I}}(\delta_\varepsilon) = (\sim A)^{\mathcal{I}}(\delta_\varepsilon) \geq 1 - 2^{-v_1}$ and hence $A^{\mathcal{I}}(\delta_\varepsilon) = 2^{-v_1}$. A similar argument shows that in every model of $\mathcal{A}_{\mathcal{P}}$, δ_ε is interpreted to the desired membership degrees of the concepts B and V_j, W_j for each $j, 1 \leq j \leq m$. Thus, this element can be used to represent the root of the tree.

7.1.3 The canonical model

We now show that every model of the axioms defined so far encodes the instance \mathcal{P} of the PCP. Let $\mathcal{T}_{\mathcal{P}} := \bigcup_{i=0}^m \mathcal{T}_{\mathcal{P}}^i$ and $\mathcal{K}_{\mathcal{P}} = \langle \mathcal{T}_{\mathcal{P}}, \mathcal{A}_{\mathcal{P}} \rangle$. We define the *canonical interpretation* as the interpretation $\mathcal{I}_{\mathcal{P}} := \langle \Delta^{\mathcal{I}_{\mathcal{P}}}, \cdot^{\mathcal{I}_{\mathcal{P}}} \rangle$ whose domain is the set

$\Delta^{\mathcal{I}_{\mathcal{P}}} = \{1, \dots, m\}^*$ of all finite words over $\{1, \dots, m\}$ and the interpretation function $\cdot^{\mathcal{I}_{\mathcal{P}}}$ is such that: (i) the individual name a is mapped to $a^{\mathcal{I}_{\mathcal{P}}} = \varepsilon$, (ii) for every $\mu \in \Delta^{\mathcal{I}_{\mathcal{P}}}$, $A^{\mathcal{I}_{\mathcal{P}}}(\mu) = 2^{-v_\mu}$, and $B^{\mathcal{I}_{\mathcal{P}}}(\mu) = 2^{-w_\mu}$, and (iii) for all $j, 1 \leq j \leq m$,

- $V_j^{\mathcal{I}_{\mathcal{P}}}(\mu) = 2^{-v_j}$ and $W_j^{\mathcal{I}_{\mathcal{P}}}(\mu) = 2^{-w_j}$.
- $R_j^{\mathcal{I}_{\mathcal{P}}}(\mu, \mu_j) = 1$ and $R_j^{\mathcal{I}_{\mathcal{P}}}(\mu, \mu') = 0$ if $\mu' \neq \mu_j$.

This interpretation is precisely the one depicted in Figure 5. It is easy to see that $\mathcal{I}_{\mathcal{P}}$ is a model of $\mathcal{K}_{\mathcal{P}}$, and is in fact witnessed, since every node has exactly one R_i -successor with degree greater than 0, for every $i, 1 \leq i \leq m$. More interesting, however, is that for every witnessed model \mathcal{I} of $\mathcal{K}_{\mathcal{P}}$, there is a homomorphism from $\mathcal{I}_{\mathcal{P}}$ to \mathcal{I} as described in the following lemma.

LEMMA 7.1.3. *Let \mathcal{I} be a witnessed model of $\mathcal{K}_{\mathcal{P}}$. There is a function $f: \Delta^{\mathcal{I}_{\mathcal{P}}} \rightarrow \Delta^{\mathcal{I}}$ such that, for every $\mu \in \Delta^{\mathcal{I}_{\mathcal{P}}}$ and every concept name C appearing in $\mathcal{A}_{\mathcal{P}}$ or in $\mathcal{T}_{\mathcal{P}}$, it holds that $C^{\mathcal{I}_{\mathcal{P}}}(\mu) = C^{\mathcal{I}}(f(\mu))$.*

Proof. The function f is built inductively on the length of μ . For $\mu = \varepsilon$, $\mathcal{A}_{\mathcal{P}}$ fixes the interpretation of all relevant concept names at $a^{\mathcal{I}}$ and hence defining $f(\varepsilon) := a^{\mathcal{I}}$ satisfies the condition of the lemma.

Let now μ be such that $f(\mu)$ is already defined. By induction, assume that $A^{\mathcal{I}}(f(\mu)) = 2^{-v_\mu}$, $B^{\mathcal{I}}(f(\mu)) = 2^{-w_\mu}$, and that for every j , where $1 \leq j \leq m$, $V_j^{\mathcal{I}}(f(\mu)) = 2^{-v_j}$, $W_j^{\mathcal{I}}(f(\mu)) = 2^{-w_j}$. Since \mathcal{I} is a witnessed model of the axiom $\langle \top \sqsubseteq \exists R_i. \top, 1 \rangle$, for all $i, 1 \leq i \leq m$ there exists a $\gamma \in \Delta^{\mathcal{I}}$ with $r^{\mathcal{I}}(f(\mu), \gamma) = 1$. As \mathcal{I} satisfies all the axioms of the form $\langle C \overset{R_i}{\rightsquigarrow} D \rangle \in \mathcal{T}_{\mathcal{P}}$, it follows that

$$A^{\mathcal{I}}(\gamma) = 2^{-v_\mu v_i} = 2^{-v_{\mu i}}, \quad B^{\mathcal{I}}(\gamma) = 2^{-w_\mu w_i} = 2^{-w_{\mu i}},$$

and for all $j, 1 \leq j \leq m$, $V_j^{\mathcal{I}}(\gamma) = 2^{-v_j}$, $W_j^{\mathcal{I}}(\gamma) = 2^{-w_j}$. Setting $f(\mu i) := \gamma$ thus satisfies the required property. \square

From this lemma it then follows that if the instance \mathcal{P} of the PCP has a solution $\mu \in \{1, \dots, m\}^*$, then every witnessed model \mathcal{I} of $\mathcal{K}_{\mathcal{P}}$ contains a node $\delta = f(\mu)$ such that $A^{\mathcal{I}}(\delta) = B^{\mathcal{I}}(\delta)$; that is, where A and B encode the same word. Conversely, if every witnessed model contains such a node, then in particular $\mathcal{I}_{\mathcal{P}}$ does, and thus \mathcal{P} has a solution. The question now is how to detect whether a node with this characteristics exists in every model.

7.1.4 Finding a solution

To detect whether there must exist a domain element that belongs to the concepts A and B to the same degree, we extend $\mathcal{T}_{\mathcal{P}}$ with axioms that further restrict $\mathcal{I}_{\mathcal{P}}$, and hence indirectly all models, to satisfy $A^{\mathcal{I}_{\mathcal{P}}}(\mu) \neq B^{\mathcal{I}_{\mathcal{P}}}(\mu)$ for every $\mu \in \{1, \dots, m\}^*$. This will ensure that the extended ontology will have a model iff \mathcal{P} has no solution.

Suppose for now that, for some $\mu \in \{1, \dots, m\}^*$, it holds that

$$2^{-v_\mu} = A^{\mathcal{I}_{\mathcal{P}}}(\mu) > B^{\mathcal{I}_{\mathcal{P}}}(\mu) = 2^{-w_\mu}.$$

We then have that $v_\mu < w_\mu$ and hence $w_\mu - v_\mu \geq 1$. It thus follows that

$$A^{\mathcal{I}\mathcal{P}}(\mu) \Rightarrow_{\Pi} B^{\mathcal{I}\mathcal{P}}(\mu) = 2^{-w_\mu} / 2^{-v_\mu} = 2^{-(w_\mu - v_\mu)} \leq 2^{-1} = 0.5.$$

Likewise, if $A^{\mathcal{I}\mathcal{P}}(\mu) < B^{\mathcal{I}\mathcal{P}}(\mu)$, we get $B^{\mathcal{I}\mathcal{P}}(\mu) \Rightarrow_{\Pi} A^{\mathcal{I}\mathcal{P}}(\mu) \leq 0.5$. Additionally, if we have $A^{\mathcal{I}\mathcal{P}}(\mu) = B^{\mathcal{I}\mathcal{P}}(\mu)$, then it is easy to verify that

$$A^{\mathcal{I}\mathcal{P}}(\mu) \Rightarrow_{\Pi} B^{\mathcal{I}\mathcal{P}}(\mu) = B^{\mathcal{I}\mathcal{P}}(\mu) \Rightarrow_{\Pi} A^{\mathcal{I}\mathcal{P}}(\mu) = 1.$$

From all this it follows that, for every $\mu \in \{1, \dots, m\}^*$,

$$A^{\mathcal{I}\mathcal{P}}(\mu) \neq B^{\mathcal{I}\mathcal{P}}(\mu) \quad \text{iff} \quad \begin{array}{l} \text{either} \quad A^{\mathcal{I}\mathcal{P}}(\mu) \Rightarrow_{\Pi} B^{\mathcal{I}\mathcal{P}}(\mu) \leq 0.5 \\ \text{or} \quad B^{\mathcal{I}\mathcal{P}}(\mu) \Rightarrow_{\Pi} A^{\mathcal{I}\mathcal{P}}(\mu) \leq 0.5. \end{array} \quad (3)$$

Thus, the instance \mathcal{P} has no solution iff for every $\mu \in \{1, \dots, m\}^*$ one of the two restrictions $A^{\mathcal{I}\mathcal{P}}(\mu) \Rightarrow_{\Pi} B^{\mathcal{I}\mathcal{P}}(\mu) \leq 0.5$ or $B^{\mathcal{I}\mathcal{P}}(\mu) \Rightarrow_{\Pi} A^{\mathcal{I}\mathcal{P}}(\mu) \leq 0.5$ is satisfied. We now define the TBox $\mathcal{T}'_{\mathcal{P}}$ that ensures this behavior in every model. Let

$$\mathcal{T}'_{\mathcal{P}} := \mathcal{T}_{\mathcal{P}} \cup \{ \langle \top \sqsubseteq \sim((A \rightarrow B) \sqcap (B \rightarrow A)), 0.5 \rangle \} \quad \mathcal{K}'_{\mathcal{P}} = \langle \mathcal{T}'_{\mathcal{P}}, \mathcal{A}_{\mathcal{P}} \rangle$$

This KB can be used to decide whether \mathcal{P} has a solution:

LEMMA 7.1.4. *The instance \mathcal{P} of the PCP has a solution iff the knowledge base $\mathcal{K}'_{\mathcal{P}}$ is not Π -consistent w.r.t. witnessed models.*

Proof. Assume that \mathcal{P} has a solution $\mu = i_1 \cdots i_k$ and let $u = v_\mu = w_\mu$. Suppose there is a witnessed model \mathcal{I} of $\mathcal{K}'_{\mathcal{P}}$. Since $\mathcal{T}_{\mathcal{P}} \subseteq \mathcal{T}'_{\mathcal{P}}$, \mathcal{I} is also a (witnessed) model of $\mathcal{K}_{\mathcal{P}}$. From Lemma 7.1.3 it follows that there is a node $\delta \in \Delta^{\mathcal{I}}$ where

$$A^{\mathcal{I}}(\delta) = A^{\mathcal{I}\mathcal{P}}(\mu) = B^{\mathcal{I}\mathcal{P}}(\mu) = B^{\mathcal{I}}(\delta).$$

Then, $A^{\mathcal{I}}(\delta) \Rightarrow_{\Pi} B^{\mathcal{I}}(\delta) = 1$ and $B^{\mathcal{I}}(\delta) \Rightarrow_{\Pi} A^{\mathcal{I}}(\delta) = 1$. This then implies that

$$(\sim((A \rightarrow B) \sqcap (B \rightarrow A)))^{\mathcal{I}}(\delta) = 0,$$

which violates the axiom $\langle \top \sqsubseteq \sim((A \rightarrow B) \sqcap (B \rightarrow A)), 0.5 \rangle$ from $\mathcal{T}'_{\mathcal{P}}$. Thus \mathcal{I} cannot be a model of $\mathcal{T}'_{\mathcal{P}}$, nor of $\mathcal{K}'_{\mathcal{P}}$.

For the converse, assume that $\mathcal{K}'_{\mathcal{P}}$ is not witnessed Π -consistent. Then $\mathcal{I}_{\mathcal{P}}$ cannot be a model of $\mathcal{K}'_{\mathcal{P}}$. Since it is a model of $\mathcal{K}_{\mathcal{P}}$, $\mathcal{I}_{\mathcal{P}}$ must violate the only axiom in $\mathcal{T}'_{\mathcal{P}} \setminus \mathcal{T}_{\mathcal{P}}$. Thus, there is a node $\mu \in \{1, \dots, m\}^*$ where $(\sim((A \rightarrow B) \sqcap (B \rightarrow A)))^{\mathcal{I}_{\mathcal{P}}}(\mu) < 0.5$ or, equivalently, where $(A^{\mathcal{I}_{\mathcal{P}}}(\mu) \Rightarrow_{\Pi} B^{\mathcal{I}_{\mathcal{P}}}(\mu)) \cdot (B^{\mathcal{I}_{\mathcal{P}}}(\mu) \Rightarrow_{\Pi} A^{\mathcal{I}_{\mathcal{P}}}(\mu)) > 0.5$. But this means that each of these residua is strictly greater than 0.5, and hence \mathcal{P} has a solution. \square

Lemma 7.1.4 implies that we can reduce the PCP to Π -consistency in the FDL \mathcal{JALCE} , and in particular, that the latter problem is undecidable.

THEOREM 7.1.5. *Witnessed Π -consistency of knowledge bases in \mathcal{JALCE} is undecidable.*

Observe that with obvious changes the proof of Theorem 7.1.5 is also valid for any FDL language of \mathcal{JALCE} type when taking as semantic of the complementation any involutive negation on $[0, 1]$ (different from the standard one). Moreover we cannot drop the condition of models being witnessed. Take the concept $B := \forall R.C \& \neg \forall R.(C \sqcap C)$, given by Hájek in [80], which is 1-satisfiable but not witnessed 1-satisfiable. In the same way the concept $\neg \Delta B$ is a tautology for witnessed models but not for all $[0, 1]$ -models.

7.2 Further undecidability results

Using similar techniques to those presented in Section 7.1, it is possible to prove undecidability of other reasoning tasks and for FDLs using other continuous t-norms for their semantics. The undecidability of $*$ -consistency holds even for less expressive FDLs. In fact, it was shown in [41, 60] that for every continuous t-norm $*$ that is not idempotent (i.e., not the minimum t-norm), $*$ -consistency is undecidable already in \mathcal{ELC} .

