UNIVERSITÉ D'ARTOIS

École Doctorale Sciences Technologies et Santé n° 585

Managing Inconsistency in Partially Ordered Information: Application to Dynamic and Explainable Access Control Models

THÈSE

présentée et soutenue publiquement le 21 novembre 2025

en vue de l'obtention du

Doctorat de l'Université d'Artois

(Spécialité Informatique)

par

Ahmed Laouar

Composition du jury

Présidente :	Meghyn Bienvenu	CNRS – LaBRI
Directeurs :	Sihem Belabbes	Université de Paris 8 – LIASD
	Salem Benferhat	Université d'Artois – CRIL
	Karim Tabia	Université d'Artois – CRIL
Rapporteurs :	Clara Bertolissi	INSA Centre-Val de Loire – LIFO
	Anthony Hunter	University College London – UCL
Examinateurs :	Sébastien Destercke	CNRS – Heudiasyc
	Laura Giordano	Università del Piemonte Orientale – DISIT

Centre de Recherche en Informatique de Lens – CNRS UMR 8188
Université d'Artois, rue Jean Souvraz, S.P. 18 F-62307, Lens Cedex France
Secrétariat : +33 (0)3 21 79 17 23
http://www.cril.univ-artois.fr

Mise en page avec memcril (B. Mazure, CRIL) et thloria (D. Roegel, LORIA).

Acknowledgements

First and foremost, I would like to sincerely thank the reviewers, Mme Clara Bertolissi and Mr Anthony Hunter, for agreeing to review my thesis and provide valuable feedback. I am also grateful to the members of the jury, Mme Meghyn Bienvenu, Mr Sébastien Destercke, and Mme Laura Giordano, for accepting to be part of the committee and for their time and expertise.

I would like to express my deepest gratitude to my supervisors: Mme Sihem Belabbes, Mr Salem Benferhat, and Mr Karim Tabia, whose insightful guidance, encouragement, support, and supervision over more than three years have been invaluable. In particular, to Salem, whose dedication and example have been an inspiration not only in academia but also in everyday life.

I am also grateful to my colleagues in CRIL, in particular fellow PhD students, for their collaboration and support throughout my research. I also wish to thank the colleagues from the labs where I conducted secondments, in Cape Town, Fès, and Can Tho, for warmly welcoming me and creating an enriching environment.

My heartfelt thanks go to my family: my parents for being a constant source of inspiration and support for academic pursuits; my sister for leading the way in higher studies; and my brothers for their encouragement and support since the beginning of my academic journey. I am also grateful to my friends, those I met in Lens, those I left behind and those I met along the way, for their companionship and encouragement throughout these years.

Finally, I wish to acknowledge the ANR VIVAH project and the Horizon Europe STARWARS project, whose support made my research possible.

To a 7 years old kid,
who said to his teacher: "I want to become a scientist..."
(before the entire class burst out laughing!)

Contents

List of Figures viii

List of Tables x

List of Algorithms xi

Introduction 1

1 Inconsistency and Uncertainty in Propositional Logic and DL-Lite: a Short Review 9

1.1 Logic-Based Representation Languages 10

1.1.1 Propositional Logic Knowledge Bases 10

1.1.2 First Order Logic Knowledge Bases 12

1.1.3 Description Logics 15

1.2 DL-Lite Knowledge Bases 21

1.2.1 Syntax 21

1.2.2 Semantics 24

1.2.3 Reasoning tasks 25

1.3 Inconsistency-Tolerant Semantics 33

1.3.1 Standard Repair-Based Semantics 34

1.3.2 Parametrised ABox repairs 38

1.3.3 Prioritised Repairs 40

1.4 Uncertainty Management 42

1.4.1 Probability Theory 43

1.4.2 Possibility Theory 44

1.4.3 Possibility Theory applied to Propositional Logic 47

1.4.4 Possibility Theory applied to DL-Lite 49

1.5 Conclusion 52

2 The Closure-Based Partially Ordered Possibilistic Repair **53**
2.1 Partially Ordered Possibilistic Knowledge Bases . . . 54
2.1.1 Total Order Extensions of a Partial Order and Compatible ABoxes . . . 56
2.1.2 The Partially Ordered Possibilistic Repair . . . 57
2.1.3 The π-accepted Assertions . . . 59
2.2 A Closure-Based Partially Ordered Possibilistic Repair . . . 60
2.2.1 The Positive Deductive Closure . . . 60
2.2.2 Closure-Based ABox Repairs . . . 62
2.2.3 The Closure-Based Partially Ordered Possibilistic Repair . . . 65
2.2.4 Equivalent Characterisation: the Dominant Supports . . . 69
2.3 Algorithmic and Experimental Evaluation . . . 76
2.3.1 The Cπ-repair Algorithm . . . 76
2.3.2 Revisiting the Cπ-repair Algorithm . . . 82
2.3.3 Experimental Evaluation . . . 87
2.4 Conclusion . . . 91

3 The Deductive Closure and the Possibilistic Repair: The Order of Application Does Not Matter **93**
3.1 Closure-Based Possibilistic Inference from Partially Ordered Knowledge Bases . . . 94
3.1.1 Quantitative Representation of Partially Ordered Possibilistic ABoxes . . . 95
3.1.2 How to Deductively Close a Partially Ordered ABox? . . . 96
3.1.3 The cπ-acceptance Method . . . 98
3.1.4 Equivalence with the Cπ-repair . . . 99
3.2 Experimental Evaluation . . . 101
3.2.1 Experimental Setting . . . 101
3.2.2 Experimental Results . . . 104
3.3 Semantics of Partially Ordered and Inconsistent ABoxes . . . 105
3.3.1 The Preferred Interpretations of a Partially Ordered Knowledge Base . . . 106
3.3.2 Semantic Properties of the π-repair . . . 108
3.4 Conclusion . . . 110

4 Rationality Properties Analysis and Beyond the Cπ-repair **111**
4.1 Rationality Properties . . . 113
4.1.1 Unconditional Inference . . . 113
4.1.2 Conditional Inference . . . 114
4.1.3 Satisfaction of System P Properties . . . 118
4.2 On the Extension of the Cπ-repair to DL-Lite$_{\mathcal{R},\sqcap}$ . . . 125

4.2.1 The DL-Lite$_{\mathcal{R},\sqcap}$ language . . . 125
4.2.2 The Cπ-repair for DL-Lite$_{\mathcal{R},\sqcap}$ knowledge bases . . . 126
4.2.3 The result of applying the "Dominant supports" characterisation on a DL-Lite$_{\mathcal{R},\sqcap}$ knowledge base . . . 128
4.3 A Lexicographic-Based Semantics for the Cπ-repair . . . 129
4.4 Conclusion . . . 134

5 Conflict-Resolution and Explanation of Access Control Models 137
5.1 Overview of Access Control . . . 139
5.1.1 A Possible Classification of Access Control Models . . . 139
5.1.2 Organisation Based Access Control (OrBAC) . . . 141
5.1.3 Inconsistency in the OrBAC Model . . . 145
5.2 A Repair-Based Semantics for OrBAC . . . 146
5.2.1 A Possibilistic OrBAC Model . . . 147
5.2.2 The OrBAC Ontology . . . 148
5.2.3 Conflict-Resolution in OrBAC . . . 154
5.3 Explaining Access Control Decisions . . . 156
5.3.1 Logical Explanations of OrBAC Policies . . . 156
5.3.2 On Explaining OrBAC Policies using LLMs . . . 158
5.4 Conclusion . . . 166

Conclusion and perspectives 167

Bibliography 173

A Developed tools 195

B SPARQL queries of the OrBAC ontology 207

List of Figures

1.1 Overview of standard ABox repairs. 35
1.2 Syntactic vs. Semantic Inference in Possibilistic Logic. 49
1.3 The possibilistic repair $\mathcal{R}_\pi(\mathcal{A}_\geq)$. 52

2.1 The Partially Preordered ABox $\mathcal{A}_\trianglerighteq$ 56
2.2 The ABoxes $\mathcal{A}_{\geq_1}, \ldots, \mathcal{A}_{\geq_m}$ compatible with $\mathcal{A}_\trianglerighteq$ 57
2.3 The total preorder extensions $\geq_1, \ldots, \geq_{13}$ of $\trianglerighteq$. 58
2.4 The process of computing the Cπ-repair 67
2.5 The possibilistic repairs $\mathcal{R}_\pi(\mathcal{A}_{\geq_1}), \ldots, \mathcal{R}_\pi(\mathcal{A}_{\geq_m})$ of ABoxes compatible with $\mathcal{A}_\trianglerighteq$. 69
2.6 The possibilistic repairs of the total preorder extensions $\mathcal{R}_\pi(\mathcal{A}_{\geq_1}), \ldots, \mathcal{R}_\pi(\mathcal{A}_{\geq_{13}})$. 70
2.7 The dominance relation. 72
2.8 The total extension $\geq_4$ of $\trianglerighteq$. 76
2.9 The Dominant Supports Characterisation 77
2.10 Distribution of the possibilistic repairs w.r.t the ABox $\mathcal{A}_\trianglerighteq$ and its deductive closure. 84
2.11 Evolution of repair time [left] and repair size [right] w.r.t. the number of conflicts (ABox **u1**). 90
2.12 Proportion of assertions in each repair vs POSet density [left]. Proportion of repair time for ABox **u.5**, using Algorithm 5 [middle] and Algorithm 7 [right]. . . 91

3.1 The Partially Preordered ABox $\mathcal{A}_\trianglerighteq$ 96
3.2 The $c\pi$-acceptance characterisation vs. the process of computing the Cπ-repair . 102
3.3 Comparison of Cπ-repair Computation Time using (2) Characterisations 104
3.4 Syntactic vs. Semantics characterisation 105
3.5 Preference relation $\rhd^\Omega$ among interpretations. 107

4.1 Unconditional inference under the Cπ-repair 113
4.2 The ABox $\mathcal{A}_{\trianglerighteq_\mathcal{O}}$ augmented with a fully certain observation $\mathcal{O}$ 115
4.3 Conditional inference under the Cπ-repair in the presence of an observation $\mathcal{O}$ . . 116
4.4 Conditioning a partial preorder $\trianglerighteq$ vs. Conditioning its total preorder extensions $\geq_1, \ldots, \geq_m$ with an observation $\mathcal{O}$ 123
4.5 The partially preordered ABox $\mathcal{A}_\trianglerighteq$, $\longrightarrow$: the strict preference $\rhd$. 128
4.6 The process of computing $\mathsf{ICLR}(\mathcal{A}_\trianglerighteq)$ 132
4.7 The Partially Preordered ABox $\mathcal{A}_\trianglerighteq$ 133
4.8 The ABoxes $\mathcal{A}_{\geq_1}, \ldots, \mathcal{A}_{\geq_m}$ compatible with $\mathcal{A}_\trianglerighteq$ 134

5.1 The OrBAC Model [Kalam *et al.* 2003] 144
5.2 Multi-layer view of the proposed OrBAC conflict resolution framework. 147
5.3 Representation of concrete entities and the context in the OrBAC ontology . . . 150

5.4 Representation of the policy rules in the OrBAC ontology . . . 151
5.5 Representation of the connection rules in the OrBAC ontology . . . 152
5.6 OrBAC ontology example . . . 153
5.7 Preferences of the OrBAC ontology example . . . 154
5.8 Multi-layer view of the proposed explanation framework. . . . 157
5.9 Natural Language Entailment score using the `bart-large-mnli` model. . . . 160
5.10 Framework for LLM-Generated Explanations of Access Control . . . 162
5.11 Number of successful explanations per interaction (threshold= 0.6) . . . 164
5.12 Impact of interaction-based prompting per model . . . 165

A.1 Main interface of the OrBAC explanation tool. . . . 196
A.2 Loading an example policy. . . . 197
A.3 List of example policies for testing. . . . 198
A.4 List of queries to test access permissions. . . . 199
A.5 Visualisation of supports for permissions and prohibitions. . . . 200
A.6 Decision outcome after conflict resolution. . . . 201
A.7 Button to generate a natural language explanation using `google/gemma-2-9b-it`. 202
A.8 Natural language explanation produced by the LLM. . . . 203
A.9 Visualisation of the abstract rules of the policy. . . . 204
A.10 Visualisation of the connection relations of the policy. . . . 205
A.11 Checking policy consistency and computing conflicts. . . . 206

B.1 The SPARQL query ls-permitted(subject, action, object) . . . 208
B.2 The SPARQL query ls-prohibited(subject, action, object) . . . 209
B.3 The permission_supports(subject, action, object) SPARQL query . . . 210
B.4 The prohibition_supports(subject, action, object) SPARQL query . . . 211
B.5 The acceptance(subject, action, object) SPARQL query . . . 215

List of Tables

1.1 Syntax of $\mathcal{ALC}$ concept constructors. 16
1.2 Syntax and semantics of some expressive DL concept and role constructors. . . . 18
1.3 The repairs $\mathcal{R}_1, \ldots, \mathcal{R}_4$ of the ABox $\mathcal{A}$ 35

2.1 The closed ABox repairs $\mathcal{R}_1, \ldots, \mathcal{R}_4$ of the ABox $\mathcal{A}$ 63
2.2 ABoxes used in our experiments: size ($\#\mathbf{A}$), conflicts number ($\#\mathbf{Cf}$) and computation time ($\mathbf{Cf(s)}$), π-repair size ($\#\boldsymbol{\pi}$) and computation time ($\boldsymbol{\pi}\mathbf{(s)}$), C$\pi$-repair size ($\#\boldsymbol{c\pi}$) and computation time ($\boldsymbol{c\pi}\mathbf{(s)}$ and $\boldsymbol{c\pi{+}{+}}\mathbf{(s)}$). 89

3.1 ABoxes in terms of: size, percentage of assertions in conflicts, number of conflicts, and time to compute the conflict set, size of the deductive closure, time to compute the standard closure, and the partially ordered closure. 103
3.2 Some interpretations of $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq}, \mathbb{L} \rangle$ 107

5.1 SWRL rules for order propagation 153
5.2 Verbalisation templates of OrBAC rules 159
5.3 A prompt example generated for the access of Ben to read dataset1. 160
5.4 Evaluation criteria for LLM-based explanations of OrBAC decisions 161
5.5 Average metric score for the zero-shot task per model 163
5.6 Average metric score for the few-shots task per model. 166

List of Algorithms

1 PerfectRef [Calvanese *et al.* 2007] 30

2 ComputeClosure 79
3 ComputeConflicts 79
4 ComputeSupports 80
5 ComputeCπ-repair (naive) 81
6 Computeπ-repair 85
7 ComputeCπ-repair (revisited) 86

Introduction

Problem statement

Inconsistency: a Knowledge Representation and Reasoning issue

Computer and information systems deal with constantly growing amounts of data flowing in different forms from various sources. Large datasets can be organised in different ways: from structured formats, such as relational databases, to unstructured formats, such as images and videos. In this context, the field of knowledge representation and reasoning provides frameworks for modelling this knowledge with the goal of drawing useful conclusions. Yet, large datasets often contain imperfect or even contradictory information.

The methods for collecting, extracting, and storing large datasets have developed rapidly, and even the mechanisms to query such datasets are highly efficient. One can take, as an example, semantic web solutions, where ontologies allow modelling domain knowledge and the relationships between entities, datasets are stored in RDF graphs, and SPARQL endpoints allow querying the large datasets. This is the case of the DBpedia ontology [Lehmann *et al.* 2015, Crum *et al.* 2024, Association 2025], where almost (5) million resource instances [1], are stored within an RDF graph and accessible through a SPARQL endpoint [2].

One of the main challenges facing this kind of applications is dealing with contradictions in the datasets. Continuing with the example of DBpedia, the querying endpoint may return query answers that are considered contradictory (or conflicting) by the DBpedia ontology. However, no clear method is in place to deal with them. This situation highlights the need for methods to manage contradictory information and address the problem to allow a more safe and reliable reasoning from this kind of dataset.

The problem of reasoning with contradictory information has been widely studied in the literature on knowledge representation and reasoning. Different frameworks have been proposed to provide expressive ways of modelling knowledge and capturing inconsistencies. Early work efforts mainly relied on propositional logic [Finkelstein *et al.* 1994, Rescher & Manor 1970, Brewka 1989]. However, the limited expressive power of the language, mainly in terms of capturing structured knowledge involving objects, their properties, and relationships, hampered the application of the developed methods on real-world datasets. Alternatively, first-order logic has been used as a representation framework [Baral *et al.* 1992]; nevertheless, the computational cost of reasoning with it made it challenging to implement. More recently, description logics, and more precisely their lightweight fragments, have been adopted as suitable formalisms for representing and reasoning with inconsistent information [Lukasiewicz *et al.* 2022, Bienvenu & Bourgaux 2022, Bienvenu & Bourgaux 2020]. Overall, the aim here is to achieve a good trade-off between the expressive power of the language and the computational cost of reasoning.

The term "inconsistency" is widely used to denote the presence of contradictory information, or the absence of a satisfiable state, in a knowledge base. Several approaches can be followed when reasoning with inconsistency. A popular approach consists in replacing the inconsistent knowledge base with some of its consistent sub-bases, also called repairs, then evaluating the queries on some or all of them, as in consistent query answering [Koutris & Wijsen 2021, Wijsen 2019, Bienvenu & Bourgaux 2017]. Similar ideas have been explored in evidence theory, where inconsistency is handled by retracting conflicting evidence prior to fusion [Jousselme *et al.* 2021, Pinto Prieto *et al.* 2024]. An alternative approach preserves the inconsistent knowledge base and alters the inference relation, as in paraconsistent logics [Grant & Hunter 2023, Esteva *et al.* 2021, Anderson & Belnap 1975]. Other more adventurous

[1] Accessed on 25th September 2025.
[2] DBpedia SPARQL endpoint: https://dbpedia.org/sparql

approaches propose rewriting some of the knowledge base's facts to restore consistency.

A related dimension is uncertainty. Information may vary in reliability and completeness. Possibility theory offers a framework for reasoning with uncertainty and incompleteness [Denœux *et al.* 2020, Dubois & Prade 2024, Dubois & Prade 2018]. Preferences are often used to capture such uncertainty, through order-based approaches [Brewka 1989, Bienvenu *et al.* 2014, Kimelfeld *et al.* 2020], weighted logics [Perez & Wijsen 2021, Baader *et al.* 2019, de Saint-Cyr & Prade 2008], or stratified knowledge bases [Benferhat *et al.* 2015].

The idea of exploring uncertainty and prioritised datasets to handle inconsistency takes advantage of the preference of some data pieces over others to enable a more informed selection of the consistent subsets. This has been achieved in several *inconsistency-tolerant semantics*, such as the preferred repair semantics [Bienvenu *et al.* 2014], the possibilistic repair and the non-defeated repairs [Benferhat *et al.* 2015], the linear-based repair [Nebel 1994], and the optimal repairs [Staworko *et al.* 2012, Bienvenu & Bourgaux 2020].

However, despite the numerous representation frameworks and available solutions, very few approaches focussed on partially ordered information in the context of reasoning with inconsistent information. Partial orders are a means to specifying a preference relation over a dataset when some data pieces are incomparable. On top of being a more general case of total orders, partial orders provide a flexible approach when it is not feasible to impose a total order on all the information pieces. They are mostly useful when only a part of a priority relation is known. Thus, known relations can be expressed, while missing relations are treated as cases of incomparability, allowing for the management of data that cannot be fully ordered.

A partial order over a dataset could be the result of multiple information sources providing distinct data and each source applying its own total order. This situation is exemplified by Brewka's work [Brewka 1989], which highlights how to derive a partial order from total specificity relations, but applied individually within different parts of a knowledge base. In the same context, distinct information sources may totally order the same dataset in different ways, based on varying criteria, such as creation timestamps or the reliability of the source itself. When combined, these different total orders can induce a partial order on the same dataset.

The main challenge with partial orders is handling incomparability. If two contradictory data pieces are comparable, one can be preferred. If they are incomparable, it is unclear which to keep in a repair. A fundamental idea when dealing with a partially ordered dataset is to extend the partial order into a family of compatible total orders, where incomparabilities are replaced with a different regular preference each time to obtain the total orders. Then, apply standard inconsistency handling techniques to each. The challenge here becomes computational, as the number of compatible total orders can be exponential, in the worst case, in the number of incomparabilities present in a partial order.

Two notable approaches address this challenge in lightweight description logics: the Elect method [Belabbes *et al.* 2021] and the partially ordered possibilistic repair [Belabbes & Benferhat 2021]. Both methods follow the above strategy of extending the partial order into a family of total orders, before applying a repair semantics to each obtained totally ordered set. Then, all the obtained repairs are intersected to provide a single repair. Both methods are efficient, yet, a main drawback in their process is the loss of some meaningful consequences that follow from all the repairs of the totally ordered sets when performing intersection to obtain a single repair that is ready for us to reason from it.

Contributions

The main framework used for the contributions of this thesis is formal ontologies and, in particular, lightweight description logics (also called DL-Lite), which underlies the Ontology Web Language (OWL 2) QL profile [Motik *et al.* 2012] and offers interesting computational properties. A DL-Lite knowledge base typically consists of two components: conceptual knowledge, captured in the TBox, and factual knowledge, represented in the ABox. The TBox contains axioms, while the ABox contains assertions. In this context, inconsistency is defined by the absence of a model for the knowledge base and is usually studied under the assumption that the TBox is well-debugged and conflict-free; whereas, the ABox is very large, constantly changing, and more prone to errors. Hence, an ABox is said to be inconsistent with respect to the TBox.

- **The Closure-Based Partially Ordered Possibilistic Repair (Cπ-repair).** The first contribution of this thesis is the proposal of a new tractable inconsistency-tolerant semantics for partially ordered DL-Lite knowledge bases. The central focus of our proposal is on the preservation of meaningful consequences in the repair process. In essence, we investigate the popular practice of deductively closing the repairs, i.e., adding to an ABox all assertions that logically follow from it, to efficiently provide a single repair for a partially ordered ABox. Our method proceeds by extending the partial order over the ABox into a family of compatible total orders to obtain as many totally ordered ABoxes. Then, it applies the standard possibilistic repair, which accounts for uncertainty to provide a single repair for each totally ordered ABox. At this level intervenes the deductive closure, which adds all assertions that can be inferred by each repair to it, before intersecting the closed repairs to render a single partially ordered repair. The obtained repair is larger than the repair computed using the partially ordered possibilistic repair method [Belabbes & Benferhat 2021]. This result underscores how our method avoids the loss of meaningful consequences.

- **An Efficient Equivalent Characterisation Using Dominant Supports.** The efficiency of our method is established through an equivalent characterisation, which avoids enumerating the total order extensions of a partial order in the process. It mainly relies on the notion of conflicts, which are inclusion-minimal inconsistent subsets of the ABox, the notion of supports, which are inclusion-minimal consistent subsets of the ABox that allow for the derivation of a given assertion, and the dominance relation, which extends the partial order defined over ABox assertions into a partial order defined over subsets of assertions. Intuitively, valid conclusions are those that have some dominant support against each conflict of the ABox.

- **The Partially Ordered Deductive Closure for Possibilistic Repairs: the Order of Application Does Not Matter.** We introduce a novel deductive closure operator for partially ordered ABoxes that safely applies the deductive closure as the first step of computing a repair for partially ordered ABoxes. The operator accounts for all preferences of inferring assertions, including those derived from incomparable assertions. The operator ensures that a single repair is obtained, equivalent to computing and closing all possibilistic repairs of the compatible total orders (Cπ-repair), but is significantly more efficient as it reduces reasoning to a series of the most basic reasoning task: consistency checking, ensuring that no conclusions follow solely from conflicting facts. Experimental evaluation shows that computing the closure before repairing improves performance compared to previous characterisations.

- **Analysis of Rationality Properties for Partially Ordered Possibilistic Repairs.** In order to complete the study of handling inconsistency in partially ordered DL-Lite knowledge

bases, we propose analysis of rationality properties for inference under the proposed method: the Cπ-repair, and its unclosed version: the π-repair [Belabbes & Benferhat 2021]. The examination of these methods compliance with a set of well-established rationality properties shows the satisfaction of unconditional inference properties. In addition, we provide an analysis of conditional inference under a fully reliable new set of assertions, also referred to as an observation. In particular, our analysis shows that the satisfaction of the properties when possibilistic query-answering is performed over the intersection of the unclosed totally ordered repairs is preserved when applying the deductive closure in the repair process.

- **Beyond DL-Lite$_\mathcal{R}$ and the Possibilistic Semantics.** Two additional steps are explored to go beyond our method: the application to richer description logics and alternative semantics. We show how it can be applied beyond the DL-Lite$_\mathcal{R}$ setting by analysing its application in the context of a richer language that allows for more expressive TBoxes, mainly by allowing conjunction in the axioms. Then, we discuss its relation to other repair semantics such as the lexicographic approach, highlighting the tractability challenges that arise.

- **Algorithmic and Experimental evaluation.** At the algorithmic level, we propose a sound and complete algorithm for the characterisation. Then, we refine our proposition by studying potential improvements to the algorithm, resulting in a more practically efficient way of computing repairs by exploiting the notions of preferred supports and preferred conflicts, in addition to the inclusion relationship between the new repair and some of its subsets. At the experimental level, we study the impact of data size, data quality, and different configurations of partial orders to emphasise the practical benefits of the refined repair algorithm.

- **Applying Inconsistency-Tolerant Repairs to Access Control Conflicts.** We demonstrate how our inconsistency-tolerant repair framework can be applied to access control models to systematically resolve conflicts between permissions and prohibitions. The method ensures that the rules supporting a granted access dominate those supporting a denied one, providing a formal, principled mechanism for conflict resolution in complex policies, without restricting expressiveness or interpretability.

- **On Explaining Dynamic Access Control Policies using LLMs.** Finally, we explore the task of generating explanations for access control decisions using Large Language Models (LLMs). Our approach grounds natural language explanations in formal logical reasoning, which leads to the proposition of an automatic evaluation strategy to assess the accuracy of natural language explanations. Experiments demonstrate the promise of this methodology, further enhanced by interactive prompting techniques. This contribution constitutes a first step in a framework that takes advantage of LLMs to generate natural language explanations in a reliable way, particularly when human-referenced explanations are costly or unavailable.

Organisation of the thesis

The manuscript is organised as follows:

Chapter 1 This chapter recalls the preliminaries of knowledge representation systems, going through representation using propositional logic, first-order logic, and description logics, before focussing on DL-Lite$_\mathcal{R}$ knowledge bases. Then, it reviews some of the well-known semantics that have been proposed to deal with inconsistent knowledge bases. Finally, it recalls the basics of possibility theory, possibilistic logic, and possibilistic DL-Lite$_\mathcal{R}$ knowledge bases.

Chapter 2 This chapter presents the first contribution of the thesis, which is the closure-based partially preordered possibilistic repair method, called the (Cπ-repair). Then, it shows its capacity to return more conclusions than its unclosed version, which is also referred to by productivity, thanks to the use of the deductive closure in the repair process. The tractability of the method is established using the notions of support and dominance, which allow us to define an equivalent characterisation that guarantees the computation of the Cπ-repair in polynomial time for DL-Lite$_\mathcal{R}$ knowledge bases. Finally, this chapter is concluded with algorithmic and experimental studies of the proposed semantics.

Chapter 3 The third chapter proposes an alternative approach to apply the deductive closure to partially preordered inconsistent DL-Lite$_\mathcal{R}$ knowledge bases. This yields a new, yet more efficient repair process that is equivalent to the Cπ-repair method. An experimental evaluation allows us to showcase the efficiency of the new method compared to the first characterisation.

Chapter 4 This chapter investigates some potential ways to go beyond the proposed repair semantics. These consist in studying the rationality properties of the proposed methods, considering description logics that are more expressive than DL-Lite$_\mathcal{R}$, and another inconsistency-tolerant semantics, which is the lexicographic preferred-based repair. This semantics was initially defined for totally preordered ABoxes, where a cardinality-based criterion is used to select preferred repairs of an inconsistent ABox.

Chapter 5 The last chapter applies our inconsistency-tolerant semantics to the domain of access control, with a focus on the Organisation-Based Access Control (OrBAC) model. The chapter introduces a repair-based conflict resolution method to handle contradictory permissions and prohibitions in OrBAC policies. It also extends the model with a natural language explanation framework, leveraging Large Language Models (LLMs) to generate clear and user-friendly justifications for access decisions, particularly in conflict scenarios.

Appendix A This appendix presents screenshots of the OrBAC explanation tool developed from the results of the last chapter of this thesis. It illustrates interface, policy visualisation, conflict resolution, and natural language explanation generations.

Appendix B This appendix depicts the SPARQL queries used in the implementation of the conflict resolution method defined for the Organisation-Based Access Control model.

Publications achieved in this thesis

International journal papers

1. Ahmed Laouar, Sihem Belabbes, and Salem Benferhat. « How to tractably compute a productive repair for possibilistic partially ordered DL-Lite_R ontologies? » In **FSS 2025**: **F**uzzy **S**ets and **S**ystems, 510:109361. ISSN: 0165-0114. DOI : https://doi.org/10.1016/j.fss.2025.109361

International conference papers

1. Ahmed Laouar and Salem Benferhat. « Closure-Based Tractable Possibilistic Inference from Partially Ordered DL-Lite Ontologies ». In **JELIA 2025**: European Conference on Logics in Artificial Intelligence. DOI : https://doi.org/10.1007/978-3-032-04587-4_15

2. Ahmed Laouar, Toky Raboanary, and Salem Benferhat. « An Ontology-Based Approach for Handling Inconsistency in Explainable and Prioritized Access Control Model ». In **SUM 2024**: **S**calable **U**ncertainty **M**anagement, pages 249-264. DOI : https://doi.org/10.1007/978-3-031-76235-2_19

3. Ahmed Laouar, Sihem Belabbes, and Salem Benferhat. « Conflict handling strategies for partially ordered access control security policies ». In **PACRIM 2024**: 2024 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing. IEEE. 2024, pp. 1–6. DOI: https://doi.org/10.1109/PACRIM61180.2024.10690228

4. Ahmed Laouar, Sihem Belabbes, and Salem Benferhat. « On the Computation of a Productive Partially Ordered Possibilistic Repair ». In **DL 2024**: 37th International Workshop on Description Logics, volume 3739. DOI: https://ceur-ws.org/Vol-3739/paper-6.pdf

5. Ahmed Laouar, Sihem Belabbes, and Salem Benferhat. « Tractable Closure-Based Possibilistic Repair for Partially Ordered DL-Lite Ontologies ». In **JELIA 2023**: European Conference on Logics in Artificial Intelligence, pages 353–368, Cham, 2023. Springer Nature Switzerland. DOI: https://doi.org/10.1007/978-3-031-43619-2_25

National conference papers

1. Sihem Belabbes, Salem Benferhat, and Ahmed Laouar. « Calcul traitable de la réparation fermée possibiliste pour les ontologies partiellement pré-ordonnées ». In **LFA 2022**: rencontres francophones sur la Logique Floue et ses Applications. URL https://www.cepadues.com/nos-articles/1012/

Chapter 1

Inconsistency and Uncertainty in Propositional Logic and DL-Lite: a Short Review

Contents

1.1 Logic-Based Representation Languages 10
1.1.1 Propositional Logic Knowledge Bases 10
1.1.2 First Order Logic Knowledge Bases 12
1.1.3 Description Logics 15
1.2 DL-Lite Knowledge Bases 21
1.2.1 Syntax 21
1.2.2 Semantics 24
1.2.3 Reasoning tasks 25
1.3 Inconsistency-Tolerant Semantics 33
1.3.1 Standard Repair-Based Semantics 34
1.3.2 Parametrised ABox repairs 38
1.3.3 Prioritised Repairs 40
1.4 Uncertainty Management 42
1.4.1 Probability Theory 43
1.4.2 Possibility Theory 44
1.4.3 Possibility Theory applied to Propositional Logic 47
1.4.4 Possibility Theory applied to DL-Lite 49
1.5 Conclusion 52

In the field of knowledge representation and reasoning, research efforts focused on developing methods to capture high-level descriptions of a domain of interest. In this context, knowledge-based systems are those capable of finding implicit consequences of explicitly represented knowledge. The research community makes a distinction between logic-based and non-logic-based representations [Baader *et al.* 2007]. In logic-based representations, reasoning mainly aims at verifying consequences in a very natural way. For instance, first-order logic is designed to support logical deduction. In contrast, non-logic-based representations provide an intuitive way

(e.g. graph structures) to visualise connections and relationships between elements of the studied domain, like in semantic networks or Bayesian networks [Darwiche & Pearl 1997].

Capturing inconsistency in knowledge systems relies on the use of formal languages to represent information about a domain of interest. The expressive power of a formal language is often governed by its syntax, which allows forming expressions and formulas, and by its semantics, which define the interpretation and meaning of the expressions. Depending on the studied type of inconsistency, one must choose the right representation language without neglecting the complexity of the reasoning behind it. The chosen language is also the basis for a set of targeted reasoning mechanisms which support consistency checking and drawing safe and reliable conclusions.

This chapter provides a comprehensive overview of the primary frameworks that support reasoning with inconsistent information, with a particular focus on propositional logic and description logic (DL). The main challenge in resolving inconsistency resides in maintaining a good balance between the expressive power of the used representation language and the complexity of reasoning. Therefore, a special focus is given to methods that achieve a reasonable balance. In particular, it focuses on the approaches defined for lightweight ontologies (DL-Lite).

1.1 Logic-Based Representation Languages

Among a wide spectrum of representation solutions, logic-based languages are a standard way to represent knowledge in a structured and formalised way. Some of the well-known logics include Propositional Logic, First-Order Logic, Description Logics, Rule-Based Languages (like Datalog), and Modal and Temporal Logics.

1.1.1 Propositional Logic Knowledge Bases

Propositional logic (PL) is a long-standing approach to foundational logical reasoning. It allows modelling simple knowledge domains and relies on truth values assignment to support reasoning (e.g. [Gallier 2015]).

Syntax. A propositional logic vocabulary $\mathcal{V}$ is built upon an alphabet, consisting of propositional variables, logical connectives, and boolean constants to denote the truth values: true ($\top$) and false ($\bot$). Propositional variables are often denoted by letters as p, q, and r, to which a boolean truth value (true or false) is assigned. The connectives are negation ($\neg$), conjunction ($\wedge$), disjunction ($\vee$), implication ($\rightarrow$) and equivalence ($\leftrightarrow$). Parentheses are used for grouping.

A propositional language is denoted by $\mathcal{L}$. The above alphabet is used recursively to build well-formed formulas (or propositional formulas). For instance, an element of the vocabulary $p \in \mathcal{V}$ is a formula, a conjunction of formulas $r \wedge q$, where $r, q \in \mathcal{V}$, is also a formula, and so on.

A literal is either a propositional variable p (positive literal) or its negation $\neg p$ (negative literal). A clause is a finite disjunction (OR) of literals, e.g. $p \vee \neg q \vee r$. A term, or a cube (in this context) is a finite conjunction (AND) of literals, e.g. $p \wedge \neg q \wedge r$. A formula is said to be in Conjunctive Normal Form (CNF), if it is formed by a conjunction of clauses, e.g. $(p \vee \neg q) \wedge (\neg r \vee s)$. A formula is said to be in Disjunctive Normal Form (DNF), if it is formed by a disjunction of terms, e.g. $(p \wedge \neg q) \vee (\neg r \wedge s)$.

Example 1.1. *Let us consider an academic example to model information about a data access management domain. The idea is to represent information about the roles of system users, the types of resources, and the relations between users and resources. Let us start with the following propositions which indicate the roles of users:*

- $p =$ FullResearcher
- $q =$ User
- $r =$ Secondee

One can express some formulas from the above propositions. For instance, $(p \wedge q) \rightarrow \neg r$*, which translates to:*

"If someone is a FullResearcher *and a* User*, then they are not a* Secondee*."*

Semantics. The semantics of propositional logic consists in evaluating formulas to their truth values (true or false). It is given in terms of interpretations. An interpretation $\mathcal{I}$ is a function that maps each propositional variable to either true or false: let $\mathcal{P}$ be the set of propositional variables, then $\mathcal{I} : \mathcal{P} \rightarrow \{\text{true}, \text{false}\}$. The set of interpretations is denoted by Ω. Interpretations are extended to propositional formulas $\phi, \psi, \ldots$, as follows:

$$\begin{aligned}
\mathcal{I}(\top) &= \text{true} \quad \text{and} \quad \mathcal{I}(\bot) = \text{false} \\
\mathcal{I}(\neg\phi) &= \text{true} \quad \text{iff} \quad \mathcal{I}(\phi) = \text{false} \\
\mathcal{I}(\phi \wedge \psi) &= \text{true} \quad \text{iff} \quad \mathcal{I}(\phi) = \text{true and } \mathcal{I}(\psi) = \text{true} \\
\mathcal{I}(\phi \vee \psi) &= \text{true} \quad \text{iff} \quad \mathcal{I}(\phi) = \text{true or } \mathcal{I}(\psi) = \text{true} \\
\mathcal{I}(\phi \rightarrow \psi) &= \text{true} \quad \text{iff} \quad \mathcal{I}(\phi) = \text{false or } \mathcal{I}(\psi) = \text{true} \\
\mathcal{I}(\phi \leftrightarrow \psi) &= \text{true} \quad \text{iff} \quad \mathcal{I}(\phi) = \mathcal{I}(\psi)
\end{aligned}$$

A formula ϕ is said to be satisfiable if there exists at least an interpretation that makes it true. An interpretation $\mathcal{I}$ satisfies ϕ, denoted $\mathcal{I} \models \phi$ if and only if $\mathcal{I}(\phi) = \text{true}$. Such interpretation is called a model of ϕ. Otherwise, $\mathcal{I}$ is a counter-model of ϕ, denoted $\mathcal{I} \not\models \phi$. A formula is said to be a tautology if it is true under all interpretations, and it is said to be a contradiction if it is false under all interpretations. Two formulas are said to be logically equivalent if they have the same set of models.

Reasoning. The main reasoning services in propositional logic include entailment, satisfiability, tautology checking, and contradiction detection. Conclusions are logically entailed as follows. Let ϕ_1 and ϕ_2 be two propositional logic formulas. ϕ_2 is said to be entailed by ϕ_1, denoted $\phi_1 \vdash \phi_2$, if for every interpretation $\mathcal{I}$ that satisfies ϕ_1, $\mathcal{I}$ also satisfies ϕ_2.

Checking whether a set of formulas logically entails another formula is achieved using resolution or truth tables. Using truth tables, in the worst case, one has to enumerate 2^n interpretations to find a model, with n being the number of propositional variables in the formula. Using resolution, one attempts to derive a contradiction by refuting the negation of the formula to be proved, using a complete and sound inference rule based on clause manipulation. Namely, ϕ_2 is a logical consequence of ϕ_1 if and only if $\phi_1 \wedge \neg\phi_2$ is unsatisfiable. This reduces the problem

of entailment to the satisfiability problem (SAT), which asks whether there exists a truth assignment that makes a propositional formula true. The complexity of the satisfiability problem of a set of propositional formulas (SAT) is NP-complete [Cook 1971], and the problem is a cornerstone of computational complexity.

The main drawback of propositional logic is its limitation in terms of expressiveness. In essence, it does not allow for the representation of quantifiers and relationships between objects. As a result, real-world complex domains are hard to model using propositional logic. However, it serves as a basis or is embedded within more expressive logics such as First-Order Logic and Description Logic.

1.1.2 First Order Logic Knowledge Bases

Propositional logic allows us to express truth values about statements. However, it does not deal with individual objects or their relationships. Hence, it is not expressive enough for most real-world reasoning. First-order logic (FOL) overcomes the limitations of propositional logic in terms of expressiveness, mainly by allowing the representation of statements, or assertions, about domain elements. It allows the representation of structured knowledge about objects, their properties and relationships. First-order logic is also known as predicate logic [Gallier 2015].

A first-order logic vocabulary is constructed from finite sets CS, FS, PS, and V, where CS is a set of constant symbols (or individuals) denoted $a_0, a_1, \ldots$. FS is a set of function symbols $f_0, f_1, \ldots,$. PS is a set of predicate symbols $P_0, P_1, \ldots,$. V is a set of variable names $x_1, x_2, \ldots,$. The sets CS, FS, and PS are countable sets.

The sets V, FS, CS, and PS are assumed to be mutually disjoint. The set of terms is defined such that: every constant in $a \in \mathsf{CS}$ and every variable in $x \in \mathsf{V}$ is a term. Each function or predicate symbol is associated with an arity denoting the number of arguments it takes. If f is an n-ary function and $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is also a term. Note that a 0-ary function is called a constant. Terms are used as arguments for predicates to form atomic formulas. For instance, $P(x_1, x_2, x_3)$ is a ternary predicate (predicate of arity 3).

An atom ϕ is an expression of the form $P(t_1, \ldots, t_n)$, such that $P \in \mathsf{PS}$ and $t_1, \ldots, t_n$ are terms as given above. A first-order language, denoted by $\mathcal{L}$, is built on a set of atoms, a set of logical connectives $(\neg)$, $(\wedge)$, $(\vee)$, $(\rightarrow)$ and $(\leftrightarrow)$, as in propositional logic, in addition to the universal $\forall$ and existential $\exists$ quantifiers, and the 0-ary predicates $\top$ and $\bot$.

Similarly to propositional logic, atoms and connectives are used recursively to build FOL formulas, as follows:

- $\top$ and $\bot$ are formulas.
- Each atom is a formula (called an atomic formula).
- For any two formulas ϕ_1 and ϕ_2, $(\phi_1 \wedge \phi_2)$, $(\phi_1 \vee \phi_2)$, $(\phi_1 \rightarrow \phi_2)$, $(\phi_1 \leftrightarrow \phi_2)$, and $\neg\phi_1$ are also well-formed formulas.
- For any variable x_i and any formula ϕ, $\forall x_i.\phi$ and $\exists x_i.\phi$ are also formulas.
- Let $=$ be a binary predicate symbol, the formulas $=(x, y)$ and $\neg(x = y)$ are called equality and inequality, and are simply denoted by $x = y$ and $x \neq y$.

Example 1.2. *Let us adapt elements of the vocabulary of Example 1.1 to first-order logic, using a data access management domain:*

- *Constants are used to refer to specific individuals, such as* Duane *(a user) and* r_1 *(a resource).*
- *Variables, such as* x*,* y*,* z*, range over individuals in the domain.*
- *Predicate symbols represent properties or relations. For instance:*
  - *Unary predicates:* $\mathsf{FullResearcher}(x)$ *means "*x *is a full researcher";*
  - *Binary predicates:* $\mathsf{owns}(x, y)$ *means "*x *owns* y*";* $\mathsf{hasConfidAccess}(x, y)$ *means "*x *has confidential access to* y*".*
- *Atoms are formed by applying predicates to terms (constants or variables), such as* $\mathsf{Confidential}(\mathsf{r}_1)$ or $\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_1)$.
- Complex formulas are built using logical connectives and quantifiers over atoms.

A sub-formula of a formula ϕ is any part of ϕ that is itself a formula, including ϕ itself. Sub-formulas are useful in syntactic analyses, such as determining variable occurrences or logical equivalences.

The scope of a quantifier (either $\forall$ or $\exists$) is the subformula to which the quantifier applies. For example, in the formula $\forall x.(P(x) \to Q(y))$, the scope of the quantifier $\forall x$ is the formula $(P(x) \to Q(y))$.

A variable occurrence in a formula is said to be bound if it lies within the scope of a quantifier that binds it (i.e., $\forall x$ or $\exists x$); otherwise, it is free. For instance, in the formula $\forall x.(P(x) \wedge Q(y))$, the variable x is bound, while y is free. A formula in which all variable occurrences are bound is called a closed formula (or a sentence), and a formula that contains at least one free variable is called an open formula.

A term that contains no variables is called a ground term. Similarly, a formula that contains no variables (i.e., a closed formula with only ground terms) is called a ground formula. These play an important role in automated reasoning and logic programming, as ground formulas have a fixed meaning once all symbols are defined.

FOL supports expressive queries over knowledge bases. For instance, conjunctive queries and logic programming with FOL fragments. It is often used to express queries in the form of formulas with free variables. These formulas act as templates for retrieving variable assignments (answers) that make the formula true in a given model. For example, a conjunctive query like $\exists y., P(x, y) \wedge Q(y)$ retrieves the values of x for which the pattern holds. Queries are central to applications such as databases, semantic web reasoning, and logical inference, where retrieving meaningful information from structured knowledge is essential.

Example 1.3. *Let us continue Example 1.2 by introducing some FOL formulas:*

- *The formula* $\exists y., \mathsf{owns}(x, y) \wedge \mathsf{Confidential}(y)$ *can be interpreted as a query asking: "Is there a resource* y *such that* y *is owned by* x *and* y *is confidential?" The answer would be a set of individuals* x *for which the formula holds in the model.*
- *The formula* $\mathsf{hasConfidAccess}(x, y) \wedge \mathsf{owns}(z, y)$ *is a query with three free variables, asking: "Which triples* (x, y, z) *satisfy the fact that* x *has confidential access to* y *and* y *is owned by* z*?".*

Such queries enable expressive retrieval patterns, making FOL a powerful tool for representing and querying knowledge.

The semantics of first-order logic is defined with respect to a structure $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty domain of discourse (a set of objects), and $\cdot^{\mathcal{I}}$ is an interpretation function that assigns:

- a specific object in $\Delta^{\mathcal{I}}$ to each constant symbol,
- a function over $\Delta^{\mathcal{I}}$ to each function symbol,
- a relation over $\Delta^{\mathcal{I}}$ to each predicate symbol.

Variables are assigned values via variable assignments σ. The truth of a formula under an interpretation and an assignment, denoted $\mathcal{I}, \sigma \models \phi$, is defined inductively. A sentence (closed formula) is said to be true in a model if it holds under all variable assignments. If a formula is true in every model, it is called logically valid.

The satisfaction relation $\mathcal{I}, \sigma \models \phi$ is defined inductively as follows, where $\mathcal{I}$ is an interpretation and σ is a variable assignment:

- $\mathcal{I}, \sigma \models P(t_1, \ldots, t_n)$ if $(t_1^{\mathcal{I}}[\sigma], \ldots, t_n^{\mathcal{I}}[\sigma]) \in P^{\mathcal{I}}$
- $\mathcal{I}, \sigma \models t_1 = t_2$ if $t_1^{\mathcal{I}}[\sigma] = t_2^{\mathcal{I}}[\sigma]$
- $\mathcal{I}, \sigma \models \neg\phi$ if $\mathcal{I}, \sigma \not\models \phi$
- $\mathcal{I}, \sigma \models (\phi_1 \wedge \phi_2)$ if $\mathcal{I}, \sigma \models \phi_1$ and $\mathcal{I}, \sigma \models \phi_2$
- $\mathcal{I}, \sigma \models (\phi_1 \vee \phi_2)$ if $\mathcal{I}, \sigma \models \phi_1$ or $\mathcal{I}, \sigma \models \phi_2$
- $\mathcal{I}, \sigma \models (\phi_1 \rightarrow \phi_2)$ if $\mathcal{I}, \sigma \not\models \phi_1$ or $\mathcal{I}, \sigma \models \phi_2$
- $\mathcal{I}, \sigma \models \forall x.\phi$ if for every $d \in \Delta^{\mathcal{I}}$, we have $\mathcal{I}, \sigma[x \mapsto d] \models \phi$
- $\mathcal{I}, \sigma \models \exists x.\phi$ if there exists $d \in \Delta^{\mathcal{I}}$ such that $\mathcal{I}, \sigma[x \mapsto d] \models \phi$

Here, the notation $t^{\mathcal{I}}[\sigma]$ denotes the interpretation of a term t under $\mathcal{I}$ and the assignment σ. The assignment $\sigma[x \mapsto d]$ denotes the function that maps x to d and behaves like σ on all other variables.

First-order logic is a core formalism for many knowledge-based systems, which allows reasoning over domains with individuals and predicates. It is an expressive logic that can represent arbitrary properties, relationships, and quantification over individuals. In addition, it supports reasoning services such as deduction, model checking, unification and proof systems. However, FOL is undecidable in general, which means that there is no complete algorithm to determine the satisfiability of arbitrary FOL formulas [Gallier 2015].

For instance, consistency checking is possible, but not always decidable, and requires syntactic restrictions. Entailment is achieved using natural deduction, resolution (the first-order version), or tableaux. However, it is semi-decidable, which means that one can demonstrate if a formula is valid in a finite number of steps, but if it is falsifiable, the procedure may run forever [Gallier 2015].

Finally, for representation languages, a trade-off (or a middle-ground) needs to be achieved between expressiveness and tractability. FOL is highly expressive, but comes with high computational cost and undecidability. FOL also serves as the foundation for logic-based AI, formal verification, ontology languages (e.g. OWL2 DL), logic programming, databases, and extensions like non-monotonic logic and modal logics.

1.1.3 Description Logics

Description logics (DLs) are a family of logic formalisms that support structured knowledge representation and automated reasoning. The primary purpose of DLs is to describe conceptual knowledge about a domain of interest in a formal and declarative way. Although DLs originate from network-based representation structures, such as semantic networks and frame-based systems, they are equipped with model-theoretic semantics and are considered decidable subsets of FOL [Baader *et al.* 2007]. DLs have been employed in several application domains, ranging from documentation systems, data integration, and databases to the semantic web, where they serve as the basis for ontology languages such as OWL 2 [Baader *et al.* 2017, Baader *et al.* 2007]. In fact, since their use in semantic web languages, DL knowledge bases have been referred to as ontologies. Furthermore, the ontology engineering application area has multiple lines of research such as ontology debugging [Ji *et al.* 2022], and ontology alignment [He *et al.* 2022], in addition to the ontology-based data access (OBDA) [Xiao *et al.* 2018], and ontology-mediated query answering (OMQA) [Bienvenu *et al.* 2019] paradigms.

Syntax

A knowledge base in the context of DLs is built upon a vocabulary consisting of three pairwise disjoint and countably infinite sets of symbols, as follows:

- $\mathsf{N_C}$ is the set of concept names which are unary predicates and represent objects that share some common characteristics.
- $\mathsf{N_R}$ is the set of role names which are binary predicates and represent relationships between objects.
- $\mathsf{N_I}$ is the set of individuals or constants referring to specific objects in the discourse domain.

Example 1.4. *Concept names can be used to represent the roles of users and the types of objects in our data access management system. For instance, some examples of concept names include* User, FullResearcher, Secondee, *and* Confidential. *Role names are used to express relations between users and resources, such as* hasAccess, hasConfidAccess, *and* hasOpenAccess, *which describe the types of access to resources by users. Individual names represent specific domain entities such as* Duane *and* dataset1.

DL Knowledge Bases

Description logics make a clear distinction between intensional and extensional knowledge. The intensional knowledge represents general knowledge about the domain of interest and is contained within a TBox, where the 'T' in TBox refers to terminological knowledge. The knowledge in the TBox consists of declarations that describe general properties of concepts, in the form of subsumption relationships among the concepts that constitute the terminology. Hence, the

TBox is also referred to as conceptual knowledge, and the subsumption relationships are called axioms. Extensional knowledge represents ground facts on specific individuals in the domain of discourse. These facts are given in the form of membership assertions, from which the name ABox is given to denote extensional knowledge.

A DL knowledge base (KB) is a pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ where $\mathcal{T}$ is a finite set of terminological axioms (TBox) and $\mathcal{A}$ is a finite set of membership assertions (ABox).

DLs use formal grammar to form complex concepts and roles from concept and role names in $\mathsf{N_C}$ and $\mathsf{N_R}$. The (complex) concepts and roles are then used to form the TBox axioms, which are also called concept (or role) inclusions. Depending on the allowed constructors in the TBox axioms, one can obtain a different fragment of DLs [Baader *et al.* 2007].

To provide a general idea of how DL TBoxes and ABoxes are formed, a brief presentation of the prototypical expressive description logic $\mathcal{ALC}$ (*Attributive Language with Complement*) is given in the following. It is a well-studied fragment and helps ground more advanced or restricted fragments. Then, we provide some further notions used in more expressive DLs. Later in this chapter, a more specific language, namely DL-Lite$_\mathcal{R}$, will be presented and discussed in detail, as it is the fragment used for the main contributions of this thesis.

$\mathcal{ALC}$ focusses mainly on the construction of complex concepts, which are formed as given in Table 1.1, where C, C_1, C_2 are concepts, and P is a role. An atomic concept is a concept name $A \in \mathsf{N_C}$, and an atomic role is a role name $P \in \mathsf{N_R}$. The logical connectives used include negation, conjunction, disjunction, qualified universal quantification, and qualified existential quantification.

Table 1.1: Syntax of $\mathcal{ALC}$ concept constructors.

Name	**Syntax**	**Description**
Top concept	$\top$	The set of all individuals
Bottom concept	$\bot$	The empty set
Atomic concept	A	A named concept (e.g. Person, Animal)
Negation	$\neg C$	The complement of a concept C
Conjunction	$C_1 \sqcap C_2$	Member of both concepts C_1 and C_2
Disjunction	$C_1 \sqcup C_2$	Member of at least one of C_1 or C_2
Existential restriction	$\exists P$	Related by P to some individual
Qualified existential restriction	$\exists P.C$	Related by P to some individual in C
Qualified universal quantification	$\forall P.C$	Related by P only to individuals in C

Note that $\exists P$ is a widely used abbreviation of $\exists P.\top$ and $\forall P$ is also an abbreviation of $\forall P.\top$.

As mentioned earlier, DL TBox axioms come in the form of concept inclusions. An $\mathcal{ALC}$ TBox $\mathcal{T}$ contains a finite set of inclusion axioms on concepts of the form:

$$C_1 \sqsubseteq C_2 \qquad\qquad C_1 \equiv C_2 (C_1 \sqsubseteq C_2 \text{ and } C_2 \sqsubseteq C_1)$$

Example 1.5. *The following are examples of $\mathcal{ALC}$ TBox axioms, using our data access management vocabulary (where concept and role names stand for their respective obvious meanings):*

1. $\mathsf{FullResearcher} \sqsubseteq \mathsf{User}$
2. $\mathsf{Secondee} \sqsubseteq \neg\, \mathsf{FullResearcher}$
3. $\exists \mathsf{hasOpenAccess} \sqsubseteq \exists \mathsf{hasAccess} \sqcap \mathsf{User}$
4. $\exists \mathsf{hasConfidAccess} \equiv \exists \mathsf{hasAccess}.\mathsf{Confidential}$

Axiom 1 states that every FullResearcher *is also a* User*, implying that full researchers are a type of users. Axiom 2 expresses that* Secondee *and* FullResearcher *are disjoint concepts: no individual can simultaneously belong to both categories. Axiom 3 specifies that any individual who has open access to some resource must be both a user and have access to that resource. This constrains open access to be a form of general access granted only to users. Axiom 4 defines confidential access precisely as having access to a resource that is considered* Confidential*. It equates confidential access with access to confidential resources.*

In addition, the ABox $\mathcal{A}$ is a finite set of membership assertions (or ground facts) of individuals of the form:

$$C(a) \qquad\qquad P(a,b)$$

Example 1.6. *Examples of $\mathcal{ALC}$ ABox assertions include the following:*

$$\mathsf{FullResearcher}(\mathsf{Ben}), \qquad \mathsf{Confidential}(\mathsf{r}_1),$$
$$\mathsf{hasConfidAccess}(\mathsf{Duane}, \mathsf{r}_1), \qquad \mathsf{hasOpenAccess}(\mathsf{Duane}, \mathsf{r}_2)$$

The first line of assertions are concept assertions that indicate individuals membership to concepts, they indicate, respectively, that Ben is a full researcher and resource r_1 *is confidential.*
The second line contains role assertions that indicate relationships between individuals. They indicate, respectively, that Duane has confidential access to r_1 *and Duane has open access to* r_2*.*

Semantics

The semantics of DLs is given in terms of interpretations. An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty domain $\Delta^{\mathcal{I}}$ (the set of individuals) and an interpretation function $\cdot^{\mathcal{I}}$ which maps:

- each individual name a to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$,
- each atomic concept A to a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$,
- each atomic role P to a binary relation $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

The interpretation function is extended to complex concept expressions according to the following rules:

$$\begin{aligned}
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} &= \emptyset \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C_1 \sqcap C_2)^{\mathcal{I}} &= C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}} \\
(C_1 \sqcup C_2)^{\mathcal{I}} &= C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}} \\
(\exists P)^{\mathcal{I}} &= \{d_1 \in \Delta^{\mathcal{I}} \mid \text{ there is } (d_1, d_2) \in P^{\mathcal{I}}\} \\
(\exists P.C)^{\mathcal{I}} &= \{d_1 \in \Delta^{\mathcal{I}} \mid \text{ there is } d_2 \in \Delta^{\mathcal{I}}, \text{ with } (d_1, d_2) \in P^{\mathcal{I}} \text{ and } d_2 \in C^{\mathcal{I}}\} \\
(\forall P.C)^{\mathcal{I}} &= \{d_1 \in \Delta^{\mathcal{I}} \mid \text{ for all } d_2 \in \Delta^{\mathcal{I}}, \text{ if } (d_1, d_2) \in P^{\mathcal{I}} \text{ then } d_2 \in C^{\mathcal{I}}\}
\end{aligned}$$

The model-theoretic semantics of DLs ensures a formal and unambiguous interpretation of knowledge bases.

Model entailment and satisfiability. An interpretation $\mathcal{I}$ is said to be a model of an inclusion axiom on concepts $C_1 \sqsubseteq C_2$, denoted by $\mathcal{I} \vDash C_1 \sqsubseteq C_2$, if $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$. An interpretation $\mathcal{I}$ is a model of a TBox $\mathcal{T}$, denoted by $\mathcal{I} \vDash \mathcal{T}$, if for every axiom α in $\mathcal{T}$, $\mathcal{I} \vDash \alpha$. A TBox $\mathcal{T}$ is incoherent if for each interpretation $\mathcal{I}$ that is a model of $\mathcal{T}$, there is $A \in \mathsf{N_C}$, respectively $P \in \mathsf{N_R}$, such that $A^{\mathcal{I}} = \emptyset$, respectively $P^{\mathcal{I}} = \emptyset$. Otherwise, $\mathcal{T}$ is coherent.

An interpretation $\mathcal{I}$ satisfies an assertion $A(a)$, denoted by $\mathcal{I} \vDash A(a)$, if $a^{\mathcal{I}} \in A^{\mathcal{I}}$. Similarly, $\mathcal{I}$ satisfies an assertion $P(a, b)$, denoted by $\mathcal{I} \vDash P(a, b)$, if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$. An interpretation $\mathcal{I}$ is a model of an ABox $\mathcal{A}$, denoted by $\mathcal{I} \vDash \mathcal{A}$, if for every assertion φ in $\mathcal{A}$, $\mathcal{I} \vDash \varphi$. A model of the knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is an interpretation $\mathcal{I}$ such that $\mathcal{I} \vDash \mathcal{T}$ and $\mathcal{I} \vDash \mathcal{A}$. A knowledge base is consistent if it admits at least one model; otherwise, it is inconsistent.

More expressive DLs

In addition to the concept constructors present in the $\mathcal{ALC}$ fragment, further constructors are used by richer DLs to achieve more expressivity. Table 1.2 summarises some of the widely used constructors, where C is a concept name, and R, R_1 and R_2 are role names.

Table 1.2: Syntax and semantics of some expressive DL concept and role constructors.

Name	**Syntax**	**Semantics**
Atomic role	P	$P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
Inverse role	P^-	$(P^-)^{\mathcal{I}} = \{(d_2, d_1) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (d_1, d_2) \in P^{\mathcal{I}}\}$
Role negation	$\neg R$	$(\neg R)^{\mathcal{I}} = (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \setminus R^{\mathcal{I}}$
Role composition	$R_1 \circ R_2$	$\{(d_1, d_3) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (d_1, d_2) \in R_1^{\mathcal{I}}$ and $(d_2, d_3) \in R_2^{\mathcal{I}}\}$
Role inclusion	$R_1 \sqsubseteq R_2$	$R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$
Transitive role	$trans(R)$	$R^{\mathcal{I}} \circ R^{\mathcal{I}} \subseteq R^{\mathcal{I}}$ (if $R(a, b)$ and $R(b, c)$, then $R(a, c)$)
Functional role	$funct(R)$	If $(d_1, d_2) \in R^{\mathcal{I}}$ and $(d_1, d_3) \in R^{\mathcal{I}}$, then $d_2 = d_3$
Unqualified number restrictions	$\geq n\, R$	$\{d_1 \mid \lvert\{d_2 \mid (d_1, d_2) \in R^{\mathcal{I}}\}\rvert \geq n\}$
	$\leq n\, R$	$\{d_1 \mid \lvert\{d_2 \mid (d_1, d_2) \in R^{\mathcal{I}}\}\rvert \leq n\}$
	$= n\, R$	$\{d_1 \mid \lvert\{d_2 \mid (d_1, d_2) \in R^{\mathcal{I}}\}\rvert = n\}$
Qualified number restrictions	$\geq n\, R.C$	$\{d_1 \mid \lvert\{d_2 \mid (d_1, d_2) \in R^{\mathcal{I}}$ and $d_2 \in C^{\mathcal{I}}\}\rvert \geq n\}$
	$\leq n\, R.C$	$\{d_1 \mid \lvert\{d_2 \mid (d_1, d_2) \in R^{\mathcal{I}}$ and $d_2 \in C^{\mathcal{I}}\}\rvert \leq n\}$
	$= n\, R.C$	$\{d_1 \mid \lvert\{d_2 \mid (d_1, d_2) \in R^{\mathcal{I}}$ and $d_2 \in C^{\mathcal{I}}\}\rvert = n\}$

Expressive DLs extend role modelling capabilities beyond simple atomic roles. The *inverse role* constructor P^- allows expressing relationships in both directions, such as owns^- to denote the relation "is owned by". For instance, if $\mathsf{owns}(\mathsf{Alice}, \mathsf{r}_1)$ means "Alice owns r_1", then $\mathsf{owns}^-(\mathsf{r}_1, \mathsf{Alice})$ expresses "r_1 is owned by Alice". Such an expression is crucial for ontologies where bidirectional relations are needed, such as in $\mathcal{SHOIN}$ (which underlies OWL DL). *Role negation* $\neg R$ enables us to state that two elements are *not* related by a role name, which is used in $\mathcal{ALCIO}$ and more expressive DLs. *Role composition* $R_1 \circ R_2$ allows chaining roles to define relationships over multiple steps, a key feature in $\mathcal{SROIQ}$, the basis of OWL 2-DL.

Other role-related constructors include *role inclusion axioms* $R_1 \sqsubseteq R_2$, allowing the specification of hierarchical relationships between roles, and *transitivity* $trans(R)$, which ensures that

if an individual is related to a second through R, and the second to a third, then the first is related to the third. These constructs are essential in DLs such as $\mathcal{SHOIQ}$, where they enable reasoning over complex relationships. *Functional roles* $funct(R)$ restrict a role to have at most one successor, commonly used in identity and reference constraints.

Regarding concept constructors, *number restrictions* introduce expressive constraints on the number of role successors an individual can have. *Unqualified number restrictions* (e.g. $\geq nR$, $\leq nR$) specify a minimum or maximum number of related individuals without restricting their type, while *qualified number restrictions* (e.g. $\geq nR.C$) impose constraints based on the concept of successor individuals. Number restrictions are key in expressive DLs like $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$, which allows cardinality-based reasoning.

These additional constructors enhance the expressiveness of DLs, enabling complex domain modelling and supporting automated reasoning tasks in ontology-based applications, including semantic web technologies, knowledge graphs, and ontology-based data access (OBDA).

Reasoning services. DL Knowledge Bases support a wide range of reasoning services. The main ones are presented below:

- **Concept satisfiability:** A concept C is satisfiable w.r.t. a TBox $\mathcal{T}$ if there exists an interpretation $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{T}$ and $C^{\mathcal{I}} \neq \emptyset$. This checks whether the concept can have instances without leading to a contradiction.

- **Concept subsumption:** A concept C_1 is subsumed by a concept C_2 w.r.t. a TBox $\mathcal{T}$, written $\mathcal{T} \models C_1 \sqsubseteq C_2$, if in every model $\mathcal{I}$ of $\mathcal{T}$, it holds that $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$.

- **Instance checking:** Given an individual 'a' and a concept C, instance checking with respect to a knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ verifies whether $\mathcal{K} \models C(a)$, i.e., whether 'a' is an instance of C in all models of $\mathcal{K}$. Similarly, given two individuals 'a_1' and 'a_2' and a role R, instance checking with respect to $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ verifies whether $\mathcal{K} \models R(a_1, a_2)$, i.e., whether the pair '(a_1, a_2)' is an instance of R in all models of $\mathcal{K}$.

- **Consistency checking:** A knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is consistent if there exists at least one interpretation $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \mathcal{A}$; otherwise, it is inconsistent. This ensures that the asserted facts do not contradict the axioms.

One particular aspect of DLs compared to relational databases (SQL DBs) relies on the adoption of the open-world assumption; this means that lack of knowledge does not imply falsity, and one cannot assume that the knowledge in the KB is complete. Note that in this thesis, we are making the *unique name assumption*, which means that different individual names refer to different individuals.

Query answering

In addition to the reasoning problems mentioned above, query answering is another crucial reasoning task in description logics. It enables the extraction of implicit information from a knowledge base with respect to a formal query language, through paradigms such as ontology-based data access (OBDA) [Xiao *et al.* 2018], and ontology-mediated query answering (OMQA) [Bienvenu *et al.* 2019]. The commonly used language for queries in these paradigms is that of conjunctive queries (CQs), and slight extensions such as unions of conjunctive queries (UCQs) and positive existential queries.

Positive existential queries $q(x_1, \ldots, x_n)$ are first-order formulas $\phi(x_1, \ldots, x_n)$ built from atomic predicates and using conjunction, disjunction, and existential quantification on variables. The syntax of such queries can be formally defined as follows:

$$t ::= y_i \mid a_i \quad \text{(terms)} \qquad\qquad \phi ::= A_k(t) \mid P_k(t_1, t_2) \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \exists y_i\, \phi$$

where y_i are variables and a_i are individual names, A_k are atomic concepts, and P_k are atomic roles.

A conjunctive query (CQ) is a special case in which the formula ϕ is a conjunction of atoms and quantification is only existential. That is, a CQ takes the form $\exists \vec{y}.\ \bigwedge_i P_k(t_i^1, t_i^2)$, where some of the terms may be variables and others constants. A Boolean Conjunctive Query (BCQ) (or ground query) contains no free variables and thus evaluates to either true or false in a given model.

Answering such queries amounts to identifying the tuples of individuals that make the query formula true in every model of the knowledge base. Thus, query answering is defined with respect to the class of interpretations satisfying the knowledge base, and is inherently grounded in the model-theoretic semantics of DLs.

Query rewriting Relational database systems (SQL DBs) are long-standing approaches to storing and managing data. The paradigms of OBDA and OMQA take advantage of the advance of database systems to store factual data (ABox assertions) and provide access to it using the well-established SQL querying systems. Conjunctive queries, as given above, can be seen as the select-from-where fragment of SQL, written in logic. Then, the query rewriting process can be used to integrate the TBox into the query. In simple terms, query rewriting consists of reducing the problem of answering a CQ q on a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$ into answering a query $q_{\mathcal{T}}$ by a database system that stores $\mathcal{A}$ as data. This allows us to extract the implicit knowledge encoded by the TBox into the query, before submitting the query to the ABox.

It is essential to note that query rewritings are usually guaranteed to exist only for inexpressive DLs. This is the main reason for the introduction and design of the lightweight DLs family DL-Lite [Calvanese *et al.* 2007, Artale *et al.* 2009, Baader *et al.* 2017]. In addition, this is one of the main reasons why this thesis puts a special focus on the DL-Lite family for inconsistency management.

Complexity measures

The computational complexity of reasoning problems in DLs can be regarded in two different ways, depending on the considered input of the problem. For decisions problems, such as instance checking and consistency checking, the input parameters are the size of the TBox $|\mathcal{T}|$ and of the ABox $|\mathcal{A}|$. Hence, the size of the knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is given in terms of the sizes of both $|\mathcal{T}|$ and $|\mathcal{A}|$ [Calvanese *et al.* 2007]. For query answering, another parameter to consider is the size of the query $|q|$. However, the size of a query is often negligible compared to the size of the TBox and of the ABox. Therefore, the standard database assumption that the size of queries is always bounded by some constant is adopted.

- *Combined complexity* considers the whole knowledge base as input for the problem and thus considers both the size of the TBox $|\mathcal{T}|$ and of the ABox $|\mathcal{A}|$.
- *Data complexity* takes as input only the size of the ABox. It considers the size of the TBox as fixed or constant. This is usually the case when the size of the TBox is negligible compared to the size of the ABox.

Note that in some work efforts [Baader *et al.* 2007], a difference is made between knowledge base complexity, where the size of the knowledge base $|\mathcal{K}|$ is regarded as input, and combined complexity, where both the sizes of the knowledge base $|\mathcal{K}|$ and of the query $|q|$ are regarded as input [Bienvenu & Bourgaux 2017]. Nevertheless, since the work presented in this thesis aligns with the OBDA and OMQA paradigms, where the main interest is data complexity, the mentioned difference is not relevant.

For the $\mathcal{ALC}$ language, the problem of consistency checking is NP-complete in data complexity. Consistency checking of knowledge bases of the more expressive DLs $\mathcal{SHOIN}$ and $\mathcal{SROIQ}$ is NP-hard in data complexity. These and other complexity results of some of the well-known DLs can be found at http://www.cs.man.ac.uk/~ezolin/dl/ [3].

1.2 DL-Lite Knowledge Bases

The lightweight family of description logics DL-Lite [Calvanese *et al.* 2007] are decidable knowledge representation languages based on first-order predicate logic. These language fragments have been designed to capture conceptual modelling constructs in applications with large datasets and where query answering is the most important reasoning task.

When presented with a query, the idea of query reformulation (or rewriting) consists in encoding the conceptual knowledge of a TBox into the query before posing it to the ABox. This separation allows taking advantage of relational database systems to store the ABox, while ensuring that reasoning on the TBox is reduced to answering SQL queries. In this section, a detailed introduction of the DL-Lite$_\mathcal{R}$ language is provided, along with the semantics of the language and the notions of inconsistency and FOL reducibility, before giving the algorithms that perform the basic reasoning tasks in DL-Lite$_\mathcal{R}$.

1.2.1 Syntax

There are several variants of DL-Lite [Artale *et al.* 2009] such as DL-Lite$_{core}$, DL-Lite$_\mathcal{R}$, DL-Lite$_\mathcal{F}$, DL-Lite$_\mathcal{A}$ and DL-Lite$_{Horn}$. In this work, we focus on DL-Lite$_\mathcal{R}$, also known as DL-Lite$^H_{core}$, the dialect that provides the logical underpinning for the OWL 2-QL profile [Motik *et al.* 2012] which is devoted to query answering. This focus is mainly motivated by the efficiency of query answering, since it can be reduced to standard database query evaluation using query rewriting [Kontchakov *et al.* 2010].

The DL-Lite$_\mathcal{R}$ language is built upon a vocabulary consisting of three sets, denoted $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_I}$, which are pairwise disjoint and countably infinite, and which contain concept names, role names, and individual names, respectively. Let $A \in \mathsf{N_C}$, $P \in \mathsf{N_R}$ and $P^- \in \mathsf{N_R}$ is the inverse of P. Let the negation symbol "$\neg$" encode complement sets (of concepts and of roles). Let the existential quantifier symbol "$\exists$" denote existential restriction. The DL-Lite$_\mathcal{R}$ language is defined according to the following rules:

$$R := P \mid P^- \qquad E := R \mid \neg R$$
$$B := A \mid \exists R \qquad C := B \mid \neg B$$

where R denotes a basic role, and E denotes a complex role. B stands for a basic concept and C is a complex concept.

[3] Last access: September 17, 2025.

Example 1.7. *Let us now provide full sets of concept names, role names, and individuals to complete our data access management example:*

- $\mathsf{N_C} = \{\mathsf{User}, \mathsf{Secondee}, \mathsf{FullResearcher}, \mathsf{Resource}, \mathsf{Confidential}\}$
  Concept names representing classes of individuals (with their obvious meaning):
  - User*: a general user in the system*
  - Secondee*: an external visiting collaborator*
  - FullResearcher*: a permanent internal researcher*
  - Resource*: any digital or data resource*
  - Confidential*: a resource with restricted access*
- $\mathsf{N_R} = \{\mathsf{hasAccess}, \mathsf{hasConfidAccess}, \mathsf{hasOpenAccess}, \mathsf{owns}\}$
  Role names describing binary relations between individuals:
  - hasAccess*: general access to a resource*
  - hasConfidAccess*: access to a confidential resource*
  - hasOpenAccess*: access to a non-confidential (open) resource*
  - owns*: ownership of a resource*
- $\mathsf{N_I} = \{\mathsf{r}_1, \mathsf{r}_2, \mathsf{Duane}, \mathsf{Ben}\}$
  Individual names representing specific entities:
  - r_1, r_2*: two specific resources*
  - Duane, Ben*: two specific users*

A DL-Lite$_\mathcal{R}$TBox is a finite set of inclusion axioms on concepts and roles as follows:

$$B \sqsubseteq C \qquad\qquad R \sqsubseteq E$$

A concept inclusion of the form $\exists P \sqsubseteq B$ requires the domain of a role P to be included in a concept B and is referred to by a "domain restriction". Similarly, a concept inclusion of the form $\exists P^- \sqsubseteq B$ requires the range of a role P to be included in a concept B and is called a "range restriction".

Inclusions on concepts and roles of the types $B_1 \sqsubseteq B_2$ and $R_1 \sqsubseteq R_2$, respectively, are called "*positive inclusion axioms*". Inclusions on concepts and roles with the negation symbol "$\neg$" in the right-hand side of the inclusion, such as $B_1 \sqsubseteq \neg B_2$ and $R_1 \sqsubseteq \neg R_2$, respectively, are called "*negative inclusion axioms*" (or disjointness axioms). In DL-Lite$_\mathcal{R}$, negative axioms make it possible to specify the disjointness of at most two concepts or two roles [Calvanese *et al.* 2010b]. The set of negative inclusion axioms of a TBox $\mathcal{T}$ is denoted $\mathcal{T}_n$ and the set of its positive inclusion axioms is denoted $\mathcal{T}_p$.

Example 1.8. *Let us describe a DL-Lite$_\mathcal{R}$ TBox based on the concept names and role*

names provided in Example 1.7:

$$\mathcal{T} = \left\{ \begin{array}{ll} 1.\ \mathsf{Secondee} \sqsubseteq \mathsf{User} & 7.\ \exists \mathsf{hasAccess}^- \sqsubseteq \mathsf{Resource} \\ 2.\ \mathsf{FullResearcher} \sqsubseteq \mathsf{User} & 8.\ \exists \mathsf{hasConfidAccess}^- \sqsubseteq \mathsf{Confidential} \\ 3.\ \mathsf{Confidential} \sqsubseteq \mathsf{Resource} & 9.\ \mathsf{hasConfidAccess} \sqsubseteq \neg\, \mathsf{hasOpenAccess} \\ 4.\ \mathsf{Secondee} \sqsubseteq \neg\, \mathsf{FullResearcher} & 10.\ \mathsf{owns} \sqsubseteq \mathsf{hasAccess} \\ 5.\ \exists \mathsf{hasAccess} \sqsubseteq \mathsf{User} & 11.\ \mathsf{hasOpenAccess} \sqsubseteq \mathsf{hasAccess} \\ 6.\ \exists \mathsf{owns} \sqsubseteq \mathsf{FullResearcher} & 12.\ \mathsf{hasConfidAccess} \sqsubseteq \mathsf{hasAccess} \end{array} \right\}$$

- *Axioms 1 and 2 state that every* Secondee *is* User *and every* FullResearcher *is also a* User *in the system, respectively.*
- *Axiom 3 states that every* Confidential *entity is also a* Resource.
- *Axiom 4 expresses that* Secondee *and* FullResearcher *are disjoint concepts: no individual can simultaneously belong to both categories.*
- *Axioms 5 and 6 specify that any individual who has access to some entity is also a* User *and that any individual that owns an entity is a* FullResearcher, *respectively.*
- *Axioms 7 and 8 specify that if an entity is accessed by someone, then that entity is a resource and that if an entity is accessed through confidential access, then it is a confidential resource, respectively.*
- *Axiom 9 expresses that* hasConfidAccess *and* hasOpenAccess *are disjoint roles: no individual can hold both types of access to the same object.*
- *Axiom 10 specifies that ownership implies access, if someone owns something, they also have access to it.*
- *Axioms 11 and 12 describe that open access and confidential access are forms of access, any open access or confidential access relationship implies an access relationship.*

A DL-Lite$_\mathcal{R}$ ABox is a finite set of membership assertions (or ground facts) of individuals on atomic concepts and atomic roles of the form:

$$A(a) \qquad P(a, b) \qquad \text{where } a \text{ and } b \text{ are individuals.}$$

Definition 1.1 (Knowledge base)**.** *A DL-Lite$_\mathcal{R}$ knowledge base (KB) is a pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ where $\mathcal{T}$ is a TBox and $\mathcal{A}$ is an ABox.*

Example 1.9. *Let us now complete our DL-Lite$_\mathcal{R}$ knowledge base by providing the following ABox $\mathcal{A}_1$:*

$$\mathcal{A}_1 = \left\{ \begin{array}{l} \mathsf{FullResearcher}(\mathsf{Ben}) \\ \mathsf{Secondee}(\mathsf{Duane}) \\ \mathsf{Confidential}(\mathsf{r}_1) \\ \mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1) \\ \mathsf{hasOpenAccess}(\mathsf{Duane}, \mathsf{r}_2) \\ \mathsf{owns}(\mathsf{Ben}, \mathsf{r}_1) \end{array} \right\}$$

The three first assertions are concept assertions stating individuals membership to concepts: Ben *is a* FullResearcher, Duane *is a secondee, and* r_1 *is a* Confidential *resource. The remaining assertions are role assertions stating that* Ben *has confidential access to* r_1, Duane *has open access to* r_2, *and* Ben *owns* r_1.

1.2.2 Semantics

Like in DLs, DL-Lite$_\mathcal{R}$ semantics are also standard set-theoretic semantics, given in terms of interpretations. For clarity reasons, we recall them in details here. An interpretation is defined as a tuple $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$, where $\Delta^\mathcal{I} \neq \emptyset$ is a non-empty domain and $\cdot^\mathcal{I}$ is an interpretation function. The function $\cdot^\mathcal{I}$ associates concept names $A \in \mathsf{N_C}$ to $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, role names $P \in \mathsf{N_R}$ to $P^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, and individual names $a \in \mathsf{N_I}$ to $a^\mathcal{I} \in \Delta^\mathcal{I}$.

In addition, the interpretation function is extended to interpret complex concepts and roles of DL-Lite$_\mathcal{R}$ as follows:

$$
\begin{aligned}
&(P^-)^\mathcal{I} = \{(y, x) \in \Delta^\mathcal{I} \times \Delta^\mathcal{I} \mid (x, y) \in P^\mathcal{I}\};\\
&(\exists R)^\mathcal{I} = \{x \in \Delta^\mathcal{I} \mid \text{there is } y \in \Delta^\mathcal{I} \text{ such that } (x, y) \in R^\mathcal{I}\};\\
&(\neg B)^\mathcal{I} = \Delta^\mathcal{I} \setminus B^\mathcal{I};\\
&(\neg R)^\mathcal{I} = \Delta^\mathcal{I} \times \Delta^\mathcal{I} \setminus R^\mathcal{I}.
\end{aligned}
$$

An interpretation $\mathcal{I}$ is a model of an inclusion axiom on concepts $B \sqsubseteq C$, denoted by $\mathcal{I} \vDash B \sqsubseteq C$, if $B^\mathcal{I} \subseteq C^\mathcal{I}$. Similarly, $\mathcal{I}$ is a model of an inclusion axiom on roles $R \sqsubseteq E$, denoted by $\mathcal{I} \vDash R \sqsubseteq E$, if $R^\mathcal{I} \subseteq E^\mathcal{I}$. An interpretation $\mathcal{I}$ is a model of a TBox $\mathcal{T}$, denoted by $\mathcal{I} \vDash \mathcal{T}$, if for every axiom α in $\mathcal{T}$, we have $\mathcal{I} \vDash \alpha$.

An interpretation $\mathcal{I}$ satisfies an assertion $A(a)$, denoted by $\mathcal{I} \vDash A(a)$, if $a^\mathcal{I} \in A^\mathcal{I}$. Similarly, $\mathcal{I}$ satisfies an assertion $P(a, b)$, denoted by $\mathcal{I} \vDash P(a, b)$, if $(a^\mathcal{I}, b^\mathcal{I}) \in P^\mathcal{I}$. An interpretation $\mathcal{I}$ is a model of an ABox $\mathcal{A}$, denoted by $\mathcal{I} \vDash \mathcal{A}$, if for every assertion φ in $\mathcal{A}$, $\mathcal{I} \vDash \varphi$. A model of the knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is an interpretation $\mathcal{I}$ such that $\mathcal{I} \vDash \mathcal{T}$ and $\mathcal{I} \vDash \mathcal{A}$. A DL-Lite$_\mathcal{R}$ knowledge base is consistent if it admits at least one model; otherwise, it is inconsistent [Calvanese *et al.* 2007].

Definition 1.2 (Entailment [Calvanese *et al.* 2007])**.** *A DL-Lite$_\mathcal{R}$ knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ logically implies an assertion φ, written $\mathcal{K} \vDash \varphi$, if all models of $\mathcal{K}$ are also models of φ. Similarly, a TBox $\mathcal{T}$ logically implies an axiom α, written $\mathcal{T} \vDash \alpha$, if all models of $\mathcal{T}$ are also models of α.*

Example 1.10. *Let us now continue our example by considering the knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}_1 \rangle$ where $\mathcal{T}$ is the TBox from Example 1.8 and $\mathcal{A}_1$ is the ABox of Example 1.9.*

Assume the domain $\Delta^{\mathcal{I}} = \{\mathsf{Ben}, \mathsf{Duane}, \mathsf{r}_1, \mathsf{r}_2\}$. *Consider the following interpretation:*

$$
\begin{aligned}
(\mathsf{User})^{\mathcal{I}} &= \{\mathsf{Ben}, \mathsf{Duane}\} \\
(\mathsf{Secondee})^{\mathcal{I}} &= \{\mathsf{Duane}\} \\
(\mathsf{FullResearcher})^{\mathcal{I}} &= \{\mathsf{Ben}\} \\
(\mathsf{Resource})^{\mathcal{I}} &= \{\mathsf{r}_1, \mathsf{r}_2\} \\
(\mathsf{Confidential})^{\mathcal{I}} &= \{\mathsf{r}_1\} \\
(\mathsf{hasAccess})^{\mathcal{I}} &= \{(\mathsf{Ben}, \mathsf{r}_1), (\mathsf{Duane}, \mathsf{r}_2)\} \\
(\mathsf{hasConfidAccess})^{\mathcal{I}} &= \{(\mathsf{Ben}, \mathsf{r}_1)\} \\
(\mathsf{hasOpenAccess})^{\mathcal{I}} &= \{(\mathsf{Duane}, \mathsf{r}_2)\} \\
(\mathsf{owns})^{\mathcal{I}} &= \{(\mathsf{Ben}, \mathsf{r}_1)\}
\end{aligned}
$$

One can check that the interpretation $\mathcal{I}$ *is a model of the knowledge base* $\mathcal{K}$.

1.2.3 Reasoning tasks

The main reasoning problems for description logics were also studied in the DL-Lite fragment, namely, concept satisfiability, concept subsumption, knowledge base satisfiability, instance checking, and query answering. In this thesis, a main focus is given to reasoning with inconsistent knowledge bases. Hence, we start by presenting how **Knowledge base satisfiability** is checked in the DL-Lite$_{\mathcal{R}}$ language. We also provide some insight on how query answering is performed in the same language, as it is used in exhibiting the minimal subsets that are responsible for the inconsistency.

Knowledge base satisfiability

The negative inclusion axioms in a description logics TBox allow defining complement sets of concepts and roles. Obviously, a knowledge base where the TBox contains only positive inclusion axioms is consistent. However, a knowledge base in which the TBox contains negative axioms might be inconsistent.

In description logics, two types of logical contradiction (or inconsistency) can be distinguished. The first is often referred to as incoherence and concerns the satisfiability of concepts and roles. More formally, a TBox $\mathcal{T}$ is incoherent if for each interpretation $\mathcal{I}$ that is a model of $\mathcal{T}$, there is $A \in \mathsf{N_C}$, respectively $P \in \mathsf{N_R}$, such that $A^{\mathcal{I}} = \emptyset$, respectively $P^{\mathcal{I}} = \emptyset$. Otherwise, $\mathcal{T}$ is coherent.

The second type is called inconsistency and is defined by the absence of a model for the knowledge base [Calvanese *et al.* 2007]. In the context of Ontology-Based Data Access (OBDA), inconsistency is studied under the assumption that the TBox was properly debugged, and hence it can be safely assumed to be coherent [Lembo *et al.* 2010, Bienvenu & Bourgaux 2017, Baget *et al.* 2016]. The semantics studied or presented in this thesis follow this reasonable assumption and focus on ABox inconsistency.

Example 1.11. *Consider now a new ABox* $\mathcal{A}_2$ *that is given by:*

$$\mathcal{A}_2 = \left\{ \begin{array}{l} \mathsf{FullResearcher}(\mathsf{Ben}) \\ \mathsf{Secondee}(\mathsf{Duane}) \\ \mathsf{Confidential}(\mathsf{r}_1) \\ \mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1) \\ \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1) \\ \mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2) \end{array} \right.$$

Using the axiom $\exists\mathsf{owns} \sqsubseteq \mathsf{FullResearcher}$*, one can infer* $\mathsf{FullResearcher}(\mathsf{Duane})$ *from the assertion* $\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$*. This means that* $\mathcal{A}_2$ *infers both* $\mathsf{Secondee}(\mathsf{Duane})$ *and* $\mathsf{FullResearcher}(\mathsf{Duane})$*. However, the axiom* $\mathsf{Secondee} \sqsubseteq \neg\,\mathsf{FullResearcher}$ *states that* Secondee *and* FullResearcher *must be disjoint. As a result, our knowledge base does not admit any model, and hence it is inconsistent.*
Furthermore, one can use the negative axiom $\mathsf{hasConfidAccess} \sqsubseteq \neg\,\mathsf{hasOpenAccess}$ *to show the disjointness of the assertions* $\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)$ *and* $\mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)$*.*

The knowledge base satisfiability problem is a decision problem where the input is composed of the TBox and the ABox, and the output is a binary response indicating whether the knowledge base is consistent or not. The semantic foundation that supports this problem is the notion of a canonical interpretation, which is a (possibly infinite) model that is built directly from the TBox and the ABox by systematically applying the implications of the axioms. It is designed to represent all the logical consequences of the knowledge base and often serves as a reference model for reasoning [Calvanese *et al.* 2007].

The existence of a canonical model for a DL-Lite$_\mathcal{R}$ knowledge base has been proven to be equivalent to the existence of an interpretation that satisfies both the negative closure of the TBox and the so-called minimal interpretation of the ABox [Calvanese *et al.* 2007]. We proceed by introducing these two notions, before providing the actual method for consistency checking.

The negative closure of a TBox, denoted $cln(\mathcal{T})$, propagates negative inclusion axioms using both positive and negative axioms to a fixed point, resulting in a set that contains all negative inclusion axioms that are logically implied by $\mathcal{T}$.

Definition 1.3 (TBox Negative closure)**.** *Let* $\mathcal{T}$ *be a DL-Lite*$_\mathcal{R}$ *TBox. The negative inclusion closure of* $\mathcal{T}$*, denoted by* $cln(\mathcal{T})$*, is the TBox defined inductively using the following rules:*

1. *All negative inclusion axioms in* $\mathcal{T}$ *are also in* $cln(\mathcal{T})$*.*
2. *If* $B_1 \sqsubseteq B_2$ *is in* $\mathcal{T}$ *and* $B_2 \sqsubseteq \neg\, B_3$ *or* $B_3 \sqsubseteq \neg\, B_2$ *is in* $cln(\mathcal{T})$*, then* $B_1 \sqsubseteq \neg\, B_3$ *is also in* $cln(\mathcal{T})$*.*
3. *If* $R_1 \sqsubseteq R_2$ *is in* $\mathcal{T}$ *and* $\exists R_2 \sqsubseteq \neg\, B$ *or* $B \sqsubseteq \neg\, \exists R_2$ *is in* $cln(\mathcal{T})$*, then* $\exists R_1 \sqsubseteq \neg\, B$ *is also in* $cln(\mathcal{T})$*.*
4. *If* $R_1 \sqsubseteq R_2^-$ *is in* $\mathcal{T}$ *and* $\exists R_2^- \sqsubseteq \neg\, B$ *or* $B \sqsubseteq \neg\, \exists R_2^-$ *is in* $cln(\mathcal{T})$*, then* $\exists R_1^- \sqsubseteq \neg\, B$ *is also in* $cln(\mathcal{T})$*.*
5. *If* $R_1 \sqsubseteq R_2$ *is in* $\mathcal{T}$ *and* $R_2 \sqsubseteq \neg\, R_3$ *or* $R_3 \sqsubseteq \neg\, R_2$ *is in* $cln(\mathcal{T})$*, then* $R_1 \sqsubseteq \neg\, R_3$ *is also in* $cln(\mathcal{T})$*.*

6. *If one of the axioms* $\exists R \sqsubseteq \neg\, \exists R$, $\exists R^- \sqsubseteq \neg\, \exists R^-$, *or* $R \sqsubseteq \neg\, R$ *is in* $cln(\mathcal{T})$, *then all three such axioms are also in* $cln(\mathcal{T})$.

Example 1.12. *Let us compute the negative closure of the TBox* $\mathcal{T}$ *from Example 1.8.*

- *The two negative inclusion axioms* $\mathsf{Secondee} \sqsubseteq \neg\, \mathsf{FullResearcher}$ *and* $\mathsf{hasConfidAccess} \sqsubseteq \neg\, \mathsf{hasOpenAccess}$ *are both in* $cln(\mathcal{T})$.
- *According to rule 2,* $\exists\mathsf{owns} \sqsubseteq \mathsf{FullResearcher}$ *and* $\mathsf{Secondee} \sqsubseteq \neg\, \mathsf{FullResearcher}$ *gives* $\exists\mathsf{owns} \sqsubseteq \neg\, \mathsf{Secondee}$ *in* $cln(\mathcal{T})$.

Therefore, the negative closure of the TBox $\mathcal{T}$ *is given by:*

$$cln(\mathcal{T}) = \left\{ \begin{array}{cc} \mathsf{Secondee} \sqsubseteq \neg\, \mathsf{FullResearcher}, & \mathsf{hasConfidAccess} \sqsubseteq \neg\, \mathsf{hasOpenAccess}, \\ \exists\mathsf{owns} \sqsubseteq \neg\, \mathsf{Secondee} & \end{array} \right\}$$

A minimal interpretation is a database-like representation of an ABox, which is mainly based on the constants and the membership assertions that are present within the ABox [Calvanese *et al.* 2007].

Definition 1.4 (Minimal interpretation)**.** *Let* $\mathcal{A}$ *be a DL-Lite*$_\mathcal{R}$ *ABox. A minimal interpretation is denoted by* $db(\mathcal{A}) = \langle \Delta^{db(\mathcal{A})}, \cdot^{db(\mathcal{A})} \rangle$ *such that:*

- $\Delta^{db(\mathcal{A})}$ *is the non-empty set consisting of all the constants occurring in* $\mathcal{A}$.
- $a^{db(\mathcal{A})} = a$, *for each constant* a.
- $A^{db(\mathcal{A})} = \{a \mid A(a) \in \mathcal{A}$ *for each atomic concept* $A\}$.
- $P^{db(\mathcal{A})} = \{(a_1, a_2) \mid P(a_1, a_2) \in \mathcal{A}$ *for each atomic role* $P\}$.

In the case of a consistent knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, the interpretation $db(\mathcal{A})$ is a minimal model of the ABox $\mathcal{A}$ [Calvanese *et al.* 2007]. The main aim of defining such an interpretation is to allow storing the ABox in a relational database management system, and allow as a consequence the construction of DL queries that can also be represented in SQL.

The notions of negative closure of a TBox $cln(\mathcal{T})$ and minimal interpretation $db(\mathcal{A})$ are used to check the satisfiability of a knowledge base. Indeed, the authors in [Calvanese *et al.* 2007] have established that a DL-Lite$_\mathcal{R}$ knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is satisfiable if and only if the interpretation $db(\mathcal{A})$ is a model of $\langle cln(\mathcal{T}), \mathcal{A} \rangle$.

