UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

PH.D. THESIS IN

INFORMATION TECHNOLOGY AND ELECTRICAL
ENGINEERING

OPTIMIZATION OF NONSTANDARD REASONING SERVICES

Iliana Mineva Petrova

Tutor: Prof. Piero A. Bonatti

XXIX CICLO
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE
DELL'INFORMAZIONE

To my angels -
my mother and grandmother

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor Piero A. Bonatti. Piero has shown tremendous support not only in the form of academic guidance but also in encouraging me when the going got tough. Thank you also to Luigi Sauro who helped shape a timid undergrad into an academic researcher.

I would also like to thank my reviewers, Laura Giordano, Birte Glimm and Uli Sattler for their patience, insightful feedback and helpful suggestions for improving my thesis.

I am very grateful to Yevgeny Kazakov and the other members of the Institute of Artificial Intelligence at the University of Ulm. Yevgeny has been an inspiration on how to conduct practical research in my area. In particular, I would like to thank Marvin, Peter, Kien, Markus and Sylvia who really went out of their way to make me feel welcome during my visit.

Back in Naples, I would like to thank Anna Corazza for her heartfelt advices during our long conversations.

Finally, a very special appreciation goes to my fiancé Gerardo for his unconditional love, understanding and support throughout this PhD - thanks for putting up with me.

Contents

List of Acronyms

The following acronyms are used throughout this text.

DL	Description Logic
FOL	First Order Logic
OWL	Web Ontology Language
RDF	Resource Description Framework
W3C	World Wide Web Consortium
OBDA	Ontology Based Data Access
RDBMS	Relational Database Management System

Chapter 1

Introduction

The increasing adoption of semantic technologies and the corresponding increasing complexity of application requirements are motivating extensions to the standard reasoning paradigms and services supported by such technologies. This thesis focuses on two of such extensions: nonmonotonic reasoning and inference-proof access control.

Concerning the former, expressing knowledge via general rules that admit exceptions is an approach that has been commonly adopted for centuries in areas such as law and science, and more recently in object-oriented programming and computer security. The experiences in developing complex biomedical knowledge bases reported in the literature show that a direct support to defeasible properties and exceptions would be of great help.

Concerning access control, there is ample evidence of the need for knowledge confidentiality measures. Ontology languages and Linked Open Data are increasingly being used to encode the private knowledge of companies and public organizations. Semantic Web techniques facilitate merging different sources of knowledge and extract implicit information, thereby putting at risk security and the privacy of individuals. But the same reasoning capabilities can be exploited to protect the confidentiality of knowledge.

Both nonmonotonic inference and secure knowledge base access rely on non-

standard reasoning procedures. This thesis is mainly about the design and realization of these algorithms in a scalable way (appropriate to the ever-increasing size of ontologies and knowledge bases), by means of a diversified range of optimization techniques. The thesis is organized as follows:

- Chapter 2 introduces the nomenclature, key ideas and definitions that are used in the rest of thesis. First the foundations of ontology languages – description logics (DLs) – their syntax and semantics, expressiveness as well as some relevant reasoning tasks are presented. Then, we briefly describe the W3C standardized language OWL (Web Ontology Language) underpinned by DLs. In section 2.4 we discuss different algorithmic approaches to reasoning with knowledge bases formalized with such languages. Finally, we give an overview over the most important state-of-the-art optimizations that are essential for achieving well-performing reasoning systems in practice.

- Chapter 3 introduces the two extensions of the standard reasoning paradigms this thesis focuses on. In particular, Section 3.1 contributes to the practical support of nonmonotonic inferences in description logics by introducing a new semantics expressly designed to model priority-based overriding. In this way, we obtain a formalism with nice logical and computational properties that constitutes an appealing solution to a large class of application needs. Section 3.2 introduce a new confidentiality model, sensitive enough to detect several novel attacks to the confidentiality of knowledge bases (KB), and a method for constructing secure KB views. Safe approximations of the background knowledge exploited in the attacks are identified that can be used to reduce the complexity of constructing of such views.

- Chapter 4 introduce optimization techniques for reasoning in $\mathscr{DL}^{\mathrm{N}}$. Module extraction algorithms can quickly select the axioms of a knowledge base that must be considered in order to answer any query formulated in a given signature of interest. We investigate the use of *module extractors* [Sattler et al., 2009] to focus reasoning on relevant axioms only. The approach is not trivial (module extractors are unsound for most nonmonotonic logics, including circumscription, default and autoepistemic logics) and re-

quires an articulated correctness proof. We further address cases in which module extraction techniques should be improved. Currently, module extraction methods are less effective when the knowledge base has nonempty ABoxes; this phenomenon is amplified in the nonmonotonic description logic $\mathscr{DL}^{\mathrm{N}}$, where reasoning requires repeated classifications of the knowledge base. We meet this point by introducing (and proving the correctness of): a "conditionally correct" module extractor for nonempty ABoxes. We further introduce a new algorithm for query answering, that is expected to exploit incremental reasoners at their best. Incremental reasoning is crucial as $\mathscr{DL}^{\mathrm{N}}$'s reasoning method iterates consistency tests on a set of KBs with large intersections. While the assertion of new axioms is processed very efficiently, the computational cost of axiom deletion is generally not negligible. The *optimistic reasoning method* described in Section 4.3 is expected to reduce the number of deletions. Such optimizations are validated experimentally through systematic scalability tests on large KBs with more than 30K axioms. Speedups exceed 1 order of magnitude. For the first time, response times compatible with real-time reasoning are obtained with nonmonotonic KBs of this size. A test case generator introduced in Section 4.1.2 and its novel validation method constitute a further contribution of this thesis.

- Chapter 5 illustrates SOVGen, a first implementation of the knowledge base confidentiality model that has been specialized to deal with a concrete e-health application. In order to maximize performance, we design several optimization techniques – module extraction and ad-hoc conjunctive query evaluation – and assess them experimentally by using realistic electronic health records that refer to real world biomedical ontologies (eg. SNOMED-CT). Considering that secure views are constructed off-line, performance analysis shows that SOVGen is already compatible with practical use in this application scenario.

- Chapter 6 summarizes the contribution of this thesis and discuss some interesting directions for further research.

The contents of the thesis have been partially published in several papers:

- [Bonatti et al., 2015a, Bonatti et al., 2017] report the new semantics for overriding in DLs in Section 3.1 of Chapter 3.
- [Bonatti et al., 2015c, Bonatti et al., 2015d] partially report the optimization techniques presented in Chapter 4.
- [Bonatti et al., 2014, Bonatti et al., 2015b] report the optimization techniques presented in Chapter 5.

I was the primary author for Publications [Bonatti et al., 2015a, Bonatti et al., 2017] in which I contributed somewhat to the theoretical aspects by validating the definitions and theorems fixing a few drawback and errors but more heavily in practical implementation and evaluation of the theoretical framework. The publications [Bonatti et al., 2015c, Bonatti et al., 2015d, Bonatti et al., 2015b] contain the main personal contributions described in Chapters 4 and 5 which contain in part work that is under revision. For publication [Bonatti et al., 2014] I was responsible for the implementation and experimental evaluation (and write-up thereof) of the algorithms central to the work.

Chapter 2

Preliminaries

The aim of this chapter is to introduce the nomenclature, key ideas and definitions that are used in the rest of thesis. We first present the foundations of description logics languages (DLs), their syntax and semantics, expressiveness as well as some relevant reasoning tasks. Then, we briefly describe the W3C standardized language OWL (Web Ontology Language) underpinned by DLs. In Section 2.4 we discuss different algorithmic approaches to reasoning with knowledge bases formalized with such languages. Finally, we give an overview over the most important state-of-the-art optimizations that are essential for achieving well-performing reasoning systems in practice.

We assume the reader to be familiar with the basic definitions for First Order Logic (FOL) and the associated model-theoretic semantics. For a detailed presentation of the main topics concerning DLs we refer the reader to "The Description Logic Handbook, Theory, Implementation, and Applications (2nd ed.)" [Baader et al., 2010].

2.1 Description Logics

Description logics are a family of knowledge representation formalisms that provide means to model the relationships between entities in a concrete domain

of interest. Generally speaking, they constitute decidable fragments of first-order logic or slight extensions thereof. The two base features that distinguish them from FOL are: (i) a special more concise and variable free syntax particularly suitable to provide high level model primitives; and (ii) the existence of practical decision procedures for key inference problems.

The basic building blocks of DLs are:

- a set $\mathsf{N_C}$ of **concept names**, also called atomic concepts, that correspond to unary predicates in FOL and are used to describe sets of objects characterized by some common properties;

- a set $\mathsf{N_R}$ of **role names**, also called atomic roles, corresponding to binary predicates in FOL and are used to describe relationships between objects; and possibly

- a set $\mathsf{N_I}$ of **individual names**, also called individuals, that correspond to FOL constants, and are used to denote concrete objects in a domain of interest.

Syntactically, the vocabulary of DLs is obtained starting form the countably infinite sets N_C, N_R and N_I with the help of particular logical symbols called constructors which allow to inductively define complex concepts and roles. The expressive power of a description logic can be identified by the different sets of concept and role constructors it allow. In the following A, B will range over concept names, C and D over (possibly compound) concepts, R and S over roles, and a, b, d and e over individual names.

Table 2.1. Syntax and semantics of DL constructs.

	Syntax	**Semantics**	**Identifier**
Concepts :			
Top	$\top$	$\Delta^{\mathscr{I}}$	
Bottom	$\bot$	$\emptyset$	
Nominals	$\{a\}$	$\{a^{\mathscr{I}}\}$	$\mathscr{O}$
Full negation	$\neg C$	$\Delta^{\mathscr{I}} \setminus C^{\mathscr{I}}$	$\mathscr{C}$
Conjunction	$C \sqcap D$	$C^{\mathscr{I}} \cap D^{\mathscr{I}}$	
Disjunction	$C \sqcup D$	$C^{\mathscr{I}} \cup D^{\mathscr{I}}$	$\mathscr{U}$
Existential restriction	$\exists R.C$	$\{d \in \Delta^{\mathscr{I}} \mid \exists e \in \Delta^{\mathscr{I}}.[(d,e) \in R^{\mathscr{I}} \wedge e \in C^{\mathscr{I}}]\}$	$\mathscr{E}$
Universal restriction	$\forall R.C$	$\{d \in \Delta^{\mathscr{I}} \mid \forall e \in \Delta^{\mathscr{I}}.[(d,e) \in R^{\mathscr{I}} \rightarrow e \in C^{\mathscr{I}}]\}$	
Self restriction	$\exists R.Self$	$\{d \in \Delta^{\mathscr{I}} \mid (d,d) \in R^{\mathscr{I}}]\}$	
Qualified	$\leq nR.C$	$\{d \in \Delta^{\mathscr{I}} \mid \#\{e \in \Delta^{\mathscr{I}}.[(d,e) \in R^{\mathscr{I}} \wedge e \in C^{\mathscr{I}}]\} \leq n\}$	
number	$\geq nR.C$	$\{d \in \Delta^{\mathscr{I}} \mid \#\{e \in \Delta^{\mathscr{I}}.[(d,e) \in R^{\mathscr{I}} \wedge e \in C^{\mathscr{I}}]\} \geq n\}$	
restriction	$= nR.C$	$\{d \in \Delta^{\mathscr{I}} \mid \#\{e \in \Delta^{\mathscr{I}}.[(d,e) \in R^{\mathscr{I}} \wedge e \in C^{\mathscr{I}}]\} = n\}$	$\mathscr{Q}$
Roles :			
Universal role	U	$\Delta^{\mathscr{I}} \times \Delta^{\mathscr{I}}$	
Inverse role	R^{-}	$\{(d,e) \mid (e,d) \in R^{\mathscr{I}}\}$	$\mathscr{I}$

Similar to any other logic, the formal semantics of description logics is given in a model-theoretic way by an *interpretation*[1] $\mathscr{I} = (\Delta^{\mathscr{I}}, \cdot^{\mathscr{I}})$, where the *domain* $\Delta^{\mathscr{I}}$ is a non-empty set of individuals and the *interpretation function* $\cdot^{\mathscr{I}}$ maps each concept name $A \in \mathsf{N_C}$ to a subset $A^{\mathscr{I}}$ of $\Delta^{\mathscr{I}}$, each role name $R \in \mathsf{N_R}$ to a binary relation $R^{\mathscr{I}}$ on $\Delta^{\mathscr{I}}$, and each individual name $a \in \mathsf{N_I}$ to an individual $a^{\mathscr{I}} \in \Delta^{\mathscr{I}}$. The extension of $\cdot^{\mathscr{I}}$ to some common compound concepts and roles is inductively defined as shown in the third column of Table 2.1, where $\#S$ denotes the cardinality of a set S. Most DLs provide two special concepts $\bot$ (the empty concept) and $\top$ (the concept under which everything falls) as shortcuts for $C \sqcap \neg C$ and $C \sqcup \neg C$, where C is some arbitrary concept.

A description logic axiom is a well-formed variable-free formula that uses some special logical operators. The most common type of axioms, their syntax

[1] Interpretations might be seen as potential "states of the world" or different "realities".

and semantics are listed in Table 2.2. As usual, concept equivalence axioms $C \equiv D$ are defined as an abbreviation for $C \sqsubseteq D$ and $D \sqsubseteq C$. In more expressive DLs one is also allowed to specify certain restrictions w.r.t. roles called role characteristics axioms. An exhaustive list of such kind of axioms, their syntax and semantics is provided in Table 2.3. The actual types of axioms available depend on and characterize the description logic in question but all DLs feature concept inclusion axioms between atomic (or possibly complex) concepts.

Table 2.2. Syntax and semantics of DL axioms.

	Syntax	**Semantics**	**Identifier**
TBox :			
Concept inclusion	$C \sqsubseteq D$	$C^{\mathscr{I}} \subseteq D^{\mathscr{I}}$	
Concept equivalence	$C \equiv D$	$C^{\mathscr{I}} = D^{\mathscr{I}}$	
ABox :			
Concept assertion	$C(a)$	$a^{\mathscr{I}} \in C^{\mathscr{I}}$	
Role assertion	$R(a, b)$	$(a^{\mathscr{I}}, b^{\mathscr{I}}) \in R^{\mathscr{I}}$	
RBox :			
Role inclusion	$R \sqsubseteq S$	$R^{\mathscr{I}} \subseteq S^{\mathscr{I}}$	$\mathscr{H}$
Role equivalence	$R \equiv S$	$R^{\mathscr{I}} = S^{\mathscr{I}}$	
Complex role inclusion	$R_1 \circ R_2 \sqsubseteq S$	$R_1^{\mathscr{I}} \circ R_2^{\mathscr{I}} \subseteq S^{\mathscr{I}}$	
Role disjointness	$Disj(R, S)$	$R^{\mathscr{I}} \cap S^{\mathscr{I}} = \emptyset$	

Table 2.3. Syntax and semantics of Role characteristic axioms.

	Syntax	**Semantics**	**Identifier**
Role characteristic :			
Functionality	$Func(R)$	$\forall d \in \Delta^{\mathscr{I}},\ \#\{e \in \Delta^{\mathscr{I}} \mid (d, e) \in R^{\mathscr{I}}\} \leq 1\}$	$\mathscr{F}$
Transitivity	$Trans(R)$	$(R^{+})^{\mathscr{I}} = R^{\mathscr{I}}$	
Reflexivity	$Refl(R)$	$\forall d \in \Delta^{\mathscr{I}}, (d, d) \in R^{\mathscr{I}}$	$\mathscr{R}$
Irreflexivity	$Irrefl(R)$	$\forall d \in \Delta^{\mathscr{I}}, (d, d) \notin R^{\mathscr{I}}$	
Symmetry	$Symm(R)$	$\{(b, a) \in R^{\mathscr{I}} \mid (a, b) \in R^{\mathscr{I}}\}$	
Asymmetry	$Asym(R)$	$\{(b, a) \notin R^{\mathscr{I}} \mid (a, b) \in R^{\mathscr{I}}\}$	

Historically one of the first DLs to be studied in depth was the so-called Attributive Language with complement ($\mathcal{ALC}$), which support the typical Boolean constructors, namely concept intersection ($\sqcap$), concept union ($\sqcup$), full negation ($\neg$), existential restriction ($\exists$) and universal restriction ($\forall$). The $\mathcal{ALC}$ concepts are defined by the following grammar, where R ranges over role names:

$$C, D ::= A \mid \top \mid \bot \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C\,,$$

The terminological knowledge of $\mathcal{ALC}$ can be expressed with General Concept Inclusions (GCI) of the form $C \sqsubseteq D$ and concept equivalence axioms $C \equiv D$. As for the representation of assertional knowledge $\mathcal{ALC}$ allows for concept $C(a)$ and role assertions $R(a, b)$.

The description logic languages follow a well-established naming convention. Based on $\mathcal{ALC}$ we obtain the names for a number of others more expressive DLs by concatenating the identifiers for the used constructors and axiom types (see the last column of Table 2.1 and 2.2). For example, a DL that extends $\mathcal{ALC}$ by a constructor that allows for inverse roles is called $\mathcal{ALCI}$. To refer to a DL where we additionally allow for role inclusion axioms of the form $\mathcal{R} \sqsubseteq \mathcal{S}$ we simply insert $\mathcal{H}$ in the name following the alphabetical order ($\mathcal{ALCHI}$). Note that some exceptions of the general naming rules exist, e.g. the DL that extend $\mathcal{ALC}$ by axioms for the definition of transitive roles is called $\mathcal{S}$. For more expressive DLs such as $\mathcal{SROIQ}$ [Horrocks et al., 2006], we additionally have the identifier $\mathcal{R}$ that stands for a range of role constructors, namely complex role inclusions, reflexivity, asymmetry, role disjointness and local reflexivity Self concept constructor and the universal role; the identifier $\mathcal{O}$ for nominal concepts and $\mathcal{Q}$ for arbitrary qualified number restrictions.

Unfortunately, already the DL $\mathcal{ALC}$ is non deterministic, i.e. due to the presence of the $\sqcup$-constructor it can be necessary to perform a case-by-case analysis for reasoning. As a consequence, all the more expressive DLs have a worst-case complexity for the reasoning time, which is at least exponential in the input size. However, in order to assure usability of reasoning systems in practice, often simpler, less expressive languages with specific computational properties are considered. In particular, tractability in such lightweight DLs is maintained imposing syntactical restrictions that disallow the representation of disjunctive

information. For example, the logic $\mathcal{EL}$ supports only $\top$, $\sqcap$, and $\exists$. Its extension $\mathcal{EL}^{\bot}$ supports also $\bot$. The logic $\mathcal{EL}$ ++ further adds *concrete domains* and some expressive role inclusions [Baader et al., 2005a]. The logic DL-lite$_R$ [Calvanese et al., 2005] supports inclusions shaped like $C \sqsubseteq D$ and $C \sqsubseteq \neg D$, where C and D range over concept names and *unqualified existential restrictions* such as $\exists R$ and $\exists R^-$ with $R \in \mathsf{N_R}$. $\mathcal{EL}$ ++ and DL-lite$_R$, respectively, constitute the foundation of the OWL2 profiles OWL2-EL and OWL2-QL. Both play an important role in applications; their worst-case complexity for the standard reasoning tasks is polynomial and therefore they can be used to model large amount of terminological and assertional knowledge (the same holds for some extensions of DL-lite$_R$, see [Artale et al., 2009]). In the following we introduce in more detail the basic DLs which are essential for describing the (new) optimization techniques and prove their correctness.

2.1.1 $\mathcal{EL}$ Family of Description Logics

Based on the original description of the DL $\mathcal{EL}^{++}$ in [Baader et al., 2005a, Baader et al., 2008], we now define the syntax and semantics of two members of the $\mathcal{EL}$ family of description logics that we consider throughout this thesis. Additionally, we define typically used restrictions for the combination of the different axioms, which are necessary to ensure tractability for the main inference problems.

Syntax and Semantics of the Description Logic $\mathcal{EL}^+_\bot$

The logic $\mathcal{EL}^+_\bot$ extends $\mathcal{EL}$ with the bottom concept and a restricted form of role-value maps. We start by providing some preliminary definitions needed to formally define the syntax and semantics of $\mathcal{EL}^+_\bot$.

Definition 2.1.1 (Restricted role-value-maps [Baader, 2003a]). A role-value-map is an expression of the form $R_1 \circ ... \circ R_m \sqsubseteq S_1 \circ ... \circ S_n$ where $m, n \geq 1$ and $R_1, ..., R_m, S_1, ..., S_n$ are role names. We say that this role-value-map is

restricted if $m = 2$ and $n = 1$[2].

Definition 2.1.2 (Role inclusions). Let $R_1 \circ ... \circ R_m \sqsubseteq S$ where $m \geq 1$ be a restricted role-value-map. We call this restricted role-value-map:

- a *role inclusion (RI)* if $m = 1$;
- a *complex role inclusion* for $m = 2$.

Definition 2.1.3 (Syntax of Individuals, Concepts and Roles in $\mathscr{EL}^+_\bot$). Let $\mathsf{N_C}$, $\mathsf{N_R}$ and $\mathsf{N_I}$ be countable infinite and pairwise disjoint sets of concept names, role names and individual names. We call $\Sigma = (\mathsf{N_C}, \mathsf{N_R}, \mathsf{N_I})$ a signature. The set of $\mathscr{EL}^+_\bot - concepts$ is the smallest set build inductively over symbols of Σ following the grammar:

$$C, D ::= A \mid \top \mid \bot \mid C \sqcap D \mid \exists R.C\,,$$

where $A \in \mathsf{N_C}$ and $R \in \mathsf{N_R}$.

Note, that together with the concept names, $\top$ and $\bot$ are also referred to as atomic concepts.

Definition 2.1.4 ($\mathscr{EL}^+_\bot$ Knowledge base) A knowledge base $\mathscr{KB}$ is is a tuple of the form $(\mathscr{T}, \mathscr{R}, \mathscr{A})$, where $\mathscr{T}$ is a *TBox*, $\mathscr{A}$ an *ABox* and $\mathscr{R}$ an *RBox*. A *(general) TBox* is a finite set of *general* concept inclusions (GCIs) $C \sqsubseteq D$ and emphgeneral concept equivalences (GCEs) $C \equiv D$. An *ABox* is a finite set of *concept assertions* $C(a)$ and *role assertions* $R(a, b)$. An *RBox* is a finite set of role inclusion $\mathscr{R} \sqsubseteq \mathscr{S}$ and complex role inclusion axioms $R_1 \circ R_2 \sqsubseteq S$.

Definition 2.1.5 (Definitorial TBox[3]) An axiom is called a *definition* of A if it is of the form $A \sqsubseteq D$ or $A \equiv D$, where A is an concept name. It is unique if $\mathscr{T}$ contains no other definition of A, and it is acyclic if D does not refer either directly or indirectly to A. A TBox $\mathscr{T}$ is called definitorial if it contains only unique,

[2]Note that the restriction $m = 2$ is not really necessary. All complexity results would still hold if on the left-hand sides are allowed compositions of $m \geq 1$ roles for an arbitrary m. However, the restriction n = 1 is crucial (cf. [Baader, 2003a]).

[3]Also called acyclic or unfoldable in the literature.

acyclic definitions. Given a definitorial TBox $\mathscr{T}$, concept names occurring on the left-hand side of a definition are called defined concepts, whereas the others are called primitive.

From a computational point of view, definitorial TBox are interesting since they usually allow for the use of simplified reasoning techniques. Consequently, reasoning w.r.t. such TBoxes is often of a lower complexity than reasoning w.r.t. general TBoxes.

Several remarks regarding the expressivity of $\mathscr{EL}^{++}$ are in order. First, the presence of the $\bot$-concept make it possible to express *concept disjointness*, which for two concepts C and D is equivalent to stating that $C \sqcap D \sqsubseteq \bot$. RIs on the other hand generalize several means of expressivity important in ontology applications:

- role hierarchies $R \sqsubseteq S$ can be expressed as $R \sqsubseteq S$;
- role equivalences $R \equiv S$ can be expressed as $R \sqsubseteq S$ and $S \sqsubseteq R$;
- transitive roles can be expressed as $R \circ R \sqsubseteq R$;
- left-identity rules can be expressed as $R \circ S \sqsubseteq S$;
- right-identity rules can be expressed as $R \circ S \sqsubseteq R$.

An interpretation $\mathscr{I}$ for $\mathscr{EL}^{+}_{\bot}$ is defined in the usual way. $\mathscr{I}$ is called a *model* of a concept C if $C^{\mathscr{I}} \neq \emptyset$. If $\mathscr{I}$ is a model of C, we also say that C is *satisfied* by $\mathscr{I}$. Additionally, if $\delta \in C^{\mathscr{I}}$ we also say that δ is in the extension of C.

Definition 2.1.6 (Semantics of $\mathscr{EL}^{+}_{\bot}$ Axioms and Knowledge bases) Let $\mathscr{I} = (\Delta^{\mathscr{I}}, \cdot^{\mathscr{I}})$ be an interpretation, then $\mathscr{I}$ *satisfies* (i) a GCI $C \sqsubseteq D$ if $C^{\mathscr{I}} \subseteq D^{\mathscr{I}}$, (ii) an assertion $C(a)$ if $a^{\mathscr{I}} \in C^{\mathscr{I}}$, (iii) an assertion $R(a,b)$ if $(a^{\mathscr{I}}, b^{\mathscr{I}}) \in R^{\mathscr{I}}$, (iv) a role inclusion $\mathscr{R} \sqsubseteq \mathscr{S}$ if $R^{\mathscr{I}} \subseteq S^{\mathscr{I}}$, and a complex role inclusion $R_1 \circ R_2 \sqsubseteq S$ if ${R_1}^{\mathscr{I}} \circ {R_2}^{\mathscr{I}} \subseteq S^{\mathscr{I}}$. Then, $\mathscr{I}$ is a (classical) *model* of a TBox $\mathscr{T}$ (resp. an ABox $\mathscr{A}$, RBox $\mathscr{R}$) if $\mathscr{I}$ *satisfies* all the axioms of $\mathscr{T}$ (resp. $\mathscr{A}$, $\mathscr{T}$). We say that $\mathscr{I}$ *satisfies* $\mathscr{K}\mathscr{B}$ if it *satisfies* both $\mathscr{T}$, $\mathscr{A}$ and $\mathscr{R}$. In this case we also say that $\mathscr{I}$ is a *model* of $\mathscr{K}\mathscr{B}$ and write $\mathscr{I} \models \mathscr{K}\mathscr{B}$.

A knowledge base is *consistent* if it has at least one model, otherwise it is inconsistent. We say that an axiom α is a consequence of a knowledge base $\mathscr{K}\mathscr{B}$, or also that $\mathscr{K}\mathscr{B}$ *entails* α and write $\mathscr{K}\mathscr{B} \models \alpha$, if every model of $\mathscr{K}\mathscr{B}$ is a model of α. In an extreme case in which a knowledge base is inconsistent, every axiom is entailed, i.e. holds vacuously in all of the (zero) interpretations that satisfy the knowledge base. Such a knowledge base is clearly of no utility, so avoiding inconsistency is a key task during modeling.

Syntax and Semantics of the Description Logic $\mathscr{EL}^{++}$

The description logic $\mathscr{EL}^{++}$ extends $\mathscr{EL}^{+}_{\bot}$ with nominals, a restricted form of concrete domains, range restrictions and reflexive roles, i.e., role inclusions of the form $\epsilon \sqsubseteq R$ [Baader et al., 2005a, Baader et al., 2008].

A nominal concept is a singleton, i.e. a concept with a single instance. In particular, nominals make it possible to capture assertional knowledge with TBox axioms: C(a) can be rewritten $\{a\} \sqsubseteq C$ and $R(a, b)$ is equivalent to $\{a\} \sqsubseteq \exists R.\{b\}$. The identity of two individuals can as well be expressed as $\{a\} \sqsubseteq \{b\}$, and their distinctness as $\{a\} \sqcap \{b\} \sqsubseteq \bot$. If necessary, the *unique name assumption* for individual names can be enforced by explicitly stating the distinctness for all relevant individual names a and b.

The concrete domain constructor provides an interface to the so-called *concrete domains*, which extend the description logic by specific predicates with built-in interpretations, such as strings, decimals, integers. In practice, it is often useful to directly refer to concrete data values from fixed domains. For instance, a lot of application scenarios require representing personal information such as names, ages, citizenships, etc. Formally, a concrete domain $\mathscr{D}$ is a pair $(\Delta^{\mathscr{D}}, \mathscr{P}^{\mathscr{D}})$ where $\Delta^{\mathscr{D}}$ is a set of values and $\mathscr{P}^{\mathscr{D}}$ is a set of *predicate names*. Every $p \in \mathscr{P}$ is associated with an arity n and extension $p^D \subseteq (\Delta^D)^n$. The link between the logic and concrete domain is established through a set of *feature names* $\mathsf{N_F}$. For each feature name $f \in \mathsf{N_F}$ and interpretation function $\mathscr{I}$, $f^{\mathscr{I}}$ is a partial function from $\Delta^{\mathscr{I}}$ to $\Delta^{\mathscr{D}}$.

Example 2.1.7 (Rational numbers $\mathbb{Q}$ [Baader et al., 2005a]) The set of

rational numbers $\mathbb{Q}$ can be formalized with a concrete domain $\mathtt{Q} = (\mathbb{Q}, \mathscr{P}^{\mathtt{Q}})$ that has as its domain the set of rational numbers $\mathbb{Q}$ and its set of predicates $\mathscr{P}^{\mathtt{Q}}$ consists of the following predicates:

- a unary predicate $\mathtt{T}_{\mathtt{Q}}$ with $(\mathtt{T}_{\mathtt{Q}})^{\mathtt{Q}} = \mathbb{Q}$;
- a unary predicates $=_{\mathtt{q}}$ and $>_{\mathtt{q}}$ for each $\mathtt{q} \in \mathbb{Q}$;
- a binary predicate $=$;
- a binary predicate $+_{\mathtt{q}}$ for each $\mathtt{q} \in \mathbb{Q}$, with $(+_{\mathtt{q}})^{\mathtt{Q}} = \{(\mathtt{q}', \mathtt{q}'') \in \mathbb{Q}^2 \mid q' + q = q''\}$.

∎

The range restrictions are a very important special case of universal value restrictions, a constructor heavily used in some large biomedical ontologies (eg. the thesaurus of the US national cancer institute (NCI)).

Definition 2.1.8 (Syntax of Individuals, Concepts and Roles in $\mathscr{EL}^{++}$). Let $\mathsf{N_C}$, $\mathsf{N_R}$, $\mathsf{N_I}$, $\mathscr{P}$ and $\mathsf{N_F}$ be countable infinite and pairwise disjoint sets of concept names, role names, individual names, predicate names and feature names. We call $\Sigma = (\mathsf{N_C}, \mathsf{N_R}, \mathsf{N_I}, \mathsf{N_F})$ a signature. The set of $\mathscr{EL}^{++} - \textit{concepts}$ is the smallest set build inductively over symbols of Σ following the grammar:

$$C, D ::= A \mid \top \mid \bot \mid \{a\} \mid C \sqcap D \mid \exists R.C \mid p(f_1, ..., f_k)$$

where $A \in \mathsf{N_C}$, $R \in \mathsf{N_R}$, $p \in \mathscr{P}$ and $f_1, ..., f_k \in \mathsf{N_F}$.

Definition 2.1.9 (Semantics of Individuals, Concepts and Roles in $\mathscr{EL}^{++}$). An *interpretation* $\mathscr{I} = (\Delta^{\mathscr{I}}, \cdot^{\mathscr{I}})$ consists of a non empty set of individuals $\Delta^{\mathscr{I}}$, the *domain* of $\mathscr{I}$ and an *interpretation function* $\cdot^{\mathscr{I}}$ which maps each concept name $A \in \mathsf{N_C}$ to a subset $A^{\mathscr{I}}$ of $\Delta^{\mathscr{I}}$, each role name $R \in \mathsf{N_R}$ to a binary relation $R^{\mathscr{I}}$ on $\Delta^{\mathscr{I}}$, each individual name $a \in \mathsf{N_I}$ to an individual $a^{\mathscr{I}} \in \Delta^{\mathscr{I}}$, and each feature name $f \in \mathsf{N_F}$ to a partial function $f^{\mathscr{I}}$ from $\Delta^{\mathscr{I}}$ to $\Delta^{\mathscr{D}}$ The extension of $\cdot^{\mathscr{I}}$ to arbitrary compound concepts and roles is inductively defined as follows:

$$
\begin{aligned}
\top^{\mathscr{I}} &= \Delta^{\mathscr{I}} \\
\bot^{\mathscr{I}} &= \emptyset \\
\{a\}^{\mathscr{I}} &= \{a^{\mathscr{I}}\} \\
(C \sqcap D)^{\mathscr{I}} &= C^{\mathscr{I}} \cap D^{\mathscr{I}} \\
(\exists R.C)^{\mathscr{I}} &= \{d \in \Delta^{\mathscr{I}} \mid \exists e \in \Delta^{\mathscr{I}}.[(d,e) \in R^{\mathscr{I}} \wedge e \in C^{\mathscr{I}}]\} \\
p(f_1,...,f_k)^{\mathscr{I}} &= \{x \in \Delta^{\mathscr{I}} \mid \exists y_1,...,y_k \in \Delta^{D_j} : f_i^{\mathscr{I}}(x) = y_i \ for\ 1 \leq i \leq k \\
&\qquad \wedge\ (y_1,...,y_k) \in p^D\}.
\end{aligned}
$$

Definition 2.1.10 ($\mathscr{EL}^{++}$ Knowledge base) A knowledge base $\mathscr{KB}$ is is a tuple of the form $(\mathscr{T}, \mathscr{R}, \mathscr{A})$, where $\mathscr{T}$ is a *TBox*, $\mathscr{A}$ an *ABox* and $\mathscr{R}$ an *RBox*. A *(general) TBox* is a finite set of *general concept inclusions (GCIs)* $C \sqsubseteq D$ and emphgeneral concept equivalences (GCEs) $C \equiv D$. An *ABox* is a finite set of *concept assertions* $C(a)$ and *role assertions* $R(a,b)$. An *RBox* is a finite set of (complex) role inclusions (RIs) $R_1 \circ R_2 \sqsubseteq S$, domain restrictions (DRs) $dom(R) \sqsubseteq C$, and range restrictions (RRs) $ran(R) \sqsubseteq C$. A special case of RIs also allowed is $\epsilon \sqsubseteq R$.