As explained before, the proof of this claim follows very closely the construction presented in Section 7.1 for reducing an instance of the PCP to consistency of a knowledge base. However, some additional technicalities are needed to remove any use of the constructors \forall and \rightarrow from the axioms used. As it can be seen from Figure 6, value restrictions are only used in the definition of the axioms $\langle C \overset{R}{\rightsquigarrow} D \rangle$. These axioms are used to transfer the membership degree of a concept to a given R -successor. We can obtain the same result using existential restrictions and strong negation, instead of value restrictions. To do this, simply redefine the expression $\langle C \overset{R}{\rightsquigarrow} D \rangle$ to stand for the two axioms $\langle \exists R. \sim D \sqsubseteq \sim C, 1 \rangle$ and $\langle \exists R. D \sqsubseteq C, 1 \rangle$. It is easy to see that Lemma 7.1.2 still holds under this definition.

The implication constructor is only used in the axiom

$$\langle \top \sqsubseteq \sim((A \rightarrow B) \sqcap (B \rightarrow A)), 0.5 \rangle$$

added to the knowledge base $\mathcal{K}_{\mathcal{P}}$ to verify the existence of a solution for \mathcal{P} . Recall that this axiom is intended to enforce that each element of the tree-model belongs to the concepts A and B to a different membership degree, and hence cannot be a solution. It is possible to remove the constructor \rightarrow with the help of auxiliary concept names that verify that A and B never receive the same interpretation. Since this construction is quite technical, we do not reproduce it here and refer the interested reader to the proofs appearing in [41]. Additionally, if the t-norm used is not Π , then the words from the instance \mathcal{P} need to be encoded in a different manner. Overall, we obtain the following theorem.

THEOREM 7.2.1. *Witnessed $*$ -consistency of \mathcal{ELC} knowledge bases is undecidable if $*$ is a non-idempotent continuous t-norm.*

Notice that strong negation is fundamental for the constructions presented so far. In fact, even for initializing the recursive construction of the tree, we need to ensure the existence of a domain element that belongs to the concepts A and B with specific degrees. Since concept assertions only provide lower bounds for the membership degrees, we require strong negation to provide a matching upper bound. It is well-known that for the Łukasiewicz t-norm, strong and weak negations coincide. Thus, a trivial consequence of Theorem 7.2.1 is that \mathbb{L} -consistency of \mathcal{NEL} knowledge bases is also undecidable. Observe that in fact, in Łukasiewicz, \mathcal{NEL} is equivalent to \mathcal{ALC} .

Let now $*$ be any non-strict continuous t-norm. It is well-known that such t-norms and their residua behave as the Łukasiewicz operators, scaled down to an initial subinterval from $[0, 1]$. More precisely, if $*$ is a non-strict continuous t-norm, then there exists a $b \in (0, 1]$ such that for all $x, y \in [0, b]$:

$$\begin{aligned} x * y &= b \cdot \left(\frac{x}{b} *_L \frac{y}{b} \right) \\ x \rightarrow_* y &= b \cdot \left(\frac{x}{b} \rightarrow_L \frac{y}{b} \right). \end{aligned}$$

Using this property, it is possible to scale the reduction of the PCP to \mathbb{L} -consistency down to the interval $[0, b]$, thus obtaining undecidability also for the nilpotent t-norm $*$. Clearly, in this scaling, we cannot consider strong negation, as there is no guarantee on how operators behave beyond the value b . Thus, this result can only hold if we allow weak negation, which is defined through the residuum and the lower bound 0.

THEOREM 7.2.2. *Witnessed $*$ -consistency of \mathfrak{NEL} knowledge bases is undecidable if $*$ is a non-strict continuous t-norm.*

We now briefly describe how to apply the same ideas described in this section to also show undecidability of subsumption and concept satisfiability. Recall that for the reduction presented in Section 7.1, we encoded each natural number v as the number $2^{-v} \in (0, 1]$. Notice that in this case, the base 2 is arbitrary, and the reduction would still apply, with minor modifications, if the encoded had the form p^v for any $p \in (0, 1)$. Moreover, the ABox $\mathcal{A}_{\mathcal{P}}$ is only used to initialize the root node to specific powers of $\frac{1}{2}$.

Suppose that we can guarantee the existence of an element δ that belongs to some concept name X to a degree $p \in (0, 1)$, then the TBox

$$\begin{aligned} \mathcal{T}_X &:= \{ \langle \top \sqsubseteq X, 0.1 \rangle, \langle \top \sqsubseteq \exists S.T \rangle, \langle X^{v_1} \overset{S}{\rightsquigarrow} A \rangle, \langle X^{w_1} \overset{S}{\rightsquigarrow} B \rangle \} \cup \\ &\quad \{ \langle X^{v_j} \overset{S}{\rightsquigarrow} V_j \rangle, \langle X^{w_j} \overset{S}{\rightsquigarrow} W_j \rangle \mid 1 \leq i, j \leq m \} \end{aligned}$$

ensures the existence of an element γ such that $A^{\mathcal{I}}(\gamma) = p^{v_1}$, $B^{\mathcal{I}}(\gamma) = p^{w_1}$, and $V_j^{\mathcal{I}}(\gamma) = p^{v_j}$ and $W_j^{\mathcal{I}}(\gamma) = p^{w_j}$ holds for every $j, 1 \leq j \leq m$; that is, this γ behaves like the root node for the search tree in the reduction of \mathcal{P} . Additionally, it guarantees that X is always interpreted to a degree greater than or equal to 0.1; in particular, at the element δ we have that $X^{\mathcal{I}}(\delta) > 0$. Using the arguments from Section 7.1, the following lemma can be easily proven.

LEMMA 7.2.3. *The instance \mathcal{P} of the PCP has a solution if and only if any of the following holds:*

1. X is Π -satisfiable to degree smaller than or equal to 0.9 w.r.t. $\mathcal{T}'_{\mathcal{P}} \cup \mathcal{T}_X$.
2. \top is not Π -subsumed by X to degree 1 w.r.t. $\mathcal{T}'_{\mathcal{P}} \cup \mathcal{T}_X$.

A simple consequence of this lemma is that Π -satisfiability and Π -subsumption are undecidable in \mathfrak{JALCE} . As before, these undecidability results can be strengthened to less expressive logics, and to a larger class of t-norms. We thus obtain the following theorem, whose detailed proof can be found in [2, 36].

THEOREM 7.2.4. *Witnessed $*$ -satisfiability and witnessed $*$ -subsumption w.r.t. TBoxes are undecidable*

- in \mathcal{ELC} if $*$ is a non-idempotent continuous t-norm and
- in \mathfrak{NEL} if $*$ is a non-strict continuous t-norm.

Notice that all these undecidability results hold only for witnessed models. In fact, the only step in which the properties of witnessed interpretations are used is to guarantee that there are R_i -successors with degree 1, using the axiom $\langle \top \sqsubseteq \exists R_i. \top \rangle$. Thus, we can weaken this restriction to consider only so-called *weakly-witnessed* interpretations. An interpretation is weakly-witnessed if, for every $\delta \in \Delta^{\mathcal{I}}$ such that $(\exists R. \top)^{\mathcal{I}}(\delta) = 1$, there is a $\gamma \in \Delta^{\mathcal{I}}$ with $R^{\mathcal{I}}(\delta, \gamma) = 1$.

When considering general models, the constructions presented so far do not work for encoding instances of the PCP. The reason for this is that the existential restriction can be satisfied through an infinite sequence of successors that get a degree arbitrarily close to 1, but this limit is never reached. In such a situation, the preconditions of Lemma 7.1.2 do not hold, and hence we cannot guarantee that the desired degrees are transferred to a specific successor.

To overcome this problem, it is proposed in [39] to encode words using intervals, rather than just numbers from $[0, 1]$. The idea is that such an interval allows for a margin of error in the transfer of values through successors with degrees close to, but not precisely equal to 1. While the main idea is the same, the additional handling of this error-margin makes the construction rather technical. Moreover, this idea has only been applied to non-strict t-norms, yielding the following result.

THEOREM 7.2.5. *For every non-strict t-norm $*$, $*$ -consistency, $*$ -satisfiability, and $*$ -subsumption are undecidable in \mathfrak{NEL} w.r.t. general models.*

7.3 Complexity bounds

We now study the complexity of reasoning in FDLs. For a historical overview of the known decidability results for these logics, we refer the interested reader to Section 8.

In order to study the complexity of reasoning in FDLs, recall first that in the classical case, reasoning in \mathcal{FL}_0 w.r.t. general TBoxes is already exponential [3, 133]; moreover, under the classical semantics, all the logics between \mathfrak{NEL} and \mathcal{ELC} from below, and \mathfrak{JALCE} from above are equivalent to the DL \mathcal{ALC} , for which the standard reasoning problems are all EXPTIME-complete. This implies that reasoning in the fuzzy variants of all these logics is already EXPTIME-hard, regardless of the t-norm chosen for their semantics. As shown earlier in this section, these lower complexity bounds are not necessarily tight, since in many cases these logics are undecidable. We now prove that in all other cases, KB consistency and concept satisfiability w.r.t. witnessed models are decidable in exponential time. Afterwards, we discuss the issues with general models, and the problem of subsumption.

Let $*$ be a strict continuous t-norm; in other words, a continuous t-norm with Gödel negation as its corresponding residuated negation. We want to show that $*$ -consistency of \mathfrak{NEL} knowledge bases is in EXPTIME. We will show a stronger result, proving this complexity upper bound for the more expressive \mathfrak{JALUE} .

Since $*$ is a strict t-norm, we know that its weak negation corresponds to the Gödel negation that maps 0 to 1 and all other values to 0. This means that its double negation $\neg\neg$ maps every positive value to 1 and the least element 0 to itself. Moreover, [62] proves that the double negation operator is an automorphism in any BL-chain. This result can be extended to existential and value restrictions, provided that the latter are witnessed.

LEMMA 7.3.1. *Let $*$ be a strict continuous t-norm. For all non-empty sets $X \subseteq [0, 1]$ it holds that*

1. $\neg\neg(\sup\{x \mid x \in X\}) = \sup\{\neg\neg x \mid x \in X\}$.
2. *If $\min\{x \mid x \in X\}$ exists, then $\neg\neg(\min\{x \mid x \in X\}) = \min\{\neg\neg x \mid x \in X\}$.*

Proof. To prove (1), observe that $\sup X = 0$ iff $X = \{0\}$, which yields

$$\begin{aligned} \neg\neg(\sup X) = 0 \quad \text{iff} \quad \sup X = 0 \quad \text{iff} \quad X = \{0\} \quad \text{iff} \quad \{\neg\neg x \mid x \in X\} = \{0\} \\ \text{iff} \quad \sup\{\neg\neg x \mid x \in X\} = 0. \end{aligned}$$

Assume now that $\min X = x_{\min}$ exists. To prove (2) observe that:

$$\begin{aligned} \neg\neg(\min X) = 0 \quad \text{iff} \quad x_{\min} = 0 \quad \text{iff} \quad 0 \in \{\neg\neg x \mid x \in X\} \\ \text{iff} \quad \min\{\neg\neg x \mid x \in X\} = 0. \quad \square \end{aligned}$$

In a natural way, the $\neg\neg$ operator induces, for each interpretation \mathcal{I} , the crisp interpretation \mathcal{I}^* defined by $A^{\mathcal{I}^*}(\delta) = \neg\neg(A^{\mathcal{I}}(\delta))$ and $R^{\mathcal{I}^*}(\delta, \gamma) = \neg\neg(R^{\mathcal{I}}(\delta, \gamma))$ for all $\delta, \gamma \in \Delta^{\mathcal{I}}$, concept name A and role name R . Let \mathcal{K} be a knowledge base. From \mathcal{K} we can construct a new (crisp) knowledge base \mathcal{K}^* . The axioms in \mathcal{K}^* are:

1. A concept assertion $\langle C(a), \neg\neg r \rangle$ for every concept assertion $\langle C(a), r \rangle$ in \mathcal{K}
2. A role assertion $\langle R(a, b), \neg\neg r \rangle$ for every role assertion $\langle R(a, b), r \rangle$ in \mathcal{K}
3. A GCI $\langle C \sqsubseteq D, \neg\neg r \rangle$ for every GCI $\langle C \sqsubseteq D, r \rangle$ in \mathcal{K} .

THEOREM 7.3.2. *A witnessed interpretation \mathcal{I} is a model of the \mathcal{K} if and only if \mathcal{I}^* is a model of \mathcal{K}^* .*

Since $*$ is strict, it follows that \mathcal{I}^* only uses the extreme membership degrees 0 and 1; that is, it is a crisp interpretation. Likewise, \mathcal{K}^* can be seen as a classical KB, since it is not graded. From this it follows that $*$ -consistency of $\mathcal{JALU}\mathcal{E}$ knowledge bases can be reduced in linear time to classical consistency in the DL \mathcal{ALC} , yielding the following result [34].

THEOREM 7.3.3. *If $*$ is a strict continuous t-norm then $*$ -consistency in $\mathcal{JALU}\mathcal{E}$ is in EXPTIME.*

The only remaining case is for logics that allow strong negation as a constructor. As stated in Theorem 7.2.1, in any logic that extends \mathcal{EL} with strong negation, $*$ -consistency is undecidable if $*$ is not the idempotent t-norm. Thus, we only need to study the case when the semantics is based on the minimum t-norm. It has been recently shown, using an automata-based method, that G-consistency of $\mathcal{JALC}\mathcal{E}$ can also be decided in

	\mathfrak{NEL}	\mathfrak{JELU}	\mathfrak{JALUE}	\mathcal{ELC}	\mathfrak{JALCE}
classical	$\mathbb{L}\oplus$	$\mathbb{L}\oplus$	$\mathbb{L}\oplus$	$\overline{\mathbb{G}}$	$\overline{\mathbb{G}}$
general	$\mathbb{L}\oplus$	$\mathbb{L}\oplus$	$\mathbb{L}\oplus$	$\overline{\mathbb{G}}$	$\overline{\mathbb{G}}$

Table 6. Undecidability of consistency in fuzzy description logics

exponential time [35]. The main insight in this approach is that the precise membership degrees are irrelevant for deciding whether a model exists; it is only necessary to know the *order* between these degrees. Since only a finite number of such orderings are possible, one can construct an automaton that accepts abstract representations of the tree-shaped models of the KB. For the full details of this construction see [35].

THEOREM 7.3.4. *G-consistency in \mathfrak{JALCE} is in EXPTIME.*

An upper bound in the complexity of $*$ -consistency also provides a complexity upper bound for the problem of $*$ -satisfiability of concepts to a degree greater than or equal to some r . Indeed, it is easy to see that a concept C is $*$ -satisfiable to degree at least r w.r.t. the knowledge base $\langle \mathcal{T}, \mathcal{A} \rangle$ if and only if the knowledge base $\langle \mathcal{T}, \{ \langle C(a), r \rangle \} \rangle$ is $*$ -consistent. If strong negation is allowed, then $*$ -satisfiability to a degree smaller than or equal to r and $*$ -subsumption can also be reduced to $*$ -consistency in the obvious manner. This yields the following result.