Knowledge base satisfiability for a given description logics fragment is said to be FOL-reducible if for every TBox $\mathcal{T}$ expressed in that logic, there exists a Boolean query q over the alphabet of $\mathcal{T}$, such that for every non-empty ABox $\mathcal{A}$, $\langle \mathcal{T}, \mathcal{A} \rangle$ is satisfiable if and only if q evaluates to false in $db(\mathcal{A})$ (the minimal interpretation of $\mathcal{A}$). The knowledge base satisfiability problem has been shown to be FOL-reducible for DL-Lite$_\mathcal{R}$ [Calvanese *et al.* 2007]. As a result, consistency checking is in AC^0 with respect to data complexity and in PTime with respect to combined complexity for DL-Lite$_\mathcal{R}$ knowledge bases [Calvanese *et al.* 2007, Artale *et al.* 2009].

Therefore, a possible method to check the satisfiability of a knowledge base proceeds by evaluating a Boolean FOL query built from the set of all negative inclusion axioms entailed by the TBox, i.e., $cln(\mathcal{T})$, over the database-like interpretation $db(\mathcal{A})$.

Before introducing the method of checking knowledge base satisfiability and defining the minimal subsets that cause inconsistency, let us recall some notions that support query answering

from DL-Lite knowledge bases. In the following section, we specify the query language to be used, the notions of entailment, query answer, and query reformulation.

Query reformulation

Another important property of DL-Lite$_\mathcal{R}$ knowledge bases is query reformulation (or query rewriting). The idea is that given a query q and a TBox $\mathcal{T}$, one can efficiently produce a query $q_\mathcal{T}$ that materialises the possible TBox axiom conclusions. Then, such a query is evaluated on the database-like interpretation $db(\mathcal{A})$.

A first-order logic query $q(\vec{x})$ over a DL-Lite$_\mathcal{R}$ knowledge base is a first-order logic formula built from the knowledge base's vocabulary (atomic concepts and roles) and a set of variables, where $\vec{x}$ denotes the tuple of distinguished (answer) variables. FOL conjunctive queries (CQs) take the form:

$$q(\vec{x}) = \exists \vec{y} \psi(\vec{x}, \vec{y})$$

where ψ is a conjunction of atoms of the forms $A(t)$ or $P(t, t')$, and t, t' are individuals or variables from $\vec{x} \cup \vec{y}$. As in FOL syntax, $A(t)$ and $P(t, t')$ are called atoms. The free variables $\vec{x}$ are called distinguished or answer variables. A Boolean conjunctive query (BCQ) is a query that does not involve any free variables. A Boolean query can be obtained from a query $q(\vec{x})$ with the free variables $\vec{x} = (x_1, \ldots, x_k)$ by replacing the free variables $\vec{x}$ with the constants (individuals) $\vec{a} = (a_1, \ldots, a_k)$. Such query is denoted by $q(\vec{a})$.

Example 1.13. *Consider the following query that retrieves a user that has access to something:*

$$q(x) = \exists y.\mathsf{User}(x) \wedge \mathsf{hasAccess}(x, y)$$

$q(x)$ is a non-Boolean FOL conjunctive query over our knowledge base from Example 1.10.

To formally reason about the evaluation of such queries, we next introduce the notion of query entailment, which characterises when a Boolean query holds in a knowledge base.

Definition 1.5 (Boolean Query Entailment)**.** *A Boolean query q is said to be satisfied by an interpretation $\mathcal{I}$, denoted $\mathcal{I} \models q$ if and only if q evaluates to true in $\mathcal{I}$.*
A Boolean query q is entailed by a knowledge base $\mathcal{K}$, denoted $\mathcal{K} \vDash q$, if and only if q is satisfied in all models of $\mathcal{K}$.

Having defined when a Boolean query is entailed, we now turn to the more general case of non-Boolean conjunctive queries by defining the answer of a query.

Definition 1.6 (Answer)**.** *A tuple of individuals $\vec{a}$ is an answer to a query $q(\vec{x})$ in an interpretation $\mathcal{I}$, denoted $\mathcal{I} \models q(\vec{a})$ if and only if $q(\vec{a})$ is satisfied in $\mathcal{I}$. The set of answers for $q(\vec{x})$ in $\mathcal{I}$ is denoted $ans(q, \mathcal{I})$.*

The notion of certain answers allows for reasoning only on the query answers that hold in every model of a knowledge base. Formally:

Definition 1.7 (Certain Answer)**.** *A tuple of individuals $\vec{a}$ is an answer to a query $q(\vec{x})$ over a knowledge base $\mathcal{K}$, denoted $\mathcal{K} \vDash q(\vec{a})$ if and only if $\vec{a}$ is an answer to $q(\vec{x})$ in every model of $\mathcal{K}$. The set of certain answers for $q(\vec{x})$ over $\mathcal{K}$ is denoted $cert(q, \mathcal{K})$.*

Now, based on the previous definitions of a minimal interpretation and certain answers, the idea of query reformulation (or re-writing) can be introduced. For a given TBox, it consists in producing an equivalent query that can be evaluated directly over the minimal interpretation of an ABox. Therefore, query answering in DL-Lite knowledge bases separates conceptual and factual knowledge by materialising the possible consequences of the TBox into the query.

Definition 1.8 (FOL Reformulation)**.** *A reformulation of an FOL query q with respect to a TBox $\mathcal{T}$ is an FOL query $q_{\mathcal{T}}$ such that $cert(q, \langle \mathcal{T}, \mathcal{A} \rangle) = ans(q_{\mathcal{T}}, db(\mathcal{A}))$.*

The result of reformulating a conjunctive query can be either a conjunctive query or a union of conjunctive queries, which will be referred to as UCQ in the following.

Several algorithms have been proposed in the literature for query reformulation in description logics [Pérez-Urbina *et al.* 2009, Rosati & Almatelli 2010, Calvanese *et al.* 2011, Chortaras *et al.* 2011, Venetis *et al.* 2012, Rodriguez-Muro *et al.* 2013, Gottlob *et al.* 2014]. The main idea here is to apply the positive inclusion axioms of the TBox to each atom of the query, in order to also query for the possible consequences of those atoms, and then unify atoms when possible. We provide the PerfectRef algorithm, which is specifically tailored for DL-Lite$_{\mathcal{R}}$ knowledge bases [Calvanese *et al.* 2007].

The algorithm takes each atom in the query and checks for all positive inclusions in $\mathcal{T}_p$ if they can be applied to that atom. In a query q, an argument of an atom is said to be *bound* if it is either a distinguished (an answer) variable, or a shared variable, i.e., a variable that occurs at least twice in the query body, or a constant. Non-distinguished non-shared variables are denoted by _. The applicability of an axiom α to an atom g is given as follows:

- α is applicable to an atom $A(x)$ if α has A on its right side.
- α is applicable to an atom $P(x_1, x_2)$ if:
  - (i) $x_2 = _$ and the right side of α is $\exists P$; or,
  - (ii) $x_1 = _$ and the right side of α is $\exists P^-$; or,
  - (iii) α is a role inclusion axiom and its right side is either P or P^-.

The result of applying α on some atom g is an atom given by $gr(g, \alpha)$ in the algorithm. Formally:

- if $g = A(x)$ and $\alpha = A_1 \sqsubseteq A$, then $gr(g, \alpha) = A_1(x)$
- if $g = A(x)$ and $\alpha = \exists P \sqsubseteq A$, then $gr(g, \alpha) = P(x, _)$
- if $g = A(x)$ and $\alpha = \exists P^- \sqsubseteq A$, then $gr(g, \alpha) = P(_, x)$
- if $g = P(x, _)$ and $\alpha = A \sqsubseteq \exists P$, then $gr(g, \alpha) = A(x)$
- if $g = P(x, _)$ and $\alpha = \exists P_1 \sqsubseteq \exists P$, then $gr(g, \alpha) = P_1(x, _)$
- if $g = P(x, _)$ and $\alpha = \exists P_1^- \sqsubseteq \exists P$, then $gr(g, \alpha) = P_1(_, x)$
- if $g = P(_, x)$ and $\alpha = A \sqsubseteq \exists P^-$, then $gr(g, \alpha) = A(x)$
- if $g = P(_, x)$ and $\alpha = \exists P_1 \sqsubseteq \exists P^-$, then $gr(g, \alpha) = P_1(x, _)$
- if $g = P(_, x)$ and $\alpha = \exists P_1^- \sqsubseteq \exists P^-$, then $gr(g, \alpha) = P_1(_, x)$
- if $g = P(x, y)$ and $\alpha = P_1 \sqsubseteq P$ or $\alpha = P_1^- \sqsubseteq P^-$, then $gr(g, \alpha) = P_1(x, y)$
- if $g = P(x, y)$ and $\alpha = P_1 \sqsubseteq P^-$ or $\alpha = P_1^- \sqsubseteq P$, then $gr(g, \alpha) = P_1(y, x)$

The PerfectRef algorithm is then given as follows:

Algorithm 1: PerfectRef [Calvanese *et al.* 2007]

Input: a conjunctive query q, a TBox $\mathcal{T}$
Output: a union of conjunctive queries PR

```
1  PR ← {q};
2  repeat
3      PR' ← PR;
4      foreach q ∈ PR' do
5          foreach g ∈ q do
6              foreach α ∈ T_p do
7                  if α is applicable to g then
8                      PR ← PR ∪ {q[g → gr(g, α)]};
9          foreach g1, g2 ∈ q do
10             if g1 and g2 unify then
11                 PR ← PR ∪ {reduce(q, g1, g2)};
12 until PR' = PR;
13 return PR;
```

Where reduce applies to q the most general unifier between g_1 and g_2 and replaces each unbound variable with _

Example 1.14. *Let us continue Example 1.13 and reformulate* $q(x) = \exists y.\mathsf{User}(x) \wedge \mathsf{hasAccess}(x, y)$ *with respect to the TBox* $\mathcal{T}$ *from Example 1.8. Some of the queries resulting from rewriting* $q(x)$ *are the following:*

$$
\begin{aligned}
\mathsf{PerfectRef}(q(x), \mathcal{T}) = \{ & \\
& \exists y.\mathsf{User}(x) \wedge \mathsf{hasAccess}(x, y), && \exists y.\mathsf{Secondee}(x) \wedge \mathsf{hasAccess}(x, y), \\
& \exists y.\mathsf{FullResearcher}(x) \wedge \mathsf{hasAccess}(x, y), && \exists y.\mathsf{User}(x) \wedge \mathsf{hasOpenAccess}(x, y), \\
& \exists y.\mathsf{User}(x) \wedge \mathsf{hasConfidAccess}(x, y), && \exists y.\mathsf{User}(x) \wedge \mathsf{owns}(x, y), \\
& \exists y.\mathsf{hasAccess}(x, y), && \exists y.\mathsf{owns}(x, y), \ \ldots\}
\end{aligned}
$$

Let us detail the steps required to obtain the reformulation $\exists y.\mathsf{owns}(x, y)$*:*

1.$\mathsf{User}(x) \wedge \mathsf{hasAccess}(x, _)$	*(y is an unbound variable, replaced with _)*
2.$\mathsf{hasAccess}(x, _) \wedge \mathsf{hasAccess}(x, _)$	*apply* $\alpha = \exists\mathsf{hasAccess} \sqsubseteq \mathsf{User}$
3.$\mathsf{hasAccess}(x, _)$	*Unify*
4.$\mathsf{owns}(x, _)$	*apply* $\alpha = \mathsf{owns} \sqsubseteq \mathsf{hasAccess}$

The DL-Lite family of description logics has been designed to guarantee the existence of FOL query reformulations. Thus, for any satisfiable TBox $\mathcal{T}$ and a query q, a FOL reformulation $q_{\mathcal{T}}$ exists and can be computed in polynomial time in the size of the TBox [Calvanese *et al.* 2007].

Conflicts

The adoption of the assumption that the TBox is coherent makes the focus of satisfiability on checking whether the ABox is consistent with respect to the TBox or not. As stated above, one can build a Boolean consistency checking query from all the negative inclusion axioms entailed by the TBox, i.e., $cln(\mathcal{T})$. In DL-Lite$_\mathcal{R}$, such a query is a union of Boolean conjunctive queries, where each conjunction represents a possible violation of a negative inclusion axiom.

Since the idea of query reformulation consists in rewriting each atom of the query by applying the positive inclusions of the TBox, and the idea of computing the TBox's negative closure $cln(\mathcal{T})$ is to apply the positive inclusion axioms of the TBox to the negative inclusions to obtain all implicit negative inclusions, we can avoid computing the negative closure of the TBox. It is enough to build a union of Boolean conjunctive queries from the negative axioms in $\mathcal{T}_n$, then re-write each Boolean conjunctive query using the PerfectRef algorithm to obtain the same union of Boolean conjunctive queries produced from $cln(\mathcal{T})$ [Bienvenu *et al.* 2019]. The resulting consistency checking query is denoted by q_{unsat}.

Example 1.15. *Let us build the consistency checking query for our example's TBox $\mathcal{T}$, which contains two negative inclusion axioms* $\mathsf{Secondee} \sqsubseteq \neg\, \mathsf{FullResearcher}$ *and* $\mathsf{hasConfidAccess} \sqsubseteq \neg\, \mathsf{hasOpenAccess}$*:*

$$\begin{aligned} q_{unsat} =& \mathsf{PerfectRef}(\exists x.\mathsf{Secondee}(x) \wedge \mathsf{FullResearcher}(x), \mathcal{T}) \vee \\ & \mathsf{PerfectRef}(\exists x, y.\mathsf{hasConfidAccess}(x,y) \wedge \mathsf{hasOpenAccess}(x,y), \mathcal{T}) \\ =& (\exists x.\mathsf{Secondee}(x) \wedge \mathsf{FullResearcher}(x)) \vee (\exists x, y.\mathsf{Secondee}(x) \wedge \mathsf{owns}(x,y)) \vee \\ & (\exists x, y.\mathsf{hasConfidAccess}(x,y) \wedge \mathsf{hasOpenAccess}(x,y)) \end{aligned}$$

Notice that the resulting query q_{unsat} has the same size of the negative closure $cln(\mathcal{T})$.

- *Consider the ABox $\mathcal{A}_1$ from Example 1.9, the query q_{unsat} evaluates to false over $db(\mathcal{A}_1)$, thus $\mathcal{A}_1$ is consistent.*
- *For the ABox $\mathcal{A}_2$ from Example 1.11, the query q_{unsat} evaluates to true over $db(\mathcal{A}_2)$, because $\mathcal{A}_2$ contains the assertions* $\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)$ *and* $\mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)$ *(and also the assertions* $\mathsf{Secondee}(\mathsf{Duane})$ *and* $\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$*).*

The other interesting task related to ABox inconsistency is to identify the ABox assertions that are responsible for the inconsistency if it holds. The negative inclusion axioms in the TBox also allow us to define the notion of conflicts which represent these specific reasons of inconsistency. An assertional conflict is an inclusion-minimal inconsistent subset of the ABox that contradicts the TBox [Calvanese *et al.* 2007].

Definition 1.9 (Conflict)**.** *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be an inconsistent KB. A conflict is a subset $\mathcal{C} \subseteq \mathcal{A}$ such that $\langle \mathcal{T}, \mathcal{C} \rangle$ is inconsistent and for all $\varphi \in \mathcal{C}$, $\langle \mathcal{T}, \mathcal{C} \setminus \{\varphi\} \rangle$ is consistent.*

We denote by $\mathsf{Cf}(\mathcal{A})$ the set of all conflicts, or *the conflict set* of $\mathcal{A}$.

Example 1.16. *Let us exhibit the conflict set of the ABox $\mathcal{A}_2$ which we showed to be*

inconsistent in Example 1.11, namely:

$$\mathsf{Cf}(\mathcal{A}_2) = \left\{ \begin{array}{l} \{\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)\}, \\ \{\mathsf{Secondee}(\mathsf{Duane}), \mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)\} \end{array} \right\}$$

Note that since the ABox $\mathcal{A}_1$ of Example 1.9 is consistent, its conflict set is empty, namely, $\mathsf{Cf}(\mathcal{A}_1) = \emptyset$.

The notion of conflicts given by Definition 1.9 can be seen as a particular case of the notion of justifications in description logics. Namely, a justification is a minimal subset of a knowledge base that is sufficient to draw a conclusion [Kalyanpur *et al.* 2007, Ji *et al.* 2022]. Formally, let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite$_\mathcal{R}$ knowledge base and ϕ be a conclusion. Then:

$$\mathcal{K}' \subseteq \mathcal{K} \text{ is a justification of } \phi \text{ if } \mathcal{K} \vDash \phi \text{ and for all } \mathcal{K}'' \subset \mathcal{K}', \mathcal{K}'' \nvDash \phi$$

In a justification, the conclusion ϕ can be an axiom, an assertion, or a set of these elements. In the case of conflicts, the conclusion is simply a contradiction. Thus, a conflict is a minimal subset that does not admit any model. In the following, we focus on conflicts as a distinct concept, as it plays an important role in inconsistency-tolerant semantics.

The conflict set of an inconsistent DL-Lite$_\mathcal{R}$ knowledge base can be computed in polynomial time in data and combined complexity [Bourgaux 2016, Bienvenu *et al.* 2019]. In addition, the number of conflicts in $\mathsf{Cf}(\mathcal{A})$ is polynomial in the ABox's size. In the worst case, it is bounded by $|\mathcal{A}|^2$. This is an important characteristic of the DL-Lite$_\mathcal{R}$ dialect, as it helps to establish the efficiency of some of the inconsistency-tolerant semantics which will be discussed later in this thesis.

Instance checking

As shown by Definition 1.2, an assertion φ follows from a knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ if all the models of $\mathcal{K}$ are also models of φ. Checking if an assertion follows from a knowledge base can be approached in two distinct ways. First, the problem can be reduced to knowledge base satisfiability as follows:

- If φ is a concept assertion, i.e., $A(a)$, we introduce a fresh concept name A' into the knowledge base by adding a new negative inclusion axiom between A and A': $A \sqsubseteq \neg A'$ and a new assertion $A'(a)$ (on the same individual of A). Then $\mathcal{K} \vDash A(a)$ if and only if $\langle \mathcal{T} \cup \{A \sqsubseteq \neg A'\}, \mathcal{A} \cup \{A'(a)\} \rangle$ is unsatisfiable.
- If φ is a role assertion, i.e., $R(a, b)$, we introduce a fresh role name R' into the knowledge base by adding a new negative inclusion axiom between R and R': $R \sqsubseteq \neg R'$ and a new assertion $R'(a, b)$ (on the same individuals of R). Then $\mathcal{K} \vDash R(a, b)$ if and only if $\langle \mathcal{T} \cup \{R \sqsubseteq \neg R'\}, \mathcal{A} \cup \{R'(a, b)\} \rangle$ is unsatisfiable.

The second method is a special case of Boolean query entailment (Definition 1.5). The idea consists in providing a Boolean query with a single atom that is the given assertion φ, then reformulating the query using the PerfectRef algorithm to obtain all the possible assertions implying φ. Then the reformulation result can be evaluated on the database-like interpretation of the ABox.

Instance checking from a consistent DL-Lite$_\mathcal{R}$ knowledge base is in AC^0 with respect to data complexity and in PTime with respect to combined complexity [Artale *et al.* 2009, Bienvenu & Bourgaux 2017].

1.3 Inconsistency-Tolerant Semantics

One of the main challenges in reasoning with formal ontologies is safely drawing conclusions in the presence of inconsistency. In fact, inferences can be limited only to consistent knowledge bases. However, such an overly cautious approach remains unrealistic in a world where large amounts of data flow to the web everyday.

Several attitudes have been discussed in the literature to deal with contradictory information. Such attitudes can be classified by the representation framework, where each setting has its most suitable approaches. They can also be classified by their level of involvement in a framework. For instance, syntactic methods often target the elements composing a knowledge base, by modifying or ignoring some of them, while semantic methods target the inference system by modifying or adding rules to its inference mechanisms. We prefer to highlight the differences in the principles behind the different attitudes, independently of the setting employed. Another complementary line of work investigates how inconsistency can be quantified, by assigning scores that reflect the severity or distribution of conflicts within a knowledge base [Thimm 2019, De Bona *et al.* 2019, Thimm 2018, Grant & Hunter 2011].

Inconsistency management approaches can be classified into three main forms:

1. Reasoning from one or multiple consistent subsets of the initial inconsistent knowledge base. This approach does not target a modification or deletion of contradictory facts, it simply ignores them. Then, it applies the standard inference relation to those consistent subsets, which are often called repairs, to derive consequences. This kind of methods have roots in propositional logic [Thimm & Santos 2024, Finkelstein *et al.* 1994, Baral *et al.* 1992, Rescher & Manor 1970], the database setting [Wijsen 2019, Staworko *et al.* 2012, Arenas *et al.* 1999], existential rules [Lukasiewicz *et al.* 2022, Lukasiewicz *et al.* 2015], and description logics [Baget *et al.* 2016, Bienvenu & Bourgaux 2017, Lembo *et al.* 2010]. This type is mostly useful with large datasets, where data are often read-only and the edition or the modification of contradictory facts is costly. A real-world example of such a situation is querying Wikidata [Vrandečić & Krötzsch 2014] based on ontologies such as the one provided by DBpedia [Lehmann *et al.* 2015].

2. Accepting inconsistency and modifying the inference relation of the underlying representation setting. Some of these methods include paraconsistent logics [Anderson & Belnap 1975], such as the three-valued semantics [Arieli *et al.* 2011, Grant & Hunter 2023] or logics based on distributive lattices [Esteva *et al.* 2021]. In the same spirit, some methods that are based on argumentation frameworks use the notion of argument (or justification) to decide in favour of a conclusion or its opposite [Besnard & Hunter 2008, Toulmin 2003, Benferhat *et al.* 1993, Dung 1995].

3. Modifying the inconsistent knowledge base by rewriting or weakening some of its information. This kind of method is, for instance, used in reasoning tolerant of exceptions, where the knowledge base is incomplete, and the system must complement it to allow inference.

In this work, we adopt the first attitude, where the challenges lie in deciding which facts must be preserved in a repair, which ones should be removed (or ignored) and whether to produce

a single or multiple repairs. In general, some methods are adventurous, in the sense that they derive conclusions from at least one repair, which is also called existential (or credulous) inference, while other methods are said to be cautious and infer the consequences that follow from all repairs, which is also called universal (also inevitable or skeptical) inference [Rescher & Manor 1970, Benferhat *et al.* 1997, Arieli *et al.* 2024].

Let us first recall the notion of repair in the context of inconsistent DL-Lite$_\mathcal{R}$ knowledge bases. Note that we only consider inclusion-maximal consistent subsets of the knowledge base.

Definition 1.10 (Repair)**.** *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite$_\mathcal{R}$ knowledge base. A subset of $\mathcal{A}$ is a inclusion-maximal consistent subset of $\mathcal{A}$, denoted $\mathcal{R}$, if:*

- $\mathcal{R} \subseteq \mathcal{A}$,
- $\langle \mathcal{T}, \mathcal{R} \rangle$ *is consistent, and;*
- *For all $\mathcal{R}'$ such that $\mathcal{R} \subset \mathcal{R}' \subseteq \mathcal{A}$, $\langle \mathcal{T}, \mathcal{R}' \rangle$ is inconsistent.*

$\mathcal{R}$ is said to be a repair of $\mathcal{A}$.

Note that in some work efforts, the notion of repair is used even if the condition of maximality is not satisfied.

Example 1.17. *Let us recall the TBox of our example, alongside the ABox from Example 1.11 (which we will refer to as $\mathcal{A}$ from now on):*

$$\mathcal{T} = \left\{ \begin{array}{ll} 1.\ \mathsf{Secondee} \sqsubseteq \mathsf{User} & 7.\ \exists \mathsf{hasAccess}^- \sqsubseteq \mathsf{Resource} \\ 2.\ \mathsf{FullResearcher} \sqsubseteq \mathsf{User} & 8.\ \exists \mathsf{hasConfidAccess}^- \sqsubseteq \mathsf{Confidential} \\ 3.\ \mathsf{Confidential} \sqsubseteq \mathsf{Resource} & 9.\ \mathsf{hasConfidAccess} \sqsubseteq \neg\, \mathsf{hasOpenAccess} \\ 4.\ \mathsf{Secondee} \sqsubseteq \neg\, \mathsf{FullResearcher} & 10.\ \mathsf{owns} \sqsubseteq \mathsf{hasAccess} \\ 5.\ \exists \mathsf{hasAccess} \sqsubseteq \mathsf{User} & 11.\ \mathsf{hasOpenAccess} \sqsubseteq \mathsf{hasAccess} \\ 6.\ \exists \mathsf{owns} \sqsubseteq \mathsf{FullResearcher} & 12.\ \mathsf{hasConfidAccess} \sqsubseteq \mathsf{hasAccess} \end{array} \right\}$$

$$\mathcal{A} = \left[\begin{array}{l} \mathsf{FullResearcher}(\mathsf{Ben}) \\ \mathsf{Secondee}(\mathsf{Duane}) \\ \mathsf{Confidential}(\mathsf{r}_1) \\ \mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1) \\ \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1) \\ \mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2) \end{array} \right]$$

Since a repair is consistent, the assertions that contradict some TBox axiom cannot belong to the same repair. The ABox $\mathcal{A}$ admits four (4) repairs, which are illustrated by Table 1.3, where the check-marks indicate the membership of an assertion to a repair.

1.3.1 Standard Repair-Based Semantics

A selection strategy can be applied to guide the choice of one or multiple repairs of an ABox for reasoning purposes. Among others, one can cite safe inference, existential inference, universal inference, and majority-based inference [Baget *et al.* 2016]. The methods for computing and

Table 1.3: The repairs $\mathcal{R}_1, \ldots, \mathcal{R}_4$ of the ABox $\mathcal{A}$

	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_3$	$\mathcal{R}_4$
FullResearcher(Ben)	✓	✓	✓	✓
Secondee(Duane)	✓		✓	
Confidential(r_1)	✓	✓	✓	✓
hasConfidAccess(Ben,r_1)	✓	✓		
hasOpenAccess(Ben,r_1)			✓	✓
owns(Duane,r_2)		✓		✓

selecting repairs are often referred to as *inconsistency-tolerant semantics*. Moreover, since the TBox is usually assumed to be coherent in the context of formal ontologies, the repairs are also called data repairs to highlight their focus on repairing the ABox with respect to the domain knowledge encoded in the TBox. Figure 1.1 provides an overview of the major standard approaches for inference in the presence of inconsistency using data repairs.

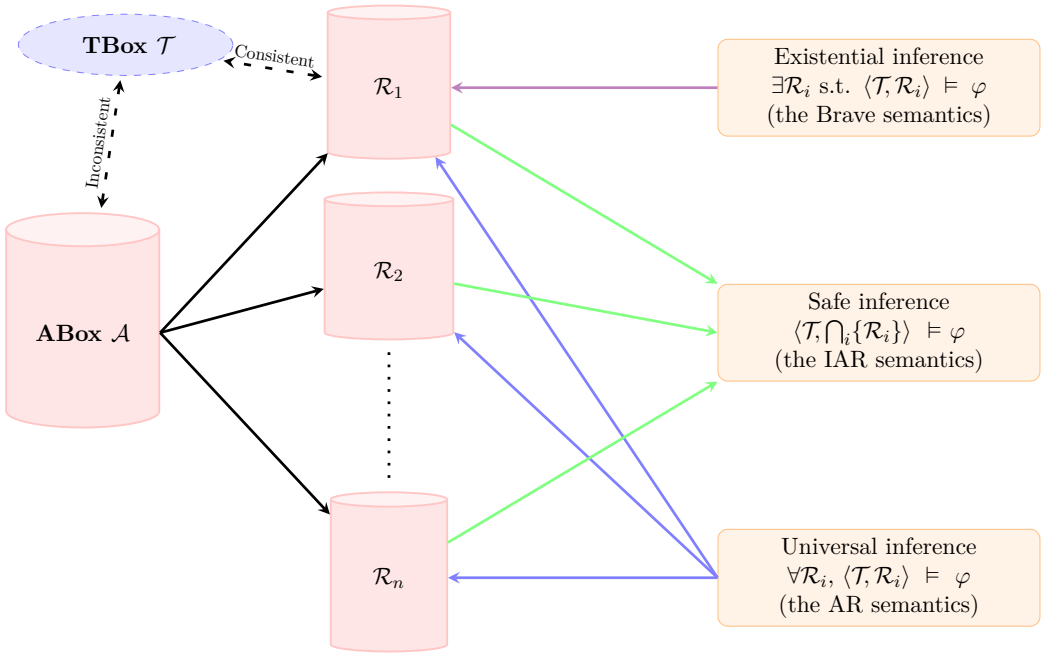


Figure 1.1: Overview of standard ABox repairs.

An inconsistency-tolerant semantics is a query answering semantics from inconsistent knowledge bases. It relies on posing queries to one or multiple data repairs rather than posing them to the inconsistent knowledge base. Note that methods that follow other attitudes, such as paraconsistent logics [Arieli *et al.* 2011], are also considered inconsistency-tolerant, since they also avoid query answers explosion (where everything becomes derivable from a contradiction).

Different aspects must be taken into account when studying inconsistency-tolerant semantics. A very important one is efficiency. Indeed, inconsistency-tolerant semantics are applied within data access frameworks, thus a reasonable computational complexity is expected from the methods to compute the repairs and draw conclusions from them. In the following, and unless specified otherwise, the same DL complexity measures (as given in Section 1.1.3) are retained, with a particular focus on data complexity.

In the remainder of this section, we consider a DL-Lite$_\mathcal{R}$ knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, which may be inconsistent, and we denote the set of the repairs of $\mathcal{A}$ by $(\mathcal{R}_1, \ldots, \mathcal{R}_m)$, as given by Definition 1.10. Note that when $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is consistent, it admits a single repair which is $\mathcal{A}$ itself, and all semantics amount to the standard logical entailment from $\mathcal{K}$.

The ABox repair (AR) semantics

Let us first start by recalling one of the most fundamental inconsistency-tolerant semantics, the ABox Repair (AR) semantics, which was defined in several settings, such as propositional logic [Arieli *et al.* 2024, Konieczny *et al.* 2019], the database setting [Wijsen 2019, Koutris & Wijsen 2021], existential rules [Lukasiewicz *et al.* 2022], and description logics [Bienvenu *et al.* 2019, Bienvenu & Bourgaux 2017]. The AR semantics draws conclusions (universally) from all the maximal consistent subsets of an inconsistent ABox. Note that AR semantics is also referred to as consistent query answering (CQA) in the case of query answering from inconsistent databases [Wijsen 2019] and description logic knowledge bases [Bienvenu & Rosati 2013].

Definition 1.11 (AR semantics)**.** *An assertion φ follows from the knowledge base $\mathcal{K}$ using the AR semantics, denoted by $\mathcal{K} \models^{\mathsf{AR}} \varphi$, if for each repair $\mathcal{R}_i$ of $\mathcal{A}$, $\langle \mathcal{T}, \mathcal{R}_i \rangle \models \varphi$. Where $\models$ denotes the standard DL-Lite$_\mathcal{R}$ entailment.*

Example 1.18. *Let us now apply the AR semantics to our knowledge base from Example 1.17:*

- *Each $\mathcal{R}_i$ contains* $\mathsf{FullResearcher}(\mathsf{Ben})$ *and* $\mathsf{Confidential}(\mathsf{r}_1)$ *which allow to derive* $\mathsf{User}(\mathsf{Ben})$ *and* $\mathsf{Resource}(\mathsf{r}_1)$*, respectively.*
- *Each $\mathcal{R}_i$ contains either* $\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)$ *or* $\mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)$*, and both assertions imply* $\mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1)$*.*
- *Each $\mathcal{R}_i$ contains either* $\mathsf{Secondee}(\mathsf{Duane})$ *or* $\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$*, and both imply* $\mathsf{User}(\mathsf{Duane})$*.*

The AR semantics presents some very desirable properties for querying inconsistent ABoxes. More precisely, it is a cautious approach in the sense that it considers all the repairs when drawing conclusions. However, inference under the AR semantics is usually intractable: conjunctive query answering and instance checking under the AR semantics over DL-Lite knowledge bases are coNP-complete with respect to data complexity [Lembo *et al.* 2010, Bienvenu 2012a].

The Intersection of ABox Repair (IAR) semantics

The IAR semantics has been proposed as an under-approximation of the AR semantics, in order to overcome its shortcomings [Lembo *et al.* 2010]. Instead of posing queries over all of the ABox repairs, the IAR semantics evaluate queries over a single consistent subset, obtained from the intersection of all the repairs.

Definition 1.12 (IAR semantics)**.** *The IAR repair of $\mathcal{A}$ is given by:*

$$\mathsf{IAR}(A) = \bigcap_i \{\mathcal{R}_i \mid \mathcal{R}_i \text{ is a repair of } \mathcal{A}\}$$

An assertion φ follows from the knowledge base $\mathcal{K}$ using the IAR semantics, denoted by $\mathcal{K} \models^{\mathsf{IAR}} \varphi$, if $\langle \mathcal{T}, \mathsf{IAR}(A) \rangle \models \varphi$. Where $\models$ denotes the standard DL-Lite$_\mathcal{R}$ entailment.

Example 1.19. *Let us continue and apply the IAR semantics on our knowledge base. The IAR repair is obtained simply from the intersection of all the repairs given in Example 1.17:*

$$\mathsf{IAR}(\mathcal{A}) = \{\mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{Confidential}(\mathsf{r}_1)\}$$

Notice that the assertions $\mathsf{User}(\mathsf{Ben})$ *and* $\mathsf{Resource}(\mathsf{r}_1)$ *are also derived using the IAR semantics, since they follow respectively from* $\mathsf{FullResearcher}(\mathsf{Ben})$ *and* $\mathsf{Confidential}(\mathsf{r}_1)$*. However, unlike the AR semantics,* $\mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1)$ *and* $\mathsf{User}(\mathsf{Duane})$ *are not derived using the IAR semantics.*

The IAR semantics has been characterised tractably in DL-Lite$_\mathcal{R}$ using the notion of free assertions. The characterisation allows one to avoid exhibiting all the repairs, which is intractable. Instead, it relies on computing the conflict set of the ABox as given per Definition 1.9. Formally, given an assertion φ:

$$\varphi \in \mathsf{IAR}(\mathcal{A}) \text{ if and only if for all } \mathcal{C} \in \mathsf{Cf}(\mathcal{A}), \varphi \notin \mathcal{C}$$

Conjunctive query answering and instance checking under the IAR semantics over DL-Lite knowledge bases is in AC^0 with respect to data complexity [Lembo *et al.* 2010].

The Brave semantics

The AR semantics is a cautious strategy because it draws conclusions from all the ABox repairs. One can follow an adventurous attitude and consider the conclusions that follow from at least one repair. This is the case with the Brave semantics [Bienvenu & Rosati 2013].

Definition 1.13 (Brave semantics)**.** *An assertion* φ *follows from the knowledge base* $\mathcal{K}$ *under the Brave semantics, denoted* $\mathcal{K} \models^{\mathsf{brave}} \varphi$*, if there is a repair* $\mathcal{R}_i$ *of* $\mathcal{A}$ *such that* $\langle \mathcal{T}, \mathcal{R}_i \rangle \models \varphi$*. Where* $\models$ *denotes the standard DL-Lite*$_\mathcal{R}$ *entailment.*

Example 1.20. *Let us now apply the Brave semantics on our knowledge base. In addition to the assertions that follow from the knowledge base under the AR semantics, the Brave semantics permits the inference of:*

- $\mathsf{Secondee}(\mathsf{Duane})$ *because it follows from the repairs* $\mathcal{R}_1$ *and* $\mathcal{R}_3$*,*
- $\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)$ *because it follows from the repairs* $\mathcal{R}_1$ *and* $\mathcal{R}_2$*,*
- $\mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)$ *because it follows from the repairs* $\mathcal{R}_3$ *and* $\mathcal{R}_4$*, and;*
- $\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$ *because it follows from the repairs* $\mathcal{R}_2$ *and* $\mathcal{R}_4$*.*

The Brave semantics is an over-approximation of the AR semantics, and inference with it is tractable in DL-Lite. Indeed, Conjunctive query answering and instance checking under the Brave semantics over DL-Lite knowledge bases is in AC^0 with respect to data complexity [Bienvenu & Rosati 2013]. However, one can see that the Brave semantics does not prevent the inference of conflicting assertions. This is the reason why it is considered adventurous.

1.3.2 Parametrised ABox repairs

We have seen the IAR and the Brave semantics which are, respectively, an under-approximation and an over-approximation of the AR semantics. In addition, other parametrised approaches that approximate the AR semantics can be found in the literature. For instance, the k-support semantics, its dual notion the k-defeaters semantics, which were defined in the context of DL-Lite knowledge bases [Bienvenu & Rosati 2013], and the k-lazy semantics, which was defined in the context of Datalog $\pm$ [Lukasiewicz *et al.* 2012].

The k-support and k-defeater semantics

A support of a query (or an assertion) in the case of the k-support semantics is defined to be a consistent subset of the ABox that allows to infer the query (or the assertion). Then, the k-support semantics can be seen as a restriction of the AR semantics, such that a maximum of k different supports can be used. Formally, the k-support semantics is defined as follows.

Definition 1.14 (k-support semantics)**.** *An assertion φ follows from the knowledge base $\mathcal{K}$ under the k-support semantics, denoted $\mathcal{K} \models^{k-\mathsf{support}} \varphi$, if there exist (not necessarily distinct) subsets $\mathcal{S}_1, \ldots, \mathcal{S}_k$ of $\mathcal{A}$ that satisfy the following conditions:*

- *each $\mathcal{S}_i$ supports φ in $\mathcal{A}$: $\langle \mathcal{T}, \mathcal{S}_i \rangle$ is consistent and $\langle \mathcal{T}, \mathcal{S}_i \rangle \models \varphi$; and,*
- *for every $\mathcal{R}$ a repair of $\mathcal{A}$ (as per Definition 1.10), there is some $\mathcal{S}_i$ with $\mathcal{S}_i \subseteq \mathcal{R}$.*

The k-support semantics is an under-approximation of the AR semantics that holds the following properties. It coincides with the IAR semantics when $k = 1$ and with the AR semantics for some k. In addition, for every $k \geq 1$, if an assertion follows under k-supports, then it also follows under $k+1$-supports. The maximum number that k can take depends on the number of supports a query has in an ABox, and this number does not depend on the number of repairs the ABox admits.

Example 1.21. *Consider the knowledge base of Example 1.17:*

- *The assertion* $\mathsf{Confidential}(\mathsf{r}_1)$ *follows from the knowledge base under the* 1*-support semantics, because it has a support, namely* $\{\mathsf{Confidential}(\mathsf{r}_1)\}$ *(the assertion itself) that belongs to every repair $\mathcal{R}_i$.*
- *The assertion* $\mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1)$ *follows from the knowledge base under the* 2*-supports semantics, because it has two supports, each belonging to two different repairs of the knowledge base, namely* $\{\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)\} \subseteq \mathcal{R}_1$, $\{\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)\} \subseteq \mathcal{R}_2$, $\{\mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)\} \subseteq \mathcal{R}_3$, *and* $\{\mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)\} \subseteq \mathcal{R}_4$.

In contrast with the k-support semantics, the k-defeater over-approximates the AR semantics. It is formally defined as follows.

Definition 1.15 (k-defeater semantics)**.** *An assertion φ follows from the knowledge base $\mathcal{K}$ under the k-defeater semantics, denoted $\mathcal{K} \models^{k-\mathsf{defeat}} \varphi$, if there is no consistent subset $\mathcal{B}$ of $\mathcal{A}$ with $|\mathcal{B}| \leq k$ such that $\langle \mathcal{T}, \mathcal{B} \cup \mathcal{S} \rangle \models \bot$ for every inclusion-minimal subset $\mathcal{S} \subseteq \mathcal{A}$ that supports φ. The set $\mathcal{B}$ is called the defeating set.*

The k-defeater semantics coincides with the Brave semantics for $k = 0$, and converges to the AR semantics in the limit. In addition, for every $k \geq 0$, if an assertion follows under the $k+1$-defeaters, then it also follows under the k-defeaters semantics.

Example 1.22. *Consider the knowledge base of Example 1.17:*

- *The assertions that follow under the Brave semantics also follow under the* 0*-defeater semantics, namely* $\mathsf{Secondee}(\mathsf{Duane})$, $\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)$, $\mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)$, *and* $\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$.
- *The assertions that follow under the* 1*-defeater semantics are the same under the AR semantics. For instance, the assertion* $\mathsf{User}(\mathsf{Duane})$ *does not admit a defeating set, i.e., a subset of* $\mathcal{A}$ *of size* 1 *that invalidates both its supports* $\{\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)\}$ *and* $\{\mathsf{Secondee}(\mathsf{Duane})\}$. *Indeed, each subset invalidates the other, but there is no single subset that invalidates both. The same applies to the remaining assertions that are entailed using the AR semantics.*

The k-lazy semantics

The k-lazy semantics relies on an alternative notion of repair (k-lazy repairs) built by processing clusters of contradictory assertions, in essence, a set of mutually conflicting ABox assertions [Lukasiewicz *et al.* 2012]. For each cluster, either an inclusion-minimal subset of at most k assertions is removed to restore consistency within the cluster, or, if no such subset exists, the entire cluster is removed. Since whole clusters may be discarded, k-lazy repairs need not coincide with standard repairs.

Definition 1.16 (Clusters and k-lazy repairs)**.** *Let* $\equiv_{\perp}^{\mathcal{K}}$ *be the smallest equivalence relation on the assertions of* $\mathcal{A}$ *such that whenever two assertions* φ_i, φ_j *occur together in a conflict of* $\mathcal{A}$*, we have* $\varphi_i \equiv_{\perp}^{\mathcal{K}} \varphi_j$*. The* clusters *of* $\mathcal{K}$ *are the equivalence classes of* $\equiv_{\perp}^{\mathcal{K}}$*.*

A k-lazy repair *of* $\mathcal{K}$ *is obtained by, for each cluster* $\mathcal{C}_i$ *of* $\mathcal{K}$*, removing either:*

1. *an inclusion-minimal subset* $\mathcal{C}_i' \subseteq \mathcal{C}_i$ *with* $|\mathcal{C}_i'| \leq k$ *such that* $\langle \mathcal{T}, \mathcal{C}_i \setminus \mathcal{C}_i' \rangle$ *is consistent, or*
2. *the whole cluster* $\mathcal{C}_i$ *if no subset* $\mathcal{C}_i' \subseteq \mathcal{C}_i$ *satisfies the first condition.*

The k-lazy semantics is then defined like the AR semantics, except that it uses the notion of k-lazy repairs instead of the classical notion of repair.

Definition 1.17 (k-lazy semantics)**.** *An assertion* φ *follows from the knowledge base* $\mathcal{K}$ *under the* k*-lazy semantics, denoted by* $\mathcal{K} \models^{\mathsf{lazy}} \varphi$*, if for all* $\mathcal{R}_i$ *a* k*-lazy repair of* $\mathcal{A}$*,* $\langle \mathcal{T}, \mathcal{R}_i \rangle \models \varphi$*.*

The k-lazy semantics coincides with the IAR semantics when $k = 0$ and under-approximates the Brave semantics for every value of k. Note that in this parametrised semantics, the value of k is an input to the problem such that an algorithm to compute lazy repairs must be provided with a value of k. In addition, for values of k that are equal or larger than the size of the largest cluster of the ABox, the same k-lazy repairs are computed. In essence, there will be no more elements that can be removed from a cluster to restore its consistency at that level (as given by the first condition of Definition 1.16).

Example 1.23. *Let us apply the k-lazy semantics on our knowledge base of Example 1.17. We first compute the clusters of $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$:*

- $\{\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)\}$
- $\{\mathsf{Secondee}(\mathsf{Duane}), \mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)\}$

There is a single 0-lazy repair, which consists of removing both clusters to restore consistency. This is equivalent to computing the IAR repair of $\mathcal{A}$. Removing one assertion from each of these clusters restores consistency in them, and hence the 1-lazy repairs of our knowledge base coincide with the standard repairs, which were given in Example1.17.

Note that the k-lazy repairs do not always coincide with standard repairs. In our example, both clusters are composed of two assertions; thus, it is sufficient to remove a single assertion from each cluster to obtain a lazy repair. If there were a cluster with strictly more than two assertions, it would require removing the entire cluster to restore consistency and compute the 1-lazy repairs. The same applies for larger clusters and larger values of k.

1.3.3 Prioritised Repairs

Another lever that has been widely used when dealing with inconsistent information in the literature is to resolve conflicts by preferring some pieces of information over others, typically via a total or partial preorder, allowing for a more informed selection of consistent subsets.

The use of an order relation over a knowledge base has been thoroughly studied in various logic-based frameworks, such as propositional logic and argumentation systems [Baral *et al.* 1992, Amgoud & Vesic 2014], databases [Perez & Wijsen 2021, Staworko *et al.* 2012, Kimelfeld *et al.* 2020, Bertossi 2011, Arenas *et al.* 1999], conditional rules and default logics [Poole 1985, Brewka 1989], existential rules [Lukasiewicz *et al.* 2023], and description logics [Bienvenu & Bourgaux 2017, Benferhat *et al.* 2015]. These studies also involve benchmarks addressing prioritised information such as the CQAPri benchmark, which provides various methods for query answering from inconsistent knowledge bases including priority-based inconsistency-tolerant semantics, such as the preferred repair semantics [Bienvenu *et al.* 2019, Bienvenu *et al.* 2014].

A strongly related line of work is defeasible reasoning in Description Logics, where conflicts are settled in a prioritised manner by ranking axioms so that more specific, or exceptional ones override more normal ones [Giordano *et al.* 2008, Giordano *et al.* 2015, Casini & Straccia 2010, Casini & Straccia 2012]. Originally developed for non-monotonic reasoning, such typicality-based mechanisms interpret defeasible axioms as holding by default unless overridden by more exceptional (higher-priority) axioms.

Before introducing the semantics that deal with prioritised knowledge bases, we start by showing how the priorities are represented. Since a DL-Lite$_\mathcal{R}$ TBox is assumed to be coherent, the axioms contained therein are considered to have the highest priority level. ABox assertions are the ones that are questionable, since they may be inconsistent with respect to the TBox. The priority levels can be represented using a set of weights (or priority degrees), thus inducing a total preorder relation between the ABox assertions. Alternatively, the preorder relation can be expressed by stratifying the ABox into layers.

Formally, a total preorder defined over $\mathcal{A}$, denoted by $\geq$, is a reflexive and transitive binary relation such that for any two assertions φ_j and φ_k in the ABox, we have $\varphi_j \geq \varphi_k$ or $\varphi_k \geq \varphi_j$.

Let $>$ be the associated strict order (defined as $\varphi_j > \varphi_k$ if and only if $\varphi_j \geq \varphi_k$ holds and $\varphi_k \geq \varphi_j$ does not hold). Let $\equiv$ be the associated equivalence relation (defined by $\varphi_k \equiv \varphi_j$ if both $\varphi_k \geq \varphi_j$ and $\varphi_j \geq \varphi_k$ hold).

The totally preordered ABox is denoted $\mathcal{A}_{\geq}$ and the total preorder relation $\geq$ can be represented with well-ordered partitions (or stratifications) of ABox assertions, of the form $(\mathcal{S}_1, \ldots, \mathcal{S}_n)$, which divides the ABox into n layers, where:

- $\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_n = \mathcal{A}_{\geq}$.
- $\mathcal{S}_1 = \{\varphi_j \in \mathcal{A}_{\geq} : \forall \varphi_k \in \mathcal{A}_{\geq}, \varphi_j \geq \varphi_k\}$.
- $\mathcal{S}_i = \{\varphi_j \in \mathcal{A}_{\geq} \setminus (\mathcal{S}_1 \cup \cdots \cup \mathcal{S}_{i-1}) : \forall \varphi_k \in \mathcal{A}_{\geq} \setminus (\mathcal{S}_1 \cup \cdots \cup \mathcal{S}_{i-1}), \varphi_j \geq \varphi_k\}$, for $i = 2, \ldots, n$.

The assertions in $\mathcal{S}_1$ are the most certain and those in $\mathcal{S}_n$ are the least certain in the ABox. The assertions in any stratum $\mathcal{S}_i$ $(i = 1, \ldots, n)$ are equally certain. We use the notation $\mathcal{K}_{\geq} = \langle \mathcal{T}, \mathcal{A}_{\geq} \rangle$ to denote a prioritised DL-Lite$_{\mathcal{R}}$ knowledge base.

The preferred repair semantics

The preferred repair semantics extends the ABox repair semantics, which answers queries universally on all the ABox repairs to the case of prioritised knowledge bases. First, let us introduce the notion of a preferred inclusion-based repair, which was used in several settings, such as propositional logic [Brewka 1989, Benferhat *et al.* 1998] and the DL-Lite framework [Bienvenu *et al.* 2014, Benferhat *et al.* 2015].

Definition 1.18 (Preferred repair)**.** *A preferred inclusion-based repair (PAR) $\mathcal{P} = P_1 \cup \ldots \cup P_n$ of $\mathcal{A}_{\geq} = \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_n$ is such that there is no inclusion-maximal subset $\mathcal{P}' = P'_1 \cup \ldots \cup P'_n$ of $\mathcal{A}_{\geq} = \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_n$, and an integer i where:*

(i) P_i is strictly included in P'_i, and;

(ii) for all $j = 1..(i-1)$, P_j is equal to P'_j.

Note that a consistent ABox $\mathcal{A}_{\geq}$ admits a single preferred repair (which is $\mathcal{A}_{\geq}$ itself).

Inference under the preferred repair semantics is given by universal querying of all the preferred repairs of a given knowledge base, which are denoted by $\mathsf{PAR}(\mathcal{A}_{\geq})$.

Definition 1.19 (Preferred repair semantics)**.** *An assertion φ follows from the knowledge base $\mathcal{K}_{\geq}$ under the preferred repair semantics, denoted $\mathcal{K}_{\geq} \models^{\mathsf{PAR}} \varphi$, if for each$\mathcal{P} \in \mathsf{PAR}(\mathcal{A}_{\geq})$, $\langle \mathcal{T}, \mathcal{P} \rangle \models \varphi$. Where $\models$ denoted the standard DL-Lite$_{\mathcal{R}}$ entailment.*

Example 1.24. *Let us introduce a prioritised version of our running example. Assume the well-ordered partition built from the same assertions in the ABox $\mathcal{A}$ of Example 1.17, such that $\mathcal{A}_{\geq} = \mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3 \cup \mathcal{S}_4$, where:*

$\mathcal{A}_{\geq} =$

Partitions	Assertions
$\mathcal{S}_1$	$\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$
$\mathcal{S}_2$	$\mathsf{FullResearcher}(\mathsf{Ben})$
$\mathcal{S}_3$	$\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)$
$\mathcal{S}_4$	$\mathsf{Secondee}(\mathsf{Duane}), \mathsf{Confidential}(\mathsf{r}_1)$

The ABox $\mathcal{A}_{\geq}$ admits two inclusion-based preferred repairs, namely $\mathsf{PAR}(\mathcal{A}_{\geq}) = \{\mathcal{R}_1, \mathcal{R}_2\}$ *such that:*

$$\mathcal{R}_1 = \{\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{Confidential}(\mathsf{r}_1)\}$$
$$\mathcal{R}_2 = \{\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{Confidential}(\mathsf{r}_1)\}$$

The lexicographic preferred-based repair

One of the main challenges faced by the preferred repair semantics is that it does not always return a single preferred assertional repair. A widely used strategy to choose a preferred repair is the lexicographic approach, in which a cardinality-based criterion was used instead of the set-inclusion criterion [Bienvenu *et al.* 2014, Benferhat *et al.* 2015].

Definition 1.20 (Lexicographic preferred-based repair)**.** *Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be two consistent subsets of $(\mathcal{S}_1, \ldots, \mathcal{S}_n)$.*

- *$\mathcal{R}_1$ is as preferred as $\mathcal{R}_2$ if $|\mathcal{R}_1 \cap \mathcal{S}_i| = |\mathcal{R}_2 \cap \mathcal{S}_i|$, for $i = 1, \ldots, n$.*
- *$\mathcal{R}_1$ is strictly more preferred than $\mathcal{R}_2$ if $|\mathcal{R}_1 \cap \mathcal{S}_i| > |\mathcal{R}_2 \cap \mathcal{S}_i|$ for some $i \in \{1, \ldots, n\}$ and $|\mathcal{R}_1 \cap \mathcal{S}_j| = |\mathcal{R}_2 \cap \mathcal{S}_j|$, for all j such that $1 \leq j < i$.*

Where $|X|$ is the cardinality of the set X.

Then $\mathcal{R} \subseteq \mathcal{A}_{\geq}$ is a lexicographic preferred-based repair of $\mathcal{A}_{\geq}$ if there is no $\mathcal{R}' \subseteq \mathcal{A}_{\geq}$ such that $\mathcal{R}'$ is strictly preferred to $\mathcal{R}$. The set of all the lexicographic preferred repairs of $\mathcal{A}_{\geq}$ is denoted $\mathsf{PAR}_{lex}(\mathcal{A}_{\geq})$.

Example 1.25. *In the case of our prioritised ABox $\mathcal{A}_{\geq}$ from Example 1.24, both of its preferred repairs have the same cardinality, namely, $|\mathcal{R}_1| = |\mathcal{R}_2| = 4$. Hence, both $\mathcal{R}_1$ and $\mathcal{R}_2$ are lexicographic preferred repairs of $\mathcal{A}_{\geq}$.*

Conjunctive query answering under the preferred repair semantics and the lexicographic preferred-based repair semantics over DL-Lite knowledge bases is coNP-complete with respect to data complexity.

1.4 Uncertainty Management

The logic-based representation languages discussed in Section 1.1 allow specification and reasoning about a domain of interest. They mainly focus on sufficiently expressing knowledge about individuals and their relations while keeping desirable computational properties to allow checking the existence of a satisfiable state of the world (or interpretation) and making inferences from a knowledge base (answering queries). However, aiming at finding such a balance between the expressive power and the computational complexity does not often cover for some important situations where the information is incomplete, imprecise, fuzzy, or uncertain.

Another important dimension for the knowledge representation and reasoning community is that of representing and reasoning about preferences and uncertainty in knowledge bases. In real-world situations, some pieces of information might be considered more reliable than others. This difference in reliability can be related to the nature of data, data originating from different sources, or simply an expression of users' preferences. The literature offers a variety of representation frameworks to capture preferences and uncertainty in data

and take advantage of them in the reasoning process. For instance, some of the main uncertainty representation frameworks, in particular those that have been applied to description logics (DLs), include weighted logics [Baader *et al.* 2019, Alviano *et al.* 2023], fuzzy DLs [Giordano *et al.* 2022, Straccia 2015, Huitzil *et al.* 2020], belief functions [Dubois *et al.* 2023, Frittella *et al.* 2020, Destercke 2018, Fagin *et al.* 2003], Spohn's Ordinal Conditional Functions (OCF) [Spohn 2014], and possibility theory [Dubois & Prade 2014, Dubois & Prade 2018, Banerjee *et al.* 2017].

In this section, we provide a brief overview of two main uncertainty representation frameworks: probability theory and possibility theory. Then, we focus on the latter by recalling its application on propositional logic and the description logic DL-Lite. Probability theory focuses on capturing conditional dependencies and representing belief change when provided with new information, whereas possibility theory is more suitable to represent vague, incomplete, or ambiguous information [Denœux *et al.* 2020]. This distinction motivates our interest in possibility theory over probability theory, taking advantage of its natural capacity to model ignorance, unlike probability theory where the auto-duality of probabilities imposes a complementary representation of belief and disbelief [Dubois & Prade 2018, Denœux *et al.* 2020].

1.4.1 Probability Theory

The main notion of probability theory is probability distributions. A probability distribution p is defined over a universe of discourse Ω such that:

$$\forall \omega_i \in \Omega, p(\omega_i) \in [0, 1]$$

A probability measure $P : 2^{\Omega} \to [0, 1]$ is a function that assigns to each event $A \subseteq \Omega$ a degree that belongs to the unit interval $[0, 1]$. It reflects the likelihood of the occurrence of an event A. A probability measure and probability distribution must satisfy the following requirements:

- **Positivity:** $\forall A \subseteq \Omega, P(A) \geq 0$, i.e., an event might be impossible in the worst case.
- **Normalisation:** $P(\Omega) = 1$, i.e., the universe of discourse is certain.
- **Additivity:** $P(A \cup B) = P(A) + P(B)$ if A and B are disjoint ($A \cap B = \emptyset$), i.e., the probability of the union of two disjoint events is equal to the sum of their individual probabilities.

In addition, the following properties follow from the above requirements:

- $\sum_{\omega \in \Omega} p(\omega) = 1$.
- $p(\emptyset) = 0$.
- $P(A) = 1 - P(\bar{A})$, A and $\bar{A}$ are complementary in Ω, i.e., $\bar{A} = \Omega \setminus A$.
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$ where $A \subseteq \Omega$ and $B \subseteq \Omega$.

Let B be an event of Ω. The probability of the realisation of an event A can be updated according to the probability of the realisation of the event B. Let $P(A \mid B)$ denote the probability of the event A knowing the event B. The transformation from $P(A)$ to $P(A \mid B)$ is called probabilistic conditioning and is computed as follows:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

A consequence of the conditioning rule is the product rule property, which is given by:

$$P(A \cap B) = P(A) * P(B \mid A) = P(B) * P(A \mid B)$$

Bayes rule allows us to represent how beliefs are updated in light of new evidence. It provides a way to compute the posterior probability of a hypothesis A given evidence B, by reversing the direction of conditional probabilities, formally:

$$P(A \mid B) = \frac{P(B \mid A) * P(A)}{P(B)}$$

Bayes rule is mostly used in classification problems and learning from statistical data.

The auto-duality of probabilities allows determining the probability of the complement of an event A, namely $P(\bar{A})$ thanks to the property $P(A) = 1 - P(\bar{A})$. However, it restricts expressiveness, since ignorance cannot be represented directly. This is the case especially when there is a need to make a distinction between the lack of belief in a fact and the belief in its negation. This property imposes a kind of closed-world assumption when modelling knowledge. This is not the case of non-dual systems, such as Dempster–Shafer theory [Dempster 1967, Shafer 1976, Frittella *et al.* 2020] or possibility theory [Denœux *et al.* 2020, Dubois & Prade 1988], where belief and disbelief are not always complements of each other.

1.4.2 Possibility Theory

Possibility theory has been a long-standing approach to reasoning with uncertain and inconsistent knowledge. It allows the representation of non-probabilistic information that can be vague, incomplete, or ambiguous [Dubois & Prade 2024, Dubois & Prade 2018]. The framework of possibility theory was introduced by Zadeh [Zadeh 1978, Zadeh 1988] and further developed by Dubois and Prade [Dubois & Prade 1988, Dubois & Prade 2024]. It relies on a possibility distribution, in addition to two distinct measures: possibility and necessity.

Possibility distribution

Let Ω be a finite set of possible worlds (or interpretations). A *possibility distribution* is a function $\pi : \Omega \to [0, 1]$ that assigns to each $\omega \in \Omega$ a degree of plausibility, called a *possibility degree*, which represents the compatibility of ω with the available knowledge encoded by π. By convention:

- $\pi(\omega) = 1$ means that ω is a *totally possible* state.
- $\pi(\omega) = 0$ means that ω is an *impossible* state.
- If $\pi(\omega) > \pi(\omega')$, then ω is considered *more plausible* or *more preferred* than ω'.

Possibility theory can express two extreme forms of knowledge:

1. *Total ignorance:* $\forall \omega \in \Omega, \pi(\omega) = 1$.
2. *Complete knowledge:* $\exists \omega_0 \in \Omega$ such that $\pi(\omega_0) = 1$ and for all $\omega \neq \omega_0$, $\pi(\omega) = 0$.

A possibility distribution π is said to be *normalised* if it admits at least one totally possible state, i.e., $\exists \omega \in \Omega$ such that $\pi(\omega) = 1$. Otherwise, it is called *sub-normalised*. The *inconsistency degree* of a sub-normalised possibility distribution π is defined by:

$$\mathsf{Inc}(\pi) = 1 - \max_{\omega \in \Omega} \pi(\omega)$$

Sub-normalisation reflects the presence of contradictions in the available knowledge encoded by the possibility distribution π.

Furthermore, possibility theory obeys the *principle of minimal specificity*, which states that any hypothesis not known to be impossible should not be ruled out [Dubois & Prade 2014]. Given two possibility distributions π and π', we say that π is *more specific* than π' if and only if $\forall \omega \in \Omega, \pi(\omega) \leq \pi'(\omega)$.

Possibility measure

Given a possibility distribution π over Ω, the *possibility measure* Π is a set function defined for any event $A \subseteq \Omega$ as:

$$\Pi(A) = \max_{\omega \in A} \pi(\omega)$$

Intuitively, $\Pi(A)$ evaluates the degree to which A is *plausible* or *compatible* with the available knowledge. It expresses whether it is possible that at least one element in A could be true.

Typical cases:

- $\Pi(A) = 1$ and $\Pi(\bar{A}) = 0$: A is *certain*.
- $\Pi(A) = 1$ and $0 < \Pi(\bar{A}) < 1$: A is *somewhat certain*.
- $\Pi(A) = 1$ and $\Pi(\bar{A}) = 1$: there is *total ignorance* about A.
- $\Pi(A) > \Pi(B)$: A is *more plausible* than B.

Properties of Π when π is normalised:

- $\max(\Pi(A), \Pi(\bar{A})) = 1$
- $\Pi(A \cap B) \leq \min(\Pi(A), \Pi(B))$
- $\Pi(A \cup B) = \max(\Pi(A), \Pi(B))$ (maximality axiom)

The possibility measure is non-additive, in the sense where: $\Pi(A \cup B) = \max(\Pi(A), \Pi(B))$ and not $\Pi(A \cup B) = \Pi(A) + \Pi(B)$.

Necessity measure

The *necessity measure* N is the dual of the possibility measure. For any event $A \subseteq \Omega$, it is defined as:

$$N(A) = 1 - \Pi(\bar{A}) = \min_{\omega \notin A}(1 - \pi(\omega))$$

The value $N(A)$ represents the degree to which A is *certainly implied* based on available knowledge. In possibility theory, for A to be somewhat certain, all elements in $\bar{A}$ must be sufficiently impossible. Importantly, $N(A) \leq \Pi(A)$ always holds.

- $N(A) = 1$ and $N(\bar{A}) = 0$: A is *certain*.
- $N(A) \in]0, 1[$ and $N(\bar{A}) = 0$: A is *somewhat certain*.
- $N(A) = 0$ and $N(\bar{A}) = 0$: there is *total ignorance* about A.

Properties of N when π is normalised:

- $\min(N(A), N(\bar{A})) = 0$
- $N(A \cap B) = \min(N(A), N(B))$
- $N(A \cup B) \geq \max(N(A), N(B))$

The use of possibility and necessity measures allows us to represent either certainty, partial ignorance, or total ignorance. The latter is achieved in the case where:

$$\Pi(A) = 1 \text{ and } \Pi(\bar{A}) = 1$$

The possibility measure $\Pi(A)$ evaluates the degree to which A is consistent with the available information (i.e., at least one element of A is plausible), while the necessity measure $N(A)$ evaluates the degree to which A is certainly implied by the available information.

Possibility theory is very suitable when measures indicate an ordering of plausibility rather than precise degrees. This ordinal scaling is one of the possible interpretations of possibility theory. It can also encode levels of surprise, the feasibility of an event, or a degree of compatibility. Such interpretations are based on the chosen representation of uncertainty. In essence, uncertainty can be represented either in extension, using possibility distributions as given above, or in a compact way using graphical models, such as Bayesian networks [Benferhat & Garcia 2002] and Credal networks [Cozman 2000]. The compact way can also be expressed in a weighted logic fashion by assigning weights to statements.

Quantitative and qualitative aspects of possibility theory

The set of weights used in the compact representation is seen as an uncertainty scale, which is often the unit interval $[0, 1]$. The compact representation can be defined either qualitatively or quantitatively. The quantitative and qualitative settings align with the notions of product-based and min-based possibility theory, respectively [Dubois & Prade 1998]. The difference between them lies in the interpretation of the uncertainty scale. The qualitative representation is appropriate when the uncertainty scale only encodes an order of plausibility between information pieces. The quantitative representation is suitable when the uncertainty degree represents a level of surprise in the sense of Spohn's OCF [Spohn 1988, Spohn 2014] or results from the transformation of a probability distribution into a possibility distribution [Dubois *et al.* 1993].

Quantitative representation. In the *quantitative* (or *numerical*) setting of possibility theory, the uncertainty is modelled using a numerical *possibility distribution* $\pi : \Omega \to [0, 1]$, as previously described. This distribution provides a graded plausibility scale for each possible world $\omega \in \Omega$, with the associated possibility and necessity measures Π and N defined over subsets of Ω. The values of $\pi(\omega)$ are typically interpreted in terms of degrees of compatibility with available knowledge or evidence, and the numerical scale allows for fine distinctions in plausibility. This representation is useful in contexts where such degrees can be assessed or estimated numerically, such as expert systems, sensor fusion, or approximate reasoning. The quantitative framework is especially powerful when integrated with fuzzy logic, enabling reasoning with vague predicates and continuous domains [Dubois & Prade 1998].

Qualitative representation. In the *qualitative* (or *ordinal*) setting of possibility theory, the focus is not on numerical degrees but on the ranking or ordering of possible worlds based on their plausibility. A qualitative possibility distribution is given by a *total preorder* $\succeq$ on Ω, where

$\omega \succeq \omega'$ means that ω is at least as plausible as ω' [Dubois & Prade 1998]. Equivalently, the qualitative setting can be captured by assigning to each world a rank or level from an ordered finite scale (e.g. $\{0, 1, \ldots, n\}$ or a linguistic scale like *impossible*, *unlikely*, *possible*, *very possible*, etc.). This representation is particularly appropriate when the uncertainty scale only encodes a degree of plausibility and when the actual numerical values do not really matter [Dubois & Prade 2014]. It is also suitable for common sense reasoning, default reasoning, or reasoning with partial or symbolic information when exact numerical values are not available or meaningful. In this context, possibility and necessity measures can be interpreted in ordinal terms, with $\Pi(A)$ reflecting the most plausible world in A, and $N(A)$ indicating that all counterexamples to A are strictly less plausible than the least plausible world in A.

1.4.3 Possibility Theory applied to Propositional Logic

Possibilistic logic is the logical counterpart of possibility theory. It is a framework that is very close to classical propositional logic, where propositions are assigned weights from the unit interval $[0, 1]$. These weights are interpreted as *certainty degrees*, representing the degree of belief or plausibility in the truth of a formula. The framework allows for reasoning about uncertain, inconsistent, and incomplete information, and it is used to model the knowledge in a way that respects the underlying possibility distribution.

Syntax

The syntax of possibilistic logic is built on top of classical propositional logic. Let $\mathcal{L}$ be a propositional language consisting of a set of propositional variables $\phi_1, \phi_2, \ldots, \phi_n$, logical connectives such as negation ($\neg$), conjunction ($\wedge$), disjunction ($\vee$), implication ($\rightarrow$), and parentheses for grouping formulas. A formula in the language $\mathcal{L}$ is called a *propositional formula*.

A possibilistic formula is a pair (ϕ, α), where ϕ is a propositional formula, and $\alpha \in [0, 1]$ is its *certainty degree*. The certainty degree α represents the degree of belief in the truth of the formula ϕ. Formally, the set of possibilistic formulas is defined as:

$$\mathcal{B} = \{(\phi_1, \alpha_1), (\phi_2, \alpha_2), \ldots, (\phi_n, \alpha_n)\}$$

where each pair (ϕ_i, α_i) is a propositional formula ϕ_i with an associated certainty degree α_i.

In propositional logic, we typically have formulas that are either true or false, but in possibilistic logic, we assign a weight to each formula to reflect how plausible or certain the formula is. This extension allows for reasoning under uncertainty, where formulas can be considered true with varying degrees of confidence.

Semantics

The semantics of a possibilistic knowledge base $\mathcal{B}$ is given in terms of a possibility distribution $\pi_{\mathcal{B}}$, which is a function $\pi_{\mathcal{B}} : \Omega \rightarrow [0, 1]$ that assigns to each interpretation (or possible world) in the set of propositional interpretations $\Omega = \{\omega_1, \ldots, \omega_m\}$ a degree of plausibility. This possibility distribution $\pi_{\mathcal{B}}$ captures the available knowledge, with $\pi_{\mathcal{B}}(\omega)$ representing the plausibility that ω is a valid world (i.e., the degree of compatibility of ω with the knowledge encoded in $\mathcal{B}$).

The possibility degree of an interpretation depends on the maximum weight of formulas falsified by that interpretation. Formally, for each interpretation $\omega \in \Omega$, its *possibility degree* is

given by:

$$\forall \omega \in \Omega, \pi_{\mathcal{B}}(\omega) = \begin{Bmatrix} 1 & \text{if } \forall (\phi_i, \alpha_i) \in \mathcal{B}, \omega \models \phi_i, \\ 1 - \max\{\alpha_i : (\phi_i, \alpha_i) \in \mathcal{B}, \omega \not\models \phi_i\} & \text{otherwise} \end{Bmatrix}$$

In classical propositional logic settings, consistency is given in terms of models; in essence, a knowledge base is considered consistent if it admits at least one model. Therefore, interpretations are distinguished into models or countermodels. Possibilistic logic is not limited to this binary model/countermodel treatment, as the possibility distribution assigns to the countermodels a degree of compatibility with the available knowledge. As a result, one can associate a *degree of inconsistency* varying between 0 and 1 to a possibilistic knowledge base.

Given a possibilistic knowledge base $\mathcal{B} = \{(\phi_i, \alpha_i) : i = 1, \ldots, n\}$, the degree of inconsistency of $\mathcal{B}$, denoted by $\mathsf{Inc}(\mathcal{B})$ can be given semantically or syntactically:

- Semantically, based on the possibility distribution associated with the interpretations: $\mathsf{Inc}(\mathcal{B}) = 1 - \max_{\omega \in \Omega} \pi_{\mathcal{B}}(\omega)$.
- Syntactically, using the notion of β-cut: $\max\{\alpha_i : \mathcal{B}_{\geq \beta} \text{ is inconsistent}\}$, where $\mathcal{B}_{\geq \beta}$ is called the β-cut of $\mathcal{B}$ and is the subset of $\mathcal{B}$ composed of formulas having weights greater than or equal to β.

Note that if $\mathcal{B}$ is consistent, which means that there is at least one interpretation ω that satisfies all the formulas in $\mathcal{B}$, $\pi_{\mathcal{B}}(\omega) = 1$, hence $\mathsf{Inc}(\mathcal{B}) = 0$. This corresponds to the case where the 0-cut, namely, $\mathcal{B}_{\geq 0}$, is consistent.

An inconsistency degree measures to what extent a knowledge base is inconsistent, with a higher value indicating more inconsistency. Possibilistic logic allows inference from an inconsistent knowledge base using an inconsistency degree.

As for the inconsistency degree, inference from a possibilistic knowledge base can be given in terms of semantics and syntax. In this setting, a distinction is made between inferring a formula as a plausible consequence and inferring it as a possibilistic consequence, where in the second case inference is made with a certainty degree. Semantically:

- A formula ϕ_i is a logical consequence of a possibilistic knowledge base $\mathcal{B}$, denoted by $\pi_{\mathcal{B}} \vdash \phi_i$ if and only if $N(\phi_i) > 0$, where $N(\phi_i)$ is the necessity degree of ϕ_i computed from the possibility distribution $\pi_{\mathcal{B}}$.
- A formula ϕ_i is a logical consequence of a possibilistic knowledge base $\mathcal{B}$ with a certainty degree α_i, denoted by $\pi_{\mathcal{B}} \vdash (\phi_i, \alpha_i)$ if and only if $N(\phi_i) \geq \alpha_i > 0$, where $N(\phi_i)$ is the necessity degree of ϕ_i computed from $\pi_{\mathcal{B}}$.

The same inferences can be given syntactically:

- A formula ϕ_i is a plausible conclusion of $\mathcal{B}$, denoted by $\mathcal{B} \vdash_p \phi_i$ if and only if $\mathcal{B}_{>\mathsf{Inc}(\mathcal{B})} \vdash \phi_i$.
- A formula ϕ_i is a possibilistic conclusion of $\mathcal{B}$, denoted by $\mathcal{B} \vdash_\pi (\phi_i, \alpha_i)$, if and only if $\mathcal{B}_{\geq \alpha_i}$ is consistent, $\mathcal{B}_{\geq \alpha_i} \vdash \phi_i$ and $\forall \alpha_j > \alpha_i$, $\mathcal{B}_{\geq \alpha_j} \nvdash \phi_i$.

Figure 1.2 illustrates how inference from a possibilistic knowledge base is done both in the semantic and the syntactic ways. In addition, as in propositional logic, inference from possibilistic knowledge bases can be performed using refutation. The plausible and possibilistic entailments can also be done by computing the inconsistency degree of the possibilistic knowledge base $\mathcal{B}$:

- $\mathcal{B} \vdash_p \phi_i$ if and only if $\mathsf{Inc}(\mathcal{B} \cup \{(\neg\phi_i, 1)\}) > \mathsf{Inc}(\mathcal{B})$.
- $\mathcal{B} \vdash_\pi (\phi_i, \alpha_i)$ if and only if $\mathsf{Inc}(\mathcal{B} \cup \{(\neg\phi_i, 1)\}) > \alpha_i$.

In summary, the semantics of possibilistic logic is grounded in possibility theory, where each interpretation is assigned a degree of compatibility with the knowledge base. Possibilistic logic allows for reasoning from inconsistent knowledge bases by deriving only conclusions that are based on formulas that are more plausible than the inconsistency degree of the knowledge base.

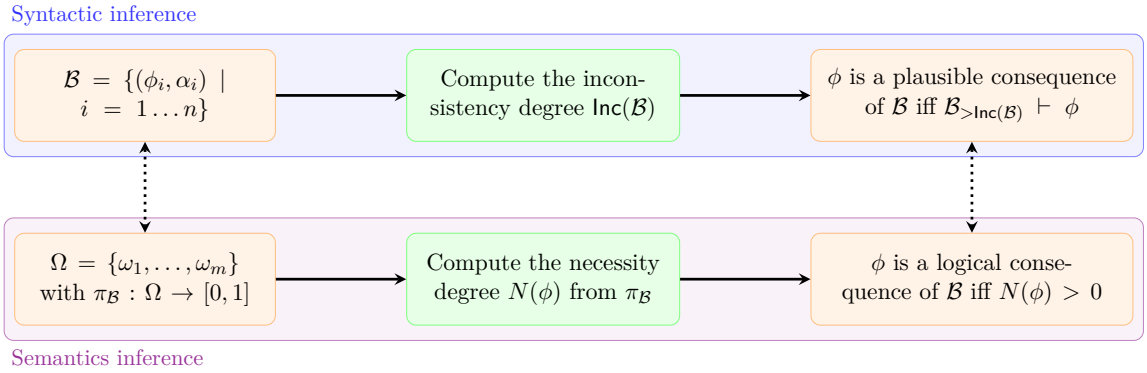


Figure 1.2: Syntactic vs. Semantic Inference in Possibilistic Logic.

1.4.4 Possibility Theory applied to DL-Lite

Description Logics have been extended with possibility theory since their early days [Hollunder 1995, Dubois *et al.* 2006]. Possibilistic DLs support uncertain reasoning and inference from a knowledge base. On their side, lightweight knowledge bases have also been extended with possibility theory [Benferhat & Bouraoui 2017]. Since the focus of this thesis is on DL-Lite$_\mathcal{R}$ knowledge bases, we briefly recall how possibilistic DL-Lite$_\mathcal{R}$ knowledge bases are defined, along with the notions of possibility distribution and inconsistency degree induced by them.

Similarly to standard possibilistic logic, possibilistic DL-Lite$_\mathcal{R}$ assigns weights from the unit interval $[0, 1]$ to a knowledge base's axioms and assertions to express their relative certainty. As mentioned in section 1.2.3, we assume that the TBox is coherent and stable [Lembo *et al.* 2010, Bienvenu & Bourgaux 2017, Baget *et al.* 2016]. Therefore, in the possibilistic setting, the TBox axioms are given the highest possible priority and considered as fully certain. Furthermore, this priority is not explicitly represented in the TBox. In contrast, ABox assertions are given explicit weights as follows:

Definition 1.21 (Possibilistic KB)**.** *A possibilistic DL-Lite$_\mathcal{R}$ knowledge base is a weighted knowledge base defined as a tuple $\mathcal{K}_\geq = \langle \mathcal{T}, \mathcal{A}_\geq \rangle$ where:*

- *$\mathcal{T}$ is a standard DL-Lite$_\mathcal{R}$ TBox in which all the TBox axioms are fully certain.*
- *$\mathcal{A}_\geq$ is a weighted ABox where each assertion is assigned a weight from the unit interval $]0, 1]$, such that:*

$$\mathcal{A}_\geq = \{(\varphi, \alpha) \mid \varphi \text{ is a DL-Lite}_\mathcal{R} \text{ assertion and } \alpha \in]0, 1]\}$$

Note that the ABox assertions with the weight $\alpha = 1$ are also considered fully certain, the assertions with a higher weight are more certain than those with a lower weight, and the assertions with the weight $\alpha = 0$ are ignored and not explicitly stated in the ABox.

Example 1.26. *Since possibilistic DL-Lite$_\mathcal{R}$ knowledge bases are also prioritised knowledge bases, let us use the same partitioning of the ABox $\mathcal{A}_\geq$ from Example 1.24 and present it as a weighted ABox by assigning a degree from the unit interval $]0,1]$ to each assertion:*

$\mathcal{A}_\geq =$

Assertion	weight
$\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$	0.8
$\mathsf{FullResearcher}(\mathsf{Ben})$	0.6
$\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)$	0.4
$\mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)$	0.4
$\mathsf{Secondee}(\mathsf{Duane})$	0.2
$\mathsf{Confidential}(\mathsf{r}_1)$	0.2

The notion of possibility distribution has also been adapted to cover possibilistic DL-Lite$_\mathcal{R}$ interpretations [Benferhat & Bouraoui 2017]. In this case, the universe of discourse Ω as given in possibility theory is the set of all standard DL-Lite$_\mathcal{R}$ interpretations of a knowledge base, where each interpretation $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle \in \Omega$. Similarly to standard possibilistic logic, a possibility distribution is a function, denoted by $\pi_{\mathcal{K}_\geq}$, that maps the universe of discourse to the unit interval $[0,1]$. A possibility degree $\pi_{\mathcal{K}_\geq}(\mathcal{I}) \in [0,1]$ is assigned to each interpretation $\mathcal{I} \in \Omega$, reflecting how compatible is the interpretation with the available knowledge. In the sense of standard DL-Lite$_\mathcal{R}$ semantics, and following the assumption of a coherent and stable TBox, if an interpretation $\mathcal{I}$ is not a model of the TBox $\mathcal{T}$, then $\mathcal{I}$ is impossible and is given the possibility degree $\pi_{\mathcal{K}_\geq} = 0$. An interpretation is also impossible if it falsifies some fully certain assertion (assigned the weight $\alpha = 1$) in the ABox. Now, if an interpretation $\mathcal{I}$ is a model of the TBox $\mathcal{T}$ and falsifies some uncertain assertions of the ABox, it is given a possibility degree according to the most important assertions that it falsifies. As a result, interpretations can be ranked based on their possibility degrees, such that for two interpretations $\mathcal{I}$ and $\mathcal{I}'$, $\mathcal{I}$ is said to be more consistent than $\mathcal{I}'$ if $\pi_{\mathcal{K}_\geq}(\mathcal{I}) > \pi_{\mathcal{K}_\geq}(\mathcal{I}')$.

Definition 1.22 (DL-Lite$_\mathcal{R}$ possibility distribution)**.** *Let $\mathcal{K}_\geq = \langle \mathcal{T}, \mathcal{A}_\geq \rangle$ be a possibilistic DL-Lite$_\mathcal{R}$ knowledge base. The possibility distribution induced by $\mathcal{K}_\geq$ is given by:*

$$\pi_{\mathcal{K}_\geq} = \begin{cases} 1 & \text{if } \forall \alpha \in \mathcal{T}, \mathcal{I} \models \alpha \text{ and } \forall (\varphi_i, \alpha_i) \in \mathcal{A}_\geq, \mathcal{I} \models \varphi_i \\ 0 & \text{if } \exists \alpha \in \mathcal{T}, \text{ such that } \mathcal{I} \not\models \alpha \\ \min\{(1-\alpha_i) \mid (\varphi_i, \alpha_i) \in \mathcal{A}_\geq, \mathcal{I} \not\models \varphi_i\} & \text{Otherwise} \end{cases}$$

If $\pi_{\mathcal{K}_\geq} = 1$, then $\mathcal{I}$ is the most preferred (or fully possible) interpretation, this reflects the consistency of a possibilistic knowledge base. A possibility distribution $\pi_{\mathcal{K}_\geq}$ is said to be *normalized* if it admits a fully possible interpretation. Otherwise, it is said to be *sub-normalised*. In such case, the normalisation degree of $\pi_{\mathcal{K}_\geq}$ is given by:

$$h(\pi_{\mathcal{K}_\geq}) = \max_{\mathcal{I} \in \Omega} \pi_{\mathcal{K}_\geq}(\mathcal{I})$$

Note that since the main interest of representing uncertainty in DL-Lite$_\mathcal{R}$ lies in capturing the priority levels between the different elements of a knowledge base, the representation of possibilistic DL-Lite$_\mathcal{R}$ given so far aligns with a min-based form of possibility theory, where the uncertainty scale solely encodes an order of plausibility between the ABox assertions. As a result, instead of assigning weights from the unit interval $]0,1]$ to the ABox assertions of a

possibilistic DL-Lite$_\mathcal{R}$ knowledge base, a total preorder relation can be defined over the ABox. Such a preorder induces a stratified ABox as given in Section 1.3.3, where the ABox is given as a well-ordered partition of the form $(\mathcal{S}_1, \ldots, \mathcal{S}_n)$, where the assertions in $\mathcal{S}_1$ are the most certain, those in $\mathcal{S}_n$ are the least certain in the ABox, and the assertions in any stratum $\mathcal{S}_i$ $(i = 1, \ldots, n)$ are equally certain. Partitioning the ABox into strata is in full agreement with qualitative possibility theory.

Possibilistic inference from DL-Lite$_\mathcal{R}$ knowledge bases

The notion of inconsistency degree has also been adapted to allow inference from inconsistent possibilistic DL-Lite$_\mathcal{R}$ knowledge bases. Since only the assertions of the ABox are assigned weights in a knowledge base, the notion of β-cut is defined in the same way such that:

- The β-cut of $\mathcal{A}_\geq$ is: $\mathcal{A}^{\geq\beta} = \{\varphi \mid (\varphi, \alpha) \in \mathcal{A}_\geq, \alpha \geq \beta\}$.
- The strict β-cut of $\mathcal{A}_\geq$ is: $\mathcal{A}^{>\beta} = \{\varphi \mid (\varphi, \alpha) \in \mathcal{A}_\geq, \alpha > \beta\}$.

As a result, the inconsistency degree of $\mathcal{A}_\geq$ is defined formally by:

$$\mathsf{Inc}(\mathcal{A}_\geq) = \begin{Bmatrix} 0 & \text{if } \langle \mathcal{T}, \mathcal{A} \rangle \text{ is consistent} \\ \beta & \text{if } \langle \mathcal{T}, \mathcal{A}^{\geq\beta} \rangle \text{ is inconsistent and } \langle \mathcal{T}, \mathcal{A}^{>\beta} \rangle \text{ is consistent} \end{Bmatrix}$$

Example 1.27. *The inconsistency degree of the ABox $\mathcal{A}_\geq$ is:*

$$\mathsf{Inc}(\mathcal{A}_\geq) = 0.4$$

because $\langle \mathcal{T}, \mathcal{A}^{\geq 0.4} \rangle$ is inconsistent and $\langle \mathcal{T}, \mathcal{A}^{>0.4} \rangle$ is consistent. Where:

$$\mathcal{A}^{\geq 0.4} = \begin{Bmatrix} \mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Ben}), \\ \mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1) \end{Bmatrix}$$

and:

$$\mathcal{A}^{>0.4} = \{\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Ben})\}$$

The possibilistic inference from DL-Lite$_\mathcal{R}$ knowledge bases is given as follows:

Definition 1.23 (Possibilistic inference)**.** *An assertion φ is a plausible consequence of a possibilistic knowledge base $\mathcal{K}_\geq = \langle \mathcal{T}, \mathcal{A}_\geq \rangle$, denoted by $\mathcal{K}_\geq \models^p \varphi$, if:*

$$\langle \mathcal{T}, \mathcal{A}^{>\mathsf{Inc}(\mathcal{A}_\geq)} \rangle \models \varphi$$

The inconsistency degree also allows computing a possibilistic repair of a knowledge base where only the assertions with a certainty degree that is strictly higher than the inconsistency degree are included in the possibilistic repair.

Definition 1.24 (Possibilistic repair)**.** *The possibilistic repair of $\mathcal{A}_\geq$, denoted by $\mathcal{R}_\pi(\mathcal{A}_\geq)$, is:*

$$\mathcal{R}_\pi(\mathcal{A}_\geq) = \{\varphi \mid (\varphi, \alpha) \in \mathcal{A}_\geq, \alpha > \mathsf{Inc}(\mathcal{A}_\geq)\}$$

Note that all the assertions contained in $\mathcal{R}_\pi(\mathcal{A}_\geq)$ have a priority degree that is strictly higher than $\mathsf{Inc}(\mathcal{A}_\geq)$. Hence, by definition, $\mathcal{R}_\pi(\mathcal{A}_\geq)$ is consistent with respect to $\mathcal{T}$. Figure 1.3 illustrates the possibilistic repair of a DL-Lite ABox.

Example 1.28. *The possibilistic repair of the ABox $\mathcal{A}_{\geq}$ of our running example is:*

$$\mathcal{R}_{\pi}(\mathcal{A}_{\geq}) = \{\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Ben})\}$$

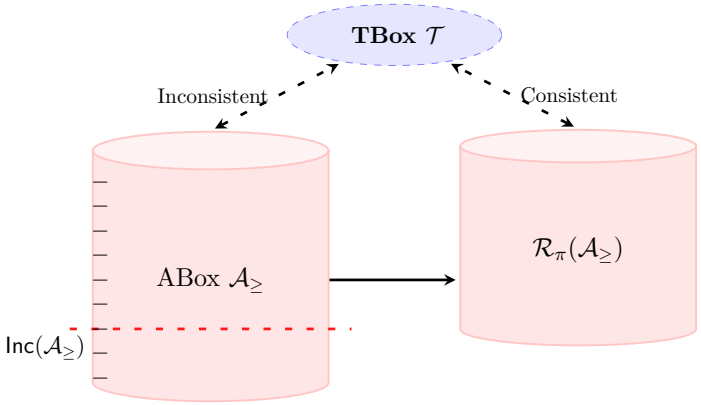


Figure 1.3: The possibilistic repair $\mathcal{R}_{\pi}(\mathcal{A}_{\geq})$.

The possibilistic semantics for weighted knowledge bases return a single repair. In practice, inference from a fixed known number of repairs is simpler. Often a single repair is considered, as we have seen in the case of the IAR and the brave semantics. Inference from other semantics with arbitrary numbers of repairs such as the AR and the preferred repair semantics is usually computationally expensive.

1.5 Conclusion

In this chapter, we provide a comprehensive overview of knowledge representation and reasoning frameworks, with a special focus on DL-Lite knowledge bases, which offer interesting computational properties when it comes to knowledge base satisfiability and query answering. In addition, we discuss some of the well-known inconsistency-tolerant semantics, which serve as a solution to reasoning in the presence of inconsistent knowledge bases. Finally, we give an insight into the particular framework of possibilistic logic for reasoning with uncertain information.

An important question that arises here concerns the possible ways of going beyond existing inconsistency-tolerant semantics, both on the representational and the computational levels. Indeed, discussed methods such as the preferred repair semantics, the lexicographic preferred-based repair, and the possibilistic repair allow inference from inconsistent and totally ordered knowledge bases. However, assuming a total order relation on data is often unrealistic, mainly due to incomplete order relations or to data originating from multiple sources. Hence, a need emerges for methods that can handle inconsistency in partially ordered knowledge bases. Such ordering relations allow for a more flexible and general way of representing preferences within data. The next chapter addresses this concern by providing a novel method for dealing with inconsistency in partially ordered knowledge bases.

Chapter 2

The Closure-Based Partially Ordered Possibilistic Repair

Contents

2.1 Partially Ordered Possibilistic Knowledge Bases 54
2.1.1 Total Order Extensions of a Partial Order and Compatible ABoxes . . . 56
2.1.2 The Partially Ordered Possibilistic Repair 57
2.1.3 The π-accepted Assertions 59
2.2 A Closure-Based Partially Ordered Possibilistic Repair 60
2.2.1 The Positive Deductive Closure 60
2.2.2 Closure-Based ABox Repairs 62
2.2.3 The Closure-Based Partially Ordered Possibilistic Repair 65
2.2.4 Equivalent Characterisation: the Dominant Supports 69
2.3 Algorithmic and Experimental Evaluation 76
2.3.1 The Cπ-repair Algorithm 76
2.3.2 Revisiting the Cπ-repair Algorithm 82
2.3.3 Experimental Evaluation 87
2.4 Conclusion 91

Dealing with inconsistent information is a challenging task, in particular, when it comes to finding a good balance between the expressive power of the representation language and the computational complexity of the proposed semantics. As shown in the first chapter, several semantics target datasets that are unordered (i.e., flat) or totally ordered (i.e., prioritised). However, information as found on the web, especially in the case of semantic web applications, often incorporates order relations that are incomplete or that stem from multiple sources with separate ranking criteria leading to partially ordered data.

Partial orders offer a more flexible way of prioritising information. They allow for handling information that cannot be fully ordered, mainly when parts of the order relation are missing or cannot be represented. They also serve as a general case for total orders, allowing thorough studies to expand the range of available semantics. Moreover, the literature does not fall short when it comes to methods for constructing partial orders when multiple ordering criteria apply to the information. For instance, Brewka's work [Brewka 1989] highlights how to obtain a partial order derived from total specificity relations but applied individually within different parts of a

dataset. In addition, the well-known Pareto principle does the same by enforcing that a piece of information must be preferred across all available criteria to be considered as more reliable, whereas other methods may suggest a less constraining combination by imposing a non-contested preference from at least a single criterion. However, the literature falls short when it comes to methods that are capable of tractably handling inconsistent partially ordered information. We find the Elect method [Belabbes *et al.* 2021], which extends the IAR semantics to the case of partially ordered ABoxes, the partially ordered possibilistic repair [Belabbes & Benferhat 2021], which extends the standard possibilistic repair, and the grounded repair [Bienvenu & Bourgaux 2020], which is inspired by argumentation frameworks [Dung 1995]. In addition, some work efforts considered partially ordered information in the propositional logic and possibilistic logic frameworks [Touazi *et al.* 2015, Yahi *et al.* 2008, Benferhat *et al.* 2004a]; however, tractability is not guaranteed in those cases. A major question that arises in such a situation is:

How to go beyond existing semantics while dealing with partially ordered information without causing additional computational costs?

In this chapter, we start by recalling the partially ordered possibilistic repair [Belabbes & Benferhat 2021], the notion of positive deductive closure, and some state-of-the-art deductive closure-based repairs. Then we address the major question by taking the following steps:

1. **The Closure-Based Partially Ordered Possibilistic Repair.** We propose a new closure-based inconsistency-tolerant semantics for partially ordered DL-Lite knowledge bases. The new semantics achieves a larger repair compared to the partially ordered possibilistic repair, thus enhancing its productivity, in addition to being in full agreement with a qualitative representation of possibility theory.

2. **Efficient Equivalent Characterisation.** We define a tractable characterisation equivalent with the introduced semantics thanks to the specification of the notions of support and dominance.

3. **Algorithmic and Experimental evaluation.** We provide an algorithmic study that revisits the steps of the proposed characterisation, taking advantage of some of its properties, before concluding the chapter with an experimental evaluation to showcase the utility of the discussed properties.

The main results of this chapter have been published in [Laouar *et al.* 2023, Laouar *et al.* 2025a].

2.1 Partially Ordered Possibilistic Knowledge Bases

A partial preorder over a knowledge base can be represented either by assigning symbolic weights to the different assertions in the knowledge base or by considering partially ordered partitions of the knowledge base. In both cases, the main interest is the order of plausibility between the assertions. The first case was discussed in several research efforts in the literature (e.g.[Dubois & Prade 2014, Belabbes & Benferhat 2021]), the idea is to consider a partially ordered set of weights, often referred to by a partially ordered set (or POSet), along with a strict partial order relation, in order to define a partially ordered uncertainty scale. In the second case, a partial preorder relation is associated to the ABox in order to represent the partially preordered ABox. The first contribution of this thesis is introduced using the second type of representation.