Definition 2.1.11 (Semantics of $\mathscr{EL}^{++}$ Axioms and Knowledge bases) Let $\mathscr{I} = (\Delta^{\mathscr{I}}, \cdot^{\mathscr{I}})$ be an interpretation, then $\mathscr{I}$ *satisfies* an axiom α, written $\mathscr{I} \models \alpha$, as defined in the following:

$$
\begin{aligned}
&\mathscr{I} \models C \sqsubseteq D && \text{if} \quad C^{\mathscr{I}} \subseteq D^{\mathscr{I}} \\
&\mathscr{I} \models C(a) && \text{if} \quad a^{\mathscr{I}} \in C^{\mathscr{I}} \\
&\mathscr{I} \models R(a,b) && \text{if} \quad (a^{\mathscr{I}}, b^{\mathscr{I}}) \in R^{\mathscr{I}} \\
&\mathscr{I} \models R_1 \circ R_2 \sqsubseteq S && \text{if} \quad {R_1}^{\mathscr{I}} \circ {R_2}^{\mathscr{I}} \subseteq S^{\mathscr{I}} \\
&\mathscr{I} \models dom(R) \sqsubseteq C && \text{if} \quad R^{\mathscr{I}} \subseteq C^{\mathscr{I}} \times \Delta^{\mathscr{I}} \\
&\mathscr{I} \models ran(R) \sqsubseteq C && \text{if} \quad R^{\mathscr{I}} \subseteq \Delta^{\mathscr{I}} \times C^{\mathscr{I}}
\end{aligned}
$$

Then, $\mathscr{I}$ is a (classical) *model* of a TBox $\mathscr{T}$ (resp. an ABox $\mathscr{A}$, RBox $\mathscr{R}$) if $\mathscr{I}$ *satisfies* all the axioms of $\mathscr{T}$ (resp. $\mathscr{A}$, $\mathscr{T}$). We say that $\mathscr{I}$ is a *model* of $\mathscr{KB}$ if it is a *model* of both $\mathscr{T}$, $\mathscr{A}$ and $\mathscr{R}$.

Please, note that the DL $\mathscr{EL}^{++}$ may be equipped with more than one concrete domains simultaneously, say $\mathscr{D}_1, ..., \mathscr{D}_n$. In such case we generally assume that

$\Delta^{\mathscr{D}_i}, ..., \Delta^{\mathscr{D}_j} = \emptyset$ for $1 \leq i < j \leq n$. Furthermore, if we want to stress the use of particular concrete domains $\mathscr{D}_1, ..., \mathscr{D}_n$ we write $\mathscr{EL}^{++}(\mathscr{D}_1, ..., \mathscr{D}_n)$ instead of $\mathscr{EL}^{++}$. Unfortunately, unrestricted use of concrete domains may have dramatic effects on the decidability and computational complexity of the underlying DL. In order to assure tractability and completeness of the standard reasoning task in $\mathscr{EL}^{++}$, we need to restrict the concrete domains to be p-admissible.

Definition 2.1.12 (P-admissible concrete domains) A concrete domain $\mathscr{D} = (\Delta^{\mathscr{D}}, \mathscr{P}^{\mathscr{D}})$ is p-admissible iff:

1. satisfiability and implication in $\mathscr{D}$ are decidable in polynomial time;
2. D is convex: if a conjunction of atoms of the form $p(f1, ..., fk)$ for $p \in \mathscr{P}^{\mathscr{D}}$ [4] implies a disjunction of such atoms, then it also implies one of its disjuncts.

Consider the concrete domain $\mathtt{Q}$ defined in Example 2.1.7. $\mathtt{Q}$ is p-admissible. In fact polynomiality of reasoning in $\mathtt{Q}$ can be shown by reduction to linear programming and convexity has been proved in [Baader et al., 2005b]. Therefore $\mathscr{EL}^{++}$ allows $\mathtt{Q}$.

To avoid intractability, we also need to impose a restriction on the structure of RBoxes. If we allow for arbitrary combinations of role inclusions and range restrictions we may easily run into decidability issues. For an RBox $\mathscr{R}$ and role names R and S we write $\mathscr{R} \models R \sqsubseteq S$ iff $R = S$ or $\mathscr{R}$ contains role inclusions $R_1 \sqsubseteq R_2, ..., R_{n_1} \sqsubseteq R_n$ with $R = R_1$ and $S = R_n$. Furthermore, we write $\mathscr{R} \models ran(R) \sqsubseteq C$ if a role name S exists such that $\mathscr{R} \models R \sqsubseteq S$ and $ran(S) \sqsubseteq C \in \mathscr{R}$. Now, to restrict intricate interplay between role inclusions and range restrictions we require that if a RI $R_1 \dot{:} ... \dot{:} R_n \sqsubseteq S$, $n \geq 1$, implies a role relationship $(a, b) \in S^{\mathscr{I}}$, then the RR on S do not impose new concept memberships of b. Formally, we have:

$$\text{If } R_1 \dot{:} ... \dot{:} R_n \sqsubseteq S \in \mathscr{R} \text{ with } n \geq 1 \text{ and } \mathscr{R} \models ran(S) \sqsubseteq C, \text{ then } \mathscr{R} \models ran(R_n) \sqsubseteq C$$

[4] If $p \in \mathscr{P}^{\mathscr{D}}$, then the $\mathscr{EL}^{++}$-concept $p(f1, ..., fk)$ can be viewed as an atomic first-order formula with variables $f1, ..., fk$. Thus, it makes sense to consider Boolean combinations of such atomic formulae, and to talk about whether such a formula is satisfiable in the first-order interpretation $\mathscr{D}$, or whether one such formula implies another one in $\mathscr{D}$.

In the reminder of the thesis, we assume that $\mathscr{EL}^{++}$ knowledge bases comply with the presented restrictions.

2.2 Reasoning Tasks and Their Reducibility

A distinguished feature of the logic-based knowledge representation formalisms is the emphasis on reasoning as a central service: reasoning allows to derive implicitly represented knowledge from the knowledge that is explicitly represented in the knowledge base. In order to be able to query this implicit knowledge one can use reasoners that provide a range of inference services for the computation of specific reasoning tasks. In this section we will review typical tasks that can be performed with DL knowledge bases and that require sophisticated reasoning. We can see that some of those tasks can be reduced to others which alleviates the task of creating tools performing those tasks.

- **Consistency Checking:** Given a knowledge base $\mathscr{KB}$, is $\mathscr{KB}$ consistent? A knowledge base $\mathscr{KB}$ is consistent (also called satisfiable) if there exists (at least one) interpretation $\mathscr{I} = (\Delta^{\mathscr{I}}, \cdot^{\mathscr{I}})$ that is a model of $\mathscr{KB}$, i.e., $\mathscr{I} \models \mathscr{KB}$.

- **Satisfiability Checking:** Given a knowledge base $\mathscr{KB}$ and a concept C, is C satisfiable w.r.t. $\mathscr{KB}$ ($\mathscr{KB} \not\models C \sqsubseteq \bot$)? A concept C is called satisfiable if it may contain individuals, i.e. there is a model $\mathscr{I} = (\Delta^{\mathscr{I}}, \cdot^{\mathscr{I}})$ of $\mathscr{KB}$ for which the extension of C is nonempty, formally: $C^{\mathscr{I}} \neq \emptyset$ for $\mathscr{I} \models \mathscr{KB}$.

- **Subsumption Checking:** Given a knowledge base $\mathscr{KB}$ and two concepts C and $\mathscr{D}$, is C subsumed by D, written $C \sqsubseteq D$, w.r.t. $\mathscr{KB}$ ($\mathscr{KB} \models C \sqsubseteq D$), i.e., for every interpretation $\mathscr{I} = (\Delta^{\mathscr{I}}, \cdot^{\mathscr{I}})$ of $\mathscr{KB}$ such that $\mathscr{I} \models \mathscr{KB}$ it is also the case that $C^{\mathscr{I}} \subseteq D^{\mathscr{I}}$. If the subsumption holds, we also say that D subsumes C. Furthermore, for a subsumption $C \sqsubseteq D$, we refer to C as the subsumee and to D as the subsumer.

- **Instance Checking (or Retrieval):** Given a knowledge base $\mathscr{KB}$, a concept C and an individual a, is a an instance of C w.r.t. $\mathscr{KB}$ ($KB \models$

$C(a)$), i.e., for every interpretation $\mathscr{I} = (\Delta^{\mathscr{I}}, \cdot^{\mathscr{I}})$ of $\mathscr{K}\mathscr{B}$ such that $\mathscr{I} \models \mathscr{K}\mathscr{B}$ it is also the case that $a^{\mathscr{I}} \in C^{\mathscr{I}}$.

Please note that the reasoning tasks so far are described for concepts, but they can obviously be defined also for roles. For example, sometimes, the term instance retrieval is also used for roles. In that case we are interested whether a pair of individual names (a, b) is an instance of a certain role R, i.e., if $(a^{\mathscr{I}}, b^{\mathscr{I}}) \in R^{\mathscr{I}}$ for every model $\mathscr{I}$ of $\mathscr{K}\mathscr{B}$. Historically, however, primarily the reasoning tasks for concepts have been considered and we also refer to the concept-based one in the following if we mention the reasoning tasks without any further specification.

For DLs that allow for nominals and the $\bot$ concept, these reasoning tasks can be reduced to each other. Consistency checking can be regarded as the main reasoning service. In practice, for logics where it is possible to have inconsistent knowledge bases, consistency checking is performed before any other reasoning. This is due to the principle of explosion, for which an inconsistent knowledge base entails every statement, which renders any derived information useless. As an example, lets consider the DL $\mathscr{EL}^{++}$, the other three standard reasoning tasks are reducible to consistency checking as follows: A concept C is unsatisfiable with respect to $\mathscr{K}\mathscr{B}$ if and only if $\mathscr{K}\mathscr{B} \cup \{C(a)\}$ is inconsistent for some fresh individual name a; A concept C is subsumed by D with respect to $\mathscr{K}\mathscr{B}$ if and only if the $\mathscr{K}\mathscr{B} \cup \{C(a)\} \cup \{D \sqcap \{a\} \sqsubseteq \bot\}$ is inconsistent for some fresh individual name a; An individual a is an instance of a concept C with respect to $\mathscr{K}\mathscr{B}$ if and only if the $\mathscr{K}\mathscr{B} \cup \{C \sqcap \{a\} \sqsubseteq \bot\}$ is inconsistent.

A prototypical reasoning task that generalizes all these reasoning tasks is often considered in practice: given a knowledge base, it can be queried by checking whether some axiom is necessarily true. More precisely, testing if $\top \sqsubseteq \bot$ holds is equivalent to inconsistency checking, $C \sqsubseteq \bot$ to unsatisfiability checking, $C \sqsubseteq D$ to subsumption checking and $C(a)$ to instance checking. It can be formalized as follows:

- **Entailment Checking:** Given a knowledge base $\mathscr{K}\mathscr{B}$ and an axiom α, $\mathscr{K}\mathscr{B}$ entails α ($KB \models \alpha$), if for every interpretation $\mathscr{I} = (\Delta^{\mathscr{I}}, \cdot^{\mathscr{I}})$ of $\mathscr{K}\mathscr{B}$ such that $\mathscr{I} \models \mathscr{K}\mathscr{B}$ it is also the case that $\mathscr{I} \models \alpha$.

Entailment checking can also be reduced to consistency checking by extending the knowledge base with a counter example of the axiom that has to be queried. If the extended knowledge base is inconsistent then the axiom is obviosly entailed.

Furthermore, DLs support inference patterns which occur in many intelligent information processing applications, and that are also used by humans to structure and understand the world: classification of concepts and individuals. Classification of concepts (called classification) determines the subsumption relationships between the concepts occurring in a given knowledge base, thus allowing to organize the concepts in a subsumption hierarchy. Classification of individuals (called realization), on the other hand, determines whether a given individual is an instance of a certain concept. It thus provides useful information on the properties of an individual. Classification and realization are considered higher level reasoning tasks as they require a range of computations:

- **Classification:** Given a knowledge base $\mathscr{K}\mathscr{B}$, compute all subsumptions between atomic concepts in $\mathscr{K}\mathscr{B}$, i.e., all the axioms $A \sqsubseteq B$ for which the concepts A and B occur in $\mathscr{K}\mathscr{B}$ and $\mathscr{K}\mathscr{B} \models A \sqsubseteq B$. The subsumption hierarchy can be obtained by a transitive reduction, i.e., by removing all axioms that represent *indirect* subsumption relations, i.e. axioms of the form $A_1 \sqsubseteq A_2$ for which a concept B exist such that $\mathscr{K}\mathscr{B} \models A_1 \sqsubseteq B$ and $\mathscr{K}\mathscr{B} \models B \sqsubseteq A_2$.

- **Realization:** Given a knowledge base $\mathscr{K}\mathscr{B}$ as input, compute all instances of (atomic) concepts i.e., all axioms $A(a)$, where the individual a and the concept A occurs in $\mathscr{K}\mathscr{B}$ and for which $\mathscr{K}\mathscr{B} \models A(a)$.

Obviously, the higher level reasoning tasks can be performed by checking the entailment $\mathscr{K}\mathscr{B} \models A \sqsubseteq B$ (resp. $\mathscr{K}\mathscr{B} \models A(a)$) for any pair A, B of concept names (resp. any concept name A and an individual a). However, such naive reduction amounts to quadratically many entailment checks and, therefore, can be impractical. Rather, one has to devise optimal deduction procedures that exploit the properties of the subsumption relation and prove their correctness with respect to the above specification.

The reasoning tasks described above are often referred to as standard reasoning tasks as they all are concerned with determining logical consequences. Beyond

those deductive tasks, there are also non-standard reasoning tasks such as induction [Lehmann, 2009], unification [Baader and Narendran, 2001], conjunctive query answering [Lutz, 2008, Lutz et al., 2009], explanation [Horridge et al., 2008], where the goal is somewhat different. In the following, we briefly describe conjunctive query answering as the other tasks are not in the focus of this thesis.

Conjunctive Query Answering

Querying KBs plays a central role in data-intensive applications. In these settings, instance retrieval is seen as a rather weak form of querying in some aspects. Although possibly complex concepts can be used as queries one can only query for tree-like relational structure, i.e., a DL concept cannot express arbitrary (cyclic) structures. Conjunctive queries (CQs) are well known in the database community [Chandra and Merlin, 1977] and constitute an expressive query language with capabilities that go far beyond the standard instance retrieval. In terms of first-order logic, the CQs are formulae from the positive existential fragment. Existentially quantified variables in a query are also called non-distinguished variables, whereas the free (or answer) variables are called distinguished. For an example, consider a knowledge base that contains the assertion $\exists hasDaughter.\exists hasDaughter(anne)$, which informally states that the individual $anne$ has a daughter who has a daughter and hence, that $anne$ is a grandmother. For this knowledge base, $anne$ is clearly an answer to the conjunctive query $hasDaughter(x, y) \wedge hasDaughter(y, z)$, with a single distinguished variable x. If all variables in the query are non-distinguished, the query answer is just true or false and the query is called a Boolean query. Given a knowledge base $\mathscr{K}\mathscr{B}$ and a Boolean CQ q, the query entailment problem is deciding whether q is true or false w.r.t. $\mathscr{K}\mathscr{B}$, i.e., whether each model of $\mathscr{K}\mathscr{B}$ provides for a suitable assignment for the variables in q. If a CQ contains distinguished variables, the answers to the query are all those tuples of individual names for which $\mathscr{K}\mathscr{B}$ entails the query that is obtained by replacing the free variables with the individual names in the answer tuple. The problem of finding all answer tuples is known as conjunctive query answering. Note that in general, solving this task is way harder than querying a classical database, as the considered DL models may be infinite in both size and number. Furthermore, conjunctive query answer-

ing is not polynomially reducible to any of the other standard reasoning tasks treated above (the worst-case complexity for the problem is usually way harder, cf. [Lutz, 2008]).

Similar to the query languages for classical databases (e.g., SQL), query languages for knowledge bases that allow to obtain all the consequences provided by the presented reasoning tasks also exist. Nowadays SPARQL, constitute a de facto standard when it comes to conjunctive query answering. SPARQL originates as a query language for RDF, where the evaluation of queries is based on simple (sub)graphs pattern matching. The recently standardized SPARQL 1.1 extension provides the so-called entailment regimes for which also implied consequences of OWL 2 ontologies w.r.t. the Direct Semantics are taken into consideration. Unfortunately, the new standard is still not fully supported by the consolidated query engines. In particular, it is typically only implemented in specific reasoning systems or as black-box extension of reasoning systems, where this more sophisticated reasoning task is reduced to the standard ones. For example, the OWL-BGP is a framework for reducing SPARQL queries[5] to standard reasoning tasks for OWL API compatible reasoners.

2.3 Relationship to the Web Ontology Language

The Web Ontology Language (OWL) is a knowledge representation language developed by the World Wide Web Consortium (W3C) working group. The underpinning logic is based on the very expressive DL $\mathcal{SROIQ}$. Defining semantics via translation into DL allows OWL to exploit well-established results from DL research regarding decidability and complexity of key inference problems and to reuse the available highly optimized DL reasoners inside OWL applications - thus accomplishing the main design goal of the language [Heflin, 2004]. The latest version of the OWL specification as standardized in 2009 is called OWL 2 [Rudolph et al., 2012].

The main building blocks of OWL are very similar to those of DLs. An OWL ontology indeed consists of the same basic elements of a DL knowledge

[5] More precisely, the basic graph patterns of SPARQL queries (BGPs) are parsed into OWL entities thus enabling their assessment under the OWL Direct Semantics Entailment Regime.

base with some minor renaming. In particular, concepts are called classes and roles are called properties. As in standard DL, OWL classes may be names or complex expressions built up from simpler classes and properties with the help of a set of constructors. The set of constructors supported by OWL together with the corresponding DL syntax is summarized in Table 2.4. However, compared with the defined syntax of $\mathscr{SROIQ}$, OWL provides additional constructors as "syntactic sugar", i.e., they can be conceived as shortcuts for which we would require several DL constructs to represent the same restriction.

Table 2.4. OWL constructors.

Constructor	**DL Syntax**	**DL Example**
ObjectComplementOf	$\neg C$	$\neg Male$
ObjectIntersectionOf	$C_1 \sqcap ... \sqcap C_n$	$Human \sqcap Female$
ObjectUnionOf	$C_1 \sqcup ... \sqcup C_n$	$Male \sqcup Female$
ObjectSomeValuesFrom	$\exists R.C$	$\exists hasChild.Student$
ObjectAllValuesFrom	$\forall R.C$	$\forall hasChild.Female$
ObjectOneOf	$\{a_1 ... a_n\}$	$\{anne, elijah\}$
ObjectHasValue	$\exists R.\{a\}$	$\exists hasDestination.\{Italy\}$
ObjectHasSelf	$\exists R.Self$	$\exists loves.Self$
ObjectMinCardinality	$\leq nR.C$	$\leq 1\ hasChild.Male$
ObjectMaxCardinality	$\geq nR.C$	$\geq 3\ hasChild.Female$
ObjectInverseOf	R^-	$hasParent^-$

Besides classes defined by the ontology, OWL further specifies a range of datatypes (mostly taken from the RDF specification [Carroll and Klyne, 2004] and the set of XML Schema Datatypes [Sperberg-McQueen et al., 2012]) that can be used in $someValuesFrom$, $allValuesFrom$, and $hasValue$ restrictions. Datatypes are unary predicates with a built-in interpretation that can be seen as a simplified version of the so-called concrete domains (see Section 2.1.1 for definitions). Mainly used in practice are basic datatypes such as *xsd:string, xsd:integer,*

xsd:double, xsd:boolean interpreted as the set of all strings, integer, decimal and Boolean values. Some important remarks regarding datatype modeling are in order. First, OWL strictly distinguishes object properties, i.e. properties that relate pairs of individuals, from data properties, i.e. properties that relate individuals to values from some datatype. Moreover, to further improve the ease of use and clarity, OWL constructs that relate to individuals have names prefixed by "*Object*" in contrast to those that relate to datatypes prefixed by "*Data*". Finally, although the constructors listed in Table 2.4 are described for classes, they can obviously be formulated also for datatypes (except from the local reflexivity one).

Table 2.5. OWL axioms.

Axiom	**DL Syntax**	**Example**
SubClassOf	$C_1 \sqsubseteq C_2$	$Mammal \sqsubseteq Animal$
EquivalentClass	$C_1 \equiv C_2$	$Woman \equiv Human \sqcap Female$
DisjointClasses	$C_1 \sqcap C_2 \sqsubseteq \bot$	$Man \sqcap Woman \sqsubseteq \bot$
ClassAssertion	$C(a)$	$Woman(anne)$
ObjectPropertyAssertion	$R(a, b)$	$hasChild(anne, christine)$
SameIndividual	$\{a\} \equiv \{b\}$	$\{turing\} \equiv \{alan\_turing\}$
SubObjectPropertyOf	$R_1 \sqsubseteq R_2$	$hasParent \sqsubseteq hasAncestor$
EquivalentObjectProperties	$R_1 \equiv R_2$	$hasBrother \equiv hasMaleSibling$
DisjointObjectProperties	$Disj(R_1, R_2)$	$Disj(hasMother, hasFather)$

As already mentioned, OWL ontologies consists of a set of axioms. Table 2.5 provides a list of axioms supported by OWL. Moreover, OWL also allows characteristics of object properties to be asserted as well as functionality of data properties. Please note that the OWL specification features much more axiom types than the ones defined in $\mathcal{SROIQ}$ knowledge bases. However, as far as the purely logical axioms are concerned, again all these axioms can be considered as syntactic sugar. For example, in OWL we can express that a set of classes is pair-

wise equivalent with a single axiom of the form $EquivalentClasses(C_1, ..., C_n)$ while we would need a set of concept equivalence axioms of the form $C_i \equiv C_j$ for $1 \leq i < j \leq n$ to represent it in $\mathcal{SROIQ}$.

Aside from the logical features, OWL ontologies further consider a number of other aspects relevant in practice that are not covered in DL knowledge bases at all. For example, OWL provide means of naming an ontology and an importing mechanism that make it possible to refer to other relevant ontologies. An OWL reasoner should therefore be able to load all imported ontologies and consider the contained axioms for reasoning. Other important extra-logical features include non-logical axioms to declare identifiers, and the possibility to add a vast range annotations to OWL axioms similar to comments in a programming language. If not stated otherwise, in the reminder of this thesis we will use the term ontology to simply refer to a document created in OWL, modeling knowledge of an application domain that is relevant for reasoning. Thereby, we will consider it to be equivalent with the arguably more appropriate term knowledge base.

The current OWL 2 standard support several serialization formats for OWL ontologies, e.g., the Manchester Syntax [Patel-Schneider and Horridge, 2012] or the OWL/XML serialization format [Parsia et al., 2012]. The notation used above is known as the Functional Style Syntax (FSS), since expressive features are written like function symbols in prefix notation [Patel-Schneider et al., 2012]. Moreover, OWL ontologies are often serialized in the RDF/XML format [Motik and Patel-Schneider, 2012], which maps the axioms of an OWL ontology to triples in graphs of the Resource Description Framework (RDF) [Raimond and Schreiber, 2014]. Among all those options, FSS represents the data model of OWL more closely, whereas RDF/XML is primary exchange syntax since it is the only mandatory to be supported by all OWL 2 applications. In order to provide compatibility with RDF, the OWL ontologies can be interpreted with the so-called OWL 2 RDF-Based Semantics [Schneider, 2012], which is slightly different in some aspects from the OWL 2 Direct Semantics [Grau et al., 2012], i.e., the model-theoretic semantics of Description Logics, even though both lead to the same consequences in many practical cases. An OWL 2 ontology that adhere to certain structural restrictions which essentially ensure that it can be translated into a $\mathcal{SROIQ}$ knowledge base is typically called OWL 2 DL ontol-

ogy and interpreted with the OWL 2 Direct Semantics. On the contrary, the term OWL 2 Full refer to OWL 2 ontologies that do not abide by any syntactic constraints and consequently can only be interpreted under the OWL 2 RDF-Based Semantics.

In addition to OWL 2 DL and OWL 2 Full, OWL 2 further specifies the so-called OWL2 Profiles [Motik et al., 2012] that offer favourable computational properties for certain application scenarios. For this purpose, OWL 2 DL is already too large, since it only admits reasoning algorithms that run in worst-case nondeterministic double-exponential time. Each profile is defined as a language fragment of OWL 2 DL and trades off different aspects of OWL's expressive power for efficiency of reasoning and/or implementational benefits.

The OWL 2 EL profile is tailored for applications employing very large but lightweight ontologies that consist mainly of terminological data. This profile captures the expressive power used by many such ontologies, in particular in the life sciences, e.g, SNOMED-CT, the NCI thesaurus, and Galen. The EL acronym reflects the profile's basis in the EL family of description logics and most specifically in the DL $\mathscr{EL}^{++}$ for which the basic reasoning problems can still be solved in worst-case polynomial time with respect to the size of the ontology. Dedicated reasoning algorithms for this profile are available and have been demonstrated to be implementable in a highly scalable way.

The profile OWL 2 QL is aimed at applications that use very large volumes of instance data, and where query answering is the most important reasoning task. OWL 2 QL enables data (assertions) stored in a standard relational database system to be queried through an ontology via a simple rewriting mechanism, i.e., by rewriting the query into an SQL query that is then answered by the RDBMS system, without any changes to the data. Ontological information is merely used in a query preprocessing step to augment the expressivity of a relational query language. The approach is known as Ontology Based Data Access (OBDA). The logical underpinning for OWL 2 QL is provided by the DL-Lite family of description logics and more precisely by DL-lite$_R$ for which sound and complete conjunctive query answering can be performed in LogSpace (more precisely, AC_0 with respect to the size of the data). As in OWL 2 EL, polynomial time algorithms can be used to implement the basic reasoning problems.

OWL 2 RL is aimed at applications that can trade the full expressivity of the language for scalability and efficiency of reasoning, as well as RDF(S) applications that need some added expressivity. Instance retrieval is the most important inference task in the profile. Applications often involve a large amount of explicit facts, which are augmented by a set of TBox axioms that is much smaller (typically by at least one order of magnitude). This situation is common for ontologies that are obtained by crawling the Semantic Web, but it is also typical for OBDA applications. The design of OWL 2 RL is inspired by Description Logic Programs, i.e., DLs that are syntactically restricted in such a way that axioms could also be read as rules in first-order Horn logic without function symbols[6]. Due to this characteristic, DLP-type logics can be considered as kinds of rule languages (hence the name OWL 2 RL) contained in DLs. Polynomial time reasoning algorithms for the standard reasoning tasks in OWL 2 RL can be implemented using rule-extended database technologies operating directly on RDF triples.

As mentioned earlier in this section, the ability to use DL reasoners to provide reasoning services for OWL applications was one of the motivations for basing the design of OWL on a DL. Different ontology design tools, both "academic" and commercial, now exploit the correspondence between OWL and $\mathcal{SROIQ}$ in order to support ontology design and maintenance by, for instance, highlighting inconsistent classes and implicit subsumption relationships. Examples of such editors include Protege[7] and TopBraid Composer[8]. The OWL API library[9] is the de-facto standard for creating and manipulating OWL ontologies nowadays. It is a Java-based framework with support for the different OWL 2 serialization formats that further provides a direct interface for reasoning systems. This enables a uniform and simple use of reasoners and supports, among others, the basic reasoning tasks (cf. Section 2.2).

[6] In practice this is accomplished by allowing different syntactic forms for subconcepts and superconcepts in concept inclusion axioms.

[7] protege.stanford.edu/

[8] http://www.topquadrant.com/tools/ide-topbraid-composer-maestro-edition/

[9] http://owlapi.sourceforge.net/

2.4 Algorithmic Approaches to DL Reasoning

A variety of reasoning techniques can be used to solve the reasoning problems introduced in Section 2.2. The majority of them originate from well-known approaches for theorem proving in the setting of First Order Logic. However, in contrast to FOL for which sound, complete and terminating reasoning methods does not exists, approaches to reasoning in Description Logics aim at providing sound and complete decision procedures. In order to understand why this is important, consider that a trivial sound and incomplete algorithm answers NO to every input, while a trivial unsound but complete algorithm answers YES to every input. Clearly, both can be seen as irrelevant if not taken together. Moreover, the adopted reasoning techniques have to be adapted in order to guarantee termination.

Most state-of-the art OWL reasoners for expressive DLs, such as Pellet [Sirin et al., 2007], FaCT++ [Tsarkov and Horrocks, 2006], and HermiT [Glimm et al., 2014], use tableau based methods first introduced by Schmidt-Schaubßand Smolka [Schmidt-Schaubßand Smolka, 1991]. Although a wide range of optimization techniques have been developed for these systems due to their long availability and usage, they are not always applicable in practice, e.g. in presence of very large knowledge bases. In order to address scenarios in which tableau algorithms perform poorly, other approaches have been investigated. Successful examples in this respect are the consequence-based saturation approaches (for sub-Boolean DLs) [Kazakov, 2009, Kazakov et al., 2014] and some works based on resolution [Hustadt et al., 2007, Kazakov and Motik, 2008, Motik and Sattler, 2006].

In this section, we present the most relevant reasoning techniques for DLs and briefly discuss their practical applications.

2.4.1 Tableau Based Calcoli

In the following we will concentrate on knowledge base consistency because, as we have seen in Section 2.2, this is a very general problem to which for more expressive Description Logics all the others can be reduced. For example, a concept C is subsumed by a concept D with respect to a knowledge base ($\mathscr{T}$,

$\mathscr{R}$,$\mathscr{A}$) if ($\mathscr{T}$, $\mathscr{R}$,$\mathscr{A} \cap \{(C \sqcup \neg D)(x)\}$) is inconsistent, where x is a new individual name. Note that forming the concept $C \sqcup \neg D$ obviously relies on having full negation in the logic.

Tableau algorithms try to prove the consistency of a knowledge base $\mathscr{KB} = (\mathscr{T}, \mathscr{R}, \mathscr{A})$ by constructing a model of $\mathscr{KB}$. A *tableau* is a graph which represents such a model, with nodes corresponding to individuals and edges corresponding to relationships between individuals. A typical algorithm start with the concrete situation described in $\mathscr{A}$ and try to construct a tableau, by inferring the existence of additional individuals or constraints on individuals implied by the axioms in $\mathscr{T}$ and $\mathscr{R}$. The inference mechanism consists of applying a set of expansion rules. For any given language Ł, a different set of expansion rules is defined. Generally, there is one to one correspondence between the expansion rules and the logical constructs of Ł. The algorithm terminates either when no further inferences are possible, or when contradictions have been detected. Non-determinism is dealt with by exploring all possible expansions.

In order to illustrate the main ideas behind the tableau procedures, a tableau algorithm for $\mathscr{ALC}$ knowledge base consistency is presented below [Baader et al., 1996, Baader et al., 2010]. We assume concepts to be in negation normal form (NNF) that is with negations only applying to concept names. An arbitrary $\mathscr{ALC}$ concept can be converted to negation normal form by pushing negations inwards using a combination of equivalences and de Morgan's laws: $\neg\exists R.A \equiv \forall R.\neg C$ and $\neg\forall R.C \equiv \exists R.\neg C$. The algorithm works on a data structure called completion forest[10], i.e., a labeled directed graph each node of which is the root of completion tree. Each node x in the graph is labeled with a set of concepts (x), whilst each edge $\langle x, y\rangle$ is labeled with a set of role names $(\langle x, y\rangle)$. When a concept C is in the label of a node x ($C \in (x)$) it represents a model in which the individual corresponding to x is in the extension of C. Furthermore, when an edge $\langle x, y\rangle$ is labeled with R ($R \in (\langle x, y\rangle)$), it represents a model in which the tuple corresponding to $\langle x, y\rangle$ is in the extension of R. A node y is called an R-successor of a node x if there is an edge $\langle x, y\rangle$ labeled with R (and x is called a predecessor of y); x is called an ancestor of y if x is the predecessor of y or there

[10]Since $\mathscr{ALC}$ has the so called forest model property, the model we aim to construct has the form of a set of trees.

exists some node z such that z is the predecessor of y and x is an ancestor of z.

Definition 2.4.1 (Clash) A completion forest contains a clash iif there is a node x such that

- $\bot(x) \subseteq (x)$, or
- $\{C, \neg C\} \subseteq (x)$ for some concept C.