THEOREM 7.3.5. *G-satisfiability and G-subsumption in \mathfrak{JALCE} are in EXPTIME. Moreover, if $*$ is a strict continuous t-norm, then $*$ -consistency to a degree greater than or equal to r in \mathfrak{JALUE} is also in EXPTIME.*

Unfortunately, this reduction does not work for deciding satisfiability to a degree smaller or equal to some r , or subsumption. It is important to notice that the proof of decidability of consistency in \mathfrak{JALUE} constructs an equi-consistent classical KB from which consistency can be decided. However, this new knowledge base \mathcal{K}^* is not necessarily equivalent to \mathcal{K} ; the latter can have many more models than the former. The translation $\mathcal{K} \rightarrow \mathcal{K}^*$ guarantees that if there is any model, then there is also a model that uses only the degrees 0 and 1. However, nothing further is known of the existence of a model that gives a degree lower than some upper bound (for satisfiability), nor of all models (for subsumption). It is still an open problem whether these problems are even decidable for the \mathfrak{JALUE} description language.

The undecidability and complexity results for $*$ -consistency are summarized in Table 6. Each cell of the table describes the class of t-norms $*$ for which $*$ -consistency in the corresponding FDL is known to be undecidable, w.r.t. witnessed and general models. In this case, $\mathbb{L}\oplus$ denotes the class of non-strict continuous t-norms; that is, those t-norms starting with the Łukasiewicz t-norm, and $\overline{\mathbb{G}}$ is the class of non-idempotent continuous t-norms, i.e., all continuous t-norms except the minimum t-norm. Cells with a gray background express that for all t-norms that are *not* in the class, $*$ -consistency is known to be decidable and EXPTIME-complete. Notice that there are two white cells for the case of general models. In these cases, it is unknown whether the problem is decidable for any t-norm that is outside the respective class.

7.4 Reasoning with finite chains

We finish this section with a brief observation on the complexity of reasoning in FDLs when the semantics is based on a finite chain. It is easy to see, using standard techniques from model theory, that when the space of truth degrees is finite, then \mathcal{JALCC} has the bounded-model property; that is, if a knowledge base \mathcal{K} has a model, then it also has a finite model with size bounded by a function on the size of \mathcal{K} . This immediately implies decidability of all the standard reasoning tasks in this logic.

We can further improve this positive complexity result. In fact, using a technique known as *unraveling*, it is also possible to prove that if the knowledge base \mathcal{K} has a model, then it also accepts a Hintikka tree [35, 42, 44]. Abstract representations of these Hintikka trees can be accepted by an automaton that uses as states all the types for the relevant concepts from \mathcal{K} , and whose transition relation verifies that the semantics of the existential and value restrictions are satisfied. Since the number of types is exponential on the size of \mathcal{K} , and deciding whether an automaton accepts at least one tree is polynomial on its number of states, this yields the following result.

THEOREM 7.4.1. *Standard reasoning in \mathcal{JALCC} over finite chains is EXPTIME-complete.*

The result in Theorem 7.4.1 matches the same complexity result for classical \mathcal{ALC} . The same matching has been proved for different languages over (not necessarily linear) finite truth algebras. Nevertheless, in [33] has been proved that the complexity of KB consistency for language \mathcal{EL} under the finite Łukasiewicz t -norm jumps from PTIME-complete in the classical case to EXPTIME-complete in the finite-valued case.

8 Some historical remarks and further reading

We now provide a brief overview of the history of fuzzy description logics.

8.1 Classical Description Logics in short

Description Logics (DLs), previously called *Terminological Logics* (e.g. [100]) and *conceptual languages* (e.g. [114]), are usually considered as an evolution of *frame-based systems* (e.g. [84, 98]) and *Inheritance Networks* (e.g. [46, 107, 135, 139]), the latter two lacking a formal (logic based) semantics. DLs have been introduced to overcome to this deficiency. In short,⁹ the history of DLs can be split into four phases:

Phase 1. During the '80s, DLs' development has been characterised by so-called *structural subsumption algorithms* and implementation of first DL systems such as KL-ONE [51, 140], KL-TWO [138], KRYPTON [48], KANDOR [104], LOOM [93], BACK [101], NIKL [89], LASSIE [67], K.REP [96] and CLASSIC [47]. Some of these algorithms have been found incomplete and the subsumption problem may be undecidable (e.g. [105, 113]), but most of the cases these systems have reduced expressivity and computational complexity of the subsumption problem (e.g. [49, 115]).

⁹ For more on classical DLs' history see [4].

Phase 2. The early '90s have been characterized by the introduction of complete *tableaux based reasoning methods* [68, 85, 114], first complexity results, first investigation on optimization methods and systems such as KRIS [6], KSAT [74], CRACK [72], whose computational complexity of the subsumption problem is in PSPACE.

Phase 3. From the mid '90s, tableaux methods and systems have been developed with good practical performance on very expressive DLs with high worst case complexity, essentially in EXPTIME, and the study of relationships to other fragments of First-Order Logic [30, 65, 112]. Example systems of this period include FACT [87], DLP, RACER [76], CEL [7] and KAON2.¹⁰

Phase 4. This latest phase, is characterised by the development and application of mature, highly optimised implementations such as PELLET [117], HERMIT [116], FACT++ [137], RACERPRO [75], ELK [90], MASTRO STUDIO [63], used as the reasoning engines for knowledge representation languages in the realm of the Semantic Web, such as OWL 2 [102] and their profiles [103].

8.2 Fuzzy Description Logics

Initially, the work on *Fuzzy DLs* (FDLs) addressed the problem to equip DLs with a fuzzy semantics. A first step was to generalize the semantics of atomic concepts and roles from crisp sets and relations to fuzzy sets and fuzzy relations, respectively, and the semantics of subsumption to the inclusion between fuzzy sets. Nevertheless this does not mean there is a wide agreement on how to generalize the semantics of complex concepts and, thus, it is not surprising that various proposals have been made.

The first attempt in this direction is [141], which proposes a generalization of [49]. The language defined in this work, called $\mathcal{F}\mathcal{T}\mathcal{S}\mathcal{L}^-$, uses concept constructors conjunction ($:$ and C_1, \dots, C_n), value restriction ($:$ all RC), restricted existential quantification ($:$ some $R \top$), modifiers ($:$ NOT, $:$ VERY, $:$ SLIGHTLY, etc.) and an early version of concrete domains. The semantics was called *test score semantics* (see [142]) as *scores* (what we nowadays call “truth values”) are assigned to concepts. However, the interesting point are the truth functions used to calculate the truth values of complex concepts. Specifically,

- It is suggested to use the min function to compute the value of a conjunction of concepts. Besides this suggestion, the author not only recognizes that any other t-norm can be used as the semantics of conjunction, but also that both lower and upper bounds for conjunctions can be computed considering min and Łukasiewicz t-norms as upper and lower bounds respectively.
- The semantics of value restriction ($:$ all RC) is defined in two alternative ways. Either through a fuzzy implication operator, as it is done nowadays, or through the notion of conditional necessity from possibility theory. The author, however, adopts the second option.

¹⁰ However, it should be noted that KAON2 maps ontologies into disjunctive logic programs; <http://kaon2.semanticweb.org/#introduction>.

- The semantics of number restriction is defined just by means of a function that normalizes the sum of the relation degree of a given individual with respect to each other individual in the domain. As an example we report the definition of the *at-least number restriction*:

$$\mu_{(\text{at-least } n \ R)}(x) = \mu_{\text{at-least-}n} \left(\sum_y \mu_R(x, y) \right),$$

where μ_X is the characteristic function of the set (relation) X and $\mu_{\text{at-least-}n}$ is defined in the following way:

$$\mu_{\text{at-least-}n}(z) = \begin{cases} 0 & \text{if } z \leq n - 1 \\ z - n + 1 & \text{if } n - 1 \leq z \leq n \\ 1 & \text{if } z \geq n, \end{cases}$$

that is quite different from the solution adopted in the latest works on FDL.

The semantics defined in [141] was general enough to leave open the adoption of a truth function for conjunction. Nevertheless, this work was inspired by practical purposes and its goal was to provide a more refined tool for knowledge representation.

Later on, in [124, 136] a more theoretic approach was considered. The evolution of the notation towards a logical-like abstraction, that was experimented by the classical DL community, influenced these works, which utilizes the same modern notation reported in Section 3.1. The language studied in [136] was called \mathcal{ALC}_{F_M} (the subindex F_M stands for infinitely many truth values). It presents, as concept constructors, conjunction \sqcap , disjunction \sqcup , value restriction $\forall R.C$, existential quantifier $\exists R.C$ and *manipulators*¹¹ $M_i C$ [136], together with a many-valued logic style semantics [77]. The choice of the truth functions for the logical connectives falls on \min and \max for conjunction and disjunction, respectively. The semantics for the existential quantifier is the one provided in Definition 3.2.1 and it is the first place where it has been defined in this way. This work is also the first (together with [124]) in defining the semantics for value restriction $\forall R.C$ by means of the so-called *Kleene–Dienes implication*, defined on $[0, 1]$ as:

$$r \rightarrow s := \max\{1 - r, s\}$$

which is a straightforward generalization of the classical one. In particular, if \mathcal{I} is an FDL interpretation, the semantics of value restriction $\forall R.C$, based on Kleene–Dienes implication, is defined in the following way:

$$(\forall R.C)^{\mathcal{I}}(v) = \inf_{w \in M} \{ \max\{1 - R^{\mathcal{I}}(v, w), C^{\mathcal{I}}(w)\} \}.$$

Finally, for the semantics of manipulators $M_i C$, unary functions on $[0, 1]$ were used, in the same way as is done in the framework of Mathematical Fuzzy Logic when the semantics of *fuzzy hedges* is defined (see Chapter VIII of this Handbook and therein references for details).

¹¹ The notion of manipulators corresponds to the logical notion of *hedges modeling linguistic modifiers* used in this chapter.

In [124], the language studied is called (and, indeed, it is) \mathcal{ALC} and the semantics adopted is the same as the one used in [136], plus a unary function that gives the semantics to concept complementation defined as:

$$\neg r := 1 - r.$$

The set of operations that includes $\min\{r, s\}$, $\max\{r, s\}$, $\max\{1 - r, s\}$ and $1 - r$ on the real unit interval is commonly denoted with the name of *Zadeh's semantics*. The strength of [124] and of its journal version [125], is that they set up a clear syntax and semantics, very close to the classical ones and relate each other without the intermediate step of many-valued first order logic, like in [136]. In this way Fuzzy Description Logic is set up as an autonomous discipline with a clearly defined syntax and semantics as the one provided in [125].

Moreover, in [124] for the first time fuzzy axioms have been defined in this way, since in previous papers the notion of fuzzy axioms was not considered. In [124], just non-strict lower bound axioms are considered. In [125] axioms stating a non-strict upper bound have been introduced as well, but strict bound axioms are not considered. Since then, some works on FDL consider strict bound axioms, like [26, 122, 126, 131] and some others do not consider strict bound axioms, like [9, 15, 22, 38, 56, 58, 127, 129]. Later, in [127], the more general framework of semantics based on lattices has been considered that are supposed to be not necessarily chains, too.

These works, indeed, opened the door to the possibility of expanding the language in order to cover the advances that had been done in the classical framework. A fuzzy semantics for concrete domains was introduced in [130]. A semantics for unqualified number restriction, role hierarchies, inverse and transitive roles was introduced in [122]. A semantics for nominals was introduced in [121]. A semantics for qualified number restriction was introduced in [16, 17] and later on discussed and improved in [24, 27].

As it is known in the framework of MFL, the presence of Kleene–Dienes implication leads to some counter-intuitive facts due to its behavior with respect to the order in $[0, 1]$. Indeed if $x \rightarrow y = (1 - x) \vee y$ then the implication is true iff either $x = 0$ or $y = 1$ that is not intuitive. Moreover if $x \geq a$ and $x \rightarrow y \geq b$ nothing could be deduced about the value of y . This means that we have no Modus Ponens rule of any type. This fact, in the setting of FDLs based on Zadeh semantics, has been pointed out by Hájek in [80], where the example of the assertion “all hotels near to the main square are expensive” is presented in order to highlight the consequences of using Kleene–Dienes implication in the semantics of value restriction. Such assertion can be formally expressed as

$$\langle \forall \text{hasNear}.\text{Expensive}(\text{MainSquare}) \geq 1 \rangle. \quad (4)$$

Here we will further develop this example, as done in [54]. Consider the following fuzzy ABox HOTELS:

$$\begin{aligned} \langle \text{hasNear}(\text{MainSquare}, \text{Hotel}_1) = 0.9 \rangle & \quad \langle \text{Expensive}(\text{Hotel}_1) = 0.9 \rangle \\ \langle \text{hasNear}(\text{MainSquare}, \text{Hotel}_2) = 0.5 \rangle & \quad \langle \text{Expensive}(\text{Hotel}_2) = 0.5 \rangle \\ \langle \text{hasNear}(\text{MainSquare}, \text{Hotel}_3) = 0.1 \rangle & \quad \langle \text{Expensive}(\text{Hotel}_3) = 0.1 \rangle. \end{aligned}$$



Figure 7. Interpretation satisfying HOTELS

The HOTELS ABox indeed depicts the ideal situation, where “for each hotel the degree of its being near to the main square equals the degree of its being expensive” and where “there is at least one hotel which is near to the main square in degree 0.5”. In Figure 7 we report an interpretation that satisfies ABox HOTELS, where the distance between MainSquare and Hotel_x is calculated as

$$1 - \text{hasNear}(\text{MainSquare}, \text{Hotel}_x).$$

In this ideal situation the truth value of assertion (4) should be 1, because hotels are at least as expensive as they lie near the main square. Now, if the truth value of (4) is calculated using the truth function of any residuated implication, its value is indeed 1, which means that the ABox $\text{HOTELS} \cup \{(4)\}$ is consistent. Nevertheless, using the truth function of Kleene–Dienes implication, the result is different. In fact, in every interpretation \mathcal{I} that is a model of HOTELS, we have that:

$$\begin{aligned} & (\forall \text{hasNear.Expensive}(\text{MainSquare}))^{\mathcal{I}} = \\ &= \inf_{v \in \Delta^{\mathcal{I}}} \{ \text{Near}^{\mathcal{I}}(\text{MainSquare}^{\mathcal{I}}, v) \Rightarrow_* \text{Expensive}^{\mathcal{I}}(v) \} \leq \\ &\leq \inf \{ \max\{1 - 0.9, 0.9\}, \max\{1 - 0.5, 0.5\}, \max\{1 - 0.1, 0.1\} \} = \\ &= \inf \{ 0.9, 0.5, 0.9 \} = \\ &= 0.5. \end{aligned}$$

Hence, the truth value of assertion (4), using Kleene–Dienes implication, is at most 0.5 in every model of HOTELS, against the intuition, reflected in HOTELS, that its truth value should be 1. Hence, the ABox $\text{HOTELS} \cup \{(4)\}$ is inconsistent.