A partial preorder $\trianglerighteq$ over a DL-Lite$_\mathcal{R}$ knowledge base is a reflexive and transitive binary relation, for which some elements may be incomparable. Given two ABox assertions φ_i and φ_j, the relation $\varphi_i \trianglerighteq \varphi_j$ means that φ_i is at least as preferred (or as certain) as φ_j, where the preorder relation $\trianglerighteq$ is given by:

- The strict order associated with $\trianglerighteq$ is denoted by $\triangleright$ such that: $\varphi_i \triangleright \varphi_j$ if and only if $\varphi_i \trianglerighteq \varphi_j$ holds and $\varphi_j \trianglerighteq \varphi_i$ does not hold.
- The associated equivalence relation is denoted by $\equiv_\trianglerighteq$ such that: $\varphi_i \equiv_\trianglerighteq \varphi_j$ if and only if both $\varphi_i \trianglerighteq \varphi_j$ and $\varphi_j \trianglerighteq \varphi_i$ hold.
- If neither $\varphi_i \trianglerighteq \varphi_j$ nor $\varphi_j \trianglerighteq \varphi_i$ holds, we say that φ_i and φ_j are incomparable and denote it by $\varphi_i \bowtie \varphi_j$.

A standard DL-Lite$_\mathcal{R}$ ABox $\mathcal{A}$ when accompanied with a partial preorder $\trianglerighteq$ is denoted by $(\mathcal{A}, \trianglerighteq)$. For the sake of simplicity, we use the notation $\mathcal{A}_\trianglerighteq$ from now on. Recall that unlike ABox assertions, the TBox axioms of a DL-Lite$_\mathcal{R}$ knowledge base are considered fully certain and thus are given the highest possible preference degree. This preference is not explicitly represented in a knowledge base.

In the following, and unless otherwise stated, we consider a partially preordered DL-Lite$_\mathcal{R}$ knowledge base denoted by $\mathcal{K}_\trianglerighteq = \langle \mathcal{T}, \mathcal{A}_\trianglerighteq \rangle$, where $\mathcal{T}$ is a standard DL-Lite$_\mathcal{R}$ TBox and $\mathcal{A}_\trianglerighteq$ is the partially preordered ABox as given above.

Example 2.1. *Let us now introduce a partially preordered version of our main example, which was originally given in Examples 1.8 (TBox) and 1.11 (ABox). We start by recalling the TBox $\mathcal{T}$:*

$$\mathcal{T} = \left\{ \begin{array}{ll} 1.\ \mathsf{Secondee} \sqsubseteq \mathsf{User} & 7.\ \exists \mathsf{hasAccess}^- \sqsubseteq \mathsf{Resource} \\ 2.\ \mathsf{FullResearcher} \sqsubseteq \mathsf{User} & 8.\ \exists \mathsf{hasConfidAccess}^- \sqsubseteq \mathsf{Confidential} \\ 3.\ \mathsf{Confidential} \sqsubseteq \mathsf{Resource} & 9.\ \mathsf{hasConfidAccess} \sqsubseteq \neg\, \mathsf{hasOpenAccess} \\ 4.\ \mathsf{Secondee} \sqsubseteq \neg\, \mathsf{FullResearcher} & 10.\ \mathsf{owns} \sqsubseteq \mathsf{hasAccess} \\ 5.\ \exists \mathsf{hasAccess} \sqsubseteq \mathsf{User} & 11.\ \mathsf{hasOpenAccess} \sqsubseteq \mathsf{hasAccess} \\ 6.\ \exists \mathsf{owns} \sqsubseteq \mathsf{FullResearcher} & 12.\ \mathsf{hasConfidAccess} \sqsubseteq \mathsf{hasAccess} \end{array} \right\}$$

The assertions in our ABox are the following:

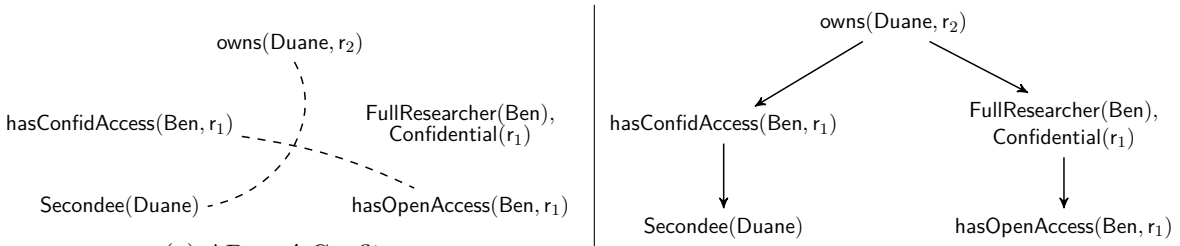
$$\mathcal{A} = \left\{ \begin{array}{l} \mathsf{FullResearcher}(\mathsf{Ben}) \\ \mathsf{Secondee}(\mathsf{Duane}) \\ \mathsf{Confidential}(\mathsf{r}_1) \\ \mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1) \\ \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1) \\ \mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2) \end{array} \right\}$$

Then, we consider the following partial preorder relation $\trianglerighteq$ over the assertions of our ABox:

- *The assertion* $\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$ *is more certain than any other assertion in* $\mathcal{A}$.
- $\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)$ *is more certain than* $\mathsf{Secondee}(\mathsf{Duane})$.

- *The assertions* FullResearcher(Ben) *and* Confidential(r_1) *are equally certain, and more certain than* hasOpenAccess(Ben, r_1).

Figure 2.1a depicts the conflict set of the ABox $\mathcal{A}$, where dashed lines (− − −) represent a conflict. Figure 2.1b depicts the partial preorder over the assertions of $\mathcal{A}_{\trianglerighteq}$, where straight arrows denote strict preference (⟶ denotes $\triangleright$).



(a) ABox $\mathcal{A}$ Conflicts

(b) Partially Preordered ABox $\mathcal{A}_{\trianglerighteq}$

Figure 2.1: The Partially Preordered ABox $\mathcal{A}_{\trianglerighteq}$

2.1.1 Total Order Extensions of a Partial Order and Compatible ABoxes

Making a decision out of an incomparability is not straightforward. Unlike strict preference or equivalence, it is not a consequence of known preference information, but rather a result of under-specification or incomplete knowledge. It often denotes the absence of order rather than neutrality. Therefore, in order to deal with situations of incomparability, one must avoid making arbitrary assumptions on the relation's nature.

A fundamental way of handling partially ordered information relies on the notion of total order extensions of a partial order. Such extensions preserve the strict ordering and the equivalence relations specified by the order relation, and extend incomparabilities in every possible way to cover every possible compatible total order. This notion has been used in several logical frameworks, such as standard possibilistic logic [Benferhat *et al.* 2004b], interval-based possibilistic logic [Benferhat *et al.* 2011], and Credal probabilistic networks [Cozman 2000].

Definition 2.1 (Total order extensions)**.** *Let $\trianglerighteq$ be a partial preorder relation over the assertions of a partially preordered DL-Lite$_{\mathcal{R}}$ ABox. $\forall \varphi_i, \varphi_j \in \mathcal{A}_{\trianglerighteq}$, a total preorder relation $\geq$, compatible with $\trianglerighteq$, is given by:*

- *if $\varphi_i \triangleright \varphi_j$, then $\varphi_i > \varphi_j$,*
- *if $\varphi_i \trianglerighteq \varphi_j$, then $\varphi_i \geq \varphi_j$, and;*
- *if $\varphi_i \bowtie \varphi_j$, then either $\varphi_i > \varphi_j$ or $\varphi_j > \varphi_i$ or $\varphi_i \equiv \varphi_j$.*

Where $\geq$ is a total preorder relation given between any two assertions in the ABox.
The set of all total preorders that are compatible with the partial preorder $\trianglerighteq$ is called a family of compatible orders and is denoted by $(\geq_1, \ldots, \geq_m)$, where m is the number of possible extensions.

The family of total preorder extensions of a partial preorder induces as many totally preordered ABoxes that are also said to be compatible with the partially preordered ABox $\trianglerighteq$.

Having m possible extensions, the resulting family of totally preordered ABoxes is denoted by:

$$\mathcal{A}_{\geq_1}, \ldots, \mathcal{A}_{\geq_m}$$

Note that the number of possible total preorder extensions is exponential (in the number of incomparabilities in a partial preorder) in the worst case.

Example 2.2. *Let us extend the partial preorder $\trianglerighteq$ over the assertions of $\mathcal{A}_{\trianglerighteq}$ from Example 2.1. One can observe the following:*

1. *any compatible total preorder $\geq_i$ returns the assertion* $\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$ *as the most certain assertion in $\mathcal{A}_{\geq_i}$, and;*
2. *extending the incomparability between* $\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)$ *and* $\mathsf{FullResearcher}(\mathsf{Ben})$, $\mathsf{Confidential}(\mathsf{r}_1)$, *results in the second most certain stratum in any $\mathcal{A}_{\geq_i}$ containing either the assertion* $\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)$, *the assertions* $\mathsf{FullResearcher}(\mathsf{Ben})$ *and* $\mathsf{Confidential}(\mathsf{r}_1)$, *or the three all together.*

As a result of the above observations, any total preorder extension $\geq_i$ would induce a totally preordered ABox that resembles one of the three extensions depicted in Figure 2.2, where straight arrows $\longrightarrow$ denote strict preference ($>$). To further show the results of the extensions, we detail all possible extensions (13) in a separate figure in the next page (Figure 2.3).

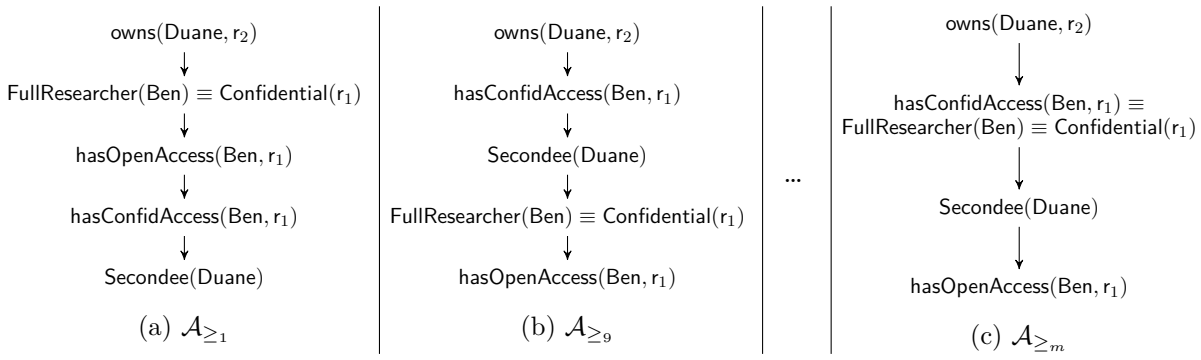


Figure 2.2: The ABoxes $\mathcal{A}_{\geq_1}, \ldots, \mathcal{A}_{\geq_m}$ compatible with $\mathcal{A}_{\trianglerighteq}$

So far, we have discussed the standard way of handling partially ordered information. In the next section, we recall one of the semantics that relies on this extension to produce a repair for inconsistent partially preordered possibilistic DL-Lite$_{\mathcal{R}}$ ABoxes.

2.1.2 The Partially Ordered Possibilistic Repair

The partially preordered possibilistic repair, also called the π-repair, was introduced to handle the case of partially preordered ABoxes using a possibility theory inference-based semantics [Belabbes & Benferhat 2021]. It mainly extends the possibilistic repair (Definition 1.24) that is unable to handle partially preordered information, using the notion of compatible total preorders.

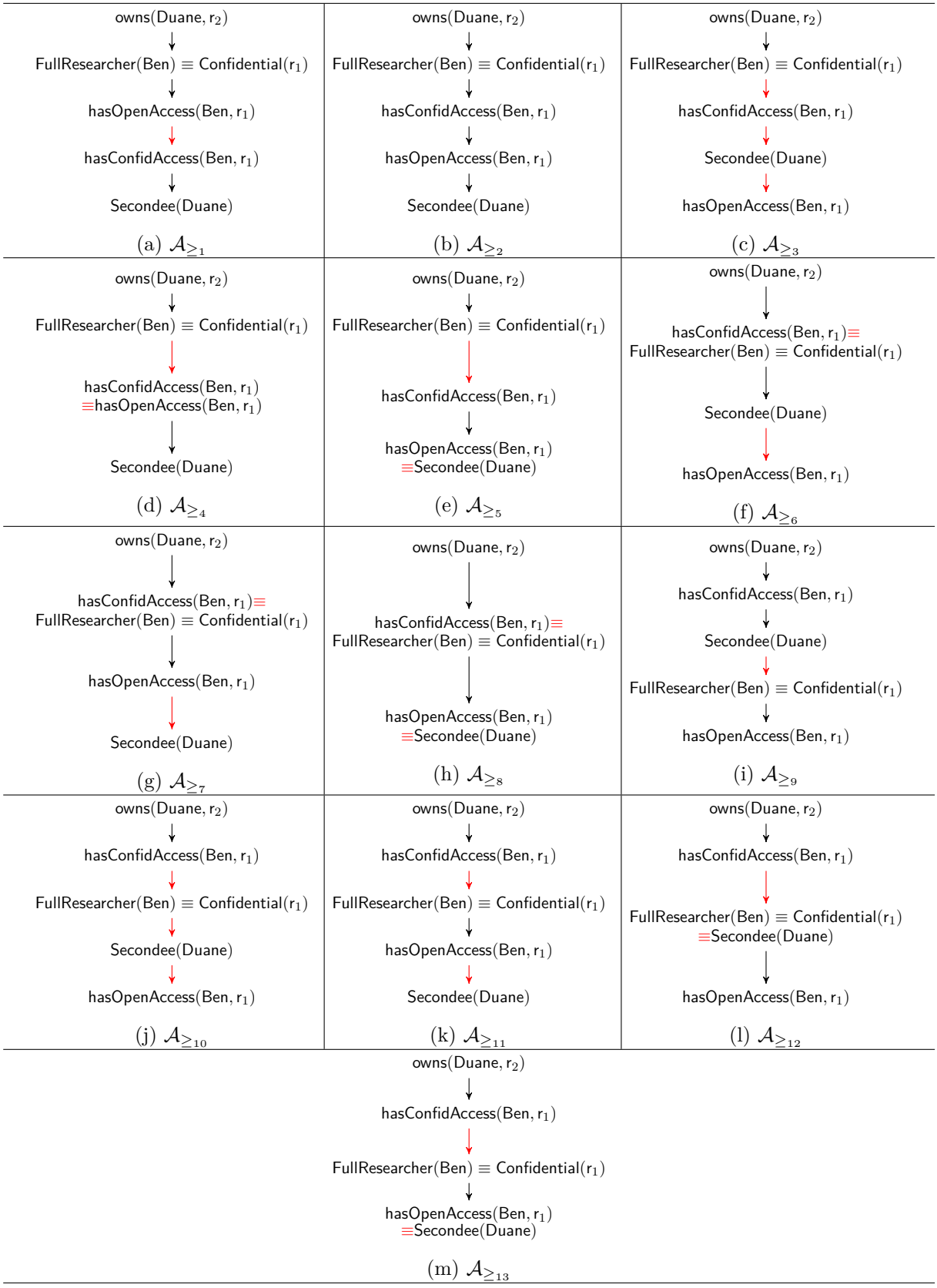


Figure 2.3: The total preorder extensions $\geq_1, \ldots, \geq_{13}$ of $\trianglerighteq$.

It proceeds by extending the partial preorder into a family of compatible total preorders, as given in Definition 2.1. Then, it computes the possibilistic repair of each resulting totally preordered compatible ABox, before intersecting all obtained repairs to get a single π-repair.

Definition 2.2 (Partially preordered π-repair)**.** *The possibilistic repair for a partially preordered ABox, denoted $\pi(\mathcal{A}_{\trianglerighteq})$, is computed as follows:*

1. *First, extend the partial preorder $\trianglerighteq$ into a family of m total preorders, denoted by $\geq_i$, for $i = 1, \ldots, m$. Then each extension $\geq_i$ defines an ABox $\mathcal{A}_{\geq_i}$ that is compatible with $\mathcal{A}_{\trianglerighteq}$.*
2. *For each compatible base $\mathcal{A}_{\geq_i}$, compute the totally preordered possibilistic repair $\mathcal{R}_\pi(\mathcal{A}_{\geq_i})$, given by Definition 1.24.*
3. *Finally, intersect the repairs $\mathcal{R}_\pi(\mathcal{A}_{\geq_i})$ to obtain the partially preordered possibilistic repair: $\pi(\mathcal{A}_{\trianglerighteq}) = \bigcap_{i=1}^{m} \mathcal{R}_\pi(\mathcal{A}_{\geq_i})$.*

One can see that the definition given above provides a naive way of computing the π-repair. This is due to the reliance on the family of total preorder extensions, which can be exponential in the worst case.

2.1.3 The π-accepted Assertions

The repair $\pi(\mathcal{A}_{\trianglerighteq})$ has been characterised tractably to avoid exhibiting the total preorder extensions $(\geq_1, \ldots, \geq_m)$ of $\trianglerighteq$, using the notion of π-accepted assertions [Belabbes & Benferhat 2021]. The characterisation relies on the conflict set of an ABox, as given in Definition 1.9. Intuitively, an assertion is π-accepted if it is strictly more certain than at least one assertion from each conflict. Note that the conflict set of a partially preordered ABox, which is denoted by $\mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$ can be computed in a similar fashion to the case of a flat DL-Lite$_\mathcal{R}$ ABox.

Definition 2.3 (π-accepted assertion)**.** *An assertion $\varphi_j \in \mathcal{A}_{\trianglerighteq}$ is π-accepted if, for all $\mathcal{C} \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$, there is $\varphi_k \in \mathcal{C}, \varphi_j \neq \varphi_k$, such that $\varphi_j \rhd \varphi_k$.*

The possibilistic repair $\pi(\mathcal{A}_{\trianglerighteq})$ is the set of all the π-accepted assertions in $\mathcal{A}_{\trianglerighteq}$, and it can be computed in polynomial time in the size of $\mathcal{A}_{\trianglerighteq}$ in DL-Lite$_\mathcal{R}$ ontologies [Belabbes & Benferhat 2021].

Example 2.3. *Let us compute the π-accepted assertions of our knowledge base from Example 2.1. The conflict set of the ABox, namely $\mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$ contains the following conflicts:*

- $\mathcal{C}_1 = \{\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)\}$.
- $\mathcal{C}_2 = \{\mathsf{Secondee}(\mathsf{Duane}), \mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)\}$.

Checking the acceptance of the assertions in $\mathcal{A}_{\trianglerighteq}$ goes as follows:

- $\mathsf{FullResearcher}(\mathsf{Ben})$ *is strictly more certain than* $\mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1) \in \mathcal{C}_1$*, however, it is not more certain than any assertion in $\mathcal{C}_2$.*
- $\mathsf{Secondee}(\mathsf{Duane})$ *is not more certain than any assertion in $\mathcal{C}_1$ or in $\mathcal{C}_2$.*
- $\mathsf{Confidential}(\mathsf{r}_1)$ *is strictly more certain than* $\mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1) \in \mathcal{C}_1$*, however, it is not more certain than any assertion in $\mathcal{C}_2$.*
- $\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)$ *is strictly more certain than* $\mathsf{Secondee}(\mathsf{Duane}) \in \mathcal{C}_2$*, however, it is not more certain than any assertion in $\mathcal{C}_1$.*

- $\mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)$ *is not more certain than any assertion in* $\mathcal{C}_1$ *or in* $\mathcal{C}_2$.
- $\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$ *is more certain than both the assertions in* $\mathcal{C}_1$ *and both assertions in* $\mathcal{C}_2$.

Since an assertion must be at least more certain than one assertion from each conflict, the only π*-accepted assertion in* $\mathcal{A}_{\trianglerighteq}$ *is* $\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$. *Therefore, the* π*-repair of* $\mathcal{A}_{\trianglerighteq}$ *is given by:*

$$\pi(\mathcal{A}_{\trianglerighteq}) = \{\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)\}$$

2.2 A Closure-Based Partially Ordered Possibilistic Repair

In addition to the use of preference relations to guide the computation of repairs, another major lever that allows to improve repair methods is the deductive closure, which allows the enrichment of an ABox with the assertions that follow from it with respect to the TBox axioms. We start this section by recalling the notion of positive deductive closure in addition to some of the well-known closure-based repair semantics. Then, we introduce our closure-based repair semantics, which combines partially preordered possibilistic knowledge bases and the deductive closure.

In order to be able to assess the impact of applying the deductive closure in a repair process, we introduce the notion of productivity, which refers to an inconsistency-tolerant semantics ability to draw more conclusions than another semantics to which it is being compared.

Definition 2.4 (Productivity)**.** *Let* s_1 *and* s_2 *be two inconsistency-tolerant semantics. We say that* s_1 *is more productive than* s_2*, if for any knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ *and any query* q*, if* $\mathcal{K} \vDash^{s_2} q$*, then* $\mathcal{K} \vDash^{s_1} q$.
Where $\vDash^s$ *denotes the inference under a given inconsistency-tolerant semantics* s.

2.2.1 The Positive Deductive Closure

The positive inclusion axioms in a DL-Lite$_\mathcal{R}$ TBox allow the derivation of implicit information from the knowledge base. The positive deductive closure of an ABox is made up of all the assertions that can be inferred from the ABox using the positive inclusion axioms of the TBox.

Definition 2.5 (Closed ABox)**.** *Let* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ *be a DL-Lite*$_\mathcal{R}$ *knowledge base. Let* $\mathcal{T}_p$ *denote the set of all the positive inclusion axioms of* $\mathcal{T}$. *The deductive closure of* $\mathcal{A}$ *with respect to the TBox* $\mathcal{T}$ *is defined as follows:*

$$\begin{aligned} cl(\mathcal{A}) \ = \ & \{B(a) \mid \langle \mathcal{T}_p, \mathcal{A} \rangle \vDash B(a) \text{ such that } B \text{ is a concept name in } \mathcal{T}, \\ & a \text{ is an individual in } \mathcal{A}\} \ \cup \ \{R(a,b) \mid \langle \mathcal{T}_p, \mathcal{A} \rangle \vDash R(a,b) \text{ such that} \\ & R \text{ is a role name in } \mathcal{T}, \ a \text{ and } b \text{ are individuals in } \mathcal{A}\}. \end{aligned}$$

Where $\vDash$ *is the standard DL-Lite*$_\mathcal{R}$ *entailment relation.*

Example 2.4. *Let us compute the deductive closure of our example's ABox* $\mathcal{A}$ *with respect to the TBox* $\mathcal{T}$. *The set of positive inclusion axioms of* $\mathcal{T}$, *denoted by* $\mathcal{T}_p$, *contains the*

following axioms:

$$\mathcal{T}_p = \left\{ \begin{array}{ll} 1.\ \mathsf{Secondee} \sqsubseteq \mathsf{User} & 7.\ \exists \mathsf{hasAccess}^- \sqsubseteq \mathsf{Resource} \\ 2.\ \mathsf{FullResearcher} \sqsubseteq \mathsf{User} & 8.\ \exists \mathsf{hasConfidAccess}^- \sqsubseteq \mathsf{Confidential} \\ 3.\ \mathsf{Confidential} \sqsubseteq \mathsf{Resource} & 10.\ \mathsf{owns} \sqsubseteq \mathsf{hasAccess} \\ 5.\ \exists \mathsf{hasAccess} \sqsubseteq \mathsf{User} & 11.\ \mathsf{hasOpenAccess} \sqsubseteq \mathsf{hasAccess} \\ 6.\ \exists \mathsf{owns} \sqsubseteq \mathsf{FullResearcher} & 12.\ \mathsf{hasConfidAccess} \sqsubseteq \mathsf{hasAccess} \end{array} \right\}$$

The positive deductive closure of $\mathcal{A}$, denoted by $cl(\mathcal{A})$, is given by:

$$cl(\mathcal{A}) = \mathcal{A} \cup \left\{ \begin{array}{l} \mathsf{User}(\mathsf{Ben}), \mathsf{User}(\mathsf{Duane}), \mathsf{Resource}(\mathsf{r}_1), \mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1), \\ \mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Duane}), \mathsf{Resource}(\mathsf{r}_2) \end{array} \right\}$$

The new assertions are obtained as follows:

TBox axioms		ABox assertions		Closure assertions
1. $\mathsf{Secondee} \sqsubseteq \mathsf{User}$	+	$\mathsf{Secondee}(\mathsf{Duane})$	$\longrightarrow$	$\mathsf{User}(\mathsf{Duane})$
2. $\mathsf{FullResearcher} \sqsubseteq \mathsf{User}$	+	$\mathsf{FullResearcher}(\mathsf{Ben})$	$\longrightarrow$	$\mathsf{User}(\mathsf{Ben})$
3. $\mathsf{Confidential} \sqsubseteq \mathsf{Resource}$	+	$\mathsf{Confidential}(\mathsf{r}_1)$	$\longrightarrow$	$\mathsf{Resource}(\mathsf{r}_1)$
6. $\exists \mathsf{owns} \sqsubseteq \mathsf{FullResearcher}$	+	$\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$	$\longrightarrow$	$\mathsf{FullResearcher}(\mathsf{Duane})$
12. $\mathsf{hasConfidAccess} \sqsubseteq \mathsf{hasAccess}$	+	$\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)$	$\longrightarrow$	$\mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1)$
10. $\mathsf{owns} \sqsubseteq \mathsf{hasAccess}$	+	$\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$	$\longrightarrow$	$\mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2)$
10. $\mathsf{owns} \sqsubseteq \mathsf{hasAccess}$ 7. $\exists \mathsf{hasAccess}^- \sqsubseteq \mathsf{Resource}$	+	$\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$	$\longrightarrow$	$\mathsf{Resource}(\mathsf{r}_2)$

Note that some assertions may be obtained in multiple ways. For instance, $\mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1)$ *can also be obtained by applying axiom 11 to the assertion* $\mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)$.

The notion of deductive closure of an ABox is closely related to the notion of *chase*, which is rooted in the database community [Maier *et al.* 1979, Johnson & Klug 1982, Calì *et al.* 2003]. The *chase* was also defined for DL-Lite$_\mathcal{R}$ knowledge bases, and consists of the set of assertions obtained by applying the positive inclusion axioms of the TBox to the ABox assertions in a certain (lexicographic) order, to come out with a unique set of assertions. The procedure of constructing the chase employs an assumed fixed infinite set of constant symbols (or individuals); hence, the number of the produced assertions may also be infinite [Calvanese *et al.* 2007]. The difference from the notion of deductive closure, which we are interested in, is the infinite set of constant symbols used in the process. Our definition of the deductive closure is limited to the individuals present in the ABox in order to guarantee the production of a finite set. This aligns with the literature on inconsistency management [Lembo *et al.* 2010, Bienvenu & Bourgaux 2017, Benferhat *et al.* 2015].

The notion of ABox completion, which was used in the context of the combined approach for reasoning with DL-Lite$_\mathcal{R}$ knowledge bases [Lutz *et al.* 2013], guarantees the production of a finite set of assertions by introducing fresh individuals in an economic way and then applies a filtering approach to avoid returning query answers with fresh individuals. However, it only applies to consistent knowledge bases. Finally, some more recent work efforts considered the

notion of saturation, which extends the notion of chase for quantified $\mathcal{EL}$ ABoxes. These are ABoxes that are naturally accompanied with a set of existential variables, and for which the introduction of fresh constant symbols is replaced with the introduction of new variables to construct the chase [Baader *et al.* 2020, Baader *et al.* 2021].

2.2.2 Closure-Based ABox Repairs

We focus on the application of the deductive closure to the ABox repair semantics. The idea is to add an additional step of closure at some level in the repair process to obtain (or preserve) some conclusions that follow under the AR semantics while maintaining tractability.

The deductive closure operator was applied both before and after computing the ABox repairs. First, it was performed directly on the initial inconsistent ABox, resulting in the CAR and ICAR semantics [Lembo *et al.* 2010]. Then, it was applied to each of the ABox repairs, resulting in the ICR semantics [Bienvenu 2012a]. We recall each of them in the following.

The CAR and ICAR semantics

The Closed ABox Repairs (CAR) semantics proceeds by closing the initial inconsistent ABox before computing its repairs, from which queries are answered. The Intersection of Closed ABox Repairs (ICAR) semantics proceeds in the same way by closing the initial inconsistent ABox, then it computes and intersects the obtained repairs to yield a single repair. In order to be able to close the inconsistent ABox, the authors in [Lembo *et al.* 2010] introduced a special closure operator, denoted $cl^*(\mathcal{A})$, which is defined as follows:

$$cl^*(\mathcal{A}) = \{\varphi \mid \exists \mathcal{B} \subseteq \mathcal{A} \text{ such that } \langle \mathcal{T}, \mathcal{B} \rangle \text{ is consistent and } \langle \mathcal{T}, \mathcal{B} \rangle \models \varphi\}$$

Example 2.5. *Let us compute the deductive closure $cl^*(\mathcal{A})$ of our ABox $\mathcal{A}$.*

$$cl^*(\mathcal{A}) = \mathcal{A} \cup \left\{ \begin{array}{l} \mathsf{User}(\mathsf{Ben}), \mathsf{User}(\mathsf{Duane}), \mathsf{Resource}(\mathsf{r}_1), \mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1), \\ \mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Duane}), \mathsf{Resource}(\mathsf{r}_2) \end{array} \right\}$$

Notice that the closure returned by the special closure operator $cl^*(\mathcal{A})$ is the same with the deductive closure computed for our ABox in Example 2.4. However, this is not always the case for DL-Lite knowledge bases. The difference between the two operators is highlighted in the following example.

Example 2.6. *Let $\mathcal{K}^* = \langle \mathcal{T}^*, \mathcal{A}^* \rangle$ be a knowledge base, where the TBox is given by:*

$$\mathcal{T}^* = \{\exists \mathsf{hasAccess} \sqsubseteq \mathsf{User} \qquad \exists \mathsf{hasAccess}^- \sqsubseteq \mathsf{Resource} \qquad \mathsf{User} \sqsubseteq \neg\, \mathsf{Resource}\}$$

and the ABox is given by the single assertion:

$$\mathcal{A}^* = \{\mathsf{hasAccess}(s_1, s_1)\}$$

One can check that $\mathcal{K}^*$ is inconsistent, because the negative inclusion axiom $\exists \mathsf{hasAccess} \sqsubseteq \neg\, \exists \mathsf{hasAccess}^-$ can be inferred from the TBox and the assertion $\mathsf{hasAccess}(s_1, s_1) \in \mathcal{A}^*$. As a consequence, we have $\mathsf{hasAccess}(s_1, s_1) \in cl(\mathcal{A}^*)$ because $\langle \mathcal{T}^*, \mathcal{A}^* \rangle \models \mathsf{hasAccess}(s_1, s_1)$ (the negative inclusion axioms of the TBox are ignored in this case). However, $\mathsf{hasAccess}(s_1, s_1) \notin cl^*(\mathcal{A}^*)$ because there is no consistent subset of $\mathcal{A}^*$ to infer

hasAccess(s_1, s_1): *the minimal subset to infer the assertion contains the assertions itself, namely* {hasAccess(s_1, s_1)}*, which is inconsistent with respect to* $\mathcal{T}^*$.

Remark. Note that the first standard definition of deductive closure can also be applied to inconsistent knowledge bases. However, it is intended to be applied on consistent knowledge bases. Moreover, in the case of consistency, both definitions amount to the standard DL-Lite$_\mathcal{R}$ entailment. Note also that the difference shown using the above example is the only difference in the case of DL-Lite$_\mathcal{R}$ knowledge bases. This is not the case for other languages that allow specifying negated assertions in the ABox (of the form $\neg A(a)$) and not the case for knowledge bases that allow incoherent TBoxes, mainly in the situation where a single atomic concept or role is unsatisfiable (having the axiom $C \sqsubseteq \neg C$ entailed by the TBox).

Now, let us recall the definition of the Closed ABox Repairs.

Definition 2.6 (Closed ABox Repair)**.** *A subset* $\mathcal{R} \subseteq cl^*(\mathcal{A})$ *is a closed ABox repair of* $\mathcal{A}$ *with respect to* $\mathcal{T}$*, if:*

(i) $\langle \mathcal{T}, \mathcal{R} \rangle$ *is consistent, and;*

(ii) there is no consistent subset $\mathcal{R}' \subseteq cl^*(\mathcal{A})$ *such that* $\mathcal{R} \cap \mathcal{A} \subsetneq \mathcal{R}' \cap \mathcal{A}$ *or* $(\mathcal{R} \cap \mathcal{A} = \mathcal{R}' \cap \mathcal{A}$ *and* $\mathcal{R} \subsetneq \mathcal{R}')$.

Example 2.7. *Let us now compute the closed ABox repairs of the ABox from our running example. The closed ABox repairs of the ABox* $\mathcal{A}$ *are illustrated in Table 2.1, where the check-marks (✓) indicate the membership of the assertions to the repairs.*

Table 2.1: The closed ABox repairs $\mathcal{R}_1, \ldots, \mathcal{R}_4$ of the ABox $\mathcal{A}$

	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_3$	$\mathcal{R}_4$
FullResearcher(Ben)	✓	✓	✓	✓
Secondee(Duane)	✓	✓		
Confidential(r_1)	✓	✓	✓	✓
hasConfidAccess(Ben,r_1)	✓		✓	
hasOpenAccess(Ben,r_1)		✓		✓
owns(Duane,r_2)			✓	✓
User(Ben)	✓	✓	✓	✓
User(Duane)	✓	✓	✓	✓
Resource(r_1)	✓	✓	✓	✓
hasAccess(Ben,r_1)	✓	✓	✓	✓
hasAccess(Duane,r_2)	✓	✓	✓	✓
FullResearcher(Duane)			✓	✓
Resource(r_2)	✓	✓	✓	✓

Now, the same semantics applied to standard ABox repairs can be applied to closed ABox repairs. In particular, universal inference from the set of the closed ABox repairs in the same style as the AR semantics yields the CAR semantics. In addition, inference from the intersection of the closed ABox repairs yields the ICAR semantics [Lembo *et al.* 2010].

Definition 2.7 (CAR semantics)**.** *An assertion φ follows from the knowledge base $\mathcal{K}$ under the CAR semantics, denoted by $\mathcal{K} \models^{\mathsf{CAR}} \varphi$, if $\forall \mathcal{R}_i$ a closed ABox repair as given per Definition 2.6, $\langle \mathcal{T}, \mathcal{R}_i \rangle \models \varphi$. Where $\models$ denotes the standard DL-Lite$_\mathcal{R}$ entailment.*

Definition 2.8 (ICAR semantics)**.** *The ICAR repair of an ABox $\mathcal{A}$ is given by:*

$$\mathsf{ICAR}(\mathcal{A}) = \bigcap \{\mathcal{R}_i \mid \mathcal{R}_i \text{ is a closed ABox repair of } \mathcal{A}\}$$

An assertion φ follows from the knowledge base $\mathcal{K}$ under the ICAR semantics, denoted by $\mathcal{K} \models^{\mathsf{ICAR}} \varphi$, if $\langle \mathcal{T}, \mathsf{ICAR}(\mathcal{A}) \rangle \models \varphi$. Where $\models$ denotes the standard DL-Lite$_\mathcal{R}$ entailment.

Example 2.8. *Let us now compute the repair induced by the ICAR semantics from the set of the closed ABox repairs of our ABox. It is sufficient to intersect all repairs, namely:*

$$\mathsf{ICAR}(\mathcal{A}) = \left\{ \begin{array}{l} \mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{Confidential}(\mathsf{r}_1), \mathsf{User}(\mathsf{Ben}), \mathsf{User}(\mathsf{Duane}), \\ \mathsf{Resource}(\mathsf{r}_1), \mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{Resource}(\mathsf{r}_2) \end{array} \right\}$$

One can see that the repair computed under the ICAR semantics is larger than the repair computed under the IAR semantics. Moreover, it also returns assertions that do not follow from the AR semantics, such as $\mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2)$.

The ICAR semantics has the advantage of being tractable, while returning assertions that follow under the AR semantics but not under the IAR semantics (impact of using the deductive closure. However, ICAR might return undesirable consequences that follow only from a conflicting assertion. For instance, in our running example, it returns the assertions $\mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2)$ and $\mathsf{Resource}(\mathsf{r}_2)$ as valid conclusions, although they only follow from $\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$, which is involved in a conflict and not returned by neither the CAR nor the ICAR semantics.

Conjunctive query answering and instance checking under the ICAR semantics over DL-Lite knowledge bases is in AC^0 with respect to data complexity [Lembo *et al.* 2010, Lembo *et al.* 2011]. Whereas, conjunctive query answering under the CAR semantics over DL-Lite knowledge bases is coNP-complete with respect to data complexity, while instance checking is in AC^0. Note that the CAR semantics and the ICAR semantics coincide for ground conjunctive queries (instance queries), that is why instance checking is in AC^0 for the CAR semantics.

In addition, note that there is a difference between the Closed ABox repairs, as given by Definition 2.6, and the standard repairs of the closed ABox $cl^*(\mathcal{A})$. This is due to the fact that the closed ABox may have a repair for which the intersection with the ABox is empty [Bienvenu & Bourgaux 2017].

The Intersection of Closed Repairs (ICR) semantics

Another way of applying the deductive closure in the process of repair-based querying of inconsistent DL-Lite$_\mathcal{R}$ knowledge bases proceeds by first computing all the ABox repairs, then computing the deductive closure of each of the repairs, before intersecting the obtained closed repairs [Bienvenu 2012a].

Definition 2.9 (ICR semantics)**.** *The ICR repair of an ABox $\mathcal{A}$ is given by:*

$$\mathsf{ICR}(\mathcal{A}) = \bigcap_i \{cl(\mathcal{R}_i) \mid \mathcal{R}_i \text{ is a repair of } \mathcal{A}\}$$

An assertion φ follows from the knowledge base $\mathcal{K}$ under the ICR semantics, denoted $\mathcal{K} \models^{\mathsf{ICR}} \varphi$, if $\langle \mathcal{T}, \mathsf{ICR}(\mathcal{A}) \rangle \models \varphi$. Where $\models$ denotes the standard DL-Lite$_\mathcal{R}$ entailment.

Note that since a repair is consistent by definition, applying the deductive closure as given per Definition 2.5, or applying the special deduction operator in the same way as the closed ABox repairs amounts to the same set.

Example 2.9. *Let us now apply the ICR semantics to our running example. We compute the standard deductive closure of each ABox repair, which were computed in Example 1.17:*

$$\begin{aligned}
cl(\mathcal{R}_1) =& \{\mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{Secondee}(\mathsf{Duane}), \mathsf{Confidential}(\mathsf{r}_1), \mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)\} \\
& \cup \{\mathsf{User}(\mathsf{Ben}), \mathsf{User}(\mathsf{Duane}), \mathsf{Resource}(\mathsf{r}_1), \mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1)\} \\
cl(\mathcal{R}_2) =& \{\mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{Confidential}(\mathsf{r}_1), \mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)\} \\
& \cup \{\mathsf{User}(\mathsf{Ben}), \mathsf{Resource}(\mathsf{r}_1), \mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2), \\
& \quad \mathsf{FullResearcher}(\mathsf{Duane}), \mathsf{User}(\mathsf{Duane}), \mathsf{Resource}(\mathsf{r}_2)\} \\
cl(\mathcal{R}_3) =& \{\mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{Secondee}(\mathsf{Duane}), \mathsf{Confidential}(\mathsf{r}_1), \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)\} \\
& \cup \{\mathsf{User}(\mathsf{Ben}), \mathsf{User}(\mathsf{Duane}), \mathsf{Resource}(\mathsf{r}_1), \mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1)\} \\
cl(\mathcal{R}_4) =& \{\mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{Confidential}(\mathsf{r}_1), \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)\} \\
& \cup \{\mathsf{User}(\mathsf{Ben}), \mathsf{Resource}(\mathsf{r}_1), \mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2), \\
& \quad \mathsf{FullResearcher}(\mathsf{Duane}), \mathsf{User}(\mathsf{Duane}), \mathsf{Resource}(\mathsf{r}_2)\}
\end{aligned}$$

The intersection of the closed repairs results in the following single ICR repair:

$$\mathsf{ICR}(\mathcal{A}) = \left\{ \begin{array}{l} \mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{Confidential}(\mathsf{r}_1), \mathsf{User}(\mathsf{Ben}), \mathsf{User}(\mathsf{Duane}), \\ \mathsf{Resource}(\mathsf{r}_1), \mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1) \end{array} \right\}$$

One can see that unlike the IAR semantics, shown in Example 1.19, the ICR semantics allows to derive assertions that follow under the AR semantics, namely, $\mathsf{User}(\mathsf{Duane})$ *and* $\mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1)$*. In addition, one can also see that the assertions* $\mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2)$ *and* $\mathsf{Resource}(\mathsf{r}_2)$ *are derived using the ICAR semantics (Example 2.8) but do not follow here under the ICR semantics.*

Conjunctive query answering and instance checking under the ICR semantics over DL-Lite knowledge bases is coNP-complete with respect to data complexity [Bienvenu 2012a]. Note that query answering under the ICR semantics amounts to standard query answering over the assertions that hold under the AR semantics.

Problem Statement

Can we define an inconsistency-tolerant semantics that applies the deductive closure in a safe way (without inferring questionable conclusions) while maintaining tractability?

2.2.3 The Closure-Based Partially Ordered Possibilistic Repair

In order to answer the main question addressed in this chapter, which is going beyond the possibilistic repair for partially preordered ABoxes, without inducing additional computational

cost, we propose the use of positive deductive closure, as given by Definition 2.5, to obtain a more productive repair.

The deductive closure can be applied at two levels for partially preordered ABoxes. The first is a direct application on the initial inconsistent ABox, in the style of the ICAR semantics [Lembo *et al.* 2010]. The main challenge facing the ICAR semantics is that it might yield questionable consequences, as we have shown in Example 2.8 (Chapter 1). We postpone the study of the implications of such an application in the case of partially preordered ABoxes to a later chapter (Chapter 3). In this part, a main focus is given to the use of deductive closure after computing the repairs of the totally preordered compatible ABoxes, in the same spirit of the ICR semantics [Bienvenu 2012a]. Note that since possibilistic repairs are consistent by definition, the use of the deductive closure as given by Definition 2.5, or as given by the special closure operator used for the ICAR semantics [Lembo *et al.* 2010], yields the same set.

The process of computing a closure-based possibilistic repair for partially preordered ABoxes is given as follows:

Step 1: Extend the partial preorder $\trianglerighteq$ into a family of m compatible total preorders, denoted $(\geq_1, \ldots, \geq_m)$. Each compatible total preorder $\geq_i$ defines a compatible ABox $\mathcal{A}_{\geq_i}$.
Step 2: Compute the possibilistic repair $\mathcal{R}_\pi(A_{\geq_i})$ of each totally preordered ABox.
Step 3: Apply the deductive closure operator on each possibilistic repair $cl(\mathcal{R}_\pi(A_{\geq_i}))$.
Step 4: Intersect all closed possibilistic repairs to obtain the closure-based repair, denoted by $c\pi(\mathcal{A}_{\trianglerighteq})$.

Figure 2.4 summarises the process given above for computing the closure-based partially preordered possibilistic repair, namely $c\pi(\mathcal{A}_{\trianglerighteq})$.

The closure-based partially preordered possibilistic repair is also called the Cπ-repair, its formal definition is given as follows.

Definition 2.10 (Cπ-repair)**.** *Let $\mathcal{A}_{\trianglerighteq}$ be a partially preordered ABox. Let $\mathcal{A}_{\geq_i}$, with $i = 1, \ldots, m$, denote all its compatible bases and let $\mathcal{R}_\pi(\mathcal{A}_{\geq_i})$ be the associated possibilistic repair. The closure-based partially preordered possibilistic repair of $\mathcal{A}_{\trianglerighteq}$, denoted $c\pi(\mathcal{A}_{\trianglerighteq})$, is obtained as follows:*

$$c\pi(\mathcal{A}_{\trianglerighteq}) = \bigcap_{i=1}^{m} \{cl(\mathcal{R}_\pi(\mathcal{A}_{\geq_i})) \mid \mathcal{A}_{\geq_i} \textit{ is compatible with } \mathcal{A}_{\trianglerighteq}.\}$$

Example 2.10. *Let us apply the Cπ-repair as given by Definition 2.10 to our example's knowledge base $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$. First, the total preorder extensions of the partial preorder $\trianglerighteq$ defined over the assertions of $\mathcal{A}_{\trianglerighteq}$ are shown in Figure 2.2. We compute the possibilistic repair of each totally preordered ABox induced by the total preorders. Figure 2.5 summarises the repairs of the main total preorder extensions, where the dashed lines (− − −) represent conflicts, straight arrows represent the strict preference defined by its total preorder extension $\geq_i$, and $\equiv$ represent equally certain assertions. In each total preorder extension repair, namely $\mathcal{R}_\pi(\mathcal{A}_{\geq_i})$, the struck-through assertions are not admitted in the possibilistic repairs. The repairs of the full total preorder extensions are given in a separate figure in the next page (Figure 2.6).*

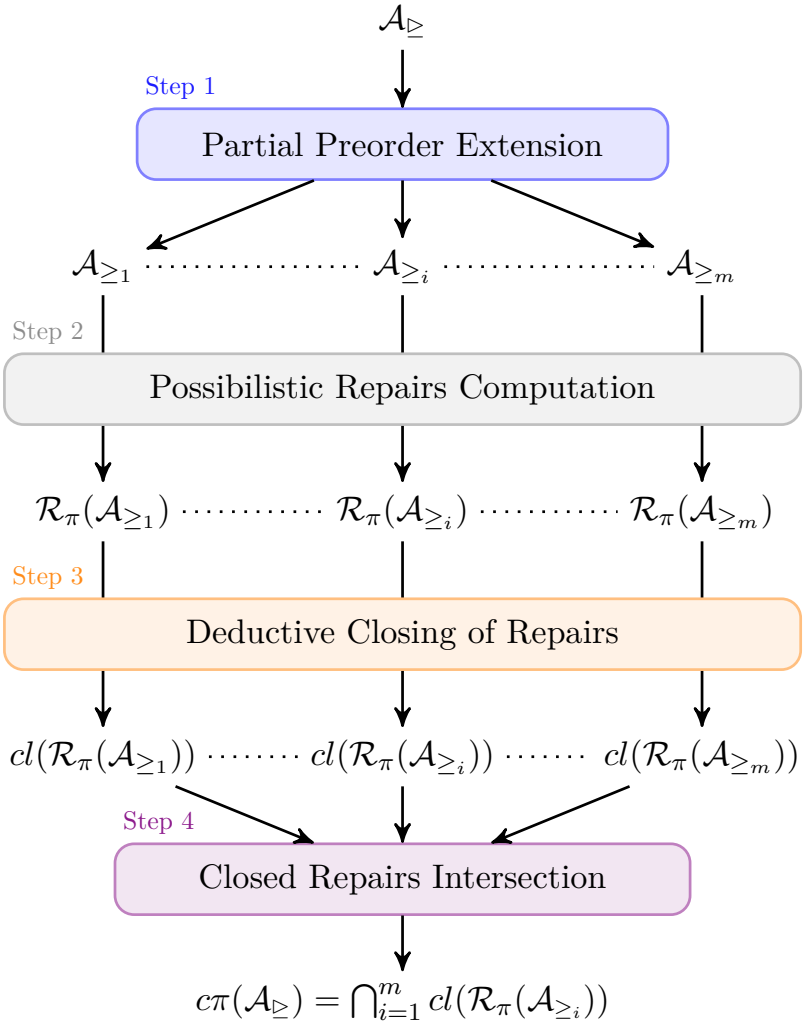


Figure 2.4: The process of computing the Cπ-repair

One can see that all repairs contain the assertion $\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$. *Moreover, each repair contains* $\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)$ *or* $\mathsf{FullResearcher}(\mathsf{Ben})$ *and* $\mathsf{Confidential}(\mathsf{r}_1)$.
The resulting repairs are as follows:

$$\begin{aligned}
\mathcal{R}_\pi(\mathcal{A}_{\geq_1}) &= \{\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{Confidential}(\mathsf{r}_1), \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)\}\\
\mathcal{R}_\pi(\mathcal{A}_{\geq_9}) &= \{\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)\}\\
&\vdots\\
\mathcal{R}_\pi(\mathcal{A}_{\geq_m}) &= \{\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{Confidential}(\mathsf{r}_1)\}
\end{aligned}$$

Now, let us apply the deductive closure operator on each repair of a total preorder extension that is compatible with $\mathcal{A}_{\trianglerighteq}$. *The resulting closed repairs are:*

$$\begin{aligned}
cl(\mathcal{R}_\pi(\mathcal{A}_{\geq_1})) =& \mathcal{R}_\pi(\mathcal{A}_{\geq_1}) \cup \{ \quad \mathsf{FullResearcher}(\mathsf{Duane}), \mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{User}(\mathsf{Duane}),\\
& \qquad \mathsf{Resource}(\mathsf{r}_2), \mathsf{User}(\mathsf{Ben}), \mathsf{Resource}(\mathsf{r}_1), \mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1)\}\\
cl(\mathcal{R}_\pi(\mathcal{A}_{\geq_9})) =& \mathcal{R}_\pi(\mathcal{A}_{\geq_9}) \cup \{ \quad \mathsf{FullResearcher}(\mathsf{Duane}), \mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{User}(\mathsf{Duane}),\\
& \qquad \mathsf{Resource}(\mathsf{r}_2), \mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{Confidential}(\mathsf{r}_1),\\
& \qquad \mathsf{User}(\mathsf{Ben}), \mathsf{Resource}(\mathsf{r}_1)\}\\
&\vdots\\
cl(\mathcal{R}_\pi(\mathcal{A}_{\geq_m})) =& \mathcal{R}_\pi(\mathcal{A}_{\geq_m}) \cup \{ \quad \mathsf{FullResearcher}(\mathsf{Duane}), \mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{User}(\mathsf{Duane}),\\
& \qquad \mathsf{Resource}(\mathsf{r}_2), \mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{User}(\mathsf{Ben}), \mathsf{Resource}(\mathsf{r}_1)\}
\end{aligned}$$

The intersection of all the closed possibilistic repairs amounts to the single Cπ-repair of $\mathcal{A}_{\trianglerighteq}$:

$$\begin{aligned}
c\pi(\mathcal{A}_{\trianglerighteq}) = \{&\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Duane}), \mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{User}(\mathsf{Duane}),\\
&\mathsf{Resource}(\mathsf{r}_2), \mathsf{User}(\mathsf{Ben}), \mathsf{Resource}(\mathsf{r}_1), \mathsf{Confidential}(\mathsf{r}_1)\}
\end{aligned}$$

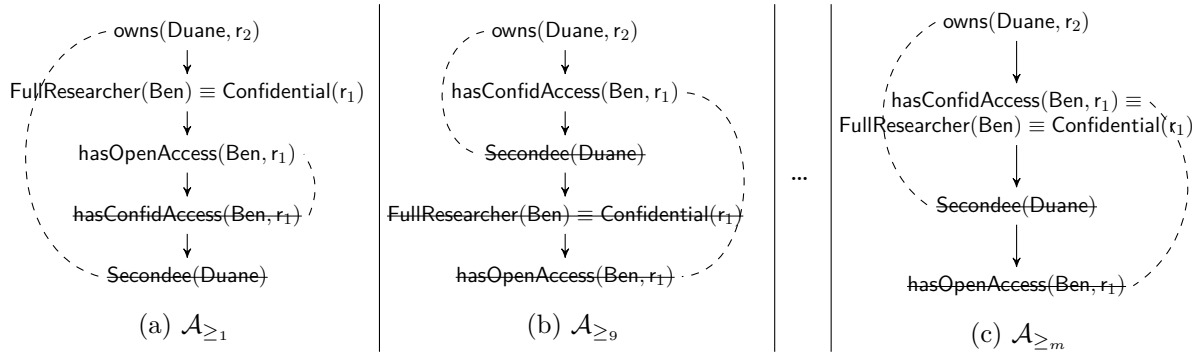
Notice that intersecting the repairs directly amounts to the π-accepted assertions. Namely, only the assertion $\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$ *is in* $\pi(\mathcal{A}_{\trianglerighteq})$. *Moreover, the application of the closure operator on the π-repair yields only assertions about* Duane *and* r_2, *resulting in the loss of the assertions about* Ben *and* r_1, *namely:*

$$cl(\pi(\mathcal{A}_{\trianglerighteq})) = \{\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Duane}), \mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{User}(\mathsf{Duane}), \mathsf{Resource}(\mathsf{r}_2)\}$$

Remark. The deductive closure can be applied to the resulting π-repair, namely $\pi(\mathcal{A}_{\trianglerighteq})$ to include all assertions that follow from it. We observe that the closure of the repair in this case is still not as large as the set computed by the Cπ-repair method (as shown by Example 2.10). This example suggests that $c\pi(\mathcal{A}_{\trianglerighteq})$ is larger than both $\pi(\mathcal{A}_{\trianglerighteq})$ and its closure $cl(\pi(\mathcal{A}_{\trianglerighteq}))$. Namely:

$$\pi(\mathcal{A}_{\trianglerighteq}) \subseteq cl(\pi(\mathcal{A}_{\trianglerighteq})) \subseteq c\pi(\mathcal{A}_{\trianglerighteq}).$$

The first results of the Cπ-repair are given by the following proposition. We establish the consistency of the repair $c\pi(\mathcal{A}_{\trianglerighteq})$, which is expected, since it results from the intersection of consistent sets. Furthermore, we show the productivity of the Cπ-repair compared to the π-repair (Definition 2.2) and its closure.



(a) $\mathcal{A}_{\geq_1}$ (b) $\mathcal{A}_{\geq_9}$ (c) $\mathcal{A}_{\geq_m}$

Figure 2.5: The possibilistic repairs $\mathcal{R}_\pi(\mathcal{A}_{\geq_1}), \ldots, \mathcal{R}_\pi(\mathcal{A}_{\geq_m})$ of ABoxes compatible with $\mathcal{A}_\trianglerighteq$

Proposition 2.1. *Let $\mathcal{K}_\trianglerighteq = \langle \mathcal{T}, \mathcal{A}_\trianglerighteq \rangle$ be partially preordered KB. Let $c\pi(\mathcal{A}_\trianglerighteq)$ and $\pi(\mathcal{A}_\trianglerighteq)$ be the Cπ-repair and the π-repair of $\mathcal{A}_\trianglerighteq$, as given by Definitions 2.10 and 2.2, respectively.*

1. *$\langle \mathcal{T}, c\pi(\mathcal{A}_\trianglerighteq) \rangle$ is consistent.*
2. *$\pi(\mathcal{A}_\trianglerighteq) \subseteq c\pi(\mathcal{A}_\trianglerighteq)$.*

Proof.

1. *The consistency of $c\pi(\mathcal{A}_\trianglerighteq)$ follows from the fact that for any totally preordered ABox $\mathcal{A}_{\geq_i}$, $i = 1, \ldots, m$, the sub-base $\mathcal{R}_\pi(\mathcal{A}_{\geq_i})$ given by Definition 1.24 is consistent. Therefore, the positive deductive closure $cl(\mathcal{R}_\pi(\mathcal{A}_{\geq_i}))$ is also consistent, and so is the intersection of all these closed repairs.*
2. *The proof of the inclusion $\pi(\mathcal{A}_\trianglerighteq) \subseteq c\pi(\mathcal{A}_\trianglerighteq)$ is immediate, as for each ABox $\mathcal{A}_{\geq_i}$ compatible with $\mathcal{A}_\trianglerighteq$, we have $\mathcal{R}_\pi(\mathcal{A}_{\geq_i}) \subseteq cl(\mathcal{R}_\pi(\mathcal{A}_{\geq_i}))$. Therefore:*

$$\bigcap_{i=1}^{m} \{\mathcal{R}_\pi(\mathcal{A}_{\geq_i}) | \mathcal{A}_{\geq_i} \text{ is compatible with } \mathcal{A}_\trianglerighteq\} \subseteq \bigcap_{i=1}^{m} \{cl(\mathcal{R}_\pi(\mathcal{A}_{\geq_i})) | \mathcal{A}_{\geq_i} \text{ compatible with } \mathcal{A}_\trianglerighteq\}$$

□

Remark. The converse of $\pi(\mathcal{A}_\trianglerighteq) \subseteq c\pi(\mathcal{A}_\trianglerighteq)$ is false. Namely, one can exhibit a knowledge base such that $c\pi(\mathcal{A}_\trianglerighteq) \not\subset \pi(\mathcal{A}_\trianglerighteq)$. Example 2.10 illustrates a case where the Cπ-repair $c\pi(\mathcal{A}_\trianglerighteq)$ is strictly larger than the π-repair $\pi(\mathcal{A}_\trianglerighteq)$. Note that there are also situations where the converse is not false. For instance, when the knowledge base is consistent, the Cπ-repair $c\pi(\mathcal{A}_\trianglerighteq)$ and the π-repair $\pi(\mathcal{A}_\trianglerighteq)$ are equivalent.

Problem Statement

The number of the total preorder extensions of a partial preorder can be exponential in the worst case. Can we avoid enumerating them in the process of computing the Cπ-repair?

2.2.4 Equivalent Characterisation: the Dominant Supports

Up to now, we have introduced a productive method to compute a repair for partially preordered knowledge bases. However, the method is naive as it relies on the exhibition of the total preorder extensions of a partial preorder. The number of these extensions is exponential in the worst

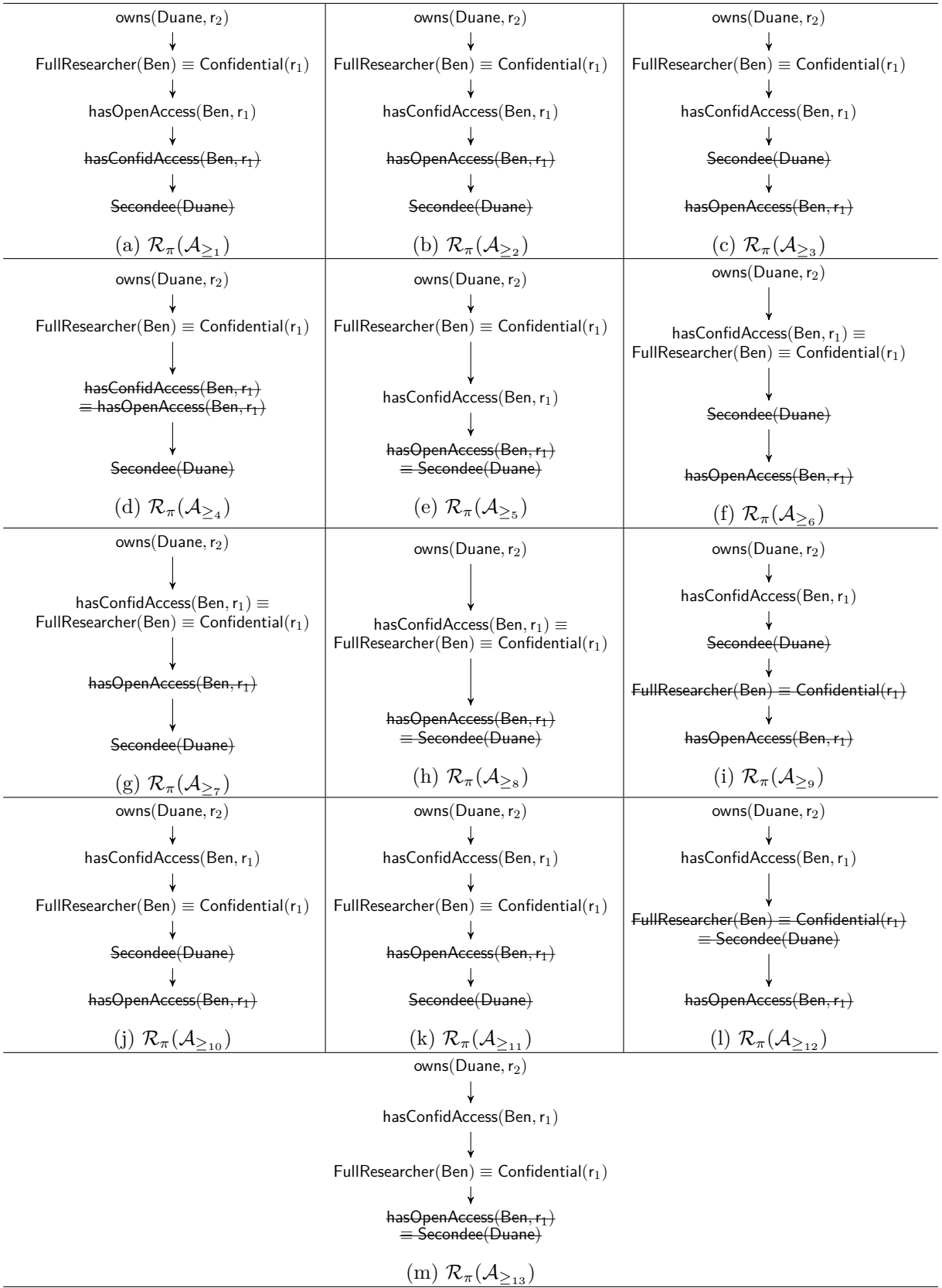


Figure 2.6: The possibilistic repairs of the total preorder extensions $\mathcal{R}_\pi(\mathcal{A}_{\geq_1}), \ldots, \mathcal{R}_\pi(\mathcal{A}_{\geq_{13}})$.

case. The aim of this part is to introduce a method that computes the Cπ-repair in tractable time, i.e., without exhibiting all total preorders that are compatible with the partial preorder defined over the initial ABox. The idea is to use the conflict set of an ABox, as given per Definition 1.9, to check if a given assertion belongs to the Cπ-repair. Our characterisation stems from the following observations:

- An assertion φ belongs to the Cπ-repair of $\mathcal{A}_{\geq}$ if it is in all the closures of the possibilistic repairs of the compatible totally preordered ABoxes, namely $cl(\mathcal{R}_\pi(\mathcal{A}_{\geq_1})), \ldots, cl(\mathcal{R}_\pi(\mathcal{A}_{\geq_m}))$.
- φ belongs to the closures $cl(\mathcal{R}_\pi(\mathcal{A}_{\geq_1})), \ldots, cl(\mathcal{R}_\pi(\mathcal{A}_{\geq_m}))$, if it can be entailed from all the possibilistic repairs $\mathcal{R}_\pi(\mathcal{A}_{\geq_1}), \ldots, \mathcal{R}_\pi(\mathcal{A}_{\geq_m})$.
- φ can be entailed from a repair $\mathcal{R}_\pi(\mathcal{A}_{\geq_i})$ if there is an inclusion-minimal subset of $\mathcal{R}_\pi(\mathcal{A}_{\geq_i})$ that **supports** it.
- This subset is in some standard possibilistic repair $\mathcal{R}_\pi(\mathcal{A}_{\geq_i})$ if all its elements are strictly more preferred than the most certain assertion that makes the totally preordered ABox $\mathcal{A}_{\geq_i}$ inconsistent.
- The subset does not need to be unique to cover all extensions.

Supports

In order to provide an efficient method to compute the Cπ-repair, we start by introducing the notion of a support of an assertion in a DL-Lite$_\mathcal{R}$ ABox.

Definition 2.11 (Support)**.** *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL-Lite$_\mathcal{R}$ KB. Let B be a concept name and R be a role name in $\mathcal{T}$. Let a and b be two individuals in $\mathcal{A}$. The subset $\mathcal{B} \subseteq \mathcal{A}$ is a support for $B(a)$ (resp. $R(a,b)$) in $\mathcal{A}$ if:*

- *$\langle \mathcal{T}, \mathcal{B} \rangle$ is consistent, and*
- *$\langle \mathcal{T}, \mathcal{B} \rangle \vDash B(a)$, (resp. $\langle \mathcal{T}, \mathcal{B} \rangle \vDash R(a,b)$), and*
- *$\forall \mathcal{B}' \subsetneq \mathcal{B}$, $\langle \mathcal{T}, \mathcal{B}' \rangle \nvDash B(a)$, (resp. $\langle \mathcal{T}, \mathcal{B}' \rangle \nvDash R(a,b)$).*

Where $\vDash$ is the standard DL-Lite$_\mathcal{R}$ inference relation.

An important characteristic of the notion of support of an assertion in a DL-Lite$_\mathcal{R}$ ABox is that the size of a support is at most one assertion. This follows from the minimality condition in the definition of a support, in addition to the fact that DL-Lite$_\mathcal{R}$ TBoxes do not allow conjunctions or qualified existential restrictions in their inclusion axioms.

The support is a special case of the notion of justification, which is largely used in the case of incoherent DL TBoxes [Kalyanpur *et al.* 2007, Ji *et al.* 2022] and is defined as the minimal subset of an ontology (a set of TBox axioms and ABox assertions) that is sufficient to draw a conclusion.

Example 2.11. *From our running example, the assertion* $\mathsf{Resource}(\mathsf{r}_1)$ *has two supports in the ABox* $\mathcal{A}_{\geq}$*:* $\{\mathsf{Confidential}(\mathsf{r}_1)\}$ *and* $\{\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)\}$.

Dominance

The other notion necessary to constitute a tractable characterisation of the Cπ-repair is the notion of dominance. This notion is a natural way to extend the partial preorder defined over ABox assertions into a partial preorder defined over subsets of assertions.

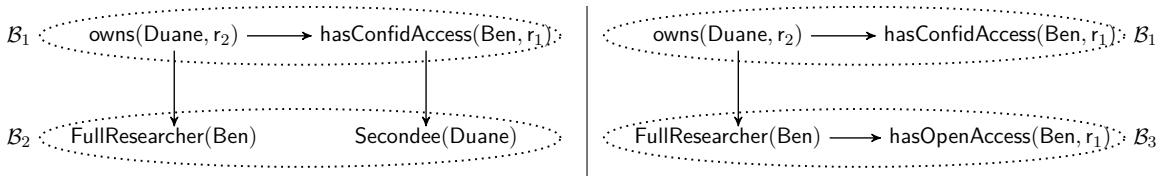
Definition 2.12 (Dominance)**.** *Let $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$ be a partially preordered KB. Let $\mathcal{B}_1 \subseteq \mathcal{A}_{\trianglerighteq}$ and $\mathcal{B}_2 \subseteq \mathcal{A}_{\trianglerighteq}$. We say that $\mathcal{B}_1$ dominates $\mathcal{B}_2$ if: for all $\varphi_j \in \mathcal{B}_1$, there is $\varphi_k \in \mathcal{B}_2$ such that $\varphi_j \rhd \varphi_k$.*

Example 2.12. *Let us use subsets of Example 2.1 ABox $\mathcal{A}_{\trianglerighteq}$ to illustrate the dominance notion, let $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$ be the following subsets of assertions:*

$$\mathcal{B}_1 = \{\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)\}$$
$$\mathcal{B}_2 = \{\mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{Secondee}(\mathsf{Duane})\}$$
$$\mathcal{B}_3 = \{\mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)\}$$

Figure 2.7 illustrates the dominance relation between the subsets $\mathcal{B}_1$ and $\mathcal{B}_2$ and between the subsets $\mathcal{B}_1$ and $\mathcal{B}_3$ where straight arrows ($\longrightarrow$) denote strict preference, namely:

- *$\mathcal{B}_1$ dominates $\mathcal{B}_2$ because* $\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$ *is more preferred to* $\mathsf{FullResearcher}(\mathsf{Ben})$ *and* $\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)$ *is more preferred to* $\mathsf{Secondee}(\mathsf{Duane})$.
- *$\mathcal{B}_1$ does not dominate $\mathcal{B}_3$ because* $\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)$ *is not more preferred to any of the assertions in $\mathcal{B}_3$, namely,* $\mathsf{FullResearcher}(\mathsf{Ben})$ *and* $\mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)$.



(a) $\mathcal{B}_1$ dominates $\mathcal{B}_2$ (b) $\mathcal{B}_1$ does not dominate $\mathcal{B}_3$

Figure 2.7: The dominance relation.

We start with the following intermediary result, which is required to establish the proof of the equivalence of the characterisation (which will be given afterwards) with the Cπ-repair method.

Lemma 2.1. *The conflict set of a DL-Lite$_\mathcal{R}$ knowledge base is independent from the ordering defined over its assertions whether the ordering is partial or total, namely, $\forall \mathcal{A}_{\geq_i}$ a totally preordered ABox that is compatible with the partially preordered ABox $\mathcal{A}_{\trianglerighteq}$, we have that $\mathsf{Cf}(\mathcal{A}_{\geq_i}) = \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$.*

Proof. *The proof of this lemma is immediate and follows from the definition of a conflict (Definition 1.9) which was given independently from any ordering.* □

We deal first with the special case of having a consistent knowledge base. In that case, inference under the Cπ-repair amounts to the standard DL-Lite$_\mathcal{R}$ entailment. Hence, it is enough to return the closed ABox $cl(\mathcal{A}_{\trianglerighteq})$ when computing the Cπ-repair.

Lemma 2.2. *Let $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$ be a consistent, partially preordered KB, and $\mathsf{Cf}(\mathcal{A}_{\trianglerighteq}) = \emptyset$. Consider $cl(\cdot)$, given by Definition 2.5. Let φ be an assertion. Then: $c\pi(\mathcal{A}_{\trianglerighteq}) = cl(\mathcal{A}_{\trianglerighteq})$. Equivalently:*

$$\varphi \in c\pi(\mathcal{A}_{\trianglerighteq}) \text{ if and only if there is } \mathcal{B} \subseteq \mathcal{A}_{\trianglerighteq} \text{ such that } \langle \mathcal{T}, \mathcal{B} \rangle \models \varphi.$$

Proof. *The proof is straightforward. Let $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$ be a consistent KB. Let $c\pi(\mathcal{A}_{\trianglerighteq})$ be as given in Definition 2.10. Since $\mathcal{A}_{\trianglerighteq}$ is consistent, for each compatible base $\mathcal{A}_{\geq_i}$, $i = 1, \ldots, m$, the possibilistic repair is simply the whole ABox, namely $\mathcal{R}_\pi(\mathcal{A}_{\geq_i}) = \mathcal{A}_{\geq_i}$ (see Definition 1.24). Hence, $c\pi(\mathcal{A}_{\trianglerighteq}) = \bigcap_{i=1}^{m} \{cl(\mathcal{A}_{\geq_i})\}$, which is equivalent to $cl(\mathcal{A}_{\trianglerighteq})$.* □

Now, we turn our focus to inconsistent knowledge bases. We use the notions of support and dominance, which we gave previously, to establish a tractable characterisation for the Cπ-repair. Intuitively, an assertion φ belongs to Cπ-repair if for every conflict $\mathcal{C}$, there is a support $\mathcal{B}$ for φ in the ABox, that dominates $\mathcal{C}$. We provide our characterisation in two separate propositions to handle both directions of equivalence with the original Cπ-repair process.

Proposition 2.2. *Let $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$ be an inconsistent, partially preordered KB. Let $\mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$ be its conflict set and let φ be an assertion. If for all $\mathcal{C} \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$, there is $\mathcal{B} \subseteq \mathcal{A}_{\trianglerighteq}$ such that:*

1. *$\mathcal{B}$ supports φ (as per Definition 2.11), and*
2. *$\mathcal{B}$ dominates $\mathcal{C}$ (as per Definition 2.12),*

then $\varphi \in c\pi(\mathcal{A}_{\trianglerighteq})$.

Proof. *Let $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$ be a partially preordered KB, $\mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$ be its non-empty conflict set, and φ be an assertion. Assume that the following condition (from our proposition) is true, and let us show that $\varphi \in c\pi(\mathcal{A}_{\trianglerighteq})$:*

for all $\mathcal{C} \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$, there is $\mathcal{B} \subseteq \mathcal{A}_{\trianglerighteq}$ such that $\mathcal{B}$ supports φ and $\mathcal{B}$ dominates $\mathcal{C}$ ... (*).

From Definition 2.10, $\varphi \in c\pi(\mathcal{A}_{\trianglerighteq})$ means that for each total preorder $\geq$ compatible with $\trianglerighteq$, we have $\langle \mathcal{T}, cl(\mathcal{R}_\pi(\mathcal{A}_{\geq})) \rangle \models \varphi$, and thus, $\langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_{\geq}) \rangle \models \varphi$. Let $\geq_1$ be a given compatible total preorder and $(\mathcal{S}_1, \ldots, \mathcal{S}_n)$ be its associated well-ordered partition. Let $\mathcal{R}_\pi(\mathcal{A}_{\geq_1}) = (\mathcal{S}_1 \cup \cdots \cup \mathcal{S}_i)$ be the possibilistic repair as given in Definition 1.24.
Now let us show that indeed $\langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_{\geq_1}) \rangle \models \varphi$. Recall from Definition 1.24 that:

- *$\langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_{\geq_1}) \rangle$ is consistent, and;*
- *$\langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_{\geq_1}) \cup \mathcal{S}_{i+1} \rangle$ is inconsistent.*

Hence, there is at least one conflict $\mathcal{C}' \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$ such that $\mathcal{C}' \subseteq (\mathcal{R}_\pi(\mathcal{A}_{\geq_1}) \cup \mathcal{S}_{i+1})$.
By (*): the assertion φ has a support $\mathcal{B}$, such that $\mathcal{B}$ dominates $\mathcal{C}'$. By Definition 2.12, $\mathcal{B}$ dominates $\mathcal{C}'$ means that for all $\phi \in \mathcal{B}$, there is $\psi \in \mathcal{C}'$ such that $\phi \rhd \psi$. Since $\geq_1$ extends $\trianglerighteq$, this also means that for all $\phi \in \mathcal{B}$, there is $\psi \in \mathcal{C}'$ such that $\phi >_1 \psi$.
Therefore, for all $\phi \in \mathcal{B}$, we have $\phi \in \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_i$. Hence, $\mathcal{B} \subseteq \mathcal{R}_\pi(\mathcal{A}_{\geq_1})$. By Definition 2.11, $\langle \mathcal{T}, \mathcal{B} \rangle \models \varphi$, so $\langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_{\geq_1}) \rangle \models \varphi$. Since this result applies to all the compatible preorders, we conclude that $\varphi \in c\pi(\mathcal{A}_{\trianglerighteq})$. □

We illustrate the result of Proposition 2.2 with the following example.

Example 2.13. *Let us use the condition of Proposition 2.2 to check whether the assertions* $\mathsf{User}(\mathsf{Ben})$ *and* $\mathsf{Resource}(\mathsf{r}_1)$ *are in* $c\pi(\mathcal{A}_{\trianglerighteq})$*, as shown by Example 2.10. We recall the conflict set of the ABox* $\mathsf{Cf}(\mathcal{A}_{\trianglerighteq}) = \{\mathcal{C}_1, \mathcal{C}_2\}$*:*

$$\mathcal{C}_1 = \{\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)\}$$
$$\mathcal{C}_2 = \{\mathsf{Secondee}(\mathsf{Duane}), \mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)\}$$

For each conflict we must exhibit a dominant support for both of $\mathsf{User}(\mathsf{Ben})$ *and* $\mathsf{Resource}(\mathsf{r}_1)$*:*

- $\mathsf{User}(\mathsf{Ben})$ *has two supports, namely,* $\{\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)\}$ *and* $\{\mathsf{FullResearcher}(\mathsf{Ben})\}$. $\{\mathsf{FullResearcher}(\mathsf{Ben})\}$ *dominates* $\mathcal{C}_1$ *and* $\{\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)\}$ *dominates* $\mathcal{C}_2$.
- $\mathsf{Resource}(\mathsf{r}_1)$ *has two supports, namely* $\{\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)\}$ *and* $\{\mathsf{Confidential}(\mathsf{r}_1)\}$. $\{\mathsf{Confidential}(\mathsf{r}_1)\}$ *dominates* $\mathcal{C}_1$ *and* $\{\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)\}$ *dominates* $\mathcal{C}_2$.

Therefore, both $\mathsf{User}(\mathsf{Ben})$ *and* $\mathsf{Resource}(\mathsf{r}_1)$ *are in* $c\pi(\mathcal{A}_{\trianglerighteq})$.

The second proposition shows the other direction of Proposition 2.2. It also covers the case of an assertion without a support, which cannot belong to the Cπ-repair.

Proposition 2.3. *Let* $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$ *be an inconsistent, partially preordered KB. Let* $\mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$ *be its conflict set and* φ *be an assertion. If* $\varphi \in c\pi(\mathcal{A}_{\trianglerighteq})$*, then for all* $\mathcal{C} \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$*, there is* $\mathcal{B} \subseteq \mathcal{A}_{\trianglerighteq}$ *such that:*

1. $\mathcal{B}$ *supports* φ *(as per Definition 2.11), and*
2. $\mathcal{B}$ *dominates* $\mathcal{C}$ *(as per Definition 2.12).*

Proof. *Let* $\varphi \in c\pi(\mathcal{A}_{\trianglerighteq})$ *and assume that the following condition is false:*

for all $\mathcal{C} \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$*, there is* $\mathcal{B} \subseteq \mathcal{A}_{\trianglerighteq}$ *such that* $\mathcal{B}$ *supports* φ *and* $\mathcal{B}$ *dominates* $\mathcal{C}$ *. . . (*)*

We distinguish two cases:

1. *There is no support for* φ*. This means that for any total preorder* $\geq$ *compatible with* $\trianglerighteq$*, there will be no support in* $\mathcal{A}_{\geq}$ *(the notion of support is independent of the fact that the ABox is flat, totally preordered or partially preordered). This also means there will be no support in* $\mathcal{R}_\pi(\mathcal{A}_{\geq})$*. Hence, this means that* φ *belongs neither to* $\mathcal{R}_\pi(\mathcal{A}_{\geq})$ *nor to its closure* $cl(\mathcal{R}_\pi(\mathcal{A}_{\geq}))$*. But this contradicts our assumption that* $\varphi \in c\pi(\mathcal{A}_{\trianglerighteq})$*.*
2. *There is at least one support for* φ*. Then, (*) is false means that there is a conflict that is not dominated by any support of* φ*. Let* $\mathcal{C}'$ *be such a conflict. Then:*

for any $\mathcal{B} \subseteq \mathcal{A}_{\trianglerighteq}$ *a support of* $\varphi, \mathcal{B}$ *does not dominate* $\mathcal{C}'$ *. . . (**)*

$\mathcal{B}$ does not dominate $\mathcal{C}'$ means that there is $\phi \in \mathcal{B}$ an assertion such that for all $\psi \in \mathcal{C}'$, either $\psi \rhd \phi$ or $\psi \equiv_{\trianglerighteq} \phi$ or $\psi \bowtie \phi$ (namely, ϕ is not strictly preferred to ψ). $\psi \equiv_{\trianglerighteq} \phi$ means that both $\psi \trianglerighteq \phi$ and $\phi \trianglerighteq \psi$ hold. The idea is then to construct a sequence of partial preorders $(\trianglerighteq_0, \trianglerighteq_1, \ldots, \trianglerighteq_m)$ where $\trianglerighteq_0 = \trianglerighteq$ and $\trianglerighteq_i$ $(i \geq 1)$ is obtained from $\trianglerighteq_{i-1}$ by transforming one case of incomparability between an element ψ of $\mathcal{C}$ and an element ϕ of

a support $\mathcal{B}$ of φ into a preference of ψ over ϕ. The last partial preorder $\trianglerighteq_m$ is such that there is no incomparability between the elements of $\mathcal{C}'$ and of $\mathcal{B}$. One can easily check that each compatible preorder of $\trianglerighteq_i$ is also a compatible preorder of any $\trianglerighteq_j$ (with $j < i$) and so of $\trianglerighteq_0 = \trianglerighteq$. One can also check that the condition (**) holds for each $\trianglerighteq_i$. At step m, $\mathcal{B}$ does not dominate $\mathcal{C}'$ means that there is $\phi \in \mathcal{B}$ such that for all $\psi \in \mathcal{C}'$ we have, either $\psi \trianglerighteq_m \phi$ or $\psi \triangleright_m \phi$ because there is no incomparability relation in $\trianglerighteq_m$. Lastly, let $\geq$ be a total preorder compatible with $\trianglerighteq_m$ and let $(\mathcal{S}_1, \ldots, \mathcal{S}_n)$ be its associated well-ordered partition and $\mathcal{R}_\pi(\mathcal{A}_\geq) = \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_i$ its repair. It is easy to see that there is no support $\mathcal{B}$ of φ such that $\mathcal{B} \subseteq \mathcal{R}_\pi(\mathcal{A}_\geq)$. Indeed, if this were the case, then $\mathcal{C}' \subseteq \mathcal{R}_\pi(\mathcal{A}_\geq)$, which is impossible since $\mathcal{R}_\pi(\mathcal{A}_\geq)$ is consistent. The fact that φ has no support in $\mathcal{R}_\pi(\mathcal{A}_\geq)$ also means that $\varphi \notin cl(\mathcal{R}_\pi(\mathcal{A}_\geq))$. This contradicts our assumption $\varphi \in c\pi(\mathcal{A}_\trianglerighteq)$.

□

The next example illustrates the situation covered by Proposition 2.3.

Example 2.14. *Let us use the assertion* $\mathsf{hasAccess(Ben, r_1)}$ *from our running example and show that it cannot belong to the Cπ-repair $c\pi(\mathcal{A}_\trianglerighteq)$. An assertion is not part of the Cπ-repair if there is a conflict $\mathcal{C} \in \mathsf{Cf}(\mathcal{A}_\trianglerighteq)$ such that none of the supports of the assertion dominates $\mathcal{C}$.*
$\mathsf{hasAccess(Ben, r_1)}$ *has two supports in $\mathcal{A}_\trianglerighteq$, namely, $\mathcal{B}_1 = \{\mathsf{hasOpenAccess(Ben, r_1)}\}$ and $\mathcal{B}_2 = \{\mathsf{hasConfidAccess(Ben, r_1)}\}$. The conflict set of the ABox $\mathsf{Cf}(\mathcal{A}_\trianglerighteq) = \{\mathcal{C}_1, \mathcal{C}_2\}$ is:*

$$\mathcal{C}_1 = \{\mathsf{hasConfidAccess(Ben, r_1)}, \mathsf{hasOpenAccess(Ben, r_1)}\}$$
$$\mathcal{C}_2 = \{\mathsf{Secondee(Duane)}, \mathsf{owns(Duane, r_2)}\}$$

Let us focus on $\mathcal{C}_1$ since $\mathcal{C}_2$ is dominated by $\mathcal{B}_2$. Both the assertions in $\mathcal{C}_1$ are not strictly less than the assertion in $\mathcal{B}_1$, $\mathsf{hasOpenAccess(Ben, r_1)}$ *is in both $\mathcal{C}_1$ and $\mathcal{B}_1$ and* $\mathsf{hasConfidAccess(Ben, r_1)}$ *is incomparable with* $\mathsf{hasOpenAccess(Ben, r_1)}$*, hence $\mathcal{B}_1$ does not dominate $\mathcal{C}_1$.*
Following the proof of Proposition 2.3, we transform the incomparability $\mathsf{hasOpenAccess(Ben, r_1)} \bowtie \mathsf{hasConfidAccess(Ben, r_1)}$ *into the equivalence* $\mathsf{hasOpenAccess(Ben, r_1)} \equiv_{\geq_4} \mathsf{hasConfidAccess(Ben, r_1)}$ *to obtain the total preorder $\geq_4$ as illustrated in Figure 2.8. At this step, neither of the supports $\mathcal{B}_1$ or $\mathcal{B}_2$ dominates the conflict $\mathcal{C}_1$. Moreover, the possibilistic repair of the obtained totally preordered ABox, denoted $\mathcal{R}_\pi(\mathcal{A}_{\geq_4})$ is given by $\mathcal{R}_\pi(\mathcal{A}_{\geq_4}) =$* $\{\mathsf{owns(Duane, r_2)}, \mathsf{FullResearcher(Ben)}, \mathsf{Confidential(r_1)}\}$*. Hence, neither of the supports of* $\mathsf{hasAccess(Ben, r_1)}$ *is in $\mathcal{R}_\pi(\mathcal{A}_{\geq_4})$. Therefore,* $\mathsf{hasAccess(Ben, r_1)}$ *does not belong to $c\pi(\mathcal{A}_\trianglerighteq)$.*

Figure 2.9 summarises the result of Propositions 2.2 and 2.3. Let us now establish the tractability of our method, which is based on the "Dominant supports" characterisation that was given by Propositions 2.2 and 2.3.

Proposition 2.4. *Let $\mathcal{K}_\trianglerighteq = \langle \mathcal{T}, \mathcal{A}_\trianglerighteq \rangle$ be a partially preordered KB and φ be an assertion. Checking if $\varphi \in c\pi(\mathcal{A}_\trianglerighteq)$ is done in polynomial time in DL-Lite$_\mathcal{R}$.*

Proof. *Proofing the tractability of checking if a given assertion φ belongs to $c\pi(\mathcal{A}_\trianglerighteq)$ comes first from our characterisation of Cπ-repair given in Propositions 2.2 and 2.3, which does not enumerate all the compatible totally preordered ABoxes. We recall that this characterisation*

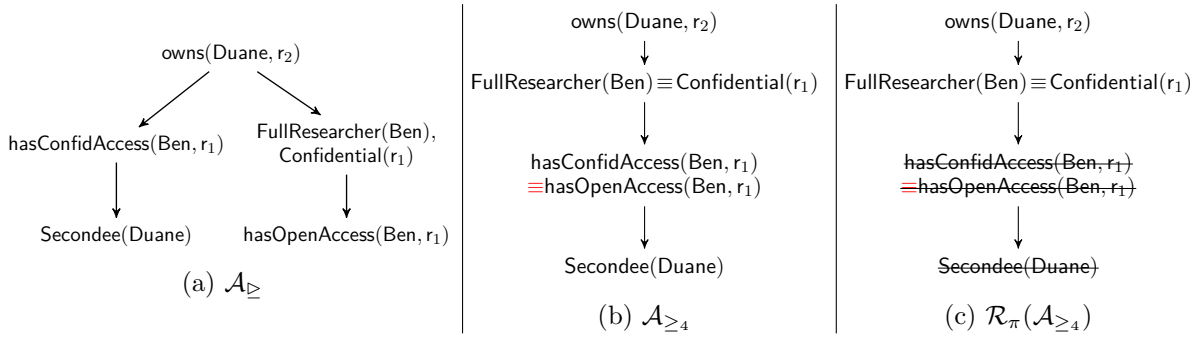


Figure 2.8: The total extension $\geq_4$ of $\trianglerighteq$.

*consists in saying that $\varphi \in c\pi(\mathcal{A}_{\trianglerighteq})$ if and only if, for any conflict $\mathcal{C}$, there is a support $\mathcal{B}$ of φ such that $\mathcal{B}$ dominates $\mathcal{C}$. Note that, in DL-Lite$_\mathcal{R}$, the size of a given conflict is at most two assertions and a given support is one assertion, so checking if $\mathcal{B}$ dominates $\mathcal{C}$ is in $O(1)$ (in DL-Lite$_\mathcal{R}$, computing the conflict set is done in polynomial time [*Calvanese et al. *2007]). Therefore, the number of conflicts and supports is bounded by $O(|\mathcal{A}_{\trianglerighteq}|^2)$. Moreover, the size of the TBox is often considered negligible compared to the size of the ABox, thus the main focus is on data complexity. Now, once the conflict set and all supports are computed, a simple algorithm iterates over the conflict set, and for each conflict $\mathcal{C}$, it parses the set of supports to check if there is some support $\mathcal{B}$ that dominates $\mathcal{C}$. Then the whole process of checking if φ belongs to Cπ-repair is achieved in polynomial time.* □

The tractability of the Cπ-repair method through the use of the "Dominant supports" characterisation completes the results of the first contribution of this thesis. It is important to note that the use of the deductive closure is usually computationally expensive in the literature. This is the case of the ICR semantics [Bienvenu 2012a] which is intractable, unlike its unclosed variant, the IAR semantics. Furthermore, the established characterisation, which is based on checking whether an assertion belongs to the Cπ-repair can be performed by progressively examining the dominance between the supports of the assertion and the conflicts of the ABox. The following section proposes algorithms to compute the Cπ-repair using the characterisation. Then, it studies how the algorithms can be enhanced for better experimental results.

2.3 Algorithmic and Experimental Evaluation

In this section, we present the algorithms we implemented for computing the closure-based partially preordered possibilistic repair (Cπ-repair) and its unclosed counterpart the π-repair. Then, we describe the experimental setting used to evaluate it, as well as the results we obtained.

2.3.1 The Cπ-repair Algorithm

A possible way of computing the Cπ-repair proceeds by applying the dominant supports characterisation on every assertion that follows from the knowledge base. Intuitively, the largest possible Cπ-repair is computed when a knowledge base is consistent (as shown by Lemma 2.2). Hence, the assertions to be checked can be obtained by closing the initial ABox, as given per Definition 2.5. After that, for each assertion, one must check if for every conflict it has a support dominating that conflict. Thus, the next step is to compute the conflict set of the ABox. The

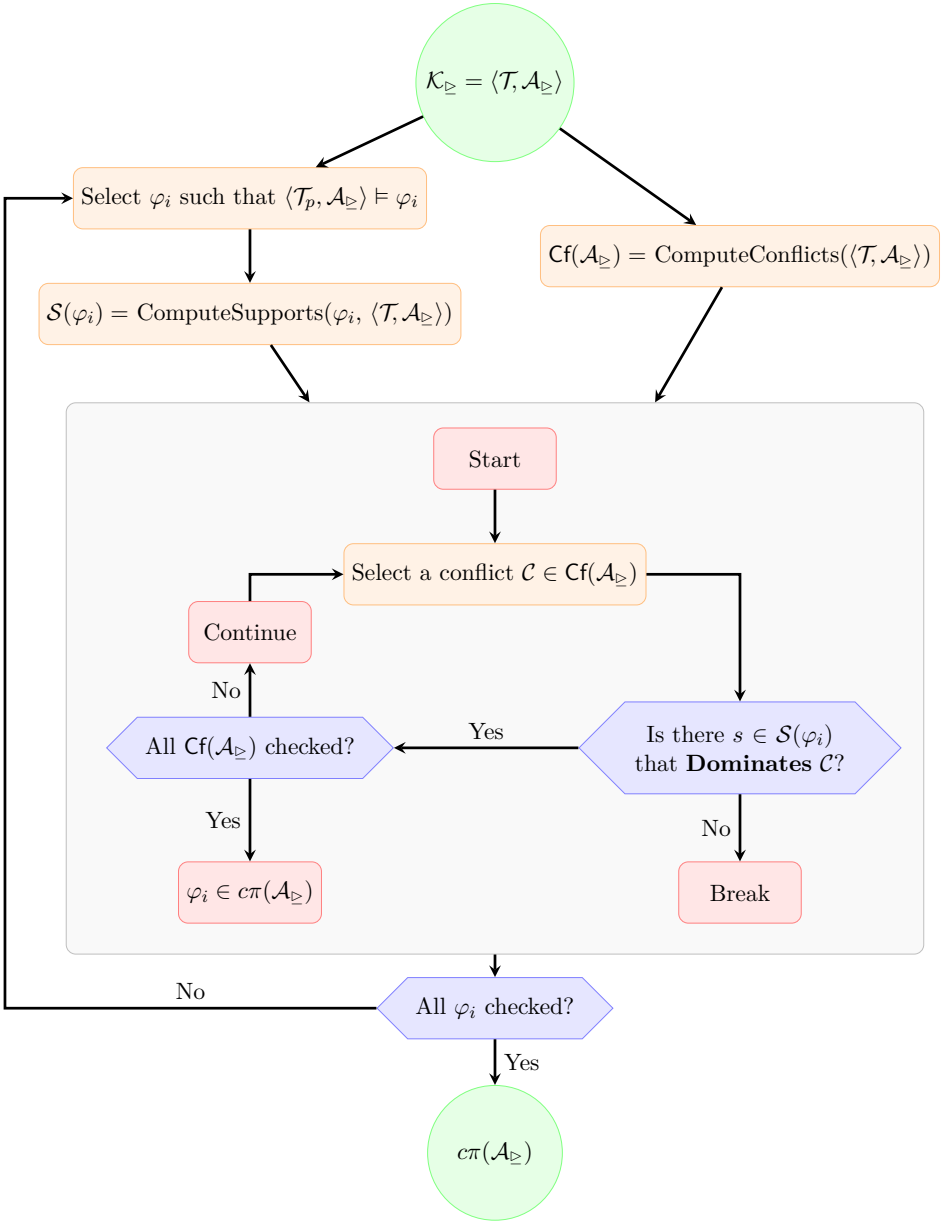


Figure 2.9: The Dominant Supports Characterisation

conflict set can be computed as a preprocessing step once for all assertions. Then, it remains to compute the set of supports of each assertion and check for every conflict if it is dominated by some support (of the assertion) to include it in the Cπ-repair.

In the following, we provide the detailed algorithms to compute the deductive closure of an ABox, as given by Definition 2.5, the conflict set of an ABox as given by Definition 1.9, and the supports of an assertion as given by Definition 2.11.