Table 2.6. Tableaux expansion rules for $\mathscr{ALC}$

$\sqcap$-rule	if	1.	$C_1 \sqcap C_2 \in (x)$
		2.	$\{C_1, C_2\} \not\subseteq (x)$
	then		set $(x) = (x) \cup \{C_1, C_2\}$
$\sqcup$-rule	if	1.	$C_1 \sqcup C_2 \in (x)$
		2.	$\{C_1, C_2\} \cap (x) = \emptyset$
	then		set $(x) = (x) \cup \{C\}$ for some $C \in \{C_1, C_2\}$
$\exists$-rule	if	1.	$\exists R.C \in (x)$
		2.	x has no R-successor y with $C \in (y)$,
	then		create a new node y with $(\langle x, y\rangle) = R$ and $(y) = \{C\}$
$\forall$-rule	if	1.	$\forall R.C \in (x)$
		2.	there is an R-successor y of x with $C \notin (y)$,
	then		set $(y) = (y) \cup \{C\}$
$\sqsubseteq$-rule	if	1.	$C_1 \sqsubseteq C_2 \in \mathscr{T}$
		2.	$C_2 \sqcup \neg C_1 \notin (x)$
	then		set $(x) = (x) \cup \{C_2 \sqcup \neg C_1\}$

For $\mathscr{ALC}$ knowledge base $\mathscr{KB} = (\mathscr{T}, \mathscr{A})$, the completion forest $\mathscr{F}_{\mathscr{A}}$ initially contains a root node x_a, with $L(x_a) = \{C \mid a : C \in \mathscr{A}\}$, for each individual name a occurring in $\mathscr{A}$, and an edge $\langle x_a, x_b\rangle$, with $(\langle x_a, x_b\rangle) = \{R \mid (a, b) : R \in \mathscr{A}\}$, for each pair (a, b) of individual names for which the set $\{R \mid (a, b) : R \in \mathscr{A}\}$

is not empty. The algorithm proceeds by exhaustive application of the expansion rules shown in Table 2.6. The rules syntactically decompose the concepts in node labels resulting either in extending node labels or in adding new leaf nodes and edges, thereby explicating the structure of a model for $\mathscr{KB}$. Notice how each rule consists of a precondition and an action, where the action is only applied if the precondition is met. For example, if $C_1 \sqcap C_2 \in (x)$, and either $C_1 \notin (x)$ or $C_2 \notin (x)$, then the $\sqcap$-rule adds C_1 and C_2 to (x); if $\exists R.C \in (x)$, and x does not yet have an R-successor labeled with C, then the $\exists$-rule generates a new R-successor node y of x with $(y) = \{C\}$. In contrast to the other rules, the $\sqcup$-rule is non deterministic: if there is a disjunctive concept $C_1 \sqcup C_2 \in (x)$ and neither $C_1 \in (x)$ nor $C_2 \in (x)$, then either C_1 or C_2 is added to (x). In practice, the algorithm may need to explore all possible choices of rule application. It backtracks and tries to apply some of the (non-deterministic) expansion rules in a different way if a *clash*, is detected, i.e., if the same individual must satisfy obviously conflicting constraints. Searching non-deterministic expansions is the main cause for poor performance of tableaux procedures.

In case of non-empty TBox, the $\sqsubseteq$-rule takes care that each node of the completion graph indeed satisfies all axioms of $\mathscr{T}$. However, the expansion process, as it is, may not terminate in presence of GCIs. In order to guarantee cycle detection a technique called *blocking* is used. Intuitively, a node X can become blocked when the sub-tree rooted in x will be "similar" to the sub-tree rooted in some predecessor y of x. In $\mathscr{ALC}$, a form of blocking known as *subset blocking* is used.

Definition 2.4.2 (Subset Blocking) A node x is *blocked* if it is either directly or indirectly blocked. A node x is *indirectly blocked* if an ancestor of x is blocked. A node x is *directly blocked* if an ancestor y of x exists such that $(x) \subseteq (y)$ and none of its ancestors is blocked.

Termination is regained by suitably modifying the set of the expansion rules: a new precondition is added that prevent application of each rule to an individual x if it is blocked. Blocking can also lead to a more complex correspondence between the completion forest and a model of the knowledge base. In particular, a branch that contains a directly blocked node x represents an infinite branch

with a regular structure in the corresponding model. More precisely, the section between the blocker and the blocked node x in the branch must be repeated or *"unraveled"* which typically leads to infinite models [Horrocks and Sattler, 1999].

A completion forest is fully expanded when none of the expansion rules can be applied. If a fully expanded and clash-free completion forest can be found, then, one can use it to build a model that witnesses the consistency of the knowledge base so the algorithm returns "$\mathscr{KB}$ is consistent". Otherwise, the obtained completion forest contains an obvious inconsistency, and thus does not represent a model, so the algorithm answers "$\mathscr{KB}$ is inconsistent".

The tableau based decision procedure for the consistency of general $\mathscr{ALC}$ knowledge bases described above runs in worst-case nondeterministic double exponential time (due to the fact that the algorithm is searching trees of worst-case exponential depth). By reusing intermediate search results a similar algorithm can be made to run in exponential time [Donini and Massacci, 2000]. However, this introduces a substantial overhead which turns out not always useful in practice.

The algorithm can be simplified if $\mathscr{T}$ is definitorial. In this case, reasoning with a knowledge base $\mathscr{KB} = (\mathscr{T}, \mathscr{A})$ can be reduced to the problem of reasoning with a knowledge base with an empty TBox by recursively unfolding the concepts used in the ABox axioms: for a concept name A, defined in $\mathscr{T}$ by an axiom $A \equiv D$, all occurrences of A in $\mathscr{A}$ can be replaced with D; for a concept name A, defined in $\mathscr{T}$ by an axiom $A \sqsubseteq D$, all occurrences of A in $\mathscr{A}$ are substituted with the concept $A' \sqcap D$, where A' is a new concept name not occurring in $\mathscr{KB}$ that represents the unspecified characteristics that differentiate A from D. The consistency of the resulting knowledge base is independent of the axioms in $\mathscr{T}$. Used in this way, *static unfolding* has the advantage that it avoids unnecessary application of $\sqsubseteq$-rule to every individual name x and TBox axiom, and the resulting search of different possible expansions. The procedure, however, can lead to an exponential increase in the size of the ABox [Nebel, 1990]. In practice, it is much more efficient to retain the structure of the knowledge base for as long as possible, and to take advantage of it during consistency checking. This can be done by using *lazy unfolding*[Lutz, 1999, Baader et al., 2010], i.e., concepts are only unfolded as required by the progress of the consistency checking algorithm. For the tableau

algorithm, this means that a defined concept A is only unfolded when it occurs in a label of a node. In general, lazy unfolding can be achieved by the additional tableau expansion rules described in Table 2.7. As in the case of static unfolding, $\sqsubseteq$-rule is no longer required[11]. Used in this way, lazy unfolding additionally avoids unfolding of irrelevant subconcepts, either because a non-deterministic expansion choice leads to a complete and clash free tree, or because a contradiction is discovered without fully expanding the tree.

Table 2.7. Lazy unfolding rules

U_1-rule	if 1.	$A \in (x)$ and $A \equiv C \in \mathscr{T}$
	2.	$C \notin (x)$
	then	set $(x) = (x) \cup \{C\}$
U_2-rule	if 1.	$\neg A \in (x)$ and $A \equiv C \in \mathscr{T}$
	2.	$\neg C \notin (x)$
	then	set $(x) = (x) \cup \{\neg C\}$
U_3-rule	if 1.	$A \in (x)$ and $A \sqsubseteq C \in \mathscr{T}$
	2.	$C \notin (x)$
	then	set $(x) = (x) \cup \{C\}$

The tableau algorithm can be easily extended to deal with a wide range of other DLs (see [Baader and Sattler, 2001] for an overview). Extending the algorithm to deal with new features is mainly a matter of adding expansion rules to deal with the new constructors (e.g., number restrictions), and more sophisticated clash and blocking condition in order to preserve both soundness and termination when using an extended rule set. A range of worst-case optimal tableau algorithms have also been proposed for several expressive DLs, such as SHIO [Nguyen, 2014], $\mathscr{SHOQ}$ [Nguyen and Golinska-Pilarek, 2014], and even $\mathscr{SHOIQ}$ [Duc et al., 2012]. Note though that other variants of tableau algorithms are usually used for actual implementations of reasoning engines due to

[11] Blocking is also no longer required as the rest of the rules only introduce concepts that are smaller than the concept triggering the rule application.

the fact that it is often not clear how important optimizations, such as dependency directed backtracking can be adapted such that they can be used with these worst-case optimal algorithms.

2.4.2 Completion and Consequence Based Saturation Procedures

Instead of building (counter)models for subsumption relations, saturation-based procedures derive logical consequences for a given knowledge base by exhaustively applying specific inference rules. Such approaches are mainly employed for DL languages that can be handled deterministically in the sense that only one model has to be considered for the computation of reasoning tasks. In particular, important examples of saturation algorithms are the polynomial-time completion-based and consequence-based procedures for the OWL 2 EL profile. Even though both variants are closely related and can be converted into one another, distinction is necessary mostly due to presentational differences.

The first completion-based saturation procedures has been developed for $\mathscr{EL}$ with terminological cycles [Baader, 2003b], GCIs and role hierarchies [Brandt, 2004]. These results were further extended to the DL $\mathscr{EL}^{++}$ [Baader et al., 2005a], and the support of reflexive roles and range restrictions [Baader et al., 2005a, Baader et al., 2008]. In addition, it is shown in [Baader et al., 2005a, Baader et al., 2008], that basically all other additions of typical DL constructors to $\mathscr{EL}^{++}$ make subsumption w.r.t. general TBoxes EXPTIME-complete. A completion-based algorithm for a given knowledge base $\mathscr{KB} = (\mathscr{T}, \mathscr{R}, \mathscr{A})$ proceeds in three steps: it first normalizes the TBox $\mathscr{T}$ and RBox $\mathscr{R}$; then translate the normalized $\mathscr{T}$ and $\mathscr{R}$ into a graph representation and finally complete the graph using a set of completion rules. Note that the algorithm actually classifies, i.e., it simultaneously computes all subsumption relationships between the concept names occurring in $\mathscr{T}$.

An $\mathscr{EL}^{+}_{\bot}$ knowledge base is in *normal form* if it only contains concept inclusions of the following forms:

$$C \sqsubseteq D, \qquad C \sqsubseteq \exists R.D, \quad C_1 \sqcap C_2 \sqsubseteq D, \quad \exists R.D \sqsubseteq C.$$

and all role inclusions are of the form $R_1 \sqsubseteq R_2$ or $R_1 \circ R_2 \sqsubseteq S$. Any $\mathscr{KB}$

can be transformed into normalized one $\mathscr{K}\mathscr{B}$' that is *conservative extension* of $\mathscr{K}\mathscr{B}$, i.e. induce the same models of $\mathscr{K}\mathscr{B}$, by repeatedly replacing complex concept and role expressions with fresh concepts and role names and adding the corresponding equivalences, followed by subsequent simplifications. Such a normal form can easily be computed in polynomial time and does not increase the size of $\mathscr{T}$ and $\mathscr{R}$ more than polynomially (cf. [Baader et al., 2005b]). Next, the algorithm build a classification graph $\mathscr{G} = (V, V \times V, S, R)$ where:

- V is the set of concept names occurring in the normalized TBox including $\{\top, \bot\}$;
- S labels nodes with set of concept names in V;
- R labels edges with sets of role names occurring in the normalized TBox and RBox.

The intuition is that these sets make implicit subsumption relationships explicit in the following sense:

- $B \in S(A)$ implies $A \sqsubseteq B$,
- $T \in R(A, B)$ implies $A \sqsubseteq \exists T.B$.

Initially, $S(A) = \{A, \top\}$ for all nodes $A \in V$ and $R(A, B) = \emptyset$ for all edges $(A, B) \in V \times V$. The labels of nodes and edges are then extended by applying the rules of Figure 2.8 until no more rule applies. The fact that subsumption in $\mathscr{EL}^+_\bot$ can be decided in polynomial time is an immediate consequence of the following observations:

1. The rules can only be applied a polynomial number of times, and each rule application is polynomial.
2. When no more rules are applicable, then $A \sqsubseteq B$ iff $B \in S(A)$.

Thus the algorithm already classifies, i.e., it simultaneously computes all subsumption relationships between the concept names occurring in $\mathscr{T}$. Furthermore, this is done in one pass compared to approaches which reduce each subsumption

to a separate consistency check, e.g. tableaux, which explains why such algorithms are used to efficiently handle (very) large knowledge bases.

In contrast to completion-based, a deduction rule in a consequence-based procedures has the shape

$$\frac{\alpha_1, ..., \alpha_n}{\alpha}$$

where $\alpha_1, ..., \alpha_n$ are axioms of the underlying logic. Derived consequences α are also represented as axioms. Hence, the input knowledge base is not required to be normalized. For example, consider a knowledge base $\mathscr{K}$ containing axioms $A \sqsubseteq B$ and $B \sqsubseteq \mathscr{C}$, the consequence $A \sqsubseteq \mathscr{C}$ of $\mathscr{K}$ can be derived by such procedure with an appropriate deduction rule.

Table 2.8. Completion rules specific to $\mathscr{EL}^+_\bot$ knowledge bases [12]

CR1	if	$D \in (C)$, $D \sqsubseteq E \in \mathscr{K}\mathscr{B}$, and $E \notin (C)$
	then	$(C) = (C) \cup \{E\}$
CR2	if	$D_1, D_2 \in (C)$, $D_1 \sqcap D_2 \sqsubseteq D \in \mathscr{K}\mathscr{B}$, and $D \notin (C)$
	then	$(C) = (C) \cup \{D\}$
CR3	if	$E \in (C)$, $E \sqsubseteq \exists R.D \in \mathscr{K}\mathscr{B}$, and $(C, D) \notin (R)$
	then	$(R) = (R) \cup \{(C, D)\}$
CR4	if	$(E, C) \in (R)$, $D_1 \in (C)$, $\exists R.D_1 \sqsubseteq D_2 \in \mathscr{K}\mathscr{B}$, and $D_2 \notin (E)$
	then	$(E) = (E) \cup \{D_2\}$
CR5	if	$(C, D) \in (R)$, $\bot \in (D)$, and $\bot \notin (C)$
	then	$(C) = (C) \cup \{\bot\}$
CR6	if	$(C, D) \in (R)$, $R \sqsubseteq S \in \mathscr{K}\mathscr{B}$, and $(C, D) \notin (S)$
	then	$(S) = (S) \cup \{(C, D)\}$
CR7	if	$(E, C) \in (R_1)$, $(C, D) \in (R_2)$, $R_1 \circ R_2 \sqsubseteq S \in \mathscr{K}\mathscr{B}$, and $(E, D) \notin (S)$
	then	$(S) = (S) \cup \{(E, D)\}$

A consequence-based algorithm for $\mathscr{EL}^+_\bot$ [Kazakov et al., 2014] – at the core of the OWL 2 EL reasoner ELK.

In [Kazakov, 2009] Kazakov extend the approach to the more expressive Horn fragment Horn-$\mathscr{SHIQ}$ (known as the CB reasoner). Further extensions to DLs with non-deterministic features have been proposed in [Simančík et al., 2011] ($\mathscr{ALCH}$), [Simancik et al., 2014] ($\mathscr{ALCI}$) and more recently for $\mathscr{ALCHI}$ in [Kazakov and Klinov, 2014]. The main idea behind these extended consequence-based procedures is to saturate all non-deterministic branches (as opposed to tableaus which performing case-by-case analysis). The consequences are only those subsumptions that can be derived regardless of the choices made in the application of the deduction rules. It is however not completely clear whether such approaches can scale well for arbitrary knowledge bases that make a frequent use of non-deterministic languages features. Moreover, to the best of our knowledge,a saturation procedure that is capable of handling DLs that allow for inverse roles and cardinality restrictions does not (yet) exist.

2.4.3 Automata Based Approaches

Automata based approaches can as well be used to decide the basic reasoning problems for DLs with the tree model property. The basic idea is to devises a translation from a knowledge base $\mathscr{KB}$ into an appropriate tree automata $\mathscr{A}$ such that $\mathscr{A}$ accepts exactly the tree models of $\mathscr{KB}$. The problem of knowledge base satisfiability will then be reduced to the emptiness test for the employed automaton model to $\mathscr{A}$. The complexity of the algorithm obtained this way depends on the complexity of the translation and the complexity of the emptiness tests. Thus various instances of the automata based approach differ not only w.r.t. the DL under consideration, but also w.r.t. the employed automaton model. For instance, the satisfiability of $\mathscr{ALC}$ knowledge bases can be decided using an alternating tree automata within exponential time[13] [Calvanese et al., 1999, Lutz and Sattler, 2000]. For very expressive description logics additional acceptance conditions may be needed such as the Büchi condi-

[12] A subset of the $\mathscr{EL}^{++}$ completion rules presented in [Baader et al., 2005a].

[13] A polynomial translation into alternating tree automata is possible, however, the emptiness test is exponential in the size of the obtained automaton.

tion [Thomas, 1990][14]. Note that although optimal worst-case decision procedure exist for different DLs, they are usually used to establish upper bound complexity results since implementations on average require an exponential encoding of the knowledge base (thus being impractical).

2.4.4 Resolution Based Approaches

The procedures based on resolution – a general theorem-proving method for first-order logic (see, e.g., [Bachmair and Ganzinger, 2001]) – are closely related to the consequence-based saturation algorithms. Similarly, resolution also works by deriving new clauses that are consequences of the original axioms. Over the years, it has been used as a decision procedure for different FOL fragments, modal logics and more recently for DLs. The resolution-based procedures for description logics [de Nivelle et al., 2000] translate DL axioms into first-order clauses (disjunctions of literals) and apply specific resolution strategies which guarantee that only a bounded number of clauses can be derived (thus ensuring termination) and, in many cases, even optimal complexity. In particular, a worst-case optimal resolution-based procedure has been defined for the expressive DL $\mathscr{SHIQ}$ and implemented in the reasoner KAON2 [Hustadt et al., 2008]. An extension to $\mathscr{SHOIQ}$ has been proposed in [Kazakov and Motik, 2008].

Despite being theoretically optimal, resolution-based procedures seem not to be able to compete in practice with modern consequence-based and tableau reasoners. For instance, KAON2 failed to classify any medical ontology in a reported evaluation [Mendez and Suntisrivaraporn, 2009]. The most plausible explanation seems to be that, despite optimizations, resolution still produces many unnecessary consequences.

[14] The Büchi automata acceptance condition requires the occurrence of infinitely many final states in every path.

Chapter 3

Nonstandard Reasoning Services

3.1 The Nonmonotonic Description Logic $\mathscr{DL}^{\mathrm{N}}$

The ontology languages at the core of the semantic web — like RDF and OWL — are based on description logics (DLs), that are fragments of first-order logic or slight extensions thereof, such as fix-point logic. Therefore, DLs inherit limitations of these well-established formalisms that include monotonicity, and the consequent inability to design knowledge bases (KBs) by describing prototypical instances whose general properties can be refined later, with suitable exceptions, that is, *incrementally*. This natural formulation approach has been commonly adopted for centuries in areas such as law and science, and more recently in programming and computer security.

For instance, many laws are formulated by adding new norms whose articles may contradict some of the articles of a previous norm. The result is a combination of old and new articles. The mechanism is similar to *overriding* in object oriented programming (OOP) languages: in law, recent norms override (part of) the old ones; in OOP the definitions in subclasses override any conflicting binding belonging to superclasses. Biologists have adopted prototypical properties and exceptions since the early days of this science. There is an obvious reason: In biology, virtually all universal properties admit some exception. For instance,

the human body has a rather precise structure: the heart is usually located in the left-hand half of the body. Still there are exceptional individuals, with so-called *situs inversus*, whose heart is located on the opposite side. Eukaryotic cells are those with a proper nucleus, by definition. Still they comprise mammalian red blood cells, that in their mature stage have no nucleus.[1] Another application of nonmonotonic DLs stems from the recent development of policy languages based on DLs [Uszok et al., 2003, Finin et al., 2008, Zhang et al., 2009, Kolovski et al., 2007]. DLs nicely capture role-based policies and facilitate the integration of semantic web policy enforcement with reasoning about semantic metadata which is often needed to check policy conditions. However, in order to formulate standard default policies such as *open* and *closed* policies[2], conflict resolution methods such as *denials take precedence*, and authorization inheritance with exceptions, it is necessary to adopt a nonmonotonic semantics (see the survey [Bonatti and Samarati, 2003] for more details).

Historically important frame systems such as LOOM (one of the ancestors of description logics) supported default properties and nonmonotonic reasoning.[3] These features were lost in the formalization process that led to the development of DLs. One of the major obstacles to the deployment of solutions based on nonmonotonic DLs is constituted by their high computational complexity (see the survey in [Bonatti et al., 2009b, Ch. 7] and the results in [Bonatti et al., 2011b]), combined with the absence of effective optimization techniques. Some of the most effective optimizations of DL reasoning, such as tableaux caching and dependency-directed backtracking [Baader et al., 2010], rely on the monotonicity of classical DLs. Given the large size of semantic web ontologies and RDF bases, it is mandatory that reasoning in nonmonotonic DLs is extremely efficient, possibly feasible in polynomial time. Unfortunately, to the best of our knowledge no nonmonotonic extension except for $\mathscr{SROEL}\,(\sqcap,\times)^{\mathbf{R}}\mathbf{T}$ [Giordano and Dupré, 2016] do preserve the tractability of low-complexity DLs [Cadoli et al., 1990, Bonatti et al., 2009a]. Usually, the complexity of nonmonotonic DL reasoning is significantly more complex than reasoning in the underlying,

[1] All of these examples are introduced and discussed in [Rector, 2004, Stevens et al., 2007].

[2] If no explicit authorization has been specified for a given access request, then an open policy permits the access while a closed policy denies it.

[3] http://www.isi.edu/isd/LOOM/documentation/LOOM-DOCS.html

monotonic DLs.

Another problem is that none of the standard nonmonotonic semantics produces exactly the set of expected consequences, and this can be verified on a range of rather simple examples (details will be given in Section 3.1.7). Circumscription, Default logic, Autoepistemic logic, Rational closure, all have complementary strengths and weaknesses that make it difficult to propose a single nonmonotonic extension as the reference semantics of nonmonotonic inheritance and overriding in description logics. In several cases, some natural, desirable inferences are missing for subtle reasons that are quite difficult to track,[4] and would make it hard for a knowledge engineer to formulate and validate complex knowledge bases.

Supporting default attributes and exceptions was important enough to look for alternative representation methods, based on classical DLs. The simplest examples can be dealt with by means of ontology design patterns [Rector, 2004, Stevens et al., 2007]. Unfortunately, these solutions do not scale to more complex examples with multiple exception dimensions, as discussed in [Rector, 2004]: The number of additional concepts introduced by the patterns may grow exponentially. Moreover, such auxiliary concepts must constitute a partition of the original concept, which requires computationally expensive constructs such as disjunction. Consequently, even if a given knowledge base belongs to some low-complexity fragment (e.g., some OWL2 profile), its nonmonotonic extension is generally not tractable.

After extensive investigations, and in vain attempts to find a proper way to address the application requirements discussed above by means of standard nonmonotonic semantics, we came to the conclusion that a new semantics is needed, tailored to nonmonotonic inheritance and overriding. Ideally, the semantics should be easy to grasp, its inferences should be reasonably predictable, and its complexity should be comparable to the complexity of the underlying monotonic description logic. Nonmonotonic features should be applicable also to important domains such as biomedical ontologies that push automated reasoning technology to its limits. A related desideratum is that nonmonotonic inferences should be implementable by re-using as much as possible the well-engineered tools and engines available for semantic web reasoning, in order to exploit all

[4] Cf. Section 3.1.7.

the sophisticated optimization techniques developed across decades of research on automated reasoning, as well as the standardization efforts carried out so far.

We select a few, well identified nonmonotonic features and design a semantics that provides a satisfactory support to those features, in the sense that it makes some recurrent representation and inference patterns easy to formulate and implement. In particular, we do not aim at covering all potentially interesting forms of nonmonotonic reasoning[5]. We focus on analogues of what McCarthy calls *communication and database storage conventions*, and *policy representation* [McCarthy, 1986]. In the communication and database storage convention perspective, nonmonotonic constructs are meant to factorize the common features of a majority of individuals and confine explicit detailed axiomatization to a restricted number of exceptional individuals in order to reduce the size and cost of knowledge bases and improve their readability. Similarly, from the policy representation perspective, the goal is a concise and neat, possibly incremental formulation of a policy that is best described by factorizing some general rules, and refining them with suitable exceptions.

A distinguishing aspects of the new approach is a *prototype oriented* semantics. In the examples taken from biomedical domains, policies, etc., the default properties associated to a concept A describe a typical member of A. We will call such a member a *prototype*.[6] The prototypical features of A are expected to be subject to *overriding* in some of the concepts B subsumed by A (the default properties specific to B may be inconsistent with those of A), but every single prototype, in our reference scenarios, is *internally coherent*, that is, the default properties that characterize a prototype are consistent with the classical axioms of the knowledge base.

On the contrary, the artificial examples occurring in the literature (especially

[5] A non-exhaustive list of such additional requirements and mechanisms taken from the literature comprises: preserving as many of the KLM axioms as possible; giving defeasible inheritance an independent philosophical foundation (e.g. probabilistic); defining overriding through complex criteria such as argumentation, or predicates over inclusion paths (as in inheritance networks); restricting entailment to the invariant consequences across a set of "admissible" priority orderings; maximizing the *number* of satisfied default properties; defining normal individuals through an absolute, global normality ordering, independent from any given concept C.

[6] To be precise, unlike the prototypes dealt with in philosophy that might not exist in the real world due to their degree of perfection, the prototypes introduced in the above domains typically correspond to a number of real instances. This is coherent with the utilitarian view underlying McCarthy's *conventions* [McCarthy, 1986].

the reductions adopted in proving lower complexity bounds) make massive use of inconsistent prototypes: one of the most frequent axiom patterns comprises one or more conflicting, nonmonotonic axioms with the same priority. From a practical perspective—in the light of our reference scenarios—such prototypes may be regarded as an abuse of nonmonotonic constructs.

Circumscription, Default logic, and Autoepistemic logic, deal with inconsistent prototypes by maximizing the number of satisfied nonmonotonic axioms. Intuitively, it is like identifying all optimal repairs of the inconsistent prototype and computing the inferences that hold for all repairs. Of course, in writing a knowledge base that models a concrete scenario, computing the invariants across all repairs is not necessarily the right approach. There may be a single meaningful application dependent way of removing the inconsistency, and the knowledge engineer should be involved in deciding how to repair the prototype.

Example 3.1.1 Consider the famous *Nixon's diamond*:

1. Quakers are normally pacifist;
2. Republicans are normally not pacifist;
3. Nixon is both a quaker and a republican.

Here the concept Nixon is associated to an inconsistent prototype consisting of the properties "pacifist" and "not pacifist" by multiple inheritance. When the above statements are formalized, multiple circumscription models exist, as well as multiple default extensions or autoepistemic expansions. Some satisfy the first statement and make Nixon a pacifist, while others apply the second statement and make Nixon not a pacifist. Each of these models (or extensions, or expansions) corresponds to an optimal repair of the prototype where Nixon satisfies exactly one of the two nonmonotonic axioms. However, history tells us that only one of them actually holds. Since the above properties of Quakers and Republicans are perfectly symmetric, this problem cannot be resolved by the logic: Choosing the right set of properties satisfied by Nixon is a knowledge engineer's task. She can fix the prototype by overriding the properties that should not be inherited, e.g. by stating that Nixon is not a pacifist. The knowledge engineer might even decide that individuals may exhibit a mix of pacifist and non-pacifist behavior,

so that who is a Quaker *and* a Republican should not be forced to be definitely a pacifist or definitely not a pacifist in all models (or extensions, or expansions) of the knowledge base. Of course, such decisions must start from the awareness that some concept (e.g. Nixon, or Quaker-and-Republican) has been inadvertently associated to an inconsistent prototype. If a nonmonotonic semantics hides this "inconsistency" by repairing it, then the whole repair process can possibly remain unnoticed by the knowledge engineer . ∎

The rationale behind highlighting inconsistent prototypes is even clearer in ontology merging activities, as illustrated by the next example, due to [Rector, 2004].

Example 3.1.2 Suppose a knowledge engineer is merging two ontologies that describe the anatomy of humans and mice, respectively. In the former ontology, the body has one prostate with three lobes, while in the latter ontology bodies have five prostates, none of which has lobes. Now consider the union of the two ontologies. The nonmonotonic semantics introduced so far in the literature would resolve the conflict between the prostate axioms by allowing each body to have either a single prostate with three lobes, or five prostates with no lobe; no conflict resolution strategy would extend the knowledge base signature. However, in this case, extending the signature is exactly what should be done: it would be better to notify the conflict to the knowledge engineer, who could then refine the concept `Body` by introducing two subclasses: `HumanBody` and `MouseBody`, each satisfying the corresponding prostate axiom. ∎

According to the above discussion, in the new semantics inconsistent prototypes will be regarded as knowledge representation errors. Identifying inconsistent prototypes is considered as a debugging step analogous to detecting inconsistent concepts. Knowledge engineers are responsible of deciding how to repair a certain prototype. Only the conflicts that can be settled by a clear priority relation between nonmonotonic axioms shall be resolved by the logic.

Another important feature of $\mathscr{DL}^{\mathrm{N}}$ is that the semantics do not maximize the sets of normal instances; this prevents the undesirable closed-world assumption effects discussed in Section 3.1.7.

To the best of our knowledge no other nonmonotonic DL has all of the above features. —actually, the overriding mechanism is novel, despite its simplicity, and

contributes of the unique computational properties of $\mathcal{DL}^{\mathrm{N}}$. The final result is that $\mathcal{DL}^{\mathrm{N}}$ is the only nonmonotonic description logic that enjoys *all* of the following properties:

- it yields the expected inferences in all the applicative examples, avoiding undesired side-effects and common shortcomings
- it supports ontology engineering by highlighting inconsistent prototypes;
- it achieves the above goals without increasing the computational complexity of the classical reasoning tasks such as subsumption and instance checking, concept consistency checking, and knowledge base consistency checking; to the best of our knowledge, $\mathcal{DL}^{\mathrm{N}}$ is the first nonmonotonic description logic that preserves the tractability of the above tasks over the $\mathcal{EL}$ family and the DL-lite family.

In this section we introduce the syntax and the semantics of the new logic of overriding called $\mathcal{DL}^{\mathrm{N}}$. Section 3.1.2 illustrate the behavior of $\mathcal{DL}^{\mathrm{N}}$ in a number of examples inspired by the intended applications of the new logic (Section 3.1.2). Automated reasoning in $\mathcal{DL}^{\mathrm{N}}$ is carried out by means of the reduction to classical description logics introduced in Section 3.1.3.Then, in Section 3.1.5, we resume the technical analysis of the new logic and show some semantic and logical properties of $\mathcal{DL}^{\mathrm{N}}$. Section 3.1.6 focuses on some guidelines for the usage $\mathcal{DL}^{\mathrm{N}}$, including representation methodologies, and elimination techniques for constructs that are extensively used in $\mathcal{DL}^{\mathrm{N}}$ but are not supported by all description logics. Finally, in Section 3.1.7, $\mathcal{DL}^{\mathrm{N}}$ is compared in detail with the other major nonmonotonic description logics and design patterns 3.1.7. Proofs are omitted in order to improve readability. The interested reader can refer to [Bonatti et al., 2015a] for more details.

3.1.1 Knowledge Bases, Defeasible Inclusions, and Overriding

Let $\mathcal{DL}$ be any classical description logic language, and $\mathcal{DL}^{\mathrm{N}}$ be the extension of $\mathcal{DL}$ with a new type of concept expressions NC for each $\mathcal{DL}$ concept C,

called *normality concepts*, meant to denote the *normal* or *prototypical instances* of C.

A $\mathscr{DL}^{\mathrm{N}}$ *interpretation* $\mathscr{I} = \langle \Delta^{\mathscr{I}}, \cdot^{\mathscr{I}} \rangle$ is any extension of a classical interpretation of $\mathscr{DL}$ such that $\mathrm{N}C^{\mathscr{I}} \subseteq C^{\mathscr{I}}$. In other words, $\mathscr{DL}^{\mathrm{N}}$ interpretations treat each normality concept $\mathrm{N}C$ as a new concept name that satisfies $\mathrm{N}C \sqsubseteq C$.

Definition 3.1.3 [$\mathscr{DL}^{\mathrm{N}}$ Knowledge base] A $\mathscr{DL}^{\mathrm{N}}$ *knowledge base* is a disjoint union $\mathscr{KB} = \mathscr{S} \cup \mathscr{D}$ where:

- $\mathscr{S}$ is a finite set of $\mathscr{DL}^{\mathrm{N}}$ axioms called *strong* or classical axioms (that may possibly comprise both inclusions and assertions)
- $\mathscr{D}$ is a finite set of *defeasible inclusions* (DIs), i.e. expressions $C \sqsubseteq_n D$ where C is a $\mathscr{DL}$ concept and D a $\mathscr{DL}^{\mathrm{N}}$ concept.

The informal meaning of $C \sqsubseteq_n D$ is: "*the normal instances of C are instances of D, unless stated otherwise*". By means of a set of DIs $C \sqsubseteq_n D_i$ $(1 \leq i \leq n)$ every concept C can be associated to a set of prototypical properties $D_1, \ldots, D_n$. A normal instance x of C should then conform to these properties *unless stated otherwise*, that is, unless a group of strong axioms and higher priority DIs forces x to satisfy $\neg D_i$, for some $i = 1, \ldots, n$. An alternative informal statement of the semantics of $C \sqsubseteq_n D$ is: "*by default, a normal member of C should satisfy D*".

Note that normality concepts and DIs are utilitarian constructs meant to factorize properties that hold for most of the entities modeled in the knowledge base, so as to minimize the amount of knowledge that must be explicitly encoded. Then the role of normality concepts and DIs is analogous to the role of inheritance and overriding in object oriented languages.

A *pre-model* of $\mathscr{KB}$ is a $\mathscr{DL}^{\mathrm{N}}$ interpretation $\mathscr{I}$ that satisfies all the axioms of its strong part $\mathscr{S}$. As usual, if $\mathscr{I}$ is a *pre-model* then we write $\mathscr{I} \models \mathscr{S}$, and if a sentence α is satisfied by all the *pre-models* of $\mathscr{KB}$ then we say that α is a logical consequence of $\mathscr{S}$ and write $\mathscr{S} \models \alpha$. We will slightly abuse notation and write $\mathscr{KB} \models \alpha$ as an equivalent of $\mathscr{S} \models \alpha$, and $C \sqsubseteq_{\mathscr{KB}} D$ as an equivalent of $\mathscr{S} \models C \sqsubseteq D$.