For these reasons, many-valued logics studied in the framework of MFL use residuated implications (satisfying $x \leq y$ iff $x \rightarrow y = 1$ and a Modus Ponens rule saying that if $x \geq a$ and $x \rightarrow y \geq b$ then $y \geq a * b$). Based on MFL, Hájek in [80] defines logical based FDLs as fragments of first order residuated many-valued logics in an analogous way as DLs are fragments of first order classical logic. In this new framework, not only the semantics of conjunction is a t-norm, but it is also recovered the idea, firstly proposed in [124, 136], of a tight relation between FDLs and first order fuzzy logic. As we have seen, with the only exception of [141], the operation min is the only function adopted as a semantics for the conjunction concept constructor until [80]. Hájek’s proposal defines a family of FDLs depending on the t-norm used to interpret strong conjunction. In the same year, Straccia defined and studied in [128–130] some FDLs where conjunction is interpreted as a t-norm different from the minimum as well.

The new framework inspired several successive works on FDL such as, [10, 18, 22, 24, 25, 27, 55, 131]. Among the ones that deepen the relationships between FDL and MFL we can find [56, 58, 59, 73].

The new framework of FDLs based on t-norms supposed also a re-thinking about the notation used in FDLs. Indeed, the use of the same notation of DLs for FDLs has been based on the fact that, in order to generalize DLs to the multi-valued framework, it seemed enough to generalize the semantics of concepts and roles to fuzzy sets and fuzzy relations. With this idea it is obvious that the same concept constructors (and, with them, the same formal languages) could be maintained in a multi-valued framework. This formalization worked indeed well when the semantics adopted as underlying truth value algebra was the Zadeh's semantics. But, since [80, 128–130], some works on FDLs began to consider the use of residuated implications as formalized in the framework of MFL. However, adopting a framework based on MFL and maintaining the same notation as in the classical case, could produce a slight confusion. This is due to several reasons related to differences between the classical and the many-valued framework. Commonly, with some exceptions, such differences include the following items:

1. Two kinds of conjunctions, with different mathematical properties, can be considered in the many-valued framework and the same holds for disjunction.
2. Implication is, in general, not definable from other connectives.
3. The quantifiers are not definable from each other by means of the duality through negation $\exists R.C \equiv \neg \forall R. \neg C$.
4. The disjunction is not definable from the residuated negation $\neg C := C \rightarrow \perp$ and the conjunction \sqcap .

All these items must be taken into account both when choosing the symbols denoting the constructors of our description languages and when building the hierarchy of fuzzy description languages, as we have already seen in Section 3.3. As an example recall that, in classical DLs, $\mathcal{AL}\mathcal{E}$ is strictly contained in $\mathcal{AL}\mathcal{C}$, while within many FDLs, by item 3 above, this is not the case.

In particular, we find worth discussing the case of implication. In classical DLs, the implication is not usually a primitive concept constructor, even though implication is often implicitly used. This is due to the fact that the implication is definable from conjunction and negation. Nevertheless, in the logic BL and many of its extensions, implication is in general not definable from other connectives. The first time that the concept constructor \rightarrow for the implication is included in the definition of the language as a primitive connective has been in [80].

The introduction of a new symbol for implication allows to utilize a concept constructor that is not otherwise definable. Moreover, it is useful in order to define, in BL and its extensions, other concept constructors like weak conjunction, weak disjunction and residuated negation.

8.3 Computational issues in a historical perspective

Like for classical DLs, also in the framework of t-norm based FDL the computational issues have marked the recent history and the development of the subject. Computational issues have been already extensively addressed in Sections 6 and 7. Here we want to summarize them in a historical perspective. For this reason, in this section we

omit the technical details, the definitions of the notions cited and any explanation of how they work. For a deeper understanding of these notions the reader is addressed to Sections 6 and 7.

Decidability has been a central matter since the beginning of the research in FDL. As in the classical case, the first problem to be dealt with has been concept subsumption. In order to deal with this problem a structural subsumption based procedure has been employed. In [49] a structural subsumption algorithm $\text{SUBS?}[a, b]$ for deciding classical concept subsumption in \mathcal{FL}^- with respect to an empty KB is presented. This kind of algorithms perform a comparison in the syntactic structure of two given concept descriptions after having transformed them in a suitable normal form. The fact that $\text{SUBS?}[a, b]$ calculates subsumption in polynomial time relies on the fact that every \mathcal{FL}^- concept C is equivalent to a \mathcal{FL}^- concept C^* where each value restriction $\forall R.C$ appears at most once for each nesting level. That is, for example concepts $\forall R.(C \sqcap D)$ and $\forall R.C \sqcap \forall R.D$ are equivalent. Moreover, the possibility of applying structural algorithms to a given calculus is due to the fact that concept conjunctions can be considered as sets of concepts. The problem of generalizing structural subsumption algorithms to the many-valued framework has been addressed in [141] and [136]. In [141] it is proven that the concept subsumption problem, with respect to empty knowledge bases, of a language denoted as \mathcal{FDSL}^- is decidable. The main result in [136] is the decidability proof for the concept subsumption problem, with respect to empty knowledge bases, of a language denoted as \mathcal{ALC}_{FM} . The structural subsumption algorithm $\text{SUBS?}[a, b]$ can be indeed consistently used in order to decide 1-subsumption¹² for $\mathbf{G}_n\text{-}\mathcal{FL}^-$. This is due to the fact that the Gödel t -norm \wedge works well with its residuum $\Rightarrow_{\mathbf{G}_n}$. That is, for every $x, y, z \in \mathbf{G}_n$ it holds that $x \Rightarrow_{\mathbf{G}_n} (y \wedge z) = (x \Rightarrow_{\mathbf{G}_n} y) \wedge (x \Rightarrow_{\mathbf{G}_n} z)$. Unfortunately, the same result does not hold for non-idempotent t -norms. In this case, there may be $x, y, z \in T$ such that $x \Rightarrow_* (y * z) \neq (x \Rightarrow_* y) * (x \Rightarrow_* z)$. In [61] the problem has been studied for the 1-subsumption problem with empty KB under the finite Łukasiewicz t -norm $*_{L_n}$. Since Łukasiewicz conjunction is not idempotent, complex concepts where just \sqcap appears as concept constructor can not be seen as sets of atomic concepts. Nevertheless, as shown in [61], complex concepts in $\mathbf{L}_n\text{-}\mathcal{FL}^-$ can be seen as *multisets* of simpler concepts, that is, different *occurrences* of atomic concepts are now seen as different elements of a given complex concept. This gives us the possibility of still defining structural subsumption algorithms for $\mathbf{L}_n\text{-}\mathcal{FL}^-$.

In [124] and [125] the decidability of the entailment problem with respect to empty TBoxes of language \mathcal{ALC} under Zadeh's semantics is proven. The procedure used is a tableaux algorithm based on the recursive production of a set of constraints until either a clash is produced or the set of constraint is complete, in the sense that no further rule can be applied to the existing set of constraints. Some years later, more general tableau algorithms were presented, managing GCIs and the more expressive FDLs *SHIN* [123] and *SHOIQ* [120] under Zadeh's semantics. The reasoner *Fire* implements a tableaux algorithm to solve several reasoning tasks in Zadeh *SHIN* [118]. If the TBox is empty, these algorithms have the same computational complexity as the tableaux algorithms

¹² Note that subsumption between two concepts in $\mathbf{G}_n\text{-}\mathcal{FL}^-$ always takes either value 0 or value 1. Therefore, speaking about $(\geq r)$ - or $(= r)$ -subsumption in $\mathbf{G}_n\text{-}\mathcal{FL}^-$ does not make sense.

for their corresponding crisp languages. If there are GCIs, it is conjectured that the algorithms do not provide a tight complexity bound. Tableau algorithms can also be used for finite t-norms if the degrees of truth are taken from a residuated De Morgan lattice. The idea here is that all the tableau rules can be applied in a non-deterministic way by producing all the combinations of degrees of truth that satisfy a restriction. An algorithm for *-consistency in \mathcal{SHI} without GCIs has been presented in [44]. Since there are exponentially many rule applications and exponentially many ways to apply a single rule, it is conjectured that this algorithm does not provide a tight complexity bound.

As explained in Section 6.2.4, the decidability of the reasoning with several FDLs have been proved by a reduction to reasoning with classical DLs. In [126] KB consistency of the language \mathcal{ALCH} with role hierarchies over Zadeh's semantics is reduced to reasoning with classical \mathcal{ALCH} . An algorithm for Z-consistency in the more expressive $\mathcal{SROIQ}(\mathbf{D})$ has also been presented in [18]. If one restricts his attention to finite chains, there are algorithms for G-consistency [20] and L-consistency [27] in \mathcal{SROIQ} . These results have been extended to *-consistency for every finite t-norm [21, 29] (where it is also possible to combine operators belonging to different finite fuzzy logics). The DeLorean reasoner implements a reasoning algorithm for fuzzy extensions of OWL 2 using Zadeh and Gödel fuzzy logics [19].

In [128] a procedure based on a reduction to the Mixed Integer Linear Programming problem is proposed in order to prove decidability of language \mathcal{ALC} extended with concrete domains and fuzzy modifiers under Zadeh's semantics or Łukasiewicz semantics for *acyclic TBoxes*. This procedure has been explained in Section 6.2.1. This family of algorithms based on a combination of tableaux rules and optimization problems has been used to solve the BED problem in more expressive logics. An algorithm to solve the witnessed BED problem in \mathcal{JALC} without TBoxes for any left-continuous t-norm was proposed in [25]. This result was later extended to consider acyclic TBoxes [60]. Similarly, an algorithm checking the *-consistency in \mathcal{SI} without TBoxes for any continuous t-norm was proposed in [119]. These algorithms can be extended to support fuzzy aggregation operators as shown in [28]. fuzzyDL [23] reasoner uses one of these algorithms to solve several reasoning tasks for Zadeh and Łukasiewicz $\mathcal{SHIF}(\mathbf{D})$.

Until [80], the algebra of truth values considered is mainly the real unit interval $[0,1]$ with operations \max , \min , $1 - x$ and Kleene–Dienes implication (also known as the Zadeh semantics). In [80] and [128–130], for the first time a t-norm based semantics is considered. Specifically, the main result proved in [80] is the decidability of the satisfiability and subsumption problems, with respect to empty knowledge bases, for language \mathcal{ALC} over a standard algebra restricted to witnessed models. The result is achieved by means of a recursive reduction of the concept satisfiability and subsumption to satisfiability and logical consequence in the corresponding propositional calculus. This procedure has been explained in Section 6.2.5. Other works where the same research line is applied are [55] and [45]. The first one tried to prove decidability of concept satisfiability in $\Pi\text{-}\mathcal{JALUE}$ without knowledge bases and with respect to so-called *quasi-witnessed models*. In recent times an error was found in the proof, but a correct proof can be found in [53]. The second one proposes an implementation in PSPACE of Algorithm 5 in the case of any finite t-norm.

Further works on the decidability of FDLs are the following. In [21] a quite expressive FDL over a set of operations that join minimum t-norm with Zadeh's operations is presented. The decidability of the KB consistency problem is proved through a recursive reduction to classical DL when the set of truth values is fixed in advance and finite. A similar result can be obtained by considering degrees of truth taken from a residuated De Morgan lattice [44]. In [34] it is proved that KB consistency for an FDL with role constructors and nominals over any t-norm with Gödel negation is linearly reducible to classical DL. In [39] it is proved undecidability of KB consistency with respect to unrestricted interpretations for a really basic FDL over Łukasiewicz t-norm. In [40] different results are provided for quite expressive FDLs over complete residuated De Morgan lattices, including some decidability results involving finite lattices. An interesting feature of this paper is the use of tableaux algorithms for the decidability results.

In the last years there have been several works dealing with undecidability of FDLs over a BL-chain determined by a continuous t-norm. Early works on FDLs based on non-idempotent t-norms, are spotted by examples of languages that do not enjoy the *finite model property* (FMP), that is a fundamental property to easily prove decidability of a calculus. In [80] it is proven that FMP fails with general models for the concept satisfiability problem without TBox in $[0, 1]_{\Pi}$ - \mathcal{JALC} . In [25] the lack of FMP is proved for knowledge base consistency in $[0, 1]_{\mathbb{L}\Pi}$ - \mathcal{ALC} . At that time a proof of undecidability was still not taken into account, because the lack of FMP was not considered to be a serious problem. On the other hand, in [25] the problem is restricted to acyclic knowledge bases and witnessed models in order to easily obtain decidability. Some years later, it was proven that the languages $[0, 1]_{\mathbb{L}}\text{-}\mathcal{ALC}$ and $[0, 1]_{\Pi}\text{-}\mathcal{JALCC}$ do not enjoy the finite model property when general TBoxes are considered, even if we consider witnessed models only [15]. Although this work does not give any undecidability result, it casts doubts on the decidability of those languages when general TBoxes are considered. The first actual undecidability result is given in [9], where undecidability of KB consistency for language $[0, 1]_{\Pi}\text{-}\mathcal{JALCC}$ is proved. The method used in this work as well as in the subsequent works on undecidability is a reduction to PCP similar to the one explained in Section 7. However, in order to prove undecidability, this work requires the use of inclusion axioms with a strict lower bound; but the authors were not able to strengthen the construction to prove undecidability of the same problem without this kind of axioms. Nevertheless the full undecidability of KB consistency for language $[0, 1]_{\Pi}\text{-}\mathcal{JALCC}$ was proved later in [10] and [11]. Note that in these works the undecidability result is proved for the cited problem when restricted both to witnessed interpretations and to strongly witnessed interpretations. In [38] the undecidability of concept satisfiability with respect to a KB for the language \mathcal{ALC} over a De Morgan chain containing the standard Łukasiewicz chain as a subalgebra is proved. In the same paper [38] it is also shown that if the De Morgan lattice considered as underlying algebra of truth values is finite, the same problem is indeed decidable. The result is achieved through a recursive reduction to a decidable problem from automata theory. Along the same line, in [60] similar results under Łukasiewicz semantics for \mathcal{ALC} are proven. In [41] a systematic study of undecidability in FDL is undertaken yielding sufficient conditions for proving undecidability in a wide class of FDLs. A quite complete overview of the decidability and undecidability results known for DLs can be found in [2, 36].