Computing the deductive closure

Since the deductive closure consists of the assertions that are entailed by the ABox and the positive axioms of the TBox, a possible procedure of computing it goes by constructing single-atom queries from each of the concept and role names appearing in the TBox. Each of these queries is then reformulated using the PerfectRef algorithm to get queries of all possible entailers of the evaluated concept or role name. Each of the queries is then evaluated on the database-like interpretation of the ABox, and the returned answers are used to build an assertion of the initial atomic concept or role, such an assertion is in the closure of the ABox. The following Algorithm 2 depicts the steps of computing the deductive closure using the described method.

Computing the conflict set

Several research efforts proposed an algorithm of computing the conflict set for inconsistent DL-Lite$_\mathcal{R}$ knowledge bases. We follow the same procedure as given in [Bienvenu *et al.* 2019]. In order to compute the set of conflicts of an ABox, the non-Boolean counterpart of the consistency checking query can be used. Therefore, the consistency checking query q_{unsat} is used with free variables to return the conflicts. Indeed, given a knowledge base $\langle \mathcal{T}, \mathcal{A} \rangle$ and its consistency checking query q_{unsat} the images of the conjunctive queries in q_{unsat} are the conflicts in $\mathsf{Cf}(\mathcal{A})$. The Algorithm 3 details the process of exhibiting the conflict set $\mathsf{Cf}(\mathcal{A})$, where:

- $\delta(.)$ is a translation function from each negative inclusion axiom to a conjunctive query as follows:
$$\delta(B_1 \sqsubseteq \neg\, B_2) = prenex(\exists x.\gamma_1(x) \wedge \gamma_2(x))$$
$$\delta(R_1 \sqsubseteq \neg\, R_2) = prenex(\exists x, y.\rho_1(x, y) \wedge \rho_2(x, y))$$
where $\gamma_i(x) = A_i(x)$ if $B_i = A_i$, $\gamma_i(x) = \exists y_i.P_i(x, y_i)$ if $B_i = \exists P_i$, and $\gamma_i(x) = \exists y_i.P_i(y_i, x)$ if $B_i = \exists P_i^-$; and $\rho_i(x, y) = P_i(x, y)$ if $R_i = P_i$, and $\rho_i(x, y) = P_i(y, x)$ if $R_i = P_i^-$ [Calvanese *et al.* 2007].

- prenex is defined by: $\mathsf{prenex}(\exists \vec{x}(\exists \vec{y} g(\vec{x}, \vec{y}) \wedge \exists \vec{y_0} g(\vec{x}, \vec{y_0}))) = \exists \vec{x} \exists \vec{y} \exists \vec{z}(g(\vec{x}, \vec{y}) \wedge g(\vec{x}, \vec{z}))with\vec{z} \cap \vec{y} = \emptyset$ [Bourgaux 2016].

- atoms$(\psi(\vec{a}))$ is a function that extracts the set of ground atomic facts (i.e., instantiated atoms) from the formula ψ after substituting the variables $\vec{y}$ with the constants $\vec{a}$.

Algorithm 3 proceeds by generating and rewriting the conjunctive queries that check for possible violations of all the negative inclusion axioms entailed by the TBox $\mathcal{T}$ in lines 2 and 3. The conjunctive queries are all of size two $|q| = 2$ (contain two atoms), because two concepts (or roles), at maximum, are involved in a negative inclusion axiom. The number of reformulations returned by each call to PerfectRef is $|\mathcal{T}|^{|q|}$ in the worst case, hence $|\mathcal{T}|^2$. The number of conjunctive queries in q_{unsat} is bounded by $O(|\mathcal{T}|^2 * |\mathcal{T}_n|)$. The algorithm then proceeds by evaluating each of the conjunctive queries in q_{unsat} on the database-like interpretation of the

Algorithm 2: ComputeClosure

Input: $\mathcal{T}$: a TBox, $db(\mathcal{A})$: the database-like interpretation of $\mathcal{A}$
Output: The deductive closure $cl(\mathcal{A})$

```
1  cl(A) ← ∅;
2  foreach B: a concept name in T do
3      q(x⃗) ← ∃x.A(x);
4      q_T(x⃗) ← PerfectRef(q(x⃗), T);
5      foreach ∃x⃗ψ(x⃗) ∈ q_T(x⃗) do
6          foreach a⃗ ∈ ans(ψ, db(A)) do
7              if atoms(B(a⃗)) ∉ cl(A) then
8                  cl(A) ← cl(A) ∪ {atoms(B(a⃗))};
9  foreach R: a role name in T do
10     q(x⃗) ← ∃x1, x2.R(x1, x2);
11     q_T(x⃗) ← PerfectRef(q(x⃗), T);
12     foreach ∃x⃗ψ(x⃗) ∈ q_T(x⃗) do
13         if arity(∃x⃗ψ(x⃗)) = 2 then
14             foreach a⃗ ∈ ans(ψ, db(A)) do
15                 if atoms(R(a⃗)) ∉ cl(A) then
16                     cl(A) ← cl(A) ∪ {atoms(R(a⃗))};
17 return cl(A);
```

Algorithm 3: ComputeConflicts

Input: $\mathcal{T}$: a TBox (where $\mathcal{T} = \mathcal{T}_n \cup \mathcal{T}_p$), $db(\mathcal{A})$: a database-like interpretation of $\mathcal{A}$
Output: The conflict set $\mathsf{Cf}(\mathcal{A})$

```
1  q_unsat ← ⊥;
2  foreach α ∈ T_n do
3      q_unsat ← q_unsat ∨ PerfectRef(δ(α), T);
4  Cf(A) = ∅;
5  foreach ∃y⃗ψ(y⃗) ∈ q_unsat do
6      foreach a⃗ ∈ ans(ψ,db(A)) do
7          Cf(A) ← Cf(A) ∪ {atoms(ψ(a⃗))};
8  foreach C ∈ Cf(A) do
9      foreach C′ ∈ Cf(A) do
10         if C′ ⊆ C then
11             Cf(A) ← Cf(A) \ C;
12 return Cf(A);
```

ABox in lines 5, 6, and 7. Since the size of each conjunctive query in q_{unsat} is 2, the number of the answers returned in line 6 is polynomial in the ABox's size in the worst case, i.e., bounded by $|\mathcal{A}|^2$. The final steps in lines 8 to 11 aims at removing the non-minimal answers. For instance, if the following axiom $\exists R^- \sqsubseteq \neg\, \exists R$ is inferred by the TBox, and the ABox contains the assertions $R(a,a), R(a,b)$, the method without these steps (8 to 11) would return both the sets $\{R(a,a), R(a,b)\}$ and $\{R(a,a)\}$, however due to the minimality condition, only $\{R(a,a)\}$ is a conflict.

Computing the supports of an assertion

In order to compute the supports of a given assertion, we proceed in an instance checking style, where all reformulations of the instance query of the assertion are verified against the ABox, and those returning true are returned by the algorithm are returned as a support. Given an assertion φ, the reformulations of the Boolean query of its instance checking query are computed using the PerfectRef algorithm (Algorithm 1). The resulting re-writings are then evaluated on the database-like interpretation of the ABox.

Algorithm 4: ComputeSupports

Input: $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$: a KB, $B(a)$, resp. $R(a_1, a_2)$: an assertion in $cl(\mathcal{A}_{\trianglerighteq})$
Output: $\mathcal{S}$: the supports of $B(a)$, resp. $R(a_1, a_2)$, in $\mathcal{A}_{\trianglerighteq}$

1 $\mathcal{S} \leftarrow \emptyset$;
2 $q(\overrightarrow{a}) \leftarrow B(a)$ // resp. $R(a_1, a_2)$;
3 PR $\leftarrow$ PerfectRef$(q, \mathcal{T})$;
4 **foreach** $q_i(\overrightarrow{a})$ *in* PR **do**
5 **if** $ans(q_i, \mathcal{I}_{\mathcal{A}_{\trianglerighteq}}) = true$ **then**
6 $\mathcal{S} \leftarrow \mathcal{S} \cup \{\{\text{atoms}(q_i(\overrightarrow{a}))\}\}$;
7 **return** $(\mathcal{S})$

A first algorithm to compute the Cπ-repair

Provided the algorithms used to compute the deductive closure of the initial ABox, its conflict set, and the supports of an assertion, one can now give the algorithm used to compute the Cπ-repair. Algorithm 5 (next page) directly goes through every assertion in $cl(\mathcal{A}_{\trianglerighteq})$ (step 4), computes its set of supports (step 5), then checks for each conflict of the ABox in Cf$(\mathcal{A}_{\trianglerighteq})$ if there is a support that dominates it (steps 6 and 7).

Note that, at this level, we assume that the dominance checking can be done efficiently by comparing the reliability of assertions. In the experimental evaluation part, we will provide the actual way of implementing the partial preorder defined over the ABox $\mathcal{A}_{\trianglerighteq}$, and discuss the possible ways to achieve it.

The following lemma establishes the complexity of applying Algorithm 5 to compute the Cπ-repair.

Lemma 2.3. *Algorithm 5 runs in polynomial time in $|\mathcal{A}_{\trianglerighteq}|$ in data complexity.*

Proof. *The closure computation in line 2 is done in linear time in $|\mathcal{A}_{\trianglerighteq}|$ in data complexity, and the number of the assertions in the closure is linear in $|\mathcal{A}_{\trianglerighteq}|$. Computing the set of conflicts in line 3 is done in polynomial time in $|\mathcal{A}_{\trianglerighteq}|$ and the size of the resulting conflict set is also*

Algorithm 5: ComputeCπ-repair (naive)

Input: $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$ // A knowledge base
Output: $c\pi(\mathcal{A}_{\trianglerighteq})$ // The Cπ-repair of $\mathcal{A}_{\trianglerighteq}$

$c\pi(\mathcal{A}_{\trianglerighteq}) \leftarrow \emptyset$ // Initialize the Cπ-repair
$cl(\mathcal{A}_{\trianglerighteq}) \leftarrow \mathsf{ComputeClosure}(\mathcal{K})$ // Compute the closure of $\mathcal{A}_{\trianglerighteq}$
$\mathsf{Cf}(\mathcal{A}_{\trianglerighteq}) \leftarrow \mathsf{ComputeConflicts}(\mathcal{K})$ // Compute the conflict set of $\mathcal{A}_{\trianglerighteq}$
if $\mathsf{Cf}(\mathcal{A}_{\trianglerighteq}) = \emptyset$ **then**
  return $cl(\mathcal{A}_{\trianglerighteq})$
else
  foreach $\varphi \in cl(\mathcal{A}_{\trianglerighteq})$ **do**
    $\mathcal{S}(\varphi) \leftarrow \mathsf{ComputeSupports}(\varphi, \mathcal{K})$ // Compute all supports of φ
    $\mathsf{visited} \leftarrow \emptyset$ // visited: the set of visited conflicts
    repeat
      select $\mathcal{C} = \{c_i, c_j\}$ from $\mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$
      $\mathsf{Cf}(\mathcal{A}_{\trianglerighteq}) \leftarrow \mathsf{Cf}(\mathcal{A}_{\trianglerighteq}) \setminus \{\mathcal{C}\}$
      $\mathsf{visited} \leftarrow \mathsf{visited} \cup \{\mathcal{C}\}$
      $\mathsf{supported} \leftarrow \mathrm{false}$ // supported: a boolean, true if φ has a support
      repeat
        select $\{s\}$ from $\mathcal{S}(\varphi)$
        $\mathcal{S}(\varphi) \leftarrow \mathcal{S}(\varphi) \setminus \{s\}$
        if $s \rhd c_i$ *or* $s \rhd c_j$ **then**
          $\mathsf{supported} \leftarrow \mathrm{true}$
      until $\mathsf{supported} = \mathit{true}$ *or* $\mathcal{S}(\varphi) = \emptyset$;
    until $\mathsf{supported}$ *is false or* $\mathsf{Cf}(\mathcal{A}_{\trianglerighteq}) = \emptyset$ // exit after parsing all the conflicts or there is no support;
    if $\mathsf{supported} = \mathit{true}$ **then**
      $C\pi(\mathcal{A}_{\trianglerighteq}) \leftarrow C\pi(\mathcal{A}_{\trianglerighteq}) \cup \{\varphi\}$ // φ is supported against each conflict
    $\mathsf{Cf}(\mathcal{A}_{\trianglerighteq}) \leftarrow \mathsf{Cf}(\mathcal{A}_{\trianglerighteq}) \cup \mathsf{visited}$
  return $c\pi(\mathcal{A}_{\trianglerighteq})$

*polynomial ($\mathcal{O}(|\mathcal{A}_{\trianglerighteq}|^2)$) in the size of the ABox [*Bienvenu et al. *2019]. The time complexity of the algorithm is determined by the loop in lines 4 to 7. The loop in line 4 iterates over the assertions in the closure. For each assertion, it computes the supports in constant time in $|\mathcal{A}_{\trianglerighteq}|$, and the number of supports is also constant in $|\mathcal{A}_{\trianglerighteq}|$. The condition in line 6 iterates over all the conflicts in the worst case. Since the number of the supports does not depend on $|\mathcal{A}_{\trianglerighteq}|$, checking the existence of a support $\exists \mathcal{B} \in \mathcal{S}(\varphi)$ in line 6 also does not depend on $|\mathcal{A}_{\trianglerighteq}|$. Checking if a support dominates a conflict is done in constant time in $|\mathcal{A}_{\trianglerighteq}|$. The rest of the lines are done in constant time per iteration. Therefore, the overall time complexity of the algorithm is polynomial in the size of the ABox $\mathcal{O}(|\mathcal{A}_{\trianglerighteq}|^3)$, confirming that Algorithm 5 runs in polynomial time in the size of the ABox (in data complexity).* □

2.3.2 Revisiting the Cπ-repair Algorithm

The procedure given by Algorithm 5 is a tractable way of computing the Cπ-repair. However, one can notice that it is a straightforward application of the characterisation and that some improvements can be made to achieve a more efficient algorithm. This involves two main observations. The first observation is related to the sets of conflicts and supports. Indeed, the dominance relation is based on the partial preorder, which is a transitive relation; hence, the dominance is also transitive. This allows us to apply the characterisation by checking only dominant sets of conflicts and supports. The second observation is the fact that the π-repair $\pi(\mathcal{A}_{\trianglerighteq})$ is included in the Cπ-repair, hence one can progressively compute the Cπ-repair by first computing the π-repair, then applying the dominant supports characterisation to the remaining assertions.

Problem Statement

Can we take advantage of the dominance relation and the inclusion of the π-repair of an ABox in its Cπ-repair to achieve a better performing algorithm?

Dominant conflicts. One can easily check that the dominance relation (Definition 2.12) is a transitive relation, namely, if $\mathcal{B}_1$ dominates $\mathcal{B}_2$ and $\mathcal{B}_2$ dominates $\mathcal{B}_3$ then $\mathcal{B}_1$ dominates $\mathcal{B}_3$. Indeed, assume that $\mathcal{B}_1$ dominates $\mathcal{B}_2$ and let ϕ be an assertion in $\mathcal{B}_1$. $\mathcal{B}_1$ dominates $\mathcal{B}_2$ means that $\exists \psi \in \mathcal{B}_2$ s.t. $\phi \rhd \psi$. Now assume also that $\mathcal{B}_2$ dominates $\mathcal{B}_3$ which means that $\forall \psi \in \mathcal{B}_2$, $\exists \gamma \in \mathcal{B}_3$ such that $\psi \rhd \gamma$. As $\rhd$ is a transitive relation, having $\phi \rhd \gamma$, this implies that $\mathcal{B}_1$ dominates $\mathcal{B}_3$.

Definition 2.13 (Dominant conflicts)**.** *The set of dominant conflicts, denoted $\mathsf{Cf}_{dom}(\mathcal{A}_{\trianglerighteq})$, is the maximal subset $\mathsf{Cf}_{dom}(\mathcal{A}_{\trianglerighteq}) \subseteq \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$ such that $\forall \mathcal{C}_i \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$, if $\exists \mathcal{C}_j \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$ such that $\mathcal{C}_j$ dominates $\mathcal{C}_i$, then $\mathcal{C}_i \notin \mathsf{Cf}_{dom}(\mathcal{A}_{\trianglerighteq})$.*

The following proposition states that it is sufficient to consider only the dominant conflicts within the conflict set to compute the Cπ-repair. This stems from the fact that the dominance relation is transitive, which follows from the transitivity of the partial preorder relation $\trianglerighteq$.

Proposition 2.5. *A given assertion φ is in $c\pi(\mathcal{A}_{\trianglerighteq})$ if and only if $\forall \mathcal{C} \in \mathsf{Cf}_{dom}(\mathcal{A}_{\trianglerighteq})$, $\exists \mathcal{B} \subseteq \mathcal{A}_{\trianglerighteq}$ such that $\mathcal{B}$ supports φ and $\mathcal{B}$ dominates $\mathcal{C}$.*

Proof. *Let φ be an assertion such that $\varphi \in c\pi(\mathcal{A}_{\trianglerighteq})$. This means that:*

$$\forall \mathcal{C} \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq}), \exists \mathcal{B} \text{ a support of } \varphi \text{ in } \mathcal{A}_{\trianglerighteq} \text{ such that } \mathcal{B} \text{ dominates } \mathcal{C} \ldots (^1)$$

Let $\mathsf{Cf}_{dom}(\mathcal{A}_{\trianglerighteq})$ be the set of dominant conflicts as per Definition 2.13. It is straightforward that if $\mathcal{C}$ is in both $\mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$ and $\mathsf{Cf}_{dom}(\mathcal{A}_{\trianglerighteq})$ then $\mathcal{C}$ is verified by Lemma 2.5. Now, let $\mathcal{C}_i$ be a conflict such that $\mathcal{C}_i \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$ but $\mathcal{C}_i \notin \mathsf{Cf}_{dom}(\mathcal{A}_{\trianglerighteq})$. This means that $\exists \mathcal{C}_j \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$ such that $\mathcal{C}_j$ dominates $\mathcal{C}_i$.
From $(^1)$, $\exists \mathcal{B}$ a support of φ such that $\mathcal{B}$ dominates $\mathcal{C}_j$. By transitivity, $\mathcal{B}$ dominates $\mathcal{C}_j$ and $\mathcal{C}_j$ dominates $\mathcal{C}_i$ means that $\mathcal{B}$ also dominates $\mathcal{C}_i$. Therefore, it is enough to verify that:

$$\mathcal{B} \text{ dominates } \mathcal{C}_j \ldots (^2)$$

Now, let φ be an assertion such that $\varphi \notin c\pi(\mathcal{A}_{\trianglerighteq})$. This means that there is at least one $\mathcal{C} \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$ such that $\forall \mathcal{B}$ a support of φ in $\mathcal{A}_{\trianglerighteq}$, $\mathcal{B}$ dominates $\mathcal{C}$ does not hold. Assume that $\mathcal{C}_i$ is a conflict such that $\forall \mathcal{B}$ a support of φ, $\mathcal{B}$ does not dominate $\mathcal{C}_i$ and $\mathcal{C}_i \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$ but $\mathcal{C}_i \notin \mathsf{Cf}_{dom}(\mathcal{A}_{\trianglerighteq})$. This means that $\exists \mathcal{C}_j \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$ such that $\mathcal{C}_j$ dominates $\mathcal{C}_i$. Assume now that $\exists \mathcal{B}$ a support of φ in $\mathcal{A}_{\trianglerighteq}$ such that $\mathcal{B}$ dominates $\mathcal{C}_j$, by the transitivity of the dominance relation, $\mathcal{B}$ also dominates $\mathcal{C}_i$. This contradicts our first assumption. Thus:

$$\text{if } \mathcal{C}_j \text{ dominates } \mathcal{C}_i \text{ and } \forall \mathcal{B} \text{ a support of } \varphi, \mathcal{B} \text{ does not dominate } \mathcal{C}_i,$$
$$\text{then } \mathcal{B} \text{ does not dominate } \mathcal{C}_j \ldots (^3)$$

From $(^2)$ and $(^3)$ we conclude that it is enough to consider the conflicts in $\mathsf{Cf}_{dom}(\mathcal{A}_{\trianglerighteq})$ in order to check if an assertion φ is in $c\pi(\mathcal{A}_{\trianglerighteq})$. □

Dominant supports.

Definition 2.14 (Dominant supports)**.** *Let $\mathcal{S}(\varphi)$ be the set of supports of an assertion φ in $\mathcal{A}_{\trianglerighteq}$. The set of dominant supports of φ, denoted $\mathcal{S}_{dom}(\varphi)$, is the maximal subset $\mathcal{S}_{dom}(\varphi) \subseteq \mathcal{S}(\varphi)$ such that $\forall \mathcal{B}_i \in \mathcal{S}(\varphi)$, if $\exists \mathcal{B}_j \in \mathcal{S}(\varphi)$ such that $\mathcal{B}_j$ dominates $\mathcal{B}_i$, then $\mathcal{B}_i \notin \mathcal{S}_{dom}(\varphi)$.*

The following proposition states that it is sufficient to consider only the set of dominant supports of any given assertion φ in order to check its membership in Cπ-repair. This result also follows from the transitivity of the partial preorder relation $\trianglerighteq$.

Proposition 2.6. *Let φ be an assertion and let $\mathcal{S}(\varphi)$ be its set of supports in $\mathcal{A}_{\trianglerighteq}$. φ is in $c\pi(\mathcal{A}_{\trianglerighteq})$ if and only if $\forall \mathcal{C} \in \mathsf{Cf}_{dom}(\mathcal{A}_{\trianglerighteq})$, $\exists \mathcal{B} \in \mathcal{S}_{dom}(\varphi)$ such that $\mathcal{B}$ dominates $\mathcal{C}$.*

Proof. *In the same way of showing that it is enough to only consider the set of the dominant conflicts to compute the Cπ-repair. We show that for any given assertion φ, it is enough to consider the set of its dominant supports to check if it belongs to the Cπ-repair. Let φ be an assertion and $\mathcal{S}(\varphi)$ be the set of its supports in $\mathcal{A}_{\trianglerighteq}$. Assume $\varphi \in c\pi(\mathcal{A}_{\trianglerighteq})$. This means that $\forall \mathcal{C} \in \mathsf{Cf}_{dom}(\mathcal{A}_{\trianglerighteq})$, $\exists \mathcal{B} \in \mathcal{S}(\varphi)$ s.t. $\mathcal{B}$ dominates $\mathcal{C}$. Let $\mathcal{S}_{dom}(\varphi)$ be the set of dominant supports of φ. It is clear that if $\mathcal{B} \in \mathcal{S}(\varphi)$ and $\mathcal{B} \in \mathcal{S}_{dom}(\varphi)$ then Lemma 2.6 is verified for $\mathcal{B}$. Now assume $\mathcal{B}_i \in \mathcal{S}(\varphi)$ but $\mathcal{B}_i \notin \mathcal{S}_{dom}(\varphi)$, this means that $\exists \mathcal{B}_j \in \mathcal{S}_{dom}(\varphi)$ such that $\mathcal{B}_j$ dominates $\mathcal{B}_i$. Assume that there is a conflict $\mathcal{C}$ s.t. $\mathcal{B}_i$ dominates $\mathcal{C}$. It also follows that $\mathcal{B}_j$ dominates $\mathcal{C}$ (by the transitivity of the dominance relation). Any assertion that is*

not in the repair $\varphi \notin c\pi(\mathcal{A}_{\trianglerighteq})$ computed by considering the whole set of its supports cannot belong to the repair when considering the set of its dominant supports $\mathcal{S}_{dom}(\varphi)$, because $\mathcal{S}_{dom}(\varphi) \subseteq \mathcal{S}(\varphi)$. □

Properties of the Cπ-repair. The following property characterises the difference between the closure of the π-repair $cl(\pi(\mathcal{A}_{\trianglerighteq}))$ and the Cπ-repair $c\pi(\mathcal{A}_{\trianglerighteq})$. In essence, if an assertion is neither accepted in the π-repair nor its closure, it must have strictly more than one support to belong to the Cπ-repair. This is formalised in the following lemma.

Lemma 2.4. *Let φ be an assertion in $c\pi(\mathcal{A}_{\trianglerighteq})$. Let $\mathcal{S}(\varphi)$ be the set of supports of φ. If $|\mathcal{S}(\varphi)| = 1$, then $\varphi \in cl(\pi(\mathcal{A}_{\trianglerighteq}))$.*

Proof. *The proof here is straightforward. Assume that $\varphi \in c\pi(\mathcal{A}_{\trianglerighteq})$ and that the number of the supports of φ is 1 with $\mathcal{S}(\varphi) = \{\phi\}$ and let us show that indeed, $\varphi \in cl(\pi(\mathcal{A}_{\trianglerighteq}))$. $\varphi \in c\pi(\mathcal{A}_{\trianglerighteq})$ means that $\forall \mathcal{C} \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$, $\{\phi\}$ dominates $\mathcal{C}$. $\{\phi\}$ dominates $\mathcal{C}$ means that $\exists \psi \in \mathcal{C}$ s.t. $\phi \rhd \psi$ (this applies to all conflicts). This collapses with the definition of the π-accepted assertions, an assertion is π-accepted if it is strictly more certain than at least a member of each conflict. Thus, $\phi \in \pi(\mathcal{A}_{\trianglerighteq})$. $\{\phi\}$ supports φ means that $\langle \mathcal{T}, \{\phi\}\rangle \models \varphi$, therefore, $\varphi \in cl(\pi(\mathcal{A}_{\trianglerighteq}))$.* □

The converse of Lemma 2.4 does not hold, a non-accepted assertion may have multiple supports.

Figure 2.10 illustrates the inclusion between the Cπ-repair, the closure of the π-repair and the π-repair itself.

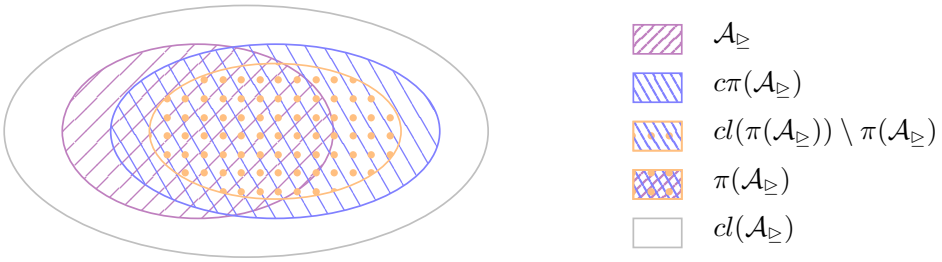


Figure 2.10: Distribution of the possibilistic repairs w.r.t the ABox $\mathcal{A}_{\trianglerighteq}$ and its deductive closure.

Before putting everything together to provide the enhanced version of Algorithm 5, we provide the algorithm to compute the π-repair, which is used in the process of progressively computing the Cπ-repair afterwards. The most efficient way of computing the π-repair has been established in [Belabbes & Benferhat 2022], and relies on $|\mathcal{A}_{\trianglerighteq}|$ consistency checks of a subset of the ABox to render the π-accepted assertions which constitute the π-repair, and which were given by Definition 2.3. The following algorithm details its computation.

Algorithm 6: Computeπ-repair

Input: DL-Lite$_\mathcal{R}$ KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}_\trianglerighteq \rangle$
Output: $\pi(\mathcal{A}_\trianglerighteq)$: the set of π-accepted assertions of $\mathcal{A}_\trianglerighteq$

1 $\pi(\mathcal{A}_\trianglerighteq) \leftarrow \emptyset$;
2 **foreach** φ *in* $\mathcal{A}_\trianglerighteq$ **do**
3 $\quad \Delta(\varphi) \leftarrow \mathcal{A}_\trianglerighteq \setminus \{\phi \mid \phi \in \mathcal{A}_\trianglerighteq \text{ s.t. } \varphi \rhd \phi\}$;
4 $\quad$ **if** Consistent$(\mathcal{T}, \{\varphi\} \cup \Delta(\varphi))$ **then** $\pi(\mathcal{A}_\trianglerighteq) \leftarrow \pi(\mathcal{A}_\trianglerighteq) \cup \{\varphi\}$;
5 **return** $\pi(\mathcal{A}_\trianglerighteq)$;

Lemma 2.5. *The algorithm (*Computeπ*-repair) runs in polynomial time and in linear space in $|\mathcal{A}_\trianglerighteq|$ (data complexity).*

Proof. *The time complexity of the algorithm is determined by the loop of line 2 which iterates over each assertion in $\mathcal{A}_\trianglerighteq$. The set Δ is computed in linear time in $|\mathcal{A}_\trianglerighteq|$. The consistency check in line 4 is equivalent to a FOL query evaluation, which means it can be checked in constant time for each assertion. Therefore, the overall time complexity of the algorithm is polynomial in $|\mathcal{A}_\trianglerighteq|$ in data complexity. The algorithm uses two data structures, $\pi(\mathcal{A}_\trianglerighteq)$ and Δ, both are linear in $|\mathcal{A}_\trianglerighteq|$ in the worst case, as they can potentially store all assertions. Therefore, the overall space complexity of the algorithm is linear in $|\mathcal{A}_\trianglerighteq|$ in data complexity.* □

Now, Algorithm 7 (next page) details the incremental computation of the Cπ-repair, relying on the previous results on dominant conflicts, dominant supports, and the inclusion of the π-repair in the Cπ-repair. Line 1 initialises the Cπ-repair with the assertions that are accepted in the π-repair. Line 2 computes the closure of the ABox, then line 3 removes the assertions that are already accepted to avoid duplicate checks. Line 4 computes the set of dominant conflicts of the ABox, as given per Definition 2.13, note that their computation can be achieved by computing the conflict set using Algorithm 3, then going through the returned set to remove dominated conflicts. The loop in line 5 then goes through the assertions remaining in $cl(\mathcal{A}_\trianglerighteq)$. It computes for each assertion its set of dominant supports as given by Definition 2.14 (their computation can be achieved in the same way dominant conflicts are achieved). Then verifies first, in lines 7 and 8, if some support is already in the Cπ-repair, either from a previous verification or in the π-repair. If that is not the case, it checks if the assertion has more than one support (condition of Lemma 2.4) in order to apply the dominant supports characterisation on it using the set of dominant conflicts and supports, in lines 11 and 12.

The following lemma establishes the complexity of applying Algorithm 7 to compute the Cπ-repair.

Lemma 2.6. *Algorithm 7 runs in polynomial time in $|\mathcal{A}_\trianglerighteq|$ (data complexity).*

Proof. *The proof of this lemma follows from the proofs of Lemmas 2.3 and 2.5. Indeed, the closure computation is the same in Algorithm 5. Computing dominant conflicts is in polynomial time and space in $|\mathcal{A}_\trianglerighteq|$. Computing the π-repair is done in polynomial time and the size of the repair is linear in $|\mathcal{A}_\trianglerighteq|$. Steps 6 to 8 are done in constant time in $|\mathcal{A}_\trianglerighteq|$. The time complexity of the algorithm is determined by the loop in line 5. In the worst case, the π-repair is empty and no dominance is found between the conflicts and between the supports of each assertion, thus, the algorithm amounts to running all the lines of Algorithm 5. Therefore, in the worst case,*

Algorithm 7: ComputeCπ-repair (revisited)

Input: $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$ // A knowledge base
Output: $c\pi(\mathcal{A}_{\trianglerighteq})$ // The Cπ-repair of $\mathcal{A}_{\trianglerighteq}$

1 $c\pi(\mathcal{A}_{\trianglerighteq}) \leftarrow$ Computeπ-repair$(\mathcal{K})$ // Initialize the Cπ-repair with the π-repair
2 $cl(\mathcal{A}_{\trianglerighteq}) \leftarrow$ ComputeClosure$(\mathcal{K})$ // Compute the closure of $\mathcal{A}_{\trianglerighteq}$
3 $cl(\mathcal{A}_{\trianglerighteq}) \leftarrow cl(\mathcal{A}_{\trianglerighteq}) \setminus c\pi(\mathcal{A}_{\trianglerighteq})$
4 $\mathsf{Cf}_{dom}(\mathcal{A}_{\trianglerighteq}) \leftarrow$ ComputeDominantConflicts$(\mathcal{K})$ // Compute the conflict set of $\mathcal{A}_{\trianglerighteq}$
5 **if** $\mathsf{Cf}_{dom}(\mathcal{A}_{\trianglerighteq}) = \emptyset$ **then**
6 **return** $cl(\mathcal{A}_{\trianglerighteq})$
7 **else**
8 **foreach** $\varphi \in cl(\mathcal{A}_{\trianglerighteq})$ **do**
9 $\mathcal{S}_{dom}(\varphi) \leftarrow$ ComputeDominantSupports$(\varphi, \mathcal{K})$ // Compute all supports of φ
10 **if** $\mathcal{S}_{dom}(\varphi) \cap c\pi(\mathcal{A}_{\trianglerighteq}) \neq \emptyset$ **then**
11 $c\pi(\mathcal{A}_{\trianglerighteq}) \leftarrow c\pi(\mathcal{A}_{\trianglerighteq}) \cup \{\varphi\}$ // Checking if it has an accepted support
12 **else**
13 **if** $|\mathcal{S}_{dom}(\varphi)| > 1$ **then**
14 visited $\leftarrow \emptyset$ // visited: the set of visited conflicts
15 **repeat**
16 select $\mathcal{C} = \{c_i, c_j\}$ from $\mathsf{Cf}_{dom}(\mathcal{A}_{\trianglerighteq})$
17 $\mathsf{Cf}_{dom}(\mathcal{A}_{\trianglerighteq}) \leftarrow \mathsf{Cf}_{dom}(\mathcal{A}_{\trianglerighteq}) \setminus \{\mathcal{C}\}$
18 visited $\leftarrow$ visited $\cup$ $\{\mathcal{C}\}$
19 supported $\leftarrow$ false // supported: a boolean, true if φ has a support
20 **repeat**
21 select $\{s\}$ from $\mathcal{S}_{dom}(\varphi)$
22 $\mathcal{S}_{dom}(\varphi) \leftarrow \mathcal{S}_{dom}(\varphi) \setminus \{s\}$
23 **if** $s \rhd c_i$ *or* $s \rhd c_j$ **then**
24 supported $\leftarrow$ true
25 **until** supported $=$ *true or* $\mathcal{S}_{dom}(\varphi) = \emptyset$;
26 **until** supported *is false or* $\mathsf{Cf}_{dom}(\mathcal{A}_{\trianglerighteq}) = \emptyset$ // exit after parsing all the conflicts or there is no support;
27 **if** supported $=$ *true* **then**
28 $C\pi(\mathcal{A}_{\trianglerighteq}) \leftarrow C\pi(\mathcal{A}_{\trianglerighteq}) \cup \{\varphi\}$ // φ is supported against each conflict
29 $\mathsf{Cf}_{dom}(\mathcal{A}_{\trianglerighteq}) \leftarrow \mathsf{Cf}_{dom}(\mathcal{A}_{\trianglerighteq}) \cup$ visited
30 **return** $c\pi(\mathcal{A}_{\trianglerighteq})$

the overall time complexity of the algorithm is polynomial in $|\mathcal{A}_{\trianglerighteq}|$ in data complexity. □

2.3.3 Experimental Evaluation

This part is reserved for the experimental evaluation of the Cπ-repair method, using Algorithms 5 and 7. We mainly study the impact of data size (ABox size), data quality (number of conflicts), and different configurations of partial orders. Our implementation aligns with some of the state-of-the-art systems of the ontology-based data access paradigm, with some differences due to the focus on partially preordered ABoxes. Let us start first by recalling some of the existing inconsistency-tolerant semantics systems before describing our experimental setting and results.

Overview of inconsistency-tolerant semantics systems

Several systems exist for querying description logic knowledge bases under inconsistency-tolerant semantics:

- The QuID system implements conjunctive query answering under the IAR semantics for DL-Lite$_{\mathcal{A}}$ knowledge bases [Rosati *et al.* 2012, Lembo *et al.* 2015]. It provides methods for query rewriting, ABox annotation, and ABox cleaning. While the first one is naturally feasible for read-only ABoxes, the second and third require modifying or creating a copy of the ABox, which are not often easy to perform.
- The SaQAI system implements an exact approach for conjunctive query answering under IAR and ICAR semantics over DL-Lite knowledge bases and an approximative approach for ICAR semantics over more expressive description logics (such as $\mathcal{EL}_{\bot nr}$) [Tsalapati *et al.* 2016]. The system applies a saturation approach after removing conflicting assertions from the ABox, this allows standard query answering from approximations of the closure of the initial ABox.
- The system of [Du *et al.* 2013] handles ground query answering under a variant of AR semantics over weighted ABoxes of the $\mathcal{SHIQ}$ description logic family. The system targets highly expressive DLs to handle conjunctive queries that reduce to ABox assertion entailment.
- The CQAPri system implements conjunctive query answering under AR and IAR semantics over DL-Lite knowledge bases, in addition to the preferred repair semantics based on prioritised set inclusion, also using AR and IAR semantics [Bienvenu *et al.* 2014, Bienvenu *et al.* 2019].
- ORBITS [Bienvenu & Bourgaux 2022] is a system that implements SAT-based algorithms and encodings for inconsistency-tolerant query answering from prioritised knowledge bases; it supports (Pareto and completion) optimal repairs under general priority relations and provides experimental comparisons across repair semantics (AR, IAR, brave), multiple SAT encodings, and alternative procedures for the same semantics.

In addition to these systems, some systems handle inconsistent relational databases, such as LinCQA [Fan *et al.* 2023], CAvSAT [Dixit & Kolaitis 2021], ConQuer [Fuxman *et al.* 2005, Fuxman & Miller 2005] and Hippo [Chomicki *et al.* 2004b, Chomicki *et al.* 2004a] which focus on tractable subcases of AR query answering, and ConsEx [Caniupán & Bertossi 2010] and EQUIP [Kolaitis *et al.* 2013] which use answer set programming and binary integer programming as reductions of AR query answering. Note that none of the cited systems is directly

comparable to our system, since they employ different semantics and do not target partially preordered ABoxes.

Experimental setting

Ontology. In order to provide a TBox for our experiments, we used the modified LUBM benchmark $\text{LUBM}^{\exists}_{20}$ TBox [Lutz *et al.* 2013] (available at https://home.uni-leipzig.de/clu/). This ontology provides an extended DL-Lite$_{\mathcal{R}}$ version of the well-known $\mathcal{ELI}$ TBox of the Lehigh University Benchmark (LUBM) that describes the university domain [Guo *et al.* 2005]. The $\text{LUBM}^{\exists}_{20}$ TBox is built upon 127 concepts, 27 roles, and 202 positive inclusions. It also removes axioms that go beyond DL-Lite$_{\mathcal{R}}$ and adds positive concept inclusion axioms involving existential restriction on the right-hand side to increase the number of axioms. However, the ontology does not contain any negative inclusion axioms, which are the main mechanism for capturing inconsistency in DLs. Hence, to allow for contradictions, we added a set of negative inclusion axioms to the TBox. Our addition resulted in 4561 negative inclusion axioms overall (size of the TBox's negative closure $|cln(\mathcal{T})|$). This number of constraints allows us to easily control the size of conflict sets used in our evaluation.

Datasets (ABoxes). We used the Extended University Data Generator (EUDG) [Lutz *et al.* 2013] to generate three ABoxes of different sizes and transformed each ABox into a SQLite database. Then, we introduced inconsistency by adding contradictory assertions in each ABox as follows. For each negative axiom inferred from the TBox, we contradict the presence of an individual in a concept assertion with probability p and in a role assertion with probability $p/2$. We used different values of p in order to create five different situations within each ABox in terms of inconsistency, which is measured by the number of conflicts. Note that we used the same process described and used for the CQAPri benchmark [Bienvenu *et al.* 2014, Bienvenu *et al.* 2019] to generate our inconsistent ABoxes. The generated ABoxes in addition to their size (**#A**), their conflict set size (**#Cf**) and the conflict set computation time (**Cf(s)**) are summarised in Table 2.2. The next step is the representation of partial preorders over the generated ABoxes.

Directed acyclic graphs for partial orders. In order to represent partially preordered ABoxes in practice, partially ordered priority degrees are assigned to the assertion. These degrees can be represented graphically using Hasse Diagrams or Directed Acyclic Graphs (DAGs). We used transitive DAGs where all arcs are explicitly represented within the graph [Gutin 2018]. The nodes of a transitive DAG represent degrees of a partially ordered set (POSet), each arc indicates the strict preference of a degree over another, and the absence of an arc expresses incompatibility of two degrees. We generated a set of random DAGs using the R language package `bnlearn` [Scutari 2009], varying in the number of nodes ($\{50, 100, 500, 1000, 2500\}$) and different probabilities for the presence of arcs $(0.1, \ldots, 0.9)$. Probabilities indicate the density of a DAG, where a less dense DAG has more occurrences of incomparability. For each execution of our algorithms, degrees from a previously generated DAG are randomly assigned to the assertions of each ABox.

Configuration. We implemented our system in Python, using the RDFLib library [Krech *et al.* 2023] to read the ontology and handle it in memory, a SQLite3 relational database management system (DBMS) [Hipp 2020] to store the ABox, and the Rapid query rewriting engine [Chortaras *et al.* 2011] to reformulate the queries used for conflicts and supports computation. Our system is publicly available at: https://github.com/ahmedlaouar/py_reasoner/

Table 2.2: ABoxes used in our experiments: size (**#A**), conflicts number (**#Cf**) and computation time (**Cf(s)**), π-repair size (**#π**) and computation time (**π(s)**), Cπ-repair size (**#$c\pi$**) and computation time (**$c\pi$(s)** and **$c\pi$++(s)**).

ID	#A	# Cf	Cf (s)	#π	π (s)	#$c\pi$	$c\pi$ (s)	$c\pi$++ (s)
u.5p5e-6	9156	111	1.08	8078	1.24	22740	27.65	11.26
u.5p1e-5	9158	387	1.42	5344	1.33	15859	30.56	18.15
u.5p5e-5	9193	1268	1.43	742	1.28	2473	30.15	30.86
u.5p5e-4	9228	3372	1.6	1102	1.41	3651	30.75	29.35
u.5p1e-3	9375	7733	1.7	442	1.57	1475	28.83	28.65
u1p5e-6	75671	753	2.51	36083	4.36	66867	203.59	53.45
u1p1e-5	75678	3049	2.77	22804	8.38	46257	261.47	74.74
u1p5e-5	75718	10388	4.14	5856	5.07	14992	267.33	130.9
u1p15e-5	75842	28553	4.84	2378	6.54	6834	258.71	168.25
u1p5e-4	76255	83863	6.5	988	6.9	2994	251.23	170.61
u5p5e-6	463349	4505	22.48	74511	39.39	167473	6635.11	2160.48
u5p1e-5	463400	8327	23.6	30204	38.89	78857	6711.89	2941.02
u5p5e-5	463684	57054	26.3	8693	35.85	25921	5772.07	3105.82
u5p1e-4	464060	121621	29.76	6213	34.12	18794	6492.42	3566.24
u5p5e-4	466785	544500	49.95	1470	95.88	4713	5868.32	3956.46

tree/DL2024-version. The experiments were performed on an Intel 3.60GHz CPU with 32 GB RAM under Ubuntu, full instructions to reproduce the experiments are provided with the implementation. We refer to the π-repair computed using Algorithm 6, and the Cπ-repair computed using Algorithms 5 and 7 by π, $c\pi$ and $c\pi$++, respectively. We consider three different cases for the evaluation:

Case 1: We analyse both the performance of the algorithms as a function of the size of the ABox and the impact of the number of conflicts on the size of the repairs and the running time (for computing the repairs). The results are presented in Table 2.2 and Figure 2.11.

Case 2: We consider different configurations for the associated DAG (POSet). We analyse the impact of changing the size and density of the DAG. This allows us to measure the difference $c\pi(\mathcal{A}_{\trianglerighteq}) \setminus cl(\pi(\mathcal{A}_{\trianglerighteq}))$. We point out the main situations where the Cπ-repair is more productive than the π-repair. The results are depicted in Figure 2.12 [left].

Case 3: We use ABox **u.5** to compare the proportion of time spent in each step of Algorithms 5 and 7. For each ABox by number of conflicts, we computed the median time from different executions. The results are presented in Figure 2.12 [middle] and [right].

Experimental results

Impact of the number of conflicts (Case 1). We first compare the running times of Algorithms 5 and 7 ($c\pi$ and $c\pi$++) for computing the Cπ-repair. The results clearly show that the revised method significantly outperforms the naive one. This is particularly evident in the case of weakly conflicting ABoxes, which correspond to (111, 387) conflicts for **u.5**, (753, 3049) for **u1**, and (4505, 8327) for **u5**. For these ABoxes, the revised method reduces the running time to about 25% of that of the naive method (see Figure 2.11 [left], for the case of (753, 3049) conflicts). The same tendency holds even for large ABoxes such as **u5** in Table 2.2.

Furthermore, in highly conflicting ABoxes, the revised method requires at most half of the time taken by the naive one. This behaviour is explained by the fact that in such cases the π-repair is smaller, and more assertions must be verified through the Cπ-repair characterisation. As shown in Figure 2.11 [left], the computation of the π-repair remains stable, since it essentially involves a constant-time verification step for each ABox assertion.

Regarding the size of the obtained repairs, we observe that a substantial portion of assertions is discarded once the number of conflicts exceeds 10% of the ABox size (1268 for **u.5**, 10388 for **u1**, and 57054 for **u5**), as illustrated in Figure 2.11 [right]. This outcome is expected and also holds under IAR semantics (see [Rosati *et al.* 2012, Bienvenu *et al.* 2014] for more details).

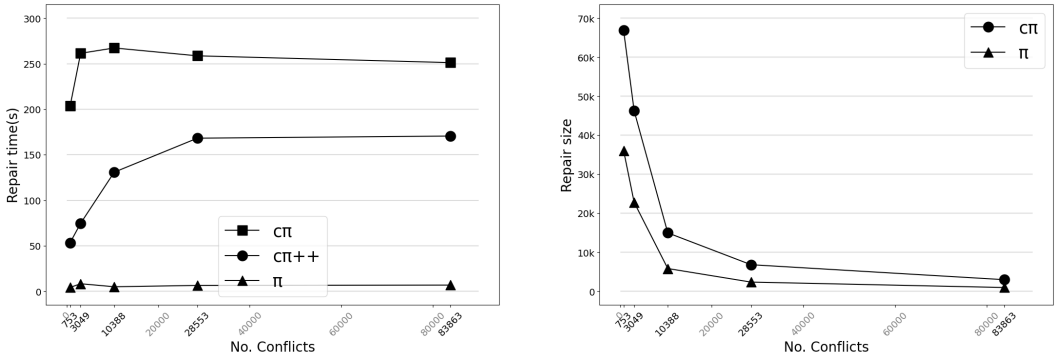


Figure 2.11: Evolution of repair time [left] and repair size [right] w.r.t. the number of conflicts (ABox **u1**).

Impact of the used partial order (Case 2). To highlight the situations where the Cπ-repair is more productive than the π-repair, we compare the sizes of the π-repair (π), its closure ($cl(\pi)$), and the additional assertions included only in the Cπ-repair (only $c\pi$), as a function of the density of the partial order (POSet). The results are shown in Figure 2.12 [left], using ABox **u1** with 753, 3049, and 10388 conflicts. The evaluation reveals that lower probabilities in the POSet—which correspond to a higher degree of incomparability—lead to a more productive Cπ-repair. Specifically, when the probability of a strict preference between two degrees is below 0.5, the Cπ-repair is larger than the π-repair. Conversely, for higher probabilities, the POSet approaches a total order, and the advantage of Cπ-repair over the π-repair diminishes. This confirms Lemma 2.4: with more incomparabilities, assertions are more likely to have different supports dominating different conflicts, which explains the additional assertions found only in the Cπ-repair. Moreover, we observe that for POSets with many incomparabilities, the Cπ-repair produces results even when the π-repair is empty (see Figure 2.12 [left], for densities 0.1-0.3). Repeated executions with varying POSet densities further show that in 80% of the cases where the π-repair contains fewer than 10% of the ABox assertions, the Cπ-repair still provides a substantial number of additional assertions. For weakly conflicting ABoxes, this added contribution appears in about 10% of the cases, while for highly conflicting ABoxes, the percentage drops to below 5%.

Proportion of time spent by Algorithms 5 and 7 (Case 3). Figure 2.12 ([middle] and [right]) reports the running time of the different steps in Algorithms 5 and 7. In both cases, the computation of the ABox closure requires constant time. The time needed to identify the set of conflicts grows slightly with the number of conflicts, which also contributes to an increase in

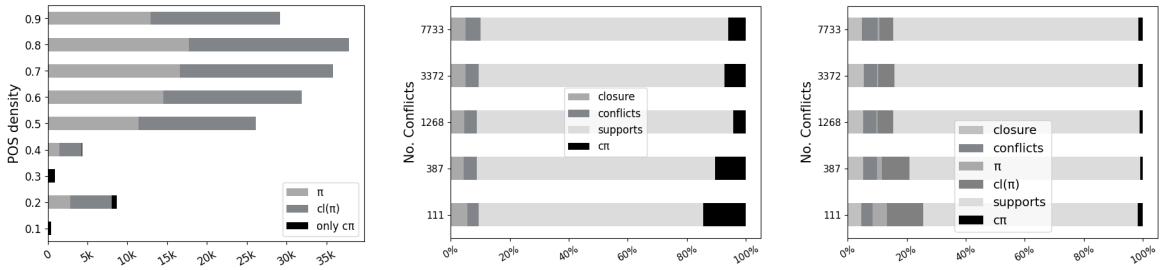


Figure 2.12: Proportion of assertions in each repair vs POSet density [left]. Proportion of repair time for ABox **u.5**, using Algorithm 5 [middle] and Algorithm 7 [right].

the overall running time. In the revised algorithm (Figure 2.12 [right]), the computation of the π-repair remains constant with respect to the number of conflicts, while computing its closure becomes comparatively more expensive in weakly conflicting ABoxes. The computation of supports requires iterating over every assertion in the ABox closure and executing multiple SQL queries to retrieve them. Although substantially faster in the revised algorithm (see Table 2.2), supports computation remains the most time-consuming step in the process.

2.4 Conclusion

Developing tractable and safe inconsistency-tolerant semantics is both crucial and challenging when there is a need for query answering from inconsistent knowledge bases. This chapter proposed a tractable and safe method for computing a closure-based possibilistic repair for partially preordered DL-Lite$_\mathcal{R}$ knowledge bases, called the Cπ-repair. It mainly shows that applying the deductive closure in the process of computing a repair for partially preordered knowledge bases yields a more productive repair compared to its unclosed version and does not suffer from any additional computational cost induced by the use of the closure. In particular, tractability is achieved thanks to an equivalent characterisation that is based on the notions of support and dominance. Intuitively, valid conclusions are those that follow from consistent subsets that dominate the conflicts of the ABox. Furthermore, an algorithmic study is conducted to take advantage of the properties of the proposed tractable characterisation and to enhance it. Finally, an experimental evaluation illustrates the impact of data size, data quality, and different configurations of partial orders on the computation of the Cπ-repair.

An important question that follows from this chapter is whether the deductive closure can be applied at a different level of the process of computing the Cπ-repair (illustrated by Figure 2.4. Indeed, the method introduced in this chapter follows the same style of the ICR semantics [Bienvenu 2012a] where the deductive closure is applied to each of the computed ABox repairs. The following chapter studies the alternative application of the deductive closure on the initial inconsistent partially preordered ABox, in the same style as the ICAR semantics [Lembo *et al.* 2010]. In addition, it provides a semantics characterisation of the inference from partially preordered and inconsistent knowledge bases.

Chapter 3

The Deductive Closure and the Possibilistic Repair: The Order of Application Does Not Matter

Contents

3.1 Closure-Based Possibilistic Inference from Partially Ordered Knowledge Bases 94
3.1.1 Quantitative Representation of Partially Ordered Possibilistic ABoxes . 95
3.1.2 How to Deductively Close a Partially Ordered ABox? 96
3.1.3 The cπ-acceptance Method 98
3.1.4 Equivalence with the Cπ-repair 99
3.2 Experimental Evaluation 101
3.2.1 Experimental Setting 101
3.2.2 Experimental Results 104
3.3 Semantics of Partially Ordered and Inconsistent ABoxes 105
3.3.1 The Preferred Interpretations of a Partially Ordered Knowledge Base . 106
3.3.2 Semantic Properties of the π-repair 108
3.4 Conclusion 110

In the previous chapter, we presented the process of repairing partially preordered DL-Lite$_\mathcal{R}$ knowledge bases, which involves the extension of the partial preorder into a family of compatible total preorders, which could be exponential in the worst case. Then, a possibilistic repair step followed by a deductive closure step are applied to each obtained totally preordered ABox, before intersecting the resulting closures and obtaining a single repair for a partially preordered inconsistent ABox. The application of the closure is done in the spirit of the ICR semantics [Bienvenu 2012a], after the computation of the repairs. Another way of applying the deductive closure in the process of repairing partially preordered ABoxes is by deductively closing the initial ABox. In this case, the deductive closure is applied as in the ICAR semantics [Lembo *et al.* 2010], where the initial ABox is closed, before computing the repairs.

In DL-Lite$_\mathcal{R}$, the tractability of the Cπ-repair method is guaranteed through the "Dominant supports" characterisation, which relies on the evaluation of the dominance of at least one support against each conflict of the ABox when checking the entailment of an assertion. However, going through sets of conflicts and supports can be impractical when dealing with very large

ABoxes. This observation was also made through the experimental evaluation of the characterisation in Chapter 2, Section 2.3, and constitutes the motivation to further explore the Cπ-repair process by prior closure of partially preordered ABoxes.

The main challenge in this case is the application of the deductive closure on a partially preordered ABox. Indeed, as given by Definition 2.5, the closure operator is performed independently of any ordering defined on the knowledge base and produces a set of assertions without their related ordering. Moreover, the use of partial preorders in this case further complicates the task. For instance, standard possibilistic logic assigns the maximum reliability level to an inferred assertion when it is entailed in multiple ways, resulting in a single reliability level. However, it is unclear how to assign a single level of reliability when assertions may be incomparable. In this chapter, we develop our contributions along three main directions:

1. **Efficient Repair Process using the Partially Ordered Deductive Closure.** We introduce a novel deductive closure operator for partially ordered ABoxes. This step produces an enriched ABox on which possibilistic inference is applied, yielding an equivalent characterisation of the Cπ-repair method. The new approach avoids explicit enumeration of assertion supports when computing repairs and reduces entailment to the most basic reasoning task: consistency checking, over subsets of the ABox.

2. **Experimental evaluation.** We provide experiments that demonstrate the practical relevance of computing the deductive closure before generating repairs, and show the advantages of the new characterisation compared to the previous "Dominant supports" characterisation.

3. **Semantic characterisation of inference.** Finally, we characterise the inference with inconsistent and partially preordered DL-Lite$_\mathcal{R}$ knowledge bases. This relies on the notion of preferred interpretations which are compared on the basis of the assertions they falsify.

The main results of this chapter have been published in [Laouar & Benferhat 2026].

3.1 Closure-Based Possibilistic Inference from Partially Ordered Knowledge Bases

This section addresses an alternative way of computing a closed repair for partially preordered inconsistent ABoxes. The idea is to apply the positive deductive closure to the initial ABox before computing the repairs, in the same style as the ICAR semantics [Lembo *et al.* 2010]. One major difference between the ICAR semantics and what this section attempts to achieve lies in the ABox being partially preordered. Thus, one must account for the order of assertions when closing the ABox. Such an application can be straightforward when dealing with comparable assertions, where a maximum can be computed. However, due to the representation of incomparability in partial preorders, a proper adaptation of the deductive closure operator must be achieved. The first part of this section addresses this particular issue by defining the partially preordered deductive closure, before showing how inference from it is done tractably.

Problem Statement

Can we tractably compute a closure-based possibilistic repair by deductively closing the initial inconsistent and partially preordered ABox? How does such a repair relate to the Cπ-repair introduced in Chapter 2?

In order to allow the deductive closure application in a more concise way, we adopt a quantitative representation of partially preordered possibilistic ABoxes, instead of the qualitative representation used in Chapter 2. In simple terms, the qualitative representation that uses a partial preorder $\trianglerighteq$ over the assertions of an ABox extends the well-ordered partition representation in the case of totally preordered ABoxes. Alternatively, weighted knowledge bases, as given by Definition 1.21 can be extended to cover partially preordered ABoxes. Both representations are equivalent, since only the plausibility ordering between assertions matters.

3.1.1 Quantitative Representation of Partially Ordered Possibilistic ABoxes

The quantitative representation of a partial preorder $\trianglerighteq$ defined on an ABox is achieved using symbolic weights assigned to ABox assertions from a partially ordered set denoted by:

$$\mathsf{U} = \{\mathbb{0}, u_1, \ldots, u_n, \mathbb{1}\}$$

The set U is equipped with a strict partial order $\rhd$, and together they form the uncertainty scale $\mathbb{L} = (\mathsf{U}, \rhd)$. By definition, $\rhd$ is an irreflexive and transitive relation. Some weights may be incomparable; that is, for any two weights $u_i, u_j \in \mathsf{U}$, $u_i \bowtie u_j$ means that neither $u_i \rhd u_j$ nor $u_j \rhd u_i$ holds. The weight $\mathbb{1}$, representing full certainty, is the greatest element of U in the sense that for every $u_i \in \mathsf{U}$, we have $\mathbb{1} \rhd u_i$. The assertions assigned the weight $\mathbb{0}$ are ignored and are not explicitly represented in the ABox.

Note that equivalence is represented by assigning the same weight to equivalently reliable assertions. Therefore, having a partial order $\rhd$ defined over the symbolic weights in U, the ABox remains partially preordered and keeps the notation $\mathcal{A}_{\trianglerighteq}$. A partially preordered DL-Lite$_{\mathcal{R}}$ KB is a triple $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq}, \mathbb{L} \rangle$, where a single weight from U is assigned to each assertion:

$$\mathcal{A}_{\trianglerighteq} = \{(\varphi_i, u_i) \mid \varphi \text{ is an assertion and } u_i \in \mathsf{U}\}$$

For simplicity, given two assertions $(\varphi_i, u_i), (\varphi_j, u_j) \in \mathcal{A}_{\trianglerighteq}$, we write $\varphi_i \rhd \varphi_j$ and $\varphi_i \bowtie \varphi_j$ to denote $u_i \rhd u_j$ and $u_i \bowtie u_j$, respectively.

Let us now adapt our running example to use partially ordered symbolic weights.

Example 3.1. *We adapt the knowledge base of Example 2.1 to use symbolic weights from a partially ordered set. We first recall the TBox $\mathcal{T}$:*

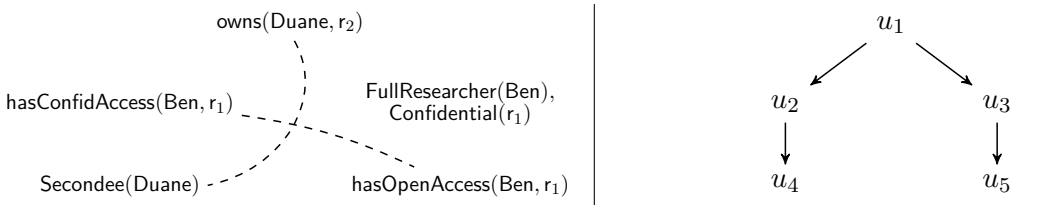
$$\mathcal{T} = \left\{ \begin{array}{ll} 1.\ \mathsf{Secondee} \sqsubseteq \mathsf{User} & 7.\ \exists \mathsf{hasAccess}^- \sqsubseteq \mathsf{Resource} \\ 2.\ \mathsf{FullResearcher} \sqsubseteq \mathsf{User} & 8.\ \exists \mathsf{hasConfidAccess}^- \sqsubseteq \mathsf{Confidential} \\ 3.\ \mathsf{Confidential} \sqsubseteq \mathsf{Resource} & 9.\ \mathsf{hasConfidAccess} \sqsubseteq \neg\, \mathsf{hasOpenAccess} \\ 4.\ \mathsf{Secondee} \sqsubseteq \neg\, \mathsf{FullResearcher} & 10.\ \mathsf{owns} \sqsubseteq \mathsf{hasAccess} \\ 5.\ \exists \mathsf{hasAccess} \sqsubseteq \mathsf{User} & 11.\ \mathsf{hasOpenAccess} \sqsubseteq \mathsf{hasAccess} \\ 6.\ \exists \mathsf{owns} \sqsubseteq \mathsf{FullResearcher} & 12.\ \mathsf{hasConfidAccess} \sqsubseteq \mathsf{hasAccess} \end{array} \right\}$$

The knowledge base is given by $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq}, \mathbb{L} \rangle$ *where the ABox assertions are given by:*

$\mathcal{A}_{\trianglerighteq} =$

Assertion	Weight
$\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$	u_1
$\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)$	u_2
$\mathsf{FullResearcher}(\mathsf{Ben})$	u_3
$\mathsf{Confidential}(\mathsf{r}_1)$	u_3
$\mathsf{Secondee}(\mathsf{Duane})$	u_4
$\mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)$	u_5

The uncertainty scale $\mathbb{L} = (\mathsf{U}, \rhd)$ *is given by the set of weights* $\mathsf{U} = \{u_1, u_2, u_3, u_4, u_5, \mathbb{1}\}$ *and the partial order* $\rhd$ *as illustrated by Figure 3.1b, where straight arrows denote strict preference* ($\longrightarrow$ *denotes* $\rhd$). *In addition, the conflict set of the ABox* $\mathcal{A}_{\trianglerighteq}$ *is recalled in Figure 3.1a, where dashed lines* ($- - -$) *represent a conflict.*



(a) ABox $\mathcal{A}$ Conflicts (b) The partial preorder $\rhd$

Figure 3.1: The Partially Preordered ABox $\mathcal{A}_{\trianglerighteq}$

3.1.2 How to Deductively Close a Partially Ordered ABox?

In order to compute the deductive closure of a partially preordered ABox, one must account for the degrees assigned to each inferred assertion. The natural choice is to assign the maximum of weights when an assertion is inferred in multiple ways. However, this assignment is not straightforward when some degrees of reliability are incomparable. For instance, in Example 3.1 the assertion $\mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1)$ can be inferred in two different ways:

1) $\langle \mathcal{T}, \{\mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)\} \rangle \vDash \mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1)$

2) $\langle \mathcal{T}, \{\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)\} \rangle \vDash \mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1)$

However, $\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)$ is assigned the weight u_2 $\mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)$ is assigned the degree u_5 and $u_2 \bowtie u_5$, which makes it impossible to select a single weight for each inferred assertion in the deductive closure.

To preserve full available information on ordering, both weights, when incomparability holds, must be retained. Indicating independent support for the inferred assertion without preference. Thus, the deductive closure of a partially ordered ABox cannot assign a single weight to each assertion. Instead, it yields an extended ABox where inferred assertions may carry sets of weights, reflecting multiple, possibly incomparable justifications. Assigning subsets of the partially ordered set U to assertions defines a new uncertainty scale comparing them. This scale is given by $\mathbb{L}^{\mathsf{U}} = (2^{\mathsf{U}}, \rhd^{\max})$, where the order $\rhd^{\max}$ is defined as follows:

$$\forall \mathcal{X}, \mathcal{Y} \subseteq \mathsf{U}, \quad \mathcal{X} \rhd^{\max} \mathcal{Y} \text{ iff } \forall u_j \in \mathcal{Y}, \exists u_i \in \mathcal{X} \text{ such that } u_i \rhd u_j$$

Since $\rhd$ is a strict order relation, one can easily check that $\rhd^{\max}$ is also an irreflexive and transitive order relation between subsets of U. The uncertainty scale $\mathbb{L}^{\mathsf{U}}$ allows us to move from an order over the elements of an ABox $\mathcal{A}_{\trianglerighteq}$ to an order over $2^{\mathcal{A}_{\trianglerighteq}}$. The following definition introduces the *partially preordered deductive closure* of a partially preordered ABox $\mathcal{A}_{\trianglerighteq}$ using the new uncertainty scale.

Definition 3.1 (Partially preordered closure)**.** *Let $\mathcal{T}_p$ denote the set of positive inclusion axioms of $\mathcal{T}$. The partially preordered deductive closure of $\mathcal{A}_{\trianglerighteq}$ w.r.t $\mathcal{T}$ is defined as:*

$$cl_{\rhd}(\mathcal{A}_{\trianglerighteq}) = \{(\varphi, \mathsf{U}(\varphi)) \mid \varphi \text{ is an assertion and } \mathsf{U}(\varphi) \neq \emptyset\}$$

where, for any assertion φ, $\mathsf{U}(\varphi)$ is the set of weights associated with its derivations, defined as:

$$\mathsf{U}(\varphi) = \{u_i \in \mathsf{U} \mid \exists(\varphi_i, u_i) \in \mathcal{A}_{\trianglerighteq} \text{ such that } \langle \mathcal{T}_p, \{\varphi_i\}\rangle \models \varphi\}$$

If there is no support for φ in $\mathcal{A}_{\trianglerighteq}$, then $\mathsf{U}(\varphi) = \emptyset$.

Note that, to align with the standard closure of Definition 2.5, each assertion φ is either a concept assertion $B(a)$ or a role assertion $R(a, b)$, where B and R are a concept and a role name in $\mathcal{T}$, and a, b are individuals mentioned in $\mathcal{A}_{\trianglerighteq}$. Furthermore, since DL-Lite$_{\mathcal{R}}$ does not allow for conjunction in TBox axioms, the minimal set to derive any assertion contains one assertion at most. Note also that if two assertions φ_i and φ_j are such that $\langle \mathcal{T}_p, \{\varphi_i\}\rangle \models \varphi_j$, then all weights that support φ_i also support φ_j; that is, $\mathsf{U}(\varphi_i) \subseteq \mathsf{U}(\varphi_j)$.

Example 3.2. *Let us apply the partially preordered closure on $\mathcal{A}_{\rhd}$ from Example 2.1. The TBox's positive inclusion axioms $\mathcal{T}_p$ are:*

$$\mathcal{T}_p = \left\{ \begin{array}{ll} 1.\ \mathsf{Secondee} \sqsubseteq \mathsf{User} & 7.\ \exists \mathsf{hasAccess}^- \sqsubseteq \mathsf{Resource} \\ 2.\ \mathsf{FullResearcher} \sqsubseteq \mathsf{User} & 8.\ \exists \mathsf{hasConfidAccess}^- \sqsubseteq \mathsf{Confidential} \\ 3.\ \mathsf{Confidential} \sqsubseteq \mathsf{Resource} & 10.\ \mathsf{owns} \sqsubseteq \mathsf{hasAccess} \\ 5.\ \exists \mathsf{hasAccess} \sqsubseteq \mathsf{User} & 11.\ \mathsf{hasOpenAccess} \sqsubseteq \mathsf{hasAccess} \\ 6.\ \exists \mathsf{owns} \sqsubseteq \mathsf{FullResearcher} & 12.\ \mathsf{hasConfidAccess} \sqsubseteq \mathsf{hasAccess} \end{array} \right\}$$

The partially preordered closure of $\mathcal{A}_{\rhd}$ is given by:

$cl_{\rhd}(\mathcal{A}_{\trianglerighteq}) =$

Assertion	Weights
$\varphi_1 = \mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$	$\{u_1\}$
$\varphi_2 = \mathsf{FullResearcher}(\mathsf{Duane})$	$\{u_1\}$
$\varphi_3 = \mathsf{User}(\mathsf{Duane})$	$\{u_1, u_4\}$
$\varphi_4 = \mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2)$	$\{u_1\}$
$\varphi_5 = \mathsf{Resource}(\mathsf{r}_2)$	$\{u_1\}$
$\varphi_6 = \mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)$	$\{u_2\}$
$\varphi_7 = \mathsf{Confidential}(\mathsf{r}_1)$	$\{u_2, u_3\}$
$\varphi_8 = \mathsf{User}(\mathsf{Ben})$	$\{u_2, u_3, u_5\}$
$\varphi_9 = \mathsf{Resource}(\mathsf{r}_1)$	$\{u_2, u_3, u_5\}$
$\varphi_{10} = \mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1)$	$\{u_2, u_5\}$
$\varphi_{11} = \mathsf{FullResearcher}(\mathsf{Ben})$	$\{u_3\}$
$\varphi_{12} = \mathsf{Secondee}(\mathsf{Duane})$	$\{u_4\}$
$\varphi_{13} = \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)$	$\{u_5\}$

3.1.3 The cπ-acceptance Method

To fully leverage the structure of partially ordered knowledge bases, we propose a new characterisation of Cπ-repair that starts with the partially preordered deductive closure defined in Definition 3.1. The key idea is the following: an assertion should only be accepted if it is consistent with all the information that is at least as reliable as itself, meaning assertions that are strictly more certain, equally certain, or incomparable. In this framework, each inferred assertion φ is annotated with the set of weights $\mathsf{U}(\varphi)$ it receives through the closure $cl_{\rhd}(\mathcal{A}_{\geq})$.

Instead of verifying consistency with the entire ABox, we narrow the scope to a relevant subset: those assertions whose weights are not strictly less reliable than any weight in $\mathsf{U}(\varphi)$. In other words, we test whether φ conflicts with what is already known and sufficiently trusted. If such a conflict arises, the assertion must be discarded from the Cπ-repair. To make this selection possible, we refine the $\rhd^{\max}$ relation to compare a set of weights $\mathsf{U}(\varphi_i) \subseteq \mathsf{U}$ with a singleton $\{u_j\}$, where $u_j \in \mathsf{U}$. The comparison is defined as follows:

$$\mathsf{U}(\varphi_i) \rhd^{\max} \{u_j\} \iff \exists u_i \in \mathsf{U}(\varphi_i) \text{ such that } u_i \rhd u_j$$

This helps us identify assertions that are strictly less certain than a given assertion $(\varphi_i, \mathsf{U}(\varphi_i)) \in cl_{\rhd}(\mathcal{A}_{\geq})$ and exclude them from the consistency check.

Definition 3.2. *Let $(\varphi_i, \mathsf{U}(\varphi_i))$ be an assertion from $cl_{\rhd}(\mathcal{A}_{\geq})$. The set of assertions that are strictly less certain than φ_i is defined as:*

$$\mathsf{SLT}(\varphi_i) = \{\varphi_j \mid (\varphi_j, u_j) \in \mathcal{A}_{\geq} \text{ and } \mathsf{U}(\varphi_i) \rhd^{\max} \{u_j\}\}$$

The complementary set containing assertions not strictly less certain than φ_i is:

$$\Delta(\varphi_i) = \mathcal{A}_{\geq} \setminus \mathsf{SLT}(\varphi_i)$$

We now introduce the crucial notion of *cπ-acceptance*, which plays a key role in characterizing whether an inferred assertion belongs to $c\pi(\mathcal{A}_{\geq})$, the closure-based partially preordered possibilistic repair given by Definition 2.10.

Definition 3.3. *An assertion $(\varphi_i, \mathsf{U}(\varphi_i)) \in cl_{\rhd}(\mathcal{A}_{\geq})$ is said to be cπ-accepted if and only if the pair $\langle \mathcal{T}, \Delta(\varphi_i) \rangle$ is consistent.*

This definition is fundamental because it reduces the problem of checking whether an inferred assertion belongs to the intersection of the closures of all repairs to a simple consistency check. Hence, the Cπ-repair of a partially preordered ABox $\mathcal{A}_{\geq}$ can be computed by applying Definition 3.3 to all assertions in $cl_{\rhd}(\mathcal{A}_{\geq})$ and collecting those that are cπ-accepted.

Example 3.3. *Let us now apply the cπ-acceptance characterisation of Definition 3.3 on the partially preordered closed ABox of Example 3.2. For each assertion φ_i, we use its set of weights $\mathsf{U}(\varphi_i)$ to exhibit $\Delta(\varphi_i)$, on which a consistency test is performed. The test*

result is indicated using (✓) if successful and (✗) otherwise.

$\varphi_1 = \mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$	$\mathsf{U}(\varphi_1) = \{u_1\}$	$\Delta(\varphi_1) = \{\varphi_1\}$	✓
$\varphi_2 = \mathsf{FullResearcher}(\mathsf{Duane})$	$\mathsf{U}(\varphi_2) = \{u_1\}$	$\Delta(\varphi_2) = \{\varphi_1\}$	✓
$\varphi_3 = \mathsf{User}(\mathsf{Duane})$	$\mathsf{U}(\varphi_3) = \{u_1, u_4\}$	$\Delta(\varphi_3) = \{\varphi_1\}$	✓
$\varphi_4 = \mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2)$	$\mathsf{U}(\varphi_4) = \{u_1\}$	$\Delta(\varphi_4) = \{\varphi_1\}$	✓
$\varphi_5 = \mathsf{Resource}(\mathsf{r}_2)$	$\mathsf{U}(\varphi_5) = \{u_1\}$	$\Delta(\varphi_5) = \{\varphi_1\}$	✓
$\varphi_6 = \mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)$	$\mathsf{U}(\varphi_6) = \{u_2\}$	$\Delta(\varphi_6) = \{\varphi_1, \varphi_6, \varphi_{11}, \varphi_7, \varphi_{13}\}$	✗
$\varphi_7 = \mathsf{Confidential}(\mathsf{r}_1)$	$\mathsf{U}(\varphi_7) = \{u_2, u_3\}$	$\Delta(\varphi_7) = \{\varphi_1, \varphi_6, \varphi_7, \varphi_{11}\}$	✓
$\varphi_8 = \mathsf{User}(\mathsf{Ben})$	$\mathsf{U}(\varphi_8) = \{u_2, u_3, u_5\}$	$\Delta(\varphi_8) = \{\varphi_1, \varphi_6, \varphi_7, \varphi_{11}\}$	✓
$\varphi_9 = \mathsf{Resource}(\mathsf{r}_1)$	$\mathsf{U}(\varphi_9) = \{u_2, u_3, u_5\}$	$\Delta(\varphi_9) = \{\varphi_1, \varphi_6, \varphi_7, \varphi_{11}\}$	✓
$\varphi_{10} = \mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1)$	$\mathsf{U}(\varphi_{10}) = \{u_2, u_5\}$	$\Delta(\varphi_{10}) = \{\varphi_1, \varphi_6, \varphi_7, \varphi_{11}, \varphi_{13}\}$	✗
$\varphi_{11} = \mathsf{FullResearcher}(\mathsf{Ben})$	$\mathsf{U}(\varphi_{11}) = \{u_3\}$	$\Delta(\varphi_{11}) = \{\varphi_1, \varphi_6, \varphi_7, \varphi_{11}, \varphi_{12}\}$	✗
$\varphi_{12} = \mathsf{Secondee}(\mathsf{Duane})$	$\mathsf{U}(\varphi_{12}) = \{u_4\}$	$\Delta(\varphi_{12}) = \{\varphi_1, \varphi_6, \varphi_7, \varphi_{11}, \varphi_{12}, \varphi_{13}\}$	✗
$\varphi_{13} = \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)$	$\mathsf{U}(\varphi_{13}) = \{u_5\}$	$\Delta(\varphi_{13}) = \{\varphi_1, \varphi_6, \varphi_7, \varphi_{11}, \varphi_{12}, \varphi_{13}\}$	✗

Successful consistency checks mean cπ-acceptance (✓). The assertion $\varphi_7 = \mathsf{Confidential}(\mathsf{r}_1)$ is cπ-accepted, since the set of weights associated with it is $\mathsf{U}(\varphi_7) = \{u_2, u_3\}$, and the set $\Delta(\varphi_7)$ involves assertions that have a weight that is more certain than, incomparable, or equivalent to both the weights u_2 and u_3. The set $\Delta(\varphi_7)$ in this case is consistent. Meanwhile, $\varphi_{10} = \mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1)$ is not cπ-accepted because the set $\Delta(\varphi_{10})$ is inconsistent due to the presence of the assertions $\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)$ and $\mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)$, which are conflicting.

One can already see that the set of the cπ-accepted assertions computed in Example 3.3 is equivalent to the Cπ-repair $c\pi(\mathcal{A}_{\trianglerighteq})$ computed in Example 2.10 (Chapter 2). The next section establishes the cπ-acceptance inference as an equivalent characterisation of the Cπ-repair.

3.1.4 Equivalence with the Cπ-repair

Applying the cπ-acceptance method on all assertions in the partially preordered deductive closure of a partially preordered ABox constitutes a repair process that is equivalent to the Cπ-repair method. This means that the deductive closure can be used equivalently before and after applying possibilistic inference when dealing with partially preordered ABoxes. This is not the case for flat knowledge bases, where applying the deductive closure before and after computing the repairs, in the ICAR and ICR semantics produces two distinct repairs, as we have seen in Chapter 1.

Therefore, the partially preordered deductive closure provides a further characterisation of the Cπ-repair that relies on a series of consistency checks of subsets of the ABox to compute the Cπ-repair. Before giving the proposition that establishes the new characterisation of the Cπ-repair, we start by an intermediary result on the relation between the supports of an assertion and its set of weights in the partially preordered deductive closure.

Lemma 3.1. *Let $(\varphi_i, \mathsf{U}(\varphi_i)) \in cl_{\trianglerighteq}(\mathcal{A}_{\trianglerighteq})$, then $\forall \mathcal{B}$ a support of φ_i in $\mathcal{A}_{\trianglerighteq}$, as per Definition 2.11, $\forall (\varphi_j, u_j) \in \mathcal{B}$, $u_j \in \mathsf{U}(\varphi_i)$.*

Proof. *Let $(\varphi_i, \mathsf{U}(\varphi_i))$ be an assertion in $cl_{\rhd}(\mathcal{A}_{\trianglerighteq})$. Let $\mathcal{B} \subseteq \mathcal{A}_{\trianglerighteq}$ be a support of φ_i in $\mathcal{A}_{\trianglerighteq}$. By Definition 2.11 of a support, a support is minimal for set inclusion. This means for DL-Lite$_{\mathcal{R}}$ knowledge bases that a support contains at most a single assertion, since conjunction is not allowed in the construction of TBox axioms. Therefore, $\forall \mathcal{B} \subseteq \mathcal{A}_{\trianglerighteq}$, a support of φ_i, $\mathcal{B} = \{(\varphi_j, u_j)\}$.*
Assume that there exists $\mathcal{B} = \{(\varphi_j, u_j)\}$ which is a support of φ_i. By definition of a support, $\langle \mathcal{T}, \{(\varphi_j, u_j)\}\rangle \models \varphi_i$.
By Definition 3.1 of the partially preordered closure, $(\varphi_i, \mathsf{U}(\varphi_i)) \in cl_{\rhd}(\mathcal{A}_{\trianglerighteq})$ means that $\exists (\varphi_j, u_j) \in \mathcal{A}_{\trianglerighteq}$ such that $\langle \mathcal{T}_p, \{(\varphi_j, u_j)\}\rangle \models \varphi_i$. This matches the condition of entailment in the definition of a support. Hence, $u_j \in \mathsf{U}(\varphi_i)$. □

Now, the following proposition establishes that each assertion from the partially preordered deductive closure that is $c\pi$-accepted as per Definition 3.3 is in the Cπ-repair.

Proposition 3.1. *Let $cl_{\rhd}(\mathcal{A}_{\trianglerighteq})$ be the partially preordered deductive closure of an ABox $\mathcal{A}_{\trianglerighteq}$. Then, an assertion $(\varphi_i, \mathsf{U}(\varphi_i)) \in cl_{\rhd}(\mathcal{A}_{\trianglerighteq})$ is $c\pi$-accepted, as per Definition 3.3, if and only if $\varphi_i \in c\pi(\mathcal{A}_{\trianglerighteq})$ (as per Definition 2.10).*

Proof. *Let φ_i be an assertion and $\mathsf{U}(\varphi_i)$ be the subset of weights associated with it in the partially preordered deductive closure $cl_{\rhd}(\mathcal{A}_{\trianglerighteq})$. Let the following set be the set of assertions which are strictly less certain than φ_i in $\mathcal{A}_{\trianglerighteq}$:*

$$\mathsf{SLT}(\varphi_i) = \{\varphi_j \mid (\varphi_j, u_j) \in \mathcal{A}_{\trianglerighteq} \text{ and } \exists u_i \in \mathsf{U}(\varphi_i) \text{ such that } u_i \rhd u_j\}$$

The set of the assertions of $\mathcal{A}_{\trianglerighteq}$ which are not strictly less certain than φ_i is:

$$\Delta(\varphi_i) = \mathcal{A}_{\trianglerighteq} \setminus \mathsf{SLT}(\varphi_i)$$

1. *Assume that $\varphi_i \in c\pi(\mathcal{A}_{\trianglerighteq})$ and let us show that the condition from our proposition holds, namely:*
   $$\langle \mathcal{T}, \Delta(\varphi_i)\rangle \text{ is consistent}$$
   Let $\mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$ be the conflict set of $\mathcal{A}_{\trianglerighteq}$. From the definition of the "Dominant supports" characterisation, given by Propositions 2.2 and 2.3, $\varphi_i \in c\pi(\mathcal{A}_{\trianglerighteq})$ means that $\forall \mathcal{C} \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$ there is $\mathcal{B} \subseteq \mathcal{A}_{\trianglerighteq}$ such that $\mathcal{B}$ is a support of φ_i and $\mathcal{B}$ dominates $\mathcal{C}$. Recall that dominance means that $\forall (\varphi_j, u_j) \in \mathcal{B}$, $\exists (\varphi_k, u_k) \in \mathcal{C}$ such that $u_j \rhd u_k$. As per Lemma 3.1, $(\varphi_j, u_j) \in \mathcal{B}$, where $\mathcal{B}$ is a support of φ_i in $\mathcal{A}_{\trianglerighteq}$ means that $u_j \in \mathsf{U}(\varphi_i)$. Since $u_j \rhd u_k$, where $(\varphi_k, u_k) \in \mathcal{C}$, this means that $\varphi_k \in \mathsf{SLT}(\varphi_i)$ (As per Definition 3.2). Hence, $\varphi_k \notin \Delta(\varphi_i)$. Since this applies to all conflicts in $\mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$, we have $\forall \mathcal{C} \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$, $\exists (\varphi_k, u_k) \in \mathcal{C}$, such that $\varphi_k \notin \Delta(\varphi_i)$, we conclude that $\langle \mathcal{T}, \Delta(\varphi_i)\rangle$ is consistent.
2. *Now assume that $\varphi_i \notin c\pi(\mathcal{A}_{\trianglerighteq})$ and let us show that the condition of our proposition is false, namely:*
   $$\langle \mathcal{T}, \Delta(\varphi_i)\rangle \text{ is inconsistent}$$
   $\varphi_i \notin c\pi(\mathcal{A}_{\trianglerighteq})$ means that there is at least one conflict $\mathcal{C}' \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$ such that $\forall (\varphi_k, u_k) \in \mathcal{C}'$, $\forall \mathcal{B}$ a support of φ_i in $\mathcal{A}_{\trianglerighteq}$, $\exists (\varphi_j, u_j) \in \mathcal{B}$, $u_j \not\rhd u_k$, $\forall (\varphi_k, u_k) \in \mathcal{C}'$. This means that either $u_k \rhd u_i$ or $u_k \bowtie u_i$, or $u_k = u_i (k = i)$. Hence, according to the definition of $\mathsf{SLT}(\varphi_i)$, $\forall (\varphi_k, u_k) \in \mathcal{C}'$, $\varphi_k \notin \mathsf{SLT}(\varphi_i)$. Therefore, $\forall (\varphi_k, u_k) \in \mathcal{C}'$, $\varphi_k \in \Delta(\varphi_i)$. Since $\mathcal{C}'$ is a conflict, this means that $\langle \mathcal{T}, \Delta(\varphi_i)\rangle$ is inconsistent.

□

The above proposition offers two major advantages. First, it avoids the need to enumerate all possible extensions of the partially preordered ABox. Second, it eliminates the need to compute the closure of each individual repair associated with those extensions. Instead, the closure is computed once, globally, over the initial structure, relying only on the weighted subsets associated with each inferred assertion. Figure 3.2 illustrates the process of computing the Cπ-repair alongside the equivalence with the new characterisation.

In DL-Lite$_\mathcal{R}$, consistency checking is performed in AC^0 with respect to data complexity [Calvanese *et al.* 2007]. Consistency testing is a crucial task in our approach, as each assertion must be checked for consistency against relevant subsets of the knowledge base. Each individual consistency test is in AC^0. The overall repair computation process, which involves iterating over all assertions and verifying their consistency, remains polynomial and tractable. The following proposition formalises the computation of the complete repair in polynomial time.

Proposition 3.2. *The computation of the set of assertions* $(\varphi, \mathsf{U}(\varphi))$ *from the partially preordered deductive closure of an ABox* $cl_{\rhd}(\mathcal{A}_{\trianglerighteq})$ *that are* $c\pi$*-accepted is achieved in polynomial time, in accordance with the data complexity of DL-Lite*$_\mathcal{R}$*.*

Proof. *The proof follows directly from the tractability of consistency checking for DL-Lite*$_\mathcal{R}$ *knowledge bases, which has been established by [*Calvanese et al. *2007,* Artale et al. *2009] (see Section 1.2.3).* □

3.2 Experimental Evaluation

We conducted experiments to evaluate the $c\pi$-acceptance characterisation of the Cπ-repair method. The experiments mainly focused on comparing the performance of the "Dominant supports" characterisation and the new characterisation in computing a repair for partially preordered DL-Lite$_\mathcal{R}$ ABoxes. Moreover, to allow for a fair comparison, we did not employ any enhancements in both methods. In particular, for the "Dominant supports" characterisation, we used Algorithm 5 which does not account for dominant sets of supports or conflicts. In the same way, we fully compute the partially preordered deductive closure in order to apply the $c\pi$-acceptance characterisation.

3.2.1 Experimental Setting

Ontology (TBox). We considered the latest ontology development version of the DBpedia project. The project uses mappings of Wikipedia infoboxes to populate the ontology. In order to preserve the same ontology provided by the project, we used interface methods for axioms retrieval and query rewriting that allow only for DL-Lite$_\mathcal{R}$ axioms in the TBox. The ontology provides negative inclusion axioms between classes and object properties, which aligns naturally with the negative inclusion axioms between concepts and roles in DL-Lite$_\mathcal{R}$. The TBox has 34653 axioms, where 27 of them are explicit negative inclusions, leading to a total of 614670 implicit and explicit negative inclusions.

Datasets (ABoxes) The ABoxes are obtained from the DBpedia ontology RDF type statements and object relations, in addition to the LHD inference (linked hypernyms) type statements from a previous version of the same ontology. The latter contains unclean inferred types, which makes it suitable for inconsistency experiments. Subsets of these datasets were extracted in

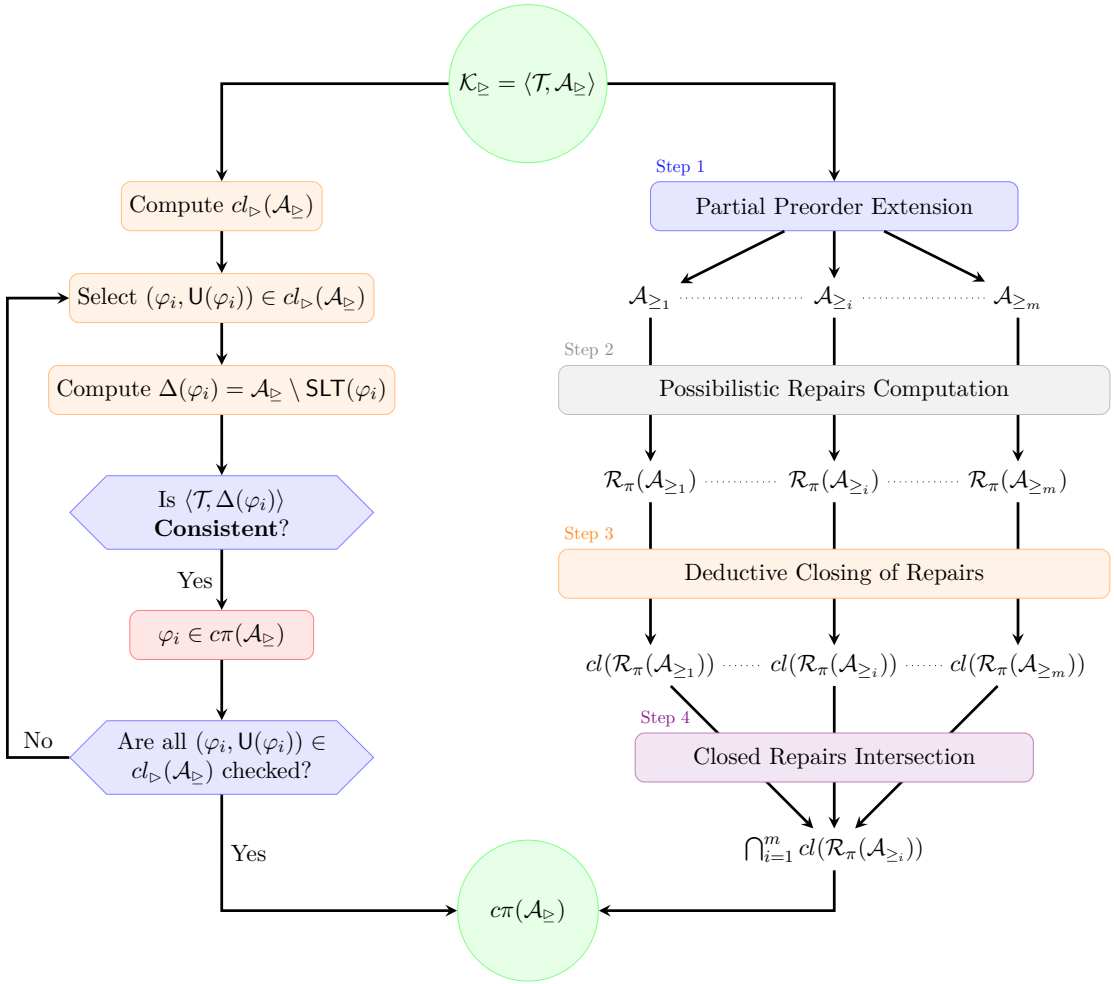


Figure 3.2: The $c\pi$-acceptance characterisation vs. the process of computing the Cπ-repair

growing size and number of conflicts to yield the ABoxes of our experiments. Details of the extracted ABoxes are given in Table 3.1, where along with the size of each ABox, we provide the percentage of distinct assertions involved in some conflict, the number of conflicts, the running time needed to extract conflict sets and the running times of computing the standard closure and the partially preordered deductive closure.

Table 3.1: ABoxes in terms of: size, percentage of assertions in conflicts, number of conflicts, and time to compute the conflict set, size of the deductive closure, time to compute the standard closure, and the partially ordered closure.

ABox ($\mathcal{A}$)	$\#\mathcal{A}$	%Cf($\mathcal{A}$)	#Cf($\mathcal{A}$)	Cf($\mathcal{A}$)(s)	$\#cl(\mathcal{A})$	$cl(\mathcal{A})$(s)	$cl_{\triangleright}(\mathcal{A})$(s)
dbr_n1e03_p5e-02	1k	8.3	72	59.29	3833	3.0	2.98
dbr_n1e03_p3e-01	1k	32.7	794	60.05	4716	2.87	3.05
dbr_n1e03_p5e-01	1k	52.4	1643	60.34	5248	3.0	2.94
dbr_n1e04_p5e-02	10k	7.03	1782	62.15	34968	2.73	3.58
dbr_n1e04_p3e-01	10k	32.36	28098	64.95	42691	3.01	3.34
dbr_n1e04_p5e-01	10k	51.56	51495	63.06	49535	3.31	3.3
dbr_n5e04_p5e-02	50k	10.68	25643	69.6	175376	2.88	4.38
dbr_n5e04_p3e-01	50k	33.02	230548	74.6	210992	3.38	6.76
dbr_n5e04_p5e-01	50k	51.79	397370	67.36	238008	3.63	7.07

Partial ordering. We defined a partial preorder over ABox assertions using three ordering criteria. First, we used two types of timestamps: derivationTimestamp from DBpedia and wikiTimestamp from Wikidata (via the Wikipedia API), both ordered by recency—more recent resources are more preferred. Second, we labelled each resource by its source: RDF statements from the latest DBpedia version, or LHD statements, which are considered less preferred. These criteria yield three orderings: $>_{\mathsf{dts}}$, $>_{\mathsf{wts}}$, and $>_{\mathsf{src}}$, based respectively on the derivation timestamp, wiki timestamp, and source dataset. These orderings are combined as follows:

$$\begin{aligned} \varphi \rhd \varphi' \text{ if } : \quad & \varphi >_{\mathsf{dts}} \varphi' \text{ or } \varphi >_{\mathsf{wts}} \varphi' \text{ or } \varphi >_{\mathsf{src}} \varphi' \\ & \wedge \\ & \neg(\varphi' >_{\mathsf{dts}} \varphi \text{ or } \varphi' >_{\mathsf{wts}} \varphi \text{ or } \varphi' >_{\mathsf{src}} \varphi) \end{aligned}$$

Configuration. We implemented our system in python using the py_reasoner tool, which was initially used in the experiments presented in Chapter 2, Section 2.3. However, we made significant changes to the way the system handles ABoxes. In particular, we changed the database management system on top of which our system is built to a PostgreSQL relational database. This was done mainly to allow for better management of the large-scale real-world data obtained from the DBpedia ontology. In addition, we preserved the use of the Rapid query rewriting engine and the RDFLib library, which is used to interface with OWL ontologies.