In order to define the semantics of DIs, some intermediate steps are required. Let $\delta = (C \sqsubseteq_n D)$, then its left-hand side C and its right-hand side D are denoted

respectively by $\mathsf{pre}(\delta)$ and $\mathsf{con}(\delta)$. An individual x in a $\mathscr{DL}^{\mathrm{N}}$ interpretation $\mathscr{I}$ satisfies δ iff either $x \notin \mathsf{pre}(\delta)^{\mathscr{I}}$ or $x \in \mathsf{con}(\delta)^{\mathscr{I}}$. A normality concept N$C$ satisfies δ in $\mathscr{I}$ iff all the elements of N$C^{\mathscr{I}}$ satisfy δ. The set of normality concepts that satisfy δ in $\mathscr{I}$ will be denoted by $\mathsf{sat}^{\mathscr{I}}(\delta)$ and defined formally as:

$$\mathsf{sat}^{\mathscr{I}}(\delta) = \{\mathrm{N}C \mid \forall x \in \mathrm{N}C^{\mathscr{I}},\ x \notin \mathsf{pre}(\delta)^{\mathscr{I}} \vee x \in \mathsf{con}(\delta)^{\mathscr{I}}\}\,.$$

It frequently happens that a concept NC cannot simultaneously satisfy two conflicting DIs. In that case, a choice is made using a *priority relation* over DIs.

A priority relation is a *strict partial order* $\prec$ such that the intended meaning of $\delta_1 \prec \delta_2$ is "δ_1 *has higher priority than* δ_2" and, in case of conflicts, it is preferable to sacrifice the lower priority DI δ_2. While the results reported in this section apply to all priority relations, from now on we will assume that $\prec$ is determined by *specificity* (unless stated otherwise):

$$\delta_1 \prec \delta_2 \text{ iff } \mathsf{pre}(\delta_1) \sqsubseteq_{\mathscr{KB}} \mathsf{pre}(\delta_2) \text{ and } \mathsf{pre}(\delta_2) \not\sqsubseteq_{\mathscr{KB}} \mathsf{pre}(\delta_1)\,. \tag{3.1}$$

In the above definition, the specific properties of $\mathsf{pre}(\delta_1)$ may override those of the more general (less specific) concept $\mathsf{pre}(\delta_2)$.

In each intended model $\mathscr{I}$ of $\mathscr{KB}$, a concept NC should satisfy a DI $\delta \in \mathscr{KB}$ unless the cost of having NC satisfy δ is unacceptable, i.e. if satisfying δ implies that either NC becomes inconsistent or some higher priority $\delta' \in \mathscr{KB}$ must be invalidated. Under these circumstances δ can be ignored for NC, and we say that δ is *overridden* in N$C/\mathscr{I}$. Since higher priority DIs, in turn, can be ignored if they are overridden, the formal definition of overriding is formulated in a recursive fashion:

Definition 3.1.4 [Overriding w.r.t. $\mathscr{KB}$, function ovd] Let $\mathscr{I}$ be a $\mathscr{DL}^{\mathrm{N}}$ interpretation. A DI δ is *overridden in* N$C/\mathscr{I}$ (w.r.t. a knowledge base $\mathscr{KB}$) iff there exists no pre-model $\mathscr{J}$ of $\mathscr{KB}$ satisfying all of the following conditions:

1. N$C \in \mathsf{sat}^{\mathscr{J}}(\delta)$,
2. N$C^{\mathscr{J}} \neq \emptyset$,
3. for all $\delta' \in \mathscr{KB}$ such that $\delta' \prec \delta$, $\mathsf{sat}^{\mathscr{J}}(\delta') \setminus \mathsf{ovd}_{\mathscr{KB}}(\mathscr{I}, \delta') \subseteq \mathsf{sat}^{\mathscr{J}}(\delta')$.

where $\mathsf{ovd}_{\mathscr{K}\mathscr{B}}(\mathscr{I}, \delta')$ denotes the set of all normality concepts ND such that δ' is overridden in N$D/\mathscr{I}$ w.r.t. $\mathscr{K}\mathscr{B}$. In order to improve readability, $\mathscr{K}\mathscr{B}$ will be omitted whenever clear from context.

In other words, δ is *not* overridden in N$C/\mathscr{I}$ iff there exists a model $\mathscr{J}$ of the strong part of $\mathscr{K}\mathscr{B}$ that represents an acceptable way of making NC satisfy δ: Indeed, by condition 1, NC satisfies δ in $\mathscr{J}$; by condition 2, NC is consistent in $\mathscr{J}$; finally, by 3, all of the non-overridden, higher priority DIs of $\mathscr{K}\mathscr{B}$ satisfied in $\mathscr{I}$ by some ND are also satisfied in $\mathscr{J}$ by the same ND (i.e., no higher priority DI is sacrificed, unless it is overridden).

The above recursive definition is well defined since the notion of overriding for δ depends only on the overriding of $\delta' \prec \delta$. In fact, if δ has maximal priority, then condition 3 is vacuously satisfied, so δ is not overridden iff there exists a pre-model $\mathscr{J}$ of $\mathscr{K}\mathscr{B}$ satisfying 1 and 2. Note also that, according to this definition of overriding, a DI δ can be blocked only by higher priority DIs. In particular, it is possible that none of the "acceptable improvements" $\mathscr{J}$ provides a global solution by satisfying also all $\delta' \in \mathscr{K}\mathscr{B}$ incomparable with δ (i.e. $\delta' \not\prec \delta$ and $\delta \not\prec \delta'$). As a consequence, $\mathscr{D}\mathscr{L}^{\mathrm{N}}$ solves only the conflicts that can be settled by the priority relation $\prec$. This is a characterizing feature of $\mathscr{D}\mathscr{L}^{\mathrm{N}}$ that distinguish it from the other nonmonotonic description logics (based on default logic, MKNF, circumscription, etc.) where a nonmonotonic axiom/rule ν can contribute to blocking an incomparable axiom/rule ν', which leads to complex conflict resolution procedures. This however is not appropriate to the inheritance with overriding approach adopted by $\mathscr{D}\mathscr{L}^{\mathrm{N}}$.

Example 3.1.5 The phrase *situs inversus* refers to humans whose heart is located on the right-hand side of the body, differently from typical humans whose heart is on the opposite side. If we agree that no heart can be simultaneously located on both sides, then a simple axiomatization in $(\mathscr{E}\mathscr{L}^{\bot})^{\mathrm{N}}$ is:

$$\texttt{Human} \sqsubseteq_n \exists\texttt{has\_heart}.\exists\texttt{has\_position}.\texttt{Left} \tag{3.2}$$

$$\texttt{SitusInversus} \sqsubseteq \texttt{Human} \tag{3.3}$$

$$\texttt{SitusInversus} \sqsubseteq \exists\texttt{has\_heart}.\exists\texttt{has\_position}.\texttt{Right} \tag{3.4}$$

$$\exists\texttt{has\_heart}.\exists\texttt{has\_position}.\texttt{Left} \sqcap \exists\texttt{has\_heart}.\exists\texttt{has\_position}.\texttt{Right} \sqsubseteq \bot\,. \tag{3.5}$$

We are going to show that in all $\mathscr{DL}^{\mathrm{N}}$-interpretations $\mathscr{I}$, (3.2) is not overridden in N`Human` and is overridden in N`SitusInversus`, due to (3.4).

The above knowledge base $\mathscr{KB}$ contains one DI $\delta =$ (3.2), so the preference relation $\prec$ is irrelevant in this example. Moreover, δ has maximal priority in $\mathscr{KB}$, so we only have to consider conditions 1 and 2 of Def. 3.1.4, as explained above. Note that a $\mathscr{DL}^{\mathrm{N}}$-interpretation $\mathscr{J}$ satisfies condition 1 of Def. 3.1.4 iff $\mathscr{J} \models \mathrm{N}C \sqsubseteq \neg\mathsf{pre}(\delta) \sqcup \mathsf{con}(\delta)$. Therefore, conditions 1 and 2 of Def. 3.1.4 are satisfied by a pre-model of $\mathscr{KB}$ iff there exists a $\mathscr{DL}^{\mathrm{N}}$-model $\mathscr{J}$ of

$$\mathscr{S} \cup \{\mathrm{N}C \sqsubseteq \neg\mathsf{pre}(\delta) \sqcup \mathsf{con}(\delta)\}$$

such that $\mathrm{N}C^{\mathscr{J}} \neq \emptyset$, where $\mathscr{S} = \{(3.3), (3.4), (3.5)\}$ is the strong part of $\mathscr{KB}$.

First, let us focus on $\mathrm{N}C =$ N`Human`. According to the above discussion, (3.2) is *not* overridden in N`Human`$/\mathscr{I}$ iff there exists a $\mathscr{DL}^{\mathrm{N}}$-model $\mathscr{J}$ of

$$\mathscr{S} \cup \{\mathrm{N}\texttt{Human} \sqsubseteq \neg\texttt{Human} \sqcup \exists\texttt{has\_heart}.\exists\texttt{has\_position}.\texttt{Left}\}$$

such that $\mathrm{N}\texttt{Human}^{\mathscr{J}} \neq \emptyset$. Such a $\mathscr{DL}^{\mathrm{N}}$-model obviously exists: for example, take a $\mathscr{J}$ where $\texttt{SitusInversus}^{\mathscr{J}} = \emptyset$, all the instances of $\texttt{Human}^{\mathscr{J}}$ have their heart on the left-hand side of the body, and $\mathrm{N}\texttt{Human}^{\mathscr{J}} = \texttt{Human}^{\mathscr{J}} \neq \emptyset$. This confirms that (3.2) is not overridden in N`Human`$/\mathscr{I}$.

On the contrary, conditions 1 and 2 cannot be possibly satisfied for $\mathrm{N}C =$ N`SitusInversus`, because (3.4), (3.5) imply that N`SitusInversus` and the right-hand side of (3.2) are disjoint; therefore N`SitusInversus` (which is subsumed by `Human`) cannot satisfy (3.2) in $\mathscr{J}$ unless $\mathrm{N}\texttt{SitusInversus}^{\mathscr{J}} = \emptyset$ (which violates condition 2). It follows that (3.2) is overridden in N`SitusInversus`$/\mathscr{I}$, for all $\mathscr{I}$. ∎

Example 3.1.6 The eukaryotic cell example is slightly different from the situs inversus example. If we decide to regard *fully developed* mammalian red blood cells (that have no nucleus) as standard mammalian red blood cells, then we have to introduce two conflicting DIs with different priority. The encoding in $\mathscr{ALC}^{\mathrm{N}}$

is:

$$\texttt{EukCell} \sqsubseteq_n \exists\,\texttt{has\_nucleus} \tag{3.6}$$

$$\texttt{MamRedBldCel} \sqsubseteq \texttt{EukCell} \tag{3.7}$$

$$\texttt{MamRedBldCel} \sqsubseteq_n \neg\exists\,\texttt{has\_nucleus}\,. \tag{3.8}$$

By (3.7), specificity yields (3.8) $\prec$ (3.6), that is, (3.8) has higher priority than (3.6). We are going to show that for all $\mathscr{D}\mathscr{L}^{\mathrm{N}}$-interpretations $\mathscr{I}$, none of the two DIs is overridden in $\mathrm{N}\texttt{EukCell}/\mathscr{I}$, while (3.6) may be overridden by (3.8) in $\mathrm{N}\texttt{MamRedBldCel}/\mathscr{I}$.

Since (3.8) is a maximal priority default, the analysis of where it is overridden can be carried out by analogy with the analysis of (3.2), in the previous example. In particular, (3.8) is overridden in $\mathrm{N}\texttt{EukCell}/\mathscr{I}$ iff there exists no $\mathscr{D}\mathscr{L}^{\mathrm{N}}$-model $\mathscr{J}$ of

$$\mathscr{S} \cup \{\mathrm{N}\texttt{EukCell} \sqsubseteq \neg\texttt{MamRedBldCel} \sqcup \neg\exists\,\texttt{has\_nucleus}\}$$

such that $\mathrm{N}\texttt{EukCell}^{\mathscr{J}} \neq \emptyset$, where $\mathscr{S} = \{(3.7)\}$. Such a $\mathscr{J}$ exists: take any $\mathscr{D}\mathscr{L}^{\mathrm{N}}$-interpretation where no eukaryotic cells have a nucleus, and $\mathrm{N}\texttt{EukCell}^{\mathscr{J}} = \texttt{EukCell}^{\mathscr{J}} \neq \emptyset$. It follows that (3.8) is not overridden in $\mathrm{N}\texttt{EukCell}/\mathscr{I}$. The reader may easily check in a similar way that (3.8) is not overridden in $\mathrm{N}\texttt{MamRedBldCel}/\mathscr{I}$, either.

Next we show that the low-priority DI (3.6) is not overridden in $\mathrm{N}\texttt{EukCell}/\mathscr{I}$. Take a $\mathscr{D}\mathscr{L}^{\mathrm{N}}$-model $\mathscr{J}$ of (3.7) (i.e. a pre-model of $\mathscr{K}\mathscr{B}$) where:

- $\texttt{EukCell}^{\mathscr{J}} \neq \emptyset$;
- all the instances of EukCell have a nucleus;
- $\texttt{MamRedBldCel}^{\mathscr{J}} = \emptyset$.

The second and first bullet, respectively, ensure that $\mathscr{J}$ satisfies conditions 1 and 2 of Def. 3.1.4 for $\delta = (3.6)$ and $NC = \mathrm{N}\texttt{EukCell}$. The last bullet ensures that the high-priority DI (3.8) is satisfied by all normality concepts; this entails condition 3 of Def. 3.1.4. It follows that (3.6) is not overridden in $\mathrm{N}\texttt{EukCell}/\mathscr{I}$.

Finally, consider $\mathrm{N}\texttt{MamRedBldCel}$ and an arbitrary $\mathscr{D}\mathscr{L}^{\mathrm{N}}$-interpretation $\mathscr{I}$. There are two possibilities:

1. NMamRedBldCel satisfies (3.8) in $\mathscr{I}$. Then it is not possible to find any $\mathscr{J}$ satisfying the conditions of Def. 3.1.4 for $\delta =$ (3.6). To see this, observe that in no pre-model $\mathscr{J}$ of $\mathscr{KB}$, NMamRedBldCel can satisfy both DIs and be nonempty. However, satisfying (3.8) is necessary to satisfy condition 3 of Def. 3.1.4 for $\delta' =$ (3.8). Therefore, (3.6) is overridden in NMamRedBldCel$/\mathscr{I}$.

2. NMamRedBldCel does *not* satisfy (3.8) in $\mathscr{I}$. Then, take a $\mathscr{DL}^{\mathrm{N}}$-interpretation $\mathscr{J}$ of $\mathscr{KB}$ such that:

   - $\mathtt{MamRedBldCel}^{\mathscr{J}} = \mathtt{EukCell}^{\mathscr{J}} \neq \emptyset$;
   - all the instances of MamRedBldCel and EukCell have a nucleus;
   - for all $\mathrm{N}C \neq$ NMamRedBldCel, $\mathrm{N}C^{J} = \emptyset$.

   The second and first bullet, respectively, ensure that $\mathscr{J}$ satisfies conditions 1 and 2 of Def. 3.1.4 for $\delta =$ (3.6) and $\mathrm{N}C =$ NMamRedBldCel. The last bullet ensures that the high-priority DI (3.8) is satisfied by all normality concepts different from NMamRedBldCel. From this fact, and since NMamRedBldCel does not satisfy (3.8) in $\mathscr{I}$, it follows that all the normality concepts that satisfy (3.8) in $\mathscr{I}$, satisfy it also in $\mathscr{J}$, so condition 3 of Def. 3.1.4 is satisfied. We conclude that (3.6) is not overridden in NMamRedBldCel$/\mathscr{I}$.

Summarizing, (3.6) may or may not be overridden in NMamRedBldCel$/\mathscr{I}$ depending on whether NMamRedBldCel satisfies (3.8) in $\mathscr{I}$. ∎

The requirement that non-overridden DIs should be satisfied by normality concepts (while overridden DIs can be ignored) naturally leads to the following notion of DI satisfaction:

Definition 3.1.7 [DI satisfaction] A $\mathscr{DL}^{\mathrm{N}}$ interpretation $\mathscr{I}$ *satisfies* a DI δ (w.r.t. a knowledge base $\mathscr{KB}$) iff for all normality concepts $\mathrm{N}C$, either δ is overridden in $\mathrm{N}C/\mathscr{I}$, or $\mathrm{N}C \in \mathtt{sat}^{\mathscr{I}}(\delta)$. If $\mathscr{I}$ satisfies δ w.r.t. $\mathscr{KB}$ then we write $\mathscr{I} \models_{\mathscr{KB}} \delta$.

Now the notion of $\mathscr{DL}^{\mathrm{N}}$ model can be defined for a full $\mathscr{KB}$ simply by stating that all the members of $\mathscr{KB}$ must be satisfied:

Definition 3.1.8 [$\mathscr{DL}^{\mathrm{N}}$ model] A $\mathscr{DL}^{\mathrm{N}}$ interpretation $\mathscr{I}$ is a $\mathscr{DL}^{\mathrm{N}}$ *model* of $\mathscr{K}\mathscr{B}$ iff $\mathscr{I}$ is a pre-model of $\mathscr{K}\mathscr{B}$ and for all DIs $\delta \in \mathscr{K}\mathscr{B}$, $\mathscr{I} \models_{\mathscr{K}\mathscr{B}} \delta$.

Let ϵ be either a $\mathscr{DL}^{\mathrm{N}}$ sentence (assertion or inclusion) or a DI. If ϵ is satisfied by all the $\mathscr{DL}^{\mathrm{N}}$ models of $\mathscr{K}\mathscr{B}$, then we say that ϵ is a $\mathscr{DL}^{\mathrm{N}}$ *consequence* of $\mathscr{K}\mathscr{B}$ and write

$$\mathscr{K}\mathscr{B} \mid\!\approx \epsilon\,.$$

3.1.2 Examples

We start with some examples where all conflicts (if any) are resolved by specificity. Most nonmonotonic description logics agree on these examples. The first one is a simple representation of the situs inversus example mentioned in the introduction.

Example 3.1.9 Consider again Example 3.1.5 (situs inversus). Recall that for all $\mathscr{DL}^{\mathrm{N}}$-interpretations $\mathscr{I}$, the unique DI of $\mathscr{K}\mathscr{B}$, (3.2), is satisfied by N`Human` and overridden in N`SitusInversus`.

As a first consequence, N`Human` must satisfy (3.2) in every $\mathscr{DL}^{\mathrm{N}}$-model of $\mathscr{K}\mathscr{B}$ (by definition), that is, all the instances of N`Human` are either not humans (which is impossible, by definition of $\mathscr{DL}^{\mathrm{N}}$-interpretations) or members of the concept $\exists$`has_heart`.$\exists$`has_position`.`Left`. As a consequence, one can derive that the heart of standard humans is located on the left-hand side of the body:

$$\mathscr{K}\mathscr{B} \mid\!\approx \mathrm{N}\texttt{Human} \sqsubseteq \exists\texttt{has\_heart}.\exists\texttt{has\_position}.\texttt{Left}\,. \tag{3.9}$$

Moreover, since the $\mathscr{DL}^{\mathrm{N}}$ models of $\mathscr{K}\mathscr{B}$ are also classical models of its strong axioms, by (3.4) we have that the instances of `SitusInversus` have their heart on the opposite side:

$$\mathscr{K}\mathscr{B} \mid\!\approx \texttt{SitusInversus} \sqsubseteq \exists\texttt{has\_heart}.\exists\texttt{has\_position}.\texttt{Right}\,. \tag{3.10}$$

This yields no inconsistency, since (3.2) is overridden in N`SitusInversus`$/\mathscr{I}$, for all models $\mathscr{I}$. Similarly, `SitusInversus` violates the properties of standard

humans without making the knowledge base inconsistent:

$$\mathscr{K}\mathscr{B} \mid\!\approx \texttt{SitusInversus} \sqsubseteq \neg\exists\texttt{has\_heart}.\exists\texttt{has\_position}.\texttt{Left}\,. \tag{3.11}$$

Moreover, as a classical consequence of the above inferences, one can further conclude that people with situs inversus are not standard humans:

$$\mathscr{K}\mathscr{B} \mid\!\approx \texttt{SitusInversus} \sqsubseteq \neg\texttt{NHuman}\,. \tag{3.12}$$

Since $\mathscr{DL}^{\mathrm{N}}$ cautiously refrains from applying (3.2) to *all* humans, in order to avoid inconsistencies with inferences like (3.11), (3.9) cannot be strengthened:

$$\mathscr{K}\mathscr{B} \not\mid\!\approx \texttt{Human} \sqsubseteq \exists\texttt{has\_heart}.\exists\texttt{has\_position}.\texttt{Left}\,.$$

Indeed, there are exceptions to the above subsumption. Obviously, $\mathscr{K}\mathscr{B}$, N`Human`, and N`SitusInversus` are all consistent: $\mathscr{K}\mathscr{B} \not\mid\!\approx \texttt{NHuman} \sqsubseteq \bot$ and $\mathscr{K}\mathscr{B} \not\mid\!\approx \texttt{NSitusInversus} \sqsubseteq \bot$. ∎

In some nonmonotonic logics, an exceptional concept like `SitusInversus`, that does not satisfy some of the standard properties of a more general concept, like `Human`, inherits *none* of the default properties of `Human`, including those that are consistent with the specific properties of `SitusInversus` (such as having a nose).[7] The next example shows that this limitation, sometimes called *inheritance blocking*, does not affect $\mathscr{DL}^{\mathrm{N}}$'s inheritance.[8]

Example 3.1.10 Extend Example 3.1.9 with the additional DI:

$$\texttt{Human} \sqsubseteq_n \exists\texttt{has\_organ}.\texttt{Nose}\,. \tag{3.13}$$

This DI has maximal priority and can be analyzed analogously to (3.2). It is easy to see that (3.13) is overridden neither in N`Human` nor in N`SitusInversus`,

[7] Such logics are discussed in the related work section.

[8] From now on, in the examples, we do not spell out the details of how inferences are derived (with the exception of Example 3.1.11). Section 3.1.3 contains the reductions to classical logic that support those inferences.

therefore both of the following inferences are valid:

$$\mathscr{K}\mathscr{B} \;\mid\!\approx\; \mathtt{NHuman} \sqsubseteq \exists \mathtt{has\_organ.Nose}\,,$$
$$\mathscr{K}\mathscr{B} \;\mid\!\approx\; \mathtt{NSitusInversus} \sqsubseteq \exists \mathtt{has\_organ.Nose}\,.$$

In other words, the property of having a nose is inherited even if (3.9), (3.11), and (3.12) make `SitusInversus` exceptional w.r.t. `Human`. ∎

Example 3.1.11 Consider Example 3.1.6 (eukarotic cells) and recall that for all $\mathscr{DL}^{\mathrm{N}}$-models $\mathscr{I}$ of $\mathscr{K}\mathscr{B}$, neither (3.6) nor (3.8) are overridden in $\mathtt{NEukCell}/\mathscr{I}$, (3.8) is not overridden in $\mathtt{NMamRedBldCel}/\mathscr{I}$, and (3.6) is overridden in $\mathtt{NMamRedBldCel}/\mathscr{I}$[9]. Then, in all such $\mathscr{I}$, `NEukCell` satisfies both DIs, while `NMamRedBldCel` satisfies only (3.8).

The result is that standard eukaryotic cells have a nucleus while standard mammalian red blood cells do not have a nucleus, as required:

$$\mathscr{K}\mathscr{B} \;\mid\!\approx\; \mathtt{NEukCell} \sqsubseteq \exists \mathtt{has\_nucleus}$$
$$\mathscr{K}\mathscr{B} \;\mid\!\approx\; \mathtt{NMamRedBldCel} \sqsubseteq \neg\exists \mathtt{has\_nucleus}\,.$$

Since `NEukCell` must satisfy both DIs, it follows by classical inferences that

$$\mathscr{K}\mathscr{B} \mid\!\approx \mathtt{NEukCell} \sqsubseteq \neg \mathtt{MamRedBldCel}\,,$$

that is, mammalian red blood cells are abnormal eukaryotic cells. %qed ∎

The next example shows how to use $\mathscr{DL}^{\mathrm{N}}$ to encode access control policies (which is another of the intended applications of nonmonotonic description logics mentioned in the introduction). It is also an example of multiple levels of exception: the requests of blacklisted staff are exceptional staff requests, that in turn are exceptional user requests.[10]

[9] Since $\mathscr{I}$ is a $\mathscr{DL}^{\mathrm{N}}$-*model* of $\mathscr{K}\mathscr{B}$, and (3.8) is not overridden in $\mathtt{NMamRedBldCel}/\mathscr{I}$, `NMamRedBldCel` must satisfy (3.8) in $\mathscr{I}$, and in this case (3.6) is necessarily overridden in $\mathtt{NMamRedBldCel}/\mathscr{I}$, as shown in Example 3.1.6.

[10] Recall that inferences are justified by the translation into classical DLs given in Section 3.1.3.

Example 3.1.12 We are going to axiomatize the following natural language policy:

- In general, users cannot access confidential files.
- Staff can read confidential files.
- Blacklisted users are not granted any access. This directive cannot be overridden.

Note that each of the above directives contradicts (and is supposed to override) its predecessor in some particular case. Authorizations can be reified as objects with attributes *subject* (the access requester), *target* (the file to be accessed), and *privilege* (such as *read* and *write*). Then the above policy can be encoded as follows in $\mathscr{ALC}$:

$$\texttt{Staff} \sqsubseteq \texttt{User} \tag{3.14}$$

$$\texttt{Blklst} \sqsubseteq \texttt{User} \tag{3.15}$$

$$\exists\texttt{subj.User} \sqcap \exists\texttt{target.Confidential} \sqsubseteq_n \neg\exists\texttt{privilege} \tag{3.16}$$

$$\exists\texttt{subj.Staff} \sqcap \exists\texttt{target.Confidential} \sqsubseteq_n \exists\texttt{privilege.Read} \tag{3.17}$$

$$\exists\texttt{subj.Blklst} \sqsubseteq \neg\exists\texttt{privilege} \tag{3.18}$$

By (3.14), specificity yields (3.17) $\prec$ (3.16), that is, (3.17) has higher priority than (3.16). With the usual analysis, it can be seen that (3.16) is not overridden in

$$\mathrm{N}(\exists\texttt{subj.User} \sqcap \exists\texttt{target.Confidential});$$

it is overridden in

$$\mathrm{N}(\exists\texttt{subj.Staff} \sqcap \exists\texttt{target.Confidential}) \quad \text{and} \quad \mathrm{N}\exists\texttt{subj.Blklst}\,.$$

Moreover, (3.17) is not overridden in $\mathrm{N}(\exists\texttt{subj.User} \sqcap \exists\texttt{target.Confidential})$, nor in $\mathrm{N}(\exists\texttt{subj.Staff} \sqcap \exists\texttt{target.Confidential})$; it is overridden only in

$$\mathrm{N}\exists\texttt{subj.Blklst}\,.$$

Thus we get the expected policy behavior:

1. Normally, access requests involving confidential files are rejected, if they come from generic users:
   $\mathscr{K}\mathscr{B} \mid\!\approx \mathrm{N}(\exists \mathtt{subj.User} \sqcap \exists \mathtt{target.Confidential}) \sqsubseteq \neg\exists \mathtt{privilege}$;
2. Normally, read operations on confidential files are permitted if the request comes from staff:
   $\mathscr{K}\mathscr{B} \mid\!\approx \mathrm{N}(\exists \mathtt{subj.Staff} \sqcap \exists \mathtt{target.Confidential}) \sqsubseteq \exists \mathtt{privilege.Read}$;
3. Blacklisted users cannot do anything (3.18), so, in particular:
   $\mathscr{K}\mathscr{B} \mid\!\approx \mathrm{N}\exists \mathtt{subj.Blklst} \sqsubseteq \neg\exists \mathtt{privilege}$.

∎

The next example illustrates an inconsistent prototype, due to an unresolvable conflict between DIs with incomparable priorities. Most nonmonotonic logics tacitly solve this conflict and fail to highlight the inconsistency.

Example 3.1.13 Consider the following variant of Nixon's diamond, expressed in $\mathscr{ALC}^{\mathrm{N}}$:

$$\mathtt{Quaker} \sqsubseteq_n \mathtt{Pacifist}\,, \tag{3.19}$$

$$\mathtt{Republican} \sqsubseteq_n \neg\mathtt{Pacifist}\,, \tag{3.20}$$

$$\mathtt{RepQuaker} \sqsubseteq \mathtt{Republican} \sqcap \mathtt{Quaker}\,. \tag{3.21}$$

The two DIs here are not comparable under specificity, that is: (3.19) $\not\prec$ (3.20) and (3.20) $\not\prec$ (3.19). Then both DIs have maximal priority, and their overriding status shall be analyzed independently, by analogy with the unique DI of the situs inversus example. Both (3.19) and (3.20) can be individually satisfied by $\mathrm{N}\mathtt{RepQuaker}$, without making it inconsistent. Then none of them is overridden in $\mathrm{N}\mathtt{RepQuaker}$. It follows that $\mathrm{N}\mathtt{RepQuaker}$ must satisfy both DIs, consequently

$$\mathscr{K}\mathscr{B} \mid\!\approx \mathrm{N}\mathtt{RepQuaker} \sqsubseteq \bot\,,$$

that is, RepQuaker is associated to an inconsistent prototype. A knowledge engineer can now repair it in several possible ways, for instance:

1. by adding $\texttt{RepQuaker} \sqsubseteq_{\mathrm{n}} \texttt{Pacifist}$, which resolves the conflict in favor of (3.19);

2. by adding $\texttt{RepQuaker} \sqsubseteq_{\mathrm{n}} \neg\texttt{Pacifist}$, which resolves the conflict in favor of (3.20);

3. by changing the axiomatization of behavior so as to permit three alternative attitudes: `Pacifist`, `NonPacifist`, and `Mixed`.

Note that even if the prototype of `RepQuaker` is inconsistent, the knowledge base is consistent, as well as many normality concepts. In particular, we have $\mathscr{KB} \not\models \mathrm{N}\texttt{Quaker} \sqsubseteq \bot$ and $\mathscr{KB} \not\models \mathrm{N}\texttt{Republican} \sqsubseteq \bot$. ∎

Example 3.1.14 Consider again Example 3.1.2. Suppose that the axiomatization of bodies and prostates in the two given ontologies has been done with DIs, so as to accommodate exceptional individuals:

$$\texttt{Body} \quad \sqsubseteq_n \quad (= 1\ \texttt{has\_organ.Prostate})\,, \tag{3.22}$$

$$\texttt{Body} \quad \sqsubseteq_n \quad (= 5\ \texttt{has\_organ.Prostate})\,. \tag{3.23}$$

In the union of the two ontologies the above DIs have the same priority, so they cannot override each other. As a consequence, N`Body` must satisfy both, in all models of $\mathscr{KB}$, therefore $\mathscr{KB} \models\!\!\approx \mathrm{N}\texttt{Body} \sqsubseteq \bot$. This makes the unresolved conflict between the two DIs visible to the knowledge engineer. The inconsistency is confined to N`Body`; $\mathscr{KB}$ is consistent. ∎

3.1.3 A Syntactic Characterization of $\models\!\!\approx$

Automate reasoning in $\mathscr{DL}^{\mathrm{N}}$ is based on a syntactic characterization of $\models\!\!\approx$. More precisely, $\models\!\!\approx$ can be reduced to classical reasoning over a $\mathscr{DL}$ knowledge base where DIs are converted into classical axioms. For this purpose, one need to select among the infinitely many concepts NC a finite set Σ of normality concepts that are relevant to the queries of interest. Such concepts are regarded as additional concept names in the resulting classical knowledge base. In order to encode DIs in classical logic, we assume that $\mathscr{DL}$ supports concept intersection

($\sqcap$) on the left-hand side of inclusions.[11]

Let Σ be a finite set of normality concepts *that comprise at least all the normality concepts explicitly occurring in* $\mathscr{K}\mathscr{B}$ plus any concepts needed for constructing the queries of interest.[12] As a special case, Σ may be exactly the set of normality concepts occurring either in $\mathscr{K}\mathscr{B}$ or in the given query.

Let $\mathscr{D}\mathscr{L}^{\Sigma}$ denote the language obtained by extending $\mathscr{D}\mathscr{L}$ with the normality concepts of Σ (treated like new concept names).

For all DIs δ and all normality concepts $\mathrm{N}C$ the classical translation of δ w.r.t. $\mathrm{N}C$ is defined as follows:

$$\delta^{\mathrm{N}C} = \big(\mathrm{N}C \sqcap \mathsf{pre}(\delta) \sqsubseteq \mathsf{con}(\delta)\big)\,.$$

Note that $\mathscr{I} \models \delta^{\mathrm{N}C}$ holds iff $\mathrm{N}C \in \mathsf{sat}^{\mathscr{I}}(\delta)$.

Next, for all sets of $\mathscr{D}\mathscr{L}$ axioms $\mathscr{S}'$ and all DIs δ, let $\mathscr{S}' \downarrow_{\prec\delta}$ denote the result of removing from $\mathscr{S}'$ all the axioms $\delta_0^{\mathrm{N}C}$ such that δ_0's priority is not higher than δ's:

$$\mathscr{S}' \downarrow_{\prec\delta} = \mathscr{S}' \setminus \{\delta_0^{\mathrm{N}C} \mid \mathrm{N}C \in \Sigma \wedge \delta_0 \not\prec \delta\}\,.$$

Finally, let $\delta_1, \ldots, \delta_z$ be an arbitrary *linearization* of $(\mathscr{D}, \prec)$, which means that $\{\delta_1, \ldots, \delta_z\} = \mathscr{D}$ and for all $i, j = 1, \ldots, z$, if $\delta_i \prec \delta_j$ then $i < j$.