Computational complexity is a problem that has been addressed only in recent years. In [31] it is proved that concept witnessed r -satisfiability with respect to a general TBox for language \mathcal{ALC} over (not necessarily linear) finite De Morgan lattices is EXPTIME-complete. The result is proved by means of a reduction to automata theory. Notice that in [31] the truth function considered for concept constructors are min and max for conjunction and disjunction respectively, a unary function that satisfies the De Morgan laws for the negation and Kleene–Dienes implication. This implies that the implication constructor and the existential quantifier are definable like in Zadeh’s semantics. In [38] the result of [31] is enhanced by adding a finite t-norm to the operations of the De Morgan lattice and using its residuum in the semantics of the value restriction and of inclusion axioms. With such semantics, the existential quantifier is no more definable as in the De Morgan lattices considered in [31]. So, the language considered in [38] is \mathcal{ALCE} . Nevertheless, concept satisfiability with respect to general knowledge bases is proved to be EXPTIME-complete. Again, the result is proved by means of a reduction to automata theory. The semantics considered in [37] is the same as in [38] and the language is \mathcal{ALCE} with inverse roles. By means of a reduction to automata theory it is proved that entailment of lower bound inclusion axioms by general knowledge bases is EXPTIME-complete. The same problem is proved to be PSPACE-complete if the TBox is acyclic. Moreover, concept satisfiability with respect to acyclic TBoxes is proved to be PSPACE-complete as well. Again, the proof is based on a recursive reduction to automata theory. The result in [37] generalizes the one in [38], not only because the language considered is more expressive, but also because the concept subsumption problem is considered, that was not considered in [38]. Moreover the problem of concept satisfiability with respect to acyclic TBoxes is considered.

The result in Theorem 7.4.1 matches the same complexity result for classical \mathcal{ALC} . The same matching has been proved for different languages, e.g., for \mathcal{ALC} over (not necessarily linear) finite De Morgan lattices [31], for \mathcal{ALCE} over finite De Morgan lattices with an added t-norm [38], for \mathcal{ALCE} with inverse roles over finite De Morgan lattices [37], for \mathcal{JALCE} with the empty KB over finite chains [45] and for \mathcal{FL}^- with an empty KB under the finite Łukasiewicz t -norm [61]. These are all languages whose complexity of the respective reasoning tasks has been proved to match the classical results. Nevertheless, in [33] has been proved that the complexity of KB consistency for language \mathcal{EL} under the finite Łukasiewicz t -norm does not match with the complexity of the same problem for classical \mathcal{EL} . Indeed, KB consistency jumps from PTIME-complete in the latter case to EXPTIME-complete in the former case.

Some complementary related work about FDLs can be found in [32, 54, 132].

Acknowledgements

We want to thank Brunella Gerla and Ulrike Sattler for their reading of the first version of the chapter and their helpful comments and corrections.

Fernando Bobillo acknowledges support by two Spanish CICYT projects: TIN2012-30939 and TIN2013-46238-C4-4-R.

Marco Cerami acknowledges support by the european project POST UP II, number CZ.1.07/2.3.00/30.0041. The present work has been written while this author was working as a postdoctoral researcher at University Palacký in Olomouc, under this project.

Francesc Esteva acknowledges support by Spanish MICINN project EdeTRI (TIN 2012-39348-C02-01) and grant 2014SGR-118 from the Generalitat de Catalunya.

Àngel García-Cerdaña acknowledges support by Spanish MICINN projects EdeTRI (TIN2012-39348-C02-01) and MTM 201125747, the CSIC project 201450E045 and grant 2014SGR-788 from the Generalitat de Catalunya.

Rafael Peñaloza has been partially supported by the German Research Foundation (DFG) under grant BA 1122/17-1 (FuzzyDL) and within the Cluster of Excellence ‘Center for Advancing Electronics Dresden’ (cfAED). Most of the work presented here was developed while he was still affiliated to TU Dresden and the Center for Advancing Electronics Dresden, Germany.

BIBLIOGRAPHY

- [1] Franz Baader. Using automata theory for characterizing the semantics of terminological cycles. *Annals of Mathematics and Artificial Intelligence*, 18(2–4):175–219, 1996.
- [2] Franz Baader, Stefan Borgwardt, and Rafael Peñaloza. On the decidability status of fuzzy \mathcal{ALC} with general concept inclusions. *Journal of Philosophical Logic*, 44(2):117–146, 2015.
- [3] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the \mathcal{EL} envelope. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005)*, pages 364–369. Morgan Kaufmann, 2005.
- [4] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, New York, NY, USA, 2003.
- [5] Franz Baader, Jan Hladik, and Rafael Peñaloza. Automata can show PSPACE results for description logics. *Information and Computation*, 206(9–10):1045–1056, 2008.
- [6] Franz Baader and Bernhard Hollunder. KRIS: Knowledge representation and inference system, system description. *ACM SIGART Bulletin*, 2:8–14, 1991.
- [7] Franz Baader, Carsten Lutz, and Boontawee Suntisrivaraporn. CEL—A polynomial-time reasoner for life science ontologies. In Ulrich Furbach and Natarajan Shankar, editors, *Automated Reasoning. Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR 2006)*, volume 4130 of *Lecture Notes in Artificial Intelligence*, pages 287–291. Springer, 2006.
- [8] Franz Baader and Werner Nutt. Basic description logics. In Franz Baader, Diego Calvanese, Deborah L. McGuinness, Danielle Nardi, and Peter F. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation, and Applications*, pages 47–100. Cambridge University Press, 2003.
- [9] Franz Baader and Rafael Peñaloza. Are fuzzy description logics with general concept inclusion axioms decidable? In *Proceedings of the 20th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, pages 1735–1742. IEEE Press, 2011.
- [10] Franz Baader and Rafael Peñaloza. GCIs make reasoning in fuzzy DL with the product t -norm undecidable. In Riccardo Rosati, Sebastian Rudolph, and Michael Zakharyashev, editors, *Proceedings of the 24th International Workshop on Description Logics (DL 2011)*, volume 745, pages 37–47. CEUR Workshop Proceedings, 2011.
- [11] Franz Baader and Rafael Peñaloza. On the undecidability of fuzzy description logics with gcis and product t -norm. In Cesare Tinelli and Viorica Sofronie-Stokkermans, editors, *Proceedings of the 8th International Symposium Frontiers of Combining Systems (FroCos 2011)*, number 6989 in *Lecture Notes in Artificial Intelligence*, pages 55–70. Springer, 2011.
- [12] Franz Baader and Ulrike Sattler. An overview of tableau algorithms for description logics. *Studia Logica*, 69:5–40, 2001.
- [13] Matthias Baaz, Christian G. Fermüller, Gernot Salzer, and Richard Zach. Labeled calculi and finite-valued logics. *Studia Logica*, 61(1):7–33, 1998.
- [14] Libor Běhounek, Petr Cintula, and Petr Hájek. Introduction to mathematical fuzzy logic. In Petr Cintula, Petr Hájek, and Carles Noguera, editors, *Handbook of Mathematical Fuzzy Logic - Volume 1*, volume 37 of *Studies in Logic, Mathematical Logic and Foundations*, pages 1–101. College Publications, London, 2011.
- [15] Fernando Bobillo, Félix Bou, and Umberto Straccia. On the failure of the finite model property in some fuzzy description logics. *Fuzzy Sets and Systems*, 172(1):1–12, 2011.

- [16] Fernando Bobillo, Miguel Delgado, and Juan Gómez-Romero. A crisp representation for fuzzy *SHOLN* with fuzzy nominals and general concept inclusions with fuzzy nominals and general concept inclusions. In Paulo Cesar G. da Costa, Kathryn B. Laskey, Kenneth J. Laskey, Francis Fung, and Michael Pool, editors, *Proceedings of the 2nd International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2006)*, volume 218. CEUR Workshop Proceedings, 2006.
- [17] Fernando Bobillo, Miguel Delgado, and Juan Gómez-Romero. Optimizing the crisp representation of the fuzzy description logic *SROLQ*. In Fernando Bobillo, Paulo Cesar G. da Costa, Claudia d'Amato, Nicola Fanizzi, Francis Fung, Thomas Lukasiewicz, Trevor Martin, Matthias Nickles, Yun Peng, Michael Pool, Pavel Smrž, and Peter Vojtás, editors, *Proceedings of the 3rd International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2007)*. CEUR Workshop Proceedings, 2007.
- [18] Fernando Bobillo, Miguel Delgado, and Juan Gómez-Romero. Crisp representation and reasoning for fuzzy ontologies. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 17(4):501–530, 2009.
- [19] Fernando Bobillo, Miguel Delgado, and Juan Gómez-Romero. DeLorean: A reasoner for fuzzy OWL 2. *Expert Systems with Applications*, 39:258–272, 2012.
- [20] Fernando Bobillo, Miguel Delgado, Juan Gómez-Romero, and Umberto Straccia. Fuzzy description logics under Gödel semantics. *International Journal of Approximate Reasoning*, 50(3):494–514, 2009.
- [21] Fernando Bobillo, Miguel Delgado, Juan Gómez-Romero, and Umberto Straccia. Joining Gödel and Zadeh fuzzy logics in fuzzy description logics. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 20(4):475–508, 2012.
- [22] Fernando Bobillo and Umberto Straccia. A fuzzy description logic with product t-norm. In *Proceedings of the 16th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2007)*, pages 652–657. IEEE Computer Society, 2007.
- [23] Fernando Bobillo and Umberto Straccia. fuzzyDL: An expressive fuzzy description logic reasoner. In *Proceedings of the 17th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2008)*, pages 923–930. IEEE Xplore. Digital Library, 2008.
- [24] Fernando Bobillo and Umberto Straccia. Towards a crisp representation of fuzzy description logics under Lukasiewicz semantics. In Aijun An, Stan Matwin, Zbigniew W. Raś, and Dominik Ślezak, editors, *Proceedings of the 17th International Symposium on Methodologies for Intelligent Systems (ISMIS 2008)*, volume 4994 of *Lecture Notes in Computer Science*, pages 309–318. Springer, 2008.
- [25] Fernando Bobillo and Umberto Straccia. Fuzzy description logics with general t-norms and datatypes. *Fuzzy Sets and Systems*, 160(23):3382–3402, 2009.
- [26] Fernando Bobillo and Umberto Straccia. Fuzzy ontology representation using OWL 2. *International Journal of Approximate Reasoning*, 52:1073–1094, 2011.
- [27] Fernando Bobillo and Umberto Straccia. Reasoning with the finitely many-valued Łukasiewicz fuzzy description logic *SROLQ*. *Information Sciences*, 181:758–778, 2011.
- [28] Fernando Bobillo and Umberto Straccia. Aggregation operators for fuzzy ontologies. *Applied Soft Computing*, 13:3816–3830, 2013.
- [29] Fernando Bobillo and Umberto Straccia. Finite fuzzy description logics and crisp representations. In Fernando Bobillo, Paulo Cesar G. da Costa, Claudia d'Amato, Nicola Fanizzi, Kathryn Laskey, Ken Laskey, Thomas Lukasiewicz, Matthias Nickles, and Michael Pool, editors, *Uncertainty Reasoning for the Semantic Web II*, volume 7123 of *Lecture Notes in Computer Science*, pages 102–121. Springer, 2013.
- [30] Alex Borgida. On the relative expressiveness of description logics and predicate logics. *Artificial Intelligence*, 82:353–367, 1996.
- [31] Stefa Borgwardt and Rafael Peñaloza. Description logics over lattices with multi-valued ontologies. In Toby Walsh, editor, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 768–773. AAAI Press, 2011.
- [32] Stefa Borgwardt. *Fuzzy Description Logics with General Concept Inclusions*. PhD thesis, Technische Universität Dresden, 2014.
- [33] Stefa Borgwardt, Marco Cerami, and Rafael Peñaloza. The complexity of subsumption in fuzzy \mathcal{EL} . In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 2812–2818. AAAI Press, 2015.
- [34] Stefa Borgwardt, Felix Distel, and Rafael Peñaloza. Gödel negation makes unwitnessed consistency crisp. In Yevgeny Kazarov, Domenico Lembo, and Frank Wolter, editors, *Proceedings of the 25th International Workshop on Description Logics (DL 2012)*, volume 846, pages 103–113. CEUR Workshop Proceedings, 2012.