The generated ABoxes are stored in a PostgreSQL database, while the TBox is loaded as an RDF graph in memory. This allows us to retrieve TBox axioms efficiently in memory. The system relies on rewritings of conjunctive queries, computed using the Rapid tool to compute conflicts and supports of assertions. The updated version of our py_reasoner system is available at: https://github.com/ahmedlaouar/py_reasoner/tree/jelia2025. The experiments were performed on an Intel Core Ultra 7 5.0GHz CPU with 32 GB RAM under Ubuntu, full instructions to reproduce the experiments are provided with the implementation.

3.2.2 Experimental Results

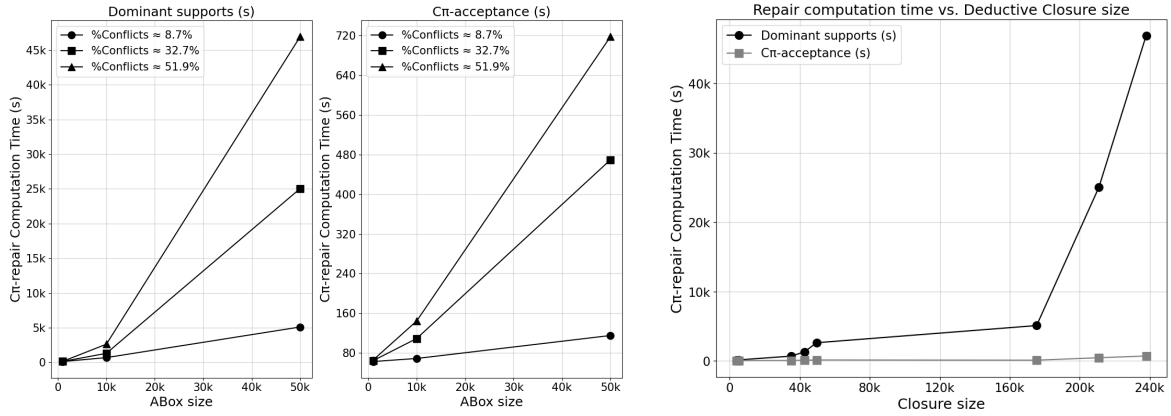


Figure 3.3: Comparison of Cπ-repair Computation Time using (2) Characterisations

We recall the implementation of the "Dominant supports" characterisation. As depicted by Algorithm 5, the conflict set of the ABox is computed first. Then, for each assertion in $cl(\mathcal{A}_{\trianglerighteq})$ (Definition 2.5), we compute its support sets, each linear in ABox size. Finally, we check that every conflict has at least one dominant support per assertion.

The new characterisation starts by computing the partially preordered closure, then checks consistency for each assertion with those not strictly less certain. This check can be done in two ways. The first uses a SQL query that detects violations of TBox negative inclusion axioms, ignoring those involving assertions less certain than the one checked. Since SQL involves I/O operations, the second runs in memory using the conflict set, identifying violations where no assertion is strictly less certain. We used the latter for weakly conflicting ABoxes, where the conflict set fits in memory.

Our experiments demonstrate the stable performance of the "$c\pi$-acceptance" method compared to the so-called "Dominant supports" method in computing the Cπ-repair. Figure 3.3 [left, middle] plots the running times across ABox sizes, grouped by percentage of conflicting assertions. The new method is at least 2× faster on small ABoxes (1k), 10× on medium (10k), and over 40× on large (50k). As both methods process each assertion in the deductive closure, we also plotted repair time against closure size in Figure 3.3 [right]. While the "Dominant supports" slows down with larger closures, the "$c\pi$-acceptance" method remains stable.

Discussion. The experimental evaluation highlights the efficiency and scalability of the proposed method. In particular, computing the Cπ-repair via the partially preordered closure avoids the exhaustive search over support sets required by the "dominant supports" method. While applying the deductive closure is inherently costly, especially under partial orders, our method mitigates this by limiting consistency checks to relevant assertion subsets. The results show that it scales gracefully with both the ABox size and the closure size, delivering substantial time savings even in large and highly conflicting datasets. This demonstrates the practical advantage of relying on a more targeted and order-aware closure.

3.3 Semantics of Partially Ordered and Inconsistent ABoxes

We start this chapter by introducing a semantics characterisation for inference in the inference of inconsistent and partially ordered ABoxes. To simplify the idea behind this part, we assume that for a standard knowledge base $\langle \mathcal{T}, \mathcal{A} \rangle$, an inconsistency-tolerant semantics s that produces a single repair $\mathcal{R}(\mathcal{A})$ is available (as in the π-repair or the Cπ-repair). Inference from $\langle \mathcal{T}, \mathcal{A} \rangle$ under s can be achieved by employing the TBox as follows $\langle \mathcal{T}, \mathcal{R}(\mathcal{A}) \rangle \models q$, where q is a query. Since the knowledge base is inconsistent, it does not admit any model. However, some of its interpretations satisfy the TBox $\mathcal{T}$ (as assumed by inconsistency-tolerant semantics). Among these interpretations, which are denoted $\Omega = \{\mathcal{I}_0, \ldots, \mathcal{I}_n\}$, there is one or more interpretations that satisfy $\mathcal{R}(\mathcal{A})$. The aim here is to find a method that identifies interpretations that satisfy the π-repair of a partially preordered ABox, without recurring to the actual computation of the repair. Figure 3.4 summarises the objective of this section.

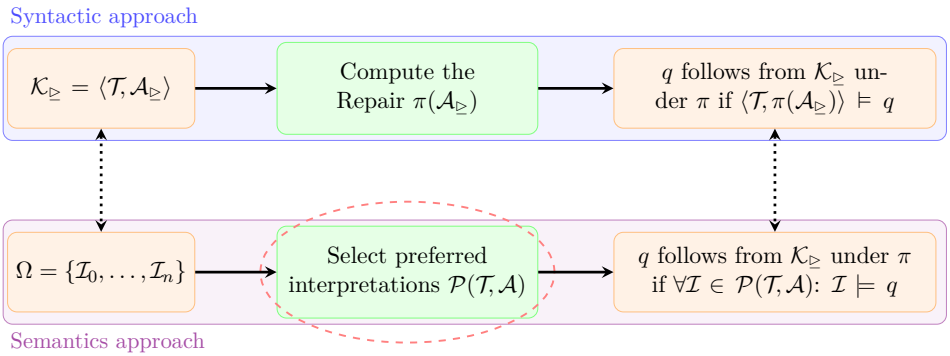


Figure 3.4: Syntactic vs. Semantics characterisation

In the standard setting, where the ABox is consistent and no priority is specified among assertions, DL-Lite$_{\mathcal{R}}$ semantics relies on the standard distinction between models and countermodels: an interpretation is a model if it satisfies both the TBox $\mathcal{T}$ and the ABox $\mathcal{A}$, and a countermodel otherwise. We can represent this distinction between models and countermodels using a binary scale consisting of two values, 1 and 0, where the value 1 is assigned to the models and the value 0 to the countermodels.

Whereas, when the ABox is partially preordered, the semantics can no longer rely on a simple binary model/countermodel classification. Instead, a degree of plausibility can be associated with each interpretation. To achieve a meaningful distinction between the interpretations of a knowledge base, we can rely on the partial preorder $\trianglerighteq$ that indicates the plausibility or preference of the assertions of an ABox. For the sake of simplicity, we adopt a quantitative representation of partially preordered possibilistic ABoxes, instead of the qualitative representation used in Chapter 2. In simple terms, the qualitative representation that uses a partial preorder $\trianglerighteq$ over the assertions of an ABox extends the well-ordered partition representation in the case of totally preordered ABoxes. Alternatively, weighted DL-Lite$_{\mathcal{R}}$ knowledge bases, as given by Definition 1.21 (Chapter 1) can be extended to cover partially preordered ABoxes. Both representations are equivalent, since only the plausibility ordering between assertions matters.

Problem Statement

Can we semantically characterise the π-repair of an ABox solely on the basis of the ABox's interpretations that satisfy the TBox?

3.3.1 The Preferred Interpretations of a Partially Ordered Knowledge Base

In order to represent the preference between interpretations, we use the uncertainty scale $\mathbb{L} = (\mathsf{U}, \rhd)$, where $\mathsf{U} = \{\mathbb{0}, u_1, \ldots, u_n, \mathbb{1}\}$, to associate a degree of plausibility to each interpretation. This approach aligns with the general framework of preferential semantics, widely used in non-monotonic reasoning. Among the interpretations, some will be considered fully acceptable (receiving the value $\mathbb{1}$), others only partially acceptable, and some entirely ruled out (receiving the value $\mathbb{0}$), typically those that violate strict TBox axioms or contradict assertions considered strictly certain.

Let Ω be the set of interpretations, and let $\mathsf{U} = \{\mathbb{0}, u_1, \ldots, u_n, \mathbb{1}\}$ be the set of partially ordered weights from the uncertainty scale $\mathbb{L}$, where $\mathbb{0}$ denotes complete impossibility, and $\mathbb{1}$ denotes full plausibility. We define a *partial possibility distribution* as a function $\sigma : \Omega \longrightarrow 2^{\mathsf{U}}$ that assigns to each interpretation $\mathcal{I} \in \Omega$ a set of plausibility levels, corresponding to the degrees of uncertainty of the formulas falsified by $\mathcal{I}$. Formally:

- If $\mathcal{I} \not\models \mathcal{T}$, or $\exists(\varphi_i, u_i) \in \mathcal{A}_{\geq}$ such that $\mathcal{I} \not\models \varphi_i$ and $u_i = \mathbb{1}$, then: $\sigma(\mathcal{I}) = \{\mathbb{0}\}$. This corresponds to interpretations that are entirely impossible.
- If $\mathcal{I} \models \mathcal{T}$ and $\mathcal{I} \models \varphi_i$ for all $\varphi_i \in \mathcal{A}$, then: $\sigma(\mathcal{I}) = \{\mathbb{1}\}$. This represents the ideal situation and the most preferred interpretations.
- Otherwise: $\sigma(\mathcal{I}) = \{u_i \in \mathsf{U} \mid \exists\, (\varphi_i, u_i) \in \mathcal{A}_{\geq}$ such that $\mathcal{I} \not\models \varphi_i\}$. This captures interpretations that are models of $\mathcal{T}$ and violate some assertions, but none with full certainty.

Since the partial possibility distribution σ assigns a set of degrees from U to each interpretation, a new uncertainty scale is obtained from $\mathbb{L}$ to allow comparison of interpretations. This scale is given by $\mathbb{L}^{\mathsf{U}} = (2^{\mathsf{U}}, \rhd^{\max})$, where the order $\rhd^{\max}$ is defined as follows:

$$\forall \mathcal{X}, \mathcal{Y} \subseteq \mathsf{U}, \quad \mathcal{X} \rhd^{\max} \mathcal{Y} \text{ iff } \forall u_j \in \mathcal{Y}, \exists u_i \in \mathcal{X} \text{ such that } u_i \rhd u_j$$

Since $\rhd$ is a strict order relation, one can easily check that $\rhd^{\max}$ is also an irreflexive and transitive order relation between subsets of U.

As a result, this new order relation allows us to specify the notion of preferred interpretations with respect to the new uncertainty scale $\mathbb{L}^{\mathsf{U}}$. Intuitively, we compare the degrees of plausibility associated with the assertions falsified by interpretations. Note that we focus only on interpretations that are models of the TBox $\mathcal{T}$, since interpretations that do not satisfy it are entirely ruled out.

Definition 3.4 (Preferred interpretation)**.** *Let Ω be the set of interpretations that satisfy the TBox of a partially preordered DL-Lite$_{\mathcal{R}}$ knowledge base $\mathcal{K}_{\geq} = \langle \mathcal{T}, \mathcal{A}_{\geq}, \mathbb{L} \rangle$. Let $\mathcal{I}$ and $\mathcal{I}'$ be two interpretations in Ω. The interpretation $\mathcal{I}$ is said to be more preferred to $\mathcal{I}'$, denoted $\mathcal{I} \rhd^{\Omega} \mathcal{I}'$ if:*

$$\sigma(\mathcal{I}') \rhd^{\max} \sigma(\mathcal{I})$$

An interpretation $\mathcal{I} \in \Omega$ belongs to the set of preferred interpretations, denoted $\mathcal{P}(\mathcal{T}, \mathcal{A})$, if there is no other interpretation $\mathcal{I}' \in \Omega$ such that $\mathcal{I}' \rhd^{\Omega} \mathcal{I}$. The set of preferred interpretations is given by:

$$\mathcal{P}(\mathcal{T}, \mathcal{A}) = \left\{ \mathcal{I} \in \Omega \,\middle|\, \nexists \mathcal{I}' \in \Omega \text{ such that } \mathcal{I}' \rhd^{\Omega} \mathcal{I} \right\}$$

Table 3.2: Some interpretations of $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq}, \mathbb{L} \rangle$

Interpretation	Falsified assertions	$\sigma(\mathcal{I}_i)$	Preferred?
$\mathcal{I}_0$	owns(Duane,r_2), hasConfidAccess(Ben,r_1), FullResearcher(Ben), Confidential(r_1), Secondee(Duane), hasOpenAccess(Ben,r_1)	$\{u_1, u_2, u_3, u_4, u_5\}$	✗
$\mathcal{I}_1$	hasConfidAccess(Ben,r_1), Secondee(Duane), hasOpenAccess(Ben,r_1)	$\{u_2, u_4, u_5\}$	✓
$\mathcal{I}_2$	hasConfidAccess(Ben,r_1), FullResearcher(Ben), Confidential(r_1), Secondee(Duane)	$\{u_2, u_3, u_4\}$	✗
$\mathcal{I}_3$	Secondee(Duane), hasOpenAccess(Ben,r_1)	$\{u_4, u_5\}$	✓
$\mathcal{I}_4$	Secondee(Duane), FullResearcher(Ben), hasOpenAccess(Ben,r_1)	$\{u_3, u_4, u_5\}$	✓
$\mathcal{I}_5$	hasConfidAccess(Ben,r_1), Secondee(Duane)	$\{u_2, u_4\}$	✓
$\mathcal{I}_6$	owns(Duane,r_2), hasOpenAccess(Ben,r_1)	$\{u_1, u_5\}$	✗
$\mathcal{I}_7$	owns(Duane,r_2), Confidential(r_1), hasConfidAccess(Ben,r_1)	$\{u_1, u_2, u_3\}$	✗

Example 3.4. *Let us now provide the preferred interpretations of the knowledge base of our example. We focus on interpretations that are model of the TBox $\mathcal{T}$. Table 3.2 shows some of these interpretations using the assertions they falsify and their partial possibility distribution.*
In order to exhibit the set of preferred interpretations, we apply the preference between interpretations with respect to the order relation $\rhd^{\max}$. We can notice that any interpretation that falsifies the most certain assertion of our ABox, owns(Duane, r_2)*, is not a preferred interpretation. In addition, any interpretation that falsifies assertions that have the weights u_2 and u_3 at the same time is also not a preferred interpretation. In Table 3.2, the preferred interpretations are indicated using a check-mark (✓) and the non-preferred interpretations are indicated by a cross-mark (✗). In addition, Figure 3.5 illustrates the preference relation between the interpretations discussed in this example, notice that Preferred interpretations (in violet) have no incoming arrows.*

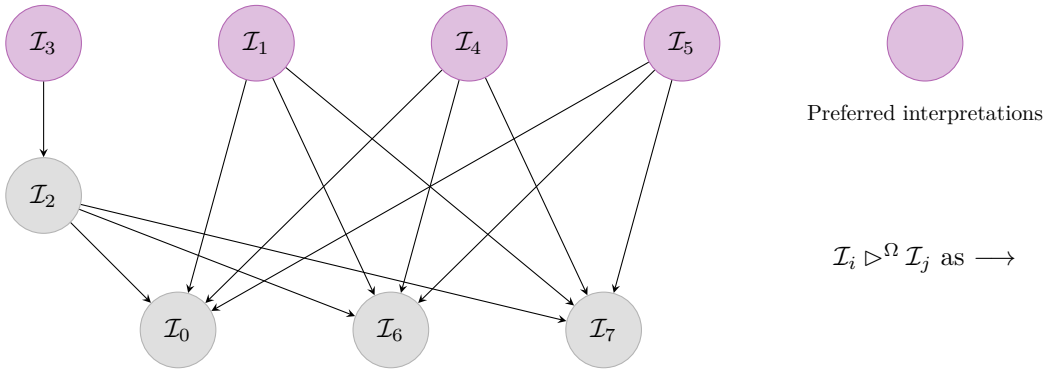


Figure 3.5: Preference relation $\rhd^{\Omega}$ among interpretations.

3.3.2 Semantic Properties of the π-repair

The notion of preferred interpretations serves as a starting point to provide a semantics characterisation for partially ordered possibilistic repairs.

The first result here connects the π-repair of an ABox $\mathcal{A}_{\trianglerighteq}$ (Definition 2.2), namely $\pi(\mathcal{A}_{\trianglerighteq})$ with the preferred interpretations of $\mathcal{A}_{\trianglerighteq}$, namely $\mathcal{P}(\mathcal{T},\mathcal{A})$. The following proposition establishes that if an interpretation $\mathcal{I}$ is a preferred interpretation, namely $\mathcal{I} \in \mathcal{P}(\mathcal{T},\mathcal{A})$ as given per Definition 3.4, then $\mathcal{I}$ is a model of $\pi(\mathcal{A}_{\trianglerighteq})$.

Proposition 3.3. *Let $\mathcal{I}$ be an interpretation in Ω. If $\mathcal{I}$ is a preferred interpretation, i.e., $\mathcal{I} \in \mathcal{P}(\mathcal{T},\mathcal{A})$, then $\mathcal{I}$ is a model of $\pi(\mathcal{A}_{\trianglerighteq})$, the π-repair of the ABox of a partially preordered knowledge base $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq}, \mathbb{L}\rangle$.*

Proof. *Let $\mathcal{I}$ be a preferred interpretation, i.e., $\mathcal{I} \in \mathcal{P}(\mathcal{T},\mathcal{A})$. Assume that $\mathcal{I}$ is not a model of $\pi(\mathcal{A}_{\trianglerighteq})$.*
$\mathcal{I}$ is not a model of $\pi(\mathcal{A}_{\trianglerighteq})$ means that there is at least one assertion $(\varphi_i, u_i) \in \mathcal{A}_{\trianglerighteq}$ that is π-accepted, i.e., $\varphi_i \in \pi(\mathcal{A}_{\trianglerighteq})$, such that $\mathcal{I} \not\models \varphi_i$.
Since $\varphi_i \in \pi(\mathcal{A}_{\trianglerighteq})$, by definition we have that $\forall \mathcal{C} \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$, $\exists (\varphi_j, u_j) \in \mathcal{C}$ such that $\varphi_i \rhd \varphi_j$ $(u_i \rhd u_j)$.
Now, let $\mathcal{B}$ be the set of assertions that are part of some conflict and are less preferred than φ_i, formally:

$$\mathcal{B} = \{(\varphi_j, u_i) \mid \exists (\varphi_j, u_j) \in \mathcal{C} \text{ such that } \varphi_i \rhd \varphi_j (\text{i.e., } u_i \rhd u_j)\}$$

Note first that $\mathcal{A}_{\trianglerighteq} \setminus \mathcal{B}$ is consistent, since at least one assertion from each conflict is included in $\mathcal{B}$. As a consequence, for each conflict $\mathcal{C} \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$, $\mathcal{C} \not\subset \mathcal{A}_{\trianglerighteq} \setminus \mathcal{B}$, thus no conflict is in $\mathcal{A}_{\trianglerighteq} \setminus \mathcal{B}$.
Now, let $\mathcal{I}'$ be a model of $\mathcal{A}_{\trianglerighteq} \setminus \mathcal{B}$. It is easy to see that $\mathcal{I}'$ is strictly preferred to $\mathcal{I}$: $\mathcal{I}' \rhd^{\Omega} \mathcal{I}$. In fact, since the assertions falsified by $\mathcal{I}'$ are exactly the assertions in $\mathcal{B}$, we have the following:

$$\forall (\varphi_j, u_j) \in \mathcal{A}_{\trianglerighteq} \text{ such that } \mathcal{I}' \not\models \varphi_j, \exists (\varphi_k, u_k) \in \mathcal{A}_{\trianglerighteq} \text{ such that } \mathcal{I} \not\models \varphi_k, \text{ with } \varphi_k \rhd \varphi_j (u_k \rhd u_j)$$

To satisfy the above condition, it is enough to consider the assertion φ_i each time, since φ_i is more preferred than all the assertions $\varphi_j \in \mathcal{B}$ which are falsified by $\mathcal{I}'$ (by construction of the set $\mathcal{B}$). Therefore, we are able to construct an interpretation $\mathcal{I}'$ that is more preferred than $\mathcal{I}$, thus $\mathcal{I} \notin \mathcal{P}(\mathcal{T},\mathcal{A})$, which contradicts our initial assumption. □

An important result of the above proposition is the fact that the assertions of the π-repair are included in the intersection of assertions that are satisfied by the preferred interpretations, as stated by the following lemma.

Lemma 3.2. *Let $\pi(\mathcal{A}_{\trianglerighteq})$ be the π-repair of the ABox of a partially preordered knowledge base $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq}, \mathbb{L}\rangle$ and $\mathcal{P}(\mathcal{T},\mathcal{A})$ be the set of preferred interpretations of the knowledge base $\mathcal{K}_{\trianglerighteq}$. Then $\pi(\mathcal{A}_{\trianglerighteq})$ is included in the intersection of the sets of assertions that are satisfied by the interpretations in $\mathcal{P}(\mathcal{T},\mathcal{A})$, formally:*

$$\pi(\mathcal{A}_{\trianglerighteq}) \subseteq \cap\{\varphi \mid \mathcal{I} \models \varphi \text{ such that } \mathcal{I} \in \mathcal{P}(\mathcal{T},\mathcal{A})\}$$

Proof. *The proof of this lemma follows directly from the proof of Proposition 3.3. Namely, since all preferred interpretations are models of $\pi(\mathcal{A}_{\trianglerighteq})$, then the intersection of the assertions these interpretations satisfy contains each assertion of $\pi(\mathcal{A}_{\trianglerighteq})$.* □

Note that preference may exist between interpretations that satisfy the π-repair of a partially preordered ABox, in particular we could have $\mathcal{I} \models \pi(\mathcal{A}_{\trianglerighteq})$, $\mathcal{I}' \models \pi(\mathcal{A}_{\trianglerighteq})$ and $\mathcal{I} \rhd^{\Omega} \mathcal{I}'$. In this case, $\mathcal{I}'$ cannot belong to $\mathcal{P}(\mathcal{T}, \mathcal{A})$. However, $\mathcal{I}'$ is still a model of $\pi(\mathcal{A}_{\trianglerighteq})$. Therefore, not all interpretations that are model of the π-repair are preferred interpretations, as given by Definition 3.4. We shall illustrate this observation in our next example.

Example 3.5. *Recall the π-repair of the ABox of our knowledge base $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq}, \mathbb{L} \rangle$:*

$$\pi(\mathcal{A}_{\trianglerighteq}) = \{\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)\}$$

The set of preferred interpretations is $\mathcal{P}(\mathcal{T}, \mathcal{A}) = \{\mathcal{I}_1, \mathcal{I}_3, \mathcal{I}_4, \mathcal{I}_5\}$. Notice that all preferred interpretations satisfy $\pi(\mathcal{A}_{\trianglerighteq})$, namely:

$$\forall \mathcal{I}_i \in \mathcal{P}(\mathcal{T}, \mathcal{A}), \mathcal{I}_i \models \pi(\mathcal{A}_{\trianglerighteq})$$

Notice also that $\mathcal{I}_2$ satisfies $\pi(\mathcal{A}_{\trianglerighteq})$, however it is not a preferred interpretation. This illustrates the observation we made above on the preference between interpretations that satisfy $\pi(\mathcal{A}_{\trianglerighteq})$.

Another property that follows from the definition of preferred interpretations with respect to the order relation $\rhd^{\Omega}$ is the fact that an interpretation that contradicts the π-repair $\pi(\mathcal{A}_{\trianglerighteq})$ is not preferred to any interpretation that satisfies $\pi(\mathcal{A}_{\trianglerighteq})$.

Lemma 3.3. *Let $\mathcal{I}$ and $\mathcal{I}'$ be two interpretations in Ω. If $\mathcal{I}$ is a model of $\pi(\mathcal{A}_{\trianglerighteq})$, the π-repair of the ABox of a partially preordered knowledge base $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq}, \mathbb{L} \rangle$, and $\mathcal{I}'$ is not a model of $\pi(\mathcal{A}_{\trianglerighteq})$, then $\mathcal{I}' \not\rhd^{\Omega} \mathcal{I}$.*

Proof. *Let $\mathcal{I}$ be an interpretation that satisfies the π-repair $\pi(\mathcal{A}_{\trianglerighteq})$, namely, $\forall \varphi \in \pi(\mathcal{A}_{\trianglerighteq})$, $\mathcal{I} \models \varphi$. Let $\mathcal{I}'$ be an interpretation that does not satisfy $\pi(\mathcal{A}_{\trianglerighteq})$, $\mathcal{I}' \not\models \pi(\mathcal{A}_{\trianglerighteq})$.*
Assume that $\mathcal{I}'$ is more preferred than $\mathcal{I}$, i.e., $\mathcal{I}' \rhd^{\Omega} \mathcal{I}$, which means that $\sigma(\mathcal{I}) \rhd^{\max} \sigma(\mathcal{I}')$.
Let (φ_i, u_i) be an assertion from the ABox $\mathcal{A}_{\trianglerighteq}$ that is π-accepted, i.e., $\varphi_i \in \pi(\mathcal{A}_{\trianglerighteq})$ and falsified by the interpretation $\mathcal{I}'$: $\mathcal{I}' \not\models \varphi_i$, hence we have that $u_i \in \sigma(\mathcal{I}')$.
Since $\sigma(\mathcal{I}) \rhd^{\max} \sigma(\mathcal{I}')$, this means that there is some other assertion $(\varphi_j, u_j) \in \mathcal{A}_{\trianglerighteq}$ that is not π-accepted, i.e., $\varphi_j \notin \pi(\mathcal{A}_{\trianglerighteq})$ and which is falsified by the interpretation $\mathcal{I}$: $\mathcal{I} \not\models \varphi_j$ such that $u_j \in \sigma(\mathcal{I})$ and $u_j \rhd u_i$ ($\varphi_j \rhd \varphi_i$).
Now, since $\varphi_j \notin \pi(\mathcal{A}_{\trianglerighteq})$, this means that there exists at least on conflict $\mathcal{C} \in \mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$ such that $\forall (\varphi_k, u_k) \in \mathcal{C}$, $\varphi_j \not\rhd \varphi_k$. This also means that $\forall (\varphi_k, u_k) \in \mathcal{C}$, $\varphi_i \not\rhd \varphi_k$ (since $\varphi_j \rhd \varphi_i$).
However, this contradicts our initial assumption that $\varphi_i \in \pi(\mathcal{A}_{\trianglerighteq})$. □

The previous lemma showed that an interpretation that falsifies $\pi(\mathcal{A}_{\trianglerighteq})$ cannot be more preferred than an interpretation that satisfies it. Nevertheless, it is possible to have two incomparable interpretations such that one satisfies the π-repair and the other falsifies it. This situation can be illustrated with a TBox that contains a single axiom, namely $\mathcal{T} = \{B \sqsubseteq \neg C\}$ and an ABox that contains three assertions $\mathcal{A}_{\trianglerighteq} = \{(A(a), u_1), (B(a), u_2), (C(a), u_3)\}$ with the

partial preorder $u_1 \rhd u_3$ and the remaining assertions incomparable. Clearly, the only assertion accepted in $\pi(\mathcal{A}_{\trianglerighteq})$ is $A(a)$ since it is more preferred to $C(a)$ which is involved in the single conflict of the ABox. Now, if we take two interpretations $\mathcal{I}_1$ and $\mathcal{I}_2$ such that $\mathcal{I}_1$ satisfies $A(a)$, the only assertion in the π-repair, and $\mathcal{I}_2$ satisfies $B(a)$ only, we obtain $\sigma(\mathcal{I}_1) = \{u_2, u_3\}$ and $\sigma(\mathcal{I}_2) = \{u_1, u_3\}$. $\mathcal{I}_1$ is not preferred to $\mathcal{I}_2$, i.e., $\sigma(\mathcal{I}_2) \not\trianglerighteq^{\max} \sigma(\mathcal{I}_1)$ because $u_1 \not\trianglerighteq u_2$ and $u_3 \not\trianglerighteq u_2$. The same applies for the converse relation, namely, $\mathcal{I}_2$ is not preferred to $\mathcal{I}_1$.

3.4 Conclusion

This chapter answers two major questions about the computation of a partially preordered possibilistic repair for DL-Lite$_{\mathcal{R}}$ knowledge bases. The first addressed question is whether we can obtain a closure-based partially preordered possibilistic repair if we start by applying the deductive closure on the initial ABox in the repair process, as in the ICAR semantics [Lembo *et al.* 2010]. By defining a notion of partially preordered deductive closure, we show that the order of applying the closure operator and computing the repair does not matter for possibilistic semantics, unlike the ICAR and ICR semantics, where closing the initial ABox or closing its repairs in the IAR semantics yields different results. The main result of this partially preordered deductive closure is a new repair process that is equivalent to the Cπ-repair defined in Chapter 2. In particular, tractability is preserved, since the problem is reduced to consistency checks over subsets of the closed ABox. This closure-based process yields the same repairs as those obtained by computing and then closing all compatible total preorder extensions of the ABox, but in a much more efficient manner. Finally, we show the efficiency of this method, in particular through comparisons with the earlier approach, which relies on exhaustive searches over the conflict set of the ABox and the support sets of assertions.

The second question concerns the possibility of semantically characterising the partially preordered possibilistic repair (π-repair), making a relation between the interpretations of an inconsistent ABox and its π-repair. Indeed, one can notice that inconsistency-tolerant semantics often focus on producing a syntactic approach relying on an alteration of an ABox that discards some of its assertions for query answering. Such approach is computationally interesting; however, it does not provide insight into the semantics of the considered repairs through the interpretations of a knowledge base. Our method's reliance on possibilistic semantics allows us to establish a connection between the ability of a possibility distribution to assign a degree of plausibility to worlds (or interpretations) and the possibilistic repair computed for partially preordered ABoxes. As a result, we achieved a semantic characterisation that allows us to check if an assertion is accepted in the π-repair of an ABox by checking if it is satisfied by the preferred interpretations of the ABox.

The next chapter addresses a different type of question, mainly that of analysing rationality properties in both standard inference from an inconsistent knowledge base and in conditional inference, which is performed in the presence of a fully certain observation. In addition, we discuss extensions of our method, the Cπ-repair, both at the representation level, by considering a richer language, and at the semantics level, by applying a cardinality-based preferred semantics instead of the possibilistic semantics to repair compatible total preorder extensions of the partially preordered ABox.

Chapter 4

Rationality Properties Analysis and Beyond the Cπ-repair

Contents

4.1 Rationality Properties 113
4.1.1 Unconditional Inference 113
4.1.2 Conditional Inference 114
4.1.3 Satisfaction of System P Properties 118
4.2 On the Extension of the Cπ-repair to DL-Lite$_{\mathcal{R},\sqcap}$ 125
4.2.1 The DL-Lite$_{\mathcal{R},\sqcap}$ language 125
4.2.2 The Cπ-repair for DL-Lite$_{\mathcal{R},\sqcap}$ knowledge bases 126
4.2.3 The result of applying the "Dominant supports" characterisation on a DL-Lite$_{\mathcal{R},\sqcap}$ knowledge base 128
4.3 A Lexicographic-Based Semantics for the Cπ-repair 129
4.4 Conclusion 134

In the previous chapters, we discussed and proposed some methods to deal with inconsistent information. In particular, we introduced an efficient method to reason with inconsistent partially ordered DL-Lite$_{\mathcal{R}}$ knowledge bases. In this chapter, we discuss some additional aspects of the methods that handle inconsistency. Among many, an important aspect is studying the behaviour of such methods. Indeed, ontologies are designed to capture constantly evolving and changing datasets [De Giacomo *et al.* 2021, Calvanese *et al.* 2010a]. Thus, we are interested in studying these methods with respect to two types of properties. The first type concerns fixed knowledge bases; how does our method handle the entailment of a conjunction (or set union) of conclusions? and what can we say about the consistency of its entailments? The answer of this last question helps us establish the fact that the developed method is safe, in the sense that it does not infer conclusions that follow solely from a conflicting fact.

Then, for the second types, one can observe that providing an inconsistent ABox with new factual assertions would change the outcome of our repair. This observation underscores the non-monotonicity of the repair and leads us to the study of the repair's behaviour with respect to a set of well-established conditional inference properties. Such a study is performed in the presence of a set of sure (or fully certain) new facts that we refer to as observation. This type of behaviour study, in the presence of inconsistency, has been the subject of the related framework of non-monotonic reasoning, in particular in the propositional logic setting [Kraus *et al.* 1990,

Konieczny *et al.* 2019, Giordano & Gliozzi 2021], but also in existential rules [Baget *et al.* 2016], description logics [Britz *et al.* 2021, Everett *et al.* 2021] and paraconsistent logics [Liu *et al.* 2023].

In addition, we go a step further with the setting in which our method is proposed by considering its application in a richer description logic. In particular, we aim for a language where conflicts are of an arbitrary size (i.e., they involve more than two assertions unlike in DL-Lite$_\mathcal{R}$). Indeed, the characterisation that we proposed in Chapter 2 relies on the computation of sets of conflicts and supports to decide whether an assertion belongs to the repair, and a DL-Lite$_\mathcal{R}$'s ABox admits conflicts of two assertions at max. Therefore, we made the choice of applying our method on the DL-Lite$_{\mathcal{R},\sqcap}$ language [Calvanese *et al.* 2006], which adds to the expressiveness of DL-Lite by considering conjunction in TBox axioms. Such an addition leads to potentially larger conflicts.

From a semantics point of view, we do not see any limitations to applying our method to such richer languages. In fact, the characterisation given by Propositions 2.2 and 2.3 is given independently of the size of a conflict and the size of a support. It remains to consider the computational complexity level, where one needs to ensure that not only the used sets (i.e., conflicts and supports) are computed in polynomial time but also that their size is polynomial in the size of the ABox, in order to ensure tractability.

Finally, in order to complete the studies around the proposed inconsistency handling method, we apply it using a different type of semantics. The core idea behind repairing partially ordered ABoxes is to extend the partial order into a family of compatible total orders, for which a standard preferred-based repair method is applied. Our method, the Cπ-repair applies the possibilistic repair, which yields a single repair for each compatible total order. However, we are interested in seeing the behaviour of the proposed approach if we replace the possibilistic semantics with a repair approach that yields an arbitrary number of repairs.
The chapter is structured around three main directions:

1. **Rationality properties analysis.** We examine to what extent the Cπ-repair and the π-repair methods satisfy well-established rationality properties, both for unconditional inference (based solely on the produced repair) and conditional inference (where a fully certain observation is added to the inconsistent ABox). The latter case is especially relevant as it provides insight into how these repairs evolve under new fully certain information.

2. **Application to richer description logics.** We investigate how the proposed characterisation interacts with more expressive fragments of the DL-Lite family. In particular, we study its application to the DL-Lite$_{\mathcal{R},\sqcap}$ language, which extends DL-Lite$_\mathcal{R}$ with conjunctions of concepts in the TBox. Indeed, the DL-Lite$_\mathcal{R}$ language forms the backbone of the OWL 2 QL profile [Motik *et al.* 2012] of the Ontology Web Language (OWL) which is devoted to query answering. However, the tractability of DL-Lite$_\mathcal{R}$ comes at the cost of expressiveness. It remains true that few methods that handle inconsistent knowledge bases exist even for the DL-Lite$_\mathcal{R}$ language, yet some interesting properties, such as expressing conjunctions of concepts in the TBox, are appealing to the community.

3. **Extension to other repair semantics.** We explore the repair process for partially preordered knowledge bases independently of the chosen repair semantics. Building on existing methods for totally preordered ABoxes, we focus on applying the lexicographic preferred-based semantics [Benferhat *et al.* 2015] as an alternative to the possibilistic semantics in the second step of the process of computing the Cπ-repair (see Figure 2.4).

The main results of this chapter have been published in [Laouar *et al.* 2025a].

4.1 Rationality Properties

We study the rationality properties of both the closure-based partially preordered possibilistic repair (Cπ-repair) and its unclosed version, the π-repair. We start by showing that these repairs satisfy the properties of the unconditional setting. Then, we show how they behave in the conditional setting. The main result of this part is the preservation of the satisfied properties when applying the deductive closure in the repair process of partially preordered possibilistic knowledge bases. In the remainder of this section, we consider a partially preordered knowledge base, namely $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$ as studied in Chapter 2 where a qualitative representation of uncertainty is used through the definition of a partial preorder relation over the ABox assertions.

4.1.1 Unconditional Inference

Let us start with unconditional query-answering, which amounts to checking whether a query q follows from the partially preordered possibilistic repair (Definition 2.2) and the closure-based partially preordered possibilistic repair (Definition 2.10), denoted by the symbols $\models^{\pi}$ and $\models^{c\pi}$, respectively. Formally:

Definition 4.1 (Unconditional inference)**.** *Let $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$ be a partially preordered knowledge base. Let q be a query:*

- *q follows from the repair $\pi(\mathcal{A}_{\trianglerighteq})$, denoted by $\mathcal{K}_{\trianglerighteq} \models^{\pi} q$ if and only if $\langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq}) \rangle \models q$.*
- *q follows from the repair $c\pi(\mathcal{A}_{\trianglerighteq})$, denoted by $\mathcal{K}_{\trianglerighteq} \models^{c\pi} q$ if and only if $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq}) \rangle \models q$.*

Where $\models$ denotes standard DL-Lite$_{\mathcal{R}}$ inference.

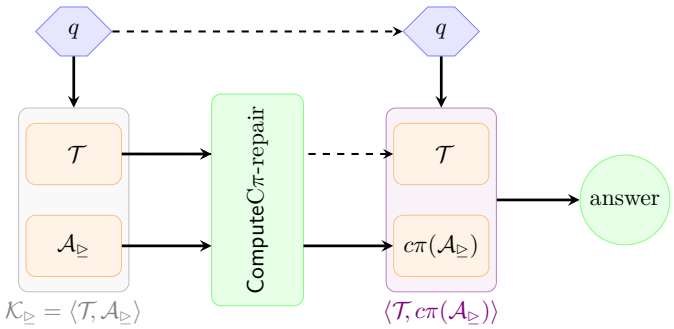


Figure 4.1: Unconditional inference under the Cπ-repair

Figure 4.1 illustrates the unconditional inference as given by Definition 4.1. Let us recall the set of rationality properties for unconditional inconsistency-tolerant entailment defined in [Baget *et al.* 2016]:

- **QCE** (Query Conjunction Elimination): if $\mathcal{K}_{\trianglerighteq} \models q_1 \wedge q_2$, then $\mathcal{K}_{\trianglerighteq} \models q_1$ and $\mathcal{K}_{\trianglerighteq} \models q_2$.
- **QCI** (Query Conjunction Introduction): if $\mathcal{K}_{\trianglerighteq} \models q_1$ and $\mathcal{K}_{\trianglerighteq} \models q_2$ then $\mathcal{K}_{\trianglerighteq} \models q_1 \wedge q_2$.
- **Cons** (Consistency): For any set of assertions $\mathcal{B}$, if $\mathcal{K}_{\trianglerighteq} \models \mathcal{B}$ then $\langle \mathcal{T}, \mathcal{B} \rangle$ is consistent.
- **ConsC** (Consistency of Conjunction): For any set of assertions $\mathcal{B}$, if $\forall \varphi \in \mathcal{B}$, $\mathcal{K}_{\trianglerighteq} \models \varphi$, then $\langle \mathcal{T}, \mathcal{B} \rangle$ is consistent.
- **ConsS** (Consistency of Support): For any set of assertions $\mathcal{B}$, if $\mathcal{K}_{\trianglerighteq} \models \mathcal{B}$ then $\exists \mathcal{A}'$, a maximally consistent subset of $\mathcal{A}_{\trianglerighteq}$ such that $\langle \mathcal{T}, \mathcal{A}' \rangle \models \mathcal{B}$.

Informally, these properties ensure that the inference behaves both in a cautious and consistent manner. **QCE** indicates that if we can safely conclude two queries together, then each of them can be concluded on its own safely. In other words, if we can conclude a conjunction, then we can separately conclude its parts. **QCI** formalises the converse of **QCE**, namely, a semantics that allow the inference of two separate conclusions must allow their inference together. **Cons** is an intuitive property, since repairs are consistent by definition, their conclusions must be consistent. **ConsC** takes consistency one step further by imposing that separate conclusions must be consistent when grouped. Finally, **ConsS** requires that a conclusion has a consistent support in the knowledge base to be inferred.

The following proposition establishes that the unconditional inference relations $\models^{\pi}$ and $\models^{c\pi}$ meet the rationality properties of unconditional inference.

Proposition 4.1. *The unconditional possibilistic inference relation $\models^{s}$ (with $s \in \{\pi, c\pi\}$) satisfies the unconditional rationality properties, namely:* ***QCE***, ***QCI***, ***Cons***, ***ConsC***, *and* ***ConsS***.

Proof. *The proof of Proposition 4.1 is straightforward. It follows from the fact that both the repairs $\pi(\mathcal{A}_{\trianglerighteq})$ and $c\pi(\mathcal{A}_{\trianglerighteq})$ are consistent (Proposition 2.1) and also follows from the fact that the unconditional inference relations $\mathcal{K}_{\trianglerighteq} \models^{\pi} q$ and $\mathcal{K}_{\trianglerighteq} \models^{c\pi} q$ amount to standard inference from consistent knowledge bases, namely $\langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq})\rangle \models q$ and $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq})\rangle \models q$, respectively:*

- *As for any consistent and certain knowledge base, the standard entailment $\models$ satisfies* ***QCE***, ***QCI***, ***Cons***, *and* ***ConsC*** *[*Baget et al. *2016]. Therefore, since $\pi(\mathcal{A}_{\trianglerighteq})$ and $c\pi(\mathcal{A}_{\trianglerighteq})$ are consistent, they both satisfy these properties.*
- *The proof of* ***ConsS*** *is also immediate. For any set of assertions $\mathcal{B}$, $\mathcal{K}_{\trianglerighteq} \models^{c\pi} \mathcal{B}$ means that $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq})\rangle \models \mathcal{B}$. Since the Cπ-repair is obtained from the intersection of the closed possibilistic repairs of the totally preordered extensions $(cl(\mathcal{R}_{\pi}(\mathcal{A}_{\geq_1})), \ldots, cl(\mathcal{R}_{\pi}(\mathcal{A}_{\geq_m})))$ of the partially preordered ABox $\mathcal{A}_{\trianglerighteq}$, a set $\mathcal{B}$ that follows from $c\pi(\mathcal{A}_{\trianglerighteq})$ can also be obtained from at least one possibilistic repair of a total preorder extension, namely $\langle \mathcal{T}, \mathcal{R}_{\pi}(\mathcal{A}_{\geq_i})\rangle \models \mathcal{B}$. Thus, there exists a maximally consistent subset of $\mathcal{A}_{\trianglerighteq}$ to entail $\mathcal{B}$. The same applies to the inference relation $\models^{\pi}$, since $\pi(\mathcal{A}_{\trianglerighteq})$ is a subset of $c\pi(\mathcal{A}_{\trianglerighteq})$.*

□

4.1.2 Conditional Inference

We now consider conditional query answering. This involves querying a partially preordered knowledge base under a given observation, represented by a set of fully reliable assertions. These assertions, denoted by $\mathcal{O}$, are assumed to be consistent with the TBox.

We write $\mathcal{O} \mid\!\sim^{s}_{\mathcal{K}_{\trianglerighteq}} q$ to indicate that q follows from the knowledge base $\mathcal{K}_{\trianglerighteq}$, under the observation $\mathcal{O}$, using the inconsistency-tolerant semantics s (here $s = \pi$ for the partially preordered possibilistic repair and $s = c\pi$ for the closure-based partially preordered possibilistic repair). A standard way to proceed is to first add $\mathcal{O}$ to the ABox with the highest priority, then apply the π-repair and the Cπ-repair methods to unconditionally answer queries from the augmented knowledge base, as given by Definition 4.1.

Let us denote by $\mathcal{K}_{\trianglerighteq_{\mathcal{O}}} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq_{\mathcal{O}}}\rangle$ the augmented knowledge base where $\mathcal{A}_{\trianglerighteq_{\mathcal{O}}} = (\mathcal{A}_{\trianglerighteq} \cup \mathcal{O})$ results from adding $\mathcal{O}$ to $\mathcal{A}_{\trianglerighteq}$ with the highest priority. The partial preorder $\trianglerighteq_{\mathcal{O}}$ over $(\mathcal{A}_{\trianglerighteq} \cup \mathcal{O})$ is obtained from $\trianglerighteq$ as follows:

(i) $\forall \varphi_i \in \mathcal{O}, \forall \varphi_j \in \mathcal{O}$: $\varphi_i \trianglerighteq_{\mathcal{O}} \varphi_j$ and $\varphi_j \trianglerighteq_{\mathcal{O}} \varphi_i$ (i.e., φ_i and φ_j are equally reliable).

(ii) $\forall \varphi_i \in \mathcal{O}, \forall \varphi_j \in \mathcal{A}_{\trianglerighteq} \setminus \mathcal{O}$: $\varphi_i \rhd_{\mathcal{O}} \varphi_j$ (i.e., every $\varphi_i \in \mathcal{O}$ is strictly more preferred than any $\varphi_j \in \mathcal{A}_{\trianglerighteq} \setminus \mathcal{O}$. This serves to give higher priority to the assertions in $\mathcal{O}$).
(iii) $\forall \varphi_i \in \mathcal{A}_{\trianglerighteq} \setminus \mathcal{O}, \forall \varphi_j \in \mathcal{A}_{\trianglerighteq} \setminus \mathcal{O}$: $\varphi_i \trianglerighteq_{\mathcal{O}} \varphi_j$ if and only if $\varphi_i \trianglerighteq \varphi_j$ (i.e., the relative ordering between the elements of $\mathcal{A}_{\trianglerighteq}$ that are not in $\mathcal{O}$ is preserved).

We use a simplified version of Example 2.1 from Chapter 2.2 to illustrate the conditional inference relations $\mid\!\sim^{\pi}_{\mathcal{K}_{\trianglerighteq}}$ and $\mid\!\sim^{c\pi}_{\mathcal{K}_{\trianglerighteq}}$.

Example 4.1. *Let $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$ be a partially preordered DL-Lite$_{\mathcal{R}}$ knowledge base, where the TBox $\mathcal{T}$ is given by:*

$$\mathcal{T} = \left\{ \begin{array}{ll} 1.\ \mathsf{Secondee} \sqsubseteq \mathsf{User} & 2.\ \mathsf{FullResearcher} \sqsubseteq \mathsf{User} \\ 3.\ \mathsf{Confidential} \sqsubseteq \mathsf{Resource} & 4.\ \mathsf{Secondee} \sqsubseteq \neg\, \mathsf{FullResearcher} \end{array} \right\}$$

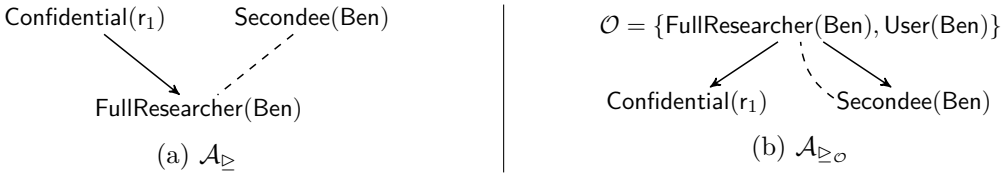
The assertions of the ABox $\mathcal{A}_{\trianglerighteq}$ are the following:

$$\mathcal{A}_{\trianglerighteq} \;=\; \boxed{\begin{array}{l} \mathsf{FullResearcher(Ben)} \\ \mathsf{Secondee(Ben)} \\ \mathsf{Confidential}(\mathsf{r}_1) \end{array}}$$

The partial preorder $\trianglerighteq$ over the assertions of the ABox is given such that $\mathsf{Secondee(Ben)} \rhd \mathsf{FullResearcher(Ben)}$ *and the remaining assertions are incomparable, namely* $\mathsf{Secondee(Ben)} \bowtie \mathsf{Confidential}(\mathsf{r}_1)$ *and* $\mathsf{FullResearcher(Ben)} \bowtie \mathsf{Confidential}(\mathsf{r}_1)$. *Figure 4.2a illustrates the partial preorder $\trianglerighteq$, where $\longrightarrow$ denotes strict preference ($\rhd$).*
Let $\mathcal{O} = \{\mathsf{FullResearcher(Ben)}, \mathsf{User(Ben)}\}$ be an observation. The augmented ABox $\mathcal{A}_{\trianglerighteq_{\mathcal{O}}} = (\mathcal{A}_{\trianglerighteq} \cup \mathcal{O})$ results from adding $\mathcal{O}$ to $\mathcal{A}_{\trianglerighteq}$ with the highest priority such that:

- $\mathsf{FullResearcher(Ben)} \rhd_{\mathcal{O}} \mathsf{User(Ben)}$ *and* $\mathsf{User(Ben)} \rhd_{\mathcal{O}} \mathsf{FullResearcher(Ben)}$.
- $\mathsf{FullResearcher(Ben)} \rhd_{\mathcal{O}} \mathsf{Secondee(Ben)}$, $\mathsf{FullResearcher(Ben)} \rhd_{\mathcal{O}} \mathsf{Confidential}(\mathsf{r}_1)$.
- $\mathsf{User(Ben)} \rhd_{\mathcal{O}} \mathsf{Secondee(Ben)}$, $\mathsf{User(Ben)} \rhd_{\mathcal{O}} \mathsf{Confidential}(\mathsf{r}_1)$.

The remaining relations of $\trianglerighteq$ are preserved in $\trianglerighteq_{\mathcal{O}}$, namely the incomparability $\mathsf{Secondee(Ben)} \bowtie \mathsf{Confidential}(\mathsf{r}_1)$. *Figure 4.2b illustrates the result of adding $\mathcal{O}$ to $\mathcal{A}_{\trianglerighteq}$ with the highest priority, where $\longrightarrow$ denotes strict preference ($\rhd$).*



(a) $\mathcal{A}_{\trianglerighteq}$ (b) $\mathcal{A}_{\trianglerighteq_{\mathcal{O}}}$

Figure 4.2: The ABox $\mathcal{A}_{\trianglerighteq_{\mathcal{O}}}$ augmented with a fully certain observation $\mathcal{O}$

Next, the definition of the partially preordered conditional query-answering relation is given as follows:

Definition 4.2 (Conditional inference)**.** *Let $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$ be a partially preordered knowledge base, $\mathcal{O}$ an observation and q a query. Then q follows from $\mathcal{K}_{\trianglerighteq}$ and $\mathcal{O}$, denoted $\mathcal{O} \mid\!\sim^{s}_{\mathcal{K}_{\trianglerighteq}} q$, if*

$\mathcal{K}_{\trianglerighteq_\mathcal{O}} \models^s q$ (with $s \in \{\pi, c\pi\}$), where $\mathcal{K}_{\trianglerighteq_\mathcal{O}} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq_\mathcal{O}} \rangle$ is the augmented knowledge base and $\trianglerighteq_\mathcal{O}$ its associated partial preorder (described in steps (i), (ii), (iii)), and $\models^s$ is the unconditional inference relation given in Definition 4.1.

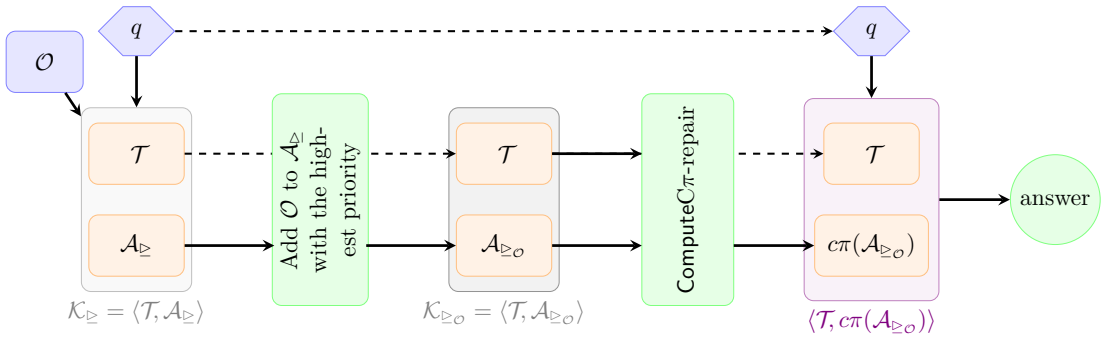


Figure 4.3: Conditional inference under the Cπ-repair in the presence of an observation $\mathcal{O}$

Figure 4.3 illustrates the conditional inference from a DL-Lite$_\mathcal{R}$ knowledge base in the presence of an observation $\mathcal{O}$ under the Cπ-repair. One can check that the conditional inference relation $\mathcal{O} \mid\!\sim^s_{\mathcal{K}_\trianglerighteq} q$ is non-monotonic under both the π-repair and the Cπ-repair methods. For instance, consider the knowledge base $\mathcal{K}_\trianglerighteq$ of Example 4.1, the conflict set of the ABox $\mathsf{Cf}(\mathcal{A}_\trianglerighteq)$ contains a single conflict, namely $\mathcal{C} = \{\mathsf{Secondee(Ben)}, \mathsf{FullResearcher(Ben)}\}$ (illustrated with dashed line in Figure 4.2). One can easily check that the assertion $\mathsf{Confidential}(\mathsf{r}_1)$ is admitted in both $\pi(\mathcal{A}_\trianglerighteq)$ and $c\pi(\mathcal{A}_\trianglerighteq)$. However, after adding the observation $\mathcal{O} = \{\mathsf{FullResearcher(Ben)}, \mathsf{User(Ben)}\}$ with the highest priority, $\mathsf{Confidential}(\mathsf{r}_1)$ is no longer admitted in $\pi(\mathcal{A}_\trianglerighteq)$ and $c\pi(\mathcal{A}_\trianglerighteq)$. That is because after obtaining the partial preorder $\trianglerighteq_\mathcal{O}$, $\mathsf{Confidential}(\mathsf{r}_1)$ does no longer have a support that is more certain than an element from the conflict $\mathcal{C}$.

One of the well-known sets of rational properties that have been proposed to study non-monotonic inference relations is System P [Kraus *et al.* 1990]. Originally defined in the context of propositional logic, it has also been adapted to DL-Lite$_\mathcal{R}$ in [Baget *et al.* 2016] (see also [Britz *et al.* 2021, Everett *et al.* 2021] for an adaptation to richer DLs). In the following, except in particular cases or when stated differently, any observation $\mathcal{O}$ is assumed to be consistent with respect to the TBox $\mathcal{T}$, i.e., $\langle \mathcal{T}, \mathcal{O} \rangle$ is consistent.

The adaptation of System P's rules is given below, where $\mathcal{K}_\trianglerighteq = \langle \mathcal{T}, \mathcal{A}_\trianglerighteq \rangle$ is a knowledge base, $\mathcal{O}_1$, $\mathcal{O}_2$, $\mathcal{O}_3$ are observations (or consistent subsets of assertions), s is an inconsistency-tolerant semantics with $s \in \{\pi, c\pi\}$, $\models$ and $\equiv$ denote standard DL-Lite$_\mathcal{R}$ inference and equivalence:

- **R** (Reflexivity) $\mathcal{O}_1 \mid\!\sim^s_{\mathcal{K}_\trianglerighteq} \mathcal{O}_1$.
- **LLE** (Left Logical Equivalence): if $\langle \mathcal{T}, \mathcal{O}_1 \rangle \equiv \langle \mathcal{T}, \mathcal{O}_2 \rangle$ and $\mathcal{O}_1 \mid\!\sim^s_{\mathcal{K}_\trianglerighteq} \mathcal{O}_3$ then $\mathcal{O}_2 \mid\!\sim^s_{\mathcal{K}_\trianglerighteq} \mathcal{O}_3$.
- **RW** (Right Weakening): if $\langle \mathcal{T}, \mathcal{O}_1 \rangle \models \mathcal{O}_2$ and $\mathcal{O}_3 \mid\!\sim^s_{\mathcal{K}_\trianglerighteq} \mathcal{O}_1$, then $\mathcal{O}_3 \mid\!\sim^s_{\mathcal{K}_\trianglerighteq} \mathcal{O}_2$.
- **Cut**: if $\mathcal{O}_1 \mid\!\sim^s_{\mathcal{K}_\trianglerighteq} \mathcal{O}_2$ and $\mathcal{O}_1 \cup \mathcal{O}_2 \mid\!\sim^s_{\mathcal{K}_\trianglerighteq} \mathcal{O}_3$, then $\mathcal{O}_1 \mid\!\sim^s_{\mathcal{K}_\trianglerighteq} \mathcal{O}_3$.
- **CM** (Cautious Monotony): if $\mathcal{O}_1 \mid\!\sim^s_{\mathcal{K}_\trianglerighteq} \mathcal{O}_2$ and $\mathcal{O}_1 \mid\!\sim^s_{\mathcal{K}_\trianglerighteq} \mathcal{O}_3$, then $\mathcal{O}_1 \cup \mathcal{O}_2 \mid\!\sim^s_{\mathcal{K}_\trianglerighteq} \mathcal{O}_3$.
- **AND**: if $\mathcal{O}_1 \mid\!\sim^s_{\mathcal{K}_\trianglerighteq} \mathcal{O}_2$ and $\mathcal{O}_1 \mid\!\sim^s_{\mathcal{K}_\trianglerighteq} \mathcal{O}_3$, then $\mathcal{O}_1 \mid\!\sim^s_{\mathcal{K}_\trianglerighteq} \mathcal{O}_2 \cup \mathcal{O}_3$.

In our analysis, we propose also to consider two additional properties, originally defined in propositional logic [Kraus *et al.* 1990, Makinson 1994], that go beyond cautious monotony (where $s \in \{\pi, c\pi\}$):

- **RM** (Rational Monotony): if $\mathcal{O}_1 \mid\!\sim^{s}_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}_2$, then $\mathcal{O}_1 \cup \mathcal{O}_3 \mid\!\sim^{s}_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}_2$ or $\langle \mathcal{T}, (\mathcal{O}_1 \cup \mathcal{O}_3 \cup s(\mathcal{A}_{\trianglerighteq}))\rangle$ is inconsistent.
- **Comp** (Completeness): if $\mathcal{O}_1 \mid\!\sim^{s}_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}_2$, then either $\mathcal{O}_1 \cup \mathcal{O}_3 \mid\!\sim^{s}_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}_2$ or $\langle \mathcal{T}, (\mathcal{O}_1 \cup \mathcal{O}_2 \cup \mathcal{O}_3 \cup s(\mathcal{A}_{\trianglerighteq}))\rangle$ is inconsistent.

Note that in the adaptation of the last two properties, we propose to use the notion of inconsistency instead of negation in the original definition of rational monotony, and use a disjunctive interpretation of **RM**. In propositional logic, **RM** is defined as: if $\alpha \mid\!\sim_{\mathcal{K}} \gamma$ and $\alpha \not\mid\!\sim_{\mathcal{K}} \neg\beta$, then $\alpha \wedge \beta \mid\!\sim_{\mathcal{K}} \gamma$, where α, β and γ are propositional logic formulas. Our adaptation consists first in rewriting **RM** equivalently in a disjunctive way: if $\alpha \mid\!\sim_{\mathcal{K}} \gamma$, then $\alpha \wedge \beta \mid\!\sim_{\mathcal{K}} \gamma$ or $\alpha \mid\!\sim_{\mathcal{K}} \neg\beta$. Lastly, we replace $\alpha \mid\!\sim_{\mathcal{K}} \neg\beta$ with $\alpha \wedge \beta$ is inconsistent with the knowledge base. Here, **RM** states that given a new observation, we can continue to believe in the previous plausible consequences of the knowledge base, or the new observation conflicts with the knowledge base.

In propositional logic, **Comp** is defined as: if $\alpha \mid\!\sim_{\mathcal{K}} \gamma$, then either $\alpha \wedge \beta \mid\!\sim_{\mathcal{K}} \gamma$ or $\alpha \wedge \beta \mid\!\sim_{\mathcal{K}} \neg\gamma$. Here, we simply replace $\alpha \wedge \beta \mid\!\sim_{\mathcal{K}} \neg\gamma$ with $\alpha \wedge \beta \wedge \gamma$ is inconsistent with the knowledge base. The **Comp** rule is stronger than **RM**, and states that given a new observation $\mathcal{O}_3$, then either $\mathcal{O}_2$ continues to be derived from both $\mathcal{O}_1$ and $\mathcal{O}_3$, or $\mathcal{O}_3$ contradicts the whole knowledge base (plus itself). Intuitively, this means that either we continue to believe in $\mathcal{O}_2$, or we should believe in its negation (there is no room for ignoring $\mathcal{O}_2$).

The following proposition summarises the results of the conditional properties:

Proposition 4.2. *Let $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq}\rangle$ be a partially preordered knowledge base, $\mathcal{O}$ be an observation and q be a query. The query-answering relations $\mid\!\sim^{\pi}_{\mathcal{K}_{\trianglerighteq}}$ and $\mid\!\sim^{c\pi}_{\mathcal{K}_{\trianglerighteq}}$ satisfy the System P properties: **R** (Reflexivity), **LLE** (Left Logical Equivalence), **RW** (Right Weakening), **Cut**, **CM** (Cautious Monotony) and **AND**.*

Remark Note that the query-answering relations $\mid\!\sim^{\pi}_{\mathcal{K}_{\trianglerighteq}}$ and $\mid\!\sim^{c\pi}_{\mathcal{K}_{\trianglerighteq}}$ fail to satisfy **RM** (Rational Monotony) and **Comp** (Completeness). The following example illustrates a case where the inference relations $\mid\!\sim^{\pi}_{\mathcal{K}_{\trianglerighteq}}$ and $\mid\!\sim^{c\pi}_{\mathcal{K}_{\trianglerighteq}}$ do not satisfy the properties **RM** and **Comp**.

Example 4.1 (Continued)**.** *Let us continue Example 4.1 and show through the addition of Observation $\mathcal{O}$ to the ABox $\mathcal{A}_{\trianglerighteq}$ as illustrated by Figure 4.2 that the conditional inference relations $\mid\!\sim^{\pi}_{\mathcal{K}_{\trianglerighteq}}$ and $\mid\!\sim^{c\pi}_{\mathcal{K}_{\trianglerighteq}}$ do not satisfy **RM** and **Comp**. The observation added with the highest priority is:*

$$\mathcal{O} = \{\mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{User}(\mathsf{Ben})\}$$

In addition, assume the following sets of assertions:

$$\mathcal{B}_1 = \emptyset, \qquad \mathcal{B}_2 = \{\mathsf{Resource}(\mathsf{r}_1)\}$$

1. *We provide a counterexample to **Rational Monotony** using the sets $\mathcal{B}_1$, $\mathcal{B}_2$ and the observation $\mathcal{O}$, namely:*
   If $\mathcal{B}_1 \mid\!\sim^{s}_{\mathcal{K}_{\trianglerighteq}} \mathcal{B}_2$, then $\mathcal{B}_1 \cup \mathcal{O} \mid\!\sim^{s}_{\mathcal{K}_{\trianglerighteq}} \mathcal{B}_2$ or $\langle \mathcal{T}, \mathcal{B}_1 \cup \mathcal{O} \cup s(\mathcal{A}_{\trianglerighteq})\rangle$ is inconsistent (where $s \in \{\pi, c\pi\}$).
   One can check that the partially preordered possibilistic repair is $\pi(\mathcal{A}_{\trianglerighteq}) = \{\mathsf{Confidential}(\mathsf{r}_1)\}$ and the $C\pi$-repair is $c\pi(\mathcal{A}_{\trianglerighteq}) = \{\mathsf{Confidential}(\mathsf{r}_1), \mathsf{Resource}(\mathsf{r}_1)\}$.
   Using the unconditional inference relations $\models^{\pi}$ and $\models^{c\pi}$, we have that $\langle \mathcal{T}, \mathcal{A}_{\trianglerighteq}\rangle \models^{\pi} \mathcal{B}_2$ and $\langle \mathcal{T}, \mathcal{A}_{\trianglerighteq}\rangle \models^{\pi} \mathcal{B}_2$. Moreover, the set $\mathcal{B}_1$ is an empty set, hence by definition of the

conditional inference:

$$\mathcal{B}_1 \mid\!\sim^{\pi}_{\mathcal{K}_{\trianglerighteq}} \mathcal{B}_2 \quad \text{and} \quad \mathcal{B}_1 \mid\!\sim^{c\pi}_{\mathcal{K}_{\trianglerighteq}} \mathcal{B}_2$$

Now, after adding the observation $\mathcal{O} = \{\mathsf{FullResearcher(Ben)}, \mathsf{User(Ben)}\}$ *to* $\mathcal{A}_{\trianglerighteq}$ *with the highest priority, as illustrated by Figure 4.2b, the π-repair and the Cπ-repair of* $\mathcal{A}_{\trianglerighteq_{\mathcal{O}}}$ *change, resulting in the sets* $\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}}}) = \{\mathsf{FullResearcher(Ben)}, \mathsf{User(Ben)}\}$ *and* $c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}}}) = \{\mathsf{FullResearcher(Ben)}, \mathsf{User(Ben)}\}$. *One can easily check that* $\mathsf{Resource}(\mathsf{r}_1)$ *is no longer entailed by* $\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}}})$ *and* $c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}}})$, *hence,* $\langle \mathcal{T}, \mathcal{A}_{\trianglerighteq_{\mathcal{O}}}\rangle \not\models^{\pi} \mathcal{B}_2$ *and* $\langle \mathcal{T}, \mathcal{A}_{\trianglerighteq_{\mathcal{O}}}\rangle \not\models^{c\pi} \mathcal{B}_2$. *Using the conditional inference relations:*

$$\mathcal{B}_1 \cup \mathcal{O} \not\mid\!\sim^{\pi}_{\mathcal{K}_{\trianglerighteq}} \mathcal{B}_2 \quad \text{and} \quad \mathcal{B}_1 \cup \mathcal{O} \not\mid\!\sim^{c\pi}_{\mathcal{K}_{\trianglerighteq}} \mathcal{B}_2$$

Besides, one can easily check that $\langle \mathcal{T}, \mathcal{B}_1 \cup \mathcal{O} \cup \pi(\mathcal{A}_{\trianglerighteq})\rangle$ *and* $\langle \mathcal{T}, \mathcal{B}_1 \cup \mathcal{O} \cup c\pi(\mathcal{A}_{\trianglerighteq})\rangle$ *are both consistent, which contradicts the definition of the* ***RM*** *property. Therefore,* ***RM*** *(Rational Monotony) is not satisfied by both of the conditional inference relations* $\mid\!\sim^{\pi}_{\mathcal{K}_{\trianglerighteq}}$ *and* $\mid\!\sim^{c\pi}_{\mathcal{K}_{\trianglerighteq}}$.

2. *Now, in the same way, we provide a counterexample to* ***Completeness*** *using the same sets* $\mathcal{B}_1$, $\mathcal{B}_2$ *and the observation* $\mathcal{O}$ *namely:*
If $\mathcal{B}_1 \mid\!\sim^{s}_{\mathcal{K}_{\trianglerighteq}} \mathcal{B}_2$, *then either* $\mathcal{B}_1 \cup \mathcal{O} \mid\!\sim^{s}_{\mathcal{K}_{\trianglerighteq}} \mathcal{B}_2$ *or* $\langle \mathcal{T}, (\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{O} \cup s(\mathcal{A}_{\trianglerighteq}))\rangle$ *is inconsistent (where* $s \in \{\pi, c\pi\}$*).*
We have already shown that $\mathcal{B}_1 \mid\!\sim^{\pi}_{\mathcal{K}_{\trianglerighteq}} \mathcal{B}_2$ *and* $\mathcal{B}_1 \mid\!\sim^{c\pi}_{\mathcal{K}_{\trianglerighteq}} \mathcal{B}_2$. *Moreover, after introducing* $\mathcal{O}$ *to* $\mathcal{A}_{\trianglerighteq}$ *with the highest priority, we cannot infer* $\mathcal{B}_2$ *any more:*

$$\mathcal{B}_1 \cup \mathcal{O} \not\mid\!\sim^{\pi}_{\mathcal{K}_{\trianglerighteq}} \mathcal{B}_2 \quad \text{and} \quad \mathcal{B}_1 \cup \mathcal{O} \not\mid\!\sim^{c\pi}_{\mathcal{K}_{\trianglerighteq}} \mathcal{B}_2$$

One can also easily check that $\langle \mathcal{T}, \mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{O} \cup \pi(\mathcal{A}_{\trianglerighteq})\rangle$ *is consistent, and* $\langle \mathcal{T}, \mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{O} \cup c\pi(\mathcal{A}_{\trianglerighteq})\rangle$ *is consistent. This contradicts the definition of* ***Comp****. Therefore,* ***Comp*** *is not satisfied by both of the conditional inference relations* $\mid\!\sim^{\pi}_{\mathcal{K}_{\trianglerighteq}}$ *and* $\mid\!\sim^{c\pi}_{\mathcal{K}_{\trianglerighteq}}$.

4.1.3 Satisfaction of System P Properties

In this section, we provide an in-depth study of the satisfaction of the System P properties by the conditional inference relations of the Cπ-repair and the π-repair, as established by Proposition 4.2. The proofs of the properties **R**, **RW** and **AND** are directly given since they are relatively easy to establish. Then, in Subsection 4.1.3 we establish intermediary results on the satisfaction of the properties **LLE**, **Cut** and **CM** in the context of totally preordered ABoxes before showing how their satisfaction is achieved in the case of partially preordered ABoxes, in Subsections 4.1.3 and 4.1.3.

Lemma 4.1. *Let* $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq}\rangle$ *be a partially preordered knowledge base,* $\mathcal{O}$ *be an observation and* q *be a query. The conditional query-answering relations* $\mid\!\sim^{\pi}_{\mathcal{K}_{\trianglerighteq}}$ *and* $\mid\!\sim^{c\pi}_{\mathcal{K}_{\trianglerighteq}}$ *satisfy the properties* ***R****,* ***RW*** *and* ***AND****.*

Proof.

1. *The proof of* ***R*** *is immediate. The property of reflexivity states that, for a given observation* $\mathcal{O}$, $\mathcal{O} \mid\!\sim^{s}_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}$ *holds (where* $s \in \{\pi, c\pi\}$*). By Definition 4.2 of the conditional*

inference, $\mathcal{O} \mid\sim^{s}_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}$ amounts to $\mathcal{K}_{\trianglerighteq_{\mathcal{O}}} \models^{s} \mathcal{O}$ holds, where $\mathcal{K}_{\trianglerighteq_{\mathcal{O}}} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq_{\mathcal{O}}} \rangle$, such that $\mathcal{A}_{\trianglerighteq_{\mathcal{O}}}$ is the augmented ABox (as described in steps (i), (ii), (iii) from Section 4.1.2), and $\models^{s}$ is the unconditional inference relation (Definition 4.1).
Let $\mathcal{A}_{\geq_{\mathcal{O}}}$ be a totally preordered ABox compatible with $\mathcal{A}_{\trianglerighteq_{\mathcal{O}}}$. Let $\mathcal{S}_1 \cup \cdots \cup \mathcal{S}_n$ be its associated well-ordered partition. One can easily see that $\mathcal{S}_1 = \mathcal{O}$. Indeed, by construction of $\trianglerighteq_{\mathcal{O}}$, we have $\forall \varphi_j \in \mathcal{O}$, $\forall \varphi_k \in \mathcal{O}$, $\varphi_j \trianglerighteq_{\mathcal{O}} \varphi_k$ and $\varphi_k \trianglerighteq_{\mathcal{O}} \varphi_j$, hence, we also have $\varphi_j \geq_{\mathcal{O}} \varphi_k$ and $\varphi_k \geq_{\mathcal{O}} \varphi_j$ since $\geq_{\mathcal{O}}$ is compatible with $\trianglerighteq_{\mathcal{O}}$. Besides, again by construction of $\trianglerighteq_{\mathcal{O}}$, $\forall \varphi_j \in \mathcal{O}$, $\forall \varphi_k \in (\mathcal{A} \setminus \mathcal{O})$, we have $\varphi_j \rhd_{\mathcal{O}} \varphi_k$, hence, $\varphi_j >_{\mathcal{O}} \varphi_k$. The possibilistic repair (Definition 1.24) of $\mathcal{A}_{\geq_{\mathcal{O}}}$ is necessarily of the form:

- *$\mathcal{R}_{\pi}(\mathcal{A}_{\geq_{\mathcal{O}}}) = \mathcal{A}_{\geq_{\mathcal{O}}}$ if $\langle \mathcal{T}, (\mathcal{S}_1 = \mathcal{O}) \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_n \rangle$ is consistent.*
- *$\mathcal{R}_{\pi}(\mathcal{A}_{\geq_{\mathcal{O}}}) = (\mathcal{S}_1 = \mathcal{O}) \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_i$ if $\langle \mathcal{T}, (\mathcal{S}_1 = \mathcal{O}) \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_i \rangle$ is consistent and $\langle \mathcal{T}, (\mathcal{S}_1 = \mathcal{O}) \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_{i+1} \rangle$ is inconsistent (with $1 \leq i < n$).*

Therefore, $\mathcal{R}_{\pi}(\mathcal{A}_{\geq_{\mathcal{O}}})$ contains at least $\mathcal{O}$ (by assumption $\langle \mathcal{T}, \mathcal{O} \rangle$ is consistent). Since this result applies to all the totally preordered ABoxes compatible with $\mathcal{A}_{\trianglerighteq_{\mathcal{O}}}$, we conclude that $\mathcal{O} \subseteq \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}}})$ (resp. $\mathcal{O} \subseteq c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}}})$). Hence, $\langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}}}) \rangle \models \mathcal{O}$ (resp. $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}}}) \rangle \models \mathcal{O}$).

2. ***RW** states that if $\langle \mathcal{T}, \mathcal{O}_1 \rangle \models \mathcal{O}_2$ and $\mathcal{O}_3 \mid\sim^{s}_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}_1$ then $\mathcal{O}_3 \mid\sim^{s}_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}_2$ (where $s \in \{\pi, c\pi\}$). The proof is immediate. Indeed, $\mathcal{O}_3 \mid\sim^{s}_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}_1$ amounts to apply the standard entailment $\langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_3}}) \rangle \models \mathcal{O}_1$ (resp. $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_3}}) \rangle \models \mathcal{O}_1$).*
   *By definition of a possibilistic repair, $\langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_3}}) \rangle$ (resp. $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_3}}) \rangle$) is consistent. Furthermore, from the result of Proposition 4.1, consistency (**Cons**) is satisfied by the unconditional inference relation, thus, we can infer that $\langle \mathcal{T}, \mathcal{O}_1 \rangle$ is consistent from $\langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_3}}) \rangle \models \mathcal{O}_1$ (resp. $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_3}}) \rangle \models \mathcal{O}_1$). Therefore, having $\langle \mathcal{T}, \mathcal{O}_1 \rangle \models \langle \mathcal{T}, \mathcal{O}_2 \rangle$, one can directly infer $\langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_3}}) \rangle \models \mathcal{O}_2$ (resp. $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_3}}) \rangle \models \mathcal{O}_2$).*
3. ***AND** states that $\mathcal{O}_1 \mid\sim^{s}_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}_2$ and $\mathcal{O}_1 \mid\sim^{s}_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}_3$, then $\mathcal{O}_1 \mid\sim^{s}_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}_2 \cup \mathcal{O}_3$ (where $s \in \{\pi, c\pi\}$). The proof of **AND** is also immediate. Indeed, $\mathcal{O}_1 \mid\sim^{s}_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}_2$ amounts to derive $\mathcal{K}_{\trianglerighteq_{\mathcal{O}_1}} \models^{s} \mathcal{O}_2$, where $\models^{s}$ is the unconditional inference relation. Similarly, $\mathcal{O}_1 \mid\sim^{s}_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}_3$ amounts to derive $\mathcal{K}_{\trianglerighteq_{\mathcal{O}_1}} \models^{s} \mathcal{O}_3$ (where $s \in \{\pi, c\pi\}$).*
   *Proposition 4.1 shows that both inference relations $\models^{\pi}$ and $\models^{c\pi}$ satisfy the **QCI** property, which states that if $\mathcal{K}_{\trianglerighteq_{\mathcal{O}_1}} \models^{s} q_1$ and $\mathcal{K}_{\trianglerighteq_{\mathcal{O}_1}} \models^{s} q_2$ then $\mathcal{K}_{\trianglerighteq_{\mathcal{O}_1}} \models^{s} q_1 \wedge q_2$. In our case, we have $\mathcal{K}_{\trianglerighteq_{\mathcal{O}_1}} \models^{\pi} \mathcal{O}_2$ and $\mathcal{K}_{\trianglerighteq_{\mathcal{O}_1}} \models^{\pi} \mathcal{O}_3$ (resp. $\mathcal{K}_{\trianglerighteq_{\mathcal{O}_1}} \models^{c\pi} \mathcal{O}_2$ and $\mathcal{K}_{\trianglerighteq_{\mathcal{O}_1}} \models^{c\pi} \mathcal{O}_3$), therefore, we conclude that $\mathcal{K}_{\trianglerighteq_{\mathcal{O}_1}} \models^{\pi} \mathcal{O}_2 \cup \mathcal{O}_3$ (resp. $\mathcal{K}_{\trianglerighteq_{\mathcal{O}_1}} \models^{c\pi} \mathcal{O}_2 \cup \mathcal{O}_3$). (Note that when $\mathcal{O}_1 = \emptyset$ (no assertions observed) and q_1 and q_2 are sets of assertions, then **AND** is equivalent to **QCI** [Baget et al. 2016]).*

□

Some preliminaries on the conditional inference for totally preordered ABoxes

In this subsection, we first establish the satisfaction of **LLE** in the context of a totally preordered ABox. We also discuss the behaviour of the conditional inference relation when previously derived results are added to the base, still within the context of totally preordered ABoxes.

In the following, we use the expression "conditioning a partial preorder $\trianglerighteq$ by $\mathcal{O}$" to denote the process of adding an observation $\mathcal{O}$ with the highest priority to a partially preordered ABox $\mathcal{A}_{\trianglerighteq}$ following steps (i), (ii), (iii) from Section 4.1.2. The obtained partial preorder $\trianglerighteq_{\mathcal{O}}$ is also called a conditioned partial preorder.

The following definition introduces the instantiation (given in steps (i), (ii), (iii) from Sec-

tion 4.1.2) of conditioning a total preorder $\geq$ by an observation $\mathcal{O}$.

Definition 4.3 (Conditioned total preorder)**.** *Let $\mathcal{A}_{\geq}$ be a totally preordered ABox, and $\mathcal{S}_1 \cup \cdots \cup \mathcal{S}_n$ be its associated well-ordered partition. Let $\mathcal{O}$ be an observation.*
The conditioned total preorder $\geq_{\mathcal{O}}$ of $\geq$ by $\mathcal{O}$ is obtained by adding $\mathcal{O}$ to $\mathcal{A}_{\geq}$ with the highest priority as follows:

1. $\forall \phi_j \in \mathcal{O}$, $\forall \phi_k \in \mathcal{O}$, $\phi_j \geq_{\mathcal{O}} \phi_k$ *and* $\phi_k \geq_{\mathcal{O}} \phi_j$.
2. $\forall \phi_j \in \mathcal{O}$, $\forall \phi_k \in (\mathcal{S}_i \setminus \mathcal{O})$ *(with* $1 \leq i \leq n$*)*, $\phi_j >_{\mathcal{O}} \phi_k$.
3. $\forall \phi_j \in (\mathcal{S}_i \setminus \mathcal{O})$, $\forall \phi_k \in (\mathcal{S}_l \setminus \mathcal{O})$ *(with* $1 \leq i, l \leq n$*)*, $\phi_j \geq_{\mathcal{O}} \phi_k$ *if and only if* $\phi_j \geq \phi_k$.

The resulting ABox is given by the well-ordered partition: $\mathcal{A}_{\geq_{\mathcal{O}}} = \mathcal{O} \cup (\mathcal{S}_1 \setminus \mathcal{O}) \cup \cdots \cup (\mathcal{S}_n \setminus \mathcal{O})$

Let us begin by establishing the property **LLE**.

Lemma 4.2 (**LLE** for totally preordered KBs)**.** *Let $\mathcal{K}_{\geq} = \langle \mathcal{T}, \mathcal{A}_{\geq} \rangle$ be a totally preordered knowledge base, $\mathcal{O}$ be an observation and q be a query. The query-answering relation $\mathcal{O} \mid\!\sim^{s}_{\mathcal{K}_{\geq}} q$ satisfies **LLE** (where $s \in \{\pi, c\pi\}$).*

Proof. *We recall that **LLE** is defined by:*
If $\langle \mathcal{T}, \mathcal{O}_1 \rangle \equiv \langle \mathcal{T}, \mathcal{O}_2 \rangle$ and $\mathcal{O}_1 \mid\!\sim^{s}_{\mathcal{K}_{\geq}} \mathcal{O}_3$ then $\mathcal{O}_2 \mid\!\sim^{s}_{\mathcal{K}_{\geq}} \mathcal{O}_3$ (where $s \in \{\pi, c\pi\}$).
In the case of totally preordered knowledge base, it is enough to show that if $\langle \mathcal{T}, \mathcal{O}_1 \rangle \equiv \langle \mathcal{T}, \mathcal{O}_2 \rangle$ then $\langle \mathcal{T}, \mathcal{R}_{\pi}(\mathcal{A}_{\geq_{\mathcal{O}_1}}) \rangle \equiv \langle \mathcal{T}, \mathcal{R}_{\pi}(\mathcal{A}_{\geq_{\mathcal{O}_2}}) \rangle$.
We will first use the following observation: let $\mathcal{S}$ be a set of assertions. Then if $\langle \mathcal{T}, \mathcal{O}_1 \rangle \equiv \langle \mathcal{T}, \mathcal{O}_2 \rangle$ and $\langle \mathcal{T}, \mathcal{O}_1 \cup \mathcal{S} \rangle$ is consistent then $\langle \mathcal{T}, \mathcal{O}_2 \cup \mathcal{S} \rangle$ is also consistent, and if $\langle \mathcal{T}, \mathcal{O}_1 \cup \mathcal{S} \rangle$ is inconsistent then $\langle \mathcal{T}, \mathcal{O}_2 \cup \mathcal{S} \rangle$ is also inconsistent …(*).
Let $\mathcal{S}_1 \cup \cdots \cup \mathcal{S}_n$ be the well-ordered partition associated with $\mathcal{A}_{\geq}$. The result of the conditioning of $\geq$ by $\mathcal{O}_1$ is given by: $\mathcal{A}_{\geq_{\mathcal{O}_1}} = \mathcal{O}_1 \cup (\mathcal{S}_1 \setminus \mathcal{O}_1) \cup \cdots \cup (\mathcal{S}_n \setminus \mathcal{O}_1)$. Assume that the possibilistic repair $\mathcal{R}_{\pi}(\mathcal{A}_{\geq_{\mathcal{O}_1}}) = \mathcal{O}_1 \cup (\mathcal{S}_1 \setminus \mathcal{O}_1) \cup \cdots \cup (\mathcal{S}_i \setminus \mathcal{O}_1)$, then, $\langle \mathcal{T}, \mathcal{O}_1 \cup (\mathcal{S}_1 \setminus \mathcal{O}_1) \cup \cdots \cup (\mathcal{S}_i \setminus \mathcal{O}_1) \rangle$ is consistent and $\langle \mathcal{T}, \mathcal{O}_1 \cup (\mathcal{S}_1 \setminus \mathcal{O}_1) \cup \cdots \cup (\mathcal{S}_{i+1} \setminus \mathcal{O}_1) \rangle$ is inconsistent.
In a trivial way, $\mathcal{O}_1 \cup (\mathcal{S}_1 \setminus \mathcal{O}_1) \cup \cdots \cup (\mathcal{S}_i \setminus \mathcal{O}_1)$ can be simplified to $\mathcal{O}_1 \cup \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_i$. Therefore, the possibilistic repair is given by $\mathcal{R}_{\pi}(\mathcal{A}_{\geq_{\mathcal{O}_1}}) = \mathcal{O}_1 \cup \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_i$.
In the same way, the conditioning of $\geq$ by $\mathcal{O}_2$ gives as result $\mathcal{A}_{\geq_{\mathcal{O}_2}} = \mathcal{O}_2 \cup (\mathcal{S}_1 \setminus \mathcal{O}_2) \cup \cdots \cup (\mathcal{S}_n \setminus \mathcal{O}_2)$. This can also be simplified by $\mathcal{A}_{\geq_{\mathcal{O}_2}} = \mathcal{O}_2 \cup \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_n$. Thus, based on the observation (*), if $\langle \mathcal{T}, \mathcal{O}_1 \cup \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_i \rangle$ is consistent then $\langle \mathcal{T}, \mathcal{O}_2 \cup \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_i \rangle$ is consistent, and if $\langle \mathcal{T}, \mathcal{O}_1 \cup \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_{i+1} \rangle$ is inconsistent then $\langle \mathcal{T}, \mathcal{O}_2 \cup \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_{i+1} \rangle$ is inconsistent. This yields a repair for $\mathcal{A}_{\geq_{\mathcal{O}_2}}$ such that $\mathcal{R}_{\pi}(\mathcal{A}_{\geq_{\mathcal{O}_2}}) = \mathcal{O}_2 \cup \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_i$. Therefore, since $\langle \mathcal{T}, \mathcal{O}_1 \rangle \equiv \langle \mathcal{T}, \mathcal{O}_2 \rangle$, we conclude that $\langle \mathcal{T}, \mathcal{R}_{\pi}(\mathcal{A}_{\geq_{\mathcal{O}_1}}) \rangle \equiv \langle \mathcal{T}, \mathcal{R}_{\pi}(\mathcal{A}_{\geq_{\mathcal{O}_2}}) \rangle$. □

Now, we introduce two properties regarding the notion of repair in totally preordered ABoxes. These two properties will be useful to establish the satisfaction of **Cut** and **CM** when dealing with partially preordered ABoxes, see Subsection 4.1.3. The first property (given by Lemma 4.3) states that, in the context of a totally preordered knowledge base, adding plausible conclusions to a possibilistic repair does not change the repair; to be more precise, the result of these additions leads to an equivalent result. More formally:

Lemma 4.3. *Let $\mathcal{K}_{\geq} = \langle \mathcal{T}, \mathcal{A}_{\geq} \rangle$ be a totally preordered knowledge base such that $\langle \mathcal{T}, \mathcal{R}_{\pi}(\mathcal{A}_{\geq}) \rangle \models f$, where f is a given assertion (not necessarily in $\mathcal{A}_{\geq}$). Let $\geq_f$ be the conditioning of $\geq$ by $\{f\}$. Then $\langle \mathcal{T}, \mathcal{R}_{\pi}(\mathcal{A}_{\geq}) \rangle \equiv \langle \mathcal{T}, \mathcal{R}_{\pi}(\mathcal{A}_{\geq_f}) \rangle$ and $\{f\} \cup \mathcal{R}_{\pi}(\mathcal{A}_{\geq}) = \mathcal{R}_{\pi}(\mathcal{A}_{\geq_f})$.*

Proof. *Assume that* $\langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_\geq)\rangle \models f$ *and let us show that indeed:*

$$\langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_\geq)\rangle \equiv \langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_{\geq_f})\rangle$$

Let $\mathcal{S}_1 \cup \cdots \cup \mathcal{S}_n$ *be the well-ordered partition associated with* $\mathcal{A}_\geq$*. By Definition 1.24, the possibilistic repair is given by* $\mathcal{R}_\pi(\mathcal{A}_\geq) = \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_i$ *such that* $\langle \mathcal{T}, \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_i\rangle$ *is consistent and* $\langle \mathcal{T}, \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_{i+1}\rangle$ *is inconsistent. From the assumption, we have* $\langle \mathcal{T}, \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_i\rangle \models f$*. The conditioning of* $\geq$ *by* $\{f\}$ *is given by* $\mathcal{A}_{\geq_f} = \{f\} \cup (\mathcal{S}_1 \setminus \{f\}) \cup \cdots \cup (\mathcal{S}_n \setminus \{f\})$*.*
Since $\langle \mathcal{T}, \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_{i+1}\rangle$ *is inconsistent, one can directly infer that* $\langle \mathcal{T}, \{f\} \cup (\mathcal{S}_1 \setminus \{f\}) \cup \cdots \cup (\mathcal{S}_{i+1} \setminus \{f\})\rangle$ *is inconsistent. Moreover, since* $\langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_\geq)\rangle \models f$ *then* $\langle \mathcal{T}, \{f\} \cup \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_i\rangle$ *is consistent, thus* $\langle \mathcal{T}, \{f\} \cup (\mathcal{S}_1 \setminus \{f\}) \cup \cdots \cup (\mathcal{S}_i \setminus \{f\})\rangle$ *is consistent. It follows that* $\mathcal{R}_\pi(\mathcal{A}_{\geq_f}) = \{f\} \cup (\mathcal{S}_1 \setminus \{f\}) \cup \cdots \cup (\mathcal{S}_i \setminus \{f\})$*. Therefore,* $\mathcal{R}_\pi(\mathcal{A}_\geq)$ *and* $\mathcal{R}_\pi(\mathcal{A}_{\geq_f})$ *are equivalent with respect to the TBox* $\mathcal{T}$*.* □

The second property is simply a generalisation of Lemma 4.3 to the case where a set of assertions (instead of just one assertion) is derived:

Lemma 4.4. *Let* $\mathcal{K}_\geq = \langle \mathcal{T}, \mathcal{A}_\geq\rangle$ *be a totally preordered knowledge base such that* $\langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_\geq)\rangle \models \mathcal{O}$*, where* $\mathcal{O}$ *is a set of assertions consistent with respect to* $\mathcal{T}$*. Let* $\geq_\mathcal{O}$ *be the conditioning of* $\geq$ *by* $\mathcal{O}$*. Then* $\langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_\geq)\rangle \equiv \langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_{\geq_\mathcal{O}})\rangle$ *and* $\mathcal{O} \cup \mathcal{R}_\pi(\mathcal{A}_\geq) = \mathcal{R}_\pi(\mathcal{A}_{\geq_\mathcal{O}})$*.*

Proof. *The proof of this lemma follows directly from the proof of Lemma 4.3, where it is enough to apply Lemma 4.3 iteratively to each assertion from* $\mathcal{O}$*.* □

On the relationship between a partial preorder and its compatible total preorders when conditioning

This subsection introduces the relationship between the set of the total preorders compatible with the initial partial preorder, and the set of the total preorders compatible with the partial preorder obtained after adding a new observation with the highest priority.

Lemma 4.5 makes it possible to establish the link between the conditioning of a partial preorder and the conditioning of the total preorders which are compatible with it. More precisely, the lemma states that:

i) if you start from a set (noted X in Lemma 4.5) of total preorders compatible with a partial preorder $\trianglerighteq$; and
ii) you condition each of the total preorders by the observation $\mathcal{O}$;
iii) then the set obtained (noted Y in Lemma 4.5) is equal to the set (noted Z in Lemma 4.5) of the total preorders compatible with the partial preorder conditioned by the observation $\mathcal{O}$.

More formally:

Lemma 4.5. *Let* $\mathcal{A}_\trianglerighteq$ *be a partially preordered ABox. Let* $\mathcal{O}$ *be an observation. Let* $\mathcal{A}_{\trianglerighteq_\mathcal{O}}$ *be the result of conditioning* $\trianglerighteq$ *by* $\mathcal{O}$*. Consider the following sets:*

- $X = \{\geq_i : \geq_i$ *is compatible with* $\trianglerighteq\}$
- $Y = \{\geq_j : \geq_j$ *is compatible with* $\trianglerighteq_\mathcal{O}\}$
- $Z = \{\geq_{i_\mathcal{O}} : \geq_{i_\mathcal{O}}$ *the result of conditioning* $\geq_i$ *by* $\mathcal{O}$ *and where* $\geq_i \in X\}$*, where orderings in* X*,* Y *and* Z *are all defined on* $\mathcal{A}$*.*

Then $Y = Z$*. The result of this lemma is illustrated in Figure 4.4.*

Proof.