Now the classical knowledge base $\mathscr{K}\mathscr{B}^{\Sigma}$ corresponding to $\mathscr{K}\mathscr{B}$ can be defined with the following inductive construction (where $i = 1, 2, \ldots, z$):

$$\mathscr{S}_0^{\Sigma} = \mathscr{S} \cup \big\{\mathrm{N}C \sqsubseteq C \mid \mathrm{N}C \in \Sigma\big\} \tag{3.24}$$

$$\mathscr{S}_i^{\Sigma} = \mathscr{S}_{i-1}^{\Sigma} \cup \big\{\delta_i^{\mathrm{N}C} \mid \mathrm{N}C \in \Sigma \text{ and } \mathscr{S}_{i-1}^{\Sigma} \downarrow_{\prec\delta_i} \cup \{\delta_i^{\mathrm{N}C}\} \not\models \mathrm{N}C \sqsubseteq \bot\big\} \tag{3.25}$$

$$\mathscr{K}\mathscr{B}^{\Sigma} = \mathscr{S}_z^{\Sigma}\,. \tag{3.26}$$

Note that the first step in constructing $\mathscr{K}\mathscr{B}^{\Sigma}$ extends $\mathscr{S}$ with the axioms $\mathrm{N}C \sqsubseteq C$ implicitly satisfied by $\mathscr{D}\mathscr{L}^{\mathrm{N}}$'s semantics. The construction then proceeds by processing the DIs $\delta_i \in \mathscr{D}$ in decreasing priority order; if adding δ_i to the higher priority $\delta_j \prec \delta_i$ that have been previously selected does not make $\mathrm{N}C$ inconsistent, as stated by (3.25),

[11] The restriction rules out a single notable description logic, namely *DL-lite*$_R$.

[12] Queries can be subsumption queries ($C \sqsubseteq D$), instance checking queries ($C(a)$ or $R(a, b)$), concept consistency queries (that can be formulated through subsumptions like $C \sqsubseteq \bot$), and knowledge base consistency queries (that can be expressed through $\top \sqsubseteq \bot$).

then δ_i^{NC} is included in $\mathscr{KB}^{\Sigma}$, otherwise δ_i^{NC} is discarded (overridden).

It can be proved that the above translation into classical reasoning yields a faithful account of subsumption and assertion checking in $\mathscr{DL}^{\mathrm{N}}$.

Theorem 3.1.15 *Let $\mathscr{KB}$ be a $\mathscr{DL}^{\mathrm{N}}$ knowledge base, α be a subsumption or an assertion in $\mathscr{DL}^{\mathrm{N}}$, and let Σ be any finite set of normality concepts including all* NC *that occur in $\mathscr{KB} \cup \{\alpha\}$. Then*

$$\mathscr{KB} \mid\!\approx \alpha \text{ iff } \mathscr{KB}^{\Sigma} \models \alpha.$$

To better illustrate the reduction to classical reasoning we apply it to the example introduced in Section 3.1.1.

Example 3.1.16 Consider again Example 3.1.9 (situs inversus). In order to infer the standard properties of humans let $\Sigma = \{\mathrm{N}\texttt{Human}\}$. The classical translation $\mathscr{KB}^{\Sigma}$ of $\mathscr{KB}$ consists of (3.3)–(3.5) (i.e. the strong part $\mathscr{S}$ of $\mathscr{KB}$) plus the two inclusions $\mathrm{N}\texttt{Human} \sqsubseteq \texttt{Human}$ and $\delta_1^{\mathrm{N}\texttt{Human}}$, where δ_1 is (3.2), so $\delta_1^{\mathrm{N}\texttt{Human}}$ equals:

$$\mathrm{N}\texttt{Human} \sqcap \texttt{Human} \sqsubseteq \exists \texttt{has\_heart}.\exists \texttt{has\_position}.\texttt{Left}\,. \tag{3.27}$$

From these axioms we get the following inferences:

$$\begin{aligned}
\mathscr{KB}^{\Sigma} &\models \mathrm{N}\texttt{Human} \sqsubseteq \exists \texttt{has_heart}.\exists \texttt{has_position}.\texttt{Left}\,,\\
\mathscr{KB}^{\Sigma} &\models \texttt{SitusInversus} \sqsubseteq \exists \texttt{has_heart}.\exists \texttt{has_position}.\texttt{Right}\,,\\
\mathscr{KB}^{\Sigma} &\models \texttt{SitusInversus} \sqsubseteq \neg\exists \texttt{has_heart}.\exists \texttt{has_position}.\texttt{Left}\,,\\
\mathscr{KB}^{\Sigma} &\models \texttt{SitusInversus} \sqsubseteq \neg\mathrm{N}\texttt{Human}\,,
\end{aligned}$$

that entail the corresponding nonmonotonic inferences (3.9), (3.10), (3.11), (3.12), by Theorem 3.1.15. ∎

Overriding in $\mathscr{DL}^{\mathrm{N}}$ can then be checked by means of $\mathscr{KB}^{\Sigma}$ and classical reasoning:

Lemma 3.1.17 *Let $\mathscr{I}$ be a $\mathscr{DL}^{\mathrm{N}}$ interpretation that satisfies $\mathscr{KB}^{\Sigma}$. For all normality concepts* N$C \in \Sigma$*, and for all DIs δ (not necessarily occurring in $\mathscr{KB}$),*

$$\delta \text{ is overridden in } \mathrm{N}C/\mathscr{I} \text{ iff } \mathscr{KB}^{\Sigma} \downarrow_{\prec\delta} \cup \{\delta^{NC}\} \models \mathrm{N}C \sqsubseteq \bot\,.$$

Interestingly, the above characterization of overriding does not depend on $\mathscr{I}$. For this reason, from now on, we drop the model and say simply "δ *is overridden in* NC", meaning that δ is overridden in N$C/\mathscr{I}$ for all $\mathscr{DL}^{\mathrm{N}}$ models $\mathscr{I}$ of $\mathscr{KB}$.

A characterization of DI inference in terms of classical reasoning can also be provided:

Theorem 3.1.18 *For all DIs δ, $\mathcal{K}\mathcal{B} \mid\!\approx \delta$ iff for all normality concepts* NC*,*

$$\text{either } \mathcal{K}\mathcal{B}^{\Sigma}\downarrow_{\prec\delta} \cup\{\delta^{\mathrm{N}C}\} \models \mathrm{N}C \sqsubseteq \bot \text{ or } \mathcal{K}\mathcal{B}^{\Sigma} \models \delta^{\mathrm{N}C},$$

where Σ contains NC *and all the normality concepts* ND *occurring in $\mathcal{K}\mathcal{B}$.*

Unfortunately, Theorem 3.1.18 cannot be immediately applied in practice to infer DIs because it requires the inspection of all the infinitely many, possible normality concepts[13]. A decision method for N-free[14] knowledge bases and DI queries is already coNP-hard for very simple $\mathcal{DL}$ such as the $\exists$-free fragment of $(\mathcal{EL}^{\bot})^{\mathrm{N}}$, that is also a fragment of *DL-lite*$^{\mathrm{N}}_{horn}$. Finding a decision method for unrestricted knowledge bases and DIs is an open problem. This however is not an issue as a closer look reveals that all of our motivating scenarios are covered by Theorem 3.1.15. In particular the complexity of nonmonotonic inference for α that range over subsumptions and assertions in a $\mathcal{EL}$ $++^{\mathrm{N}}$ knowledge base remains PTIME:

Assumption: in the following results, either $\prec$ is determined by specificity, as formalized by (3.1), or the input comprises an extensional description of $\prec$; in this case, checking whether $\delta_i \prec \delta_j$ has linear cost.

Theorem 3.1.19 *Let $\mathcal{K}\mathcal{B}$ range over $\mathcal{EL}$ $++^{\mathrm{N}}$ knowledge bases, and α range over $\mathcal{EL}$ $++^{\mathrm{N}}$ subsumptions and assertions[15]. Then checking $\mathcal{K}\mathcal{B} \mid\!\approx \alpha$ is in PTIME.*

Similar results hold for all $\mathcal{DL}$ whose subsumption problem is tractable, provided that $\sqcap$ can occur in the left-hand side of inclusions (so as to enable the transformation of $\mathcal{K}\mathcal{B}$ into $\mathcal{K}\mathcal{B}^{\Sigma}$). Another logic with this property is DL-lite$_{Horn}$ [Artale et al., 2009]. More tractable cases can be found in [Artale et al., 2009, Table 2].
The complexity of more expressive description logics, whose inference problems are ExpTime-complete[16] is also preserved.

[13] A different normality concept correspond to each of the infinitly many complex concepts that can be contructed in the reference $\mathcal{DL}$.

[14] A knowledge bases is *N-free* if it contains no occurrences of any normality concepts.

[15] Concept satisfiability will not be explicitly dealt with, because it can be naturally reduced to the complement of subsumption checking in the usual way: C is satisfiable w.r.t. $\mathcal{K}\mathcal{B}$ (i.e. there exists a $\mathcal{DL}^{\mathrm{N}}$ model $\mathcal{I}$ of $\mathcal{K}\mathcal{B}$ such that $C^{\mathcal{I}} \neq \emptyset$) iff $\mathcal{K}\mathcal{B} \not\mid\!\approx C \sqsubseteq \bot$. Similarly, $\mathcal{K}\mathcal{B}$ consistency will not be dealt with, because it can be reduced to checking whether $\mathcal{K}\mathcal{B} \mid\!\approx \top \sqsubseteq \bot$.

[16] Comprise all the logics lying between $\mathcal{ALC}^{\mathrm{N}}$, on one side, and $\mathcal{SHOQ}^{\mathrm{N}}$, $\mathcal{SHIO}^{\mathrm{N}}$, or $\mathcal{SHIQ}^{\mathrm{N}}$ on the other side.

Theorem 3.1.20 *Let $\mathscr{DL}$ be a description logic whose subsumption problem is in ExpTime, and let α range over subsumptions and assertions in $\mathscr{DL}^{\mathrm{N}}$. Then deciding $\mathscr{KB} \mid\!\approx \alpha$ in $\mathscr{DL}^{\mathrm{N}}$ is in ExpTime, too. Moreover, if $\mathscr{DL}$'s subsumption problem is ExpTime-complete then so is deciding $\mathscr{KB} \mid\!\approx \alpha$.*

As far as the description logic underlying OWL2 is concerned: $\mathscr{SROIQ}$'s inference belongs to a nondeterministic complexity class, so:

Theorem 3.1.21 *Deciding $\mathscr{KB} \mid\!\approx \alpha$ in $\mathscr{SROIQ}^{\mathrm{N}}$ is in $\mathrm{P}^{\mathrm{N2ExpTime}}$ if α ranges over subsumption and assertion queries.*

Next we characterize the complexity of N-free DI inference.

Theorem 3.1.22 [17] *Let $\mathscr{KB}$ range over $(\mathscr{EL}^{\bot})^{\mathrm{N}}$ knowledge bases, and δ range over $(\mathscr{EL}^{\bot})^{\mathrm{N}}$ DIs. Then checking $\mathscr{KB} \mid\!\approx \delta$ is coNP-hard. Similarly for DL-lite$^{\mathrm{N}}_{horn}$ knowledge bases and DIs. The theorem still holds if the input $\mathscr{KB}$ and δ are* N*-free and $\exists$-free.*

For description logics, whose inference problems are ExpTime-complete, the nondeterministic search performed by the polynomial time Turing machine with ExpTime oracle can be turned into an ExpTime deterministic search, that preserves the cost of the oracle's tests; so we immediately obtain:

Theorem 3.1.23 *Let $\mathscr{DL}$ be a description logic whose subsumption checking problem is in ExpTime, and suppose that deciding $\prec$ has the same complexity. Then* N*-free DI inference in $\mathscr{DL}^{\mathrm{N}}$ is in ExpTime, too. Moreover, if $\mathscr{DL}$'s subsumption checking problem is ExpTime complete, then so is* N*-free DI inference in $\mathscr{DL}^{\mathrm{N}}$.*

In particular, if $\prec$ is specificity, then N*-free DI inference is ExpTime-complete for all the logics ranging from $\mathscr{ALC}^{\mathrm{N}}$ to any of $\mathscr{SHOQ}^{\mathrm{N}}$, $\mathscr{SHIO}^{\mathrm{N}}$, or $\mathscr{SHIQ}^{\mathrm{N}}$.*

Finally, an upper complexity bound for $\mathscr{SROIQ}^{\mathrm{N}}$ can be given:

Corollary 3.1.24 N*-free DI inference in the logic $\mathscr{SROIQ}^{\mathrm{N}}$ is in $\mathrm{coNP}^{\mathrm{N2ExpTime}}$, provided that deciding $\prec$ is in $\mathrm{coNP}^{\mathrm{N2ExpTime}}$, too (which holds if $\prec$ is specificity).*

The complete set of complexity results for the $\mathscr{DL}^{\mathrm{N}}$ logics are reported in Table 3.1.

[17] Note that the results of Theorem 3.1.19 and this theorem are non in conflict. The former applies on subsumption and assertion inference, while the latter applies on DI inference.

Table 3.1. Summary of complexity results

$\mathscr{DL}$ complexity	$\mathscr{DL}^{\mathrm{N}}$ complexity: subsumption and assertion checking	knowledge base and concept consistency	N-free DI inference
P	P	P	coNP
ExpTime	ExpTime	ExpTime	ExpTime
N2ExpTime	$\mathrm{P}^{\mathrm{N2ExpTime}}$	$\mathrm{P}^{\mathrm{N2ExpTime}}$	$\mathrm{coNP}^{\mathrm{N2ExpTime}}$

All results hold for specificity and other priority relations in $\mathrm{P}^{\mathscr{C}}$, where $\mathscr{C}$ is the complexity of subsumption in $\mathscr{DL}$

3.1.4 Reasoning about Individuals

Some nonmonotonic description logics adopt two different approaches for reasoning about TBoxes and ABoxes, e.g. [Casini and Straccia, 2013]. On the contrary, $\mathscr{DL}^{\mathrm{N}}$ treat TBox and ABox reasoning in a uniform way. Recall that the default properties of a concept C can be found by proving inclusions like $\mathrm{N}C \sqsubseteq D$, as shown in the examples discussed so far. Similarly, the default properties of an individual a can be found by proving inclusions $\mathrm{N}\{a\} \sqsubseteq D$.

Example 3.1.25 Extend the situs inversus example (Ex. 3.1.9) with an ABox containing

$$\texttt{Human}(\texttt{Ann})\,,$$
$$\texttt{SitusInversus}(\texttt{Bob})\,.$$

Let δ_1 be the unique DI in $\mathscr{KB}$, that is, (3.2). Using (3.25) it is easy to check that $\delta_1^{\mathrm{N}\{\texttt{Ann}\}}$ belongs to $\mathscr{KB}^{\Sigma}$, while $\delta_1^{\mathrm{N}\{\texttt{Bob}\}}$ does not, because the properties of `SitusInversus` are inconsistent with δ_1 (i.e. δ_1 is overridden in the concept $\mathrm{N}\{\texttt{Bob}\}$).

Accordingly, both $\mathrm{N}\{\texttt{Ann}\}$ and $\mathrm{N}\{\texttt{Bob}\}$ are satisfiable w.r.t. $\mathscr{KB}^{\Sigma}$, and the hearts

of Ann and Bob are located where it should be expected:

$$\mathscr{K}\mathscr{B} \;\mid\!\approx\; \mathrm{N}\{\texttt{Ann}\} \sqsubseteq \exists \texttt{has\_heart}.\exists \texttt{has\_position}.\texttt{Left}\,, \tag{3.28}$$

$$\mathscr{K}\mathscr{B} \;\mid\!\approx\; \mathrm{N}\{\texttt{Bob}\} \sqsubseteq \exists \texttt{has\_heart}.\exists \texttt{has\_position}.\texttt{Right}\,. \tag{3.29}$$

∎

It is worth noting that DIs only state what standard individuals look like; DIs do not force any individual to be standard.[18] Accordingly, a concept $\mathrm{N}\{\mathtt{a}\}$ may happen to be empty, when a contingently violates its default profile. An important advantages of this approach is that if the prototype associated to a nominal $\{a\}$ is inconsistent, then only $\mathrm{N}\{a\}$ is inconsistent; the knowledge base remains globally consistent. This behavior make it easier to identify inconsistent prototypes.

Of course, if $\mathrm{N}\{a\}$ is satisfiable w.r.t. $\mathscr{K}\mathscr{B}$ (as it happens with $\mathrm{N}\{\texttt{Ann}\}$ and $\mathrm{N}\{\texttt{Bob}\}$ in the above example), then a could be safely asserted to be normal by adding the assertion $\mathrm{N}\{a\}(a)$ to $\mathscr{K}\mathscr{B}$. Note, however, that the queries $\mathrm{N}\{a\} \sqsubseteq D$ simply constitute an alternative way of inspecting particular models: these queries are consequences of $\mathscr{K}\mathscr{B}$ iff a belongs to D in all the $\mathscr{DL}^{\mathrm{N}}$-models where a satisfies its default properties, that is, $\mathrm{N}\{a\}(a)$ holds. So, an alternative phrasing of the query $\mathrm{N}\{a\} \sqsubseteq D$ is: "if a is normal, then it satisfies D".

If a $\mathscr{DL}$ and its inference engine do not support nominals, then it may be necessary to reduce the above problem to an inference problem that does not involve nominals. In the rest of this subsection, we assume that

1. *$\mathscr{K}\mathscr{B}$ contains no nominals;*
2. *$\{a\}$ is the only nominal contained in Σ, that is, for all $\mathrm{N}\{b\} \in \Sigma$, $b = a$.*

The transformation tr consists of two steps:

Definition 3.1.26 Let $tr_1(\mathscr{K}\mathscr{B}^{\Sigma})$ be the set of axioms obtained from $\mathscr{K}\mathscr{B}^{\Sigma}$ by removing $\mathrm{N}\{a\} \sqsubseteq \{a\}$, and replacing each $\delta_i^{\mathrm{N}\{a\}}$ with

$$\delta_i^{\{a\}} = \{a\} \sqcap \mathsf{pre}(\delta_i) \sqsubseteq \mathsf{con}(\delta_i)\,.$$

Let $tr_2(\mathscr{K}\mathscr{B}^{\Sigma})$ be the set of axioms obtained from $tr_1(\mathscr{K}\mathscr{B}^{\Sigma})$ by adding the assertion $F(a)$, where F is fresh concept name, and replacing each $\delta_i^{\{a\}}$ with $\delta_i^{F} = F \sqcap \mathsf{pre}(\delta_i) \sqsubseteq \mathsf{con}(\delta_i)$.

[18] This cautious behavior prevents undesired CWA effects, such as those reported in Section 3.1.7.

Since the translation is correct reasoning about nominals can be reduced to an instance problem in a $\mathscr{DL}$ theory without nominals.

Theorem 3.1.27 *$\mathscr{KB} \mid\!\approx \mathrm{N}\{a\} \sqsubseteq C$ iff $tr_(\mathscr{KB}^{\Sigma}) \models C(a)$.*

3.1.5 The Logic of $\mathscr{DL}^{\mathrm{N}}$

In the following we illustrate some logical properties of $\mathscr{DL}^{\mathrm{N}}$. Obviously, $\mathscr{DL}^{\mathrm{N}}$ extends classical logic, by definition:

If $\mathscr{S} \models \alpha$ then $\mathscr{S} \cup \mathscr{D} \mid\!\approx \alpha$.

The first two results show that concepts and axioms can be replaced by (classical) equivalents without affecting any inference. The first result holds for all priority relations.

Theorem 3.1.28 *Let $\mathscr{S}$ and $\mathscr{S}'$ be classically equivalent sets of inclusions and assertions. Then, for all sets of DIs $\mathscr{D}$,*

1. *$\mathscr{I}$ is a $\mathscr{DL}^{\mathrm{N}}$ model of $\mathscr{S}' \cup \mathscr{D}$ iff $\mathscr{I}$ is a $\mathscr{DL}^{\mathrm{N}}$ model of $\mathscr{S} \cup \mathscr{D}$;*
2. *for all subsumptions / assertions / DIs ϵ, $\mathscr{S}' \cup \mathscr{D} \mid\!\approx \epsilon$ iff $\mathscr{S} \cup \mathscr{D} \mid\!\approx \epsilon$.*

The next theorem, instead, requires a mild assumption on priority relations. Roughly speaking, $\prec$ should be insensitive to substitutions with logical equivalents:

Definition 3.1.29 A priority relation $\prec$ for $\mathscr{KB}$ is *semantic* iff $\delta_1 \prec \delta_2$ and $C \equiv_{\mathscr{KB}} D$ imply $\delta_1' \prec \delta_2'$, where each δ_i' is obtained from δ_i by replacing some occurrences of C with D.

Note that the specificity-based relation is a semantic priority relation.

Theorem 3.1.30 *Suppose that $C \equiv D$ is a (classically) valid equivalence and let $\overline{\mathscr{KB}}$ be a knowledge base obtained from $\mathscr{KB}$ by replacing some occurrences of C with D. If $\prec$ is semantic, then*

1. *$\mathscr{I}$ is a $\mathscr{DL}^{\mathrm{N}}$ model of $\overline{\mathscr{KB}}$ iff $\mathscr{I}$ is a $\mathscr{DL}^{\mathrm{N}}$ model of $\mathscr{KB}$;*
2. *for all subsumptions / assertions / DIs ϵ, $\overline{\mathscr{KB}} \mid\!\approx \bar{\epsilon}$ iff $\mathscr{KB} \mid\!\approx \epsilon$,*

where $\bar{\epsilon}$ is obtained from ϵ by replacing some occurrences of C with D.

Note that the above two theorems are not redundant. The former allows for more general restructuring of the strong part, while the latter supports replacements within DIs.

Finally, concerning semantics, $\mathscr{DL}^{\mathrm{N}}$ preserves the finite model property:

Theorem 3.1.31 *If $\mathscr{DL}$ enjoys the finite model property, then $\mathscr{DL}^{\mathrm{N}}$ enjoys it, too.*

3.1.6 Some Methodological Guidelines for KR&R in $\mathscr{DL}^{\mathrm{N}}$

Disciplined usage of normality concepts

Although most of the results hold for unrestricted knowledge bases, the intended use of the strong part $\mathscr{S}$ of $\mathscr{KB}$ is specifying classically valid axioms that do not involve normality concepts, because such concepts are supposed to be defined by the DIs in $\mathscr{D}$. By adhering to this discipline, the so called *canonical knowledge base* is obtained:

Definition 3.1.32 A knowledge base $\mathscr{KB} = \mathscr{S} \cup \mathscr{D}$ is *canonical* if $\mathscr{S} \subseteq \mathscr{DL}$ (i.e. normality concepts do not occur in the strong part of $\mathscr{KB}$).

One of the interesting properties of canonical knowledge bases is that classical and nonmonotonic inferences are neatly separated. By carefully formulating queries, it is possible to distinguish valid consequences from nonmonotonic inferences:

Theorem 3.1.33 *If $\mathscr{KB} = \mathscr{S} \cup \mathscr{D}$ is a canonical $\mathscr{DL}^{\mathrm{N}}$ knowledge base, then for all subsumption or assertions $\alpha \in \mathscr{DL}$, $\mathscr{KB} \mid\!\approx \alpha$ iff $\mathscr{S} \models \alpha$.*

Clearly, the thesis of Theorem 3.1.33 is not valid if $\mathscr{KB}$ is not canonical.

A related interesting property of canonical knowledge bases is that they preserve classical consistency, in the following sense:

Theorem 3.1.34 *Let $\mathscr{KB} = \mathscr{S} \cup \mathscr{D}$ be a canonical $\mathscr{DL}^{\mathrm{N}}$ knowledge base. Then $\mathscr{KB}$ is satisfiable iff $\mathscr{S}$ is (classically) satisfiable.*

Normal attributes for standard instances

One may wonder whether the attributes of a standard individual should be normal as well. In general, this is not the case.

Example 3.1.35 Consider the following scenario: People are usually honest, and lawyers' customers are people. However, lawyers' customers cannot be assumed to be honest, by default. They are more evenly distributed, and it would not be appropriate to assume that they are *not* honest, either. Accordingly, from the knowledge base:

$$\begin{array}{lll} \texttt{Lawyer} & \sqsubseteq & \forall \texttt{customer}.\texttt{Person} \\ \texttt{Person} & \sqsubseteq_n & \texttt{Honest} \end{array}$$

$\mathscr{DL}^{\mathrm{N}}$ does *not* derive $\mathrm{N}\texttt{Lawyer} \sqsubseteq \forall \texttt{customer}.\texttt{Honest}$. This happens for two reasons: (i) nothing in the semantics forces roles to range over normal individuals only; (ii) atypical

individuals may exist, since $\mathscr{D}.\mathscr{L}^{\mathrm{N}}$ does not induce any closed-world effects, as discussed later. ∎

However, as shown by the following example, having roles range over standard individuals makes perfect sense in some scenarios. Most of the other nonmonotonic description logics are unable to express such restrictions.

Example 3.1.36 The situs inversus example might be alternatively formulated by describing normal human hearts, and asserting that typically humans have normal human organs:

$$\begin{aligned}
\texttt{HumanOrgan} &\equiv \exists \texttt{has_organ}^{-}.\texttt{Human}\,, \\
\texttt{HumanHeart} &\equiv \texttt{HumanOrgan} \sqcap \texttt{Heart}\,, \\
\texttt{Human} &\sqsubseteq \exists \texttt{has_organ}.\texttt{Heart}\,, \\
\texttt{HumanHeart} &\sqsubseteq_n \exists \texttt{has_position}.\texttt{Left}\,, && (3.30)\\
\texttt{Human} &\sqsubseteq_n \forall \texttt{has_organ}.\mathrm{N}\,\texttt{HumanOrgan}\,. && (3.31)
\end{aligned}$$

The first DI (3.30) is not overridden in $\mathrm{N}\,\texttt{HumanOrgan}$, so the second DI (3.31) makes the organs of typical humans satisfy $\neg\texttt{HumanHeart} \sqcup \exists\texttt{has\_position}.\texttt{Left}$. Consequently, the above knowledge base yields the expected inference that the heart of typical humans is placed on the left-hand side of the body:

$$\mathscr{K}\mathscr{B} \mid\!\approx \mathrm{N}\texttt{Human} \sqsubseteq \exists\texttt{has\_organ}.(\texttt{Heart} \sqcap \exists\texttt{has\_position}.\texttt{Left})\,.$$

∎

More precisely, a nonmonotonic design pattern exists that given a knowledge base $\mathscr{K}\mathscr{B} = \mathscr{S} \cup \mathscr{D}$ with (possibly defeasible) N-free $\forall$-restrictions $C_i \sqsubseteq_{[n]} \forall R_i.D_i$ $(1 \leq i \leq n)$, further restricts each role R_i to the standard members of D_i preserving the DIs in $\mathscr{D}$, that is, making role values standard only if the DIs in $\mathscr{D}$ permit to do so. This can be accomplished by introducing in $\mathscr{K}\mathscr{B}$ the new DIs $\delta_i = C_i \sqsubseteq_n \forall R_i.\mathrm{N}D_i$ $(1 \leq i \leq n)$ and setting their priorities as follows:

- the new DIs are compared with each other by means of specificity, as in (3.1);
- for each new DI δ_i and all $\delta \in \mathscr{D}$, let $\delta \prec \delta_i$ (i.e. the new DIs have lower priority than all of the explicit DIs in $\mathscr{K}\mathscr{B}$).

In this way, if $\mathscr{K}\mathscr{B} = \{A \sqsubseteq \forall R.B,\ B \sqsubseteq_n C\}$, then the new DI $\delta_1 = A \sqsubseteq_n \forall R.\mathrm{N}B$ introduced by the design pattern makes it possible to entail $\mathscr{K}\mathscr{B} \mid\!\approx \mathrm{N}A \sqsubseteq \forall R.C$.

Clearly, if the new DIs were not given lower priority than the explicit DIs, then in examples like this an unresolvable conflict would arise between the DIs of $\mathscr{KB}$ and those introduced by the design pattern; consequently NA would become inconsistent.

3.1.7 Comparison with Other Nonmonotonic DLs

In this section we assume the reader to be familiar with the nonmonotonic description logics compared with $\mathscr{DL}^{\mathrm{N}}$. Their syntax and semantics can be found in [Baader and Hollunder, 1995a, Baader and Hollunder, 1995b, Casini and Straccia, 2010, Bonatti et al., 2011b, Donini et al., 2002, Giordano et al., 2013b].

Circumscribed Description Logics

In this section we compare $\mathscr{DL}^{\mathrm{N}}$ with the circumscribed DLs dealt with in [Bonatti et al., 2011b], that can be regarded as a fragment of the more general framework analyzed in [Bonatti et al., 2009b]. The syntax adopted in [Bonatti et al., 2011b] supports defeasible inclusions; however, there are no normality predicates and each DI $C \sqsubseteq_n D$ affects directly the extension of C.

An important semantic difference between $\mathscr{DL}^{\mathrm{N}}$ and circumscribed $\mathscr{DL}$ is that circumscription—roughly speaking—cannot create individuals. This happens because the model preference relation at the core of Circumscription's semantics makes two models $\mathscr{I}$ and $\mathscr{J}$ comparable only if they have the same domain. This property has subtle consequences that may be difficult to predict.

Example 3.1.37 Consider the following domain description:

1. All dentists have an assistant;
2. Normally, a dentist's assistant is not a dentist;
3. Ann is a dentist.

The natural encoding of the above sentences is:

$$\begin{aligned} &\texttt{Dentist} \quad \sqsubseteq \quad \exists \texttt{has\_assistant}\,, \\ &\texttt{Dentist} \quad \sqsubseteq_n \quad \forall \texttt{has\_assistant}.\neg\texttt{Dentist}\,, \\ &\texttt{Dentist(Ann)}\,. \end{aligned} \tag{3.32}$$

From the above inclusions, one would expect the conclusion

$$\texttt{Dentist} \sqsubseteq \exists \texttt{has\_assistant}.\neg\texttt{Dentist}. \tag{3.33}$$

Indeed, in $\mathscr{ALC}^{\mathrm{N}}$, the consequence relation $\mathrel{|\!\approx}$ infers the corresponding subsumption

$$\texttt{NDentist} \sqsubseteq \exists \texttt{has\_assistant}.\neg\texttt{Dentist} \tag{3.34}$$

On the contrary, (3.33) is not entailed under Circumscription. The reason is a peculiar counterexample $\mathscr{I}$ whose domain consists of a single individual d. The strong axiom forces Ann to be the assistant of herself; then (3.33) is not satisfied by $\mathscr{I}$. Since $\mathscr{I}$ is comparable only with other models with the same domain, Circumscription cannot "improve" $\mathscr{I}$ by introducing a new assistant that is not a dentist. The reason why $\mathscr{DL}^{\mathrm{N}}$ does not suffer from this problem is that overriding is checked against models $\mathscr{J}$ that may have a different domain. Since (3.32) can be satisfied by extending the domain, the singleton interpretation $\mathscr{I}$ is discarded by $\mathscr{DL}^{\mathrm{N}}$ and (3.34) can be inferred. In order to solve this well-known problem of Circumscription, in [Bonatti et al., 2011a] has been suggested to introduce additional axioms to make the targets of existential restrictions nonempty. In the dentist example, the additional axiom could be $\top \sqsubseteq \exists \texttt{aux}.\neg\texttt{Dentist}$ (where `aux` is a fresh role name). ∎

In order to adopt Circumscription, knowledge engineers must decide for each predicate whether it should be fixed or variable. Fixed predicates cannot be affected by Circumscription, while variable predicates can. It has been proved that roles should not be fixed, otherwise becomes undecidable [Bonatti et al., 2009b, Bonatti et al., 2011b]. As far as concepts are concerned, it has been observed that a closed-world assumption (CWA) affect all variable concepts with some exceptional property. A variable concept C may thus become empty or restricted to the only named individuals in the knowledge base that belong to C [Bonatti et al., 2009b, Bonatti et al., 2010, Bonatti et al., 2011a]. Similar CWA effects are usually undesirable in DLs.

Example 3.1.38 Consider again the situs inversus example. This is a formulation in Circumscribed $\mathscr{EL}^{\bot}$:

$$\texttt{Human} \sqsubseteq_n \exists \texttt{has\_heart}.\exists \texttt{position}.\texttt{Left}$$

$$\texttt{SitusInversus} \sqsubseteq \texttt{Human} \sqcap \exists \texttt{has\_heart}.\exists \texttt{position}.\texttt{Right}$$

$$\exists \texttt{has\_heart}.\exists \texttt{position}.\texttt{Left} \sqcap \exists \texttt{has\_heart}.\exists \texttt{position}.\texttt{Right} \sqsubseteq \bot\,.$$

$$\texttt{SitusInversus}(\texttt{Bob})\,.$$

If `SitusInversus` is variable, then Circumscription maximizes the set of individuals satisfying $\texttt{Human} \sqsubseteq_n \exists \texttt{has\_heart}.\exists \texttt{position}.\texttt{Left}$ by restricting $\texttt{SitusInversus}^{\mathscr{I}}$ to

$\{\mathtt{Bob}^{\mathscr{I}}\}$. The effects of this minimization become visible if we add properties to Bob. For example, extend the ABox with $\mathtt{Blond}(\mathtt{Bob})$. Then Circumscription yields

$$\mathsf{Circ}_F(\mathscr{K}\mathscr{B}) \models \mathtt{SitusInversus} \sqsubseteq \mathtt{Blond}\,.$$

An obvious solution consists in making all concept names fixed. However, in this case, some care must be taken, otherwise it might be impossible to define default attributes at all, as shown in the following example.

Example 3.1.39 Consider a simplified policy example, where $\mathscr{K}\mathscr{B}$ consists of the single DI

$$\mathtt{UserRequest} \sqsubseteq_{\mathrm{n}} \exists \mathtt{decision}.\mathtt{Deny}\,.$$

If $\mathtt{Deny}$ is fixed, then Circumscription cannot infer $\mathtt{UserRequest} \sqsubseteq \exists \mathtt{decision}.\mathtt{Deny}$ because there exists a model $\mathscr{I}$ of the circumscription of $\mathscr{K}\mathscr{B}$ in which $\mathtt{Deny}$ is empty. Since $\mathtt{Deny}$ is fixed, $\mathscr{I}$ can only be compared with other models where $\mathtt{Deny}$ is empty. It follows that Circumscription cannot force $\exists \mathtt{decision}.\mathtt{Deny}$ to be satisfied. This problem can be addressed as shown in Example 3.1.37, by introducing additional axioms that make $\mathtt{Deny}$ nonempty. Several of the examples in Section 3.1.2 show that $\mathscr{D}\mathscr{L}^{\mathrm{N}}$ does not exhibit any similar difficulties. Again, the reason is that overriding is defined in terms of models that may have different domains. ∎

In principle, Circumscription applies default properties to all individuals, including the "implicit" that are not denoted by any constant name, and exist because of existential quantification. However, this does not imply that in a concept $\exists R.B$ role R ranges over the normal instances of B; the behavior of Circumscription, in this respect, is essentially context dependent and uncontrollable.