- [35] Stefan Borgwardt, Felix Distel, and Rafael Peñaloza. Decidable Gödel description logics without the finitely-valued model property. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Proceedings of the 14th Conference on Principles of Knowledge Representation and Reasoning (KR 2014)*, pages 228–237. AAAI Press, 2014.
- [36] Stefan Borgwardt, Felix Distel, and Rafael Peñaloza. The limits of decidability in fuzzy description logics with general concept inclusions. *Artificial Intelligence*, 218:23–55, 2015.
- [37] Stefan Borgwardt and Rafael Peñaloza. Finite lattices do not make reasoning in \mathcal{ALCC} harder. In Fernando Bobillo, Rommel Carvalho, Paulo Cesar G. da Costa, Claudia d’Amato, Nicola Fanizzi, Kathryn B. Laskey, Kenneth J. Laskey, Thomas Lukasiewicz, Trevor Martin, Matthias Nickles, and Michael Pool, editors, *Proceedings of the 7th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2011)*, volume 778, pages 51–62. CEUR Workshop Proceedings, 2011.
- [38] Stefan Borgwardt and Rafael Peñaloza. Fuzzy ontologies over lattices with t-norms. In Riccardo Rosati, Sebastian Rudolph, and Michael Zakharyashev, editors, *Proceedings of the 24th International Workshop on Description Logics (DL 2011)*, pages 70–80. CEUR Workshop Proceedings, 2011.
- [39] Stefan Borgwardt and Rafael Peñaloza. Non-Gödel negation makes unwitnessed consistency undecidable. In Yevgeny Kazarov, Domenico Lembo, and Frank Wolter, editors, *Proceedings of the 25th International Workshop on Description Logics (DL 2012)*, volume 846, pages 411–421. CEUR Workshop Proceedings, 2012.
- [40] Stefan Borgwardt and Rafael Peñaloza. A tableau algorithm for fuzzy description logics over residuated De Morgan lattices. In Markus Kroetzsch and Umberto Straccia, editors, *Proceedings of the 6th International Conference on Web Reasoning and Rule Systems (RR 2012)*, volume 7497 of *Lecture Notes in Computer Science*, pages 9–24. Springer, 2012.
- [41] Stefan Borgwardt and Rafael Peñaloza. Undecidability of fuzzy description logics. In Gerhard Brewka, Thomas Eiter, and Sheila A. McIlraith, editors, *Proceedings of the 13th Conference on Principles of Knowledge Representation and Reasoning (KR 2012)*, pages 232–242. AAAI Press, 2012.
- [42] Stefan Borgwardt and Rafael Peñaloza. The complexity of lattice-based fuzzy description logics. *Journal on Data Semantics*, 2(1):1–19, 2013.
- [43] Stefan Borgwardt and Rafael Peñaloza. Positive subsumption in fuzzy \mathcal{EL} with general t-norms. In Francesca Rossi, editor, *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)*, pages 789–795. AAAI Press, 2013.
- [44] Stefan Borgwardt and Rafael Peñaloza. Consistency reasoning in lattice-based fuzzy description logics. *International Journal of Approximate Reasoning*, 55(9):1917–1938, 2014.
- [45] Félix Bou, Marco Cerami, and Francesc Esteva. Concept satisfiability in finite-valued fuzzy description logics is PSPACE-complete. In *Logic, Algebra and Truth Degrees 2012*, pages 44–48, 2012.
- [46] Ronald J. Brachman. On the epistemological status of semantic networks. In Nicholas V. Findler, editor, *Associative Networks*, pages 3–50. Academic Press, 1979.
- [47] Ronald J. Brachman, Alexander Borgida, Deborah L. McGuinness, and Lori A. Resnick. CLASSIC: A structural data model for objects. In James Clifford, Bruce Lindsay, and David Maier, editors, *Proceedings of the 1989 ACM SIGMOD International Conference on Management of Data*, pages 59–67. Association for Computing Machinery Publication, 1989.
- [48] Ronald J. Brachman, Richard E. Fikes, and Hector J. Levesque. KRYPTON: Integrating terminology and assertion. In *Proceedings of the 3rd Conference of the American Association for Artificial Intelligence (AAAI 1983)*, pages 31–35. AAAI Press, 1983.
- [49] Ronald J. Brachman and Hector J. Levesque. The tractability of subsumption in frame based description languages. In *Proceedings of the 4th Conference of the American Association for Artificial Intelligence (AAAI 1984)*, pages 34–37. AAAI Press, 1984.
- [50] Ronald J. Brachman and Hector J. Levesque, editors. *Readings in Knowledge Representation*. Morgan Kaufmann, 1985.
- [51] Ronald J. Brachman and James G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, 1985.
- [52] Manuela Busaniche and Franco Montagna. Hájek’s logic BL and BL-algebras. In Petr Cintula, Petr Hájek, and Carles Noguera, editors, *Handbook of Mathematical Fuzzy Logic - Volume 1*, volume 37 of *Studies in Logic, Mathematical Logic and Foundations*, pages 355–447. College Publications, London, 2011.
- [53] Marco Cerami. Fuzzy Description Logics under a Mathematical Fuzzy Logic Point of View. To appear. http://www.iiia.csic.es/files/pdfs/monografia_cerami.pdf.
- [54] Marco Cerami. *Fuzzy Description Logics from a Mathematical Fuzzy Logic point of view*. PhD thesis, Universitat de Barcelona, 2012. <http://www.iiia.csic.es/files/pdfs/tesi-cerami.pdf>.

- [55] Marco Cerami, Francesc Esteva, and Félix Bou. Decidability of a description logic over infinite-valued product logic. In Fangzhen Lin, Ulrike Sattler, and Mirosław Truszczyński, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of KR 2010*, pages 203–213. AAAI Press, 2010.
- [56] Marco Cerami, Francesc Esteva, and Àngel García-Cerdaña. On finitely valued fuzzy description logics: The Łukasiewicz case. In Salvatore Greco, Bernadette Bouchon-Meunier, Giulianella Coletti, Mario Fedrizzi, Benedetto Matarazzo, and Ronad R. Yager, editors, *Proceedings of the 14th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2012)*, pages 235–244. Springer, 2012.
- [57] Marco Cerami, Àngel García-Cerdaña, and Francesc Esteva. On the relations between fuzzy description logics and multi-valued modal logics. Forthcoming.
- [58] Marco Cerami, Àngel García-Cerdaña, and Francesc Esteva. From classical description logic to n -graded fuzzy description logics. In *Proceedings of the 19th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2010)*, pages 1506–1513. IEEE Xplore. Digital Library, 2010.
- [59] Marco Cerami, Àngel García-Cerdaña, and Francesc Esteva. On finitely-valued fuzzy description logics. *International Journal of Approximate Reasoning*, 55(9):1890–1916, 2014.
- [60] Marco Cerami and Umberto Straccia. On the (un)decidability of fuzzy description logics under Łukasiewicz t -norms. *Information Sciences*, 227:1–21, 2013.
- [61] Marco Cerami and Umberto Straccia. Complexity sources in fuzzy description logic. In Meghyn Bienvenu, Magdalena Ortiz, Riccardo Rosati, and Mantas Simkus, editors, *Proceedings of the 27th International Workshop on Description Logics (DL 2014)*, volume 1193, pages 421–433. CEUR Workshop Proceedings, 2014.
- [62] Roberto Cignoli and Antoni Torrens. Glivenko like theorems in natural expansions of BCK-logic. *Mathematical Logic Quarterly*, 50(2):111–125, 2003.
- [63] Cristina Civili, Marco Console, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Lorenzo Lepore, Riccardo Mancini, Antonella Poggi, Riccardo Rosati, Marco Ruzzi, Valerio Santarelli, and Domenico Fabio Savo. Mastro studio: Managing ontology-based data access applications. *Proceedings of VLDB Endowment*, 6(12):1314–1317, 2013.
- [64] Bernardo Cuenca-Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter F. Patel-Schneider, and Ulrike Sattler. OWL 2: The next step for OWL. *Journal of Web Semantics*, 6(4):309–322, 2008.
- [65] Giuseppe De Giacomo and Maurizio Lenzerini. Boosting the correspondence between description logics and propositional dynamic logics. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI 1994)*, pages 205–212. AAAI Press/MIT Press, 1994.
- [66] Miguel Delgado, M. Dolores Ruiz, Daniel Sánchez, and M. Amparo Vila. Fuzzy quantification: a state of the art. *Fuzzy Sets and Systems*, 242:1–30, 2014.
- [67] Premkumar Devanbu, Ronald J. Brachman, Peter J. Selfridge, and Bruce W. Ballard. LASSIE: A knowledge-based software information system. *Communications of the Association for Computing Machinery*, 34(5):36–49, 1991.
- [68] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Werner Nutt. The complexity of concept languages. In James Allen, Ronald J. Brachman, Erik Sandewall, Hector J. Levesque, Ray Reiter, and Richard Fikes, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR 1991)*, pages 151–162. Morgan Kaufmann, 1991.
- [69] Francesc Esteva, Lluís Godo, Petr Hájek, and Mirko Navara. Residuated fuzzy logics with an involutive negation. *Archive for Mathematical Logic*, 39(2):103–124, 2000.
- [70] Francesc Esteva, Lluís Godo, and Enrico Marchioni. Fuzzy logics with enriched language. In Petr Cintula, Petr Hájek, and Carles Noguera, editors, *Handbook of Mathematical Fuzzy Logic - Volume 2*, volume 38 of *Studies in Logic, Mathematical Logic and Foundations*, pages 627–711. College Publications, London, 2011.
- [71] Francesc Esteva, Lluís Godo, and Carles Noguera. Expanding the propositional logic of a t -norm with truth-constants: Completeness results for rational semantics. *Soft Computing*, 14(3):273–284, 2010.
- [72] Enrico Franconi. CRACK. In Enrico Franconi, Giuseppe De Giacomo, Robert M. MacGregor, Werner Nutt, and Christopher A. Welty, editors, *Proceedings of the International Workshop on Description Logics (DL 1998)*, pages 58–59. CEUR Workshop Proceedings, 1998.
- [73] Àngel García-Cerdaña, Eva Armengol, and Francesc Esteva. Fuzzy description logics and t -norm based fuzzy logics. *International Journal of Approximate Reasoning*, 51(6):632–655, 2010.
- [74] Franco Giunchiglia and Roberto Sebastiani. A SAT-based decision procedure for ALC. In Luigia Calucci Aiello, Jon Doyle, and Stuart C. Shapiro, editors, *Proceedings of the 6th International Conference on the Principles of Knowledge Representation and Reasoning (KR 1996)*, pages 304–314. Morgan Kaufmann, 1996.

- [75] Volker Haarslev, Kay Hidde, Ralf Möller, and Michael Wessel. The racerpro knowledge representation and reasoning system. *Semantic web*, 3(3):267–277, 2012.
- [76] Volker Haarslev and Ralf Möller. RACER system description. In Rajeev Gore, Alexander Leitsch, and Tobias Nipkow, editors, *Proceedings of the 15th International Joint Conference on Automated Reasoning (IJCAR 2001)*, volume 2083 of *Lecture Notes in Artificial Intelligence*, pages 701–705. Springer, 2001.
- [77] Reiner Hähnle. Advanced many-valued logics. In *Handbook of Philosophical Logic*, pages 297–395. Kluwer, 2 edition, 2001.
- [78] Petr Hájek. *Metamathematics of Fuzzy Logic*, volume 4 of *Trends in Logic*. Kluwer, Dordrecht, 1998.
- [79] Petr Hájek. On very true. *Fuzzy Sets and Systems*, 124(3):329–333, 2001.
- [80] Petr Hájek. Making fuzzy description logic more general. *Fuzzy Sets and Systems*, 154(1):1–15, 2005.
- [81] Petr Hájek. Computational complexity of t-norm based propositional fuzzy logics with rational truth constants. *Fuzzy Sets and Systems*, 157(5):677–682, 2006.
- [82] Petr Hájek. What does mathematical fuzzy logic offer to description logic? In Elie Sanchez, editor, *Fuzzy Logic and the Semantic Web*, volume 1 of *Capturing Intelligence*, pages 91–100. Elsevier, Amsterdam, 2006.
- [83] Petr Hájek, Franco Montagna, and Carles Noguera. Arithmetical complexity of first-order fuzzy logics. In Petr Cintula, Petr Hájek, and Carles Noguera, editors, *Handbook of Mathematical Fuzzy Logic - Volume 2*, volume 38 of *Studies in Logic, Mathematical Logic and Foundations*, pages 853–908. College Publications, London, 2011.
- [84] Patrick J. Hayes. The logic of frames. In Dieter Metzging, editor, *Frame Concepts and Text Understanding*, pages 46–61. W. de Gruyter, Berlin, 1979. Reprinted in [50] pages 287–295.
- [85] Bernhard Hollunder, Werner Nutt, and Manfred Schmidt-Schauss. Subsumption algorithms for concept description languages. In Luigia Calucci Aiello, editor, *Proceedings of the 9th European Conference on Artificial Intelligence (ECAI 1990)*, pages 348–353. Pitman, 1990.
- [86] John E. Hopcroft and Richard M. Karp. An $n^5/2$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2:225–231, 1973.
- [87] Ian Horrocks. Using an expressive description logic: FaCT or fiction? In *Proceedings of the 8th International Conference on the Principles of Knowledge Representation and Reasoning (KR 1998)*, pages 636–647. Morgan Kaufmann, 1998.
- [88] Ian Horrocks, Oliver Kutz, and Ulrike Sattler. The even more irresistible *SRCTLQ*. In Patrick Doherty, John Mylopoulos, and Christopher Welty, editors, *Proceedings of the 10th International Conference of Knowledge Representation and Reasoning (KR 2006)*, pages 452–457. AAAI Press, 2006.
- [89] Thomas S. Kaczmarek, Raymond Bates, and Gabriel Robins. Recent developments in NIKL. In *Proceedings of the 5th Conference of the American Association for Artificial Intelligence (AAAI 1986)*, pages 978–985. AAAI Press, 1986.
- [90] Yevgeny Kazakov, Markus Krötzsch, and František Simančík. The incredible ELK - from polynomial procedures to efficient reasoning with \mathcal{EL} ontologies. *Journal of Automated Reasoning*, 53(1):1–61, 2014.
- [91] Erich Peter Klement, Radko Mesiar, and Endre Pap. *Triangular Norms*, volume 8 of *Trends in Logic*. Kluwer, Dordrecht, 2000.
- [92] Thomas Lukasiewicz and Umberto Straccia. Description logic programs under probabilistic uncertainty and fuzzy vagueness. *International Journal of Approximate Reasoning*, 50(6):837–853, 2009.
- [93] Robert MacGregor and R. Bates. The Loom knowledge representation language. Technical Report ISI/RS-87-188, University of Southern California, Information Science Institute, Marina del Rey, 1987.
- [94] Gaspar Mayor and Joan Torrens. On a class of operators for expert systems. *International Journal of Intelligent Systems*, 8:771–778, 1993.
- [95] Gaspar Mayor and Joan Torrens. Triangular norms in discrete settings. In Erich Peter Klement and Radko Messiar, editors, *Logical, Algebraic, Analytic, and Probabilistic Aspects of Triangular Norms*, pages 189–230. Elsevier, Amsterdam, 2005.
- [96] Eric Mays, Robert Dionne, and Robert Weida. K-REP system overview. *SIGART Bulletin*, 2(3):93–97, 1991.
- [97] George Metcalfe, Nicola Olivetti, and Dov M. Gabbay. *Proof Theory for Fuzzy Logics*, volume 36 of *Applied Logic Series*. Springer, 2008.
- [98] Marvin Minsky. A framework for representing knowledge. In John Haugeland, editor, *Mind Design*. MIT Press, 1981.
- [99] Paul S. Mostert and Allen L. Shields. On the structure of semigroups on a compact manifold with boundary. *The Annals of Mathematics, Second Series*, 65:117–143, 1957.