1. *Let us show that $Z \subseteq Y$. Let $\geq_i \in X$ be a total preorder compatible with $\trianglerighteq$. Let us show that its conditioned preorder $\geq_{i_\mathcal{O}}$ preserves the ordering relation in $\trianglerighteq_\mathcal{O}$ (namely, $\geq_{i_\mathcal{O}}$ is compatible with $\trianglerighteq_\mathcal{O}$). The ordering between the elements of $\mathcal{O}$ is immediately preserved in $\geq_{i_\mathcal{O}}$ by the Definition 4.3 of the conditioning. In the same way, the ordering between the elements of $\mathcal{O}$ and the elements in $\mathcal{A} \setminus \mathcal{O}$ is also preserved in $\geq_{i_\mathcal{O}}$. The partial preorder $\trianglerighteq_\mathcal{O}$ is given by:*

   - *$\forall \phi \in \mathcal{O}$, $\forall \psi \in \mathcal{O}$: $\phi \trianglerighteq_\mathcal{O} \psi$ and $\psi \trianglerighteq_\mathcal{O} \phi$, this ordering is preserved in $\geq_{i_\mathcal{O}}$ by step 1 of the conditioning ...$(^1)$.*
   - *$\forall \phi \in \mathcal{O}$, $\forall \psi \in \mathcal{A} \setminus \mathcal{O}$: $\phi \rhd_\mathcal{O} \psi$, this ordering is preserved in $\geq_{i_\mathcal{O}}$ by step 2 of the conditioning ...$(^2)$.*

   It remains to show that the ordering between the elements in $\mathcal{A} \setminus \mathcal{O}$ is also preserved in $\geq_{i_\mathcal{O}} \in Z$. For these elements, the partial preorder $\trianglerighteq_\mathcal{O}$ is simply equal to $\trianglerighteq$, since $\forall \phi \in \mathcal{A} \setminus \mathcal{O}$, $\forall \psi \in \mathcal{A} \setminus \mathcal{O}$: $\phi \trianglerighteq_\mathcal{O} \psi$ if and only if $\phi \trianglerighteq \psi$.
   Since $\geq_i \in X$ is compatible with $\trianglerighteq$, the ordering is preserved by the extension such that: if $\phi \rhd \psi$ then $\phi > \psi$ and if $\phi \trianglerighteq \psi$ and $\psi \trianglerighteq \phi$ then $\phi \geq \psi$ and $\psi \geq \phi$.
   Finally, for any conditioned total preorder $\geq_{i_\mathcal{O}} \in Z$, the step 3 of the conditioning guarantees that the ordering $\geq_i$ of the elements in $\mathcal{A} \setminus \mathcal{O}$ is preserved. Therefore, $\geq_{i_\mathcal{O}} \in Z$ also preserves the ordering $\trianglerighteq$ of the elements in $\mathcal{A} \setminus \mathcal{O}$ ...$(^3)$.
   From $(^1)$, $(^2)$, $(^3)$, we conclude that any $\geq_{i_\mathcal{O}} \in Z$ is compatible with $\trianglerighteq_\mathcal{O}$.

2. *Let $\geq_j \in Y$. Let us show that $Y \subseteq Z$. Let us first focus on the sub-ordering of $\geq_j$ only defined on $\mathcal{A} \setminus \mathcal{O}$. Recall that the ordering $\geq_j$ is obtained by extending the partial preorder $\trianglerighteq_\mathcal{O}$ such that $\forall \phi \in \mathcal{A} \setminus \mathcal{O}$, $\forall \psi \in \mathcal{A} \setminus \mathcal{O}$, if $\phi \trianglerighteq_\mathcal{O} \psi$ then $\phi \geq_j \psi$ (a compatible total preorder preserves the ordering of the partial preorder by definition). $\trianglerighteq_\mathcal{O}$ is obtained by adding $\mathcal{O}$ to $\mathcal{A}_\trianglerighteq$ with the highest priority. For the assertions in $\mathcal{A} \setminus \mathcal{O}$, $\trianglerighteq_\mathcal{O}$ preserves then the ordering of $\trianglerighteq$. Therefore, it is also the case for the assertions in $\mathcal{A} \setminus \mathcal{O}$, $\geq_j$ preserves the ordering of $\trianglerighteq_\mathcal{O}$ ...$(^1)$.*

   It remains to consider the elements in $\mathcal{O}$ in the ordering $\geq_j$. The ordering in $\trianglerighteq_\mathcal{O}$ is preserved by any $\geq_j \in Y$. The assertions in $\mathcal{O}$ are equally reliable and strictly more certain than the assertions in $\mathcal{A} \setminus \mathcal{O}$ ...$(^2)$.

   From $(^1)$, the sub-ordering of $\geq_j$ only defined on $\mathcal{A} \setminus \mathcal{O}$ is compatible with $\trianglerighteq$. From $(^2)$, adding the assertions in $\mathcal{O}$ to $\mathcal{A}_{\geq_j}$ with the highest priority collapses with the conditioning of a compatible with $\trianglerighteq$ (i.e., $\geq_i \in X$) by $\mathcal{O}$ to obtain $\geq_{i_\mathcal{O}} \in Z$. We conclude that $\geq_j \in Y$ is also compatible with $\trianglerighteq_\mathcal{O}$.

□

Satisfaction of LLE, Cut and CM for partially preordered ABoxes

Now, we are ready to provide the proofs of the satisfaction of the remaining properties, namely, **LLE**, **Cut** and **CM**, which are based on the properties presented in the above subsections 4.1.3 and 4.1.3. Namely, the properties for totally preordered ABoxes, and the relationship between the set of the total preorders compatible with the initial partial preorder, and the set of the total preorders compatible with the partial preorder obtained after adding a new observation

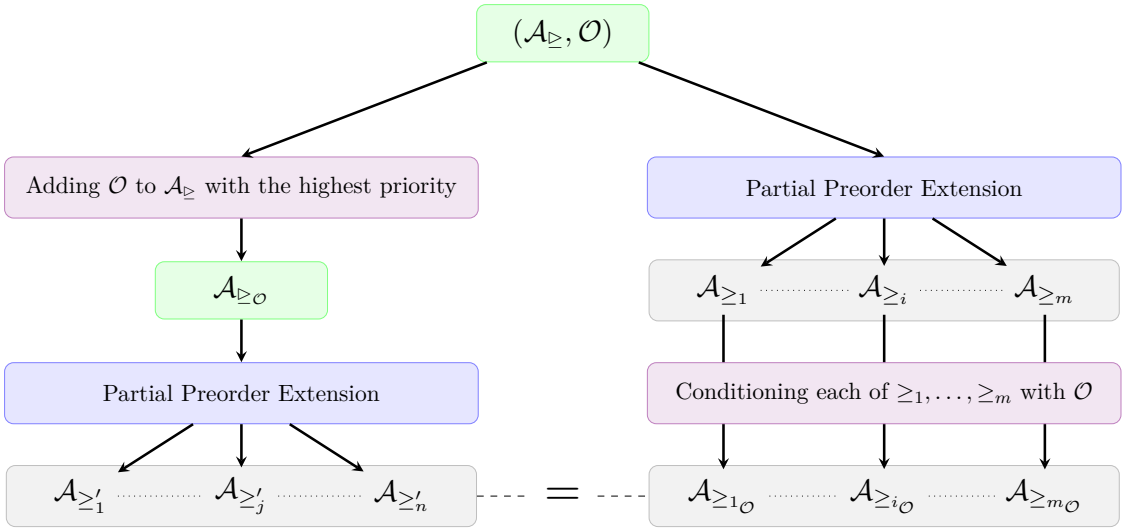


Figure 4.4: Conditioning a partial preorder $\trianglerighteq$ vs. Conditioning its total preorder extensions $\geq_1, \ldots, \geq_m$ with an observation $\mathcal{O}$

with the highest priority.

More precisely, using Lemmas 4.2 and 4.5, we can show that **LLE** (Left Logical Equivalence) is also valid in the case of partial preorders.

Lemma 4.6 (**LLE** for partially preordered KBs)**.** *Let $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$ be a partially preordered knowledge base, $\mathcal{O}$ be an observation and q be a query. The query-answering relations $\mid\!\sim^{\pi}_{\mathcal{K}_{\trianglerighteq}}$ and $\mid\!\sim^{c\pi}_{\mathcal{K}_{\trianglerighteq}}$ satisfy **LLE**.*

Proof. *Let $\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}}$ and $\mathcal{A}_{\trianglerighteq_{\mathcal{O}_2}}$ be the result of adding $\mathcal{O}_1$ and $\mathcal{O}_2$ respectively to $\mathcal{A}_{\trianglerighteq}$ with the highest priority. In order to prove the satisfaction of **LLE** for both inference relations $\mid\!\sim^{\pi}_{\mathcal{K}_{\trianglerighteq}}$ and $\mid\!\sim^{c\pi}_{\mathcal{K}_{\trianglerighteq}}$, it is enough to show that if $\langle \mathcal{T}, \mathcal{O}_1 \rangle \equiv \langle \mathcal{T}, \mathcal{O}_2 \rangle$ then $\langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}}) \rangle \equiv \langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_2}}) \rangle$ (resp. $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}}) \rangle \equiv \langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_2}}) \rangle$).*
Based on Lemma 4.5, each $\geq_{\mathcal{O}_1}$ (resp. $\geq_{\mathcal{O}_2}$) that is compatible with $\trianglerighteq_{\mathcal{O}_1}$ (resp. $\trianglerighteq_{\mathcal{O}_2}$) can be obtained equivalently by conditioning a compatible $\geq$ with $\trianglerighteq$ by $\mathcal{O}_1$ (resp. $\mathcal{O}_2$).
Based on the result of Lemma 4.2, $\forall \mathcal{A}_{\geq}$ compatible with $\mathcal{A}_{\trianglerighteq}$, if $\langle \mathcal{T}, \mathcal{O}_1 \rangle \equiv \langle \mathcal{T}, \mathcal{O}_2 \rangle$ then $\langle \mathcal{T}, \mathcal{R}_{\pi}(\mathcal{A}_{\geq_{\mathcal{O}_1}}) \rangle \equiv \langle \mathcal{T}, \mathcal{R}_{\pi}(\mathcal{A}_{\geq_{\mathcal{O}_2}}) \rangle$.
*It follows that, by intersection, if $\langle \mathcal{T}, \mathcal{O}_1 \rangle \equiv \langle \mathcal{T}, \mathcal{O}_2 \rangle$ then $\langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}}) \rangle \equiv \langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_2}}) \rangle$. Hence, **LLE** is satisfied by the partially preordered possibilistic repair.*
*In the same way, $\forall \mathcal{A}_{\geq}$ compatible with $\mathcal{A}_{\trianglerighteq}$, if $\langle \mathcal{T}, \mathcal{O}_1 \rangle \equiv \langle \mathcal{T}, \mathcal{O}_2 \rangle$ then $\langle \mathcal{T}, cl(\mathcal{R}_{\pi}(\mathcal{A}_{\geq_{\mathcal{O}_1}})) \rangle \equiv \langle \mathcal{T}, cl(\mathcal{R}_{\pi}(\mathcal{A}_{\geq_{\mathcal{O}_2}})) \rangle$. Hence, by intersection, if $\langle \mathcal{T}, \mathcal{O}_1 \rangle \equiv \langle \mathcal{T}, \mathcal{O}_2 \rangle$ then $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}}) \rangle \equiv \langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_2}}) \rangle$. Therefore, **LLE** is satisfied by the Cπ-repair.* □

Finally, the next lemma shows that the properties **CUT** and **CM** (Cautious Monotony) hold for partially preordered knowledge bases, thanks to Lemmas 4.3 and 4.4.

Lemma 4.7. *Let $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$ be a partially preordered knowledge base, $\mathcal{O}$ be an observation and q be a query. The query-answering relations $\mid\!\sim^{\pi}_{\mathcal{K}_{\trianglerighteq}}$ and $\mid\!\sim^{c\pi}_{\mathcal{K}_{\trianglerighteq}}$ satisfy **Cut** and **CM**.*

Proof.

1. ***Cut*** *states that if* $\mathcal{O}_1 \mid\!\sim^s_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}_2$ *and* $\mathcal{O}_1 \cup \mathcal{O}_2 \mid\!\sim^s_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}_3$ *then* $\mathcal{O}_1 \mid\!\sim^s_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}_3$ *(where* $s \in \{\pi, c\pi\}$*).*

   - *Proof for* $s = \pi$*, we already showed in Lemmas 4.3 and 4.4 that, for a totally preordered ABox* $\mathcal{A}_{\geq}$*, if* $\langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_{\geq})\rangle \models \mathcal{O}$ *then* $\langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_{\geq})\rangle \equiv \langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_{\geq_{\mathcal{O}}})\rangle$*.*

     This result can be generalized to the partially preordered possibilistic repair. By definition, if $\langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq})\rangle \models \mathcal{O}$ *then* $\forall \geq$ *compatible with* $\trianglerighteq$*,* $\langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_{\geq})\rangle \models \mathcal{O}$*. This means that* $\forall \mathcal{A}_{\geq}$*,* $\langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_{\geq})\rangle \equiv \langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_{\geq_{\mathcal{O}}})\rangle$*. Therefore, by intersection, we obtain* $\langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq})\rangle \equiv \langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}}})\rangle$*.*

     Let $\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}}$ *be the ABox obtained through the conditioning of* $\trianglerighteq$ *by* $\mathcal{O}_1$*. Let* $\mathcal{A}_{\trianglerighteq_{\mathcal{O}_{1,2}}}$ *be the ABox obtained through the conditioning of* $\trianglerighteq$ *by* $\mathcal{O}_1$ *and* $\mathcal{O}_2$*. Based on the previous result, one can infer that if* $\langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}})\rangle \models \mathcal{O}_2$ *then* $\langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}})\rangle \equiv \langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_{1,2}}})\rangle$*. Therefore, for a set of assertions* $\mathcal{O}_3$ *if* $\langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_{1,2}}})\rangle \models \mathcal{O}_3$*, then* $\langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}})\rangle \models \mathcal{O}_3$*.*

   - *When* $s = c\pi$*, thanks to the use of the deductive closure, showing that if* $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}})\rangle \models \mathcal{O}_2$ *then* $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}})\rangle \equiv \langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_{1,2}}})\rangle$ *can be done directly. In fact,* $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}})\rangle \models \mathcal{O}_2$ *means that for any totally preordered ABox* $\mathcal{A}_{\geq_{\mathcal{O}_1}}$ *compatible with* $\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}}$*:* $\langle \mathcal{T}, cl(\mathcal{R}_\pi(\mathcal{A}_{\geq_{\mathcal{O}_1}}))\rangle \models \mathcal{O}_2$*. Which means that* $\mathcal{O}_2 \subseteq cl(\mathcal{R}_\pi(\mathcal{A}_{\geq_{\mathcal{O}_1}}))$*, hence,* $\mathcal{O}_2 \subseteq c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}})$*. Therefore,* $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}})\rangle \equiv \langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_{1,2}}})\rangle$ *is verified. We conclude that if* $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_{1,2}}})\rangle \models \mathcal{O}_3$ *then* $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}})\rangle \models \mathcal{O}_3$*.*

2. ***CM*** *(Cautious Monotony) states that* $\mathcal{O}_1 \mid\!\sim^s_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}_2$ *and* $\mathcal{O}_1 \mid\!\sim^s_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}_3$ *then* $\mathcal{O}_1 \cup \mathcal{O}_2 \mid\!\sim^s_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}_3$ *(where* $s \in \{\pi, c\pi\}$*). Recall that* $\mathcal{O}_1 \mid\!\sim^s_{\mathcal{K}_{\trianglerighteq}} \mathcal{O}_2$ *amounts to applying the standard entailment* $\langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}})\rangle \models \mathcal{O}_2$ *(resp.* $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}})\rangle \models \mathcal{O}_2$*).*

   - *For* $s = \pi$*, we start from the result of Lemmas 4.3 and 4.4 for totally preordered knowledge bases. If* $\langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_{\geq})\rangle \models \mathcal{O}$ *then* $\mathcal{O} \cup \mathcal{R}_\pi(\mathcal{A}_{\geq}) = \mathcal{R}_\pi(\mathcal{A}_{\geq_{\mathcal{O}}})$*.*

     This result can be generalized to the partially preordered possibilistic repair. Let $\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}}$ *be the ABox obtained through the conditioning of* $\trianglerighteq$ *by* $\mathcal{O}_1$*. Let* $\mathcal{A}_{\trianglerighteq_{\mathcal{O}_{1,2}}}$ *be the ABox obtained through the conditioning of* $\trianglerighteq$ *by* $\mathcal{O}_1$ *and* $\mathcal{O}_2$*. By definition, if* $\langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}})\rangle \models \mathcal{O}_2$ *then* $\forall \geq_{\mathcal{O}_1}$ *compatible with* $\trianglerighteq_{\mathcal{O}_1}$*,* $\langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_{\geq_{\mathcal{O}_1}})\rangle \models \mathcal{O}_2$*. This means that* $\forall \mathcal{A}_{\geq_{\mathcal{O}_1}}$*,* $\mathcal{O}_2 \cup \mathcal{R}_\pi(\mathcal{A}_{\geq_{\mathcal{O}_1}}) = \mathcal{R}_\pi(\mathcal{A}_{\geq_{\mathcal{O}_{1,2}}})$ *(by Lemmas 4.3 and 4.4). Thus, by intersection, we obtain* $\mathcal{O}_2 \cup \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}}) = \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_{1,2}}})$*. Therefore,* $\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}}) \subseteq \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_{1,2}}})$*. One can infer directly that if* $\langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}})\rangle \models \mathcal{O}_3$*, then* $\langle \mathcal{T}, \pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_{1,2}}})\rangle \models \mathcal{O}_3$*.*

   - *For* $s = c\pi$*, the proof that if* $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}})\rangle \models \mathcal{O}_2$ *then* $c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}}) \subseteq c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_{1,2}}})$ *is immediate thanks to the use of the deductive closure.*

     The entailment $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}})\rangle \models \mathcal{O}_2$ *means that* $\forall \geq_{\mathcal{O}_1}$ *compatible with* $\trianglerighteq_{\mathcal{O}_1}$*,* $\langle \mathcal{T}, \mathcal{R}_\pi(\mathcal{A}_{\geq_{\mathcal{O}_1}})\rangle \models \mathcal{O}_2$*. From Lemma 4.4,* $\mathcal{O}_2 \cup \mathcal{R}_\pi(\mathcal{A}_{\geq_{\mathcal{O}_1}}) = \mathcal{R}_\pi(\mathcal{A}_{\geq_{\mathcal{O}_{1,2}}})$*. This means that for any* $\mathcal{A}_{\geq_{\mathcal{O}_1}}$*,* $\mathcal{R}_\pi(\mathcal{A}_{\geq_{\mathcal{O}_1}}) \subseteq \mathcal{R}_\pi(\mathcal{A}_{\geq_{\mathcal{O}_{1,2}}})$*. By introducing the deductive closure:* $cl(\mathcal{R}_\pi(\mathcal{A}_{\geq_{\mathcal{O}_1}})) \subseteq cl(\mathcal{R}_\pi(\mathcal{A}_{\geq_{\mathcal{O}_{1,2}}}))$*. Thus, by intersection,* $c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}}) \subseteq c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_{1,2}}})$*.*
     Therefore, one can also infer that if $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_1}})\rangle \models \mathcal{O}_3$ *then* $\langle \mathcal{T}, c\pi(\mathcal{A}_{\trianglerighteq_{\mathcal{O}_{1,2}}})\rangle \models \mathcal{O}_3$*.*

□

4.2 On the Extension of the Cπ-repair to DL-Lite$_{\mathcal{R},\sqcap}$

In this section, we discuss the applicability of the Cπ-repair method to more expressive representation languages. The idea is to find out to what extent the same process used for repairing partially preordered DL-Lite$_{\mathcal{R}}$ knowledge bases is applicable to richer languages without major changes to the proposed semantics, except for what the new language requires. This study involves two levels; the first is representational, where the richer language's ABox must be the same as the DL-Lite$_{\mathcal{R}}$ ABox. Hence, the main changes concern the syntax of the TBox, where further constructors of the concept and role inclusions are allowed. In addition, the richer language must support a deductive closure notion of its ABoxes and ensure the existence of a consistency checking mechanism in the same style of DL-Lite$_{\mathcal{R}}$ (namely, an ABox being inconsistent with a coherent TBox).

The second level is computational. In Chapter 2, we proposed an equivalent characterisation of the Cπ-repair method, which we called the "Dominant supports" characterisation and which we have shown to be tractable in DL-Lite$_{\mathcal{R}}$. This characterisation avoids the enumeration of all the total preorders that are compatible with a partial preorder. Hence, the main challenge here is to ensure that such a characterisation is also tractable when dealing with a richer language. Indeed, the DL-Lite$_{\mathcal{R}}$ language guarantees that the size of a conflict is at most two assertions [Calvanese *et al.* 2010b], and the size of a conflict set remains polynomial in the size of an ABox. This is not the case for richer languages, where a conflict might contain more than two assertions. In such a situation, even when computing the set of conflicts is done in polynomial time, always in the ABox's size, the "Dominant supports" requires going through those conflicts. Hence, the number of conflicts must also remain polynomial in the worst case to preserve tractability.

For the purpose of showing how Cπ-repair behaves with richer languages, we target a language that is slightly more expressive than DL-Lite$_{\mathcal{R}}$ and which allows the specification of conjunctions on the left-hand side of TBox axioms, which is not permitted by DL-Lite$_{\mathcal{R}}$. The presence of conjunctions in the TBox axioms has consequences for the structure of conflicts, which are no longer only binary. The language is DL-Lite$_{\mathcal{R},\sqcap}$ which was specified in [Calvanese *et al.* 2006]. Note that both consistency checking and query answering under classical semantics (case of consistency) are FOL reducible in DL-Lite$_{\mathcal{R},\sqcap}$ [Calvanese *et al.* 2006]. In the remainder of this section we show that it is possible to define the Cπ-repair method and its "Dominant supports" characterisation for DL-Lite$_{\mathcal{R},\sqcap}$ knowledge bases. However, tractability using this characterisation is not guaranteed, due to having an arbitrary number of assertions in a conflict.

Remark. The tractability offered by DL-Lite$_{\mathcal{R}}$ for reasoning and query answering comes at the cost of expressiveness. In particular, it does not allow conjunctions on the left side of TBox axioms, but they can be represented on the right side, where $B \sqsubseteq C_1 \sqcap C_2$ is equivalent to $B \sqsubseteq C_1$ and $B \sqsubseteq C_2$. In addition, union cannot be used on the right side of axioms but can be represented on the left side, where $B_1 \sqcup B_2 \sqsubseteq C$ is equivalent to the axioms $B_1 \sqsubseteq C$ and $B_2 \sqsubseteq C$.

4.2.1 The DL-Lite$_{\mathcal{R},\sqcap}$ language

Let us briefly recall the language DL-Lite$_{\mathcal{R},\sqcap}$, which is an extension of DL-Lite$_{\mathcal{R}}$ with the addition of two complex concept constructors (highlighted in bold below). More formally, the language

DL-Lite$_{\mathcal{R},\sqcap}$ is defined by :

$$B := A \mid \exists R \mid \boldsymbol{B_1 \sqcap B_2} \qquad C := A \mid \exists R \mid \neg A \mid \neg\exists R \mid \boldsymbol{\exists R.A}$$
$$R := P \mid P^- \qquad E := R \mid \neg R$$

A and P are atomic concepts and roles, respectively. B, C, R and E are basic concepts, complex concepts, basic roles, and complex roles, respectively. The negation symbol $\neg$ denotes complement sets, and the existential quantifier $\exists$ denotes existential restriction. In addition, $B_1 \sqcap B_2$ denotes the conjunction of concepts and $\exists R.A$ denotes qualified existential quantification.

A knowledge base is a pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ where $\mathcal{T}$ is a TBox containing a finite set of inclusion axioms of the form: $B \sqsubseteq C$ and $R \sqsubseteq E$, and $\mathcal{A}$ is the ABox, which contains a finite set of assertions (or ground facts) of the form: $A(a)$ and $P(a, b)$, where a and b are individuals. Concerning semantics, the equations of the interpretation function (of DL-Lite$_{\mathcal{R}}$) given in Section 1.2 of Chapter 1 hold, in addition to the two following equations for the two newly added connectors:

$$(B_1 \sqcap B_2)^{\mathcal{I}} = B_1^{\mathcal{I}} \cap B_2^{\mathcal{I}};$$
$$(\exists R.A)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} \text{ such that } (x, y) \in R^{\mathcal{I}} \text{ and } y \in A^{\mathcal{I}}\}.$$

4.2.2 The Cπ-repair for DL-Lite$_{\mathcal{R},\sqcap}$ knowledge bases

Let us now show that the Cπ-repair method is applicable to the DL-Lite$_{\mathcal{R},\sqcap}$ language both from the semantic and characterisation perspectives. This application results from the fact that the key concepts used in the Cπ-repair and its "Dominant supports" characterisation hold the same definitions as in DL-Lite$_{\mathcal{R}}$. These notions are the deductive closure of an ABox, the conflict set, the support of an assertion, and the dominance relation.

1. The ABox is defined in the same way in both languages. As a result, the concept of partially preordered ABox and its compatible total preorders remains identical to that presented in Section 2.1.

2. In both languages, a knowledge base is consistent if it admits at least one model, and is inconsistent otherwise. As a result, the notion of conflict given in Definition 1.9 remains the same. Of course, the mechanisms for testing consistency in DL-Lite$_{\mathcal{R},\sqcap}$ and DL-Lite$_{\mathcal{R}}$ are not the same, but this does not affect the definition of conflicts. Similarly, since the notion of possibilistic repair, given in Definition 1.24, in the total order used in DL-Lite$_{\mathcal{R}}$ is based solely on consistency tests, this definition adapts directly to the language DL-Lite$_{\mathcal{R},\sqcap}$. We denote by $\mathcal{R}_\pi^\sqcap$ the result of the repair, as given by Definition 1.24, but using the language DL-Lite$_{\mathcal{R},\sqcap}$ to test the consistency of a knowledge base.

3. The notion of positive deductive closure, as given in Definition 2.5, is immediately applicable to a DL-Lite$_{\mathcal{R},\sqcap}$ knowledge base. Indeed, this definition is general and to instantiate it in the DL-Lite$_{\mathcal{R},\sqcap}$ language, it suffices to use the standard entailment relation (from sets of positive axioms) of the DL-Lite$_{\mathcal{R},\sqcap}$ language. We denote by $cl_\sqcap(\cdot)$ the result of the closure of the ABox, as given by Definition 2.5, but using the inference relation of language DL-Lite$_{\mathcal{R},\sqcap}$.

4. From these first observations, we can already see that the computation of Cπ-repair, from a semantic point of view, using repairs on all total compatible orders applies directly to the language DL-Lite$_{\mathcal{R},\sqcap}$, as follows:

$$c\pi(\mathcal{A}_\trianglerighteq) = \bigcap_{i=1}^{m} \{cl_\sqcap(\mathcal{R}_\pi^\sqcap(\mathcal{A}_{\geq_i})) \mid \mathcal{A}_{\geq_i} \text{ is compatible with } \mathcal{A}_\trianglerighteq\}$$

Where $\mathcal{A}_{\geq_i}$ denotes a totally preordered ABox that is compatible with $\mathcal{A}_{\trianglerighteq}$.

5. Finally, the definition of the notion of support, given by Definition 2.11, can be applied as is in the language DL-Lite$_{\mathcal{R},\sqcap}$, with the understanding that the standard inference relation of DL-Lite$_{\mathcal{R},\sqcap}$ is used instead of that of DL-Lite$_{\mathcal{R}}$. Similarly, the definition of dominance, as presented in Definition 2.12, applies to subsets of the ABox (whether binary or not), and therefore its application does not require any adaptation to the language DL-Lite$_{\mathcal{R},\sqcap}$. These two observations allow us to use the syntactic characterisation given in Propositions 2.2 and 2.3, which remains valid for the language DL-Lite$_{\mathcal{R},\sqcap}$. The proofs of Propositions 2.2 and 2.3 also hold in the case of a DL-Lite$_{\mathcal{R},\sqcap}$ knowledge base, since they have been established for the case of conflicts and supports of arbitrary sizes.

Let us now illustrate the application of the Cπ-repair on a DL-Lite$_{\mathcal{R},\sqcap}$ knowledge base. Consider the example of three (3) research teams working on two (2) (different) work packages each. We must respect the following conditions:

1. Each work package is led by a lab member, and;

2. A lab member cannot lead all the work packages across teams.

We employ the individuals wp_1, wp_2, wp_3, wp_4, wp_5 and wp_6 to represent the work packages and the roles Leads_1, Leads_2, and Leads_3 to designate the leader of a work package in each team. We use the TBox axiom $\exists\mathsf{Leads}_1 \sqcap \exists\mathsf{Leads}_2 \sqsubseteq \neg\,\exists\mathsf{Leads}_3$ to express the second condition "a lab member cannot lead all work packages across teams", more precisely, the existential quantification over the three roles Leads_1, Leads_2 and Leads_3 must be disjoint.

Example 4.2. *Let $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$ be a DL-Lite$_{\mathcal{R},\sqcap}$ knowledge base given by the TBox:*

$$\mathcal{T} = \{\exists\mathsf{Leads}_1 \sqcap \exists\mathsf{Leads}_2 \sqsubseteq \neg\,\exists\mathsf{Leads}_3\}$$

and the ABox $\mathcal{A}_{\trianglerighteq}$, such that:

$$\mathcal{A}_{\trianglerighteq} = \boxed{\begin{array}{c} \mathsf{Leads}_1(\mathsf{Mary}, \mathsf{wp}_1), \\ \mathsf{Leads}_1(\mathsf{Mary}, \mathsf{wp}_2), \\ \mathsf{Leads}_2(\mathsf{Mary}, \mathsf{wp}_3), \\ \mathsf{Leads}_2(\mathsf{Mary}, \mathsf{wp}_4), \\ \mathsf{Leads}_3(\mathsf{Mary}, \mathsf{wp}_5), \\ \mathsf{Leads}_3(\mathsf{Mary}, \mathsf{wp}_6) \end{array}}$$

Figure 4.5 illustrates the partial preorder $\trianglerighteq$ over the ABox $\mathcal{A}_{\trianglerighteq}$.
Let us now exhibit the conflict set of the ABox $\mathcal{A}_{\trianglerighteq}$, using the axiom $\exists\mathsf{Leads}_1 \sqcap \exists\mathsf{Leads}_2 \sqsubseteq \neg\,\exists\mathsf{Leads}_3$. We can easily check that the conflicts are of the form:

$$\mathcal{C} = \{\mathsf{Leads}_1(\mathsf{Mary}, \mathsf{wp}_i), \mathsf{Leads}_2(\mathsf{Mary}, \mathsf{wp}_j), \mathsf{Leads}_3(\mathsf{Mary}, \mathsf{wp}_k)\}$$

Where $i \in \{1,2\}$, $j \in \{3,4\}$ and $k \in \{5,6\}$. Namely, there are exactly (8) conflicts:

$$\mathsf{Cf}(\mathcal{A}_{\trianglerighteq}) = \left\{ \begin{array}{l} \{\mathsf{Leads}_1(\mathsf{Mary},\mathsf{wp}_1), \mathsf{Leads}_2(\mathsf{Mary},\mathsf{wp}_3), \mathsf{Leads}_3(\mathsf{Mary},\mathsf{wp}_5)\}, \\ \{\mathsf{Leads}_1(\mathsf{Mary},\mathsf{wp}_1), \mathsf{Leads}_2(\mathsf{Mary},\mathsf{wp}_3), \mathsf{Leads}_3(\mathsf{Mary},\mathsf{wp}_6)\}, \\ \{\mathsf{Leads}_1(\mathsf{Mary},\mathsf{wp}_1), \mathsf{Leads}_2(\mathsf{Mary},\mathsf{wp}_4), \mathsf{Leads}_3(\mathsf{Mary},\mathsf{wp}_5)\}, \\ \{\mathsf{Leads}_1(\mathsf{Mary},\mathsf{wp}_1), \mathsf{Leads}_2(\mathsf{Mary},\mathsf{wp}_4), \mathsf{Leads}_3(\mathsf{Mary},\mathsf{wp}_5)\}, \\ \{\mathsf{Leads}_1(\mathsf{Mary},\mathsf{wp}_2), \mathsf{Leads}_2(\mathsf{Mary},\mathsf{wp}_3), \mathsf{Leads}_3(\mathsf{Mary},\mathsf{wp}_5)\}, \\ \{\mathsf{Leads}_1(\mathsf{Mary},\mathsf{wp}_2), \mathsf{Leads}_2(\mathsf{Mary},\mathsf{wp}_3), \mathsf{Leads}_3(\mathsf{Mary},\mathsf{wp}_6)\}, \\ \{\mathsf{Leads}_1(\mathsf{Mary},\mathsf{wp}_2), \mathsf{Leads}_2(\mathsf{Mary},\mathsf{wp}_4), \mathsf{Leads}_3(\mathsf{Mary},\mathsf{wp}_5)\}, \\ \{\mathsf{Leads}_1(\mathsf{Mary},\mathsf{wp}_2), \mathsf{Leads}_2(\mathsf{Mary},\mathsf{wp}_4), \mathsf{Leads}_3(\mathsf{Mary},\mathsf{wp}_6)\} \end{array} \right\}$$

Indeed, the assertions $\mathsf{Leads}_1(\mathsf{Mary},\mathsf{wp}_1)$, $\mathsf{Leads}_2(\mathsf{Mary},\mathsf{wp}_3)$ and $\mathsf{Leads}_3(\mathsf{Mary},\mathsf{wp}_5)$ contradict the axiom $\exists\mathsf{Leads}_1 \sqcap \exists\mathsf{Leads}_2 \sqsubseteq \neg\, \exists\mathsf{Leads}_3$ and form the first conflict, the same applies for the other conflicts.
Let us now compute the Cπ-repair of $\mathcal{A}_{\trianglerighteq}$:

- *The assertions $\mathsf{Leads}_1(\mathsf{Mary},\mathsf{wp}_1)$ and $\mathsf{Leads}_1(\mathsf{Mary},\mathsf{wp}_2)$ are both strictly preferred to all the remaining assertions, and thus, each one of them forms a support for itself, and dominates all the conflicts.*
- *The assertion $\mathsf{Leads}_2(\mathsf{Mary},\mathsf{wp}_3)$ does not belong to $c\pi(\mathcal{A}_{\trianglerighteq})$, because it is the only support for itself, and it does not dominate the conflict $\{\mathsf{Leads}_1(\mathsf{Mary},\mathsf{wp}_2), \mathsf{Leads}_2(\mathsf{Mary},\mathsf{wp}_4), \mathsf{Leads}_3(\mathsf{Mary},\mathsf{wp}_6)\}$, namely, not strictly preferred to an assertion from the conflict. The remaining assertions do not belong to $c\pi(\mathcal{A}_{\trianglerighteq})$ and can be checked in the same way.*

Therefore, the Cπ-repair of $\mathcal{A}_{\trianglerighteq}$ is given by:

$$c\pi(\mathcal{A}_{\trianglerighteq}) = \{\mathsf{Leads}_1(\mathsf{Mary},\mathsf{wp}_1),\ \mathsf{Leads}_1(\mathsf{Mary},\mathsf{wp}_2)\}$$

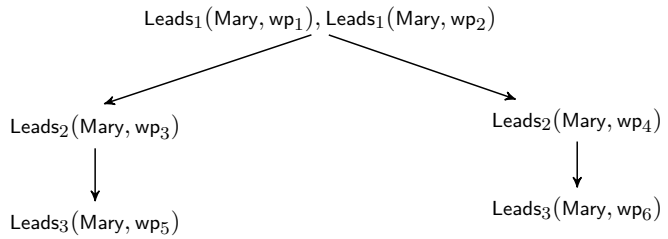


Figure 4.5: The partially preordered ABox $\mathcal{A}_{\trianglerighteq}$, $\longrightarrow$: the strict preference $\rhd$.

4.2.3 The result of applying the "Dominant supports" characterisation on a DL-Lite$_{\mathcal{R},\sqcap}$ knowledge base

The main difference between the application of the "Dominant supports" characterisation on a DL-Lite$_{\mathcal{R}}$ knowledge base and a DL-Lite$_{\mathcal{R},\sqcap}$ knowledge base is the size of the conflicts and supports. In particular, each conflict in a DL-Lite$_{\mathcal{R}}$ knowledge base has a size of at most

2. However, when working in the DL-Lite$_{\mathcal{R},\sqcap}$ language, a conflict can be of any size. In the previous example, for instance, the conflicts contained three (3) assertions.

This observation impacts the size of the set of all conflicts, which can be exponential with respect to the size of the ABox. In contrast, in DL-Lite$_{\mathcal{R}}$, this size is always polynomial with respect to the size of the ABox. This statement is illustrated in the following example.

Example 4.2 (Continued)**.** *In the knowledge base $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$, we limited ourselves to 3 research teams and 2 work packages by team. This also limits the number of concepts involved in the disjointness axiom used in the TBox. Assume now an arbitrary number of teams, then, the disjointness axiom becomes $\exists \mathsf{Leads}_1 \sqcap \exists \mathsf{Leads}_2 \sqcap \ldots \sqcap \exists \mathsf{Leads}_{k-1} \sqsubseteq \neg \exists \mathsf{Leads}_k$, where k represents the number of research teams (number of concepts ($\exists \mathsf{Leads}_i$) involved in the disjointness axiom). In this case, with an arbitrary number (p) of work packages per team, the size of the conflict set can be given by:*

$$|\mathsf{Cf}(\mathcal{A}_{\trianglerighteq})| = O(p^k)$$

This is because conflicts are obtained by combining any k-tuple of assertions, each from a different role Leads_i.

The above example shows how the size of the conflicts can grow exponentially with respect to the two factors used to construct the ABox, namely k and p. Thus, the application of the "Dominant supports" characterisation can be impractical when dealing with conflicts of an arbitrary size.

4.3 A Lexicographic-Based Semantics for the Cπ-repair

In the last section of this chapter, we study the application of a different inconsistency-tolerant semantics in the process of repairing a partially preordered DL-Lite$_{\mathcal{R}}$ knowledge base. The Cπ-repair is obtained via the intersection of the closed possibilistic repairs of the totally preordered ABoxes that are compatible with the initial partially preordered ABox. Before studying the replacement of the possibilistic repair with a different semantics, we first distinguish the semantics that are not directly based on repairs, such as paraconsistent multivalued description logics. These approaches do not align with the spirit of the methods discussed in this thesis. Therefore, their adaptation is not immediate for reasoning with partially ordered ABoxes. In the context of multivalued semantics, the ABox remains unchanged, but the query-answering mechanisms need to be adapted, which can generate additional computational costs. A necessary first step would be to start with an existing multivalued semantics for standard DL-Lite$_{\mathcal{R}}$ (for example, the 4-valued semantics provided in [Zhou *et al.* 2012]), and then extend it to the possibilistic framework with a totally ordered ABox. This requires an adaptation of existing work (for a flat ABox) to define preferred 4-valued canonical models. The final step involves taking all the extensions of the total orders and defining the 4-valued canonical models of the partial ABox as the union of the preferred 4-valued canonical models of each total ABox extension.

For inconsistency-tolerant semantics based on repairs, the extension is also feasible from a semantic point of view. However, it should be noted that the strategy we used in our approach produces a single repair for each total order. In the general case, using a different strategy, each total order may return multiple repairs. We aim to adapt our method with the inconsistency-tolerant semantics known as lexicographic or cardinality-based semantics [Benferhat *et al.* 2015], which was recalled in Definition 1.20. It is important to note that whether

for semantics based on repairs or not, the challenge is to have an equivalent characterisation (without generating all the extensions of the partial order) as provided for the Cπ-repair using Propositions 2.2 and 2.3.

The lexicographic preferred-based repair is a cardinality-based repair that allows selecting the larger preferred repairs of a totally preordered ABox. An important consideration to make when applying this repair in the Cπ-repair process is that unlike the possibilistic repair, it might return more than one preferred repair. Therefore, we propose computing the intersection of the lexicographic preferred-based repairs of a totally preordered compatible ABox, before applying the deductive closure step. We start by recalling Definition 1.20 of a lexicographic-based repair.

Definition 4.4 (Lexicographic preferred-based repair (recalled))**.** *Let $\mathcal{K}_{\geq} = \langle \mathcal{T}, \mathcal{A}_{\geq} \rangle$ be a totally preordered DL-Lite$_{\mathcal{R}}$ knowledge base where the ABox is given by its well-ordered partitions $(\mathcal{S}_1, \ldots, \mathcal{S}_n)$.*
Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be two consistent subsets of $(\mathcal{S}_1, \ldots, \mathcal{S}_n)$.

- *$\mathcal{R}_1$ is as preferred as $\mathcal{R}_2$ if $|\mathcal{R}_1 \cap \mathcal{S}_i| = |\mathcal{R}_2 \cap \mathcal{S}_i|$, for $i = 1, \ldots, n$.*
- *$\mathcal{R}_1$ is strictly preferred to $\mathcal{R}_2$ if $|\mathcal{R}_1 \cap \mathcal{S}_i| > |\mathcal{R}_2 \cap \mathcal{S}_i|$ for some $i \in \{1, \ldots, n\}$ and $|\mathcal{R}_1 \cap \mathcal{S}_j| = |\mathcal{R}_2 \cap \mathcal{S}_j|$, $\forall j$ such that $1 \leq j < i$.*

Where $|X|$ is the cardinality of the set X.

Then $\mathcal{R} \subseteq \mathcal{A}_{\geq}$ is a lexicographical preferred-based repair of $\mathcal{A}_{\geq}$ if there is no $\mathcal{R}' \subseteq \mathcal{A}_{\geq}$ such that $\mathcal{R}'$ is strictly preferred to $\mathcal{R}$. The set of all the lexicographic preferred repairs of $\mathcal{A}_{\geq}$ is denoted $\mathsf{PAR}_{lex}(\mathcal{A}_{\geq})$.

In the following, we denote the intersection of all the lexicographic preferred-based repairs of $\mathcal{A}_{\geq}$ by:

$$\mathsf{IPR}_{lex}(\mathcal{A}_{\geq}) = \bigcap \{\mathcal{R} | \mathcal{R} \in \mathsf{PAR}_{lex}(\mathcal{A}_{\geq})\}$$

The lexicographical repair can be extended to handle partially preordered ABoxes by deductively closing all the sets obtained from the intersection of the lexicographic preferred-based repairs of the totally preordered ABoxes compatible with the initial partially preordered ABox. Formally, for a given partially preordered knowledge base $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$:

$$\mathsf{ICLR}(\mathcal{A}_{\trianglerighteq}) = \bigcap_{i=1}^{m} \{cl(\mathsf{IPR}_{lex}(\mathcal{A}_{\geq_i})) \mid \mathcal{A}_{\geq_i} \text{ is compatible with } \mathcal{A}_{\trianglerighteq}\}$$

Where $\mathsf{ICLR}(\mathcal{A}_{\trianglerighteq})$ denotes the Intersection of Closed Lexicographic Preferred-Based Repairs of a partially preordered ABox.

The process of applying the lexicographic preferred-based repair to a partially preordered knowledge base $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$ is depicted by Figure 4.6, where each step is given as follows:

Step 1: First, extend the partially preordered ABox $\mathcal{A}_{\trianglerighteq}$ into a family of totally preordered ABoxes, denoted $A_{\geq_1}, \ldots, A_{\geq_m}$.
Step 2: Then, compute the set of lexicographical preferred-based repairs $\mathsf{PAR}_{lex}(\mathcal{A}_{\geq_i})$ for each obtained totally preordered ABox $\mathcal{A}_{\geq_i}$.
Step 3: Compute the intersection of lexicographical preferred-based repairs $\mathsf{IPR}_{lex}(\mathcal{A}_{\geq_i})$ of each obtained set of lexicographical preferred-based repairs $\mathsf{PAR}_{lex}(\mathcal{A}_{\geq_i})$.
Step 4: Compute the deductive closure, as per Definition 2.5, of each intersection of lexicographical preferred-based repairs $cl(\mathsf{IPR}_{lex}(\mathcal{A}_{\geq_i}))$.
Step 5: Intersect all closed repairs to obtain a single closure-based repair $\mathsf{ICLR}(\mathcal{A}_{\trianglerighteq})$ (the Intersection of Closed Lexicographic Preferred-Based Repairs of a partially preordered ABox).

Let us use our running example from Chapter 2 to illustrate the application of lexicographic preferred-based repair on partially preordered DL-Lite$_{\mathcal{R}}$ knowledge bases.

Example 4.3. *We start by recalling the knowledge base $\mathcal{K}_{\trianglerighteq} = \langle \mathcal{T}, \mathcal{A}_{\trianglerighteq} \rangle$ from Example 2.1. The TBox $\mathcal{T}$ is given by the following axioms:*

$$\mathcal{T} = \left\{ \begin{array}{ll} 1.\ \mathsf{Secondee} \sqsubseteq \mathsf{User} & 7.\ \exists \mathsf{hasAccess}^{-} \sqsubseteq \mathsf{Resource} \\ 2.\ \mathsf{FullResearcher} \sqsubseteq \mathsf{User} & 8.\ \exists \mathsf{hasConfidAccess}^{-} \sqsubseteq \mathsf{Confidential} \\ 3.\ \mathsf{Confidential} \sqsubseteq \mathsf{Resource} & 9.\ \mathsf{hasConfidAccess} \sqsubseteq \neg\, \mathsf{hasOpenAccess} \\ 4.\ \mathsf{Secondee} \sqsubseteq \neg\, \mathsf{FullResearcher} & 10.\ \mathsf{owns} \sqsubseteq \mathsf{hasAccess} \\ 5.\ \exists \mathsf{hasAccess} \sqsubseteq \mathsf{User} & 11.\ \mathsf{hasOpenAccess} \sqsubseteq \mathsf{hasAccess} \\ 6.\ \exists \mathsf{owns} \sqsubseteq \mathsf{FullResearcher} & 12.\ \mathsf{hasConfidAccess} \sqsubseteq \mathsf{hasAccess} \end{array} \right\}$$

The assertions in our ABox are the following:

$$\mathcal{A}_{\trianglerighteq} = \left\{ \begin{array}{l} \mathsf{FullResearcher}(\mathsf{Ben}) \\ \mathsf{Secondee}(\mathsf{Duane}) \\ \mathsf{Confidential}(\mathsf{r}_1) \\ \mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1) \\ \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1) \\ \mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2) \end{array} \right\}$$

Then, we consider the following partial preorder relation $\trianglerighteq$ over the assertions of our ABox:

- *The assertion* $\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)$ *is more certain than any other assertion in* $\mathcal{A}$.
- $\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)$ *is more certain than* $\mathsf{Secondee}(\mathsf{Duane})$.
- *The assertions* $\mathsf{FullResearcher}(\mathsf{Ben})$ *and* $\mathsf{Confidential}(\mathsf{r}_1)$ *are equally certain, and more certain than* $\mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)$.

The conflict set of $\mathcal{A}_{\trianglerighteq}$ is given by $\mathsf{Cf}(\mathcal{A}_{\trianglerighteq}) = \{\mathcal{C}_1, \mathcal{C}_2\}$ where:

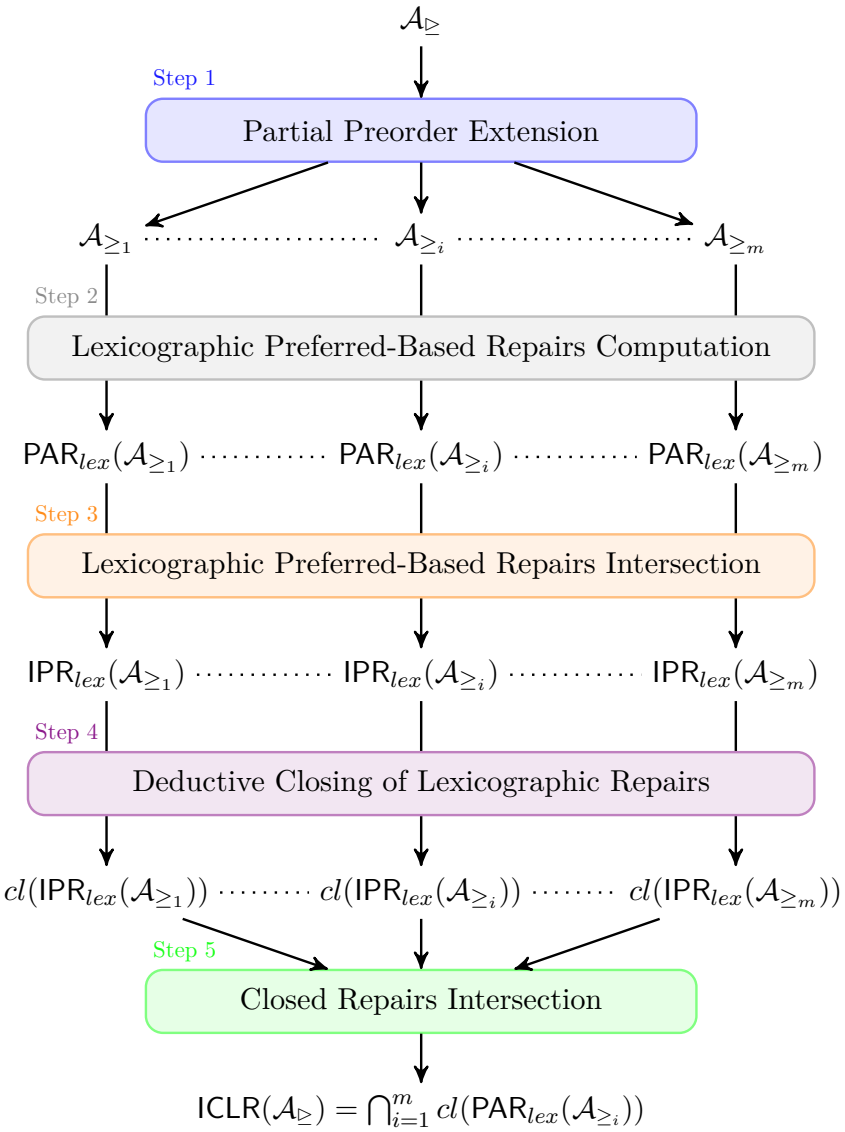
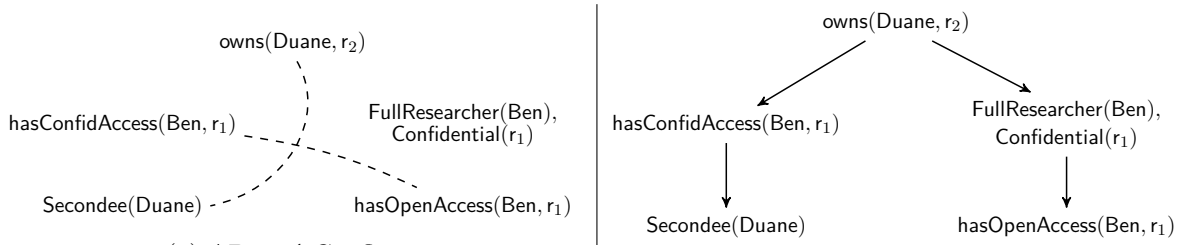


Figure 4.6: The process of computing $\mathsf{ICLR}(\mathcal{A}_{\trianglerighteq})$



(a) ABox $\mathcal{A}$ Conflicts

(b) Partially Preordered ABox $\mathcal{A}_{\trianglerighteq}$

Figure 4.7: The Partially Preordered ABox $\mathcal{A}_{\trianglerighteq}$

- $\mathcal{C}_1 = \{\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{Secondee}(\mathsf{Duane})\}$*; and,*
- $\mathcal{C}_2 = \{\mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)\}$.

Figure 4.7 depicts the conflict set $\mathsf{Cf}(\mathcal{A}_{\trianglerighteq})$ *and the partial preorder* $\trianglerighteq$.
We also recall some of the total preorder extensions of the partial preorder which are depicted in Figure 4.8.
The lexicographic preferred-based repairs of each depicted totally preordered extension of the ABox $\mathcal{A}_{\trianglerighteq}$ *are given as follows:*

$$\begin{aligned}
\mathsf{PAR}_{lex}(\mathcal{A}_{\trianglerighteq_1}) &= \{\{\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{Confidential}(\mathsf{r}_1), \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)\}\} \\
\mathsf{PAR}_{lex}(\mathcal{A}_{\trianglerighteq_2}) &= \{\{\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{Confidential}(\mathsf{r}_1), \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1)\}, \\
&\quad \{\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{Confidential}(\mathsf{r}_1), \mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1)\}\} \\
\mathsf{PAR}_{lex}(\mathcal{A}_{\trianglerighteq_m}) &= \{\{\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{Confidential}(\mathsf{r}_1)\}\}
\end{aligned}$$

Note that in our example, a single totally preordered compatible ABox admits two lexicographic preferred-based repairs, which is illustrated here by $\mathcal{A}_{\trianglerighteq_2}$.
The intersection of the two obtained repairs gives the set $\mathsf{IPR}_{lex}(\mathcal{A}_{\trianglerighteq_2}) = \{\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{Confidential}(\mathsf{r}_1)\}$.
Now, applying the deductive closure on each resulting repair results in the following sets:

$$\begin{aligned}
cl(\mathsf{IPR}_{lex}(\mathcal{A}_{\trianglerighteq_1})) = \{&\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Duane}), \mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{User}(\mathsf{Duane}), \\
&\mathsf{Resource}(\mathsf{r}_2), \mathsf{hasOpenAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1), \\
&\mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{Confidential}(\mathsf{r}_1), \mathsf{User}(\mathsf{Ben}), \mathsf{Resource}(\mathsf{r}_1)\} \\
cl(\mathsf{IPR}_{lex}(\mathcal{A}_{\trianglerighteq_2})) = \{&\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Duane}), \mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{User}(\mathsf{Duane}), \\
&\mathsf{Resource}(\mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{Confidential}(\mathsf{r}_1), \mathsf{User}(\mathsf{Ben}), \mathsf{Resource}(\mathsf{r}_1)\} \\
cl(\mathsf{IPR}_{lex}(\mathcal{A}_{\trianglerighteq_m})) = \{&\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Duane}), \mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{User}(\mathsf{Duane}), \\
&\mathsf{Resource}(\mathsf{r}_2), \mathsf{hasConfidAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{hasAccess}(\mathsf{Ben}, \mathsf{r}_1), \mathsf{FullResearcher}(\mathsf{Ben}), \\
&\mathsf{Confidential}(\mathsf{r}_1), \mathsf{User}(\mathsf{Ben}), \mathsf{Resource}(\mathsf{r}_1)\}
\end{aligned}$$

The intersection of the above closed repairs returns the following consistent subset of the ABox:

$$\mathsf{ICLR}(\mathcal{A}_{\trianglerighteq}) = \left\{ \begin{matrix} \mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Duane}), \mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{User}(\mathsf{Duane}), \\ \mathsf{Resource}(\mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Ben}), \mathsf{Confidential}(\mathsf{r}_1), \mathsf{User}(\mathsf{Ben}), \mathsf{Resource}(\mathsf{r}_1) \end{matrix} \right\}$$

The application of the π-repair given by Definition 2.2 and the Cπ-repair given by Definition 2.10 on this example's knowledge base gives the following sets:

$$\pi(\mathcal{A}_{\trianglerighteq}) = \{\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2)\}$$
$$c\pi(\mathcal{A}_{\trianglerighteq}) = \{\mathsf{owns}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{FullResearcher}(\mathsf{Duane}), \mathsf{hasAccess}(\mathsf{Duane}, \mathsf{r}_2), \mathsf{User}(\mathsf{Duane}), \mathsf{Resource}(\mathsf{r}_2), \mathsf{Confidential}(\mathsf{r}_1), \mathsf{User}(\mathsf{Ben}), \mathsf{Resource}(\mathsf{r}_1)\}$$

In this example, we can already see that the method based on the lexicographic preferred-based repairs yields a repair that is larger than both the π-repair and the Cπ-repair by returning the assertion $\mathsf{FullResearcher}(\mathsf{Ben})$.

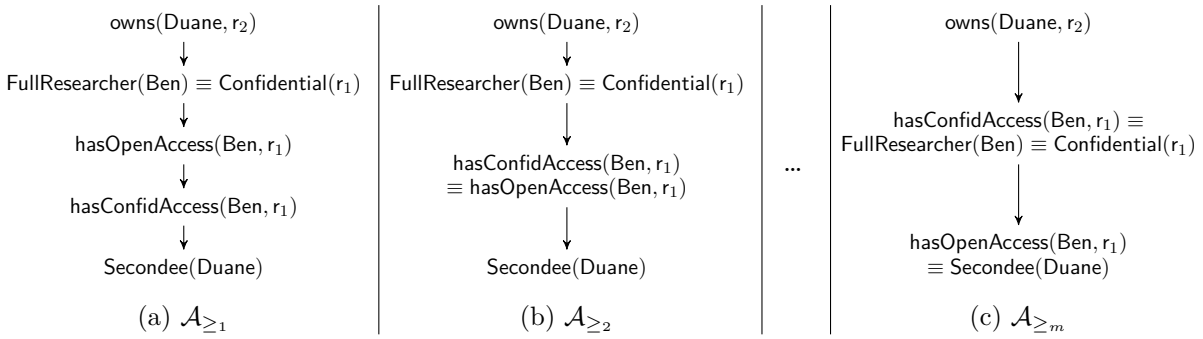


Figure 4.8: The ABoxes $\mathcal{A}_{\geq_1}, \ldots, \mathcal{A}_{\geq_m}$ compatible with $\mathcal{A}_{\trianglerighteq}$

The previous example shows a situation where using a lexicographic preferred-based repair semantics instead of the possibilistic repair in the process of repairing partially preordered knowledge bases yields a larger (or more productive) repair. Further investigation is required to evaluate the productivity of the repair process illustrated in Figure 4.6 over the one given by the Cπ-repair method. However, the main issue in the case of the lexicographic preferred-based repair semantics is computational. The tractability of the Cπ-repair method for DL-Lite$_\mathcal{R}$ knowledge bases is ensured via the "Dominant supports" characterisation, and the same characterisation does not apply in this case. Therefore, since even in the case of a total preorder relation over the ABox, the lexicographic preferred-based repair semantics is not tractable [Benferhat *et al.* 2015], finding a tractable characterisation when dealing with partially preordered ABoxes is not guaranteed.

4.4 Conclusion

This chapter completes the study of the Cπ-repair method by exploring its behaviour with respect to some well-established rationality properties and extension in two directions beyond its initial formulation: on the representation level and on the semantics level.

The analysis of rationality properties for both Cπ-repair and its unclosed variant, the π-repair, establishes that both methods satisfy unconditional properties, as well as System P rules. Although these properties may appear natural or minimal, other inconsistency-tolerant semantics do not necessarily guarantee them. For instance, the closure-based semantics CAR and ICAR do not satisfy consistency of support (**ConsS**), unless the conclusion is a ground query (i.e., a single assertion) [Baget *et al.* 2016, Bienvenu & Bourgaux 2017]. In addition, the so-called majority-based semantics (a query is valid if it is obtained from the majority of

the repairs of an inconsistent ABox) does not satisfy these minimal properties. More precisely, in [Baget *et al.* 2016], majority-based inference has been shown to not satisfy the Cut, CM, and AND properties. Another example in which System P is not satisfied is the Brave semantics, where a query is valid if it follows from one repair. Regarding the rational monotony (RM) property, we observe that its non-satisfaction is consistent with results from standard possibilistic propositional logic and non-monotonic relations. In particular, a representation theorem from KLM [Kraus *et al.* 1990] shows that any non-monotonic inference relation satisfying System P properties in the same time with rational monotony is necessarily representable by a total order on the set of interpretations of propositional logic.

Then, we illustrate how the Cπ-repair can be extended to richer representation languages. In particular, we adapt the Cπ-repair process and the dominant supports characterisation to DL-Lite$_{\mathcal{R},\sqcap}$, which extends DL-Lite$_{\mathcal{R}}$ with the ability to represent conjunction in TBox axioms. While the representation level adapts naturally, computational challenges arise: the tractability of the dominant supports method relies on conflict sets being polynomial in the size of the ABox, which is not always guaranteed for richer languages such as DL-Lite$_{\mathcal{R},\sqcap}$.

Finally, we examine the applicability of Cπ-repair to alternative semantics. We showed that the repair process can be defined under lexicographic preferred-based semantics [Benferhat *et al.* 2015]. However, the characterisations established for the possibilistic setting no longer hold, and tractability is not preserved. Even when dealing with a total preorder on the ABox, lexicographic preferred-based repairs remain intractable, and hence tractability is not guaranteed, at least using the same characterisation, for the partially preordered case.

Chapter 5

Conflict-Resolution and Explanation of Access Control Models

Contents

5.1 Overview of Access Control 139
5.1.1 A Possible Classification of Access Control Models 139
5.1.2 Organisation Based Access Control (OrBAC) 141
5.1.3 Inconsistency in the OrBAC Model 145
5.2 A Repair-Based Semantics for OrBAC 146
5.2.1 A Possibilistic OrBAC Model 147
5.2.2 The OrBAC Ontology 148
5.2.3 Conflict-Resolution in OrBAC 154
5.3 Explaining Access Control Decisions 156
5.3.1 Logical Explanations of OrBAC Policies 156
5.3.2 On Explaining OrBAC Policies using LLMs 158
5.4 Conclusion 166

This chapter presents an application of some of our methods for inconsistency management in the context of access control models. A particular focus is given to explaining the decisions of access control models, both in the presence and in the absence of inconsistency. In the context of access control, there are several mechanisms to ensure that only authorised users can (properly) access the information systems. Thus, computer systems require, via a security policy, the restriction of access to some resources depending on multiple factors.

In this case, a security policy is a formal set of rules and guidelines that defines how an organisation manages, protects, and controls access to its information systems, resources, and assets to ensure confidentiality, integrity, and availability. Access control models are the tool for specifying and enforcing a security policy [Hu *et al.* 2014, Penelova 2021, Ameer *et al.* 2022, Bertolissi & Fernández 2022]. In general, an access control model focuses on making the decision of granting or denying a *subject* (user) the right to perform an *action* on some *object* (resource). This is the case of the Harrison, Ruzzo, Ullman model (HRU) [Harrison *et al.* 1976]. Moreover, for a model to be more general, scalable, and expressive, one may resort to more compact specifications by grouping users into roles, such as in Role Based Access

Control (RBAC) [Ameer *et al.* 2022], specifying contexts of access, such as in Organisation-Based Access Control (OrBAC) [Kalam *et al.* 2003], or relying on user and object attributes to make a decision, such as in Attribute-Based Access Control (ABAC) [Hu *et al.* 2014].

Most traditional access control models, such as HRU, support only permission rules, implicitly assuming a closed-world perspective where the absence of permission implies denial. In contrast, more expressive frameworks like ABAC and OrBAC introduce both permission and prohibition rules, enabling richer specifications and scalable administration of complex systems in which access decisions are inferred rather than explicitly enumerated. However, this expressiveness comes at the cost of inconsistency: When both types of rule apply to the same access request, conflicting conclusions may arise. To resolve such contradictions, the literature proposes a variety of strategies, ranging from administrator-driven resolution [Bringhenti *et al.* 2023] and collaborative decision aggregation via threshold-based models [Anton *et al.* 2024] to argumentation-based approaches that treat policies as competing arguments [Boella *et al.* 2005, Perrussel *et al.* 2007]. Within OrBAC, early solutions relied on logical encodings with possibilistic or lexicographic inference [Benferhat *et al.* 2003, Benferhat & El Baida 2004], although these suffered from limited formal grounding or high computational cost, while alternative proposals imposed ordering relations or separation constraints to avoid rule clashes [Cuppens *et al.* 2007] at the expense of policy flexibility.

Furthermore, the literature on access control largely involved policy specification, enforcement architectures, conflict resolution, and even automated policy learning, focusing on what decision to return rather than *explaining* how and why that decision was made. This gap may be due to the sensitive nature of the matter in question. For instance, it seems unnecessary to explain to a user why they were refused access to some resources. In addition, it seems unnecessary to explain the decision to a user who was granted the permission. Yet, an explanation mechanism would simplify the tasks of debugging and analysing security problems, whether they are caused by shortcomings of the representation framework or by the policy itself.

While access control systems traditionally operate under the assumption that "decisions speak for themselves", this stance becomes increasingly untenable in complex or collaborative environments. Modern systems involve layered delegation, context-aware rules, conflict resolution strategies, and sometimes probabilistic or priority-based reasoning. In such settings, returning a bare `Permit` or `Deny` decision provides limited value, especially when administrators must audit policies, justify outcomes to stakeholders, or ensure compliance with organisational or legal standards. Even within interpretable policy models like OrBAC, understanding which rules were applied, how conflicts were resolved, or why certain permissions were derived can be far from obvious when hundreds of rules interact. This is particularly true in the presence of exception handling or inheritance mechanisms, where reasoning is not strictly local but propagates through multiple layers of abstraction.

The growing demand for transparency, accountability, and trust in AI-driven and automated decision systems strengthens the case for explainability in access control. In eXplainable AI (XAI), distinctions are made between global explanations, which describe how a system behaves overall, and local explanations, which justify individual outcomes [Arya *et al.* 2019]. Access control requires both: administrators need global insight to verify policy soundness, while users or auditors may require local explanations to understand why a specific request was accepted or rejected. Moreover, explanation styles vary: trace-based or rule-based justifications may suffice for administrators, while more abstract, contrastive ("Why allowed rather than denied?") or counterfactual ("Would access be granted if role X were added?") explanations [Chari *et al.* 2020] may be more appropriate for higher-level stakeholders. Integrating such explanation mechanisms into access control is therefore not merely a usability refinement, but a necessary

step toward robust, auditable, and trustworthy security policy management.

In this chapter, we first survey existing access control models before focusing on the OrBAC model. We then address conflict resolution and explanation in OrBAC by encoding it as an ontology. The conflict resolution approach is inspired by the inconsistency-tolerant semantics introduced in previous chapters. Building on this encoding, we define trace-based logical explanations of access decisions and, in a separate step, propose a framework for generating and evaluating natural language explanations using large language models. The main results of this chapter have been published in [Laouar *et al.* 2025b, Laouar *et al.* 2024].

5.1 Overview of Access Control

Access control models are mechanisms for specifying and enforcing security policies within organisations. They ensure that the resources of an information system are only used in the scope in which they are intended to be used. Different types of rules have been defined in the literature of access control to regulate access to sensitive information and control the flow of authorisations.

An access control system is defined upon sets of subjects (users), which represent active entities, actions such as *read* and *write*, and objects (resources), which are passive entities, such as files and databases. Most access control models are underspecified in the sense that they require some adaptations before being put into practice. In addition, not all models are directly comparable because different models meet different needs depending on the application field. In the following, we provide a classification of access control models, including examples of the systems that implement those models if available. Before that, let us recall the principal access right types or deontic modalities that can be defined by the discussed models.

- **Permissions**: explicit authorisation rules, present in all models. Models that only implement permissions are based on the closed-world assumption, where anything that is not permitted is implicitly prohibited.
- **Prohibitions**: explicit negative permission rules. Often represents the opposite of a permission. The simultaneous use of permissions and prohibitions in a model could lead to conflicts [Al-Kahtani & Sandhu 2004, Jajodia *et al.* 2001, Anton *et al.* 2024].
- **Obligations**: usually specified in terms of a subject that should perform a particular action on a target when a specified condition is met [Kalam *et al.* 2003, Charalambides *et al.* 2005].
- Other possible rules include, but are not limited to, recommendation, delegation, …

5.1.1 A Possible Classification of Access Control Models

Models in extension

Models in extension, such as HRU (Harrison-Ruzzo-Ullman) [Harrison *et al.* 1976], TAM (Typed Access Matrix) [Sandhu 1992], and the Take-Grant Model [Bishop 1981], focus on explicitly representing and manipulating permissions to capture how rights can be created, transferred, or revoked within a system. The HRU model extends the classical access matrix by allowing dynamic changes to the set of rights through commands, making it expressive enough to model practical security policies, but also leading to undecidability in safety analysis. The TAM model refines this approach by associating types with subjects and objects, which constrains

how rights may evolve and enables better decidability results for safety properties. The Take-Grant model adopts a graph-based view, where nodes represent subjects or objects, and edges represent rights, emphasising the transfer of permissions through the "take" and "grant" rules. A practical illustration is found in shared file systems, where a user (subject) may "grant" another user read access to a file (object), or "take" access from an inherited permission chain, reflecting how delegation and propagation of rights can be analysed through these models.

Discretionary Access Control

The discretionary access control (DAC) model is based on user identity and the concept of ownership, whereby an owner is responsible for assigning privileges for their objects to other subjects. It consists of three main elements: subjects, objects, and an access matrix that associates subjects with objects and the access rights that connect them. This model is used in operating systems such as Linux to manage file systems. The Lampson [Lampson 1974] and Graham-Dening [Graham & Denning 1971] models were the first DAC models. These models are administered using basic commands to create and delete subjects and objects and to add and revoke privileges. A very important issue here is safety; in other words, whether there is a sequence of commands that would allow a user to have a given right over an object.

Mandatory Access Control

The Mandatory Access Control (MAC) model is a strict model of access control. It emerged in military contexts, particularly in the context of multilevel security (MLS). The policy is uniformly enforced on subjects, and rights are assigned only through the central administration of the system. It is based on classification levels for subjects and objects. In practice, the Bell-LaPadula [Bell 1973] and Biba Integrity Model [Biba 1977] are the most common models, where the concept of dominance relationships is introduced. For example, the Bell-LaPadula model imposes two main rules: no read up: no subject can read an object with a classification level higher than its authorisation level and no write down: no subject can write to an object with a classification level lower than its authorisation level. The Biba model focuses on data integrity and reinforces the opposite of these rules (no write up and no read down).

Role-Based Access Control

The Role-Based Access Control (RBAC) model [Sandhu *et al.* 2000] abstracts subjects through their roles, which are dictated by the security policy. Privileges are assigned indirectly so that subjects are assigned their roles, and privileges are granted at the role level. Consequently, when a specific role is assigned to a subject, it obtains the privileges associated with that role. This model is the most widely used in practice, where there are several implementations, particularly in database management systems and enterprise resource management systems. The concept of roles allows for a compact and easy representation of the policy, which also eases the administrative work load. The RBAC model has been extended in multiple ways:

- "Flat RBAC" (RBAC0): This is the basic RBAC model. The NIST model requires that the roles assigned to a user can be determined, as well as the users assigned to a specific role.
- 'Hierarchical RBAC" (RBAC1): In this model, the roles are organised into a hierarchical structure (directed acyclic graph or tree), where "higher" roles inherit all the permissions

from connected "lower" roles. Hierarchical RBAC is subdivided into general and restricted hierarchical RBAC.

- "Constrained RBAC" (RBAC2): This extension adds the application of separation of duties to roles. In the NIST RBAC model, the separation of duties can be static or dynamic. RBAC1 and RBAC2 are independent of each other in the sense that a system can implement one without the other.
- "Symmetric RBAC" (RBAC3): This is a complete RBAC model that incorporates RBAC0, RBAC1, and RBAC2. In addition, RBAC3 allows for a partial order hierarchy between roles and not just a total order structure.

In addition, the RBAC model was extended with negative authorisations to specify both permissions and prohibitions [Al-Kahtani & Sandhu 2004].

Attribute Based Access Control

Attribute-Based Access Control [Yuan & Tong 2005] is a model in which access rights are granted to users based on policies that combine a set of arbitrary attributes possessed by the access requester with a selected set of attributes of the object. ABAC allows administrators to apply an access control policy without complete prior knowledge of the specific subject, using other data points that could be strong indicators of identity [Jin *et al.* 2012]. When combined with other attributes, indicators can form the basis for sufficient trust in the identity and ownership of the device to authorise access to services and transactions. In addition to subject and object attributes, this model supports environmental attributes, which are useful for systems that depend on the current state of the system and sensors for external events. ABAC is considered an expressive model because it allows for a multitude of concepts to be expressed through attributes.

Category-Based Access Control

Category-Based Access Control (CBAC) is an access control paradigm where permissions are defined over categories of subjects, objects, and actions, rather than individual entities. Categories are typically constructed from attributes, providing a structured and scalable way to capture policies. A notable extension is C-ABAC [Fernández *et al.* 2019], which gives a formal specification of Attribute-Based Access Control (ABAC) using the CBAC meta-model [Barker 2009]. In C-ABAC, categories are defined through Boolean conditions over attributes of users, resources, and environments, allowing flexible grouping. Authorisations are then assigned to categories through relations, enabling concise policy specification while supporting extensions such as prohibitions and obligations. From an administrative perspective, the authors in [Bertolissi *et al.* 2020] introduced admin-CBAC, an administration model that formalises how categories, permissions, and delegation rules can be managed consistently in dynamic environments, thereby strengthening CBAC as a practical and maintainable framework.

5.1.2 Organisation Based Access Control (OrBAC)

Since its proposition, the Organisation-Based Access Control OrBAC [Kalam *et al.* 2003] attracted a lot of interest in the specification and modelling of security policies. Thanks to its ability to express a large set of access control rules while retaining a compact and abstract definition of the policy. In addition, concrete access rules are inferred in OrBAC. In the literature,

there are several recent applications of the OrBAC model, notably in domains such as Security Protocols [Adelin *et al.* 2022], IoT [Laamech *et al.* 2022] and Safety Engineering [Chouchani *et al.* 2022]. Moreover, the model has been extended in different ways [Abou El Kalam & Deswarte 2006, Bouij-Pasquier *et al.* 2015].

OrBAC is described using an entity relation diagram. The basic concepts in access control are subjects, which represent users, objects, which are passive entities like resources and files, and actions which are the basic operations performed by a subject on an object, such as read and write. OrBAC [Kalam *et al.* 2003] considers these concepts as concrete entities and abstracts each of them into abstract entities. Subjects that fulfil the same functions are grouped in roles, objects that satisfy common properties are grouped in views, and actions that share the same principles are grouped in activities. In addition, OrBAC considers organisations as concrete concepts and always defines rules in an organisation.

Concrete entities are linked to abstract entities using the connection relations Employ, Use and Consider. Employ(org,s,r) means that the subject "s" plays the role "r" within the organisation "org". Use(org,o,v) means that the object "o" is used in the view "v" within the organization "org". Consider(org,α,a) means that the organisation "org" considers that the action "α" fall within the activity "a". Moreover, OrBAC is said to be dynamic because it uses contexts. A privilege may be granted in one context and denied in another. It can represent environmental variables, such as timeOfDay, or a particular state like emergency. Contexts are defined between a subject, an action and an object in an organisation using the relation Define such that Define(org,s,α,o,c) means that in the organization "org", the context "c" holds between the subject "s", the object "o" and the action "α".

Policy rules, which are also called abstract privileges, are defined using the relations Permission, Prohibition, Obligation and Recommendation. Each rule admits five arguments; Permission(org, r, a, v, c) means that the organization "org" grants the role "r" the permission to perform the activity "a" on the view "v" if the context "c" holds. The remaining rules are defined similarly.

Access rights, or concrete privileges, are derived for concrete entities, namely subjects, actions, and objects. They are represented with the relations Is-permitted, Is-prohibited, Is-obliged, and Is-recommended. The concrete privilege Is-permitted(s,α,o) means that the subject "s" is allowed to perform the action "α" on the object "o". Figure 5.1 summarises the relational diagram of the OrBAC model.

Definition 5.1 (OrBAC policy)**.** *An* OrBAC *policy is a tuple of the form* $\mathcal{O} = \langle \mathcal{B}, \mathcal{P}, \mathcal{U} \rangle$*, where* $\mathcal{B}$ *is a set of basic concrete and abstract concepts,* $\mathcal{P}$ *is the set of abstract rules* Permission*,* Prohibition*,* Obligation *and* Recommendation*, and* $\mathcal{U}$ *is a set of rules of the form* Employ*,* Use*,* Consider *and* Define*.*

Access rights, or concrete privileges, are derived for concrete entities, namely subjects, actions, and objects. They are represented with the relations Is-permitted, Is-prohibited, Is-obliged, and Is-recommended. The concrete privilege Is-permitted(s,α,o) means that the subject "s" is allowed to perform the action "α" on the object "o".

Definition 5.2 (Privilege derivation)**.** *Concrete permissions are derived from an* OrBAC *policy using the following rule:*
$\forall$org, $\forall$s, $\forall\alpha$, $\forall$o, $\forall$r, $\forall$a, $\forall$v, $\forall$c,
Permission(org,r,a,v,c) $\wedge$ Employ(org,s,r) $\wedge$ Use(org,o,v) $\wedge$ Consider(org, α, a) $\wedge$ Define(org,s,α,o,c)
$\models$ Is-permitted(s,α,o)
The privileges Is-prohibited, Is-obliged, and Is-recommended are derived similarly.

In the following, we denote by $\mathcal{O} \models \mathsf{Is\text{-}permitted}(\mathsf{s}, \alpha, \mathsf{o})$ the fact that a concrete permission is derived and by $\mathcal{O} \not\models \mathsf{Is\text{-}permitted}(\mathsf{s}, \alpha, \mathsf{o})$ the fact that a concrete permission is not derived from the policy $\mathcal{O}$, respectively.

Hierarchies in the OrBAC model

Organisations are often structured hierarchically, allowing a parent entity to pass permissions or prohibitions to a child entity. In the OrBAC model, this inheritance has been studied in [Cuppens *et al.* 2004], where abstract rules can be inferred through hierarchies of roles, views, activities, and organisations. In our work, we adopt the same hierarchical principles but focus on deriving concrete privileges (Is-permitted, Is-prohibited). Organisational hierarchies are captured by the predicate $\mathsf{subOrganisationOf}(\mathsf{org}_1, \mathsf{org}_2)$, ensuring that rules defined in a parent organisation also apply to its sub-organisation. Similarly, role hierarchies are handled via two relations:

- Sub-roles (SubRole), where a role inherits both permissions and prohibitions of its parent.
- Senior-roles (SeniorRole), where senior roles inherit permissions of junior ones, while juniors inherit prohibitions from seniors.

Both relations are organisation-specific, ensuring consistency with other OrBAC constructs and allowing flexible, context-dependent role hierarchies.

OrBAC policy example

Let us now introduce an OrBAC policy example to illustrate the previous notions. We consider a staff-exchange research project, where sensitive data is accessed by users from multiple organisations. This requires elaborate access control systems, as multiple roles and different contexts can hold at the same time. For instance, some project data are internal to one organisation and therefore cannot be accessed by users from other organisations. However, some exceptions are defined, such as access by team leaders or project coordinators.

Example 5.1. *Let $\mathcal{O} = \langle \mathcal{B}, \mathcal{P}, \mathcal{U} \rangle$ be an* OrBAC *policy, where the sets of abstract rules $\mathcal{P}$ and the connection relations $\mathcal{U}$ are given by:*

$$\mathcal{P} = \left\{ \begin{array}{l} \mathsf{Permission}(\mathsf{consortium}, \mathsf{coordinator}, \mathsf{consult}, \mathsf{data}, \mathsf{management}), \\ \mathsf{Permission}(\mathsf{univ1}, \mathsf{teamLeader}, \mathsf{consult}, \mathsf{data}, \mathsf{management}), \\ \mathsf{Prohibition}(\mathsf{consortium}, \mathsf{staffMember}, \mathsf{consult}, \mathsf{internalData}, \mathsf{default}), \\ \mathsf{subOrganisationOf}(\mathsf{univ1}, \mathsf{consortium}), \quad \mathsf{subOrganisationOf}(\mathsf{inst1}, \mathsf{consortium}) \end{array} \right\}$$

$$\mathcal{U} = \left\{ \begin{array}{ll} \mathsf{Employ}(\mathsf{consortium}, \mathsf{coordinator}, \mathsf{Ben}), & \mathsf{Employ}(\mathsf{univ1}, \mathsf{teamLeader}, \mathsf{Ben}), \\ \mathsf{Employ}(\mathsf{inst1}, \mathsf{staffMember}, \mathsf{Ben}), & \mathsf{Use}(\mathsf{consortium}, \mathsf{data}, \mathsf{dataset1}), \\ \mathsf{Define}(\mathsf{univ1}, \mathsf{Ben}, \mathsf{read}, \mathsf{dataset1}, \mathsf{management}), & \mathsf{Use}(\mathsf{inst1}, \mathsf{internalData}, \mathsf{dataset1}), \\ \mathsf{Define}(\mathsf{inst1}, \mathsf{Ben}, \mathsf{read}, \mathsf{dataset1}, \mathsf{default}), & \mathsf{Consider}(\mathsf{consortium}, \mathsf{consult}, \mathsf{read}) \end{array} \right\}$$

Using Definition 5.2, both the privileges:

$$\mathcal{O} \models \mathsf{Is\text{-}permitted}(\mathsf{Ben}, \mathsf{read}, \mathsf{dataset1}) \quad \text{and} \quad \mathcal{O} \models \mathsf{Is\text{-}prohibited}(\mathsf{Ben}, \mathsf{read}, \mathsf{dataset1})$$

are inferred from $\mathcal{O}$.

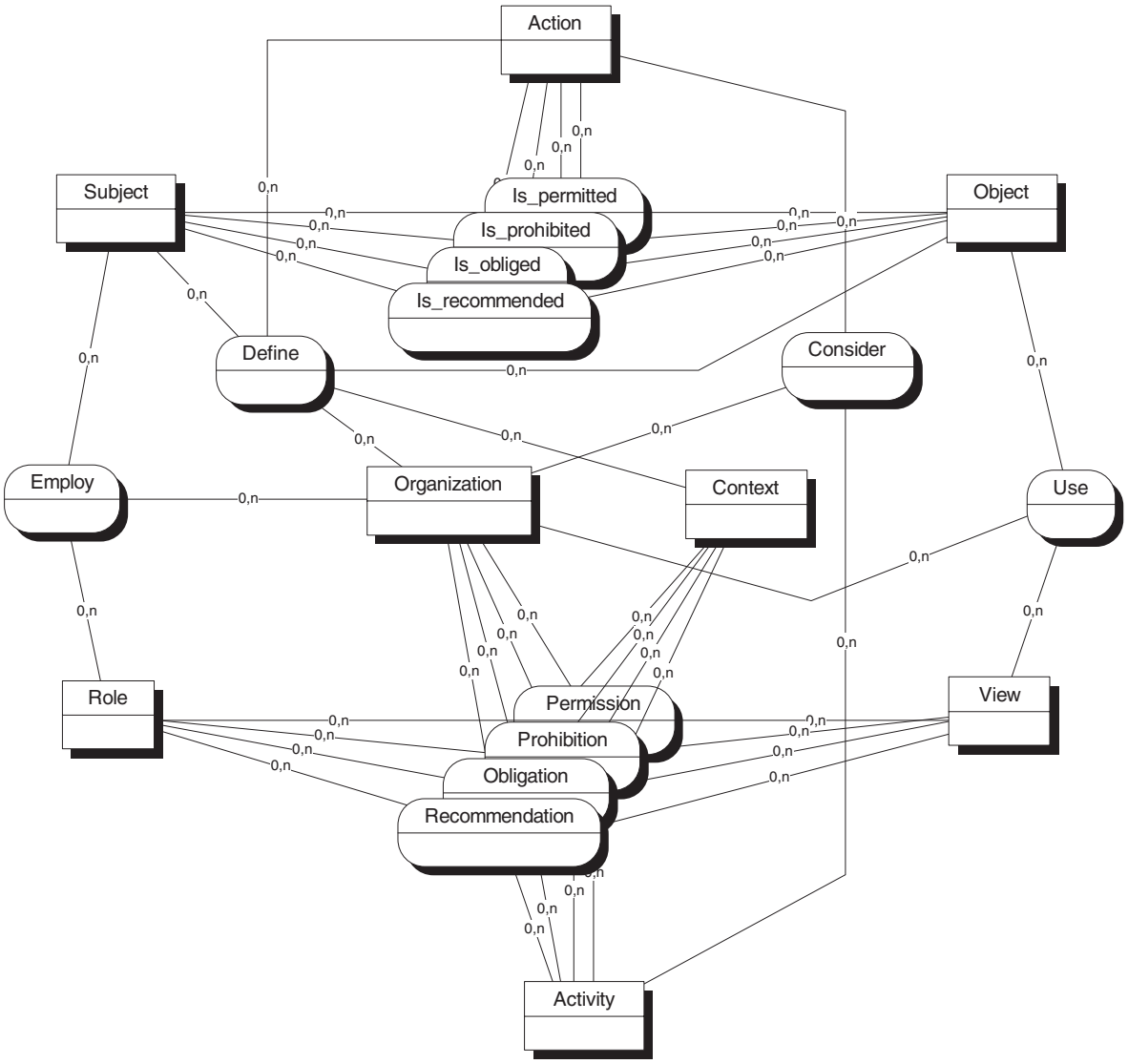


Figure 5.1: The OrBAC Model [Kalam *et al.* 2003]

5.1.3 Inconsistency in the OrBAC Model

The OrBAC model, as well as several other models such as RBAC and ABAC, allow the use of multiple types of access rights, making the model expressive. However, the simultaneous use of contradictory access rights, such as permission and prohibition, could lead to access conflicts. In practice, access conflicts may be the result of shortcomings in the representation framework or even design mistakes. Moreover, these access conflicts can be a normal result of the inclusion of rules with exceptions and general and specific rules in the security policy. For the OrBAC model, the definition of dynamic rules through the use of contexts can also be the result of conflicts when two contradictory contexts hold at the same time. Inconsistency might also be a direct result of the use of hierarchies in a policy. For instance, in our running example, the inference of both concrete permission and prohibition is due to the application of the rules of the parent organisation consortium.

Definition 5.3 (Inconsistent OrBAC policy)**.** *Let $\mathcal{O} = \langle \mathcal{B}, \mathcal{U}, \mathcal{P} \rangle$ be an* OrBAC *policy. $\mathcal{O}$ is said to be inconsistent if there exists a subject "*s*", an object "*o*" and an action "α" such that both* Is-permitted(s,α,o) and Is-prohibited(s,α,o) are derived from $\mathcal{O}$. Otherwise, $\mathcal{O}$ is said to be consistent.

Example 5.1 (Continued)**.** *The* OrBAC *policy of our example, namely $\mathcal{O}$, is inconsistent because it infers both* Is-permitted *and* Is-prohibited *privileges for the subject* Ben*, the action* read*, and the object* dataset1*.*

Before discussing the possible ways and the solutions proposed in the literature to address inconsistency in OrBAC policies, let us define the notion of conflict in an OrBAC policy. We proceed in the same way for inconsistent description logic knowledge bases, where a conflict is defined as an inclusion-minimal subset of a knowledge base that allows the entailment of a contradiction. In OrBAC, a contradiction is the derivation of both permission and prohibition for the same subject, action, and object. Hence, a conflict can be defined as a minimal subset of policy rules that entails a contradiction. In essence, a subset of the union of the sets of rules $\mathcal{P}$ and $\mathcal{U}$.

Definition 5.4 (OrBAC conflict)**.** *Let $\mathcal{O} = \langle \mathcal{B}, \mathcal{U}, \mathcal{P} \rangle$ be an* OrBAC *policy. Let $\mathcal{C} \subseteq \mathcal{U} \cup \mathcal{P}$ be a subset of rules from $\mathcal{O}$. $\mathcal{C}$ is a conflict in $\mathcal{O}$ if:*

- $\langle \mathcal{B}, \mathcal{C} \rangle \models$ Is-permitted(s, α, o) *and* $\langle \mathcal{B}, \mathcal{C} \rangle \models$ Is-prohibited(s, α, o)*; and,*
- $\forall T(\cdot) \in \mathcal{C}$, $\langle \mathcal{B}, \mathcal{C} \setminus \{T(\cdot)\} \rangle \not\models$ Is-permitted(s, α, o) *or* $\langle \mathcal{B}, \mathcal{C} \setminus \{T(\cdot)\} \rangle \not\models$ Is-prohibited(s, α, o}).

Example 5.1 (Continued)**.** *Consider the following subset of rules from our example's*

policy $\mathcal{O}$:

$$\mathcal{C} = \left\{ \begin{array}{l} \mathsf{Permission}(\mathsf{univ1}, \mathsf{teamLeader}, \mathsf{consult}, \mathsf{data}, \mathsf{management}) \\ \mathsf{Prohibition}(\mathsf{consortium}, \mathsf{staffMember}, \mathsf{consult}, \mathsf{internalData}, \mathsf{default}) \\ \mathsf{subOrganisationOf}(\mathsf{inst1}, \mathsf{consortium}) \\ \mathsf{subOrganisationOf}(\mathsf{univ1}, \mathsf{consortium}) \\ \mathsf{Employ}(\mathsf{inst1}, \mathsf{staffMember}, \mathsf{Ben}) \\ \mathsf{Employ}(\mathsf{univ1}, \mathsf{teamLeader}, \mathsf{Ben}) \\ \mathsf{Define}(\mathsf{inst1}, \mathsf{Ben}, \mathsf{read}, \mathsf{dataset1}, \mathsf{default}) \\ \mathsf{Define}(\mathsf{univ1}, \mathsf{Ben}, \mathsf{read}, \mathsf{dataset1}, \mathsf{management}) \\ \mathsf{Use}(\mathsf{consortium}, \mathsf{data}, \mathsf{dataset1}) \\ \mathsf{Use}(\mathsf{inst1}, \mathsf{internalData}, \mathsf{dataset1}) \\ \mathsf{Consider}(\mathsf{consortium}, \mathsf{consult}, \mathsf{read}) \end{array} \right\}$$

$\mathcal{C}$ is a conflict of $\mathcal{O}$ because

$$\langle \mathcal{B}, \mathcal{C} \rangle \models \mathsf{Is\text{-}permitted}(\mathsf{Ben}, \mathsf{read}, \mathsf{dataset1}) \quad \text{and} \quad \langle \mathcal{B}, \mathcal{C} \rangle \models \mathsf{Is\text{-}prohibited}(\mathsf{Ben}, \mathsf{read}, \mathsf{dataset1})$$

Problem Statement

Two important questions arise at this level:

- How can we efficiently make a decision from a conflicting policy, endowing the model with the capacity to handle contradictory rules, without compromising its safety?
- How can we explain the origin of the problem or the outcome decision, in order to allow a potential correction or an improvement to the described policy?

The following sections attempt to answer these questions, first by providing an ontology that supports conflict resolution in OrBAC. Then, building on that, the last section provides a specialised framework to generate natural language explanations, relying on large language models (LLMs), and to evaluate the outcome explanations, relying on formal logical explanations.

5.2 A Repair-Based Semantics for OrBAC

In this section, we introduce our methods for conflict resolution in the OrBAC model. First, we introduce a possibilistic version of OrBAC. Then, we propose the OrBAC ontology, which is an ontology that translates the OrBAC model into OWL (Web Ontology Language). This representation allows us to overcome the shortcomings of the existing conflict resolution methods, taking advantage of existing tractable mechanisms that have desirable properties. Our solution, which is inspired by inconsistency-tolerant semantics, deals with conflicts as part of the OrBAC model and always guarantees that a decision is made.

To provide a clear roadmap for this section, Figure 5.2 illustrates the layered structure of our access control conflict resolution framework. At the foundation lies the OrBAC basic model, structured around four internal units: concrete entities, abstract entities, connection relations, and policy rules. This first layer is represented using an ontology. On top of this, hierarchies

of organisations and rules are introduced to capture inheritance principles. The preferences are then acquired in the ontology using SWRL, enabling the expression of priorities in the application of the policy. Access rights are derived through SPARQL queries, which formalise the decision process. Finally, conflicts among policies are resolved at the highest layer using the Acceptance method.

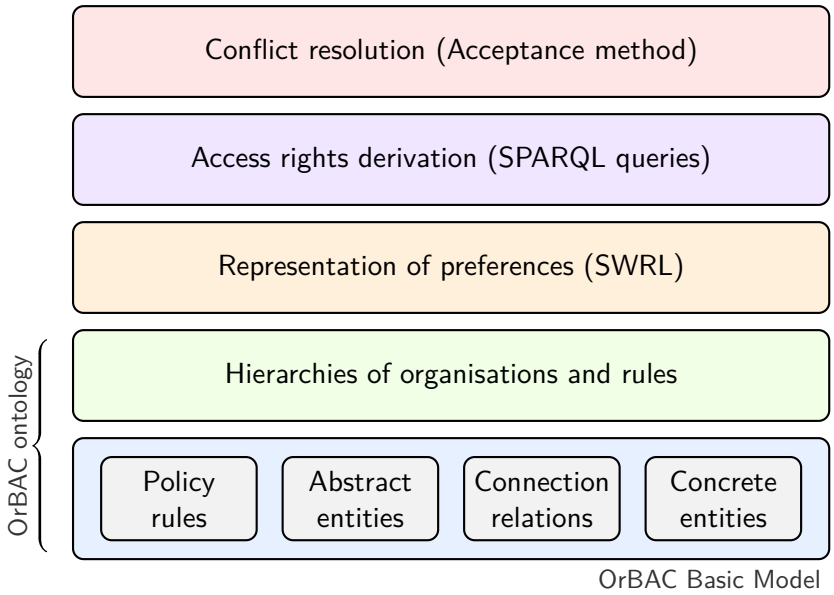


Figure 5.2: Multi-layer view of the proposed OrBAC conflict resolution framework.