Example 3.1.40 Assume that all concept names are variable. Consider the $\mathscr{K}\mathscr{B}$

$$\begin{aligned} A &\sqsubseteq \exists R.B \\ B &\sqsubseteq_n C\,, \\ B' &\sqsubseteq B \\ B' &\sqsubseteq_n \neg C\,, \end{aligned}$$

and the query $Q = A \sqsubseteq \exists R.C$, which might be expected to hold because of the first two inclusions. Circumscription maximizes the set of individuals satisfying both DIs by making B' empty and having all instances of B satisfy C; consequently, Q is entailed

by $\mathscr{K}\mathscr{B}$. However, if $\mathscr{K}\mathscr{B}$ is extended with any axiom that forces B' to be nonempty (such as $\top \sqsubseteq \exists S.B'$ or $B'(a)$), then in every model of $\mathscr{K}\mathscr{B}$, B contains some individual that satisfies $\neg C$. Semantics lets R range freely over *all* the members of B, so Q does not hold anymore. A similar behavior emerges when concept names are fixed. ∎

So, with Circumscription, knowledge engineers have neither a direct way of restricting the range of R to the normal instances of B, nor any way of preventing such restriction when it is not appropriate. For this reason, Circumscription is not guaranteed to handle correctly Example 3.1.35 and Example 3.1.36 (the result is context dependent).

Concerning conflict resolution, Circumscription deals with inconsistent prototypes by fixing them in all possible ways.

Example 3.1.41 Consider again Example 3.1.1 (Nixon's diamond). Using circumscribed $\mathscr{E}\mathscr{L}^{\bot}$, it can be formalized as follows:

$$\texttt{Quarker} \sqsubseteq_n \exists \texttt{has\_behavior.Pacifist}\,,$$

$$\texttt{Republican} \sqsubseteq_n \exists \texttt{has\_behavior.NonPacifist}\,,$$

$$\exists \texttt{has\_behavior.Pacifist} \sqcap \exists \texttt{has\_behavior.NonPacifist} \sqsubseteq \bot\,,$$

$$\texttt{Nixon} \sqsubseteq \texttt{Quaker} \sqcap \texttt{Republican}\,.$$

Here `Nixon` is associated to an inconsistent prototype by multiple inheritance. Each model of circumscribed $\mathscr{K}\mathscr{B}$ corresponds to an optimal repair of the prototype where `Nixon` satisfies exactly one of the above DIs. Therefore, Circumscription entails

$$\texttt{Nixon} \sqsubseteq \exists \texttt{has\_behavior.Pacifist} \sqcup \exists \texttt{has\_behavior.NonPacifist}\,,$$

and makes concept `Nixon` consistent. In $\mathscr{D}\mathscr{L}^{\mathrm{N}}$, instead, since the two DIs have the same priority, the conflict is not resolved and $\mathscr{D}\mathscr{L}^{\mathrm{N}}$ makes it evident as follows:

$$\mathscr{K}\mathscr{B} \mid\!\approx \mathrm{N}\,\texttt{Nixon} \sqsubseteq \bot\,.$$

∎

We argued that the $\mathscr{D}\mathscr{L}^{\mathrm{N}}$ conflict handling method is safer. It has also better computational properties, as shown by Theorem 3.1.19 and the complexity analysis in [Bonatti et al., 2011b]. In particular, $\mathscr{E}\mathscr{L}$ $++^{\mathrm{N}}$ is tractable, while circumscribed $\mathscr{E}\mathscr{L}^{\bot}$ needs further restrictions of various sort to reduce its complexity [Bonatti et al., 2011b]. A prototypical implementation for $\mathscr{A}\mathscr{L}\mathscr{C}\mathscr{O}$ can be found in [Grimm and Hitzler, 2009].

Default Description Logics

Some of the earliest nonmonotonic DLs are based on Default logic and prioritized extensions thereof [Baader and Hollunder, 1995a, Baader and Hollunder, 1995b]. Undecidability issues led the authors of these papers to restrict the application of default rules only to the individuals that are explicitly named in the knowledge base. This means that the implicit individuals introduced by existential quantification are not subject to nonmonotonic axioms. As a consequence, Default DLs cannot encode Example 3.1.36, because the range of role *has_organ* consists of implicit individuals to which default rules do not apply.

In case of conflicts between different default rules, Default logic adopts the repair-like approach sketched in the introduction.

Example 3.1.42 In Default logic, Nixon's diamond can be encoded with the following default rules and axioms:

$$\frac{\texttt{Quaker}(X) : \mathrm{M}(\exists \texttt{has\_behavior.Pacifist})(X)}{(\exists \texttt{has\_behavior.Pacifist})(X)}$$

$$\frac{\texttt{Republican}(X) : \mathrm{M}(\exists \texttt{has\_behavior.NonPacifist})(X)}{(\exists \texttt{has\_behavior.NonPacifist})(X)}$$

$$\exists \texttt{has\_behavior.Pacifist} \sqcap \exists \texttt{has\_behavior.NonPacifist} \sqsubseteq \bot\,,$$

$$(\texttt{Quaker} \sqcap \texttt{Republican})(\texttt{Nixon})\,.$$

There exist two default extensions, containing $(\exists \texttt{has\_behavior.Pacifist})(\texttt{Nixon})$ and $(\exists \texttt{has\_behavior.NonPacifist})(\texttt{Nixon})$, respectively. ∎

We have already discussed the potential drawbacks of this approach, in terms of representation error repairs. Moreover—due to this behavior—it is not hard to see that even if the underlying description logic is tractable (e.g. $\mathscr{EL}^{\bot}$), the complexity of credulous reasoning is NP-hard, and the complexity of skeptical reasoning is coNP-hard while $\mathscr{DL}^{\mathrm{N}}$ preserves tractability.

Autoepistemic Description Logics

Autoepistemic description logics [Donini et al., 2002] have many properties in common with the description logics based on Default logic. Hence, we provide only a brief summary of their features highlighting the relationships between the two approaches.

The autoepistemic approach is based on a version of MKNF (the logic of minimal knowledge and negation as failure) where all interpretations have the same domain: a denumerable set of constants called *standard names*. The examples of Section 3.1.7 can be transformed into analogous examples for MKNF by replacing each default rule

$$\frac{\alpha : \mathrm{M}\beta}{\beta}$$

with the MKNF inclusion $\mathrm{K}\alpha \sqcap \neg\mathrm{A}\neg\beta \sqsubseteq \mathrm{K}\beta$, where K is an epistemic operator and $\neg$A is essentially negation as failure.

The above nonmonotonic inclusions define the defeasible properties β of individuals. Apparently, since the above inclusions need not be grounded on syntactic domains, they apply to all individuals (including those that are not denoted by any constant occurring in the knowledge base). However, the precondition $\mathrm{K}\alpha$ is very strong. Only individuals that are denoted by a constant occurring in the knowledge base can satisfy it, so in practice nonmonotonic rules apply only to such individuals, as it happens with Default logic. It follows that Autoepistemic description logics cannot deal correctly with Example 3.1.36 as well.

Conflict handling is based on repairs and may yield multiple models that correspond to the alternative default extensions of default theories. It is not hard to see that the computational complexity of subsumption is at least coNP-hard, even if the underlying DL is tractable.

To the best of our knowledge, no prioritized version of Autoepistemic description logics has been introduced so far.

Conditional Entailment

$\mathscr{DL}^{\mathrm{N}}$ exhibits a superficial syntactic similarity with conditional knowledge bases and their conditional entailment semantics [Geffner and Pearl, 1992]. Conditional knowledge bases are sets of classical sentences and defaults $\phi \rightarrow \psi$; these expressions are syntactic analogues of strong inclusions and DIs, respectively. Given a priority relation $\prec$ over the set of defaults, the models of a conditional knowledge base are those that maximize the set of satisfied defaults; such models are called $\prec$-preferred models. A sentence is *conditionally entailed* by a knowledge base $\mathscr{KB}$ if it is satisfied by all the $\prec$-preferred models of $\mathscr{KB}$, for all *admissible* preference orderings. A preference ordering is admissible iff each set of defaults in $\mathscr{KB}$ that is in conflict with another default $d \in \mathscr{KB}$, contains a default $d' \prec d$.

Conditional entailment has never been extended to description logics. For a fixed

relation $\prec$, the models of circumscribed description logic with variable predicates—as presented in [Bonatti et al., 2011b]—would be close analogues of $\prec$-preferred models; however, it is not clear how to extend the propositional notion of admissible ordering to this setting. This makes it more difficult to compare $\mathscr{DL}^{\mathrm{N}}$ and conditional entailment. In particular, it is not possible to assess how a description logic based on conditional entailment would handle Examples 3.1.35 and 3.1.36 (and, more generally, how much control would be possible on the default properties of role ranges). The analogies with circumscribed DLs raise the concern that a conditional description logic might inherit Circumscription's limitations, in this respect.

The proof theory of conditional entailment is *argumentation*, which is significantly more complex than the simple iterative reduction to classical logic used for $\mathscr{DL}^{\mathrm{N}}$ subsumptions and assertions. In the propositional case, the complexity of conditional entailment is complete for the second level of the polynomial hierarchy, even if the underlying monotonic logic is tractable [Eiter and Lukasiewicz, 2000].

The notion of admissible ordering, and conditional entailment's quantification over such orderings, have two effects: On the one hand, no explicit priority needs to be specified. The priority induced by admissible orderings is always grounded on specificity (explicit priorities are more flexible). On the other hand, with this approach, specificity is determined by both strong *and* defeasible axioms. Sometimes, this notion of specificity solves conflicts than cannot be resolved by the version based solely on strong axioms.

Example 3.1.43 Consider two analogous knowledge bases, encoded as a conditional knowledge base and a $\mathscr{DL}^{\mathrm{N}}$ knowledge base, respectively:

$$\begin{array}{lcllcl} a & \rightarrow & b & A & \sqsubseteq_n & B \\ a & \rightarrow & c & A & \sqsubseteq_n & C \\ b & \rightarrow & \neg c & B & \sqsubseteq_n & \neg C\,. \end{array}$$

Using conditional entailment's priorities, the first default makes a more specific than b, thereby giving itself and the second default higher priority than the third one. As a consequence, conditional entailment yields $a \supset c$ and solves the conflict between the second and third defaults in favor of the former. On the contrary, in $\mathscr{DL}^{\mathrm{N}}$, the three DIs are all incomparable, as there are no strong inclusions. So the conflict is not resolved, and $\mathscr{KB} \not\models \mathrm{N}A \sqsubseteq \bot$. ∎

In other cases, our notion of specificity-based priority solves conflicts that conditional

entailment cannot resolve.

Example 3.1.44 Consider the following two analogous knowledge bases, inspired by the Nixon's diamond, encoded as a conditional knowledge base and a $\mathscr{DL}^{\mathrm{N}}$ knowledge base, respectively:

$$\begin{array}{rcl@{\qquad}rcl} q & \rightarrow & p & Q & \sqsubseteq_n & P \\ r & \rightarrow & \neg p & R & \sqsubseteq_n & \neg P \\ \mathit{true} & \rightarrow & q \wedge r & \top & \sqsubseteq_n & Q \sqcap R\,. \end{array}$$

The conditional knowledge base is inconsistent, because no priority ordering is admissible, while the $\mathscr{DL}^{\mathrm{N}}$ knowledge base is consistent, and the last DI is overridden by the other two, because it defines a default property for the less specific concept of all ($\top$). ∎

Note that the explicit priorities adopted in Circumscribed DLs, default DLs and $\mathscr{DL}^{\mathrm{N}}$ are more flexible, as they are not necessarily confined to specificity, and allow the encoding of nonmonotonic design patterns such as the default role range pattern illustrated after Example 3.1.36. Nonetheless, making priorities depend on DIs, and analyzing the consequences on the expressiveness and the complexity of $\mathscr{DL}^{\mathrm{N}}$, are interesting subjects for further research.

Rational Closure, Typicality

A first comparison with the description logics based on rational closure and typicality derives from the analysis of the relationships between DI inference and the *rational closure properties* [Lehmann and Magidor, 1992] — whose adaptation to our syntax is illustrated in Figure 3.1 — that extend the KLM axioms for preferential entailment [Kraus et al., 1990]. In Figure 3.1, each rule with premises δ_1, δ_2 and conclusion δ_3 should be interpreted as follows:

$$\textit{If } \mathscr{K}\mathscr{B} \mid\!\approx \delta_1 \textit{ and } \mathscr{K}\mathscr{B} \mid\!\approx \delta_2\textit{, then } \mathscr{K}\mathscr{B} \mid\!\approx \delta_3. \tag{3.35}$$

Theorem 3.1.45 *Axiom (REF) is valid. Axiom (OR) holds if the priority relation is specificity. Axioms (CT), (CM), (LLE), (RW), and (RM) are not valid.*

Some of them are not valid in $\mathscr{DL}^{\mathrm{N}}$, and hold only in the absence of overriding.

(REF)	$C \sqsubseteq_n C$	Reflexivity
(CT)	$\dfrac{C \sqsubseteq_n D \quad C \sqcap D \sqsubseteq_n E}{C \sqsubseteq_n E}$	Cut (Cumulative Transitivity)
(CM)	$\dfrac{C \sqsubseteq_n D \quad C \sqsubseteq_n E}{C \sqcap D \sqsubseteq_n E}$	Cautious Monotony
(LLE)	$\dfrac{C \sqsubseteq_n E \quad C \equiv D}{D \sqsubseteq_n E}$	Left Logical Equivalence
(RW)	$\dfrac{C \sqsubseteq_n D \quad D \sqsubseteq E}{C \sqsubseteq_n E}$	Right Weakening
(OR)	$\dfrac{C \sqsubseteq_n E \quad D \sqsubseteq_n E}{C \sqcup D \sqsubseteq_n E}$	Left Disjunction
(RM)	$\dfrac{C \sqsubseteq_n E \quad C \not\sqsubseteq_n \neg D}{C \sqcap D \sqsubseteq_n E}$	Rational Monotony

Figure 3.1. Rational closure axioms. Statements $X \sqsubseteq_{(n)} Y$ should be interpreted as $\mathscr{K}\mathscr{B} \mid\!\approx X \sqsubseteq_{(n)} Y$, for a fixed $\mathscr{K}\mathscr{B}$.

A simple inspection of the proof of Theorem 3.1.45 shows that the rational closure properties that are not valid in $\mathscr{DL}^{\mathrm{N}}$ are in sharp contrast with the very idea of specificity-based overriding. Consider (CT), first: a concept A more specific than C may strongly satisfy $\neg D$ so that $C \sqsubseteq_n D$ is overridden in NA. Then, since NA is not contained in D, $C \sqcap D \sqsubseteq_n E$ cannot be applied to infer that the members of NA satisfy E. As a consequence, there is no reason to infer that NA satisfies $C \sqsubseteq_n E$. The following example instantiates this situation in a concrete representation domain, showing that extending $\mathscr{DL}^{\mathrm{N}}$ with (CT) can lead to undesirable inferences.

Example 3.1.46 In several countries (e.g. Mexico, Norway and Brazil) military service is mandatory for male citizens (except for special cases such as mental disorders). After military training, citizens become *reservists*, and shall join the army again in case of war. This can be formalized with the following DIs:

$$\texttt{MaleCitizen} \sqsubseteq_n \texttt{HasMilitaryTraining} \tag{3.36}$$

$$\texttt{MaleCitizen} \sqcap \texttt{HasMilitaryTraining} \sqsubseteq_n \texttt{Reservist}\,. \tag{3.37}$$

The exceptions to the above rules include minors:

$$\texttt{MinorMaleCitizen} \sqsubseteq \texttt{MaleCitizen} \tag{3.38}$$

$$\texttt{MinorMaleCitizen} \sqsubseteq \neg\texttt{HasMilitaryTraining}\,. \tag{3.39}$$

Axiom (3.39) should prevent (3.37) from being applied to minors, that is, it should *not* be possible to conclude that N $\texttt{MinorMaleCitizen} \sqsubseteq \texttt{Reservist}$ (indeed, this is what happens with $\mathscr{DL}^{\mathrm{N}}$).

On the contrary, by applying (CT) to (3.36) and (3.37), one obtains:

$$\texttt{MaleCitizen} \sqsubseteq_n \texttt{Reservist}\,, \tag{3.40}$$

whose right-hand side is consistent with the properties of $\texttt{MinorMaleCitizen}$ formalized by (3.38) and (3.39). Then (3.40) would not be overridden and it would be possible to conclude that minors are normally reservists (N $\texttt{MinorMaleCitizen} \sqsubseteq \texttt{Reservist}$). ∎

Theorem 3.1.45's argument for showing that (CM) does not hold is similar: A has a top priority DI $A \sqsubseteq_n \neg E$ that overrides the premise $C \sqsubseteq_n E$, thereby removing the main reason for inferring $C \sqcap D \sqsubseteq_n E$. Analogously, in the counterexamples for (RW) and (RM), the strong properties of A override the first premise of the inference rule. Finally consider (LLE): in this rule, the problem is that the equivalence $C \equiv D$ is

assumed to be a defeasible inference by our semantics; therefore, in general, it can be invalidated by overriding and, as a consequence, the conclusion of (LLE) is not logically supported.

It is easy to see that overriding is indeed the only reason why some rational closure properties fail. To see this, for all rules listed in Figure 3.1 with premises δ_1, δ_2 and conclusion δ, consider the following *weak interpretation*:

> For all normality concepts $\mathrm{N}X$: if $\mathscr{K}\mathscr{B} \not\approx \delta_1$, $\mathscr{K}\mathscr{B} \not\approx \delta_2$, and neither δ_1 nor δ_2 are overridden in $\mathrm{N}X$, then for all $\mathscr{DL}^{\mathrm{N}}$-models $\mathscr{I}$ of $\mathscr{K}\mathscr{B}$, $\mathrm{N}X$ satisfies the conclusion δ (i.e. $\mathrm{N}X \in \mathsf{sat}^{\mathscr{I}}(\delta)$).

In the next theorem, all of the above axioms hold under the weak (overriding free) interpretation.

Theorem 3.1.47 *All axioms in Figure 3.1 hold under their weak interpretation.*

Recall that the reason why (LLE) is not valid, in general, under the strong interpretation formalized by (3.35), is that the premise $C \equiv D$ is a defeasible consequence. It was meant to be a tautology in the original KLM axioms. Actually, (LLE) is valid under the strong interpretation (3.35) not only if $C \equiv D$ is a tautology, but also when it is a strong consequence of $\mathscr{K}\mathscr{B}$, provided that the priority relation is semantic.

Theorem 3.1.48 *Axiom (LLE) holds when the assumption $C \equiv D$ is interpreted as $\mathscr{K}\mathscr{B} \models C \equiv D$, and $\prec$ is semantic.*

The logic $\mathscr{ALC} + \mathbf{T}_{\min}$ [Giordano et al., 2009b, Giordano et al., 2013a, Giordano et al., 2013b] features a typicality operator $\mathbf{T}$ analogous to N, that gives a comparable degree of flexibility in specifying the default properties of role ranges (with some exceptions, cf. Example 3.1.50 below). The equivalent of our DI $C \sqsubseteq_n D$ is $\mathbf{T}(C) \sqsubseteq D$, i.e. there is no special symbol for defeasible inclusions. The monotonic semantics of the typicality operator is essentially a preferential semantics [Giordano et al., 2009b]; its nonmonotonic extension [Giordano et al., 2013b] is a minimal model semantics that maximizes the extension of $\mathbf{T}$. $\mathscr{ALC} + \mathbf{T}_{\min}$ satisfies the KLM axioms (REF), (LLE), (CM), and (OR) [Giordano et al., 2009b], and hence it formally differs from $\mathscr{DL}^{\mathrm{N}}$ where (CM) is not universally valid (cf. 3.1.45). A stronger semantics satisfying also rational monotony, originally introdiced in [Britz et al., 2008], is discussed in [Giordano et al., 2013b, Sec. 7.2]; it is argued that this semantics is too strong and yields undesirable results. In $\mathscr{DL}^{\mathrm{N}}$, also (CM) yields undesirable results, which show

that (CM) is not compatible with the notion of "overriding as plain inconsistency" adopted in $\mathscr{DL}^{\mathrm{N}}$.

In general, $\mathscr{ALC} + \mathbf{T}_{\min}$ resolves conflicts like Circumscription does: In Example 3.1.13 (reformulated in $\mathscr{ALC} + \mathbf{T}_{\min}$) it would be possible to conclude neither $\mathbf{T}(\texttt{RepQuaker}) \sqsubseteq \texttt{Pacifist}$ nor the alternative inclusion $\mathbf{T}(\texttt{RepQuaker}) \sqsubseteq \neg\texttt{Pacifist}$, and $\mathbf{T}(\texttt{RepQuaker})$ would be satisfiable. Only direct conflicts such as $\mathbf{T}(A) \sqsubseteq C$ and $\mathbf{T}(A) \sqsubseteq \neg C$ would make $\mathbf{T}(A)$ inconsistent (so, in Example 3.1.14, $\mathbf{T}(\texttt{Body}) \sqsubseteq \bot$ holds). Another similarity with Circumscription is the CWA effect on exceptional concepts:

Example 3.1.49 The $\mathscr{ALC} + \mathbf{T}_{\min}$ knowledge base

$$\texttt{Whale} \sqsubseteq \texttt{Mammal}$$
$$\mathbf{T}(\texttt{Mammal}) \sqsubseteq \exists\texttt{habitat}.\texttt{Land}$$
$$\mathbf{T}(\texttt{Whale}) \sqsubseteq \neg\exists\texttt{habitat}.\texttt{Land}$$

entails that there are no whales, that is: $\texttt{Whale} \sqsubseteq \bot$. ∎

Similar effects may force role ranges to be normal.

Example 3.1.50 The above knowledge base entails $\mathbf{T}(\texttt{Mammal}) \equiv \texttt{Mammal}$, and in this case it is not possible to distinguish $\exists R.\mathbf{T}(\texttt{Mammal})$ from $\exists R.\texttt{Mammal}$. In other words, R's ranges necessarily over typical instances. ∎

Similarly to the other preferential semantics, $\mathscr{ALC} + \mathbf{T}_{\min}$ is affected by inheritance blocking. Here is an example.

Example 3.1.51 Consider the following variant of the penguins-and-birds example:

1. Penguins are birds;
2. Birds normally fly;
3. Birds normally have wings;
4. Penguins do not fly.

$\mathscr{ALC} + \mathbf{T}_{\min}$ does not infer that penguins have wings; the property of not flying makes penguins exceptional birds and prevents them from inheriting *any* of the prototypical properties of birds [Giordano et al., 2013b, Sec. 7.3]. In $\mathscr{DL}^{\mathrm{N}}$, overriding is selective,

instead. Let $\mathscr{KB}$ be the natural encoding of the example:

$$\texttt{Penguin} \sqsubseteq \texttt{Bird}$$
$$\texttt{Bird} \sqsubseteq_{\mathrm{n}} \texttt{Flying}$$
$$\texttt{Bird} \sqsubseteq_{\mathrm{n}} \exists \texttt{has\_wing}$$
$$\texttt{Penguin} \sqsubseteq \neg \texttt{Flying}\,.$$

It is not hard to see that $\mathscr{KB} \not\models \mathrm{N}\,\texttt{Penguin} \sqsubseteq \exists \texttt{has\_wing}$, as expected. ∎

Analogously, $\mathscr{ALC} + \mathbf{T}_{\min}$ does not deal correctly with Example 3.1.36.

Similarly to conditional entailment, in $\mathscr{ALC} + \mathbf{T}_{\min}$ priorities are implicit and grounded in specificity. In the analogue of Example 3.1.43, namely,

$$\mathbf{T}(A) \sqsubseteq B \qquad \mathbf{T}(A) \sqsubseteq C \qquad \mathbf{T}(B) \sqsubseteq \neg C$$

$\mathscr{ALC} + \mathbf{T}_{\min}$ is able to resolve the conflict: $\mathbf{T}(A)$ is satisfiable and $\mathbf{T}(A) \sqsubseteq C$ is entailed.

The computational complexity of reasoning in $\mathscr{ALC} + \mathbf{T}_{\min}$ has been analyzed in some selected cases. So far, tractable fragments have not been identified. Some restricted fragments (called *left local*) fall within the second level of the polynomial hierarchy [Giordano et al., 2009a, Giordano et al., 2012].

More nonmonotonic DLs based on rational closure can be found in [Casini et al., 2013a, Casini et al., 2013b, Casini and Straccia, 2010]. Differently from $\mathscr{ALC} + \mathbf{T}_{\min}$, they have a special inclusion operator for DIs but no equivalent of N. They satisfy the Rational closure axioms and are subject to inheritance blocking, like $\mathscr{ALC} + \mathbf{T}_{\min}$ (Example 3.1.51 is not dealt with correctly). Moreover, they are not able to infer any standard property about role values (Example 3.1.36 cannot be encoded). No tractable fragments are known; the available results show that complexity is preserved only for ExpTime-hard logics.

These logics have been refined in [Casini and Straccia, 2013] to remove inheritance blocking. The new approach has a syntactic nature (no model-theoretic semantics) and is articulated in two stages:

- in the first stage, the knowledge base is converted into a boolean inheritance network (introduced in [Casini and Straccia, 2013] itself); a set of nonmonotonic inclusions is derived using the network; in general, some of these inclusions cannot be derived by rational closure due to inheritance blocking;

- in the second stage, rational closure is applied to the knowledge base extended with the defeasible inclusions derived in the first stage.

With this method, the effects of inheritance blocking are limited, as the inheritance networks recovers some of the missing inferences; for instance, in Example 3.1.51, this logic can infer that penguins have wings. The Rational closure axioms are satisfied, because the second stage performs a standard rational closure.

The two-stage logic has limited reasoning abilities on role ranges. For example, from the purely classical knowledge base $\{A \sqsubseteq \exists R.B,\ B \sqsubseteq C\}$, the TBox construction of [Casini and Straccia, 2013] does not yield the classical consequence $A \sqsubseteq \exists R.C$ because strong axioms are internalized. This problem has been fixed in [Britz et al., 2013]. However, the solution does not suffice to infer $A \sqsubseteq \exists R.C$ from $\{A \sqsubseteq \exists R.B,\ B \sqsubseteq_n C\}$. We argued that this inference is not always desirable (Example 3.1.35); the point here is that there is no obvious way of achieving it when it is desired (as in Example 3.1.36). The authors of [Britz et al., 2013] leave this issue as an open problem.

Priorities are implicit and determined by specificity, similarly to conditional entailment. Both logics resolve the conflicts in Examples 3.1.43 and 3.1.44 in the same way and return the same inferences.

When specificity does not settle a conflict, the conflict is repaired, similarly to Default and Autoepistemic logics, Conditional entailment, and $\mathscr{ALC}+\mathbf{T}_{\min}$; in particular, in Examples 3.1.13 and 3.1.14, no inconsistency is reported.

In [Casini and Straccia, 2013, Appendix A], the two-stage logic is tried on a number of examples discussed by Sandewall [Sandewall, 2010]. In several of these examples, strong inclusions are either absent or too scarce to prioritize DIs in $\mathscr{DL}^{\mathrm{N}}$. Consequently, $\mathscr{DL}^{\mathrm{N}}$ treats these examples similarly to Example 3.1.43: all DIs are mutually incomparable, and conflicts cannot be resolved by overriding. In the other examples, $\mathscr{DL}^{\mathrm{N}}$ yields the same results as the two-stage logic. There is one exception, namely Example A.8, which appears to be intrinsically problematic. Sandewall proposes a few alternative consequences, corresponding to different ways of resolving conflicts between incomparable defaults. The two-stage logic derives none of them, while $\mathscr{DL}^{\mathrm{N}}$ detects the unresolvable conflict by entailing a subsumption $\mathrm{N}t \sqsubseteq \bot$ [Bonatti et al., 2015a].

The complexity of the two-stage logic has been determined for the nonmonotonic extension of $\mathscr{ALC}$. Basically, the exponential overhead of nonmonotonic inferences (partially due to a brute force search over all the possible permutations of the constants occurring in the ABox, each of which may yield a different deductive closure) is absorbed by the ExpTime complexity of monotonic $\mathscr{ALC}$ reasoning. So far, no tractable fragment has been identified.

Probabilistic, Nonmonotonic Description Logics

These logics, introduced in [Lukasiewicz, 2008], extend classical DLs with *conditional constraints* $(D \mid C)[l, u]$ whose intended meaning is: "by default, the typical instances of C belong to D with probability $p \in [l, u]$". When $l = u = 1$, such conditional constraints are reminiscent of DIs $C \sqsubseteq_n D$, however their behavior differs in many respects.

First of all, probabilistic description logics exhibit some inferences with a paraconsistent flavor.[19] For instance, the knowledge base

$$\top \sqsubseteq \exists R.A \tag{3.41}$$

$$(\neg A \mid \top)[1, 1] \tag{3.42}$$

has a model, even if (3.41) states that some instance of A exists, and (3.42) apparently states that no individuals belong to A. The reason is that (3.42) applies only to a non-denotable subset of individuals, that can be considered as typical individuals. Next, consider the knowledge base with TBox

$$\top \sqsubseteq \forall R.\{a\} \tag{3.43}$$

and a probabilistic ABox that asserts the following constraint on individual b

$$(\exists R.A \mid \top)[1, 1] \tag{3.44}$$

and the following constraint on individual c

$$(\exists R.\neg A \mid \top)[1, 1]\,. \tag{3.45}$$

This knowledge base is consistent and entails both the constraint $(\exists R.(\{a\} \sqcap A) \mid \top)[1, 1]$ for b, and $(\exists R.(\{a\} \sqcap \neg A) \mid \top)[1, 1]$ for c, respectively. Note, however, that if b satisfied $\exists R.(\{a\} \sqcap A)$ and c satisfied $\exists R.(\{a\} \sqcap \neg A)$ at the same time, then a should satisfy both A and $\neg A$ (a contradiction). Therefore the two inferences about b and c should be regarded as members of different deductive closures.

The "typical" individuals, i.e. those that are subject to conditional constraints, are not allowed to occur in any axiom of the knowledge base. Accordingly, in the above knowledge base, a must necessarily be a "classical" individual that is not subject to any

[19] Such contradictory inferences may be regarded as analogues of credulous inference in the logics with multiple deductive closures, such as Default and Autoepistemic logics.

conditional constraint (while $\mathscr{DL}^{\mathrm{N}}$ permits to use $\{a\}$ in the left-hand side of DIs, for all individuals a). This is an obstacle to applying defaults to role values. More generally, it turns out that probabilistic DLs cannot reason about "normal" attribute values. For instance, the probabilistic TBox

$$(\exists R.A \mid \top)[1,1] \tag{3.46}$$

$$(B \mid A)[1,1] \tag{3.47}$$

does not entail $(\exists R.B \mid \top)[1,1]$, that is, the default property (3.47) does not apply to the values of role R in (3.46). Since normal individuals cannot be denoted in this logic, there seems to be no way of restricting role values to typical individuals.

Probabilistic DLs induce CWA effects on exceptional classes. For example the knowledge base

$$\texttt{Whale} \sqsubseteq \texttt{Mammal} \tag{3.48}$$

$$(\exists \texttt{habitat.Land} \mid \texttt{Mammal})[1,1] \tag{3.49}$$

$$(\neg\exists \texttt{habitat.Land} \mid \texttt{Whale})[1,1] \tag{3.50}$$

entails $(\neg\texttt{Whale} \mid \top)[1,1]$, that is, there are no whales.

The conflicts that cannot be resolved by specificity result in either inconsistent concepts or inconsistent knowledge bases. As a first example, consider the following encoding of Nixon's diamond:

$$(\texttt{Pacifist} \mid \texttt{Quaker})[1,1] \tag{3.51}$$

$$(\neg\texttt{Pacifist} \mid \texttt{Republican})[1,1]\,. \tag{3.52}$$

This probabilistic TBox entails that no one is both a quaker and a republican: $(\neg(\texttt{Quaker} \sqcap \texttt{Republican}) \mid \top)[1,1]$. If $\texttt{Quaker} \sqcap \texttt{Republican}$ were forced to be nonempty, e.g. by adding

$$(\texttt{Quaker} \sqcap \texttt{Republican} \mid \top)[0.1,1]\,, \tag{3.53}$$

then the entire knowledge base would be inconsistent.

Priorities over conditional constraints are automatically derived based on a notion of specificity, similar to those adopted by conditional entailment and rational closure. They all behave in the same way on Examples 3.1.43 and 3.1.44.

Probabilistic DLs do not preserve tractability [Lukasiewicz, 2008, Thm. 6.4(c)].

On the positive side, probabilistic DLs do not suffer from inheritance blocking (cf. [Lukasiewicz, 2008, Example 4.17]), and satisfy suitable probabilistic variants of the KLM axioms [Lukasiewicz, 2008, Thm. 4.19, 4.20].

Design Patterns

Let us apply the ontology design pattern (ODP) for exceptions to the Eukaryotic Cell example, as illustrated in [Stevens et al., 2007, Fig. 8, 9]. Axiomatization details follow the approach illustrated in [Rector, 2004, Sec. 2.1.3, 2.2]. The resulting knowledge base, denoted by $\mathscr{KB}$, consists of the following inclusions:

$$\texttt{RedBldCel} \sqsubseteq \texttt{EukCell} \tag{3.54}$$

$$\texttt{MamRedBldCel} \sqsubseteq \texttt{RedBldCel} \tag{3.55}$$

$$\texttt{AvianRedBldCel} \sqsubseteq \texttt{RedBldCel} \tag{3.56}$$

$$\texttt{MamRedBldCel} \sqsubseteq \neg\exists\,\texttt{nucleus} \tag{3.57}$$

$$\texttt{AvianRedBldCel} \sqsubseteq \exists\,\texttt{status.Normal} \tag{3.58}$$

$$\texttt{TypicalEukCell} \equiv \texttt{EukCell} \sqcap \exists\,\texttt{status.Normal} \tag{3.59}$$

$$\texttt{TypicalEukCell} \sqsubseteq \exists\,\texttt{nucleus} \tag{3.60}$$

$$\texttt{ATypicalEukCell} \equiv \texttt{EukCell} \sqcap \neg\texttt{TypicalEukCell} \tag{3.61}$$

$$\texttt{TypicalRBC} \equiv \texttt{RedBldCel} \sqcap \exists\,\texttt{status.Normal} \tag{3.62}$$

$$\texttt{ATypicalRBC} \equiv \texttt{RedBldCel} \sqcap \neg\texttt{TypicalRBC}\,. \tag{3.63}$$

Basically, the design pattern includes a complete axiomatization of all the primitive biological concepts (3.54–3.58) plus a definition of the typical and atypical cases for these concepts (3.59–3.63). Eukaryotic cells are partitioned into typical and atypical instances; red blood cells span over both partitions, so they must be split into typical and atypical instances as well.