- [100] Bernhard Nebel and Christof Peltason. Terminological reasoning and information management. In D. Karagianis, editor, *Artificial Intelligence and Information Systems: Integration Aspects*, volume 474 of *Lecture Notes in Computer Science*, pages 181–212. Springer, 1991.
- [101] Bernhard Nebel and Kai von Luck. Hybrid reasoning in BACK. In Zbigniew W. Raś and Lorenza Saitta, editors, *Proceedings of the 3rd International Symposium on Methodologies for Intelligent Systems (ISMIS 1988)*, pages 260–269. North-Holland, Amsterdam, 1988.
- [102] OWL 2 Web Ontology Language Document Overview. <http://www.w3.org/TR/2009/REC-owl2-overview-20091027/>. W3C, 2009.
- [103] OWL 2 Web Ontology Language Profiles. <http://www.w3.org/TR/2009/REC-owl2-profiles-20091027/>. W3C, 2009.
- [104] Peter F. Patel-Schneider. Small and be beautiful in knowledge representation. Technical Report 660, Fairchild, Palo Alto, CA, 1984.
- [105] Peter F. Patel-Schneider. Undecidability of subsumption in NIKL. *Artificial Intelligence*, 39:263–272, 1989.
- [106] Emil L. Post. A variant of a recursively unsolvable problem. *Bulletin of the American Mathematical Society*, 52:264–268, 1946.
- [107] M. Ros Quillian. Word concepts: A theory and simulation of some basic capabilities. *Behavioral Science*, 12:410–430, 1967.
- [108] Steven W. Reyner. An analysis of a good algorithm for the subtree problem. *SIAM Journal on Computing*, 6(4):730–732, 1977.
- [109] Daniel Sánchez and Andrea G.B. Tettamanzi. Generalizing quantification in fuzzy description logics. In B. Reusch, editor, *Computational Intelligence, Theory and Applications. Proceedings 8th Fuzzy Days in Dortmund*, Advances in Soft Computing Series, pages 397–411. Springer, 2004.
- [110] Daniel Sánchez and Andrea G.B. Tettamanzi. Fuzzy quantification in fuzzy description logics. In Elie Sanchez, editor, *Fuzzy Logic and the Semantic Web*, Capturing Intelligence, chapter 8, pages 135–159. Elsevier, 2006.
- [111] Ulrike Sattler, Diego Calvanese, and Ralf Molitor. Relationship with other formalisms. In Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation and Applications*, pages 137–177. Cambridge University Press, 2003.
- [112] Klaus Schild. A correspondence theory for terminological logics: Preliminary report. In John Mylopoulos and Raymond Reiter, editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI 1991)*, pages 466–471. Morgan Kaufmann, 1991.
- [113] Manfred Schmidt-Schauß. Subsumption in KL-ONE is undecidable. In Ronald J. Brachman, Hector J. Levesque, and Raymond Reiter, editors, *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning (KR 1989)*, pages 421–431. Morgan Kaufmann, 1989.
- [114] Manfred Schmidt-Schauß and Gert Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.
- [115] Fabrizio Sebastiani and Umberto Straccia. A computationally tractable terminological logic. In Thore Danielsen, editor, *Proceedings of the 3rd Scandinavian Conference on Artificial Intelligence (SCAI 1991)*, pages 307–315. IOS, 1991.
- [116] Rob Shearer, Boris Motik, and Ian Horrocks. Hermit: A highly-efficient owl reasoner. In Catherine Dolbear, Alan Ruttenberg, and Ulrike Sattler, editors, *Proceedings of the 5th International Workshop on OWL: Experiences and Directions (OWLED 2008)*, page 91. CEUR Workshop Proceedings, 2008.
- [117] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical OWL-DL reasoner. *Web Semantics*, 5(2):51–53, 2007.
- [118] Giorgos Stoilos, Nikos Simou, Giorgos Stamou, and Stefanos Kollias. Uncertainty and the semantic web. *IEEE Intelligent Systems*, 21(5):84–87, 2006.
- [119] Giorgos Stoilos and Giorgos Stamou. A framework for reasoning with expressive continuous fuzzy description logics. In Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, and Ulrike Sattler, editors, *Proceedings of the 22nd International Workshop on Description Logics (DL 2009)*. CEUR Workshop Proceedings, 2009.
- [120] Giorgos Stoilos and Giorgos Stamou. Reasoning with fuzzy extensions of OWL and OWL 2. *Knowledge and Information Systems*, 40(1):205–242, 2014.
- [121] Giorgos Stoilos, Giorgos Stamou, Vassilis Tzouvaras, Jeff Z. Pan, and Ian Horrocks. Fuzzy OWL: Uncertainty and the semantic web. In Bernardo Cuenca Grau, Ian Horrocks, Bijan Parsia, and Peter F. Patel-Schneider, editors, *Proceedings of the 2nd International Workshop of OWL (OWLED 2005)*. CEUR Workshop Proceedings, 2005.

- [122] Giorgos Stoilos, Giorgos B. Stamou, Jeff Z. Pan, Vassilis Tzouvaras, and Ian Horrocks. The fuzzy description logic $f\text{-}\mathcal{SHLN}$. In Paulo Cesar G. da Costa, Kathryn B. Laskey, Kenneth J. Laskey, and Michael Pool, editors, *Proceedings of the 1st International Workshop on Uncertainty Reasoning For the Semantic Web (URSW 2005)*. CE, 2005.
- [123] Giorgos Stoilos, Giorgos B. Stamou, Jeff Z. Pan, Vassilis Tzouvaras, and Ian Horrocks. Reasoning with very expressive fuzzy description logics. *Journal of Artificial Intelligence Research*, 30:273–320, 2007.
- [124] Umberto Straccia. A fuzzy description logic. In Jack Mostow and Chuck Rich, editors, *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI 1998)*, pages 594–599. AAAI Press, 1998.
- [125] Umberto Straccia. Reasoning within fuzzy description logics. *Journal of Artificial Intelligence Research*, 14:137–166, 2001.
- [126] Umberto Straccia. Transforming fuzzy description logic into classical description logics. In Jose J. Alferes and Joao Leite, editors, *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA 2004)*, volume 3229 of *Lecture Notes in Computer Science*, pages 385–399. Springer, 2004.
- [127] Umberto Straccia. Uncertainty in description logics: A lattice-based approach. In *Proceedings of the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2004)*, pages 251–258. Editrice Università La Sapienza, 2004.
- [128] Umberto Straccia. Description logics with fuzzy concrete domains. In Fahiem Bacchus and Tommi Jaakkola, editors, *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI 2005)*, pages 559–567. AUAI Press, 2005.
- [129] Umberto Straccia. Fuzzy ALC with fuzzy concrete domains. In Ian Horrocks, Ulrike Sattler, and Frank Wolter, editors, *Proceedings of the International Workshop on Description Logics (DL 2005)*, pages 96–103. CEUR Workshop Proceedings, 2005.
- [130] Umberto Straccia. Towards a fuzzy description logic for the semantic web (preliminary report). In Asunción Gómez-Pérez and Jérôme Euzenat, editors, *Proceedings of the 2nd European Semantic Web Conference (ESWC 2005)*, volume 3532 of *Lecture Notes in Computer Science*, pages 167–181. Springer, 2005.
- [131] Umberto Straccia. A fuzzy description logic for the semantic web. In Elie Sanchez, editor, *Fuzzy Logic and the Semantic Web, Capturing Intelligence*, chapter 4, pages 73–90. Elsevier, 2006.
- [132] Umberto Straccia. *Foundations of Fuzzy Logic and Semantic Web Languages*. CRC Studies in Informatics Series. Chapman and Hall/CRC, Boca Raton, FL, 2013.
- [133] David Toman and Grant E. Weddell. On reasoning about structural equality in XML: A description logic approach. *Theoretical Computer Science*, 336(1):181–203, 2005.
- [134] Vicenç Torra and Yasuo Narukawa. *Modeling decisions - information fusion and aggregation operators*. Springer, 2007.
- [135] David S. Touretzky. *The Mathematics of Inheritance Systems*. Pitman, London, GB, 1986.
- [136] Christopher B. Tresp and Ralf Molitor. A description logic for vague knowledge. In Henri Prade, editor, *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI 1998)*, pages 361–365. Wiley, 1998.
- [137] Dmitry Tsarkov and Ian Horrocks. Fact++ description logic reasoner: System description. In Ulrich Furbach and Natarajan Shankar, editors, *Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR 2006)*, pages 292–297. Berlin, Heidelberg, 2006. Springer.
- [138] Marc Vilain. KL-TWO, a hybrid knowledge representation system. Technical Report 5694, BBN Laboratories, March 1984.
- [139] William A. Woods. What's in a link: Foundations for semantic networks. In Daniel G. Bobrow and Allan Collins, editors, *Representation and Understanding: Studies in Cognitive Science*, pages 35–82. Academic Press, 1975. Reprinted in [50] pages 35–82.
- [140] William A. Woods and James G. Schmolze. The KL-ONE family. *Computers & Mathematics with Applications*, 23(2–5):133–177, 1992.
- [141] John Yen. Generalizing term subsumption languages to fuzzy logics. In John Mylopoulos and Raymond Reiter, editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI 1991)*, pages 472–477. Morgan Kaufmann, 1991.
- [142] Lotfi A. Zadeh. Test-score semantics for natural languages. In Jan Horecký, editor, *Proceedings of the 9th Conference on Computational Linguistics (COLING 1982)*, pages 425–430. Academia, 1982.

FERNANDO BOBILLO
Department of Computer Science and Systems Engineering
University of Zaragoza
C. Maria de Luna 1
50018 Zaragoza, Spain
Email: fbobillo@unizar.es

MARCO CERAMI
Department of Computer Science
Palacký University Olomouc
17. listopadu 2
771 46 Olomouc, Czech Republic
Email: marco.cerami@upol.cz

FRANCESC ESTEVA
Artificial Intelligence Research Institute (IIIA)
Spanish National Research Council (CSIC)
Campus de la Universitat Autònoma de Barcelona s/n
08193 Bellaterra, Catalonia, Spain
Email: esteva@iiia.csic.es

ÀNGEL GARCÍA-CERDAÑA
Artificial Intelligence Research Institute (IIIA)
Spanish National Research Council (CSIC)
Campus de la Universitat Autònoma de Barcelona s/n
08193 Bellaterra, Catalonia, Spain

and

Information and Communication Technologies Department
Universitat Pompeu Fabra
Tànger 122-140
08018 Barcelona, Catalonia, Spain
Email: angel@iiia.csic.es

RAFAEL PEÑALOZA
KRDB Research Centre
Free University of Bozen-Bolzano
Piazza Domenicani 3
I-39100 Bozen-Bolzano BZ, Italy
Email: rafael.penaloza@unibz.it

UMBERTO STRACCIA
Istituto di Scienze e Tecnologie dell'Informazione (ISTI)
Consiglio Nazionale delle Ricerche (CNR)
Via G. Moruzzi 1
56124 Pisa, Italy
Email: straccia@isti.cnr.it

INDEX

- A, *see* Abelian logic
- Abelian logic, 994–996
 - hypersequent system, 999
- ABox, 1121
- adjunction condition, 1065
- aggregated concept, 1131
- aggregation operator, 1112, 1131
- \mathcal{AL} , 1117
- \mathcal{ALC} , 1117, 1139, 1141, 1144,
 - 1167, 1170, 1172, 1173
- \mathcal{ALCE} , 1173
- \mathcal{ALCH} , 1146
- algebra
 - BL, 1030
 - canonical, 1111
 - Gödel, 1031
 - MV, 1065
 - partial algebra associated with
 - a d.p.r. tomonoid, 933
 - PMV^+ -algebra, 1226
 - product, 1031
 - SBL, 1031
 - Wajsberg, 1031
- alpha-cut, 1146
- ambiguity, 1240, 1276
- arity of a symbol, 1124
- artificial precision, 1267
- assertability, 1248, 1261, 1264, 1276
- assertional box, 1121
- assignment, 1125
- automata-based algorithm, 1144, 1172
- averaging operation, 1276
- axiom, 1133
 - Henkin, 1091
 - logical, 1078
- backtracking game, *see* semantic game
- Best Entailment Degree, 1122, 1136,
 - 1141
- Best Satisfiability Degree, 1122, 1136
- \mathcal{BG} -game, *see* semantic game,
 - backtracking game
- \mathcal{BL} -game, *see* semantic game,
 - backtracking game
- blurry boundaries, 1237, 1246,
 - 1249–1251, 1257, 1258
- bookmaking
 - on Boolean events, 1209
 - on many-valued conditional events, 1227
 - on many-valued events, 1209
- borderline cases, 1238, 1244, 1246,
 - 1249–1251, 1257, 1258
- cancellative hoop logic, 993
- canonical model, 1156
- Cayley tomonoid, 949
- chain, 931
 - finite, 1164
 - R-chain, 934
 - naturally ordered, 936
 - ordinal sum of R-chains, 936
 - residuated, 931
 - standard, 1110
 - finite, 1111
- channel
 - asymmetric, 1043
 - multichannel, 1045, 1048
- CHL, *see* cancellative hoop logic
- classical semantic picture, 1247, 1268
- closeness, 1257, 1262, 1268
- coherence criterion
 - complexity, 1216
 - for Boolean events, 1209
 - for many-valued conditional events, 1228
 - for many-valued events, 1209
- comparison class, 1239

- complexity
 - computational, 1152, 1161, 1173
- composition tomonoid, 948
- concept
 - aggregated, 1131
 - atomic, 1114
 - complementation, 1167
 - conjunction, 1165, 1168, 1169
 - weak, 1169
 - constructor, 1128, 1169
 - disjunction, 1166, 1169
 - weak, 1169
 - empty, 1114
 - existential quantification, 1114
 - implication, 1114, 1169
 - modifier, 1131, 1171
 - satisfiability, 1136, 1171, 1173
 - strong complementary, 1114
 - strong intersection, 1114
 - strong union, 1114
 - subsumption, 1136, 1170–1173
 - universal, 1114
 - value restriction, 1114
 - weak complementary, 1114
 - weak intersection, 1114
 - weak union, 1114
- concrete domain, 1167, 1171
- conjunction
 - lattice, 1255
 - n -ary, 1154
 - strong, 1255, 1266, 1275
 - weak, 1255, 1275
- consequence
 - operation, 1068
 - semantic, 1073
 - syntactic, 1070
- consistency, 1171
 - of a knowledge base, 1122
- constant
 - object, 1066
 - truth, 1066
- constraint, 1170
- constructors, 1114
- context sensitivity, 1239
- contextualism, 1240, 1249
- datatype, 1132
- decidability, 1152, 1170
- degree
 - function, 1269
 - functionality, 1255, 1272
 - of membership, 1251, 1275
 - of property possession, 1251
 - of truth, 1251, 1260, 1267, 1275
 - abstract, 1252, 1255
 - concrete, 1252
 - ordinal approach, 1270
 - structure, 1251, 1261
 - of typicality, 1275
 - provability, 1074, 1100
- Delta operator, *see* operator,
 - Monteiro–Baaz
- determinately operator, 1276
- diagram, 1095
- dialogue game
 - Lorenzen’s game, 978
- dialogue principles, *see* semantic game
- discriminating payoff function, 986,
 - 989, 991–993, 995
- disjunctive game states, 995, 998
- divisibility condition, 1107
- \mathcal{E} -game, *see* semantic game, explicit
 - evaluation game
- \mathcal{ELC} , 1117, 1159
- encoding
 - Gilbert, 1045
- entailment, 1122, 1136
 - *-entailment, 1122
- epistemicism, 1241, 1258, 1268
 - fuzzy, 1268
- evaluated formal proof, 1070
- evaluation game, *see* semantic game
- extension
 - admissible, 1245, 1247
 - elementary, 1095
 - of a fuzzy theory, 1083
 - conservative, 1083
 - of a predicate, 1237
 - of a valuation, 1244, 1255
 - strong, 1094