5.2.1 A Possibilistic OrBAC Model

A crucial difference between our representation of uncertainty and previous approaches [Cuppens *et al.* 2007, Benferhat *et al.* 2003, Benferhat & El Baida 2004] is the explicit separation between fully certain and uncertain rules. In our framework, abstract policy rules (Permission, Prohibition, etc.) are considered fully certain, while connection rules (Employ, Use, Consider, and Define) are possibly uncertain, in a uniform way across the policy, analogous to the TBox–ABox separation in lightweight ontologies. Uncertain rules are further partially preordered, providing flexibility in handling heterogeneous policy elements: While orders can be naturally defined among similar entities (e.g., roles or contexts), it is often unnatural to compare different types of entity, making a partial preorder a more realistic and adaptable choice.

The partial preorder over an OrBAC policy is represented by assigning to the uncertain rules weights from a non-empty partially ordered set denoted by $\mathsf{U} = u_1, \ldots, u_n, \mathbb{1}$. This set U is equipped with a strict partial order $\rhd$, to form the uncertainty scale $\mathbb{L} = (\mathsf{U}, \rhd)$. By definition, $\rhd$ is an irreflexive and transitive relation. Some weights may be incomparable; that is, for any two weights $u_i, u_j \in \mathsf{U}$, $u_i \bowtie u_j$ means that neither $u_i \rhd u_j$ nor $u_j \rhd u_i$ holds. The weight $\mathbb{1}$, representing full certainty, is the greatest element of U in the sense that for every $u_i \in \mathsf{U}$, where $u_i \neq \mathbb{1}$, we have $\mathbb{1} \rhd u_i$.

Definition 5.5 (Possibilistic OrBAC policy)**.** *A possibilistic* OrBAC *policy is a tuple* $\mathcal{O}_{\mathbb{L}} = \langle \mathcal{B}, \mathcal{P}, \mathcal{U}, \mathbb{L} \rangle$ *such that* $\mathcal{B}$ *is a set of basic concrete and abstract concepts,* $\mathcal{P}$ *is a set of abstract (fully certain) rules* Permission*,* Prohibition*,* Obligation *and* Recommendation*, and* $\mathcal{U}$ *is a set of*

(uncertain) rules of the form Employ, Use, Consider *and* Define, *and* $\mathbb{L} = (\mathsf{U}, \rhd)$ *is an uncertainty scale, where a single weight from* U *is assigned to each policy uncertain rule.*

The following example adapts the policy $\mathcal{O}$ of Example 5.1 by adding weights to its uncertain rules.

Example 5.2. *Let* $\mathcal{O}_{\mathbb{L}} = \langle \mathcal{B}, \mathcal{P}, \mathcal{U}, \mathbb{L} \rangle$ *be a possibilistic version of the policy* $\mathcal{O}$ *from Example 5.1, where the sets of abstract rules* $\mathcal{P}$ *is the same as in* $\mathcal{O}$*, namely:*

$$\mathcal{P} = \left\{ \begin{array}{l} \mathsf{Permission}(\mathsf{consortium}, \mathsf{coordinator}, \mathsf{consult}, \mathsf{data}, \mathsf{management}), \\ \mathsf{Permission}(\mathsf{univ1}, \mathsf{teamLeader}, \mathsf{consult}, \mathsf{data}, \mathsf{management}), \\ \mathsf{Prohibition}(\mathsf{consortium}, \mathsf{staffMember}, \mathsf{consult}, \mathsf{internalData}, \mathsf{default}), \\ \mathsf{subOrganisationOf}(\mathsf{univ1}, \mathsf{consortium}), \quad \mathsf{subOrganisationOf}(\mathsf{inst1}, \mathsf{consortium}) \end{array} \right\}$$

The connection rules in $\mathcal{U}$ *are assigned weights from the uncertainty scale* $\mathbb{L} = (\mathsf{U}, \rhd)$*, where the set of weights is given by* $\mathsf{U} = \{w_1, w_2, w_3, w_4, w_5, \mathbb{1}\}$*, and the partial order* $\rhd$ *is defined such that:*

- $\mathbb{1} \rhd w_1 \rhd w_2 \rhd w_3$
- $\mathbb{1} \rhd w_4 \rhd w_5$
- *The remaining relations are incomparable, namely* $w_1 \bowtie w_4$*,* $w_1 \bowtie w_5$*,* $w_2 \bowtie w_4$*,* $w_2 \bowtie w_5$*,* $w_3 \bowtie w_4$*, and* $w_3 \bowtie w_5$*.*

The resulting set of rules in $\mathcal{U}$ *after adding the weights is given as follows:*

$$\mathcal{U} = \left\{ \begin{array}{ll} \mathsf{Employ}(\mathsf{consortium}, \mathsf{coordinator}, \mathsf{Ben}, w_1) & \mathsf{Define}(\mathsf{inst1}, \mathsf{Ben}, \mathsf{read}, \mathsf{dataset1}, \mathsf{default}, w_5) \\ \mathsf{Employ}(\mathsf{univ1}, \mathsf{teamLeader}, \mathsf{Ben}, w_2) & \mathsf{Use}(\mathsf{consortium}, \mathsf{data}, \mathsf{dataset1}, \mathbb{1}) \\ \mathsf{Employ}(\mathsf{inst1}, \mathsf{staffMember}, \mathsf{Ben}, w_3) & \mathsf{Use}(\mathsf{inst1}, \mathsf{internalData}, \mathsf{dataset1}, \mathbb{1}) \\ \mathsf{Define}(\mathsf{univ1}, \mathsf{Ben}, \mathsf{read}, \mathsf{dataset1}, \mathsf{management}, w_4) & \mathsf{Consider}(\mathsf{consortium}, \mathsf{consult}, \mathsf{read}, \mathbb{1}) \end{array} \right\}$$

Remark. Note that the ordering of uncertain rules comes from the ordering of abstract entities. For instance, a coordinator is considered to be more privileged than a teamLeader. Then, each rule that attributes the role coordinator is considered to be more preferred to any rule that attributes the role teamLeader. Note also that the rules for assigning roles and the rules defining contexts are considered incomparable. Automatic priority assignment strategies have been discussed in [Cuppens *et al.* 2007], however, no actual strategy has been put into practice. In the following section, we introduce our ordering propagation strategy that is inspired by the discussion in [Cuppens *et al.* 2007], along with the OrBAC ontology.

5.2.2 The OrBAC Ontology

We propose an ontological encoding of the OrBAC model. Such an encoding facilitates the representation of uncertainty over a policy and simplifies the reasoning over a policy to answer SPARQL queries. A very similar ontology has been defined for OrBAC in [de Vergara *et al.* 2008, de Vergara *et al.* 2009]. The difference lies in the way hierarchies are handled, in addition to the integration of our representation of uncertainty. Note that the cited ontology has been used in a different context, which is that of detecting and reacting to network attacks.

In addition, the use of ontologies to specify access control policies has been widely explored, particularly for managing web resources or protecting linked data [Kirrane *et al.* 2017]. Existing solutions, such as WAC [W3C 2012], S4AC [Costabello *et al.* 2012], G-FAF [Kirrane *et al.* 2013], and lightweight vocabularies [Sacco *et al.* 2011, Sacco & Breslin 2012], provide mechanisms for expressing permissions, authorisations, and priorities, sometimes combined with query rewriting techniques [Kirrane *et al.* 2020]. While these approaches offer useful tools for specifying access rules, they are often limited in handling conflicts or require elaborate strategies when dynamic rules and hierarchies are involved.

An ontology can be encoded in OWL [Dean & (eds.) 2004], which provides different syntaxes for representing knowledge on the Semantic Web. In practice, RDF triples are often used to encode OWL axioms, as well as additional constructs such as class hierarchies and property constraints. We briefly recall the main elements that are relevant for our work, while noting that OWL also supports richer notions such as complex concepts and property axioms. An ontology is defined over finite sets of classes C, object properties P, and individuals I. Classes are represented with triples of the form $\langle x, \mathsf{rdf:type}, \mathsf{owl:Class}\rangle$ for $x \in C$, while subclass relations are expressed as $\langle x, \mathsf{rdfs:subClassOf}, y\rangle$ for $x, y \in C$. Object properties are declared through triples $\langle x, \mathsf{rdf:type}, \mathsf{owl:ObjectProperty}\rangle$ for $x \in P$, with their domain and range specified by $\langle x, p, y\rangle$ where $p \in \{\mathsf{rdfs:domain}, \mathsf{rdfs:range}\}$, $x \in P$, and $y \in C$. Additional characteristics can also be attached, such as $\langle x, \mathsf{rdf:type}, \mathsf{owl:TransitiveProperty}\rangle$ to indicate transitivity. Assertions about individuals are expressed as triples $\langle x, p, y\rangle$, where $x \in I$, p is a property, and y is either another individual or a literal. Reasoning over ontologies can be extended using the Semantic Web Rule Language (SWRL) [Horrocks *et al.* 2004] and SPARQL [Harris & (eds.) 2013] queries. SWRL rules are expressed as implications of the form $\varphi \rightarrow \phi$, where φ and ϕ are conjunctions of atoms built from classes and properties. SPARQL queries provide a practical means to extract or check knowledge: ASK queries return a boolean answer indicating whether a pattern has a match; SELECT queries return bindings of variables that satisfy the pattern; and UNION queries combine results of multiple patterns.

Ontology classes and object properties

Representing OrBAC concrete and abstract entities. The ontology distinguishes between concrete and abstract concepts:

- **Concrete entities**:
  - orbac:Object, orbac:Subject, orbac:Action represent objects, subjects, and actions.
  - orbac:Organisation represents organisations and is defined as a subclass of orbac:Subject.
  - All concrete entities are subclasses of orbac:ConcreteConcept.
- **Abstract entities**:
  - orbac:Activity, orbac:View, orbac:Role represent activities, views, and roles, respectively.
  - All abstract entities are subclasses of orbac:AbstractConcept.
  - orbac:Context represents contexts.

Figure 5.3 summarises the concrete entities and their connection with the context entity.

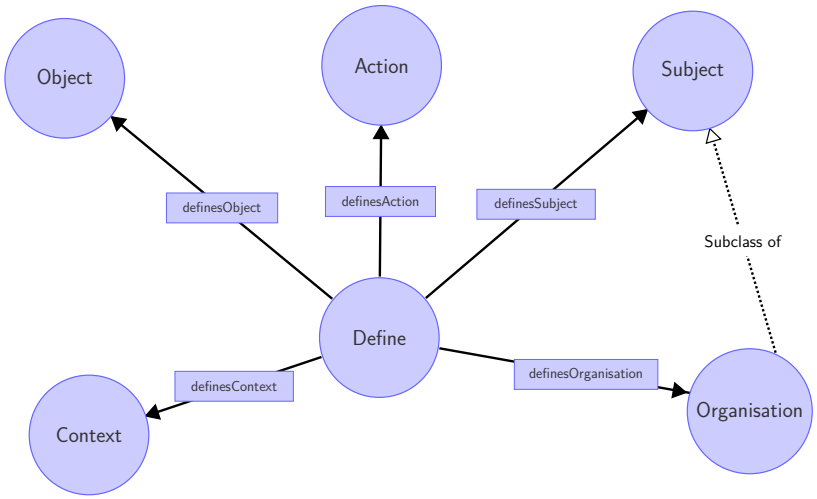


Figure 5.3: Representation of concrete entities and the context in the OrBAC ontology

Representing policy rules. Abstract policy rules (or *access types*) are represented using:

- orbac:Permission, orbac:Prohibition, orbac:Obligation, orbac:Recommendation,

all of which are subclasses of orbac:AccessType. Each access type instance is linked to exactly one entity of each kind via the following object properties:

- orbac:accessTypeOrg – connects to an organisation,
- orbac:accessTypeRole – connects to a role,
- orbac:accessTypeActivity – connects to an activity,
- orbac:accessTypeView – connects to a view,
- orbac:accessTypeContext – connects to a context.

Figure 5.4 depicts the representation of the policy rules within the OrBAC ontology.

Representing connection rules. Connection rules link concrete entities with their abstract counterparts, and define whether a context applies to a subject, action, and object in an organisation. They are represented as subclasses of orbac:ConnectionRule:

- orbac:Consider, connected via:
  - orbac:considersActivity, orbac:considersAction, orbac:considersOrg.
- orbac:Use, connected via:
  - orbac:usesView, orbac:usesObject, orbac:usesEmployer.
- orbac:Employ, connected via:
  - orbac:employesRole, orbac:employesEmployee, orbac:employesEmployer.

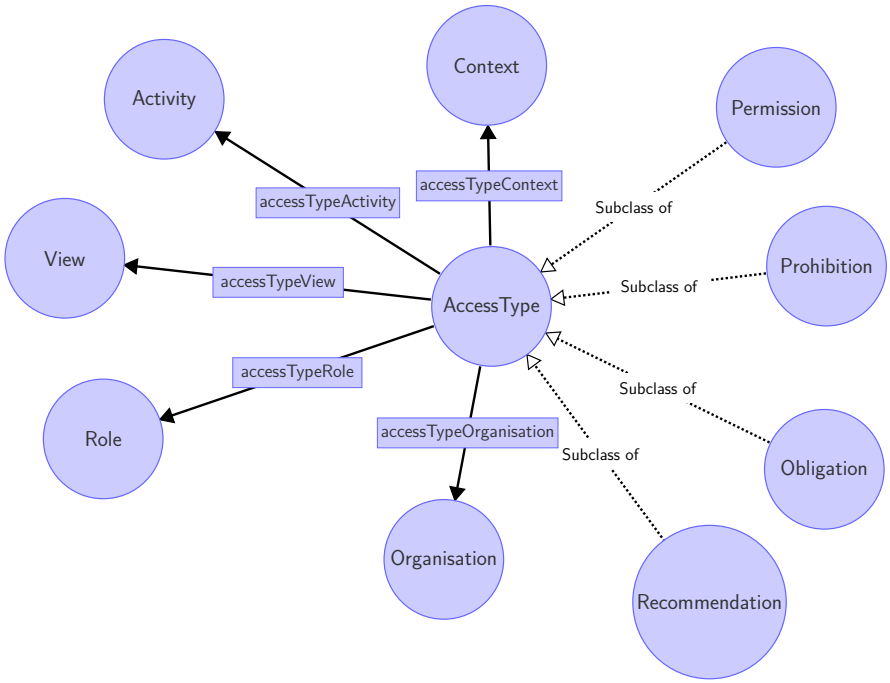


Figure 5.4: Representation of the policy rules in the OrBAC ontology

- orbac:Define, connected via:
  - orbac:definesSubject, orbac:definesAction, orbac:definesObject, orbac:definesContext.

Figure 5.5 illustrates the representation of the connection rules between the abstract and concrete entities, in addition to hierarchies of roles and organisations within the OrBAC ontology.

Representing hierarchies of organisations and roles. The organisational hierarchy is represented using the orbac:subOrganisationOf object property, which links two organisations. Every rule or relation that is defined in a parent organisation holds for the suborganisation. For instance, if $\langle$perm1, orbac:accessTypeOrg, consortium$\rangle \wedge \langle$univ1, orbac:subOrganisationOf, consortium$\rangle \rightarrow \langle$perm1, orbac:accessTypeOrg, univ1$\rangle$. The inverse does not hold.

Role hierarchies are encoded as follows:

- orbac:SubRole and orbac:SeniorRole classes as subclasses of the orbac:Role class.
- The property orbac:hasParent links a sub-role or a senior-role to a parent role.
- Sub-roles and senior-roles are defined in an organisation using the property orbac:subRoleOrg, which links a sub-role or a senior-role to an organisation.

Representing uncertainty. Uncertainty is encoded using a preference relation defined between individuals of the connection relations, namely Employ, Consider, Use, and Define. The abstract policy rules are assumed to be fully certain. Defining a preference between individuals of the uncertain rules is achieved using the transitive object property $\langle x$, orbac:isPreferredTo, $y\rangle$,

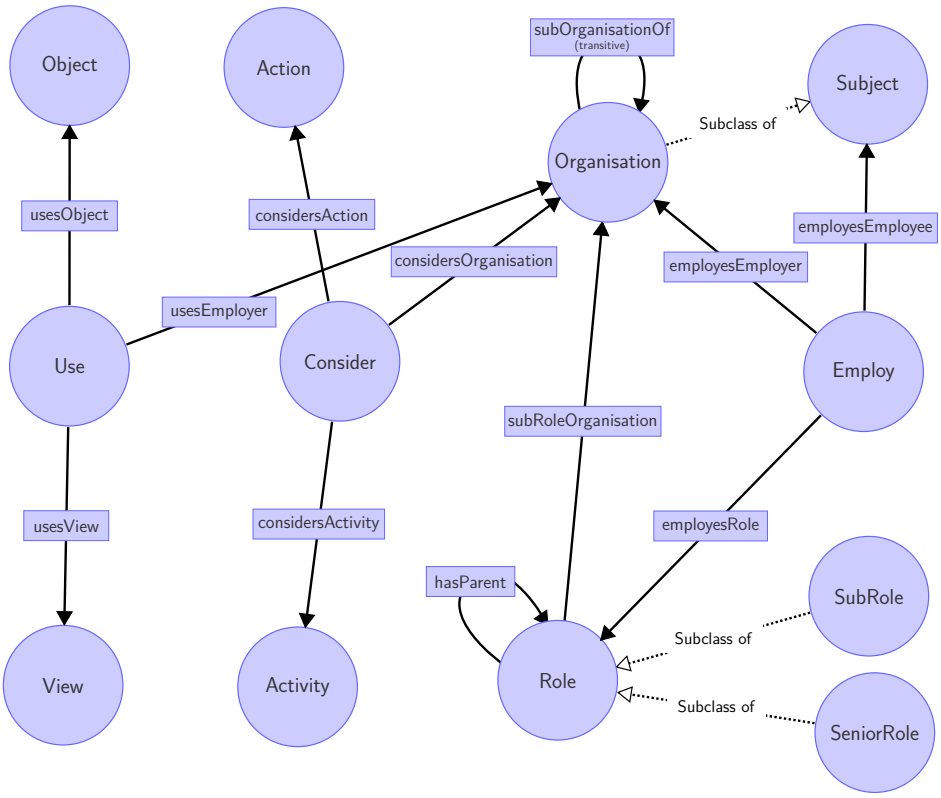


Figure 5.5: Representation of the connection rules in the OrBAC ontology

which means x is more preferred to y. Equivalence holds when both $\langle x$, orbac:isPreferredTo, $y\rangle$ and $\langle y$, orbac:isPreferredTo, $x\rangle$ hold and incomparability is the absence of both relations.

The following describes how preferences are automatically assigned to the connection rules. The main method proceeds by ordering abstract concepts (e.g. roles). The order is then propagated using the SWRL rules. Rules R1 and R2 in Table 5.1 are examples of how the ordering is propagated from orbac:Role individuals to orbac:Employ individuals, and from orbac:Context individuals to orbac:Define individuals, respectively. Another method relies on hierarchies to define preferences. Mainly, a sub-role and a senior-role are considered to be more preferred than their parent roles. In addition, rules defined in suborganisations have a higher preference. Rule R3 in Table 5.1 illustrates how to propagate the ordering from hierarchies, such that ?role1 and ?role2 are roles, if ?role2 is a parent role of ?role1, then ?role1 is preferred to ?role2. SWRL rules are executed efficiently using Drools [Proctor 2012].

Example 5.3. *Let us translate our Example's 5.1 to the* OrBAC *ontology. Figure 5.6 depicts the different example triples. In addition, Figure 5.7 depicts the relations used to represent the partial preorder over the uncertain rules of the policy.*

Derivation of privileges in the OrBAC ontology. The derivation of concrete privilege, which is given by Definition 5.2, is achieved using a SPARQL query specific for each access type, namely Is-permitted, Is-prohibited, Is-obliged, and Is-recommended.

Table 5.1: SWRL rules for order propagation

R1: orbac:Role(?role1) ∧ orbac:Role(?role2) ∧ orbac:isPreferredTo(?role1,?role2) ∧ orbac:employesRole(?employ1,?role1) ∧ orbac:employesRole(?employ2,?role2) → orbac:isPreferredTo(?employ1,?employ2)
R2: orbac:Context(?context1) ∧ orbac:Context(?context2) ∧ orbac:isPreferredTo(?context1, ?context2) ∧ orbac:definesContext(?define1, ?context1) ∧ orbac:definesContext(?define2, ?context2) → orbac:isPreferredTo(?define1, ?define2)
R3: orbac:hasParent(?role1, ?role2) → orbac:isPreferredTo(?role1, ?role2)

```
:perm1 rdf:type orbac:Permission.
:perm1 :accessTypeOrg :consortium.
:perm1 :accessTypeRole :coordinator.
:perm1 :accessTypeActivity :consult.
:perm1 :accessTypeView :data.
:perm1 :accessTypeContext :management.

:perm2 rdf:type orbac:Permission.
:perm2 :accessTypeOrg :univ1.
:perm2 :accessTypeRole :teamLeader.
:perm2 :accessTypeActivity :consult.
:perm2 :accessTypeView :data.
:perm2 :accessTypeContext :management.

:prohi1 rdf:type orbac:Prohibition.
:prohi1 :accessTypeOrg :consortium.
:prohi1 :accessTypeRole :staffMember.
:prohi1 :accessTypeActivity :consult.
:prohi1 :accessTypeView :internalData.
:prohi1 :accessTypeContext :default.

:univ1 :subOrganisationOf :consortium.
:inst1 :subOrganisationOf :consortium.

:cons1 rdf:type orbac:Consider.
:cons1 :considersOrg :consortium.
:cons1 :considersActivity :consult.
:cons1 :considersAction :read.

:emp1 rdf:type orbac:Employ.
:emp1 :employesEmployer :consortium.
:emp1 :employesRole :coordinator.
:emp1 :employesEmployee :Ben.
```

```
:emp2 rdf:type orbac:Employ.
:emp2 :employesEmployer :univ1.
:emp2 :employesRole :teamLeader.
:emp2 :employesEmployee :Ben.

:emp3 rdf:type orbac:Employ.
:emp3 :employesEmployer :inst1.
:emp3 :employesRole :staffMember.
:emp3 :employesEmployee :Ben.

:use1 rdf:type orbac:Use.
:use1 :usesEmployer :consortium.
:use1 :usesView :data.
:use1 :usesObject :dataset1.

:use2 rdf:type orbac:Use.
:use2 :usesEmployer :inst1.
:use2 :usesView :internalData.
:use2 :usesObject :dataset1.

:def1 rdf:type orbac:Define.
:def1 :definesOrg :univ1.
:def1 :definesContext :management.
:def1 :definesSubject :Ben.
:def1 :definesAction :read.
:def1 :definesObject :dataset1.

:def2 rdf:type orbac:Define.
:def2 :definesOrg :inst1.
:def2 :definesContext :default.
:def2 :definesSubject :Ben.
:def2 :definesAction :read.
:def2 :definesObject :dataset1.
```

Figure 5.6: OrBAC ontology example

The evaluation of the query Is-permitted is true when a concrete permission is derived for the given subject, action, and object. The derivation of the remaining privileges is done in the same way. Note that in the case of inconsistency, both the queries Is-permitted and Is-prohibited evaluate to true. The full used queries are provided in Appendix B.

```
:use1 :isPreferredTo :emp1.
:use2 :isPreferredTo :emp1.
:cons1 :isPreferredTo :emp1.

:emp1 :isPreferredTo :emp2.
:emp2 :isPreferredTo :emp3.
```

```
:use1 :isPreferredTo :def1.
:use2 :isPreferredTo :def1.
:cons1 :isPreferredTo :def1.


:def1 :isPreferredTo :def2.
```

Figure 5.7: Preferences of the OrBAC ontology example

5.2.3 Conflict-Resolution in OrBAC

Supports and Dominance in OrBAC

We define a conflict resolution approach that deals with conflicts on the concrete level. This means that potential conflicts between the abstract rules, as defined by [Cuppens *et al.* 2007] are tolerated in a policy. Clear conflicts between abstract rules, where both permission and prohibition relations are defined for the same abstract policy elements, are not allowed in the model by enforcing a disjointWith relation between the classes of permission and prohibition.

We deal with conflicts when an access query is posed to the system. This allows for less restrained modelling of a security policy. Moreover, it highlights the need for approaches that deal with query-answering from large-scale read-only datasets, where revision is impractical or even impossible. In the case of access control, this applies when the policy is not modifiable or when updating the policy is costly or time-consuming.

In order to resolve access conflicts, we introduce the acceptance method for OrBAC policies. The method is inspired by the "Dominant supports" characterisation, and is concerned with inconsistent partially ordered DL-Lite$_\mathcal{R}$ knowledge bases. Here, the method is applied locally by evaluating the supports of permission and the supports of prohibition for a given subject, action, and object, against each other. It requires each support of a prohibition to be dominated by a support of permission to grant access. The dominance relation is defined the same way as dominance is defined for partially preordered DL-Lite$_\mathcal{R}$ knowledge bases (Definition 2.12). In essence, it extends the partial order defined over the rules of the policy to a partial order defined over subsets of the policy. It consists of having, for each rule of the dominant set, at least one strictly less preferred rule in the dominated set.

The notion of support allows us to address the issue of access conflicts. A support for a privilege is the minimal rule set required to infer it. We distinguish between supports of permissions and supports of prohibitions for the sake of simplicity.

Definition 5.6 (Support)**.**

- *A support of the concrete permission* Is-permitted(s,α,o) is a minimal subset of rules required to derive it: s_{pm} = {Permission(org,r,a,v,c), Employ(org,s,r,w_i), Use(org,o,v,w_j), Consider(org,α,a,w_k), Define(org,s,α,o,c,w_l)}.
- A support of the concrete prohibition Is-prohibited(s,α,o) is a minimal subset of rules required to derive it: s_{ph} = {Prohibition(org,r,a,v,c), Employ(org,s,r,w_i), Use(org,o,v,w_j), Consider(org,α,a,w_k), Define(org,s,α,o,c,w_k)}.

Computing the supports of a privilege is achieved by executing a SPARQL query on the given policy represented by an OrBAC ontology, ensuring tractability. In essence, the SPARQL query identifies all minimal sets of rules that correspond to a valid derivation path, according to the semantics of the OrBAC policy. Thus, one privilege may have multiple supports. The queries used to compute supports are provided in Appendix B.

Example 5.4. *Consider the policy $\mathcal{O}_{\mathbb{L}}$ from Example 5.2. The permission* ls-permitted(Ben, read, dataset1) has two supports, and the prohibition ls-prohibited(Ben, read, dataset1) has one support. Namely:

$s_{pm_1} = \{$ Permission(consortium, coordinator, consult, data, management), Employ(consortium, coordinator, Ben, w_1), Consider(consortium, consult, read), Use(consortium, data, dataset1), Define(univ1, Ben, read, dataset1, management, w_4), subOrganisationOf(univ1, consortium)$\}$

$s_{pm_2} = \{$ Permission(univ1, teamLeader, consult, data, management), Employ(univ1, teamLeader, Ben, w_2), Consider(consortium, consult, read), Use(consortium, data, dataset1), Define(univ1, Ben, read, dataset1, management, w_4), subOrganisationOf(univ1, consortium)$\}$

$s_{ph_1} = \{$ Prohibition(consortium, staffMember, consult, internalData, default), Employ(inst1, staffMember, Ben,w_3), Consider(consortium, consult, read), Use(inst1, internalData, dataset1), Define(inst1, Ben, read, dataset1, default, w_5), subOrganisationOf(inst1, consortium)$\}$

Having addressed support computation, we now introduce the dominance relation in OrBAC policies.

Definition 5.7 (Dominance)**.** *Let $s_1 \subseteq \mathcal{U} \cup \mathcal{P}$ and $s_2 \subseteq \mathcal{U} \cup \mathcal{P}$ be subsets of rules from a policy $\mathcal{O}_{\mathbb{L}}$. We say that s_1 dominates s_2 if for all $T_1(\cdot, w_i) \in s_1$, there is $T_2(\cdot, w_j) \in s_2$ such that $w_i \rhd w_j$, where $T_1(\cdot, w_i)$, $T_2(\cdot, w_j)$ in $\mathcal{U}$.*

The acceptance method

In order to simplify uncertainty handling and reduce preference checks, we restrict supports to rules with weights other than $\mathbb{1}$, since $\mathbb{1}$ always dominates any other weight in $\mathbb{L}$ and does not require comparison. The acceptance method checks if a privilege is granted as follows.

Definition 5.8 (Acceptance)**.** *An access is accepted for a given subject s, with an action α on an object o, if:*

- *$\mathcal{O}_{\mathbb{L}} \models$ ls-permitted$(\mathsf{s}, \alpha, \mathsf{o})$ and $\mathcal{O}_{\mathbb{L}} \not\models$ ls-prohibited$(\mathsf{s}, \alpha, \mathsf{o})$.*
- *$\mathcal{O}_{\mathbb{L}} \models$ ls-permitted$(\mathsf{s}, \alpha, \mathsf{o})$ and $\mathcal{O}_{\mathbb{L}} \models$ ls-prohibited$(\mathsf{s}, \alpha, \mathsf{o})$, and for all s_{ph} supports of prohibition for $(\mathsf{s}, \alpha, \mathsf{o})$, there is s_{pm} a support of permission for $(\mathsf{s}, \alpha, \mathsf{o})$, such that s_{pm} dominates s_{ph}, as per Definition 5.7.*

Otherwise, access is not accepted.

Example 5.5. *Let us apply the acceptance method to the policy $\mathcal{O}_{\mathbb{L}}$ of Example 5.2. The support of the prohibition s_{ph_1} must be dominated by at least one permission support for the permission to be accepted. Applying dominance (Definition 5.7), s_{pm_1} dominates s_{ph_1} because $w_1 \rhd w_3$ and $w_4 \rhd w_5$. Therefore, the permission* ls-permitted(Ben, read, dataset1) *is accepted.*

In the OrBAC ontology, acceptance checking can be achieved using a single SPARQL query, without computing the supports of permission and prohibition and then comparing their preference. The query returns false if no permission is inferred. Otherwise, it returns true if no prohibition exists. If both permission and prohibition exist, the query returns true if for each prohibition there is a permission that dominates it. Otherwise, it returns false. The acceptance query is provided in Appendix B.

5.3 Explaining Access Control Decisions

The literature falls short when it comes to explaining access control security policies. The models are often limited to returning a binary response by granting or denying access. In this section, we equip the OrBAC model with explanation mechanisms. In Example 5.5, it is not enough to provide the OrBAC derivation and acceptance rules to explain why Ben is allowed to read dataset1. The problem is that even an OrBAC expert might be lost when the number of rules involved in making the decision is large and when conflicts of access are handled.

The literature on eXplainable AI (XAI) provides several explanation approaches depending on the type of system studied [Arya *et al.* 2019, Arrieta *et al.* 2020]. For instance, on the one hand, black-box models employ a post-hoc approach where an auxiliary method is used to explain a model after it has been trained. On the other hand, directly interpretable models are those that are understandable due to their transparent nature. The OrBAC model, along with the conflict resolution approach that extends it, belongs to this class of interpretable models. Indeed, decisions of OrBAC can be logically traced back to the rules that led to it. However, the model itself can be large, complex, and difficult to debug. In order to achieve our aim, two major questions must be answered.

The first one is what to explain? algorithms or results. In the case of algorithms, this means explaining how the system works by providing a full analysis of its properties for every possible input and outcome. The case of explaining the results focusses on answering the question: Why did the system produce a particular result for a particular input? In eXplainable AI (XAI), explaining the results is also called outcome or local explanation, whereas explaining the algorithms is called generating global explanations [Arya *et al.* 2019]. In both global and local cases, several other types of explanation can be used, including trace-based, counterfactual, contrastive, and scientific explanations [Chari *et al.* 2020]. The second question concerns who the explanation is aimed at and for what reason. Indeed, there are crucial differences when providing explanations to regular users or to administrators in access control. It often involves hiding a part of or the whole security policy to preserve it.

This section is divided into two parts. We start by formally defining the logical explanations of the decisions made in OrBAC in the presence and absence of conflicts. Then, we propose a general framework for generating and evaluating natural language explanations of access control decisions using large language models. The main contribution is an evaluation method that does not rely on reference texts or human annotations. This represents an original and practical solution to the challenge of assessing LLM-generated explanations. Using logical explanations and natural language inference (NLI) scores, our framework ensures clarity and accuracy while producing user-friendly outputs. Unlike existing work, which often depends on pre-defined reference explanations, our method offers an automated alternative. The implementation exploits the OrBAC ontology, which offers efficient reasoning and supports the evaluation process. In addition, we conduct a detailed experimental study using well-known language models under zero-shot, few-shots, and interactive prompting settings. To further clarify the roadmap of this section, Figure 5.8 provides a layered overview of our access control explanation framework.

5.3.1 Logical Explanations of OrBAC Policies

Explainability has been modestly addressed for access control security mechanisms. A body of work considered using explanations to detect conflicts. For instance, the authors in [Turkmen *et al.* 2013] introduced an SAT-based method that uses explanations to detect and resolve conflicts in physical access control policies. Although the representation differs from OrBAC, a conflict

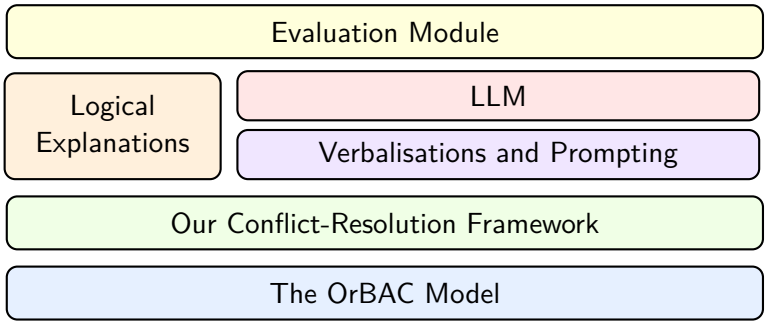


Figure 5.8: Multi-layer view of the proposed explanation framework.

is defined in the same manner, and the explanations are defined at the SAT encoding level. Other research efforts addressed explanation as a method for anomaly detection in policy-based systems [Charalambides *et al.* 2005].

We propose a formal definition of explanations for the OrBAC model. We start by defining explanations in the case of a conflict-free decision, before moving to explanations when the decision is made in a conflicting situation using the acceptance method of Definition 5.8.

Definition 5.9 (Conflict-free explanation)**.**

- *If* $\mathcal{O}_{\mathbb{L}} \models \mathsf{Is\text{-}permitted}(\mathsf{s}, \alpha, \mathsf{o})$ *and* $\mathcal{O} \not\models \mathsf{Is\text{-}prohibited}(\mathsf{s}, \alpha, \mathsf{o})$*, then a minimal explanation of the inference* $\mathcal{O}_{\mathbb{L}} \models \mathsf{Is\text{-}permitted}(\mathsf{s}, \alpha, \mathsf{o})$ *is one of its supports. The set of all the explanations is the set of all the supports* $\{s_{pm_1}, \ldots, s_{pm_n}\}$*.*

- *If* $\mathcal{O}_{\mathbb{L}} \models \mathsf{Is\text{-}prohibited}(\mathsf{s}, \alpha, \mathsf{o})$ *and* $\mathcal{O} \not\models \mathsf{Is\text{-}permitted}(\mathsf{s}, \alpha, \mathsf{o})$*, then a minimal explanation of the inference* $\mathcal{O}_{\mathbb{L}} \models \mathsf{Is\text{-}prohibited}(\mathsf{s}, \alpha, \mathsf{o})$ *is one of its supports. The set of all the explanations is the set of all the supports* $\{s_{ph_1}, \ldots, s_{ph_m}\}$*.*

An explanation in a conflict-free situation, for a grant or a denial decision, is simply a support. An explanation in a conflict situation depends more on the rule used to resolve the conflicts. In our case, the method used is local, which means that only the supports of the concrete concepts examined are involved in the decision making. However, a minimal explanation in a conflict case depends on the decided outcome (accepted or not). On the one hand, each prohibition's support must have at least one dominating permission's support for the access to be accepted. On the other hand, it is enough to have one prohibition's support not dominated by any permission's support to reject the access.

Definition 5.10 (Conflict explanation)**.** *Let* $\{s_{pm_1}, \ldots, s_{pm_n}\}$ *be the non-empty set of supports of* $\mathsf{Is\text{-}permitted}(\mathsf{s}, \alpha, \mathsf{o})$ *and* $\{s_{ph_1}, \ldots, s_{ph_m}\}$ *be the non-empty set of supports of* $\mathsf{Is\text{-}prohibited}(\mathsf{s}, \alpha, \mathsf{o})$ *in* $\mathcal{O}_{\mathcal{L}}$*:*

- *If permission* $\mathsf{Is\text{-}permitted}(\mathsf{s}, \alpha, \mathsf{o})$ *is accepted, then a minimal explanation is a set of tuples* $\mathcal{E} = \{(s_{pm_i}, s_{ph_1}), \ldots, (s_{pm_j}, s_{ph_m})\}$*, with* $1 \leq i, j \leq n$*, and for each tuple* (s_{pm_i}, s_{ph_k})*, with* $1 \leq k \leq m$*,* s_{pm_i} *dominates* s_{ph_k}*.*

- *If permission* $\mathsf{Is\text{-}permitted}(\mathsf{s}, \alpha, \mathsf{o})$ *is not accepted, then a minimal explanation is a set of tuples* $\mathcal{E} = \{(s_{pm_1}, s_{ph_j}), \ldots, (s_{pm_n}, s_{ph_j})\}$ *such that* $1 \leq j \leq m$*, and for each tuple* (s_{pm_i}, s_{ph_j})*, with* $1 \leq i \leq n$*,* s_{pm_i} *does not dominate* s_{ph_j}*.*

Example 5.6. *We continue with Example 5.5. The acceptance of* ls-permitted(Ben, read, dataset1) has two minimal explanations, namely: $\mathcal{E}_1 = \{(s_{pm_1}, s_{ph_1})\}$ and $\mathcal{E}_2 = \{(s_{pm_2}, s_{ph_1})\}$. Indeed, s_{pm_1} dominates s_{ph_1} because $w_1 \rhd w_3$ and $w_4 \rhd w_5$, and s_{pm_2} dominates s_{ph_1} because $w_2 \rhd w_3$ and $w_4 \rhd w_5$.

The size of a minimal explanation (number of pairs (s_{pm_i}, s_{ph_j})) is equal to the number of prohibition supports $|\{s_{ph_i}\}_{i=1}^{m}| = m$ in the case of acceptance, and $|\{s_{pm_i}\}_{i=1}^{n}| = n$ in the case of non-acceptance. This yields the following results:

- **Worst-case acceptance explanations.** Assume that for all $(i,k) \in \{1,\ldots,n\}\times\{1,\ldots,m\}$, s_{pm_i} dominates s_{ph_k}, then the number of minimal explanations for the acceptance of the access query ls-permitted$(\mathsf{s},\alpha,\mathsf{o})$ is exactly n^m.
- **Worst-case non-acceptance explanations.** Assume that for all $(i,k) \in \{1,\ldots,n\} \times \{1,\ldots,m\}$, s_{pm_i} does not dominate s_{ph_k}, then the number of minimal explanations for the rejection of ls-permitted$(\mathsf{s},\alpha,\mathsf{o})$ is exactly m.

The logical explanations can be extracted via a SPARQL query, which structurally mirrors the acceptance query, but returns relevant support elements instead of a Boolean result.

5.3.2 On Explaining OrBAC Policies using LLMs

This section focusses mainly on how the explanations are presented to the users. Although logical explanations are correct, their structure is often unfamiliar to users and policy administrators, making it difficult to grasp the true reasons behind a decision. The goal is to offer explanations that are more understandable (or intelligible) and user-friendly, while maintaining clarity and accuracy.

Large language models (LLMs) are powerful tools for natural language generation. In this work, we explore their use in generating explanations for OrBAC decisions. This task poses several challenges, including how policy elements are presented to the model, how the generated explanations can be evaluated, what evaluation criteria should be used, and how to effectively prompt the LLM. In addition, our aim is to answer the following research questions:

- How well can language models recognise the logical explanations involved in the reasoning process and properly explain the decision made out of conflicts?
- How to evaluate the generated explanations without reference texts?
- How can the answers be improved via interactions with the LLM? What is the number of interactions needed to get a satisfactory answer?

Verbalisation and prompting

To properly provide the elements required for explaining an acceptance or rejection decision to an LLM, we rely on "back-off" templates to verbalise the rules involved in the decision-making process. This approach aims to reduce the risk of hallucinations and biases that may arise when providing raw logical expressions or when using pre-trained fill-mask models like RoBERTa [Liu *et al.* 2019].

The templates, summarised in Table 5.2, convert OrBAC rules into natural language assertions, following established practices for verbalising logical expressions [Li *et al.* 2019]. This

Rule	Verbalisation
Permission(org,r,a,v,c) / Prohibition(org,r,a,v,c)	The organisation **org** grants the role **r** the **permission/prohibition** to perform the activity **a** on the view **v** if the context **c** holds
Employ(org,s,r,w_1)	The organisation **org** employs the **subject** s in the role **r**
Use(org,o,v,w_2)	The organisation **org** uses the **object** o in the view **v**
Consider(org,α,a,w_3)	The organisation **org** considers the **action** α as an **a** activity
Define(org,s,α,o,c,w_4)	The context **c** holds between the **subject** s, the **action** α, and the **object** o in the organisation **org**
subOrganisationOf(org_1, org_2)	The organisation org_1 is a sub-organisation of the organisation org_2

Table 5.2: Verbalisation templates of OrBAC rules

method aligns with the way LLMs learn token meaning from contextual usage. Each prompt includes: (1) the overall logic, (2) permission supports, (3) prohibition supports, (4) preferences, (5) the decision, and (6) a request for explanation. Due to input length limitations of language models, only elements relevant to an access query are included, rather than the full policy. The following Table 5.3 summarises an example of the prompt used for explaining the access of Ben to read dataset1 from our OrBAC running example.

Evaluation metrics

The primary challenge for natural language generation is the evaluation process. In the literature, different evaluation strategies were employed. Human feedback is usually the main contender. Grammar-Based Metrics (GBM) [Novikova *et al.* 2017] since they assess an important aspect of NLG and do not require reference texts. In addition to the metrics that compare natural text generations with reference-texts, either generated by humans or well-established generation methods. In this type, we find word overlap metrics (WOMs) [Novikova *et al.* 2017], semantic similarity and embedding metrics [Clinciu *et al.* 2021, Novikova *et al.* 2017], and natural language entailment confidence scores [Dušek & Kasner 2020]. The latter is the best at understanding the context and semantics of a sentence.

The key challenge in our case is the absence of textual reference datasets to evaluate the generated explanations. High-quality references are often costly and difficult to produce at scale, especially when specific complex logical rules are involved. To address this, we leverage the available structured logical explanations as an alternative form of reference. We decompose these logical explanations into minimal components (i.e., individual support elements), verbalise them into natural language, and treat them as hypotheses in a natural language inference (NLI) setup. The LLM-generated explanation is used as the premise. The resulting scores are then aggregated to assess the quality of the generated explanation. Figure 5.9 illustrates how natural language inference models work.

Table 5.4 summarises the proposed evaluation metrics, each designed to assess a specific aspect of the generated explanation. Coverage and completeness measure the model's ability to capture all elements of a minimal explanation and to retrieve all relevant logical explanations, respectively. Correctness verifies whether the model predicts the correct decision outcome, while

Overall Logic	Explain the condition under which access is granted. Access is granted if and only if, for each support of a prohibition, there exists a corresponding support for a permission where the permission's support dominates the prohibition's. Dominance means that each element of the support of the permission is strictly preferred to at least one element of the support of the prohibition. Here are the different elements which compose the decision:
Permission Supports	List of the elements that compose the permission support: (support 1): The organisation consortium grants the role coordinator the Permission to perform the activity consult on the view data if the context management holds. The organisation consortium employs Ben in the role coordinator. The organisation consortium considers read as a consult activity. The organisation consortium uses dataset1 in the view data. The context management holds between Ben, read, and dataset1 in the organisation univ1. univ1 is a sub-organisation of consortium. (support 2): The organisation univ1 grants the role teamLeader the Permission to perform the activity consult on the view data if the context management holds. The organisation univ1 employs Ben in the role teamLeader. The organisation consortium considers read as a consult activity. The organisation consortium uses dataset1 in the view data. The context management holds between Ben, read, and dataset1 in the organisation univ1. univ1 is a sub-organisation of consortium.
Prohibition Supports	List of the elements composing the prohibition support: (support 1): The organisation consortium grants the role staffMember the Prohibition to perform the activity consult on the view internalData if the context default holds. The organisation inst1 employs Ben in the role staffMember. The organisation consortium considers read as a consult activity. The organisation inst1 uses dataset1 in the view internalData. The context default holds between Ben, read, and dataset1 in the organisation inst1. inst1 is a sub-organisation of consortium.
Preferences	The relation "The organisation consortium employs Ben in the role coordinator" is preferred to "The organisation inst1 employs Ben in the role staffMember" because: coordinator is preferred to staffMember. The relation "The organisation univ1 employs Ben in the role teamLeader." is preferred to "The organisation inst1 employs Ben in the role staffMember" because: teamLeader is preferred to staffMember. The relation "The context management holds between Ben, read, and dataset1 in the organisation univ1." is preferred to "The context default holds between Ben, read, and dataset1 in the organisation inst1." because: management is preferred to default.
Outcome decision	The outcome of the Logic inference is: the permission for Ben to perform read on dataset1 is granted.
Request for Explanation	Using the provided logic, supports, and preferences. Please explain the decision-making process (why the access was granted or denied). Please use the structured format below: 1. **Decision Rule**: [Provide the decision rule used for the decision-making process.] 2. **Outcome**: [Describe the decision outcome clearly and concisely.] 3. **Relations and Preferences**: [Explain the different relations and preferences that influenced the decision. Include all relevant information.] Make sure to follow this structure exactly and ensure each section is complete.

Table 5.3: A prompt example generated for the access of Ben to read dataset1.

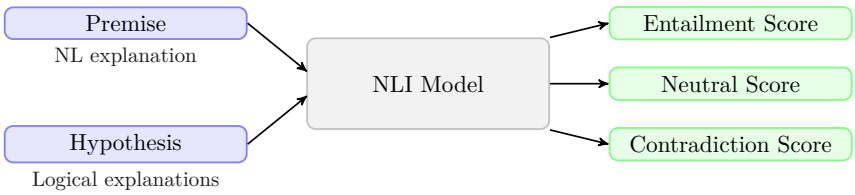


Figure 5.9: Natural Language Entailment score using the `bart-large-mnli` model.

clarity evaluates how well the model communicates the reasoning process (e.g. acceptance and dominance rules). To compute these scores, we use a natural language inference (NLI) model, specifically the 'zero-shot classification' model 'bart-large-mnli' [Facebook AI 2020], which

outputs entailment, contradiction, and neutral scores [Dušek & Kasner 2020]. For coverage and completeness, verbalised components of the logical explanation are used as hypotheses, while segments of the generated explanation serve as premises. Correctness and clarity are evaluated by feeding the full generated explanation as a premise, with the expected outcome or rule verbalisation as the hypothesis. Furthermore, we follow the hallucination detection method from [Dušek & Kasner 2020], using the full set of known facts as a premise and checking whether the generated explanation contradicts them. Finally, grammatical accuracy is assessed using the language_tool_python grammar checker [Morris 2020].

Description	**Formulas** (acceptance case)
The maximum average NLI entailment score across all logical explanations $\mathcal{E}_k$, where each score is the entailment score of the verbalised support pairs (s_{pm_i}, s_{ph_j}) from the LLM's answer.	coverage : (where $\mathcal{E}_1, \ldots, \mathcal{E}_l$ are acceptance explanations) $\max_{1 \leq k \leq l} \left(\frac{1}{m} \sum_{(s_{pm_i}, s_{ph_j}) \in \mathcal{E}_k} \mathsf{NLI}(\mathsf{verb}(s_{pm_i}, s_{ph_j}), \mathsf{answer}) \right)$
The average NLI entailment score of all logical explanations $\mathcal{E}_k$, where each score is the entailment score of the verbalised support pairs (s_{pm_i}, s_{ph_j}) from the LLM's answer.	completeness : (where $\mathcal{E}_1, \ldots, \mathcal{E}_l$ are acceptance explanations) $\frac{1}{m \cdot l} \sum_{k=1}^{l} \left(\sum_{(s_{pm_i}, s_{ph_j}) \in \mathcal{E}_k} \mathsf{NLI}(\mathsf{verb}(s_{pm_i}, s_{ph_j}), \mathsf{answer}) \right)$
The NLI entailment score of the right outcome (granted or denied) from the LLM's answer.	correctness: $\mathsf{NLI}(\mathsf{verb}(\mathsf{outcome}), \mathsf{answer})$
The NLI entailment score of the acceptance method's description from the LLM's answer.	clarity : $\mathsf{NLI}(\mathsf{verb}(\mathsf{acceptance\text{-}method}), \mathsf{answer})$
The NLI contradiction score of the LLM's answer from all the facts examined in decision making.	hallucination : $\mathsf{NLI}(\mathsf{answer}, \mathsf{verb}(\mathsf{all\text{-}facts}))$

Table 5.4: Evaluation criteria for LLM-based explanations of OrBAC decisions

Experiments

Datasets and Configuration. To evaluate natural language explanations, two synthetic policy datasets were created: one, from which the running example is drawn, is based on a staff-exchange project, and another modelling a classical hospital policy inspired by prior OrBAC work [Kalam *et al.* 2003, Cuppens *et al.* 2007, Cuppens *et al.* 2004]. Both are provided as `.owl` ontology files instantiating the OrBAC ontology. The staff-exchange dataset includes 76 access conflicts, while the hospital dataset contains 50. The experiments used the Hugging Face Transformers library [Wolf *et al.* 2020] on dedicated GPU nodes (e.g. A40, L40S, H100, RTX A5000, V100) with at least 128 GB RAM, ensuring sufficient memory and acceleration

capabilities to support large-scale natural language explanation generation. The implementation relies on an OWL ontology to perform reasoning efficiently. The ontology facilitates prompting of the LLMs and allows integration with existing Semantic Web tools. Reasoning tasks are supported by RDFlib for querying and Drools for forward chaining inference.

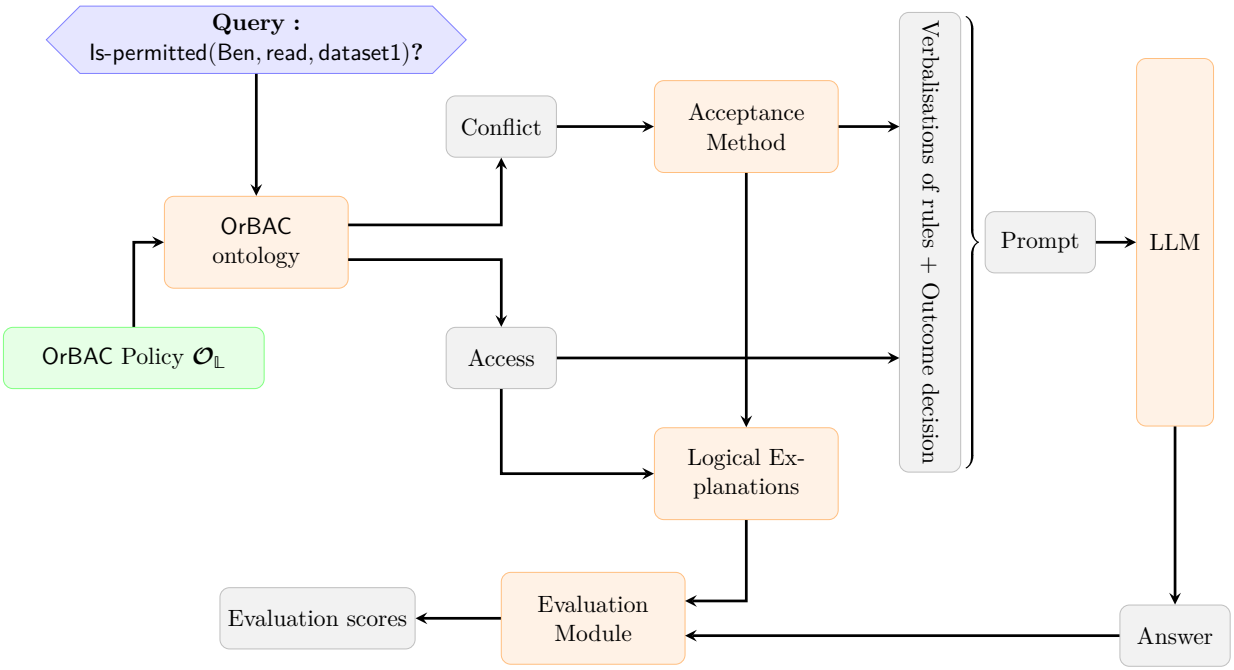


Figure 5.10: Framework for LLM-Generated Explanations of Access Control

Figure 5.10 presents our explanation framework. Access queries are evaluated using the OrBAC ontology and a policy, resulting in either an access decision (grant/deny) or a conflict, which is resolved via the acceptance method (Definition 5.8). In both cases, the verbalisations of the supports (Table 5.2) form a prompt for the LLM, while the logical explanations (Definitions 5.9 and 5.10) are generated in parallel. The evaluation module compares the LLM output with the logical explanation using our criteria (Table 5.4). This framework is used in the experimental study.

Experimental setup. We designed three different levels of experiments:

1. Evaluation of the average performance of explanations per model. This step gives us insight into the models that perform best in the zero-shot explanation generation task. The results are summarised in Table 5.5.
2. Evaluation of the models' ability to improve their results after receiving a follow-up prompt. We used a quality threshold of (0.6) for the values of the three main metrics: coverage, correctness, and clarity. The idea is to request a modification of the low-scoring parts of the explanation for up to four (4) interactions. The results of this evaluation are depicted in Figures 5.11 and 5.12.
3. We provide the models with an example of an explanation, in the style of a few-shot prompt, to compare their performance with the zero-shot task. Details of this part are given in Table 5.6.

Results and analysis. We used six language models, including base versions and, when available, their instruction-tuned (IT) variants. All are generative pre-trained models ranging from 1.5B to 9B parameters. The models and their average zero-shot explanation results are shown in Table 5.5.

Model	Coverage	Completeness	Correctness	Clarity	Hallucination	Grammar
deepseek-r1:1.5B	0.28	0.24	0.56	0.88	**0.18**	**0.98**
deepseek-r1:7B	0.49	0.42	0.86	0.86	0.26	0.95
gemma-2-9B	0.53	0.45	0.89	0.84	0.35	0.97
gemma-2-9B(IT)	0.46	0.37	**0.92**	0.92	0.38	0.95
llama-3.2-3B	**0.68**	**0.58**	0.87	0.75	0.46	0.9
llama-3.2-3B(IT)	0.61	0.52	0.84	**0.94**	0.2	0.94

Table 5.5: Average metric score for the zero-shot task per model

We focus on coverage scores, ensuring that at least one minimal logical explanation is met by the LLM-explanation. Other metrics prevent edge cases: correctness (right explanation but wrong outcome), clarity (whether the inference rule is clear), and hallucination (irrelevant text). The completeness score is informative, showing us how many of the possible explanations are captured.

A particular observation for the zero-shot experiment is that base models outperformed their instruction-tuned variants in coverage and completeness. Moreover, only two of the six prompted models averaged more than 0.6 coverage score, and, as expected, completeness averaged lower values than coverage. All models except 'deepseek-r1:1.5B' scored above 0.8 in correctness. Moreover, when it comes to clearly stating the inference rule used (clarity) and the grammar score, all the models averaged a good to excellent score.

In the zero-shot setting, the best performer was 'llama-3.2-3B', with top average coverage and solid correctness and clarity, despite higher hallucination scores, which remained below concerning levels. The zero-shot setting shows how well models capture elements of a minimal explanation. The two remaining settings aim to improve the obtained results. First, by highlighting the parts that need improvement in second chances. Then, simply using a few-shots setting, where an example of an explanation is provided to the models from the beginning.

In the interaction-based experiment, the model is given a history of interactions with follow-up sentences highlighting areas to improve: explanation details (supports and preferences) if coverage fails, the right outcome if correctness fails, and the inference rule if clarity fails. The aim is to answer the following questions:

- How many explanations were successful after each interaction?
- What is the average (coverage) score change across interactions? In other words, do interactions improve the coverage score? and to what extent?
- What percentage of retried explanations met the quality threshold (0.6), per metric, after each interaction?

Figure 5.11 shows that instruction-tuned variants outperform base models. 'gemma-2-9B(IT)' matches its base counterpart after three interactions. 'llama-3.2-3B(it)' overtakes its base version by the second interaction. Multiple rounds also let 'deepseek-r1:7B' reach performance comparable to the two 'gemma-2' models.

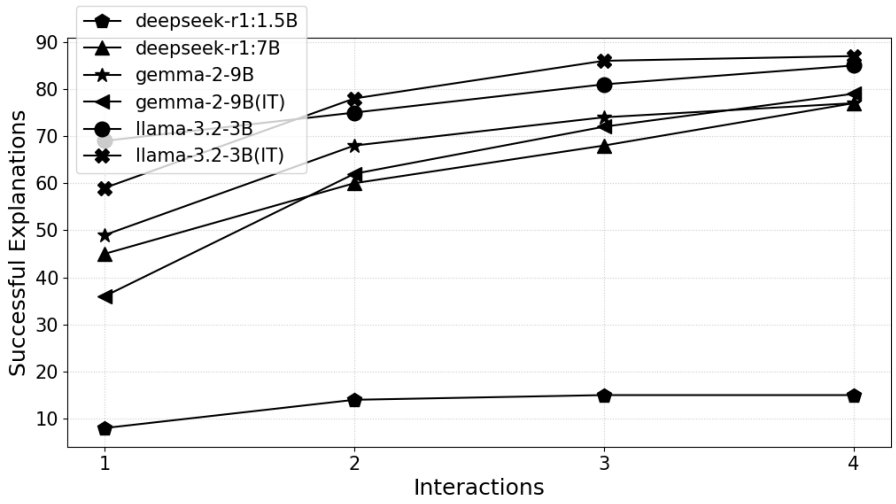


Figure 5.11: Number of successful explanations per interaction (threshold= 0.6)

To evaluate score differences between interactions, we compared the score changes of retried explanations between steps n and $n-1$ (Figure 5.12a). A notable improvement in average coverage appears after the first interaction ($\Delta(1 \rightarrow 2)$). In contrast, correctness and clarity scores tend to drop, which is expected given their already high zero-shot values. Note that in the multi-interaction process, the entire answer is re-evaluated. As a result, interactions might lower a previously high metric, as reflected by the negative change values in Figure 5.12.

Another way to assess change is by measuring the percentage of re-tried explanations meeting the quality threshold. Like the average score change, negative correctness and clarity changes appear across most models (Figure 5.12b). For coverage, the target of re-prompting, over 30% of failed cases became successful after the first retry with 'gemma-2-9B(IT)'. Around 20% improvement was observed for 'deepseek-r1:7B', 'gemma-2-9B', and 'llama-3.2-3B(IT)'. Surprisingly, the top performer, 'llama-3.2-3B', showed no improvement in coverage and only a slight gain in successful retries. Overall, the first re-prompting improved average coverage and is worth applying, though further retries had limited effect.

The final analysis involves few-shot prompting. Zero-shot prompts averaged around 1250 tokens in size, while few-shot prompts averaged 2080, due to the inclusion of a full explanation example. Larger prompts are harder for smaller models to process, even within their context window, reflected by the drop in scores for 'deepseek-r1:1.5B' (Table 5.6). 'deepseek-r1:7B' showed slight improvements across all metrics. Instruction-tuned variants outperformed base models in few-shot settings. The top scorer was 'gemma-2-9B(IT)', with its coverage rising from 0.46 to 0.65 when given an example. 'llama-3.2-3B(IT)' retained similar performance to its zero-shot results. Both 'gemma-2-9B' and 'llama-3.2-3B' had lower coverage and completeness scores than in the zero-shot task.

Discussion. The experimental results provide further insight into the usefulness of employing logical explanations to evaluate LLM-generated outputs. It represents a novel alternative to traditional evaluation approaches, where human annotators or pre-existing reference texts must be used in NLG evaluation. Furthermore, the obtained results show how structured guidance via interactive and few-shots prompting has the ability to enhance explanations quality. The tested

(a) Average score change of retried explanations per interaction

(b) Percentage difference of retried explanations per interaction.

Figure 5.12: Impact of interaction-based prompting per model

Model	Coverage	Completeness	Correctness	Clarity	Hallucination	Grammar
deepseek-r1:1.5B	0.22	0.19	0.46	0.85	**0.17**	0.97
deepseek-r1:7B	0.51	0.44	0.69	0.87	0.29	0.93
gemma-2-9B	0.45	0.37	0.75	0.84	0.37	**0.98**
gemma-2-9B(IT)	**0.65**	**0.58**	**0.94**	**0.88**	0.33	0.95
llama-3.2-3B	0.31	0.27	0.59	0.65	0.42	0.95
llama-3.2-3B(IT)	0.59	0.5	0.89	0.86	0.37	0.9

Table 5.6: Average metric score for the few-shots task per model.

settings emphasise two main results. Few-shot prompting improves results for larger models but worsens smaller ones, suggesting a trade-off between prompt complexity and model capacity, and interactive prompting showed early gains but limited marginal improvements beyond the first retry. The method shows promise for larger use in making complex reasoning decisions more accessible.

5.4 Conclusion

In this chapter, we extend the study of inconsistency-tolerant semantics to the domain of access control. We introduced a conflict resolution method that addresses the contradictions arising from simultaneous permissions and prohibitions. Unlike existing approaches that constrain the representation to prevent conflicts, our method leverages the principle of dominant supports, ensuring that permission supports prevail over prohibition supports in the presence of conflict. This approach provides a flexible and semantically grounded alternative to restrictive strategies while preserving the expressive power of access control models.

Beyond conflict resolution, we also tackled the problem of explaining access control decisions in natural language. Building on formal logical explanations, we proposed a methodology that integrates Large Language Models (LLMs) to generate user-friendly textual explanations. Our approach couples logical explanations with natural language entailment measures to automatically evaluate explanation quality, including aspects such as hallucination and grammatical correctness. Experimental results with widely used language models demonstrated the promise of this method, particularly under zero-shot and few-shot settings, and highlighted the effectiveness of interactive prompting in refining explanation quality.

Together, these contributions illustrate how inconsistency-tolerant reasoning can enhance both the robustness and the transparency of access control systems, providing a foundation for future work on explainable and reliable policy enforcement.

Conclusion and perspectives

Summary of contributions

This thesis tackles managing inconsistency in lightweight description logics, when the data are partially ordered, with the purpose of defining tractable methods. The problem is approached by considering three aspects. The first is the choice of the representation language. Some languages provide desirable expressive power, but this may lead to higher reasoning complexity. Another important aspect is the choice of semantics. The literature offers a wide spectrum of well-established options with different levels of caution and computational cost. Finally, particular attention is given to the treatment of partial orders. This is especially challenging because, unlike in fully ordered information, it is not always possible to determine whether one piece of information is preferred over another.

Description logics have emerged as attractive representation languages for data on the semantic web. However, they are hampered by their high complexity of reasoning. This gave birth to lightweight description logics (DL-Lite), in which queries are guaranteed to map into the family of select-from-where SQL queries. In this kind of representation, a separation is made between the conceptual (TBox) and the factual (ABox) knowledge; an ABox may be inconsistent and is queried with respect to a well-debugged and consistent TBox. Moreover, special focus was given to approaches that restore consistency of an ABox with respect to a TBox by considering one or multiple of its subsets. In such a context, a repair is defined as a maximal consistent subset of the ABox, and the approaches are referred to by inconsistency-tolerant semantics.

We propose using two levers in order to extend the inference under the classical notion of repair. The first involves computing the deductive closure of the ABox, which derives all the facts entailed by the positive axioms of the TBox. This allows for the recovery of conclusions that are not explicitly present in all repairs. Deductive closure must be used with caution, as it could allow the inference of unwanted conclusions, such as in ICAR semantics [Lembo *et al.* 2010], or increase the complexity of the method, such as in ICR semantics [Bienvenu 2012b]. The second lever involves the widely used strategy of assigning priorities to handle inconsistency. The idea is to resolve conflicts by preferring certain assertions, typically via a total or partial preorder, enabling a more informed selection of consistent subsets.

Our first contribution involves the extension of the possibilistic semantics to partially ordered DL-Lite$_\mathcal{R}$ knowledge bases while applying a deductive closure operation in the process. Partial orders are fundamentally handled by considering a family of extending total orders that preserve the strict ordering and equivalence of a partial order and transform incomparabilities into one of the total order relations. Such an extension yields as many compatible totally ordered ABoxes, for which a possibilistic repair is computed. Then, the possibilistic repairs are deductively closed, before intersecting their closures to obtain a single repair for a partially ordered ABox. The main results of this application involve the definition of the Cπ-repair method, which computes a repair that is larger than its unclosed version.

In addition, a tractable characterisation is established for the method via the proposition of the notions of support and dominance. In essence, the "Dominant supports" characterisation evaluates the supports of an assertion against inclusion-minimal inconsistent subsets of the ABox, also called conflicts. Namely, each conflict of the ABox must be dominated by at least one support of the assertion, for this latter to be accepted in the repair.

The "Dominant supports" characterisation is then evaluated at the algorithmic and experimental levels. The algorithmic study allowed us to refine the definition by considering the most preferred sets of supports and conflicts interns of the used partial order, then gradually computing the repair through the computation of the π-accepted assertions first, which result from the unclosed version of the repair and are computed more efficiently. In the experimental

evaluation, we evaluated the impact of data size (ABox size), data quality (number of conflicts), and different configurations of partial orders on the computation of the Cπ-repair. We have mainly shown the benefits of employing the refined version of the characterisation compared to naively applying the algorithm.

Our quest for tractable characterisations was then extended thanks to the study of applying the deductive closure directly to partially ordered information. In the spirit of ICAR semantics [Lembo *et al.* 2010], we proposed the partially preordered deductive closure, which extends the deductive closure to be applied to partially preordered ABoxes. The new closure operator takes into account every level of reliability related to the entailment of an assertion. Then, it reduces the computation of the Cπ-repair to the most basic reasoning task, which is consistency checking. Intuitively, it evaluates the consistency of the set of assertions that have reliability levels that are not strictly less certain than the levels of reliability associated with the inference of a given assertion from the partially preordered deductive closure of the inconsistent ABox. Unlike previous approaches, such as ICAR and ICR semantics, the early application of deductive closure does not introduce unwanted consequences or degrade the quality of the repairs. In contrast, it allows us to preserve tractability by reducing the problem to consistency checks over subsets of the closed ABox. Experimental evaluation showed the efficiency of this method particularly through comparisons with the "Dominant supports" characterisation that rely on exhaustive searches over the conflict sets of the ABox and the support sets of assertions.

In order to finalise the study around the Cπ-repair method, we propose going beyond it in three different ways. The first is the study of unconditional and conditional rationality properties for the Cπ-repair method and the unclosed version, namely the π-repair method. Both methods satisfy the unconditional properties and the System P rules. If these propositions seem natural, even minimal, some inconsistency-tolerant semantics do not always satisfy them. For example, the so-called majority semantics (a query is valid if it is obtained from the majority of the repairs of an inconsistent ABox) does not satisfy these minimal properties. More precisely, in [Baget *et al.* 2016], majority-based inference has been shown to not satisfy the Cut, CM, and AND properties, even in DL-Lite$_{\mathcal{R}}$. Another example in which System P is not satisfied is existential inference, where a query is valid if it follows from one repair. On the non-satisfaction of the rational monotony (RM) property, the result is expected if we draw a parallel with standard possibilistic propositional logic and with the properties of non-monotonic relations. Indeed, there is a representation theorem (KLM [Kraus *et al.* 1990]) that shows that any non-monotonic relation which satisfies System P and RM is necessarily representable by a total order on the set of interpretations of propositional logic.

The proposed method, i.e., the Cπ-repair, is applicable to richer representation languages, and its computation process is also applicable to other semantics. We have shown that through adaptation of the Cπ-repair and the dominant supports characterisation to the DL-Lite$_{\mathcal{R},\sqcap}$ language, which goes slightly beyond DL-Lite by introducing the conjunction in the TBox axioms. Although the representation level is easily adapted to DL-Lite$_{\mathcal{R},\sqcap}$, the application of the characterisation can be challenging at the computational level. This is because the tractability of the dominant supports method is guaranteed thanks to having conflict sets that are polynomial in the size of the ABox, which does not always hold for richer languages, such as DL-Lite$_{\mathcal{R},\sqcap}$. Furthermore, the proposed repair process can also be achieved by replacing the possibilistic semantics with the lexicographic preferred-based repair [Benferhat *et al.* 2015]. However, the same characterisations do not apply in this case. Moreover, even in the case of a total preorder on the ABox, the lexicographic preferred-based repair semantics is not tractable. Hence, finding a tractable characterisation when dealing with partially preordered ABoxes is not guaranteed.

The last chapter of this thesis was reserved for the application of an inconsistency-tolerant

semantics style method to resolve conflicts in access control models. In the area of access control, conflicts may occur due to allowing the expression on contradictory access types such as permission and prohibition. State-of-the-art solutions tend to restrict the representation by adding rules that constrain the model to avoid its evolution on a path that results in conflicts. We define a conflict resolution method that, in the presence of an access conflict, ensures the dominance of permission supports over prohibition supports, in the spirit of the dominant supports method.

Our last contribution consists of explaining access control decisions using Large Language Models (LLMs). In essence, we addressed the issue of providing natural language explanations for access control decisions. We described an automatic evaluation strategy for the generated explanations. Our method relies on formal logical explanations to provide confidence scores for the different parts of a textual explanation. Namely, these parts are used to obtain natural language entailment scores from the generated natural language explanation. We also evaluated the hallucination and grammar levels. Experiments with a set of some famous and relatively small language models showed very promising results. We evaluated the performance of these models on the provided task using zero-shot and few-shot settings. In addition, an interactive prompting strategy was established to improve the low-scoring parts of an explanation.

Perspectives

We conclude this dissertation by discussing possible directions for future work.

The natural extension of the work proposed in this thesis consists in studying the application of the possibilistic semantics to boarder representation languages. Indeed, the extension to richer languages we introduced in Chapter 4 was limited to DL-Lite$_{\mathcal{R},\sqcap}$ and to the first characterisation, which relies on the computation of sets of conflicts and supports. Meanwhile, the more recent characterisation, which relies on consistency checking, is more compelling and promising when it comes to richer languages. Future work involves adapting and applying that characterisation to the different members of the DL-Lite family.

Moreover, we have studied the application of different repair semantics, in particular the lexicographic repair, only on the representation level. A more detailed study is needed to see the performance of a the closure-based version of the lexicographic repair, in addition to other semantics, such as the linear repair [Nebel 1994], the preferred-repair semantics [Benferhat *et al.* 2015], and the Elect method [Belabbes *et al.* 2021], particularly when dealing with partially ordered information and comparing them with the possibilistic repairs (Cπ-repair and π-repair). Furthermore, we plan to extend the application of our method to related frameworks, such as the one of databases, where inconsistency arises from the violation of database dependencies and integrity constraints [Wijsen 2019].

Another compelling direction for future work consists in studying the behaviour of our inconsistency-tolerant semantics, and related ones, in the presence of a consequence (or set of assertions) that should be removed. Several recent works have explored this direction in the context of inconsistent Quantified $\mathcal{EL}$ ABoxes [Baader *et al.* 2024, Baader *et al.* 2022]. Our study of behaviour focuses on conditional inference in the presence of fully certain new information, which represents one direction of evolution in ABoxes; it would be interesting to also investigate how our repair method performs in the opposite direction, i.e., when the presented information must be removed.

One of the main future work directions of the access control part consists in studying further types of explanations for access control decisions and further frameworks of reasoning with inconsistency. In fact, the presented work constitutes a first step into the field of eXplainable

AI, in particular taking advantage of the capacities of large language models and using them as explainers (or helpers), when decisions are made out of complex systems and rich knowledge. In addition, the literature provides several types of explanation, including trace-based, counterfactual, contrastive, and scientific explanations [Chari *et al.* 2020]. In this thesis, we give a special focus to trace-based explanations and their utility in defining evaluation procedures for natural language explanations generated with large language models. Meanwhile, other explanations, such as counterfactuals, can also be useful when decisions are made out of conflicts. Moreover, the presented framework for generating natural language explanations can also be extended to inconsistency-tolerant semantics. However, further adaptions are needed, since the involved rules (TBox axioms) are often more varied and complex than OrBAC rules.

The final perspective considered in this thesis is of an applicative nature. It consists of extending the approaches developed for inconsistency management to the domain of geographic information systems (GIS) for waste water networks. In such systems, each component, for example, manholes, pipes, pumps, is represented in independent databases, which store information about the objects and their attributes. For pipes, these attributes may include diameter, material, installation date, and so on. In the Starwars project (www.cril.fr/starwars), several GISs were integrated, showing that combining these databases (e.g. in Shapefile format) results in inconsistencies with respect to business rules. For instance, some rules require that each pipe end be connected to a manhole, yet the merged databases revealed pipes with unconnected ends, isolated manholes, or manholes positioned very close to each other. These inconsistencies indicate that the databases do not always comply with business rules. Future work will aim at applying or adapting the approaches proposed in this thesis to such contexts, enabling the automatic detection and management of these inconsistencies.

Bibliography

[Abou El Kalam & Deswarte 2006] Anas Abou El Kalam and Yves Deswarte. « Multi-orbac: A new access control model for distributed, heterogeneous and collaborative systems ». In *Proceedings of the IEEE Symposium on Systems and Information Security*, 2006.

[Adelin *et al.* 2022] Rémi Adelin, Cyrius Nugier, Éric Alata, Vincent Nicomette, Vincent Migliore, and Mohamed Kaâniche. « Facing emerging challenges in connected vehicles: a formally proven, legislation compliant, and post-quantum ready security protocol ». *Journal of Computer Virology and Hacking Techniques*, 18(4):425–452, 2022.

[Al-Kahtani & Sandhu 2004] Mohammad A Al-Kahtani and Ravi Sandhu. « Rule-based RBAC with negative authorization ». In *20th Annual Computer Security Applications Conference*, pages 405–415. IEEE, 2004.

[Alviano *et al.* 2023] Mario Alviano, Laura Giordano, and Daniele Theseider Dupré. « Complexity and scalability of defeasible reasoning with typicality in many-valued weighted knowledge bases ». In *European Conference on Logics in Artificial Intelligence*, pages 481–497. Springer, 2023.

[Ameer *et al.* 2022] Safwa Ameer, James Benson, and Ravi Sandhu. « Hybrid approaches (ABAC and RBAC) toward secure access control in smart home IoT ». *IEEE transactions on dependable and secure computing*, 20(5):4032–4051, 2022.

[Amgoud & Vesic 2014] Leila Amgoud and Srdjan Vesic. « Rich preference-based argumentation frameworks ». *International Journal of Approximate Reasoning*, 55(2):585–606, 2014.

[Anderson & Belnap 1975] A.R. Anderson and N.D. Belnap. « Entailment: The Logic of Relevance and Neccessity, Vol. I », 1975.

[Anton *et al.* 2024] Alba Martinez Anton, Clara Bertolissi, and Jean-Marc Talbot. « Solving Access Control Conflicts in Multi-User Systems ». In Sabrina De Capitani di Vimercati and Pierangela Samarati, editors, *Proceedings of the 21st International Conference on Security and Cryptography, SE-*

CRYPT 2024, Dijon, France, July 8-10, 2024, pages 492–499. SCITEPRESS, 2024.

[Arenas *et al.* 1999] M. Arenas, E. Leopoldo. Bertossi, and J. Chomicki. « Consistent Query Answers in Inconsistent Databases ». In *Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Philadelphia, Pennsylvania, USA*, pages 68–79, 1999.

[Arieli *et al.* 2011] Ofer Arieli, Arnon Avron, and Anna Zamansky. « Maximal and premaximal paraconsistency in the framework of three-valued semantics ». *Studia Logica*, 97:31–60, 2011.

[Arieli *et al.* 2024] Ofer Arieli, Kees van Berkel, Badran Raddaoui, and Christian Strasser. « Deontic Reasoning Based on Inconsistency Measures ». In *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning*, pages 71–81, 8 2024.

[Arrieta *et al.* 2020] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, and others. « Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI ». *Information fusion*, 58:82–115, 2020.

[Artale *et al.* 2009] A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. « The DL-Lite Family and Relations ». *Journal of Artificial Intelligence Research (JAIR)*, 36:1–69, 2009.

[Arya *et al.* 2019] Vijay Arya, Rachel KE Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C Hoffman, Stephanie Houde, Q Vera Liao, Ronny Luss, Aleksandra Mojsilović, and others. « One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques ». *arXiv preprint arXiv:1909.03012*, 2019.

[Association 2025] DBpedia Association. « DBpedia Ontology (DBO) — DBpedia Association ». https://www.dbpedia.org/resources/ontology/, 2025. Accessed: 2025-10-03.

[Baader *et al.* 2007] 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, 2 edition, 2007.

[Baader *et al.* 2017] Franz Baader, Ian Horrocks, Carsten Lutz, and Uli Sattler. *An introduction to description logic*. Cambridge University Press, 2017.

[Baader *et al.* 2019] Franz Baader, Andreas Ecke, Gabriele Kern-Isberner, and Marco Wilhelm. « The Complexity of the Consistency Problem in the Probabilistic Description Logic ». In *International Symposium on Frontiers of Combining Systems*, pages 167–184. Springer, 2019.

[Baader *et al.* 2020] Franz Baader, Francesco Kriegel, Adrian Nuradiansyah, and Rafael Peñaloza. « Computing compliant anonymisations of quantified ABoxes wrt EL policies ». In *International Semantic Web Conference*, pages 3–20. Springer, 2020.

[Baader *et al.* 2021] Franz Baader, Patrick Koopmann, Francesco Kriegel, and Adrian Nuradiansyah. « Computing Optimal Repairs of Quantified ABoxes wrt Static EL TBoxes. ». In *CADE*, pages 309–326, 2021.

[Baader *et al.* 2022] Franz Baader, Francesco Kriegel, and Adrian Nuradiansyah. « Error-tolerant reasoning in the description logic EL based on optimal repairs ». In *International Joint Conference on Rules and Reasoning*, pages 227–243. Springer, 2022.

[Baader *et al.* 2024] Franz Baader, Francesco Kriegel, and Adrian Nuradiansyah. « Inconsistency-and Error-Tolerant Reasoning wrt Optimal Repairs of Ontologies ». In *International Symposium on Foundations of Information and Knowledge Systems*, pages 3–22. Springer, 2024.

[Baget *et al.* 2016] Jean-François Baget, Salem Benferhat, Zied Bouraoui, Madalina Croitoru, Marie-Laure Mugnier, Odile Papini, Swan Rocher, and Karim Tabia. « Inconsistency-Tolerant Query Answering: Rationality Properties and Computational Complexity Analysis ». In *European Conference Logics in Artificial Intelligence, JELIA*, volume 10021 of *LNCS*, pages 64–80, 2016.

[Banerjee *et al.* 2017] Mohua Banerjee, Didier Dubois, Lluis Godo, and Henri Prade. « On the relation between possibilistic logic and modal logics of belief and knowledge ». *Journal of Applied Non-Classical Logics*, 27(3-4):206–224, 2017.

[Baral *et al.* 1992] Chitta Baral, Sarit Kraus, Jack Minker, and Venkataram Subram Subrahmanian. « COMBINING KNOWLEDGE BASES CONSISTING OF FIRST-ORDER THEORIES ». *Computational intelligence*, 8(1):45–71, 1992.

[Barker 2009] Steve Barker. « The next 700 access control models or a unifying meta-model? ». In *Proceedings of the 14th ACM symposium on Access control models and technologies*, pages 187–196, 2009.

[Belabbes & Benferhat 2021] Sihem Belabbes and Salem Benferhat. « Computing a Possibility Theory Repair for Partially Preordered Inconsistent Ontologies ». *IEEE Transactions on Fuzzy Systems*, pages 1–10, 2021.

[Belabbes & Benferhat 2022] Sihem Belabbes and Salem Benferhat. « Characterizing the Possibilistic Repair for Inconsistent Partially Ordered Assertions ». In *Information Processing and Management of Uncertainty in Knowledge-Based Systems - 19th International Conference, IPMU 2022, Milan, Italy, Proceedings, Part II*, volume 1602, pages 652–666, 2022.

[Belabbes *et al.* 2021] Sihem Belabbes, Salem Benferhat, and Jan Chomicki. « Handling inconsistency in partially preordered ontologies: the Elect method ». *Journal of Logic and Computation*, 31(5):1356–1388, 2021.

[Bell 1973] David E Bell. « Secure computer systems: Mathematical foundations and model ». *Technical Report ESD-TR-73-278-1*, 1973.

[Benferhat & Bouraoui 2017] Salem Benferhat and Zied Bouraoui. « Min-based possibilistic DL-Lite ». *Journal of Logic and Computation*, 27(1):261–297, 2017.

[Benferhat & El Baida 2004] Salem Benferhat and Rania El Baida. « A prioritized-based approach to handling conflicts in access control ». In *16th IEEE International Conference on Tools with Artificial Intelligence*, pages 286–293. IEEE, 2004.

[Benferhat & Garcia 2002] Salem Benferhat and Laurent Garcia. « Handling Locally Stratified Inconsistent Knowledge Bases ». *Studia Logica*, 70(1):77–104, 2002.

[Benferhat *et al.* 1993] Salem Benferhat, Didier Dubois, and Henri Prade. « Argumentative inference in uncertain and inconsistent knowledge bases ». In *Uncertainty in Artificial Intelligence*, pages 411–419. Elsevier, 1993.

[Benferhat *et al.* 1997] Salem Benferhat, Didier Dubois, and Henri Prade. « Some syntactic approaches to the handling of inconsistent knowledge bases: A comparative study part 1: The flat case ». *Studia Logica*, 58:17–45, 1997.

[Benferhat *et al.* 1998] Salem Benferhat, Didier Dubois, and Henri Prade. « Some syntactic approaches to the handling of inconsistent knowledge bases: A comparative study part 2: The prioritized case ». *Logic at work: essays dedicated to the memory of Helena Rasiowa*, 24(Chapter 26):473–511, 1998.

[Benferhat *et al.* 2003] Salem Benferhat, Rania El Baida, and Frédéric Cuppens. « A stratification-based approach for handling conflicts in access control ». In *Proceedings of the eighth ACM symposium on Access control models and technologies*, pages 189–195, 2003.

[Benferhat *et al.* 2004a] Salem Benferhat, Sylvain Lagrue, and Odile Papini. « Reasoning with partially ordered information in a possibilistic logic framework ». *Fuzzy Sets and Systems*, 144(1):25–41, 2004.

[Benferhat *et al.* 2004b] Salem Benferhat, Sylvain Lagrue, and Odile Papini. « Reasoning with partially ordered information in a possibilistic logic framework ». *Fuzzy Sets and Systems*, 144(1):25–41, 2004.

[Benferhat *et al.* 2011] Salem Benferhat, Julien Hué, Sylvain Lagrue, and Julien Rossit. « Interval-based possibilistic logic ». In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, Page 750–755. AAAI Press, 2011.

[Benferhat *et al.* 2015] Salem Benferhat, Zied Bouraoui, and Karim Tabia. « How to Select One Preferred Assertional-Based Repair from Inconsistent and Prioritized DL-Lite Knowledge Bases? ». In *International Joint Conference on Artificial Intelligence (IJCAI), Buenos Aires, Argentina*, pages 1450–1456, 2015.

[Bertolissi & Fernández 2022] Clara Bertolissi and Maribel Fernández. « Modular composition of access control policies: a framework to build multi-site multi-level combinations ». In *Proceedings of the 27th ACM on Symposium on Access Control Models and Technologies*, pages 7–18, 2022.

[Bertolissi *et al.* 2020] Clara Bertolissi, Maribel Fernández, and Bhavani Thuraisingham. « Admin-CBAC: An Administration Model for Category-Based Access Control ». In Vassil Roussev, Bhavani Thuraisingham, Barbara Carminati, and Murat Kantarcioglu, editors, *CODASPY '20: Tenth ACM Conference on Data and Application Security and Privacy, New Orleans, LA, USA, March 16-18, 2020*, pages 73–84. ACM, 2020.

[Bertossi 2011] Leopoldo E. Bertossi. *Database Repairing and Consistent Query Answering*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.

[Besnard & Hunter 2008] Philippe Besnard and Anthony Hunter. *Elements of argumentation*, volume 47. MIT press Cambridge, 2008.

[Biba 1977] Kenneth J Biba. « Integrity considerations for secure computer systems ». Technical report, MITRE Corp., 1977.

[Bienvenu & Bourgaux 2017] Meghyn Bienvenu and Camille Bourgaux. « Inconsistency-Tolerant Querying of Description Logic Knowledge Bases ». In *Reasoning Web: Logical Foundation of Knowledge Graph Construction and Query Answering*, volume 9885, pages 156–202. LNCS. Springer, 2017.

[Bienvenu & Bourgaux 2020] Meghyn Bienvenu and Camille Bourgaux. « Querying and Repairing Inconsistent Prioritized Knowledge Bases: Complexity Analysis and Links with Abstract Argumentation ». In *Principles of Knowledge Representation and Reasoning (KR), Virtual Event*, pages 141–151, 2020.

[Bienvenu & Bourgaux 2022] Meghyn Bienvenu and Camille Bourgaux. « Querying Inconsistent Prioritized Data with ORBITS: Algorithms, Implementation, and Experiments », 2022.

[Bienvenu & Rosati 2013] Meghyn Bienvenu and Riccardo Rosati. « Tractable Approximations of Consistent Query Answering for Robust Ontology-based Data Access ». In *IJCAI*, pages 775–781, 2013.

[Bienvenu *et al.* 2014] Meghyn Bienvenu, Camille Bourgaux, and François Goasdoué. « Querying Inconsistent Description Logic Knowledge Bases under Preferred Repair Semantics ». In *AAAI*, pages 996–1002, 2014.

[Bienvenu *et al.* 2019] Meghyn Bienvenu, Camille Bourgaux, and François Goasdoué. « Computing and explaining query answers over inconsistent DL-Lite knowledge bases ». *Journal of Artificial Intelligence Research*, 64:563–644, 2019.

[Bienvenu 2012a] Meghyn Bienvenu. « On the complexity of consistent query answering in the presence of simple ontologies ». In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pages 705–711, 2012.

[Bienvenu 2012b] Meghyn Bienvenu. « On the Complexity of Consistent Query Answering in the Presence of Simple Ontologies ». In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[Bishop 1981] Matt Bishop. « Hierarchical take-grant protection systems ». *ACM SIGOPS Operating Systems Review*, 15(5):109–122, 1981.

[Boella *et al.* 2005] Guido Boella, Joris Hulstijn, and Leendert van der Torre. « Argument games for interactive access control ». In *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, pages 751–754. IEEE, 2005.

[Bouij-Pasquier *et al.* 2015] Imane Bouij-Pasquier, Abdellah Ait Ouahman, Anas Abou El Kalam, and Mina Ouabiba de Montfort. « SmartOrBAC security and privacy in the Internet of Things ». In *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, pages 1–8. IEEE, 2015.

[Bourgaux 2016] Camille Bourgaux. « *Inconsistency handling in ontology-mediated query answering* ». PhD thesis, Université Paris Saclay (COmUE), 2016.

[Brewka 1989] Gerhard Brewka. « Preferred Subtheories: An Extended Logical Framework for Default Reasoning ». In *IJCAI, Detroit, USA*, pages 1043–1048, 1989.

[Bringhenti *et al.* 2023] Daniele Bringhenti, Lucia Seno, and Fulvio Valenza. « An Optimized Approach for Assisted Firewall Anomaly Resolution ». *IEEE Access*, 11:119693–119710, 2023.

[Britz *et al.* 2021] Katarina Britz, Giovanni Casini, Thomas Meyer, Kody Moodley, Uli Sattler, and Ivan Varzinczak. « Principles of KLM-style Defeasible Description Logics ». *ACM Trans. Comput. Log.*, 22(1):1:1–1:46, 2021.

[Calì *et al.* 2003] Andrea Calì, Domenico Lembo, and Riccardo Rosati. « On the decidability and complexity of query answering over inconsistent and incomplete databases ». In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 260–271, 2003.

[Calvanese *et al.* 2006] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. « Data Complexity of Query Answering in Description Logics ». In Patrick Doherty, John Mylopoulos, and Christopher A. Welty, editors, *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006*, pages 260–270. AAAI Press, 2006.

[Calvanese *et al.* 2007] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. « Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Family ». *Journal of Automated Reasoning*, 39(3):385–429, 2007.

[Calvanese *et al.* 2010a] Diego Calvanese, Evgeny Kharlamov, Werner Nutt, and Dmitriy Zheleznyakov. « Evolution of DL- Lite knowledge bases ». In *The Semantic Web–ISWC 2010: 9th International Semantic Web Conference, ISWC 2010, Shanghai, China, November 7-11, 2010, Revised Selected Papers, Part I 9*, pages 112–128. Springer, 2010.

[Calvanese *et al.* 2010b] Diego Calvanese, Evgeny Kharlamov, Werner Nutt, and Dmitriy Zheleznyakov. « Evolution of DL-Lite Knowledge Bases ». In *International Semantic Web Conference (1), Shanghai, China*, pages 112–128, 2010.

[Calvanese *et al.* 2011] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. « The MASTRO system for ontology-based data access ». *Semantic web*, 2(1):43–53, 2011.

[Caniupán & Bertossi 2010] Monica Caniupán and Leopoldo Bertossi. « The consistency extractor system: Answer set programs for consistent query answering in databases ». *Data & Knowledge Engineering*, 69(6):545–572, 2010.

[Casini & Straccia 2010] Giovanni Casini and Umberto Straccia. « Rational Closure for Defeasible Description Logics ». In *European Conference on Logics in Artificial Intelligence*, 2010.

[Casini & Straccia 2012] Giovanni Casini and Umberto Straccia. « Lexicographic closure for defeasible description logics ». In *Proc. of Australasian Ontology Workshop*, volume 969, pages 28–39, 2012.

[Charalambides *et al.* 2005] Marinos Charalambides, Paris Flegkas, George Pavlou, Arosha K Bandara, Emil C Lupu, Alessandra Russo, N Dulav, Morris Sloman, and Javier Rubio-Loyola. « Policy conflict analysis for quality of service management ». In *Sixth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'05)*, pages 99–108. IEEE, 2005.

[Chari *et al.* 2020] Shruthi Chari, Oshani Seneviratne, Daniel M Gruen, Morgan A Foreman, Amar K Das, and Deborah L McGuinness. « Explanation ontology: a model of explanations for user-centered AI ». In *International Semantic Web Conference*, pages 228–243. Springer, 2020.

[Chomicki *et al.* 2004a] Jan Chomicki, Jerzy Marcinkowski, and Slawomir Staworko. « Computing consistent query answers using conflict hypergraphs ». In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 417–426, 2004.

[Chomicki *et al.* 2004b] Jan Chomicki, Jerzy Marcinkowski, and Slawomir Staworko. « Hippo: A system for computing consistent answers to a class of SQL queries ». In *International Conference on Extending Database Technology*, pages 841–844. Springer, 2004.

[Chortaras *et al.* 2011] Alexandros Chortaras, Despoina Trivela, and Giorgos Stamou. « Optimized query rewriting for OWL 2 QL ». In

Automated Deduction–CADE-23: 23rd International Conference on Automated Deduction, Wrocław, Poland, July 31-August 5, 2011. Proceedings 23, pages 192–206. Springer, 2011.

[Chouchani *et al.* 2022] Nadia Chouchani, Sana Debbech, and Matthieu Perin. « Model-based safety engineering for autonomous train map ». *Journal of Systems and Software*, 183:111082, 2022.

[Clinciu *et al.* 2021] Miruna-Adriana Clinciu, Arash Eshghi, and Helen Hastie. « A Study of Automatic Metrics for the Evaluation of Natural Language Explanations ». In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2376–2387, Online, April 2021. Association for Computational Linguistics.

[Cook 1971] Stephen A. Cook. « The Complexity of Theorem-Proving Procedures ». In Michael A. Harrison, Ranan B. Banerji, and Jeffrey D. Ullman, editors, *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 151–158, Shaker Heights, Ohio, USA, May 1971. ACM.