The inference engine classifies avian red blood cells as typical red blood cells and typical eukaryotic cells; it classifies mammalian red blood cells as atypical red blood cells and atypical eukaryotic cells. Note that this is possible only if every concept is explicitly associated to a property that determines its normality or abnormality, as in (3.57–3.58) and (3.62). In other words, typical red blood cells do not *automatically* inherit the properties of typical eukaryotic cells, and avian red blood cells do not automatically inherit the properties of typical red blood cells. As a further example, if the single

additional axiom

$$\texttt{ReptileRedBldCel} \sqsubseteq \texttt{RedBldCel} \tag{3.64}$$

were added to $\mathscr{K}\mathscr{B}$, then (without any further assertions about the normality of reptile red blood cells, or their having a nucleus), `ReptileRedBldCel` would be classified neither as `TypicalEukCell` nor as `ATypicalEukCell`.

The natural encoding of the above knowledge base in $\mathscr{D}\mathscr{L}^{\mathrm{N}}$, denoted by $\mathscr{K}\mathscr{B}'$, consists only of:

- the strong axioms (3.54–3.57);
- the defeasible version of (3.60): $\texttt{EukCell} \sqsubseteq_n \exists\,\texttt{nucleus}$.

The other axioms need not be included explicitly, because $\mathscr{D}\mathscr{L}^{\mathrm{N}}$ is able to *infer* the properties of typical and atypical eukaryotic cells without any further directions:

$$\mathscr{K}\mathscr{B}' \mid\!\approx \mathrm{N}\texttt{AvianRedBldCel} \sqsubseteq \exists\,\texttt{nucleus} \quad \mathscr{K}\mathscr{B}' \mid\!\approx \mathrm{N}\texttt{RedBldCel} \sqsubseteq \exists\,\texttt{nucleus}\,.$$

Similarly, typical reptile red blood cells would automatically inherit the standard property of eukaryotic cell:

$$\mathscr{K}\mathscr{B}' \cup \{(3.64)\} \mid\!\approx \mathrm{N}\texttt{ReptileRedBldCel} \sqsubseteq \exists\,\texttt{nucleus}$$

If the first ontology, $\mathscr{K}\mathscr{B}$, were queried for the entities that have a nucleus (i.e. the concepts subsumed by $\exists\,\texttt{nucleus}$), then the query would return:

$$\texttt{TypicalEukCell},\ \texttt{TypicalRBC},\ \texttt{AvianRedBldCel}\,.$$

Similarly, the $\mathscr{D}\mathscr{L}^{\mathrm{N}}$ ontology $\mathscr{K}\mathscr{B}'$ would return:

$$\mathrm{N}\texttt{EukCell},\ \mathrm{N}\texttt{RedBldCel},\ \mathrm{N}\texttt{AvianRedBldCel}\,.$$

Thus, $\mathscr{D}\mathscr{L}^{\mathrm{N}}$ does support the factorization of common default properties, with a remarkable reduction of additional axioms even in this simple example.

The ontology design pattern is vulnerable to some trivial errors [Rector, 2004, Sec. 2.2]. If a concept A is subsumed by B_1 and B_2, and the typical instances of B_1 and B_2 are modeled using the same `status` role, then the instances of A that are typical with respect to B_1 become typical also w.r.t. B_2, and vice versa. A similar problem may occur if $\mathscr{K}\mathscr{B}$ is constructed as the union of two independently developed knowledge bases. Clearly, $\mathscr{D}\mathscr{L}^{\mathrm{N}}$ is not subject to the same vulnerabilities, since it needs no such explicit flags.

Next, we consider more complex scenarios. Rector introduces an example with multiple levels of exceptions (as in the policy example) in [Rector, 2004, Sec. 2.3]: it consists in a drug knowledge base capable of keeping track of interactions and contraindications. Rector writes:

> [...] *to be safe, we want to express interactions and contraindications at the most general level possible and inherit them by default, to be overridden if necessary.*

Clearly, the ODP for exceptions does not help in this respect, as shown by the reptile cells example: as new drug subtypes are added to the ontology, the knowledge engineer *must* explicitly tell whether they are typical or atypical, or (equivalently) whether they have the typical contraindication of their superclasses, otherwise the default contraindications are neither inherited nor overridden.

Using the ODP, a drug type `A` with contraindication X and a subtype `B` that shares with A the same contraindication would be represented as follows:

$$\mathtt{B} \sqsubseteq \mathtt{A} \tag{3.65}$$

$$\mathtt{TypicalA} \equiv \mathtt{A} \sqcap \exists\, \mathtt{status.Normal} \tag{3.66}$$

$$\mathtt{ATypicalA} \equiv \mathtt{A} \sqcap \neg \mathtt{TypicalA} \tag{3.67}$$

$$\mathtt{TypicalA} \sqsubseteq \mathtt{ContraindicationX} \tag{3.68}$$

$$\mathtt{B} \sqsubseteq \exists\, \mathtt{status.Normal}\,. \tag{3.69}$$

Let $\mathscr{KB}_1$ be the above knowledge base. Now suppose a new contraindication Y for A is discovered and must be included in $\mathscr{KB}_1$. There are three possible, alternative approaches:

1. The new default contraindication is incrementally added to $\mathscr{KB}_1$:

   $$\mathtt{TypicalA} \sqsubseteq \mathtt{ContraindicationY}\,.$$

   However, this formalization suffers from *inheritance blocking* (cf. Example 3.1.51): for instance, any subtype of `A` that overrides contraindication X is an atypical `A`, and hence it does not inherit contraindication Y, either. On the contrary, the incremental approach would work perfectly well in $\mathscr{DL}^{\mathrm{N}}$, that does not suffer from inheritance blocking (it would be sufficient to add $\mathtt{A} \sqsubseteq_n \mathtt{ContraindicationY}$).

2. The partition of `A` into two classes (`TypicalA` and `ATypicalA`) is replaced by a partition into four classes, corresponding to all the subsets of the default prop-

erties that can be satisfied by a drug. With this approach, a drug is allowed to inherit a subset of the contraindications for `A`. However, this approach introduces an exponential number of additional concepts. Moreover, asserting that a drug subtype `C` overrides contraindication X is not enough, because it does not tell the classifier whether contraindication Y should be overridden, too.

3. Concept `TypicalA` is replaced by a set of partially overlapping concepts ($i = 1, \ldots, n$, where n is the number of contraindications for A):

$$\begin{aligned} \texttt{TypicalA}_i &\equiv \texttt{A} \sqcap \exists \texttt{status}_i.\texttt{Normal} \\ \texttt{TypicalA}_i &\sqsubseteq \texttt{Contraindication}_i\,. \end{aligned}$$

This approach introduces only a linear number of new concepts. However, for each drug type, the knowledge engineer should assert which properties $\exists\texttt{status}_i.\texttt{Normal}$ hold, otherwise the classifier cannot infer the corresponding contraindications. Then the question is: what is the added value of this ODP? Would it be better to directly assert which contraindications apply to each drug?

Accordingly, in [Stevens et al., 2007] it is stated that:

> [The ODP for exceptions] *suffices for simple exceptions. However, exceptions can be piled upon exceptions, eventually leading to a combinatorial explosion. Worse still, some cells, such as muscle cells, have many nuclei. This means we would have to model a three way split with zero, one or many nuclei in a cell.*

Finally, suppose a new subtype $\texttt{B}'$ of drug `B` is discovered, that does not have contraindication X. In $\mathscr{DL}^{\mathrm{N}}$, this update can be handled incrementally, by extending the knowledge base $\{\texttt{B} \sqsubseteq \texttt{A},\ \texttt{A} \sqsubseteq_n \texttt{ContraindicationX}\}$ with:

$$\begin{aligned} \texttt{B}' &\sqsubseteq \texttt{B} \\ \texttt{B}' &\sqsubseteq_n \neg\texttt{ContraindicationX}\,. \end{aligned}$$

This update cannot be handled incrementally with the ODP: axiom (3.69) must be retracted and replaced with the definition of `TypicalB` and `ATypicalB`.

Summarizing, $\mathscr{DL}^{\mathrm{N}}$ overcomes the following drawbacks of the ODP for exceptions:

1. the ODP cannot effectively factorize common default properties, nor achieve real default inheritance;
2. the ODP does not support incremental refinements and extensions;

3. the definitions of typical and atypical concepts make use of computationally expensive constructs;
4. the ODP tends to clutter the knowledge base with auxiliary concepts and roles;
5. the additional symbols introduce more error possibilities.

It should be remarked that all the other nonmonotonic DLs that do not suffer from inheritance blocking share with $\mathscr{DL}^{\mathrm{N}}$ the same advantages over the ODP, with the exception of tractability: all of the above examples fall within the tractable fragment of $\mathscr{DL}^{\mathrm{N}}$, while no tractable fragments are known for the other logics.

Rule-based Approaches

The frameworks that combine logic programming (or similar languages) and description logics typically support nonmonotonic constructs similar to negation as failure in rule bodies. Some of these approaches are based on MKNF with standard domain [Motik and Rosati, 2010] and hence can be compared with $\mathscr{DL}^{\mathrm{N}}$ as discussed in Section 3.1.7. Others, such as [Eiter et al., 2008, Eiter et al., 2005], are based on a loose semantic integration of rules and DLs. They follow the standard approach to conflict resolution, analogous to considering all possible repairs. Therefore, tractability can only be achieved by means of well-known syntactic restrictions developed in logic programming, such as stratifiability, that prevent conflicts. In most rule-based systems, priorities between nonmonotonic rules and specificity-based conflict resolution are not immediately supported. Some systems (such as DLV) attach weights to soft constraints and compute models that maximize the weights of satisfied constraints [Leone et al., 2006]. All the approaches based on logic programming and answer set programming need rules to be grounded on some syntactic domain. In [Eiter et al., 2008, Eiter et al., 2005] such domain consists of the individual constants occurring in the knowledge base. Then nonmonotonic rules do not apply to implicit individuals; this yields the same effects discussed in Section 3.1.7.

3.2 Secure Knowledge Base Views

There is ample evidence of the need for knowledge confidentiality measures [Cuenca Grau, 2010]. Ontology languages and Linked Open Data are increasingly being used to encode the private knowledge of companies and public organizations. Semantic Web techniques make possible to merge different sources of knowledge and extract implicit information, putting on risk security and privacy of individuals. Even the authors of public ontologies may want to hide some axioms to capitalize on their formalization efforts. Several approaches have been proposed in order to tackle the confidentiality requirements that arise form these scenarios.

The most natural way of preserving confidentiality in a knowledge base $\mathscr{KB}$ is checking that its answers to user queries do not entail any secret. Conceptually, the queries of a user u are answered against u's view $\mathscr{KB}_u$ of the knowledge base, where $\mathscr{KB}_u$ is a maximal subset of $\mathscr{KB}$ that entails no secret. However, there exist attacks that cannot be prevented this way. The user may exploit various sources of background knowledge and metaknowledge to reconstruct the hidden part of the knowledge base. In order to illustrate some possible attacks to this mechanism, let us formalize the above *naive confidentiality model* (NCM)[20]. I consists of: the knowledge base $\mathscr{KB}$ ($\mathscr{KB} \subseteq \mathscr{L}$); a set of users U; a *view* $\mathscr{KB}_u \subseteq \mathscr{KB}$ for each $u \in U$; a set of *secrecies* $S_u \subseteq \mathscr{L}$ for each $u \in U$. Secrecies are axioms that may or may not be entailed by $\mathscr{KB}$; if they do, then they are called *secrets* and must not be disclosed to u. Revealing that a secrecy is *not* entailed by $\mathscr{KB}$ is harmless, cf. [Biskup and Bonatti, 2001].

A view $\mathscr{KB}_u$ is *secure* if and only if it does not entail any secret $Cn(\mathscr{KB}_u) \cap S_u = \emptyset$. A view $\mathscr{KB}_u$ is *maximal secure* if it is secure and there exists no K such that $\mathscr{KB}_u \subset K \subseteq \mathscr{KB}$ and $Cn(K) \cap S_u = \emptyset$.

Attacks using background knowledge. Frequently, part of the knowledge about the application domain is not axiomatized in $\mathscr{KB}$, therefore checking that $Cn(\mathscr{KB}_u) \cap S_u = \emptyset$ does not suffice in practice to protect confidentiality. For example, suppose that $\mathscr{KB}_u = \{SSN(John, 12345),\ SSN(user123, 12345),\ OncologyPatient(user123)\}$ and there is only one secret $S_u = \{OncologyPatient(John)\}$. $\mathscr{KB}_u$ does not entail $OncologyPatient(John)$, so according to the confidentiality model $\mathscr{KB}_u$ is secure. However, it is common knowledge that a Social Security Number uniquely identifies a person. As a consequence, the user can infer that $John = user123$, and hence the secret.

This example shows that *incomplete axiomatizations, e.g. failing to model (inverse)*

[20] This usage of term "*model*" is common in Security & Privacy.

functionality constraints, may constitute a security vulnerability in the NCM. Indeed, if $\mathscr{K}\mathscr{B}_u$ encoded the uniqueness of SSN then $\mathscr{K}\mathscr{B}_u$ would be recognized as insecure and the attack would be blocked. In other examples, the additional knowledge used to infer secrets may be stored in a public ontology or RDF repository, that opens the way to automatize confidentiality violations.

Attacks to complete knowledge. Suppose an attacker knows that $\mathscr{K}\mathscr{B}$ encodes complete knowledge about a certain set of axioms. Then she might be able to reconstruct some secrets from the "I don't know" answers of a maximal secure view $\mathscr{K}\mathscr{B}_u$.

Example 3.2.1 Consider a company's knowledge base that defines a concept *Employee* and a role *works_for* that describes which employees belong to which of the n departments of the company, $d_1, \ldots, d_n$. The $\mathscr{K}\mathscr{B}$ consists of assertions like:

$$Employee(e) \qquad (3.70) \qquad\qquad works\_for(e, d_i) \qquad (3.71)$$

where each employee e belongs to exactly one department d_i. A user u is authorized to see all assertions but the instances of (3.71) with $i = n$, because d_n is a special department, devoted to controlling the other ones. So S_u (the set of secrecies for u) is the set of all assertions $works\_for(e, d_n)$.

Here a unique maximal secure view $\mathscr{K}\mathscr{B}_u$ exists which contains all the instances of (3.70), together with all the instances of (3.71) such that $i \neq n$. It is easy to see that $\mathscr{K}\mathscr{B}_u$ is secure according to NCM because $Cn(\mathscr{K}\mathscr{B}_u) \cap S_u = \emptyset$. However, note that $works\_for(e, d_n) \in Cn(\mathscr{K}\mathscr{B})$ iff $Employee(e) \in Cn(\mathscr{K}\mathscr{B}_u)$ and for all $i = 1, \ldots, n$, $works\_for(e, d_i) \notin Cn(\mathscr{K}\mathscr{B}_u)$. In other words, the members of d_n are all the employees that apparently work for no department. Using this property (based on the knowledge that for each employee e, $\mathscr{K}\mathscr{B}$ contains exactly one assertion $works\_for(e, d_i)$) and the knowledge of the protection mechanism (i.e. maximal secure views), that we assume to be known by attackers by *Kerchoff's principle*[21]. ∎

In practice, it is not hard to identify complete knowledge. A hospital's $\mathscr{K}\mathscr{B}$ is expected to have complete knowledge about which patients are in which ward; a company's $\mathscr{K}\mathscr{B}$ is likely to encode complete information about its employees, etc.

Other approaches filter query answers rather than publishing a subset of $\mathscr{K}\mathscr{B}$ [Chen and Stuckenschmidt, 2009, Knechtel and Stuckenschmidt, 2010,

[21] "A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.", originally stated by Auguste Kerckhoffs, the principal was later reformulated by Claude Shannon as "the enemy knows the system", i.e., "systems should be designed under the assumption that the enemy will immediately gain full familiarity with them"

Tao et al., 2010]. We call our abstraction of this method *naive answer confidentiality model* (NACM). It is obtained from the NCM by replacing the views $\mathcal{K}\mathcal{B}_u \subseteq \mathcal{K}\mathcal{B}$ with *answer views* $\mathcal{K}\mathcal{B}_u^a \subseteq Cn(\mathcal{K}\mathcal{B})$. The main difference is that $\mathcal{K}\mathcal{B}_u^a$ is not required to be a subset of $\mathcal{K}\mathcal{B}$ and conceptually $\mathcal{K}\mathcal{B}_u^a$ may be infinite. $\mathcal{K}\mathcal{B}_u^a$ is *secure* iff $Cn(\mathcal{K}\mathcal{B}_u^a) \cap S_u = \emptyset$.

One may easily verify that NACM is vulnerable to the two kinds of attacks illustrated for the NCM. Furthermore, it is also vulnerable to a third kind of attacks, illustrated below.

Attacks to the signature. Suppose the user knows the signature of $\mathcal{K}\mathcal{B}$ well enough to identify a symbol σ that does not occur in $\mathcal{K}\mathcal{B}$. First assume that σ is a concept name. It can be proved that:

Proposition 3.2.2 *If $\mathcal{K}\mathcal{B}_u^a$ is a maximal secure answer view and σ is a concept name not occurring in $\mathcal{K}\mathcal{B}$, then for all secrecies $C \sqsubseteq D \in S_u$, $\mathcal{K}\mathcal{B}_u^a \models C \sqcap \sigma \sqsubseteq D$ iff $\mathcal{K}\mathcal{B} \models C \sqsubseteq D$.*

The problem is that although $C \sqcap \sigma \sqsubseteq D$ does not entail the secret inclusion $C \sqsubseteq D$, still a smart attacker knows that the former inclusion cannot be proved unless $\mathcal{K}\mathcal{B}$ entails also the latter (then maximal secure answer views generally fail to protect secrets). This attack can be easily adapted to the case where σ is a role name. In practice, it is not necessary to be sure that σ does not occur in $\mathcal{K}\mathcal{B}$. The attacker may make a sequence of educated guesses (say, by trying meaningless long strings, or any word that is clearly unrelated to the domain of the $\mathcal{K}\mathcal{B}$); after a sufficient number of trials, the majority of answers should agree with the "real" answer with high probability. Rejecting queries whose signature is not contained in $\mathcal{K}\mathcal{B}$'s signature mitigates this kind of attacks but it leaks $\mathcal{K}\mathcal{B}$'s signature and it does not provide a complete solution. The attacker may still guess a σ which is logically unrelated to C and D and carry out a similar attack.

In the following section we introduce a confidentiality model that takes both background knowledge and metaknowledge into account (Section 3.2.1) and defines a method for computing secure knowledge views that generalizes some previous approaches. Section 3.2.2 and 3.2.3 illustrate a safe approximation of the user's background knowledge and metaknowledge. Finally, in Section 3.2.5, the approach is compared with other existing confidentiality preserving frameworks.

3.2.1 A Meta-safe Confidentiality Model

A confidentiality model that makes the vulnerabilities illustrated above visible, by taking into account background knowledge and metaknowledge was first introduced in [Bonatti and Sauro, 2013]. The framework is compatible with any description logic language $\mathscr{L}$ that enjoys compactness (needed by Theorem 3.2.21) and has decidable reasoning problems (e.g., $\mathscr{ALC}$, $\mathscr{EL}$, $\mathscr{SHIQ}$, etc.).

In the following, for all *knowledge bases*[22] $K \subseteq \mathscr{L}$, the logical consequences of K will be denoted by $Cn(K)$ ($K \subseteq Cn(K) \subseteq \mathscr{L}$). In particular, $Cn(K)$ contains all the subsumptions and assertions entailed by K (corresponding to subsumption and instance checks, respectively).

A *bk-model* $\mathscr{M} = \langle \mathscr{KB}, U, f, \langle S_u, PKB_u, BK_u \rangle_{u \in U} \rangle$ consists of a knowledge base $\mathscr{KB} \subseteq \mathscr{L}$, a set of users U, plus:

- a *filtering function* $f : \wp(\mathscr{L}) \times U \to \wp(\mathscr{L})$, mapping each knowledge base K and each user u on a view $f(K, u) \subseteq Cn(K)$;
- for all $u \in U$:
  - a finite set of secrecies $S_u \subseteq \mathscr{L}$;
  - a set of axioms $BK_u \subseteq \mathscr{L}$, encoding the users' background knowledge;
  - a set of *possible knowledge bases* $PKB_u \subseteq \wp(\mathscr{L})$ (users' metaknowledge).[23]

The view of $\mathscr{KB}$ released to a user u is $f(\mathscr{KB}, u)$. PKB represents the knowledge bases that are compatible with the user's metaknowledge regardless of the choice of any specific metalanguage.

Definition 3.2.3 A filtering function f is *secure* (w.r.t. $\mathscr{M}$) iff for all $u \in U$ and all $s \in S_u$, there exists $K \in PKB_u$ such that:

1. $f(K, u) = f(\mathscr{KB}, u)$;
2. $s \notin Cn(K \cup BK_u)$.

Intuitively, if f is safe according to Def. 3.2.3, no user u can conclude that any secret s is entailed by the $\mathscr{KB}$ she is interacting with—enhanced with the background knowledge BK_u. More precisely, by point 1, $\mathscr{KB}$ and K have the same observable behavior, and K is a possible knowledge base for u since $K \in PKB_u$; therefore, as far as u knows, the

[22] Real knowledge bases are finite, but this restriction is not technically needed until Section 3.2.3.

[23] In practice, bk-models are finite, and filterings computable, but no such assumption will be technically needed until Section 3.2.3.

knowledge base might be K. Moreover, by point 2, K and the background knowledge BK_u do not suffice to entail the secret s.

In the rest of the section we tacitly assume that no secret is violated a priori, that is, for all secrets $s \in S_u$ there exists $K \in PKB_u$ such that $s \notin Cn(K \cup BK_u)$.[24] As a consequence, there exists at least one secure f, namely, the constant filtering function that always returns an empty set. *In order to improve readability, we shall omit the user u from subscripts and argument lists* whenever u is irrelevant to the context.

The attacks discussed in the previous section can be easily formalized in this setting. So, in general, the maximal secure views of NCM are not secure according to Def. 3.2.3.

Example 3.2.4 Example 3.2.1 can be formalized as follows: The set of secrecies S is the set of all assertions $works\_for(e, d_n)$; $BK = \emptyset$ and PKB is the set of all the knowledge bases K that consist of assertions like (3.70) and (3.71), and such that for each axiom $Employee(e)$, K contains exactly one corresponding axiom $works\_for(e, d_i)$ and vice versa. The filtering function f maps each $K \in PKB$ on the maximal subset of K that entails none of S's members, that is, $f(K) = K \setminus S$ (by definition of PKB).

Note that f is injective over PKB, so condition 1 of Def. 3.2.3 is satisfied only if $K = \mathscr{K}\mathscr{B}$. So, if $\mathscr{K}\mathscr{B}$ contains at least one secret, then the conditions of Def. 3.2.3 cannot be satisfied, that is, maximal secure views are not secure in this model. Indeed, $\mathscr{K}\mathscr{B}$ can be reconstructed from the secure view by observing that $\mathscr{K}\mathscr{B} = f(\mathscr{K}\mathscr{B}) \cup \{works\_for(e, d_n) \mid Employee(e) \in f(\mathscr{K}\mathscr{B}) \wedge \forall i = 1, \ldots, n, works\_for(e, d_i) \notin f(\mathscr{K}\mathscr{B})\}$. - ∎

Similarly, the formalizations of the other attacks yield injective filtering functions.

Next, we define a *secure filtering function*. It is formulated as an iterative process based on a *censor*. The censor is a boolean function that decides for each axiom whether it should be obfuscated in order to preserve confidentiality. The censor's role includes deciding "secondary protection", that is, which additional axioms—besides those that entail a secret—should be obfuscated as well.

The iterative construction manipulates pairs $\langle X^+, X^- \rangle \in \wp(\mathscr{L}) \times \wp(\mathscr{L})$ that represent a meta constraint on possible knowledge bases: a knowledge base K *satisfies* $\langle X^+, X^- \rangle$ iff K entails all the sentences in X^+ and none of those in X^-. Formally, $Cn(K) \supseteq X^+$ and $Cn(K) \cap X^- = \emptyset$.

Let PAX (the set of *possible axioms*) be the set of axioms that may occur in the knowledge base according to the user's knowledge, i.e. $PAX = \bigcup_{K' \in PKB} K'$. Let $\nu = |PAX| + 1$ if PAX is finite and $\nu = \omega$ otherwise; let $\alpha_1, \alpha_2, \ldots, \alpha_i, \ldots$ be an arbitrary

[24] Conversely, no filtering function can conceal a secret that is already known by the user.

enumeration of PAX $(i < \nu)$.[25] The secure view construction for a knowledge base K in a bk-model $\mathscr{M}$ consists of the following, inductively defined sequence of pairs $\langle K_i^+, K_i^- \rangle_{i \geq 0}$:

- $\langle K_0^+, K_0^- \rangle = \langle \emptyset, \emptyset \rangle$, and for all $1 \leq i < \nu$, $\langle K_{i+1}^+, K_{i+1}^- \rangle$ is defined as follows:
  - if $censor_{\mathscr{M}}(K_i^+, K_i^-, \alpha_{i+1}) = true$ then let $\langle K_{i+1}^+, K_{i+1}^- \rangle = \langle K_i^+, K_i^- \rangle$;-
  - if $censor_{\mathscr{M}}(K_i^+, K_i^-, \alpha_{i+1}) = false$ and $K \models \alpha_{i+1}$ then $\langle K_{i+1}^+, K_{i+1}^- \rangle = \langle K_i^+ \cup \{\alpha_{i+1}\}, K_i^- \rangle$;
  - otherwise let $\langle K_{i+1}^+, K_{i+1}^- \rangle = \langle K_i^+, K_i^- \cup \{\alpha_{i+1}\} \rangle$.

Finally, let $K^+ = \bigcup_{i<\nu} K_i^+$, $K^- = \bigcup_{i<\nu} K_i^-$, and $f_{\mathscr{M}}(K, u) = K^+$.

Observe that the inductive construction aims at finding maximal sets K^+ and K^- that (i) partly describe what does / does not follow from K (as K satisfies $\langle K^+, K^- \rangle$ by construction), and (ii) do not trigger the censor (the sentences α_{i+1} that trigger the censor are included neither in K^+ nor in K^-, cf. the induction step).

In order to define the censor we need an auxiliary definition that captures all the sentences that can be entailed from a given pair $\langle X^+, X^- \rangle$ analogous to those adopted in the iterative construction enriched by the user's background knowledge BK and metaknowledge PKB: Let $Cn_{\mathscr{M}}(X^+, X^-)$ be the set of all axioms $\alpha \in \mathscr{L}$ such that

$$\text{for all } K' \in PKB \text{ such that } K' \text{ satisfies } \langle X^+, X^- \rangle, \alpha \in Cn(K' \cup BK). \tag{3.72}$$

Consequently, the censor can be defined as follows: For all $X^+, X^- \subseteq \mathscr{L}$ and $\alpha \in \mathscr{L}$,

$$censor_{\mathscr{M}}(X^+, X^-, \alpha) = \begin{cases} true & \text{if there exists } s \in S \text{ s.t. either } s \in Cn_{\mathscr{M}}(X^+ \cup \{\alpha\}, X^-) \text{ or } s \in Cn_{\mathscr{M}}(X^+, X^- \cup \{\alpha\}); \\ false & \text{otherwise.} \end{cases}$$

In other words, the censor checks whether telling either that α is derivable or that α is not derivable to a user aware that the knowledge base satisfies $\langle X^+, X^- \rangle$, restricts the set of possible knowledge bases enough to conclude that a secret s is entailed by the knowledge base, the background knowledge BK and metaknowledge PKB.

Note that the censor obfuscates α_{i+1} if *any* of its possible answers entail a secret, independently of the actual contents of K (the two possible answers "yes" and "no" correspond to conditions $s \in Cn_{\mathscr{M}}(X^+ \cup \{\alpha\}, X^-)$ and $s \in Cn_{\mathscr{M}}(X^+, X^- \cup \{\alpha\})$,

[25] Later it will become clear how to restrict the construction to finite sequences, by approximating PAX.

respectively). In this way, roughly speaking, the knowledge bases that entail s are given the same observable behavior as those that don't. Under a suitable continuity assumption on $Cn_{\mathscr{M}}$, this enforces confidentiality:

Theorem 3.2.5 *If* $Cn_{\mathscr{M}}(\mathscr{K}\mathscr{B}^+, \mathscr{K}\mathscr{B}^-) \subseteq \bigcup_{i<\nu} Cn_{\mathscr{M}}(\mathscr{K}\mathscr{B}_i^+, \mathscr{K}\mathscr{B}_i^-)$*, then* $f_{\mathscr{M}}$ *is secure w.r.t.* $\mathscr{M}$*.*

3.2.2 Approximating Users' Knowledge

Of course, the actual confidentiality of a filtering $f(\mathscr{K}\mathscr{B}, u)$ depends on a careful definition of the user's background knowledge and metaknowledge, that is, BK_u and PKB_u. If background knowledge is not exactly known, as it typically happens, then it can be safely approximated by *overestimating* it. More background knowledge means larger BK_u and smaller PKB_u, which leads to the following comparison relation $\leq_k$ over bk-models:

Definition 3.2.6 Given two bk-models $\mathscr{M} = \langle \mathscr{K}\mathscr{B}, U, f, \langle S_u, PKB_u, BK_u\rangle_{u\in U}\rangle$ and $\mathscr{M}' = \langle \mathscr{K}\mathscr{B}', U', f', \langle S'_u, PKB'_u, BK'_u\rangle_{u\in U'}\rangle$, we write $\mathscr{M} \leq_k \mathscr{M}'$ iff

1. $\mathscr{K}\mathscr{B} = \mathscr{K}\mathscr{B}'$, $U = U'$, $f = f'$, and $S_u = S'_u$ (for all $u \in U$);
2. for all $u \in U$, $PKB_u \supseteq PKB'_u$ and $BK_u \subseteq BK'_u$.

Then a bk-model $\mathscr{M}$ can be safely approximated by any $\mathscr{M}'$ such that $\mathscr{M} \leq_k \mathscr{M}'$:

Proposition 3.2.7 *If* f *is secure w.r.t.* $\mathscr{M}'$ *and* $\mathscr{M} \leq_k \mathscr{M}'$*, then* f *is secure w.r.t.* $\mathscr{M}$*.*

So, a generic advice for estimating BK consists in including as many pieces of relevant knowledge as possible, for example:

(i) modeling as completely as possible the integrity constraints satisfied by the data, as well as role domain and range restrictions and disjointness constraints;

(ii) including in BK all the relevant public sources of formalized relevant knowledge (such as ontologies and triple stores).

While background knowledge is dealt with in the literature, the general metaknowledge encoded by PKB is novel in [Bonatti and Sauro, 2013].

3.2.3 Approximating and Reasoning about Possible Knowledge Bases

In the following, we focus on real world situations where *the knowledge base* $\mathscr{K}\mathscr{B}$ *is finite* and *so are all the components of bk-models.* Restricting PKB_u to contain only finite knowledge bases turns out to guarantee the decidability of $f_{\mathscr{M}}$.

A language for defining *PKB* is a necessary prerequisite for the practical implementation of the framework and a detailed complexity analysis of the secure filtering function $f_{\mathscr{M}}$. *PKB* can be expressed as the set of all theories that are contained in a given set of *possible axioms* *PAX* [26] and satisfy a given, finite set *MR* of *metarules* like:

$$\alpha_1, \ldots, \alpha_n \Rightarrow \beta_1 \mid \ldots \mid \beta_m \quad (n \geq 0, m \geq 0)\,, \tag{3.73}$$

where all α_i and β_j are in $\mathscr{L}$ $(1 \leq i \leq n, 1 \leq j \leq m)$. Informally, (3.73) means that if $\mathscr{K}\mathscr{B}$ entails $\alpha_1, \ldots, \alpha_n$ then $\mathscr{K}\mathscr{B}$ entails also some of $\beta_1, \ldots, \beta_m$. If r denotes rule (3.73), then let $body(r) = \{\alpha_1, \ldots, \alpha_n\}$ and $head(r) = \{\beta_1, \ldots, \beta_m\}$. A rule r is *Horn* if $|head(r)| \leq 1$. Sets of similar metarules can be succinctly specified using *metavariables* that can be placed wherever individual constants may occur, i.e., as arguments of assertions, and in nominals. A metarule with such variables abbreviates the set of its *ground instantiations*: Given a $K \subseteq \mathscr{L}$, let $ground_K(MR)$ be the ground (variable-free) instantiation of *MR* where metavariables are uniformly replaced by the individual constants occurring in K in all possible ways.

Example 3.2.8 Let $MR = \{\, \exists R.\{X\} \Rightarrow A(X) \,\}$, where X is a metavariable, and let $K = \{\, R(a,b) \,\}$. Then $ground_K(MR) = \{\, (\exists R.\{a\} \Rightarrow A(a)),\ (\exists R.\{b\} \Rightarrow A(b)) \,\}$. ∎

A set of axioms $K \subseteq \mathscr{L}$ *satisfies* a ground metarule r, formally $K \models_m r$, if either $body(r) \not\subseteq Cn(K)$ or $head(r) \cap Cn(K) \neq \emptyset$.