- finite model property, 1172
- first order
 - atomic formula, 1124
 - closed formula, 1124
 - closed term, 1124
 - consequence, 1125
 - formula, 1124
 - signature, 1124
 - signature associated to a
 - description language, 1126
 - structure, 1124
 - term, 1124
 - theory, 1124
 - value of a formula, 1125
- \mathcal{FL}^- , 1117, 1151
- \mathcal{FL}_0 , 1117
- formula
 - evaluated, 1069
 - provable
 - effectively, 1074
- frame-based system, 1164
- Frege, 1276
- function
 - membership, 1064
 - Post, 1037
- fuzzy
 - quantifier, 1012
 - set, 1064, 1252
 - theory, 1074
- \mathcal{G} -game, *see* semantic game,
 - Giles's game
- \mathcal{GA} -game, *see* semantic game,
 - Giles's game for A
- game
 - BL, 1055
 - evaluation, *see* semantic game
 - Gödel, 1048
 - Hintikka, 970
 - Pelc, 1051
 - product, 1051
 - record, 1046, 1052, 1054, 1056
 - equivalent, 1047
 - semantic, 969
 - state, 1035, 1037–1039
 - character, 1040
 - conservation law, 1040, 1044
 - weight, 1040, 1044
 - Ulam-Rényi, 1029, 1033
 - asymmetric, 1043
- generality, 1240
- Giles's game, *see* semantic game
- Gödel
 - conditional, 1266
 - logic, 969, 970, 975, 1003, 1255, 1275
- graded
 - assertion axioms, 1121
 - axiom, 1167
 - concept assertion axiom, 1121
 - concept inclusion axiom, 1121
 - inclusion axioms, 1121
 - positive satisfiability of a concept, 1122
 - role assertion axiom, 1121
 - satisfiability of a concept, 1122
 - subsumption, 1122
- graded consequence operation
 - compact, 1070
- \mathcal{H} -game, *see* semantic game,
 - Hintikka's game
- \mathcal{H} -mv-game, *see* semantic game,
 - Hintikka's game for MVL
- hedge, 1131, 1132, 1166
- Hintikka's game, *see* semantic game
- Hintikka's principle, *see* semantic game
- hoop
 - Wajsberg, 1032
- hypersequent, 995
- \mathfrak{JALC} , 1171
- \mathfrak{JALCE} , 1144, 1153, 1160, 1172
- $\mathfrak{JAL\mathcal{E}}$, 1172
- \mathfrak{JALUE} , 1148, 1161
- \mathcal{IG} -game, *see* semantic game,
 - backtracking game
- \mathcal{III} -game, *see* semantic game,
 - backtracking game
- implicit backtracking game, *see* semantic game, backtracking game

- individual equality, 1135
- inheritance network, 1164
- instance
 - of a concept, 1126
 - of a description, 1126
- interpretation
 - domain, 1115
 - function, 1115
 - *-interpretation, 1115
- intuitions, 1274
- jolt problem, 1268
- Kleene–Dienes
 - implication, 1166–1168, 1173
- Kleene–Zadeh
 - logic, 971, 975, 1008
- knowledge base, 1121
- KZ, *see* Kleene–Zadeh logic
- Ł, *see* Łukasiewicz logic
- lattice
 - residuated, 930, 1064
 - commutative, 931
 - enriched, 1097
 - integral, 931
- local reflexivity concept, 1130
- location problem, 1241, 1244, 1248, 1268
- logic
 - KZ, *see* Kleene–Zadeh logic
 - A, *see* Abelian logic
 - Abelian, 994–996
 - BL, 1045, 1055
 - cancellative hoop, 993
 - Gödel, 969, 970, 975, 1003, 1048, 1255
 - Kleene–Zadeh, 971, 975, 1008
 - Łukasiewicz, 969, 970, 975, 1001, 1009, 1012, 1038, 1255, 1266, 1275
 - product, 970, 1006, 1051, 1255, 1275
 - rational Pavelka, 1099
 - SBL, 1055
 - weak Łukasiewicz, *see* Kleene–Zadeh, logic
 - Zadeh, 1253, 1277
- Łukasiewicz
 - conditional, 1265
 - logic, 969, 975, 1001, 1009, 1012, 1038, 1255, 1266, 1275
 - weak, *see* Kleene–Zadeh, logic
 - strong conjunction, 975, 976, 980, 1011
 - strong disjunction, 975, 976, 1011
 - t-norm, 1108
- manipulator, 1166
- maximality principle, 1073
- meaning, 1247
 - and use, 1242, *see* location problem
 - determination of, 1242, *see* location problem
- membership function, 1064
- Mixed Integer Linear Programming, 1141, 1171
- model, 1074
 - acceptable, 1247, 1271
 - canonical, 1156
 - classical, 1250
 - fuzzy, 1250
 - intended, 1247, 1249, 1256, 1267
 - *-model of an interpretation, 1122
 - of a knowledge base, 1122
 - of an interpretation, 1122
 - s-valuational, 1244
 - theoretic semantics, 1247
 - theory, 1247
 - T_* -model of a first order theory, 1125
 - witnessed, 1125
- modus ponens, 1079, 1266
- multiplication, 1065
- MV-algebra, 1065
 - with internal state, 1218
 - relation to states of MV-algebras, 1221
 - subdirectly irreducible, 1220

- natural language quantifiers, 1013
- negation function, 1108
 - Gödel, 1108
 - involution, 1108
 - standard, 1108
 - strong, 1108
 - weak, 1108
- \mathfrak{NEL} , 1117, 1160
- Nmatrix, *see* non-deterministic matrix
- nominal, 1129, 1167, 1172
- non-deterministic matrix, 973, 1025
- number restriction, 1129, 1130, 1166
 - qualified, 1167
 - unqualified, 1167
- object
 - constant, 1124
 - variable, 1124
- operation
 - consequence, 1068
 - graded, 1070
 - syntactic, 1068
 - logically fitting, 1097
- operator
 - aggregation, 1112, 1131
 - determinately, 1276
 - logical, 1253
 - Monteiro–Baaz, 1111
- OWL 2, 1135, 1165
- paradox
 - sorites, 1098
- payoff principles, *see* semantic game
- PFL, *see* Philosophers' Fuzzy Logic
- Philosophers' Fuzzy Logic, 1254, 1255, 1266, 1272, 1276
- plurivaluationism, 1247, 1259
 - fuzzy, 1271
- poset
 - almost complete, 943
- Post correspondence problem, 1153, 1172
- pragmatics, 1276
- precisification, 1245, 1249, 1276
- precision, 1237
- precomplement, 1254
- predicate, 1237
 - borderline case, 1238
 - extension, 1237, 1244
 - language, 1124
 - negative case, 1238
 - positive case, 1238
- principle
 - maximality, 1073
 - of limited liability,
 - see* semantic game
- product
 - logic, 970, 1006, 1255, 1275
 - t-norm, 1108
- prototypicality, *see* degree of typicality
- provability degree, 1074, 1100
- quantifier, 1169
 - binary, 1013
 - deliberate choice, 1020
 - existential, 1166
 - fuzzy, 1012
 - natural language, 1013
 - proportionality, 1012, 1013, 1022
 - semi-fuzzy, 1011
 - blind choice, 1014–1020
 - deliberate choice, 1020–1022
 - universal, 1120
- question, 1034, 1045
 - even split, 1044
- R-chain, 934
 - naturally ordered, 936
 - ordinal sum of R-chains, 936
- \mathcal{R} -game, *see* semantic game, random choice game
- rational Pavelka logic, 1099
- reasoning task, 1135
- reduction, 1145, 1148
- residuated
 - implication, 1168, 1169
 - negation, 1169
- residuated lattice, *see* lattice, residuated
- residuum, 1065
 - of a t-norm, 1107

Riesz decomposition property, 937

role

- assertion, 1134
- asymmetric, 1134
- atomic, 1114
- constructor, 1131, 1172
- disjoint, 1134
- functional, 1133
- hierarchy, 1133, 1167
- inverse, 1131, 1167, 1173
- irreflexive, 1134
- modified, 1132
- modifier, 1132
- reflexive, 1134
- symmetric, 1134
- transitive, 1133, 1134, 1167
- universal, 1132

rtv, 1271

rule

- generalization, 1079
- inference, 1069
 - sound, 1072
- truth constant introduction, 1079

satisfiability

- of a first order formula, 1125
- of an axiom, 1121
- *-satisfiability of an axiom, 1121

semantic

- game, 969
 - backtracking game, 1001, 1003
 - dialogue principles, 986
 - explicit evaluation game for \mathbb{L} , 975, 977
 - Giles's game, 978, 980, 981, 983, 992
 - Hintikka's game, 970, 971, 975
 - Hintikka's game for KZ, 971
 - Hintikka's game for MVL, 972
 - Hintikka's principle, 973, 1010
 - payoff principles, 984, 986
 - principle of limited liability, 980, 987, 994, 1014–1017, 1019–1022
 - random choice game, 1007

indeterminacy, 1248, 1257

plurality, 1248

web, 1165

semantics

test score, 1165

sentence, 1124

set

blurry, 1269

crisp, 1251

fuzzy, 1064, 1252

rubbery, 1271

sftv, 1267

sharpening, 1244, 1245, 1256

SHL, 1144, 1171

SHIF(D), 1171

SHOLN(D), 1135

signature for description languages,
1114

SMV-algebra, *see* MV-algebra with
internal state

sorites

argument, 1238, 1266

paradox, 1098, 1238

solution, 1240, 1241, 1246,
1248, 1249, 1261

series, 1238

susceptibility, 1238, 1257, 1263

SROIQ(D), 1135, 1171

state, 1191

conditional, 1227

examples, 1195

extreme, 1205

faithful, 1194, 1207

internal, 1218

invariant, 1206, 1207

of ℓ -groups, 1201

of MV-algebras, 1191

of finitely presented algebras,
1196

properties, 1192

space, 1205

strategy, 1034, 1036, 1039, 1049, 1053

two batch, 1041

structural subsumption algorithm,
1151, 1164, 1170

- structure, 1072
 - first order, 1124
 - expanded, 1094
 - isomorphic, 1095
 - T_* -structure, 1124
- substructure, 1094
 - elementary, 1095
 - strong, 1094
 - weak, 1094
- subvaluationism, 1245, 1255
- supertruth, 1245
- supervaluationism, 1245, 1255, 1260
 - degree-theoretic, 1246, 1255
- symbols
 - for functionals, 1124
 - for predicates, 1124
- syntax
 - evaluated
 - complete, 1075
 - sound, 1075
- T_* , 1111
- t-conorm, 1110
 - Łukasiewicz, 1110
 - divisible finite, 1111
 - dual, 1110
 - finite, 1111
 - maximum, 1110
 - probabilistic sum, 1110
- t-norm, 1107, 1254
 - Łukasiewicz, 1108
 - continuous, 1107
 - divisible finite, 1110
 - finite, 1110
 - fuzzy logics, 1254
 - idempotent, 1108, 1159
 - left-continuous, 944
 - minimum, 1108
 - nilpotent, 1108
 - product, 1108
 - residuum, 1254
 - strict, 1108, 1160
- tableau algorithm, 1137, 1165, 1172
- tableau and optimization problem
 - algorithm, 1142
- tautology, 1073
 - T_* -tautology, 1125
- TBox, 1121
- terminological box, *see* TBox
- Theorem
 - de Finetti's
 - for SMV-algebras, 1224
 - for classical events, 1209
 - for many-valued conditional events, 1229
 - for many-valued events, 1211
 - existence of invariant faithful states, 1207
 - extension of states, 1202
 - from MV-algebras to ℓ -groups, 1201
 - integral representation of states, 1199, 1203
- theory
 - fuzzy, 1074
 - complete, 1087, 1088
 - consistent, 1081
 - contradictory, 1081
 - Henkin, 1091
- tolerance, 1257, 1258, 1262
- tomonoid, 931
 - Cayley, 949
 - commutative, 931
 - composition, 948
 - d.p.r., 933
 - negative, 931
 - positive, 931
 - q.n.c., 943
 - Archimedean, 947
 - coextension, 946
 - congruence, 945
 - filter, 944
 - quotient, 946
 - quantic, 943
 - residuated, 931
 - divisible, 933
- triangular co-norm, *see* t-conorm
- triangular norm, *see* t-norm

truth

- constant, 1066, 1124
- degree, *see* degree of truth
- depresser, 1112
- functionality,
 - see* degree functionality
- probability of, 1246
- similarity in respect of, 1258, 1260, 1261, 1271
- simpliciter, 1247, 1249
- stresser, 1112
- supertruth, 1245
- table
 - Łukasiewicz, 1245
 - Bočvar, 1245
 - Kleene strong, 1245
 - Kleene weak, 1245
 - tripartite, 1244
- value
 - designated, 1245
 - gap, 1244, 1260
 - of a first order formula, 1125
 - third, 1244, 1260

uncertainty, 1239

undecidability, 1153, 1159, 1172

universe, 1124

vagueness, 1237

- characteristics, 1237, 1256
- definition, 1257
- higher-order, 1258
- metaphysical, 1237
- nature of, 1267
- theory of, 1240

validity, 1265

valuation

- classical, 1244
- fuzzy, 1123, 1255
- three-valued, 1244
- truth, 1072

value

- of a game, 972
- of a term, 1125
- restriction, 1165

variable

- bound, 1124
- free, 1124

vocabulary

- logical, 1250
- nonlogical, 1250

Zadeh

- conditional, 1266
- logic, 1253, 1277
- semantics, 1139, 1167, 1170