[Costabello *et al.* 2012] Luca Costabello, Serena Villata, Nicolas Delaforge, and Fabien Gandon. « Linked data access goes mobile: Context-aware authorization for graph stores ». In *LDOW-5th WWW Workshop on Linked Data on the Web-2012*, 2012.

[Cozman 2000] Fabio G Cozman. « Credal networks ». *Artificial Intelligence Journal*, 120:199–233, 2000.

[Crum *et al.* 2024] Elias Crum, Antonio De Santis, Manon Ovide, Jiaxin Pan, Alessia Pisu, Nicolas Lazzari, and Sebastian Rudolph. « Enriching Ontologies with Disjointness Axioms using Large Language Models ». In *KBC-LM/LM-KBC@ ISWC*, 2024.

[Cuppens *et al.* 2004] Frédéric Cuppens, Nora Cuppens-Boulahia, and Alexandre Miège. « Inheritance hierarchies in the Or-BAC Model and application in a network environment ». *Proc. Foundations of Computer Security (FCS04)*, pages 41–60, 2004.

[Cuppens *et al.* 2007] Frédéric Cuppens, Nora Cuppens-Boulahia, and Meriam Ben Ghorbel. « High Level Conflict Management Strategies in Advanced Access Control Models ». *Electronic Notes in Theoretical Computer Science*, 186:3–26, 2007. Proceedings of the First Workshop in Information and Computer Security (ICS 2006).

[Darwiche & Pearl 1997] Adnan Darwiche and Judea Pearl. « On the logic of iterated belief revision ». *Artificial intelligence*, 89(1-2):1–29, 1997.

[De Bona *et al.* 2019] Glauber De Bona, John Grant, Anthony Hunter, and Sébastien Konieczny. « Classifying inconsistency measures using graphs ». *Journal of Artificial Intelligence Research*, 66:937–987, 2019.

[De Giacomo *et al.* 2021] Giuseppe De Giacomo, Xavier Oriol, Riccardo Rosati, and Domenico Fabio Savo. « Instance-level update in DL-Lite ontologies through first-order rewriting ». *Journal of Artificial Intelligence Research*, 70:1335–1371, 2021.

[de Saint-Cyr & Prade 2008] Florence Dupin de Saint-Cyr and Henri Prade. « Handling uncertainty and defeasibility in a possibilistic logic setting ». *International Journal of Approximate Reasoning*, 49(1):67–82, 2008.

[de Vergara *et al.* 2008] Jorge E López de Vergara, Enrique Vázquez, and Javier Guerra. « Security policy instantiation to react to network attacks-An ontology-based approach using OWL and SWRL ». In *International Conference on Security and Cryptography*, volume 2, pages 78–83. SCITEPRESS, 2008.

[de Vergara *et al.* 2009] Jorge E López de Vergara, Enrique Vázquez, Antony Martin, Samuel Dubus, and Marie-Noëlle Lepareux. « Use of Ontologies for the Definition of Alerts and Policies in a Network Security Platform. ». *J. Networks*, 4(8):720–733, 2009.

[Dean & (eds.) 2004] Mike Dean and Guus Schreiber (eds.). « OWL Web Ontology Language Reference ». World Wide Web Consortium, Recommendation REC-owl-ref-20040210, February 2004.

[Dempster 1967] Arthur P. Dempster. « Upper and lower probabilities induced by a multivalued mapping ». *The Annals of Mathematical Statistics*, 38:325–339, 1967.

[Denœux *et al.* 2020] Thierry Denœux, Didier Dubois, and Henri Prade. Representations of uncertainty in artificial intelligence: Probability and possibility. In *A guided tour of artificial intelligence research: Volume I: Knowledge representation, reasoning and learning*, pages 69–117. Springer, 2020.

[Destercke 2018] Sébastien Destercke. « A generic framework to include belief functions in preference handling and multi-criteria decision ». *International Journal of Approximate Reasoning*, 98:62–77, 2018.

[Dixit & Kolaitis 2021] Akhil A Dixit and Phokion G Kolaitis. « Consistent answers of aggregation queries using SAT solvers ». *arXiv preprint arXiv:2103.03314*, 2021.

[Du *et al.* 2013] J. Du, G. Qi, and Y. Shen. « Weight-based consistent query answering over inconsistent SHIQ knowledge bases ». *Knowledge and Information Systems*, 34(2):335–371, 2013.

[Dubois & Prade 1988] D. Dubois and H. Prade. « Possibility Theory ». *Plenum Press, New-York*, 1988.

[Dubois & Prade 1998] Didier Dubois and Henri Prade. « *Possibility Theory: Qualitative and Quantitative Aspects* », Chapter 6, pages 169–226. Springer Netherlands, Dordrecht, 1998.

[Dubois & Prade 2014] Didier Dubois and Henri Prade. « Possibilistic Logic - An Overview ». *Computational Logic*, 9:197–255, 01 2014.

[Dubois & Prade 2018] Didier Dubois and Henri Prade. « A Crash Course on Generalized Possibilistic Logic ». In Davide Ciucci, Gabriella Pasi, and Barbara Vantaggi, editors, *Scalable Uncertainty Management - 12th International Conference, SUM, Milan, Italy*, volume 11142 of *Lecture Notes in Computer Science*, pages 3–17. Springer, 2018.

[Dubois & Prade 2024] Didier Dubois and Henri Prade. « Reasoning and learning in the setting of possibility theory-overview and perspectives ». *International Journal of Approximate Reasoning*, 171:109028, 2024.

[Dubois *et al.* 1993] Didier Dubois, Henri Prade, and Sandra Sandri. « On possibility/probability transformations ». *Fuzzy logic: State of the art*, pages 103–112, 1993.

[Dubois *et al.* 2006] Didier Dubois, Jérôme Mengin, and Henri Prade. Possibilistic uncertainty and fuzzy features in description logic. A preliminary discussion. In *Capturing Intelligence*, volume 1, pages 101–113. Elsevier, 2006.

[Dubois *et al.* 2023] Didier Dubois, Lluis Godo, and Henri Prade. « An elementary belief function logic ». *Journal of Applied Non-Classical Logics*, 33(3-4):582–605, 2023.

[Dung 1995] Phan Minh Dung. « On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games ». *Artificial intelligence*, 77(2):321–357, 1995.

[Dušek & Kasner 2020] Ondřej Dušek and Zdeněk Kasner. « Evaluating Semantic Accuracy of Data-to-Text Generation with Natural Language Inference ». In *Proceedings of the 13th International Conference on Natural Language Generation (INLG 2020)*, pages 131–137, Online, December 2020.

[Esteva *et al.* 2021] Francesc Esteva, Aldo Figallo-Orellano, Tommaso Flaminio, and Lluis Godo. « Logics of formal inconsistency based on distributive involutive residuated lattices ». *Journal of Logic and Computation*, 31(5):1226–1265, 2021.

[Everett *et al.* 2021] Lloyd Everett, Emily Morris, and Thomas Meyer. « Explanation for KLM-Style Defeasible Reasoning ». In Edgar Jembere, Aurona J. Gerber, Serestina Viriri, and Anban W. Pillay, editors, *Artificial Intelligence Research - Second Southern African Conference, SACAIR 2021, Durban, South Africa, 2021, Proceedings*, volume 1551 of *Communications in Computer and Information Science*, pages 192–207. Springer, 2021.

[Facebook AI 2020] Facebook AI. « facebook/bart-large-mnli ». https://huggingface.co/facebook/bart-large-mnli, 2020. Accessed: 2025-05-11.

[Fagin *et al.* 2003] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning About Knowledge*. MIT Press, 2003.

[Fan *et al.* 2023] Zhiwei Fan, Paraschos Koutris, Xiating Ouyang, and Jef Wijsen. « LinCQA: Faster consistent query answering with linear time guarantees ». *Proceedings of the ACM on Management of Data*, 1(1):1–25, 2023.

[Fernández *et al.* 2019] Maribel Fernández, Ian Mackie, and Bhavani Thuraisingham. « Specification and analysis of ABAC policies via the category-based metamodel ». In *Proceedings of the Ninth ACM conference on data and application security and privacy*, pages 173–184, 2019.

[Finkelstein *et al.* 1994] Anthony CW Finkelstein, Dov Gabbay, Anthony Hunter, Jeff Kramer, and Bashar Nuseibeh. « Inconsistency handling in multiperspective specifications ». *IEEE Transactions on Software Engineering*, 20(8):569–578, 1994.

[Frittella *et al.* 2020] Sabine Frittella, Krishna Manoorkar, Alessandra Palmigiano, Apostolos Tzimoulis, and Nachoem Wijnberg. « Toward a Dempster-Shafer theory of concepts ». *International Journal of Approximate Reasoning*, 125:14–25, 2020.

[Fuxman & Miller 2005] Ariel D Fuxman and Renée J Miller. « First-order query rewriting for inconsistent databases ». In *International Conference on Database Theory*, pages 337–351. Springer, 2005.

[Fuxman *et al.* 2005] Ariel Fuxman, Elham Fazli, and Renée J Miller. « Conquer: Efficient management of inconsistent databases ». In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 155–166, 2005.

[Gallier 2015] Jean H Gallier. *Logic for computer science: foundations of automatic theorem proving*. Courier Dover Publications, 2015.

[Giordano & Gliozzi 2021] Laura Giordano and Valentina Gliozzi. « A reconstruction of multipreference closure ». *Artificial Intelligence*, 290:103398, 2021.

[Giordano *et al.* 2008] Laura Giordano, Valentina Gliozzi, Nicola Olivetti, and Gian Luca Pozzato. « Reasoning about typicality in preferential description logics ». In *European workshop on logics in artificial intelligence*, pages 192–205. Springer, 2008.

[Giordano *et al.* 2015] Laura Giordano, Valentina Gliozzi, Nicola Olivetti, and Gian Luca Pozzato. « Semantic characterization of rational closure: From propositional logic to description logics ». *Artificial Intelligence*, 226:1–33, 2015.

[Giordano *et al.* 2022] Laura Giordano, Valentina Gliozzi, and Daniele Theseider Dupré. « A conditional, a fuzzy and a probabilistic interpretation of self-organizing maps ». *Journal of Logic and Computation*, 32(2):178–205, 2022.

[Gottlob *et al.* 2014] Georg Gottlob, Giorgio Orsi, and Andreas Pieris. « Query rewriting and optimization for ontological databases ». *ACM Transactions on Database Systems (TODS)*, 39(3):1–46, 2014.

[Graham & Denning 1971] G Scott Graham and Peter J Denning. « Protection: principles and practice ». In *Proceedings of the May 16-18, 1972, spring joint computer conference*, pages 417–429, 1971.

[Grant & Hunter 2011] John Grant and Anthony Hunter. « Measuring consistency gain and information loss in stepwise inconsistency resolution ». In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 362–373. Springer, 2011.

[Grant & Hunter 2023] John Grant and Anthony Hunter. « Semantic inconsistency measures using 3-valued logics ». *International Journal of Approximate Reasoning*, 156:38–60, 2023.

[Guo *et al.* 2005] Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. « LUBM: A benchmark for OWL knowledge base systems ». *Journal of Web Semantics*, 3(2-3):158–182, 2005.

[Gutin 2018] Gregory Gutin. « Acyclic digraphs ». *Classes of Directed Graphs*, pages 125–172, 2018.

[Harris & (eds.) 2013] Steve Harris and Andy Seaborne (eds.). « SPARQL 1.1 Query Language ». World Wide Web Consortium, Recommendation REC-sparql11-query-20130321, March 2013.

[Harrison *et al.* 1976] Michael A Harrison, Walter L Ruzzo, and Jeffrey D Ullman. « Protection in operating systems ». *Communications of the ACM*, 19(8):461–471, 1976.

[He *et al.* 2022] Yuan He, Jiaoyan Chen, Denvar Antonyrajah, and Ian Horrocks. « BERTMap: A BERT-Based Ontology Alignment System ». *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(5):5684–5691, Jun. 2022.

[Hipp 2020] Richard D Hipp. « SQLite », 2020.

[Hollunder 1995] Bernhard Hollunder. « An alternative proof method for possibilistic logic and its application to terminological logics ». *International Journal of Approximate Reasoning*, 12(2):85–109, 1995.

[Horrocks *et al.* 2004] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. « SWRL: A Semantic Web Rule Language Combining OWL and RuleML ». World Wide Web Consortium, Member Submission SUBM-SWRL-20040521, May 2004.

[Hu *et al.* 2014] Vincent C Hu, David Ferraiolo, Rick Kuhn, Adam Schnitzer, Kenneth Sandlin, Robert Miller, Karen Scarfone, and others. « Guide to attribute based access control (ABAC) definition and considerations ». *NIST special publication*, 800(162):1–54, 2014.

[Huitzil *et al.* 2020] Ignacio Huitzil, Fernando Bobillo, Juan Gómez-Romero, and Umberto Straccia. « Fudge: Fuzzy ontology building with consensuated fuzzy datatypes ». *Fuzzy Sets and Systems*, 401:91–112, 2020.

[Jajodia *et al.* 2001] Sushil Jajodia, Pierangela Samarati, Maria Luisa Sapino, and VS Subrahmanian. « Flexible support for multiple access control policies ». *ACM Transactions on Database Systems (TODS)*, 26(2):214–260, 2001.

[Ji *et al.* 2022] Qiu Ji, Weizhuo Li, Shiqi Zhou, Guilin Qi, and Yuanfang Li. « Benchmark construction and experimental evaluations for incoherent ontologies ». *Knowledge-Based Systems*, 239:108090, 2022.

[Jin *et al.* 2012] Xin Jin, Ram Krishnan, and Ravi Sandhu. « A unified attribute-based access control model covering DAC, MAC and RBAC ». In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 41–55. Springer, 2012.

[Johnson & Klug 1982] David S Johnson and Anthony Klug. « Testing containment of conjunctive queries under functional and inclusion

dependencies ». In *Proceedings of the 1st ACM SIGACT-SIGMOD symposium on Principles of database systems*, pages 164–169, 1982.

[Jousselme *et al.* 2021] Anne-Laure Jousselme, Frédéric Pichon, Nadia Ben Abdallah, and Sébastien Destercke. « A note about entropy and inconsistency in evidence theory ». In *International Conference on Belief Functions*, pages 215–223. Springer, 2021.

[Kalam *et al.* 2003] Anas Abou El Kalam, R El Baida, Philippe Balbiani, Salem Benferhat, Frédéric Cuppens, Yves Deswarte, Alexandre Miege, Claire Saurel, and Gilles Trouessin. « Organization based access control ». In *Proceedings POLICY 2003. IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, pages 120–131. IEEE, 2003.

[Kalyanpur *et al.* 2007] Aditya Kalyanpur, Bijan Parsia, Matthew Horridge, and Evren Sirin. « Finding all justifications of OWL DL entailments ». In *The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007+ ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings*, pages 267–280. Springer, 2007.

[Kimelfeld *et al.* 2020] Benny Kimelfeld, Ester Livshits, and Liat Peterfreund. « Counting and enumerating preferred database repairs ». *Theoretical Computer Science*, 837:115–157, 2020.

[Kirrane *et al.* 2013] Sabrina Kirrane, Ahmed Abdelrahman, Alessandra Mileo, and Stefan Decker. « Secure manipulation of linked data ». In *The Semantic Web–ISWC 2013: 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I 12*, pages 248–263. Springer, 2013.

[Kirrane *et al.* 2017] Sabrina Kirrane, Alessandra Mileo, and Stefan Decker. « Access Control and the Resource Description Framework: A Survey ». *Semantic Web*, 8(2):311–352, 2017.

[Kirrane *et al.* 2020] Sabrina Kirrane, Alessandra Mileo, Axel Polleres, and Stefan Decker. « Query Based Access Control for Linked Data ». *arXiv preprint arXiv:2007.00461*, 2020.

[Kolaitis *et al.* 2013] Phokion G Kolaitis, Enela Pema, and Wang-Chiew Tan. « Efficient querying of inconsistent databases with binary integer programming ». *Proceedings of the VLDB Endowment*, 6(6):397–408, 2013.

[Konieczny *et al.* 2019] Sébastien Konieczny, Pierre Marquis, and Srdjan Vesic. « Rational inference relations from maximal consistent subsets selection ». In *Twenty-Eighth International Joint Conference on Artificial Intelligence {IJCAI-19}*, pages 1749–1755, 2019.

[Kontchakov *et al.* 2010] Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, and Michael Zakharyaschev. « The Combined Approach to Query Answering in DL-Lite ». In *Principles of Knowledge Representation and Reasoning (KR), Toronto, Canada*, pages 247–257, 2010.

[Koutris & Wijsen 2021] Paraschos Koutris and Jef Wijsen. « Consistent query answering for primary keys in datalog ». *Theory of Computing Systems*, 65(1):122–178, 2021.

[Kraus *et al.* 1990] Sarit Kraus, Daniel Lehmann, and Menachem Magidor. « Nonmonotonic Reasoning, Preferential Models and Cumulative Logics ». *Artificial Intelligence Journal*, 44(1-2):167–207, 1990.

[Krech *et al.* 2023] Daniel Krech, Gunnar AAstrand Grimnes, Graham Higgins, Jörn Hees, Iwan Aucamp, Niklas Lindström, Natanael Arndt, Ashley Sommer, Edmond Chuc, Ivan Herman, Alex Nelson, Jamie McCusker, Tom Gillespie, Thomas Kluyver, Florian Ludwig, Pierre-Antoine Champin, Mark Watts, Urs Holzer, Ed Summers, Whit Morriss, Donny Winston, Drew Perttula, Filip Kovacevic, Remi Chateauneu, Harold Solbrig, Benjamin Cogrel, and Veyndan Stuart. « RDFLib », 2023.

[Laamech *et al.* 2022] Nouha Laamech, Manuel Munier, and Congduc Pham. « Idsm-o: An iot data sharing management ontology for data governance ». In *Proceedings of the 14th International Conference on Management of Digital EcoSystems*, pages 88–95, 2022.

[Lampson 1974] Butler W Lampson. « Protection ». *ACM SIGOPS Operating Systems Review*, 8(1):18–24, 1974.

[Laouar & Benferhat 2026] Ahmed Laouar and Salem Benferhat. « Closure-Based Tractable Possibilistic Inference from Partially Ordered DL-Lite Ontologies ». In Giovanni Casini, Besik Dundua, and Temur Kutsia, editors, *Logics in Artificial Intelligence*, pages 240–256, Cham, 2026. Springer Nature Switzerland.

[Laouar *et al.* 2023] Ahmed Laouar, Sihem Belabbes, and Salem Benferhat. « Tractable Closure-Based Possibilistic Repair for Partially Ordered DL-Lite Ontologies ». In Sarah Gaggl, Maria Vanina Martinez, and Magdalena Ortiz, editors, *Logics in Artificial Intelligence*, pages 353–368, Cham, 2023. Springer Nature Switzerland.

[Laouar *et al.* 2024] Ahmed Laouar, Sihem Belabbes, and Salem Benferhat. « Conflict Handling Strategies for Partially Ordered Access

Control Security Policies ». In *2024 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pages 1–6, 2024.

[Laouar *et al.* 2025a] Ahmed Laouar, Sihem Belabbes, and Salem Benferhat. « How to tractably compute a productive repair for possibilistic partially ordered DL-LiteR ontologies? ». *Fuzzy Sets and Systems*, 510:109361, 2025.

[Laouar *et al.* 2025b] Ahmed Laouar, Toky Raboanary, and Salem Benferhat. « An Ontology-Based Approach for Handling Inconsistency in Explainable and Prioritized Access Control Models ». In Sébastien Destercke, Maria Vanina Martinez, and Giuseppe Sanfilippo, editors, *Scalable Uncertainty Management*, pages 249–264, Cham, 2025. Springer Nature Switzerland.

[Lehmann *et al.* 2015] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and others. « Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia ». *Semantic web*, 6(2):167–195, 2015.

[Lembo *et al.* 2010] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. « Inconsistency-Tolerant Semantics for Description Logics ». In *Web Reasoning and Rule Systemss, RR 2010, Bressanone/Brixen, Italy*, pages 103–117, 2010.

[Lembo *et al.* 2011] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. « Query rewriting for inconsistent DL-Lite ontologies ». In *International Conference on Web Reasoning and Rule Systems*, pages 155–169. Springer, 2011.

[Lembo *et al.* 2015] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. « Inconsistency-tolerant query answering in ontology-based data access ». *Journal of Web Semantics*, 33:3–29, 2015.

[Li *et al.* 2019] Na Li, Zied Bouraoui, and Steven Schockaert. « Ontology completion using graph convolutional networks ». In *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part I 18*, pages 435–452. Springer, 2019.

[Liu *et al.* 2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke

Zettlemoyer, and Veselin Stoyanov. « Roberta: A robustly optimized bert pretraining approach ». *arXiv preprint arXiv:1907.11692*, 2019.

[Liu *et al.* 2023] Xiaolong Liu, Philippe Besnard, and Sylvie Doutre. « Paraconsistent inference relations induced from inconsistency measures ». *International Journal of Approximate Reasoning*, 152:183–197, 2023.

[Lukasiewicz *et al.* 2012] Thomas Lukasiewicz, Maria Vanina Martinez, and Gerardo I Simari. Inconsistency handling in Datalog+/- ontologies. In *ECAI 2012*, pages 558–563. IOS Press, 2012.

[Lukasiewicz *et al.* 2015] Thomas Lukasiewicz, Maria Vanina Martinez, Andreas Pieris, and Gerardo Simari. « From Classical to Consistent Query Answering under Existential Rules ». *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), Feb. 2015.

[Lukasiewicz *et al.* 2022] Thomas Lukasiewicz, Enrico Malizia, Maria Vanina Martinez, Cristian Molinaro, Andreas Pieris, and Gerardo I. Simari. « Inconsistency-tolerant query answering for existential rules ». *Artificial Intelligence*, 307:103685, 2022.

[Lukasiewicz *et al.* 2023] Thomas Lukasiewicz, Enrico Malizia, and Cristian Molinaro. « Complexity of Inconsistency-Tolerant Query Answering in Datalog+/– under Preferred Repairs ». In *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning*, pages 472–481, 8 2023.

[Lutz *et al.* 2013] C. Lutz, I. Seylan, D. Toman, and F. Wolter. « The Combined Approach to OBDA: Taming Role Hierarchies Using Filters ». In *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, Australia*, pages 314–330, 2013.

[Maier *et al.* 1979] David Maier, Alberto O Mendelzon, and Yehoshua Sagiv. « Testing implications of data dependencies ». *ACM Transactions on Database Systems (TODS)*, 4(4):455–469, 1979.

[Makinson 1994] David Makinson. General patterns in nonmonotonic reasoning. In C. J. HOGGER DOV M. GABBAY and J. A. ROBINSON, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3, Nonmonotonic Reasoning and Uncertain Reasoning*, Chapter 2, pages 35–110. Oxford University Press, 1994.

[Morris 2020] Jack Morris. « language_tool_python ». https://pypi.org/project/language-tool-python/, 2020. Accessed: 2025-05-11.

[Motik *et al.* 2012] Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, and Carsten Lutz. « OWL 2 Web Ontology Language Profiles. W3C Recommendation ». Available at https://www.w3.org/TR/owl2-profiles/, 11 December 2012.

[Nebel 1994] Bernhard Nebel. « Base Revision Operations and Schemes: Semantics, Representation and Complexity ». In Anthony G. Cohn, editor, *Proceedings of the Eleventh European Conference on Artificial Intelligence, Amsterdam, The Netherlands, August 8-12, 1994*, pages 341–345. John Wiley and Sons, Chichester, 1994.

[Novikova *et al.* 2017] Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. « Why we need new evaluation metrics for NLG ». *arXiv preprint arXiv:1707.06875*, 2017.

[Penelova 2021] Maria Penelova. « Access control models ». *Cybernetics and Information Technologies*, 21(4):77–104, 2021.

[Perez & Wijsen 2021] Horacio Tellez Perez and Jef Wijsen. « Generalized Weighted Repairs ». In *International Conference on Flexible Query Answering Systems*, pages 67–81. Springer, 2021.

[Pérez-Urbina *et al.* 2009] Héctor Pérez-Urbina, Ian Horrocks, and Boris Motik. « Efficient query answering for OWL 2 ». In *The Semantic Web-ISWC 2009: 8th International Semantic Web Conference, ISWC 2009, Chantilly, VA, USA, October 25-29, 2009. Proceedings 8*, pages 489–504. Springer, 2009.

[Perrussel *et al.* 2007] Laurent Perrussel, Sylvie Doutre, Jean-Marc Thévenin, and Peter McBurney. « A persuasion dialog for gaining access to information ». In *International Workshop on Argumentation in Multi-Agent Systems*, pages 63–79. Springer, 2007.

[Pinto Prieto *et al.* 2024] Daira Pinto Prieto, Ronald de Haan, and Sebastien Destercke. « How to Efficiently Decombine Belief Functions? ». In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 91–102. Springer, 2024.

[Poole 1985] David Poole. « On the Comparison of Theories: Preferring the Most Specific Explanation. ». In *IJCAI*, volume 85, pages 144–147. Citeseer, 1985.

[Proctor 2012] Mark Proctor. « Drools: a rule engine for complex event processing ». In *Applications of Graph Transformations with Industrial Relevance: 4th International Symposium, AGTIVE 2011, Budapest, Hungary, October 4-7, 2011, Revised Selected and Invited Papers 4*, pages 2–2. Springer, 2012.

[Rescher & Manor 1970] Nicholas Rescher and Ruth Manor. « On inference from inconsistent premisses ». *Theory and decision*, 1(2):179–217, 1970.

[Rodriguez-Muro *et al.* 2013] Mariano Rodriguez-Muro, Roman Kontchakov, and Michael Zakharyaschev. « Ontology-based data access: Ontop of databases ». In *International Semantic Web Conference*, pages 558–573. Springer, 2013.

[Rosati & Almatelli 2010] Riccardo Rosati and Alessandro Almatelli. « Improving Query Answering over DL-Lite Ontologies ». In Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczynski, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010*. AAAI Press, 2010.

[Rosati *et al.* 2012] Riccardo Rosati, Marco Ruzzi, Mirko Graziosi, and Giulia Masotti. « Evaluation of techniques for inconsistency handling in OWL 2 QL ontologies ». In *International Semantic Web Conference*, pages 337–349. Springer, 2012.

[Sacco & Breslin 2012] Owen Sacco and John G Breslin. « PPO & PPM 2.0: Extending the privacy preference framework to provide finer-grained access control for the web of data ». In *Proceedings of the 8th International Conference on Semantic Systems*, pages 80–87, 2012.

[Sacco *et al.* 2011] Owen Sacco, Alexandre Passant, and Stefan Decker. « An access control framework for the web of data ». In *2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 456–463. IEEE, 2011.

[Sandhu *et al.* 2000] Ravi Sandhu, David Ferraiolo, and Richard Kuhn. « The NIST Model for Role-Based Access Control: Towards a Unified Standard ». In *Proceedings of the Fifth ACM Workshop on Role-Based Access Control*, RBAC '00, Page 47–63, New York, NY, USA, 2000. Association for Computing Machinery.

[Sandhu 1992] Ravi S Sandhu. « The typed access matrix model. ». In *S&P*, pages 122–136, 1992.

[Scutari 2009] Marco Scutari. « Learning Bayesian networks with the bnlearn R package ». *arXiv preprint arXiv:0908.3817*, 2009.

[Shafer 1976] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[Spohn 1988] Wolfgang Spohn. « Ordinal conditional functions: A dynamic theory of epistemic states ». In *Causation in decision,*

belief change, and statistics: Proceedings of the Irvine Conference on Probability and Causation, pages 105–134. Springer, 1988.

[Spohn 2014] Wolfgang Spohn. *The Laws of Belief - Ranking Theory and Its Philosophical Applications*. Oxford University Press, 2014.

[Staworko *et al.* 2012] S. Staworko, J. Chomicki, and J. Marcinkowski. « Prioritized repairing and consistent query answering in relational databases ». *AMAI*, 64(2-3):209–246, 2012.

[Straccia 2015] Umberto Straccia. All about fuzzy description logics and applications. In *Reasoning Web International Summer School*, pages 1–31. Springer, 2015.

[Thimm & Santos 2024] Matthias Thimm and Jandson Santos Ribeiro Santos. « Reasoning with maximal consistent signatures ». *arXiv preprint arXiv:2408.17190*, 2024.

[Thimm 2018] Matthias Thimm. « On the evaluation of inconsistency measures ». *Measuring Inconsistency in Information*, 73:19–60, 2018.

[Thimm 2019] Matthias Thimm. « Inconsistency measurement ». In *International Conference on Scalable Uncertainty Management*, pages 9–23. Springer, 2019.

[Touazi *et al.* 2015] Fayçal Touazi, Claudette Cayrol, and Didier Dubois. « Possibilistic reasoning with partially ordered beliefs ». *Journal of Applied Logic*, 13(4):770–798, 2015.

[Toulmin 2003] Stephen E Toulmin. *The uses of argument*. Cambridge university press, 2003.

[Tsalapati *et al.* 2016] Eleni Tsalapati, Giorgos Stoilos, Giorgos Stamou, and George Koletsos. « Efficient Query Answering over Expressive Inconsistent Description Logics ». In *International Joint Conference on Artificial Intelligence (IJCAI), New York, USA*, pages 1279–1285, 2016.

[Turkmen *et al.* 2013] Fatih Turkmen, Simon Foley, Barry O'Sullivan, William Fitzgerald, Tarik Hadzic, Stylianos Basagiannis, and Menouer Boubekeur. « Explanations and relaxations for policy conflicts in physical access control ». In *2013 IEEE 25th International Conference on Tools with Artificial Intelligence*, pages 330–336. IEEE, 2013.

[Venetis *et al.* 2012] Tassos Venetis, Giorgos Stoilos, and Giorgos B Stamou. « Incremental query rewriting for OWL 2 QL ». In *Proceedings of the 2012 International Workshop on Description Logics*, 2012.

[Vrandečić & Krötzsch 2014] Denny Vrandečić and Markus Krötzsch. « Wikidata: a free collaborative knowledgebase ». *Commun. ACM*, 57(10):78–85, September 2014.

[W3C 2012] W3C. « Basic Access Control ontology (ACL) », 18-12-2012.

[Wijsen 2019] Jef Wijsen. « Foundations of query answering on inconsistent databases ». *ACM SIGMOD Record*, 48(3):6–16, 2019.

[Wolf *et al.* 2020] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. « Transformers: State-of-the-Art Natural Language Processing ». In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[Xiao *et al.* 2018] Guohui Xiao, Diego Calvanese, Roman Kontchakov, Domenico Lembo, Antonella Poggi, Riccardo Rosati, and Michael Zakharyaschev. « Ontology-Based Data Access: A Survey ». In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 5511–5519. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[Yahi *et al.* 2008] Safa Yahi, Salem Benferhat, Sylvain Lagrue, Mariette Sérayet, and Odile Papini. « A Lexicographic Inference for Partially Preordered Belief Bases ». In Gerhard Brewka and Jérôme Lang, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR 2008, Sydney, Australia, September 16-19, 2008*, pages 507–517. AAAI Press, 2008.

[Yuan & Tong 2005] Eric Yuan and Jin Tong. « Attributed based access control (ABAC) for web services ». In *IEEE International Conference on Web Services (ICWS'05)*. IEEE, 2005.

[Zadeh 1978] Lotfi Asker Zadeh. « Fuzzy sets as a basis for a theory of possibility ». *Fuzzy sets and systems*, 1(1):3–28, 1978.

[Zadeh 1988] Lotfi Asker Zadeh. « Fuzzy logic ». *Computer*, 21(4):83–93, 1988.

[Zhou *et al.* 2012] Liping Zhou, Houkuan Huang, Guilin Qi, Yue Ma, Zhisheng Huang, and Youli Qu. « Paraconsistent query answering over DL-Lite ontologies ». *Web Intelligence and Agent Systems: An International Journal*, 10(1):19–31, 2012.

Appendix A

Developed tools

Screenshots of the OrBAC Explanation Tool

Tool Interface and Example Policies

Figure A.1 shows the main interface of the OrBAC explanation tool, providing a starting point for users to explore and test policy explanations.

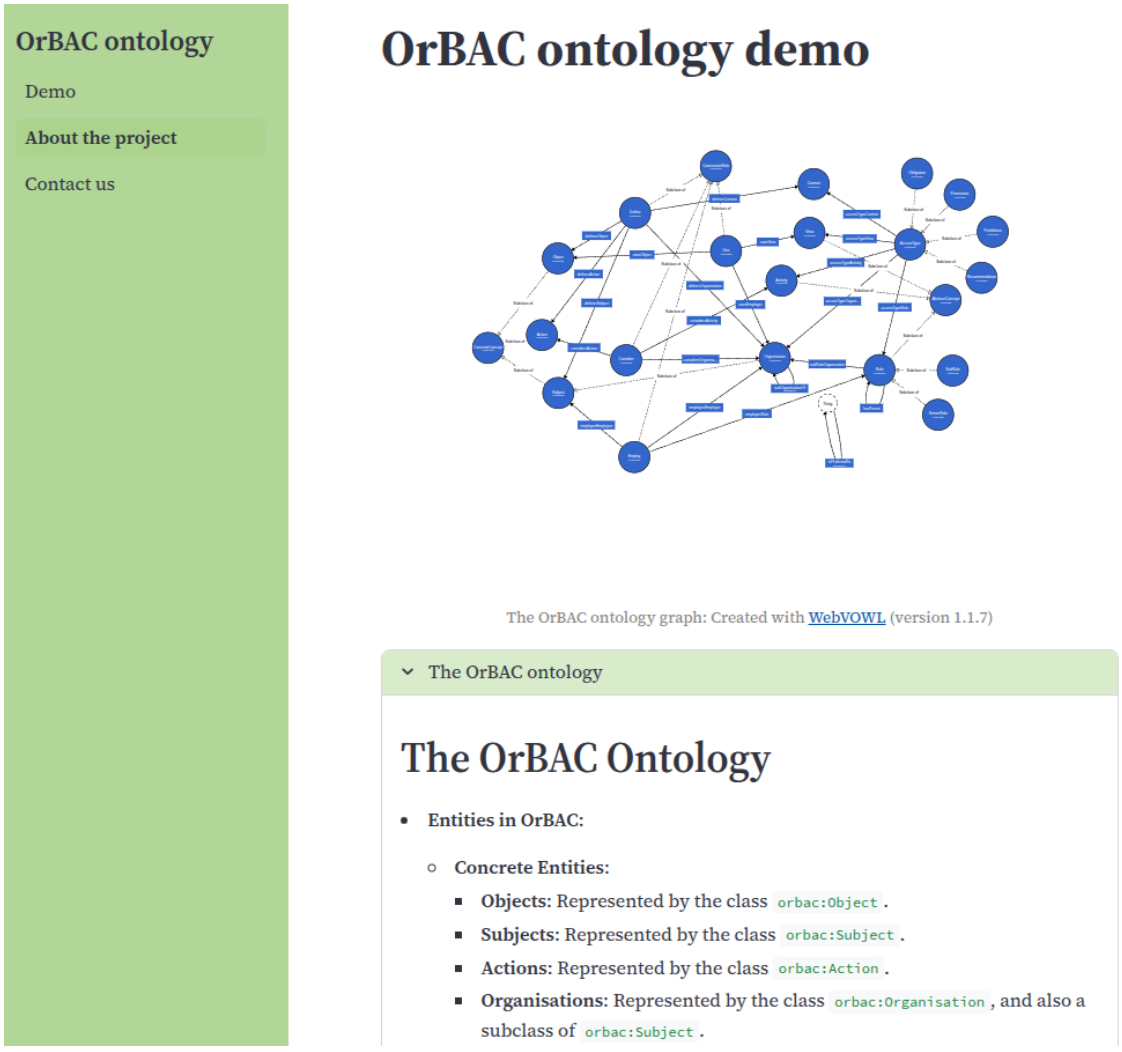


Figure A.1: Main interface of the OrBAC explanation tool.

Figure A.2 illustrates the option for the user to load an example policy. Future versions will allow creating and loading new policies.

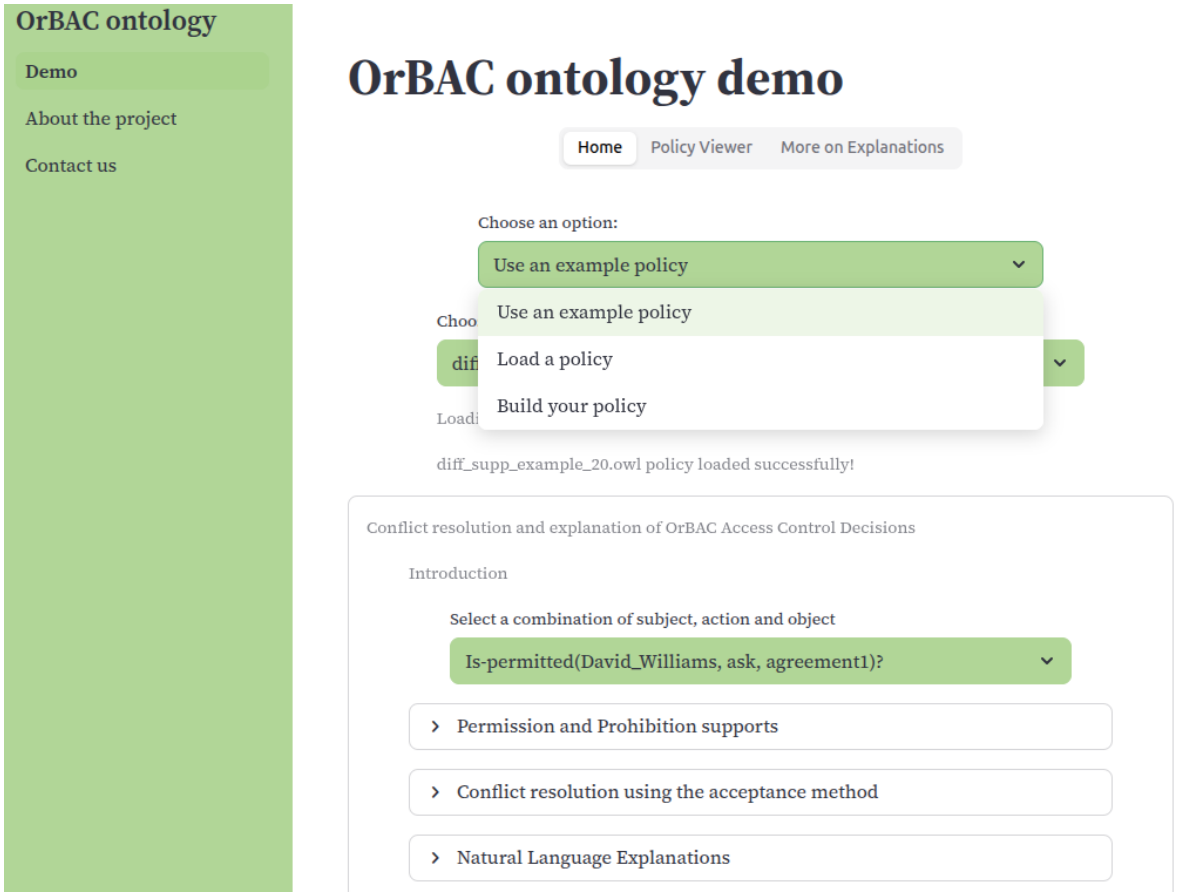


Figure A.2: Loading an example policy.

Figure A.3 shows the list of available policies that a user can try within the tool.

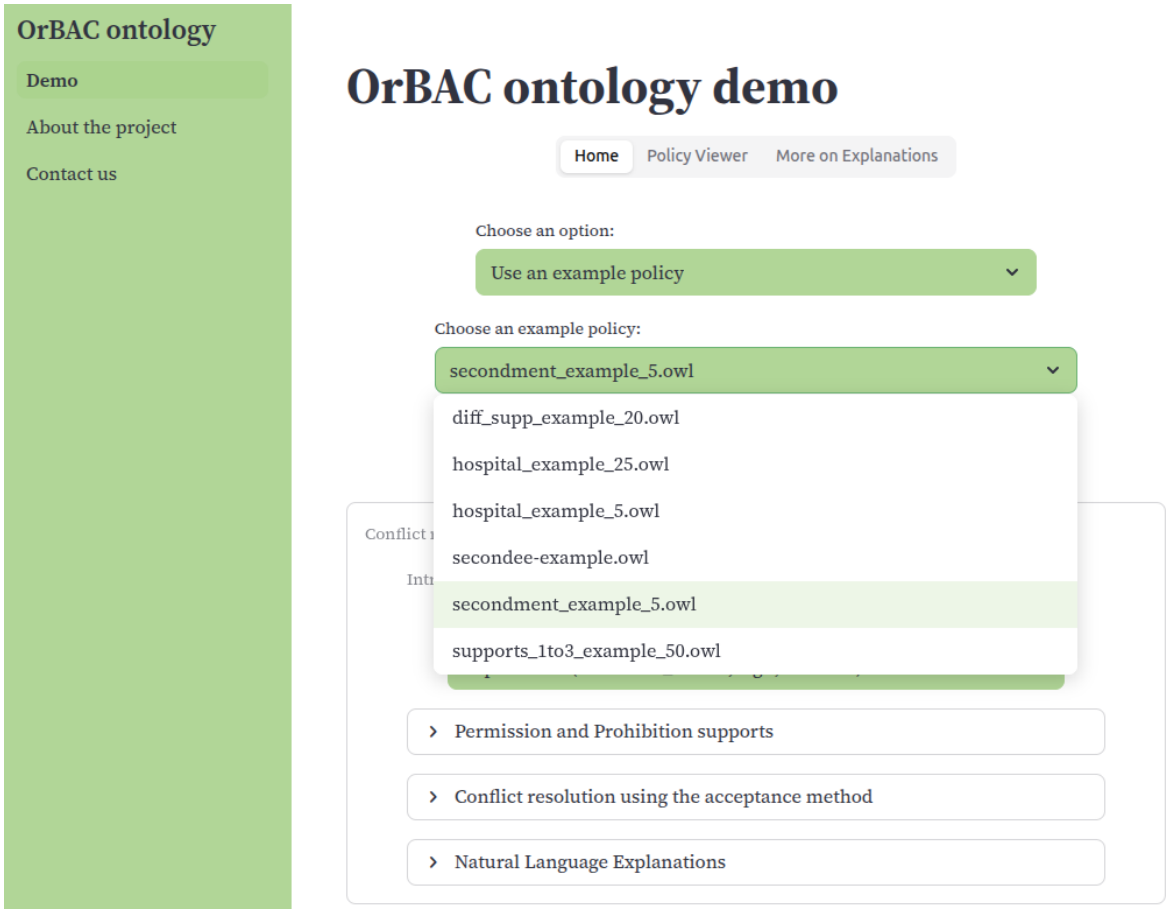


Figure A.3: List of example policies for testing.

Figure A.4 displays the set of queries that the user can try to evaluate permissions in the loaded policy.

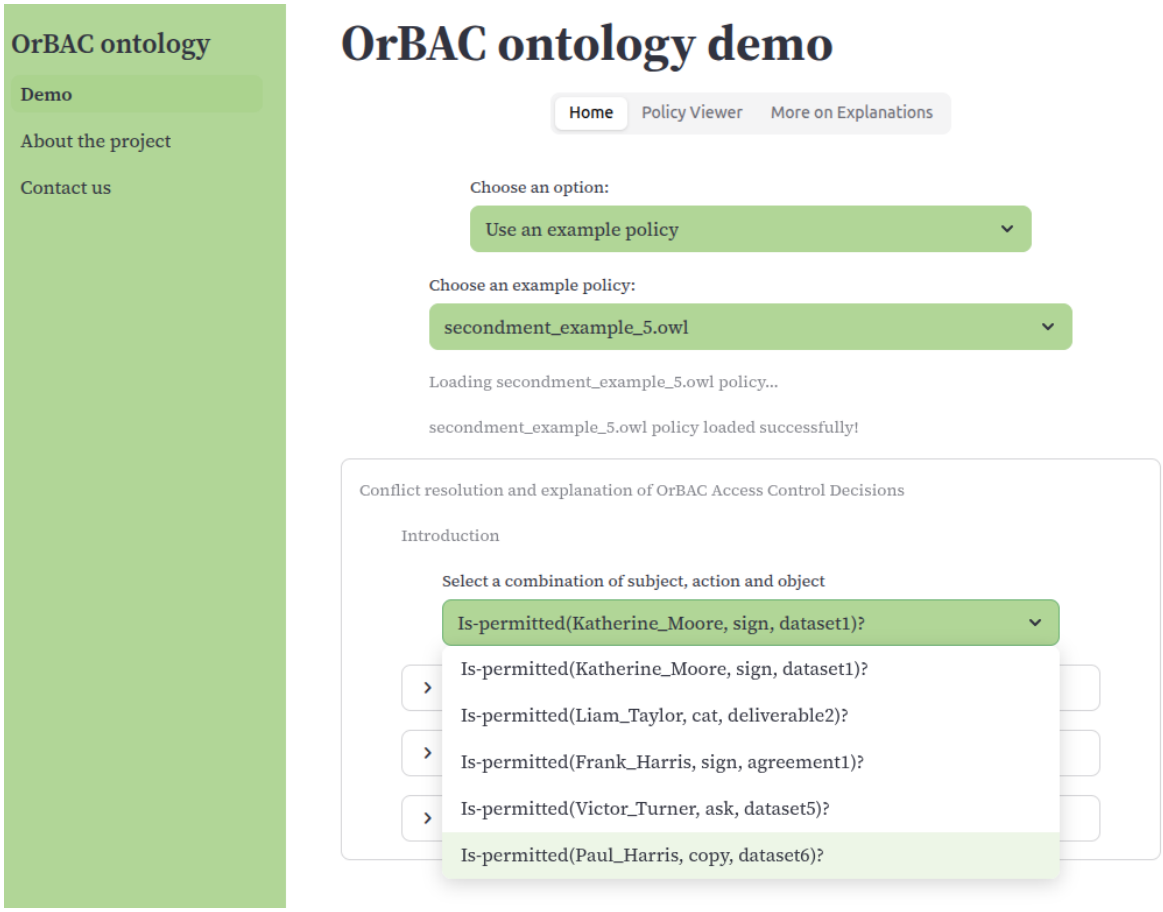


Figure A.4: List of queries to test access permissions.

Supports and Conflict Resolution

Figure A.5 shows the supports of permissions and prohibitions for the selected query, allowing the user to inspect which rules contribute to the decision.

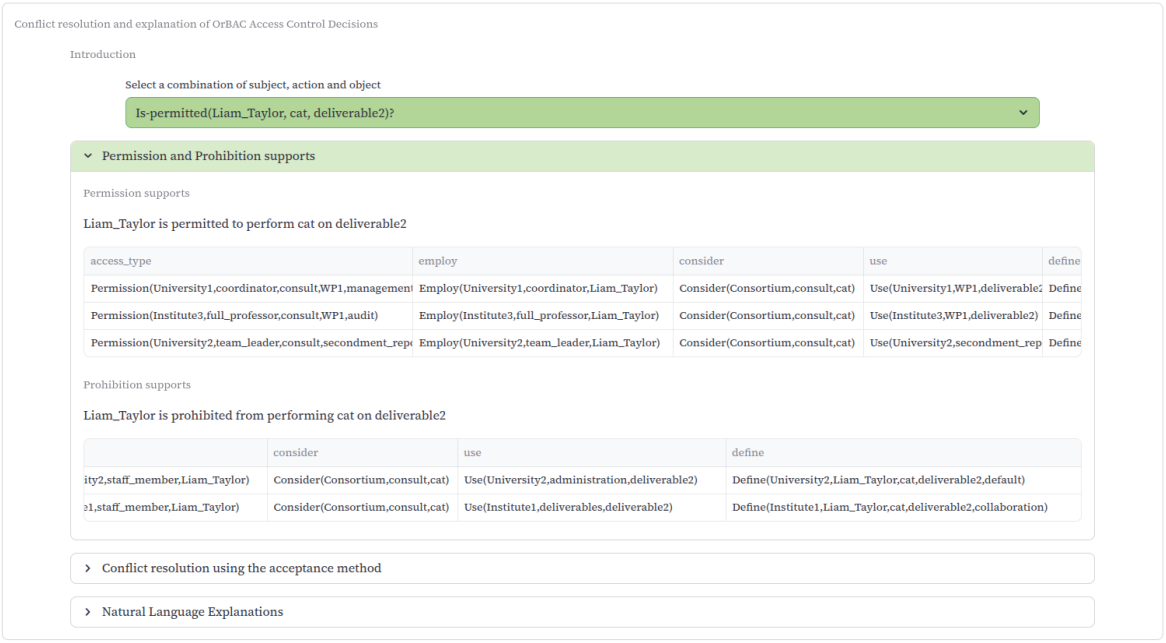


Figure A.5: Visualisation of supports for permissions and prohibitions.

Figure A.6 displays the outcome of conflict resolution for the selected query, showing which decision prevails.

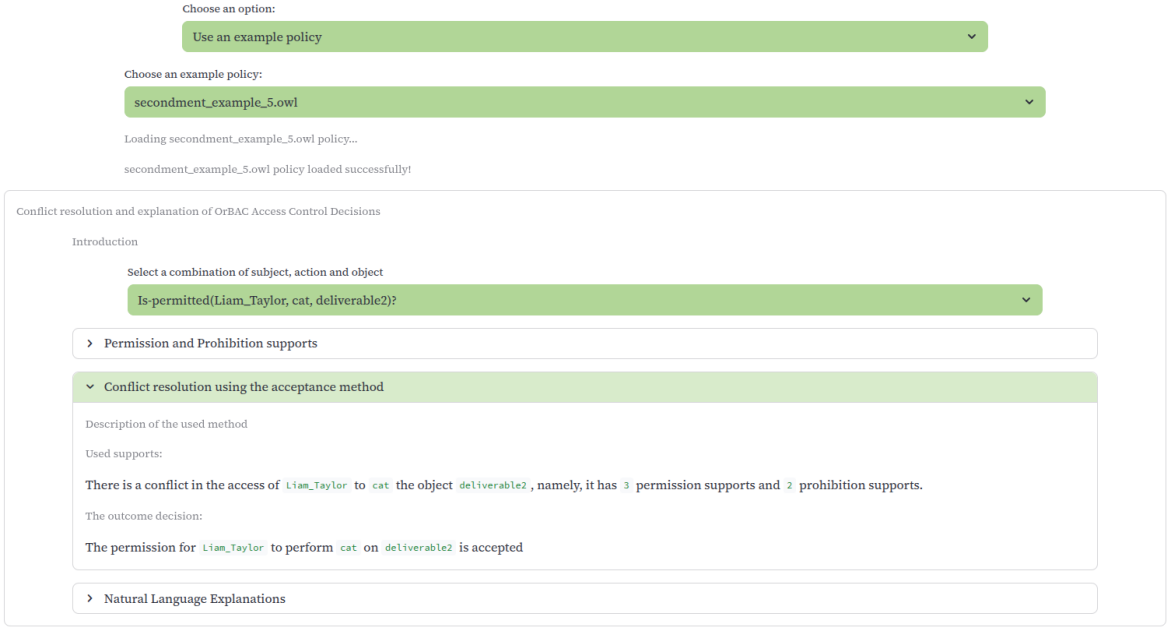


Figure A.6: Decision outcome after conflict resolution.

Generating Natural Language Explanations

Figure A.7 shows the button to prompt the large language model `google/gemma-2-9b-it` (via a HuggingFace token at https://huggingface.co/nebius) to generate a natural language explanation for the selected query.

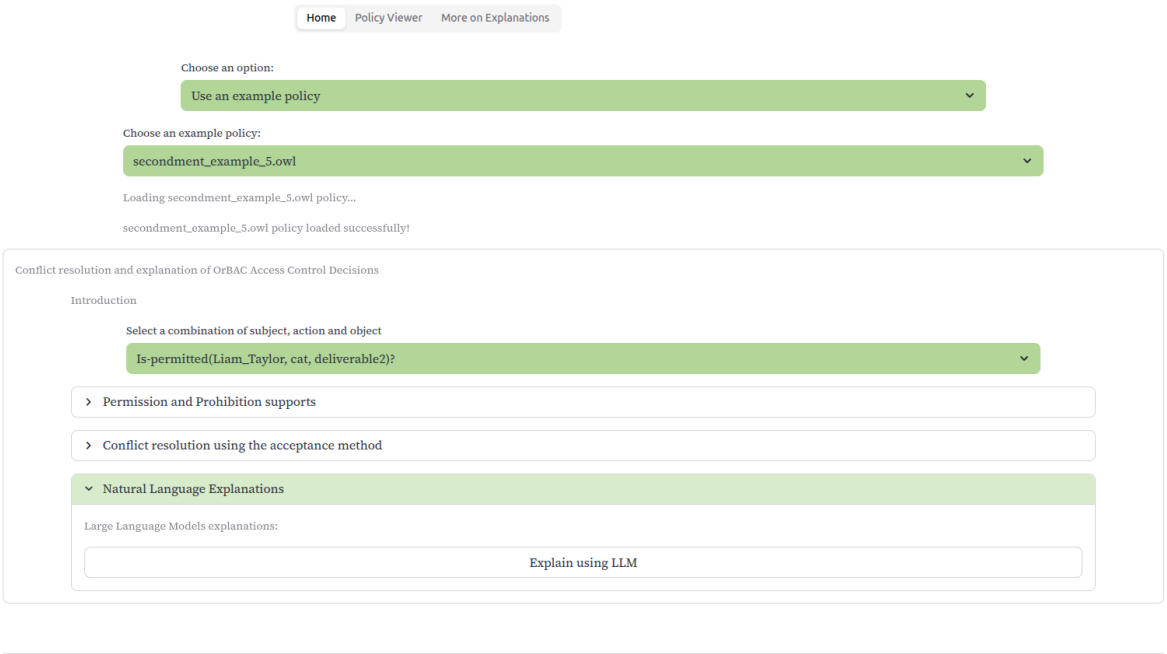


Figure A.7: Button to generate a natural language explanation using `google/gemma-2-9b-it`.

Figure A.8 presents an example explanation output generated by the LLM, describing the access control decision following the conflict-resolution process.

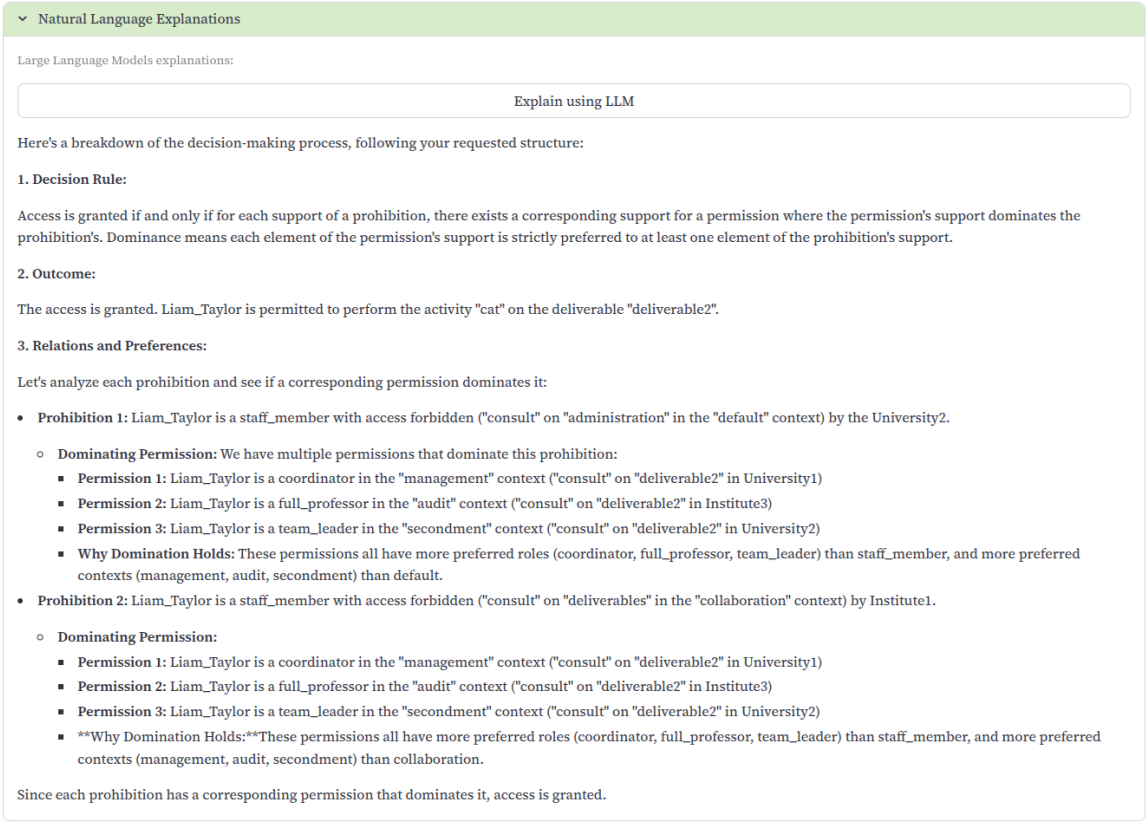


Figure A.8: Natural language explanation produced by the LLM.

Policy Visualisation

Figure A.9 displays the tables summarising the abstract rules of the policy.

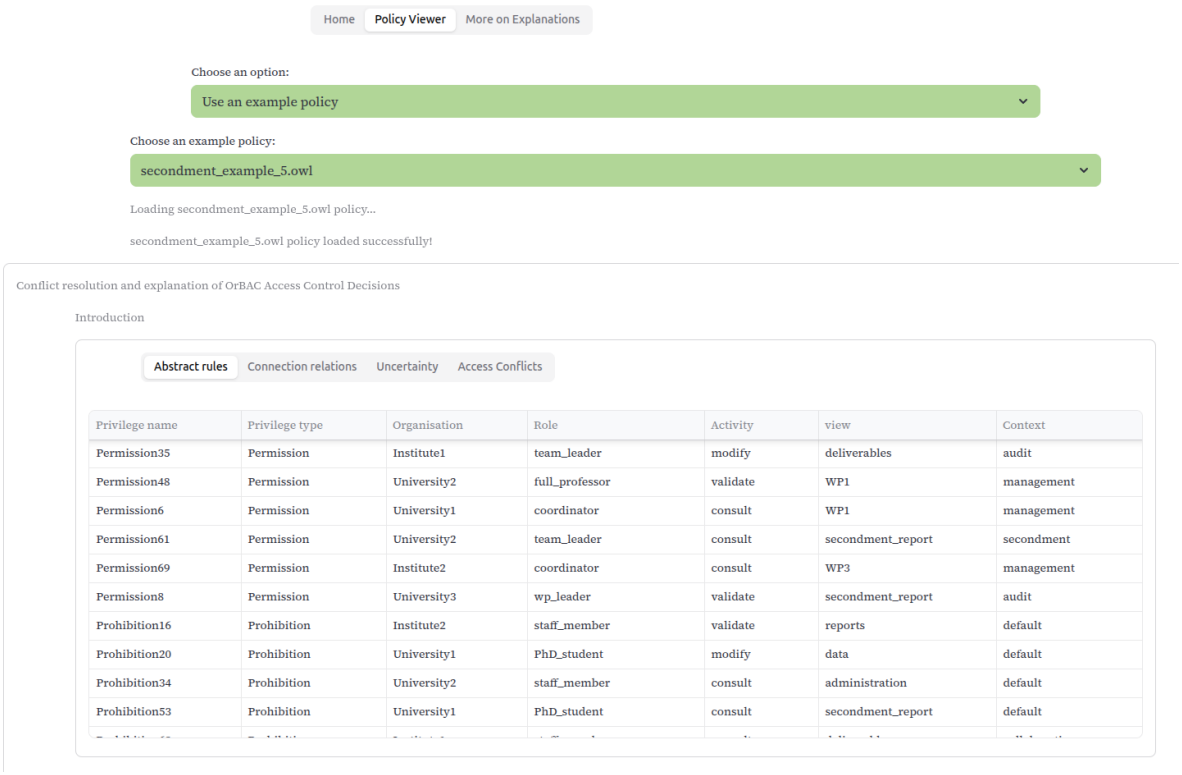
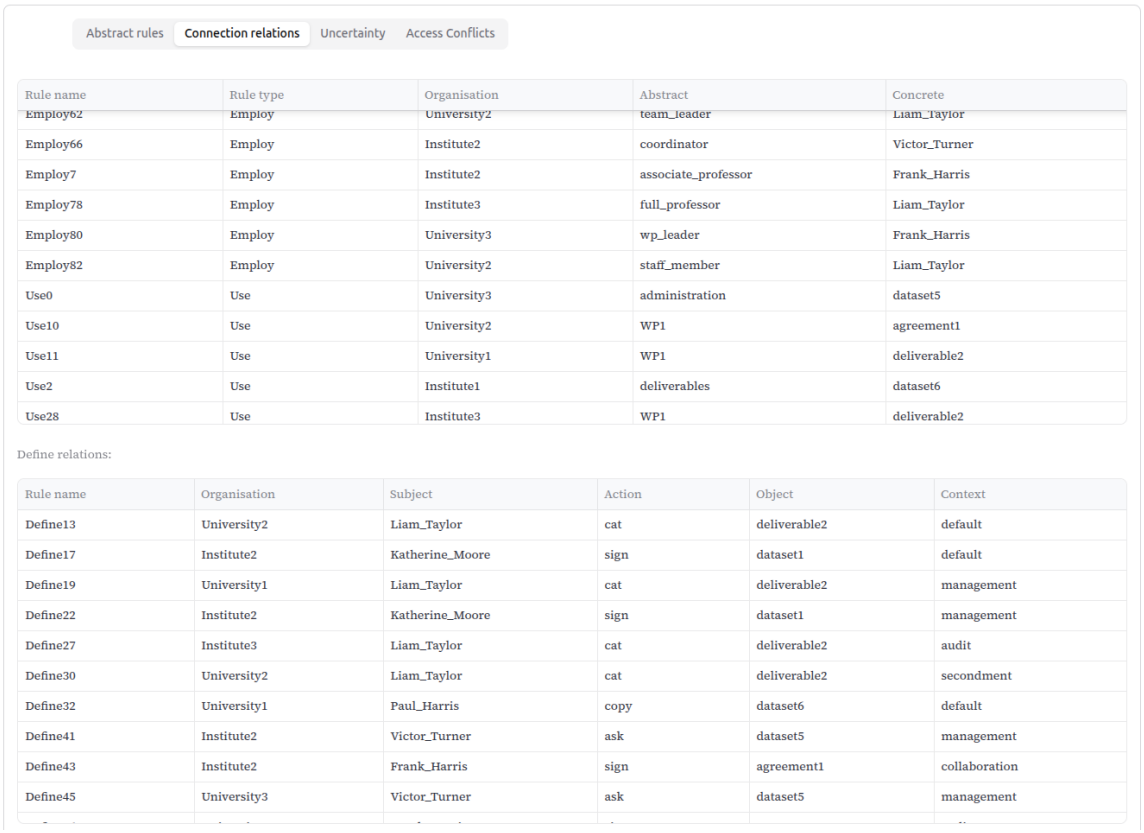


Figure A.9: Visualisation of the abstract rules of the policy.

Figure A.10 shows the tables summarising the connection relations defined in the policy.



Abstract rules | Connection relations | Uncertainty | Access Conflicts

Rule name	Rule type	Organisation	Abstract	Concrete
Employ62	Employ	University2	team_leader	Liam_Taylor
Employ66	Employ	Institute2	coordinator	Victor_Turner
Employ7	Employ	Institute2	associate_professor	Frank_Harris
Employ78	Employ	Institute3	full_professor	Liam_Taylor
Employ80	Employ	University3	wp_leader	Frank_Harris
Employ82	Employ	University2	staff_member	Liam_Taylor
Use0	Use	University3	administration	dataset5
Use10	Use	University2	WP1	agreement1
Use11	Use	University1	WP1	deliverable2
Use2	Use	Institute1	deliverables	dataset6
Use28	Use	Institute3	WP1	deliverable2

Define relations:

Rule name	Organisation	Subject	Action	Object	Context
Define13	University2	Liam_Taylor	cat	deliverable2	default
Define17	Institute2	Katherine_Moore	sign	dataset1	default
Define19	University1	Liam_Taylor	cat	deliverable2	management
Define22	Institute2	Katherine_Moore	sign	dataset1	management
Define27	Institute3	Liam_Taylor	cat	deliverable2	audit
Define30	University2	Liam_Taylor	cat	deliverable2	secondment
Define32	University1	Paul_Harris	copy	dataset6	default
Define41	Institute2	Victor_Turner	ask	dataset5	management
Define43	Institute2	Frank_Harris	sign	agreement1	collaboration
Define45	University3	Victor_Turner	ask	dataset5	management

Figure A.10: Visualisation of the connection relations of the policy.

Consistency Checking and Conflict Computation

Figure A.11 illustrates the functionality to check the consistency of a policy or compute its conflicts, providing insight into potential conflicts within the rules.

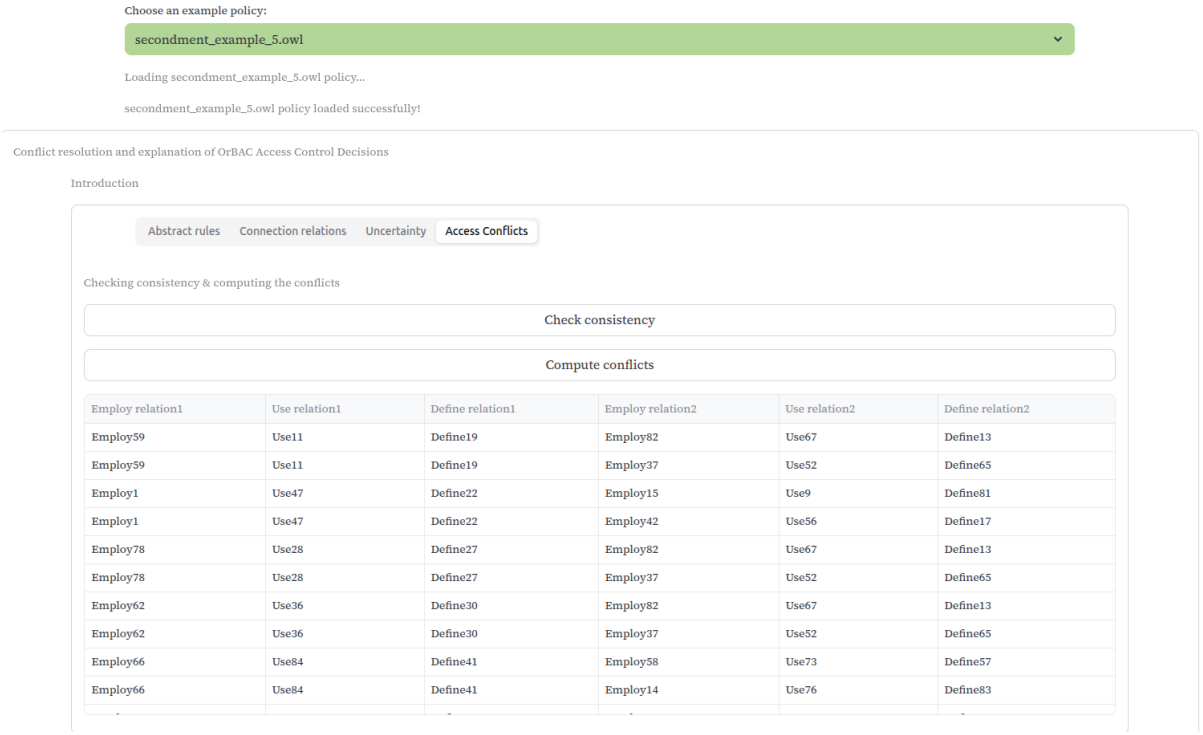


Figure A.11: Checking policy consistency and computing conflicts.

Appendix B

SPARQL queries of the OrBAC ontology

```
ASK {
  ?employ rdf:type orbac:Employ.
  ?employ orbac:employesEmployee :{subject}.
  {?employ orbac:employesEmployer ?org.
  } UNION {
  ?employ orbac:employesEmployer ?org2.
  ?org orbac:subOrganisationOf* ?org2.
  }{
  ?employ orbac:employesRole ?r.
  } UNION {
  ?employ orbac:employesRole ?r2.
  {?r2 rdf:type orbac:SubRole.} UNION  {?r2 rdf:type orbac:SeniorRole.}
  ?r2 orbac:hasParent+ ?r.
  ?r2 orbac:subRoleOrganisation ?org.
  }
  ?use rdf:type orbac:Use.
  ?use orbac:usesObject :{object}.
  ?use orbac:usesView ?v.
  {
  ?use orbac:usesEmployer ?org.
  } UNION {
  ?use orbac:usesEmployer ?org2.
  ?org orbac:subOrganisationOf* ?org2.
  }
  ?consider rdf:type orbac:Consider.
  ?consider orbac:considersAction :{action}.
  ?consider orbac:considersActivity ?a.
  {
  ?consider orbac:considersOrganisation ?org.
  } UNION {
  ?consider orbac:considersOrganisation ?org2.
  ?org orbac:subOrganisationOf* ?org2.
  }
  ?define rdf:type orbac:Define.
  ?define orbac:definesSubject :{subject}.
  ?define orbac:definesAction :{action}.
  ?define orbac:definesObject :{object}.
  ?define orbac:definesContext ?c.
  {
  ?define orbac:definesOrganisation ?org.
  } UNION {
  ?define orbac:definesOrganisation ?org2.
  ?org orbac:subOrganisationOf* ?org2.
  }
  {?perm rdf:type orbac:Permission.} UNION {?perm rdf:type orbac:Obligation.}
  ?perm orbac:accessTypeRole ?r.
  ?perm orbac:accessTypeActivity ?a.
  ?perm orbac:accessTypeView ?v.
  ?perm orbac:accessTypeContext ?c.
  {?perm orbac:accessTypeOrganisation ?org.
  } UNION {
  ?perm orbac:accessTypeOrganisation ?org2.
  ?org orbac:subOrganisationOf* ?org2.
}}
```

Figure B.1: The SPARQL query ls-permitted(subject, action, object)

```
ASK  {
  ?use rdf:type orbac:Use .
  ?use orbac:usesObject :{object}.
  ?use orbac:usesView ?v .
  ?consider rdf:type orbac:Consider .
  ?consider orbac:considersAction :{action} .
  ?consider orbac:considersActivity ?a .
  ?define rdf:type orbac:Define .
  ?define orbac:definesSubject :{subject} .
  ?define orbac:definesAction :{action} .
  ?define orbac:definesObject :{object}.
  ?define orbac:definesContext ?c .
  ?proh rdf:type orbac:Prohibition .
  ?proh orbac:accessTypeRole ?r .
  ?proh orbac:accessTypeActivity ?a .
  ?proh orbac:accessTypeView ?v .
  ?proh orbac:accessTypeContext ?c .
  {?employ orbac:employesEmployer ?org .
  } UNION {
  ?employ orbac:employesEmployer ?org2 .
  ?org orbac:subOrganisationOf* ?org2 .
  }
  {?use orbac:usesEmployer ?org .
  } UNION {
  ?use orbac:usesEmployer ?org2 .
  ?org orbac:subOrganisationOf* ?org2 .
  }
  {?consider orbac:considersOrganisation ?org .
  } UNION {
  ?consider orbac:considersOrganisation ?org2 .
  ?org orbac:subOrganisationOf* ?org2 .
  }{
  ?define orbac:definesOrganisation ?org .
  } UNION {
  ?define orbac:definesOrganisation ?org2 .
  ?org orbac:subOrganisationOf* ?org2 .
  }{
  ?proh orbac:accessTypeOrganisation ?org .
  } UNION {
  ?proh orbac:accessTypeOrganisation ?org2 .
  ?org orbac:subOrganisationOf* ?org2 .
  }
  ?employ rdf:type orbac:Employ .
  ?employ orbac:employesEmployee :{subject} .
  {?employ orbac:employesRole ?r .} UNION  {?employ orbac:employesRole ?r2 .
  {
  ?r2 rdf:type orbac:SubRole .
  ?r2 orbac:hasParent+ ?r .
  ?r2 orbac:subRoleOrganisation ?org .
  } UNION {
  ?r rdf:type orbac:SeniorRole .
  ?r orbac:hasParent+ ?r2 .
  ?r orbac:subRoleOrganisation ?org .}}}
```

Figure B.2: The SPARQL query ls-prohibited(subject, action, object)

```
SELECT DISTINCT ?permission ?employ ?consider ?use ?define
WHERE  {
  ?use rdf:type orbac-owl:Use .
  ?use orbac-owl:usesObject :{object}.
  ?use orbac-owl:usesView ?v .
  ?consider rdf:type orbac-owl:Consider .
  ?consider orbac-owl:considersAction :{action} .
  ?consider orbac-owl:considersActivity ?a .
  ?define rdf:type orbac-owl:Define .
  ?define orbac-owl:definesSubject :{subject} .
  ?define orbac-owl:definesAction :{action} .
  ?define orbac-owl:definesObject :{object}.
  ?define orbac-owl:definesContext ?c .
  {?permission rdf:type orbac-owl:Permission .} UNION {?permission rdf:type orbac-
      owl:Obligation .}
  ?permission orbac-owl:accessTypeRole ?r .
  ?permission orbac-owl:accessTypeActivity ?a .
  ?permission orbac-owl:accessTypeView ?v .
  ?permission orbac-owl:accessTypeContext ?c .
  {
  ?employ orbac-owl:employesEmployer ?org .
  } UNION  {
  ?employ orbac-owl:employesEmployer ?org2 .
  ?org orbac-owl:subOrganisationOf+ ?org2 .
  }{
  ?use orbac-owl:usesEmployer ?org .
  } UNION  {
  ?use orbac-owl:usesEmployer ?org2 .
  ?org orbac-owl:subOrganisationOf+ ?org2 .
  }{
  ?consider orbac-owl:considersOrganisation ?org .
  } UNION  {
  ?consider orbac-owl:considersOrganisation ?org2 .
  ?org orbac-owl:subOrganisationOf+ ?org2 .
  }{
  ?define orbac-owl:definesOrganisation ?org .
  } UNION  {
  ?define orbac-owl:definesOrganisation ?org2 .
  ?org orbac-owl:subOrganisationOf+ ?org2 .
  }{
  ?permission orbac-owl:accessTypeOrganisation ?org .
  } UNION  {
  ?permission orbac-owl:accessTypeOrganisation ?org2 .
  ?org orbac-owl:subOrganisationOf+ ?org2 .
  }
  ?employ rdf:type orbac-owl:Employ .
  ?employ orbac-owl:employesEmployee :{subject} .
  {
  ?employ orbac-owl:employesRole ?r .
  } UNION  {
  ?employ orbac-owl:employesRole ?r2 .
  {?r2 rdf:type orbac-owl:SubRole .} UNION  {?r2 rdf:type orbac-owl:SeniorRole .}
  ?r2 orbac-owl:hasParent+ ?r .
  ?r2 orbac-owl:subRoleOrganisation ?org .}}
```

Figure B.3: The permission_supports(subject, action, object) SPARQL query

```
SELECT DISTINCT ?prohibition ?employ ?consider ?use ?define
WHERE  {
  ?use rdf:type orbac-owl:Use .
  ?use orbac-owl:usesObject :{object}.
  ?use orbac-owl:usesView ?v .
  ?consider rdf:type orbac-owl:Consider .
  ?consider orbac-owl:considersAction :{action} .
  ?consider orbac-owl:considersActivity ?a .
  ?define rdf:type orbac-owl:Define .
  ?define orbac-owl:definesSubject :{subject} .
  ?define orbac-owl:definesAction :{action} .
  ?define orbac-owl:definesObject :{object}.
  ?define orbac-owl:definesContext ?c .
  ?prohibition rdf:type orbac-owl:Prohibition .
  ?prohibition orbac-owl:accessTypeRole ?r .
  ?prohibition orbac-owl:accessTypeActivity ?a .
  ?prohibition orbac-owl:accessTypeView ?v .
  ?prohibition orbac-owl:accessTypeContext ?c .
  {
  ?employ orbac-owl:employesEmployer ?org .
  } UNION  {
  ?employ orbac-owl:employesEmployer ?org2 .
  ?org orbac-owl:subOrganisationOf+ ?org2 .
  }{
  ?use orbac-owl:usesEmployer ?org .
  } UNION  {
  ?use orbac-owl:usesEmployer ?org2 .
  ?org orbac-owl:subOrganisationOf+ ?org2 .
  }{
  ?consider orbac-owl:considersOrganisation ?org .
  } UNION  {
  ?consider orbac-owl:considersOrganisation ?org2 .
  ?org orbac-owl:subOrganisationOf+ ?org2 .
  }{
  ?define orbac-owl:definesOrganisation ?org .
  } UNION  {
  ?define orbac-owl:definesOrganisation ?org2 .
  ?org orbac-owl:subOrganisationOf+ ?org2 .
  }{
  ?prohibition orbac-owl:accessTypeOrganisation ?org .
  } UNION  {
  ?prohibition orbac-owl:accessTypeOrganisation ?org2 .
  ?org orbac-owl:subOrganisationOf+ ?org2 .
  }
  ?employ rdf:type orbac-owl:Employ .
  ?employ orbac-owl:employesEmployee :{subject} .
  {?employ orbac-owl:employesRole ?r .} UNION  {?employ orbac-owl:employesRole ?r2
      .{
  ?r2 rdf:type orbac-owl:SubRole .
  ?r2 orbac-owl:hasParent+ ?r .
  ?r2 orbac-owl:subRoleOrganisation ?org .
  } UNION {
  ?r rdf:type orbac-owl:SeniorRole .
  ?r orbac-owl:hasParent+ ?r2 .
  ?r orbac-owl:subRoleOrganisation ?org .}}}
```

Figure B.4: The prohibition_supports(subject, action, object) SPARQL query

```
ASK {
  # First check that at least one permission exists # --- (Permission or Obligation)
      ---
  ?use rdf:type orbac:Use .
  ?use orbac:usesObject :{object}.
  ?use orbac:usesView ?v .
  ?consider rdf:type orbac:Consider .
  ?consider orbac:considersAction :{action} .
  ?consider orbac:considersActivity ?a .
  ?define rdf:type orbac:Define .
  ?define orbac:definesSubject :{subject} .
  ?define orbac:definesAction :{action} .
  ?define orbac:definesObject :{object}.
  ?define orbac:definesContext ?c .
  {?perm rdf:type orbac:Permission .} UNION {?perm rdf:type orbac:Obligation .}
  ?perm orbac:accessTypeRole ?r .
  ?perm orbac:accessTypeActivity ?a .
  ?perm orbac:accessTypeView ?v .
  ?perm orbac:accessTypeContext ?c .
  {
  ?employ orbac:employesEmployer ?org .
  } UNION   {
  ?employ orbac:employesEmployer ?org2 .
  ?org orbac:subOrganisationOf+ ?org2 .
  }
  {
  ?use orbac:usesEmployer ?org .
  } UNION   {
  ?use orbac:usesEmployer ?org2 .
  ?org orbac:subOrganisationOf+ ?org2 .
  }
  {
  ?consider orbac:considersOrganisation ?org .
  } UNION   {
  ?consider orbac:considersOrganisation ?org2 .
  ?org orbac:subOrganisationOf+ ?org2 .
  }
  {
  ?define orbac:definesOrganisation ?org .
  } UNION   {
  ?define orbac:definesOrganisation ?org2 .
  ?org orbac:subOrganisationOf+ ?org2 .
  }
  {
  ?perm orbac:accessTypeOrganisation ?org .
  } UNION   {
  ?perm orbac:accessTypeOrganisation ?org2 .
  ?org orbac:subOrganisationOf+ ?org2 .
  }
  ?employ rdf:type orbac:Employ .
  ?employ orbac:employesEmployee :{subject} .
  ?employ orbac:employesRole ?r .

  {
    FILTER EXISTS {
      # Prohibition pattern
      ?useBiss rdf:type orbac:Use .
      ?useBiss orbac:usesObject :{object}.
```

```
?useBiss orbac:usesView ?vBiss .
?considerBiss rdf:type orbac:Consider .
?considerBiss orbac:considersAction :{action} .
?considerBiss orbac:considersActivity ?aBiss .
?defineBiss rdf:type orbac:Define .
?defineBiss orbac:definesSubject :{subject} .
?defineBiss orbac:definesAction :{action} .
?defineBiss orbac:definesObject :{object}.
?defineBiss orbac:definesContext ?cBiss .
?proh rdf:type orbac:Prohibition .
?proh orbac:accessTypeRole ?rBiss .
?proh orbac:accessTypeActivity ?aBiss .
?proh orbac:accessTypeView ?vBiss .
?proh orbac:accessTypeContext ?cBiss .
{
?employBiss orbac:employesEmployer ?orgBiss .
} UNION   {
?employBiss orbac:employesEmployer ?orgBiss2 .
?orgBiss orbac:subOrganisationOf+ ?orgBiss2 .
}
{
?useBiss orbac:usesEmployer ?orgBiss .
} UNION   {
?useBiss orbac:usesEmployer ?orgBiss2 .
?orgBiss orbac:subOrganisationOf+ ?orgBiss2 .
}
{
?considerBiss orbac:considersOrganisation ?orgBiss .
} UNION   {
?considerBiss orbac:considersOrganisation ?orgBiss2 .
?orgBiss orbac:subOrganisationOf+ ?orgBiss2 .
}
{
?defineBiss orbac:definesOrganisation ?orgBiss .
} UNION   {
?defineBiss orbac:definesOrganisation ?orgBiss2 .
?orgBiss orbac:subOrganisationOf+ ?orgBiss2 .
}
{
?proh orbac:accessTypeOrganisation ?orgBiss .
} UNION   {
?proh orbac:accessTypeOrganisation ?orgBiss2 .
?orgBiss orbac:subOrganisationOf+ ?orgBiss2 .
}
?employBiss rdf:type orbac:Employ .
?employBiss orbac:employesEmployee :{subject} .
?employBiss orbac:employesRole ?rBiss .
# With strict domination
FILTER EXISTS {
  # Strictly dominating permission pattern
  ?use rdf:type orbac:Use .
  ?use orbac:usesObject :{object}.
  ?use orbac:usesView ?v .
  ?consider rdf:type orbac:Consider .
  ?consider orbac:considersAction :{action} .
  ?consider orbac:considersActivity ?a .
  ?define rdf:type orbac:Define .
  ?define orbac:definesSubject :{subject} .
  ?define orbac:definesAction :{action} .
```

```
      ?define orbac:definesObject :{object}.
      ?define orbac:definesContext ?c .
      { ?perm rdf:type orbac:Permission . } UNION { ?perm rdf:type orbac:
          Obligation . }
      ?perm orbac:accessTypeRole ?r .
      ?perm orbac:accessTypeActivity ?a .
      ?perm orbac:accessTypeView ?v .
      ?perm orbac:accessTypeContext ?c .
      {
      ?employ orbac:employesEmployer ?org .
      } UNION {
      ?employ orbac:employesEmployer ?org2 .
      ?org orbac:subOrganisationOf+ ?org2 .
      }
      {
      ?use orbac:usesEmployer ?org .
      } UNION {
      ?use orbac:usesEmployer ?org2 .
      ?org orbac:subOrganisationOf+ ?org2 .
      }
      {
      ?consider orbac:considersOrganisation ?org .
      } UNION {
      ?consider orbac:considersOrganisation ?org2 .
      ?org orbac:subOrganisationOf+ ?org2 .
      }
      {
      ?define orbac:definesOrganisation ?org .
      } UNION {
      ?define orbac:definesOrganisation ?org2 .
      ?org orbac:subOrganisationOf+ ?org2 .
      }
      {
      ?perm orbac:accessTypeOrganisation ?org .
      } UNION {
      ?perm orbac:accessTypeOrganisation ?org2 .
      ?org orbac:subOrganisationOf+ ?org2 .
      }
      ?employ rdf:type orbac:Employ .
      ?employ orbac:employesEmployee :{subject} .
      ?employ orbac:employesRole ?r .

      ?employ orbac:isPreferredTo ?employBiss .
      FILTER NOT EXISTS { ?employBiss orbac:isPreferredTo ?employ . }
      ?define orbac:isPreferredTo ?defineBiss .
      FILTER NOT EXISTS { ?defineBiss orbac:isPreferredTo ?define . }
    }
  }
} UNION {
  FILTER NOT EXISTS {
  # Prohibition pattern
  ?useBiss rdf:type orbac:Use .
  ?useBiss orbac:usesObject :{object}.
  ?useBiss orbac:usesView ?vBiss .
  ?considerBiss rdf:type orbac:Consider .
  ?considerBiss orbac:considersAction :{action} .
  ?considerBiss orbac:considersActivity ?aBiss .
  ?defineBiss rdf:type orbac:Define .
  ?defineBiss orbac:definesSubject :{subject} .
```

```
    ?defineBiss orbac:definesAction :{action} .
    ?defineBiss orbac:definesObject :{object}.
    ?defineBiss orbac:definesContext ?cBiss .
    ?proh rdf:type orbac:Prohibition .
    ?proh orbac:accessTypeRole ?rBiss .
    ?proh orbac:accessTypeActivity ?aBiss .
    ?proh orbac:accessTypeView ?vBiss .
    ?proh orbac:accessTypeContext ?cBiss .
    {
    ?employBiss orbac:employesEmployer ?orgBiss .
    } UNION   {
    ?employBiss orbac:employesEmployer ?orgBiss2 .
    ?orgBiss orbac:subOrganisationOf+ ?orgBiss2 .
    }
    {
    ?useBiss orbac:usesEmployer ?orgBiss .
    } UNION   {
    ?useBiss orbac:usesEmployer ?orgBiss2 .
    ?orgBiss orbac:subOrganisationOf+ ?orgBiss2 .
    }
    {
    ?considerBiss orbac:considersOrganisation ?orgBiss .
    } UNION   {
    ?considerBiss orbac:considersOrganisation ?orgBiss2 .
    ?orgBiss orbac:subOrganisationOf+ ?orgBiss2 .
    }
    {
    ?defineBiss orbac:definesOrganisation ?orgBiss .
    } UNION   {
    ?defineBiss orbac:definesOrganisation ?orgBiss2 .
    ?orgBiss orbac:subOrganisationOf+ ?orgBiss2 .
    }
    {
    ?proh orbac:accessTypeOrganisation ?orgBiss .
    } UNION   {
    ?proh orbac:accessTypeOrganisation ?orgBiss2 .
    ?orgBiss orbac:subOrganisationOf+ ?orgBiss2 .
    }
    ?employBiss rdf:type orbac:Employ .
    ?employBiss orbac:employesEmployee :{subject} .
    ?employBiss orbac:employesRole ?rBiss .
    }
  }
}
```

Figure B.5: The acceptance(subject, action, object) SPARQL query

Résumé

La gestion des informations contradictoires pose des défis importants en matière de représentation des connaissances et de raisonnement. L'incohérence peut être le résultat de la fusion d'informations extraites de différentes sources ou de la présence de règles comportant des exceptions. Cependant, le raisonnement à partir de bases de connaissances incohérentes n'est pas trivial, car les contradictions peuvent conduire à des déductions arbitraires. En logique propositionnelle, toute conclusion peut être déduite d'une contradiction. Par conséquent, il est essentiel d'utiliser des mécanismes appropriés de traitement des incohérences pour permettre le raisonnement en présence d'informations incohérentes ou contradictoires.

Le problème de gestion des informations contradictoires a été étudié dans plusieurs contextes dans la littérature [1,2,3]. L'un d'entre eux est celui des informations partiellement ordonnées, où certains éléments d'information peuvent avoir des degrés de fiabilité incomparables. En conséquence, il n'est pas possible de décider quel élément d'information doit être sélectionné en présence d'incohérence, ce qui fait du raisonnement une tâche difficile. Peu de méthodes éfficaces ont été proposées pour faire face à cette situation spécifique. En outre, le traitement des incompatibilités peut entraîner la perte de certaines conséquences, ce qui fait que certaines réponses aux requêtes sont manquantes. L'objectif de cette thèse est d'étudier et d'étendre les sémantiques tolérantes à l'incohérence. Plus précisément, nous prévoyons de proposer de nouvelles méthodes pour traiter efficacement l'incohérence dans le cas d'informations partiellement ordonnées.

Nous prévoyons également d'étudier l'application de ces méthodes pour gérer les conflits dans les modèles de contrôle d'accès. Les modèles de contrôle d'accès, qui sont des moyens très importants de protection des données à caractère personnel, constituent une application de ces méthodes. Dans ce cas, des incohérences peuvent survenir en raison de l'utilisation simultanée de règles de permission et d'interdiction. De nombreux modèles ont été proposés pour gérer ce type de conflits. Cependant, soit ils procèdent à l'ajout d'un grand nombre de contraintes pour éviter les conflits, soit ils nécessitent une assistance humaine pour les résoudre. En outre, ces modèles n'ont pas la capacité de fournir des explications sur les décisions prises. L'objectif de l'application d'une sémantique tolérante à l'incohérence dans ce cas est d'abord d'assurer l'ajout et la modification continus et sûrs de règles dans un modèle, tout en gérant l'incohérence, et ensuite de doter ces modèles de mécanismes d'explicabilité. Finalement, nous proposons d'étendre ces solutions de gestion des politiques de sécurité afin d'intégrer la notion d'explicabilité.

Références:

[1] Meghyn Bienvenu and Camille Bourgaux. Inconsistency-tolerant querying of description logic knowledge bases. In Reasoning Web: Logical Foundation of Knowledge Graph Construction and Query Answering, volume 9885, pages 156-202. LNCS. Springer, 2016.

[2] Jean-François Baget, Salem Benferhat, Zied Bouraoui, Madalina Croitoru, Marie-Laure Mugnier, Odile Papini, Swan Rocher, and Karim Tabia. Inconsistency-tolerant query answering: Rationality properties and computational complexity analysis. In European Conference Logics in Artificial Intelligence, JELIA, volume 10021 of LNCS, pages 64-80, 2016.

[3] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-tolerant semantics for description logics. In Web Reasoning and Rule

Systems, RR 2010, Bressanone/Brixen, Italy, pages 103-117, 2010.

Mots-clés: Ontologies légères, Ordres partiels, Réparation des données, Gestion des incohérences, Contrôle d'accès, Représentation de connaissances, IA explicable.

Abstract

The management of contradictory information presents significant challenges in knowledge representation and reasoning. Inconsistency can arise from merging information retrieved from different sources or from the presence of rules with exceptions. However, reasoning from inconsistent knowledge bases is not trivial, as contradictions can lead to arbitrary deductions. In propositional logic, any conclusion can be deduced from a contradiction. As a consequence, it is essential to use appropriate inconsistency handling mechanisms to enable reasoning in the presence of inconsistent or conflicting information.

The literature has studied the problem of contradictory information in several contexts [1,2,3]. One of them is that of partially ordered information, where some pieces of information might have incomparable degrees of reliability. As a result, one cannot decide which piece of information to select in the presence of inconsistency, making reasoning a challenging task. Few tractable methods were proposed to deal with this specific situation. Furthermore, dealing with incomparabilities might cause the loss of some consequences, leading to missing some query answers. The aim of this thesis is to investigate and extend tractable inconsistency-tolerant semantics. More specifically, we plan to propose new methods to efficiently handle inconsistency in the case of partially ordered information.

We also plan to investigate the application of such methods to handle conflicts in access control models. Access control models are very important means of protecting personal data. In this case, inconsistency may arise due to the simultaneous use of permission and prohibition rules. Many models were proposed to handle this type of conflicts. However, they either proceed to add a large set of constraints to avoid the conflicts, or require human assistance to overcome them. Moreover, these models lack the capacity to provide explanations for the taken decisions. The aim of applying inconsistency-tolerant semantics in this case is to first ensure the continuous and safe addition and modification of rules in a model, while dealing with inconsistency, and then equip these models with explainability mechanisms. Finally, we propose to extend these security policy management solutions to incorporate the notion of explainability.
References:
[1] Meghyn Bienvenu and Camille Bourgaux. Inconsistency-tolerant querying of description logic knowledge bases. In Reasoning Web: Logical Foundation of Knowledge Graph Construction and Query Answering, volume 9885, pages 156-202. LNCS. Springer, 2016.
[2] Jean-François Baget, Salem Benferhat, Zied Bouraoui, Madalina Croitoru, Marie-Laure Mugnier, Odile Papini, Swan Rocher, and Karim Tabia. Inconsistency-tolerant query answering: Rationality properties and computational complexity analysis. In European Conference Logics in Artificial Intelligence, JELIA, volume 10021 of LNCS, pages 64-80, 2016.
[3] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-tolerant semantics for description logics. In Web Reasoning and Rule Systems, RR 2010, Bressanone/Brixen, Italy, pages 103-117, 2010.

Keywords: Lightweight ontologies, Partial orders, Data Repairs, Inconsistency management, Access control, Knowledge Representation, Explainable AI.