Example 3.2.9 Let A, B, C be concept names and R be a role name. The axiom set $K = \{A \sqsubseteq \exists R.B, A \sqsubseteq C\}$ satisfies $A \sqsubseteq \exists R \Rightarrow A \sqsubseteq B \mid A \sqsubseteq C$ but not $A \sqsubseteq \exists R \Rightarrow A \sqsubseteq B$. ∎

We write $K \models_m MR$ if K satisfies all the metarules in $ground_K(MR)$. Therefore the formal definition of *PKB* now becomes:

$$PKB = \{K \mid K \subseteq PAX \wedge K \models_m MR\}\,. \tag{3.74}$$

According to Prop. 3.2.7, *PAX* can be approximated in a conservative way. Two alternative definitions are possible:

1. $PAX_0 = \mathscr{K}\mathscr{B}$ (i.e., as a minimalistic choice only the axioms of $\mathscr{K}\mathscr{B}$ are considered as possible axioms. By Prop. 3.2.7, this choice is safe also w.r.t. any larger *PAX* where *at least* the axioms of $\mathscr{K}\mathscr{B}$ are regarded as possible axioms);

[26] Differently from Section 3.2.1, here *PKB* is defined in terms of *PAX*.

2. $PAX_1 = \mathscr{K}\mathscr{B} \cup \bigcup_{r \in ground_{\mathscr{K}\mathscr{B}}(MR)} head(r)$.

Remark 3.2.10 The second definition is most natural when metarules are automatically extracted from $\mathscr{K}\mathscr{B}$ with rule mining techniques, that typically construct rules using material from the given $\mathscr{K}\mathscr{B}$ (then rule heads occur in $\mathscr{K}\mathscr{B}$).

Example 3.2.11 Consider again Example 3.2.1. The user's metaknowledge about $\mathscr{K}\mathscr{B}$'s completeness can be encoded with:

$$Employee(X) \Rightarrow works\_for(X, d_1) \mid \ldots \mid works\_for(X, d_n)\,, \tag{3.75}$$

where X is a metavariable. First let $PAX = PAX_1$. The secure view $f_{\mathscr{M}}(\mathscr{K}\mathscr{B})$ depends on the enumeration order of PAX. If the role assertions $works\_for(e, d_i)$ precede the concept assertions $Employee(e)$, then, in a first stage, the sets $\mathscr{K}\mathscr{B}_j^+$ are progressively filled with the role assertions with $d_i \neq d_n$ that belong to $\mathscr{K}\mathscr{B}$, while the sets $\mathscr{K}\mathscr{B}_j^-$ accumulate all the role assertions that do not belong to $\mathscr{K}\mathscr{B}$. In a second stage, the sets $\mathscr{K}\mathscr{B}_j^+$ are further extended with the concept assertions $Employee(e)$ such that e does not work for d_n. The role assertions $works\_for(e, d_n)$ of $\mathscr{K}\mathscr{B}$ and the corresponding concept assertions $Employee(e)$ are neither in $\mathscr{K}\mathscr{B}^+$ nor in $\mathscr{K}\mathscr{B}^-$. Note that the final effect is equivalent to removing from $\mathscr{K}\mathscr{B}$ all the axioms referring to the individuals that work for d_n.

Next suppose that the role assertions $works\_for(e, d_i)$ follow the concept assertions $Employee(e)$, and that each $works\_for(e, d_i)$ follows all $works\_for(e, d_k)$ such that $k < i$. Now all the assertions $Employee(e)$ of $\mathscr{K}\mathscr{B}$ enter $\mathscr{K}\mathscr{B}^+$, and all axioms $works\_for(e, d_i)$ with $i < n - 1$ enter either $\mathscr{K}\mathscr{B}^+$ or $\mathscr{K}\mathscr{B}^-$, depending on whether they are members of $\mathscr{K}\mathscr{B}$ or not. Finally, the assertions $works\_for(e, d_i) \in Cn(\mathscr{K}\mathscr{B})$ with $i \in \{n-1, n\}$ are inserted neither in $\mathscr{K}\mathscr{B}^+$ nor in $\mathscr{K}\mathscr{B}^-$, because the corresponding instance of (3.75) with $X = e$ has the body in $\mathscr{K}\mathscr{B}^+$ and the first $n - 2$ alternatives in the head in $\mathscr{K}\mathscr{B}^-$, therefore a negative answer to $works\_for(e, d_{n-1})$ would entail the secret $works\_for(e, d_n)$ by (3.75). This triggers the censor for all assertions $works\_for(e, d_{n-1})$. Summarizing, with this enumeration ordering it is possible to return the complete list of employees; the members of d_n are protected by hiding also which employees belong to d_{n-1}.

Finally, let $PAX = PAX_0$. Note that in this case all possible knowledge bases are subsets of $\mathscr{K}\mathscr{B}$, that contains exactly one assertion $works\_for(e, d_{i(e)})$ for each employee e. To satisfy (3.75), every $K \in PKB$ containing $Employee(e)$ must contain also $works\_for(e, d_{i(e)})$. It follows that $f_{\mathscr{M}}$ must remove all references to the individuals that work for d_n, as it happens with the first enumeration of PAX_1. ∎

Definition 3.2.12 A bk-model $\mathscr{M}$ is *canonical* if for all users $u \in U$, PAX_u is either PAX_0 or PAX_1 and PKB_u is defined by (3.74) for a given MR_u. Moreover, $\mathscr{M}$ is *in a description logic DL* if for all $u \in U$, all the axioms in $\mathscr{KB}$, PKB_u, BK_u, and S_u belong to DL.

By definition the size of PAX_0 and PAX_1 is polynomial in the size of $\mathscr{KB} \cup MR$, therefore PKB is finite and exponential in the size of $\mathscr{KB} \cup MR$. Finiteness implies the continuity hypothesis on $Cn_{\mathscr{M}}$ of Theorem 3.2.5, and hence:

Theorem 3.2.13 *If $\mathscr{M}$ is canonical, then $f_{\mathscr{M}}$ is secure with respect to all $\mathscr{M}' \leq_k \mathscr{M}$.*

The complexity of constructing the secure view $f_{\mathscr{M}}(\mathscr{KB})$ when the underlying description logic is tractable, like $\mathscr{EL}$ and DL-lite, depends on the number of variables in MR.

Lemma 3.2.14 *If the axioms occurring in MR and K are in a DL with tractable subsumption and instance checking, then checking $K \models_m MR$ is:*

1. *in P if either MR is ground or there exists a fixed bound on the number of distinct variables in MR;*
2. *coNP-complete otherwise.*

With Lemma 3.2.14, one can prove the following two lemmas.

Lemma 3.2.15 *Let $\mathscr{M}$ range over canonical bk-models. If $\mathscr{M}$, s, X^+, and X^- are in a DL with tractable subsumption/instance checking, and the number of distinct variables in MR is bounded by a constant, then checking whether $s \in Cn_{\mathscr{M}}(X^+, X^-)$ is:*

1. *in P if MR is Horn and $PAX = PAX_1$;*
2. *coNP-complete if either MR is not Horn or $PAX = PAX_0$.*

Lemma 3.2.16 *Let $\mathscr{M}$ be a canonical bk-model. If $\mathscr{M}$, s, X^+, and X^- are in a DL with tractable entailment problems, and there is no bound on the number of variables in the metarules of MR, then checking $s \in Cn_{\mathscr{M}}(X^+, X^-)$ is:*

1. *in P^{NP} if MR is Horn and $PAX = PAX_1$;*
2. *in Π_2^p if either MR is not Horn or $PAX = PAX_0$.*

The value of $censor(X^+, X^-, \alpha)$ can be computed straightforwardly by iterating the tests $s \in Cn_{\mathscr{M}}(X^+ \cup \{\alpha\}, X^-)$ and $s \in Cn_{\mathscr{M}}(X^+, X^- \cup \{\alpha\})$ for all secrets $s \in S$. Since the set of secrets is part of the parameter $\mathscr{M}$ of the filtering function, the number of iterations is polynomial in the input and the complexity of the censor is dominated by the complexity of $Cn_{\mathscr{M}}()$. The latter is determined by Lemma 3.2.15 and Lemma 3.2.16, so we immediately get:

Corollary 3.2.17 *Let $\mathscr{M}$ be a canonical bk-model and suppose that $\mathscr{M}$, X^+, X^-, and α are in a DL with tractable entailment problems. If the number of distinct variables in MR is bounded by a constant, then computing $censor(X^+, X^-, \alpha)$ is:*

- *in P if MR is Horn and $PAX = PAX_1$;*
- *coNP-complete if either MR is not Horn or $PAX = PAX_0$.*

If there is no bound on the number of variables in the metarules of MR, then computing $censor(X^+, X^-, \alpha)$ is:

- *in P^{NP} if MR is Horn and $PAX = PAX_1$;*
- *in Π_2^p if either MR is not Horn or $PAX = PAX_0$.*

The overall complexity of filtering functions is given by:

Theorem 3.2.18 *If $\mathscr{M}$ is a canonical bk-models in a DL with tractable entailment problems, then computing $f_{\mathscr{M}}(\mathscr{KB})$ is:*

1. *P-complete if the number of distinct variables in the rules of MR is bounded, MR is Horn, and $PAX = PAX_1$;*
2. *P^{NP}-complete if the number of distinct variables in MR is bounded, and either MR is not Horn or $PAX = PAX_0$;*
3. *in P^{NP} if the variables in MR are unbounded, MR is Horn, and $PAX = PAX_1$;*
4. *in Δ_3^p if MR is not restricted and $PAX \in \{PAX_0, PAX_1\}$.*

Theorem 3.2.19 *Computing $f_{\mathscr{M}}(\mathscr{KB})$ over canonical $\mathscr{M}$ in a DL with ExpTime entailment (e.g. $\mathscr{ALC}\mathscr{Q}\mathscr{O}$, $\mathscr{ALC}\mathscr{I}\mathscr{O}$, $\mathscr{ALC}\mathscr{Q}\mathscr{I}$, $\mathscr{SHO}\mathscr{Q}$, $\mathscr{SHI}\mathscr{O}$, $\mathscr{SHI}\mathscr{Q}$), is still in ExpTime.*

Theorem 3.2.20 *Computing $f_{\mathscr{M}}(\mathscr{KB})$ over canonical $\mathscr{M}$ in $\mathscr{SROI}\mathscr{Q}(\mathscr{D})$ is in $coNP^{N2ExpTime}$.*

The interested reader can refer to [Bonatti and Sauro, 2013] for further details and formal proofs of the complexity results.

3.2.4 Relationships with the NCM

The meta-secure framework can be regarded as a natural generalization of the NCM. The main result—roughly speaking—demonstrates that the NCM model can be essentially regarded as a special case of the meta-secure framework where $PKB \supseteq \wp(\mathcal{KB})$ and $BK = \emptyset$. In this case $f_{\mathscr{M}}$ is secure even if $\mathscr{M}$ is not assumed to be canonical.

Theorem 3.2.21 *Let $\mathscr{M} = \langle \mathcal{KB}, U, f_{\mathscr{M}}, \langle S_u, PKB_u, BK_u \rangle_{u \in U} \rangle$. If $PKB = \wp(\mathcal{KB})$, $BK = \emptyset$, and $\mathcal{KB}$ is finite, then*

1. *$Cn_{\mathscr{M}}(\mathcal{KB}^+, \mathcal{KB}^-) = \bigcup_{i<\nu} Cn_{\mathscr{M}}(\mathcal{KB}_i^+, \mathcal{KB}_i^-)$.*
2. *For all enumerations of PAX, the corresponding $f_{\mathscr{M}}(\mathcal{KB}, u)$ is logically equivalent to a maximal secure view $\mathcal{KB}_u$ of $\mathcal{KB}$ according to the NCM; conversely, for all maximal secure view $\mathcal{KB}_u$ of $\mathcal{KB}$ (according to the NCM) there exists an enumeration of PAX such that the resulting $f_{\mathscr{M}}(\mathcal{KB}, u)$ is logically equivalent to $\mathcal{KB}_u$.*
3. *$f_{\mathscr{M}}$ is secure w.r.t. $\mathscr{M}$ and w.r.t. any $\mathscr{M}' = \langle \mathcal{KB}, U, f_{\mathscr{M}}, \langle S_u, PKB'_u, BK'_u \rangle_{u \in U} \rangle$ such that $PKB' \supseteq \wp(\mathcal{KB})$ and $BK' = \emptyset$.*

By this correspondence, one can immediately obtains complexity bounds for the NCM from those for PAX_1 and Horn, bounded-variable *MR*.

3.2.5 Related Work

Early works focused on ontologies for security [Wishart et al., 2005, Blanco et al., 2008] and policy encoding with description logics [Uszok et al., 2003, Kolovski et al., 2007]. The pros and cons of this approach and a comparison with Datalog as a language expressly related to the representation and reasoning tasks involved in policy authoring, enforcement, and management have been discussed in [Bonatti, 2010]. The Datalog-based approaches appear currently more powerful and mature than those based on pure DLs, although the ongoing research on the latter, e.g. employing specifically designed non monotonic DLs as , might change the picture in a near future.

The framework presented in this chapter has a different goal though: protecting the confidentiality of knowledge. Some of the early approaches to this problem focused on access control models for XML documents; the focus was on selecting and encrypting portions of the document's syntax tree without considering inference mechanisms. [Fundulaki and Marx, 2004] introduce XPath 1.0 as a formal specification language for XML access control policies and survey and formalize the semantics of the existing approaches in the literature that were originally expressed in

natural language [Bertino and Ferrari, 2002, Damiani et al., 2000, Damiani et al., 2002, Gabillon and Bruno, 2002, Murata et al., 2003].

Early access control models for RDF triple stores, on the other hand, define high-level specification languages that enable fine-grained control of access permissions (at triple level) but do not deal with inference as well, e.g. [Abel et al., 2007, Flouris et al., 2010]. In the following we briefly discuss the approaches that do.

Baader et al. [Baader et al., 2009], Eldora et al. [Eldora et al., 2011], and Knechtel and Stuckenschmidt [Knechtel and Stuckenschmidt, 2010] attach security level labels to axioms and users to determine which subset of the knowledge base can be seen by each subject.

Instead of creating an exponentially many different sub-ontologies, one for every user, a label for each consequence is derived in a way that a comparison between the user and the consequence label determines whether the consequence is entailed form the corresponding sub-ontology. Reasoning then generalizes to the task of finding so-called boundary label for each implicit consequence of the ontology.

However, in [Baader et al., 2009, Eldora et al., 2011] axiom labels are not derived from the set of secrets; knowledge engineers are responsible for checking ex post that no confidential knowledge is entailed. In case of leakage, the labels can be modified with a revision tool based on pinpointing [Baader and Peñaloza, 2010]. On the contrary, the mechanism presented in this chapter automatically selects which axioms shall be hidden in order to produce a secure view. This issue is partially tackled in [Knechtel and Stuckenschmidt, 2010], by a procedure to optimally repair a given axiom labeling so that access restrictions defined in terms of queries can be enforced.

These works pursue the construction of maximal secure views so they are potentially vulnerable to the attacks based on background and metaknowledge. Similar considerations hold for [Tao et al., 2010] where secrecy-preserving query answering in $\mathscr{EL}\ \mathscr{KB}$ is based on the idea of constructing secrecy envelopes by inverting the tableau expansion rules.

Chen and Stuckenschmid [Chen and Stuckenschmidt, 2009] propose an alternative query rewriting approach for enforcing access restrictions in the context of SPARQL queries, while the TBox is assumed to be completely public. Filter conditions are automatically added to the user queries so to suppress such answers the user is not supposed to see which results in removing some individuals entirely. In general, this may be secure against metaknowledge attacks (cf. Example 3.2.11) but comes to a price: usually more knowledge than necessary remains hidden. Furthermore, no methodology is provided for selecting the individuals to be removed given a target set of secrets.

In [Bao et al., 2007], $\mathscr{K}\mathscr{B}$ is partitioned into a visible part $\mathscr{K}\mathscr{B}_v$ and a hidden part $\mathscr{K}\mathscr{B}_h$ which contains the information that is sensitive from a privacy point of view. Conceptually, this is analogous to axiom labeling in the above approaches. However, the confidentiality methodology seems to work only under the assumption that the signatures of $\mathscr{K}\mathscr{B}_v$ and $\mathscr{K}\mathscr{B}_h$ are disjoint. In particular, a strong safety reasoner prevents from disclose only consequence that can be drawn from the hidden knowledge alone. Formulae implied by a combination of $\mathscr{K}\mathscr{B}_v$ and $\mathscr{K}\mathscr{B}_h$ are not considered. Certainly the axioms of $\mathscr{K}\mathscr{B}_h$ whose signature is included in the signature of $\mathscr{K}\mathscr{B}_v$ cannot be protected in general. A partition-based approach is taken in [Cuenca Grau and Motik, 2009], too. The focus is on reusing ontologies with hidden content. In order to enable reasoning on $\mathscr{K}\mathscr{B}_v \cup \mathscr{K}\mathscr{B}_h$ without providing physical access, the axioms of $\mathscr{K}\mathscr{B}_h$ are accessed via an oracle (i.e., a limited query interface), thus allowing $\mathscr{K}\mathscr{B}_v$ to import $\mathscr{K}\mathscr{B}_h$ "by query." Some serious restrictions that preclude the existence of an import-by-query algorithm include: the TBox of $\mathscr{K}\mathscr{B}_v$ is not *semantically* modular w.r.t. the shared signature; the presence of nominals in $\mathscr{K}\mathscr{B}_h$ and (atomic) roles in the shared signature. It is also not discussed how to select the hidden part $\mathscr{K}\mathscr{B}_h$ given a set of target secrets which includes the issue of deciding secondary protection.

Similarly, in [Stouppa and Studer, 2009] only ex-post confidentiality verification methods are provided. In their framework the background knowledge is modeled it as a part of the knowledge base while the equivalent of PKB is the set of all knowledge bases that include a given set of publicly known axioms. Consequently, in some cases their verification method is vulnerable to the attacks to complete knowledge, that are based on more complex (conditional) metaknowledge (cf. Example 3.2.4 and Example 3.2.11) that cannot be encoded in their framework.

In [Cuenca Grau and Horrocks, 2008] Cuenca Grau and Horrocks investigate knowledge confidentiality from a probabilistic perspective: enlarging the public view should not change the probability distribution over the possible answers to a query q that represents the set of secrets. In [Cuenca Grau and Horrocks, 2008] users can query the knowledge base only through a predefined set of views (the approach presented in Section 3.2.1 place no such restriction, instead). As an analogue of our BK, they assume that part of the TBox is visible. A probability distribution P over the set of knowledge bases plays a role similar to metaknowledge. However, their confidentiality condition allows P to be replaced with a different P' after enlarging the public view, so at a closer look P does not really model the user's a priori knowledge about the knowledge base that should remain constant, differently from PKB. The authors consider also ontology updates, however in that case confidentiality is enforced only if updates are restricted

to an analogue of conservative extensions.

The framework adopted throughout the thesis is inspired by the literature on *controlled query evaluation* (CQE).

CQE is a prominent formal framework for confidentiality enforcement. Sensitive information is declaratively specified by means of a confidentiality policy and enforced by a censor: when given a user query, a censor checks whether returning the answer might lead to an information leakage, in which case it returns a distorted answer. Given a user query, in general also dependent on the history and thus on the current view a priori knowledge and the answers to previous queries, the censor computationally check whether returning the answer might lead to an information leakage to determine the need of a distortion. Then, as indicated by the outcomes of the checks, the censor form the answer such that, from the user's point of view, it remains indistinguishable what the correct answer would have been. A censor following the basic refusal approach first checks whether the correct answer could already be inferred from the current view; if this is not the case, then the censor inspects both the query α and its negation $\neg\alpha$: if returning any of them would lead to a direct violation of the confidentiality policy, then the answer is formed by weakening the correct answer into a tautology expressing "tertium non datur" (which is abbreviated by a keyword *mum* and interpreted as a refusal notification). A censor following the basic lying approach only inspects the correct truth evaluation of the query sentence α regarding a stronger violation condition, namely whether the disjunction of all policy elements would be entailed in order to ensure consistent answers. A censor following the basic combined approach — refusal and lying — first inspects the evaluation of the query α; if it would lead to a direct violation then the censor additionally inspects the negation $\neg\alpha$: if also that negation would lead to a violation, then the answer sentence is formed by weakening the correct answer into a tautology (or mum); otherwise the negation is returned as a lie.

The CQE paradigm was first proposed in [Sicherman et al., 1983] by Sichermann et al. and was later studied by Biskup, Bonatti, Kraus and Subrahmanian [Biskup and Bonatti, 2001, Biskup and Bonatti, 2004a, Biskup and Bonatti, 2004b], etc. CQE in the context of incomplete databases was studied by Biskup and Weibert [Biskup and Weibert, 2008]. These foundational works on CQE assume that both the information in the system and user queries are represented in propositional logic. Early works on non-propositional CQE are [Biskup and Bonatti, 2007, Biskup et al., 2010].

The method we presented is based on lies and/or refusals. Technically it uses *lies*, because rejected queries are not explicitly marked by the special answer *mum*. However, the censor resembles the classical refusal censor, so the properties of $f_{\mathscr{M}}$ are not

subsumed by any of the basic CQE approaches. For instance, unlike the CQE methods that use lies, $f_{\mathscr{M}}(KB, u)$ encodes only correct knowledge (that is entailed by $\mathscr{KB}$), and it is secure whenever users do not initially know any secret while lies-based CQE further require that no *disjunction* of secrets should be known a priori. Unlike the refusal method, $f_{\mathscr{M}}$ can handle *cover stories* because users are not told that some queries are obfuscated. As an additional advantage, the method needs not to adapt existing engines to handle nonstandard answers like *mum*. Finally, the CQE approaches do not deal specifically with DL knowledge bases, metaknowledge, and related complexity analysis. For an overview see [Biskup, 2016].

More recently CQE for ontologies has been studied in [Cuenca Grau et al., 2013, Grau et al., 2014, Grau et al., 2015]. The framework is adaptation of the existing work on CQE for incomplete databases based on a novel class of censors, called view-definable. Background knowledge is formalized as an OWL 2 RL ontology and assumed to be fully known to all users whereas a dataset formalized as a set of concept and role assertions is assumed to be hidden. A confidentiality policy is represented as a set of assertions logically entailed by the ontology and the dataset. Users access the system by means of restricted query interface that allows to formulate arbitrary conjunctive queries.

The main idea behind view-defined censors, is to modify the dataset by anonymizing occurrences of constants and adding or removing facts, whenever needed[27]. Such modified dataset constitutes an (anonymization) view that encodes the information in the system relevant to the censor's output for any user query. The authors adopt the basic case of the CQE paradigm where the censor only filters out answers that could lead to a policy violation. The focus is on optimal censors, which maximize answers to queries while ensuring confidentiality of the policy. However optimal view-based censors are not guaranteed to exist since the optimality requirement may lead to infinite views, even for $\mathscr{EL}$ and ontologies. In [Cuenca Grau et al., 2013] Cuenca Grau at. all identify a guarded fragment of OWL 2 RL for which these limitations can be circumvented.

In [Grau et al., 2014] the framework is further extended to arbitrary CQs as policies rather than plain concept and role assertions and a novel class of censors, called obstruction censors. Obstruction censors are defined by a set of *forbidden query patterns* where all answers instantiating such patterns should not be disclosed to users. Obstruction censors do not require data modification and are well-suited for applications such as OBDA, where data is managed by an RDBMS.

[27] View-defined censors may also require materialization of implicit data, and hence are well-suited for applications where materialization is feasible; the approach presented in Section 3.2.1 does not impose similar restrictions.

In [Grau et al., 2015] the authors compare the expressive power of obstruction with that of view censors and establish their theoretical limitations. In particular, determining the existence of an optimal view is undecidable even for Datalog ontologies. On the other hand, computing obstructions realizing optimal views for linear Datalog and OWL 2 QL ontologies is possible in polynomial time.

The framework presented by Cuenca Grau et. al. in [Cuenca Grau et al., 2013, Grau et al., 2014, Grau et al., 2015] can be seen as complementary to the approach adopted in this thesis. In particular, access to external sources of background or meta-knowledge knowledge is not taken in consideration. Consequently, their verification method is vulnerable to the attacks presented in Section 3.2.

In [Grau and Kostylev, 2016] Cuenca Grau and Kostylev define notions of safe and optimal anonymizations of RDF graphs for privacy-preserving data publishing (PPDP)[28] of Linked Data. In this context safety ensures that the anonymized data can be published with provable protection guarantees against linking attacks, whereas optimality ensures that it preserves as much information from the original data as possible, while satisfying the safety requirement. An anonymized RDF graph G can be obtained from the original graph G_0 by replacing some occurrences of IRIs in triples with blank nodes. The sensitive information in G_0 (referred to as a policy) is represented by a SPARQL query. Policy compliance ensures that the sensitive information remains protected when the anonymized data is considered locally. However, It provides no guarantee against disclosure once the anonymized data is released on the Web and can be linked with arbitrary external sources.

To address this limitation an additional safety requirement that take account of the dataset union, i.e. the merge of G with external graphs, is defined to ensure that G can be released with provable protection guarantees against linkage attacks. This approach is similar to how background knowledge is treated in [Bonatti and Sauro, 2013]. In order to deal with situations when a dataset contains relations for which a smart attacker could easily gather complete information (cf Example 3.2.4 and Example 3.2.11), the policy compliance is evaluated under closed-world semantics as well.

Although, the framework appears to provide safety guarantees against linking attacks and attacks to complete knowledge, it does not yet capture OWL 2 ontologies, which are extensively used in applications to enrich the semantics of RDF graphs. The introduction of such ontologies seem to lead to significant technical challenges, especially in combination with closed-world semantics [Grau and Kostylev, 2016].

[28] PPDP refers to the problem of protecting individual privacy against disclosure while at the same time ensuring that published dataset remains practically useful for analysis.

Chapter 4

Optimizing the Computation of Overriding in DLs

In the previous chapter we introduced a new family of nonmonotonic Description Logics (DLs), which supports *normality concepts* NC to denote the normal/ prototypical instances of a concept C, and prioritized *defeasible inclusions* (DIs) $C \sqsubseteq_n D$ that mean (roughly speaking): "*by default, the instances of* C *satisfy* D*, unless stated otherwise*", that is, unless some higher priority axioms entail $C \sqcap \neg D$; in that case, $C \sqsubseteq_n D$ is *overridden*. The prototypical instances of C are required to satisfy all the DIs that are not overridden in C.

Given the negligible number of applications based on nonmonotonic logics deployed so far, $\mathscr{DL}^{\mathrm{N}}$ has been designed to address real-world problems and concrete knowledge engineering needs. Fortunately, at least in the biomedical domain, the literature contains several extensive discussions of such needs and how nonmonotonic reasoning may address them [Rector, 2004, Stevens et al., 2007]. A discussion of how nonmonotonic reasoning may address needs in (semantic web) policy formulation can be found in [Woo and Lam, 1993]. As we have already seen in Section 3.1 distinguishing features are: (*i*) $\mathscr{DL}^{\mathrm{N}}$ adopts the simplest possible criterion for overriding, that is, inconsistency with higher priority axioms; (*ii*) all the normal instances of a concept C conform to the same set of default properties, sometimes called *prototype*; (*iii*) the conflicts between DIs that cannot be resolved with priorities are regarded as knowledge representation

Table 4.1. Partial comparison with other nonmonotonic DL

Features	CIRC	DEF	AEL	TYP	RAT		PR	$\mathscr{DL}^{\mathrm{N}}$
no inheritance blocking	✓	✓	✓			✓	✓	✓
no CWA effects		✓	✓		✓	✓		✓
fine-grained control on role ranges				smtm				✓
detects inconsistent prototypes				smtm			✓	✓
preserves tractability								✓(*)

(*) It holds for subsumption, assertion checking, concept consistency, KB consistency.

errors and are to be fixed by the knowledge engineer (typically, by adding specific DIs). No traditional nonmonotonic logic satisfies (*i*), and very few satisfy (*ii*) or (*iii*). $\mathscr{DL}^{\mathrm{N}}$ behaves very well on applicative examples due to the following consequences of (*i*)–(*iii*) (a comparison with other nonmonotonic DLs with respect to these features is summarized in Table 4.1):[1]

No inheritance blocking: In several nonmonotonic logics a concept with exceptional properties inherits *none* of the default properties of its superclasses. This phenomenon is known as *inheritance blocking*.

Undesired Closed World Assumption effects: In some nonmonotonic DLs, an exceptional concept is shrinked to the individuals that explicitly belong to it; it may possibly become inconsistent.

Control on role ranges: Unlike most nonmonotonic DLs, $\mathscr{DL}^{\mathrm{N}}$ axioms can specify whether a role should range only over normal individuals or not.

Detect inconsistent prototypes: $\mathscr{DL}^{\mathrm{N}}$ facilitates the identification of all conflicts that cannot be resolved with priorities (via consistency checks over normality concepts), because their correct resolution is application dependent and should require human intervention.

Besides solving the above issues, $\mathscr{DL}^{\mathrm{N}}$ is the first nonmonotonic DL known to preserve the tractability of low-complexity DLs such as $\mathscr{EL}^{++}$ and *DL-lite* (underlying the OWL2-EL and OWL2-QL profiles). This opens the way to processing very large nonmonotonic KBs within these fragments.

The attractiveness of the $\mathscr{EL}$ family in the context of this thesis is twofold: on the one hand subsumption is decidable in polynomial time; on the other hand, its expressive power is sufficient for many important applications of ontologies. In particular,

[1] The abbreviations CIRC, DEF, AEL, TYP, RAT and PR stand respectively for circumscribed, default, autoepistemic DLs, DLs of typicality, rational closure and probabilistic nonmonotonic DLs.

$\mathscr{EL}$ ++ is well-suited for the design of life science ontologies, and many of todays largest ontologies are formulated in this language. Examples include the Systematized Nomenclature of Medicine, Clinical Terms (SNOMED CT), the Gene Ontology that can be seen as an acyclic $\mathscr{EL}$ TBox with one transitive role and large parts of the Galen Medical Knowledge Base (Galen) and many Open Biomedical Ontologies (OBO), e.g. the Chemical Entities of Biological Interest (ChEBI), the e-Mouse Atlas Project (EMAP), the Foundational Model of Anatomy (FMA), the Fly Anatomy, and the Molecule Role ontology to name a few. An emphasis is due on SNOMED CT which comprises about four hundred thousand axioms and is the standardized clinical terminology adopted by health care sectors in several countries (cf. Section 5.1.2).

Given the massive size of these ontologies, it is mandatory that reasoning in nonmonotonic DLs be extremely efficient. Unfortunately, asymptotic tractability alone, does not suffice for practical purposes. For example, even the classification of the SNOMED CT ontology computed by performing (quadratically many) subsumption tests between every pair of its 300 000 concepts will take an estimated 25 000 hours (almost 3 years) to compute all subsumptions, assuming that every test takes a constant time, say just 1 millisecond. Clearly, reasoning over a nonmonotonic version of SNOMED CT, although still polynomial, is harder and cannot be regarded as practical.

In this chapter we first describe a prototype implementation of $\mathscr{DL}^{\mathrm{N}}$, together with a preliminary, experimental scalability analysis carried out on many large KBs (with more than 20K concept names and over 30K general concept inclusions). Currently there are no "real" knowledge bases encoded in a nonmonotonic DL, because standard DL technology does not support nonmonotonic reasoning. The nonmonotonic KBs encoded in the hybrid rule+DL system DLV-Hex [Drabent et al., 2009] are not suited to our purposes because they do not feature default inheritance due to a restriction of the language: DL predicates cannot occur in rule heads, so rules cannot be used for encoding default inheritance. A systematic approach to transform selected classical subsumptions into defeasible in existing ontologies is provided in [Casini et al., 2015]. The approach relies on the presence of unsatisfiable classes occurring on the left hand side of GCIs. The set of defeasible axioms for each ontology correspond precisely to the set of axioms that "cause" the unsatisfiability of each of these classes. A critical observation made by the authors is that incoherence in classical ontologies is usually the result of erroneous modelling. Given the large emphasis placed on debugging incoherence in the last decade, the number of unsatisfiable classes that elude a debugging phase is expected to be rather low.[2] As a consequence the nonmonotonic part of the modified ontologies

[2] An evidence of the truth of this assumption is provided by the average ratio of resulting

results quite small and can hardly permit an in depth study of the overhead introduced by nonmonotonic reasoning which. Then synthetic test cases are the only choice for evaluating our algorithms. Test suites are obtained by suitably modifying two large biomedical ontologies: the Gene Ontology and Fly Anatomy. From a methodological point of view, we make an effort to validate the test case generator by analyzing the structure of the synthetic knowledge bases and their classification.

As we shall see, although the preliminary results are promising, (so far, no other implemented nonmonotonic logic has been tested on KBs of this size with comparable results); still, as the amount of defeasible inclusions increase query response time raises enough to call for improvements. In the rest of the chapter, we study different optimization techniques to improve $\mathscr{DL}^{\mathrm{N}}$ query response time:

1. Many of the axioms in a large KB are expected to be irrelevant to the given query. We investigate the use of *module extractors* [Martin-Recuerda and Walther, 2014, Sattler et al., 2009] to focus reasoning on relevant axioms only. Note that module extractors are unsound for most nonmonotonic logics, including circumscription, default and autoepistemic logics.
2. We introduce a new algorithm for query answering, that is expected to exploit incremental reasoners at their best. Incremental reasoning is crucial as $\mathscr{DL}^{\mathrm{N}}$'s reasoning method iterates consistency tests on a set of KBs with large intersections. While the assertion of new axioms is processed very efficiently, the computational cost of axiom deletion is generally not negligible. We introduce an *optimistic reasoning method* that is expected to reduce the number of deletions.

We further contribute to the research on module extraction by improving it over some problematic cases. More precisely, in Section 4.2.3 and 4.2.4 we introduce two optimization methods that are not specific to $\mathscr{DL}^{\mathrm{N}}$ and apply also to classical DL reasoning:

- a new module extraction algorithm that discards significantly more axioms in the presence of nonempty ABoxes. This method is correct under the assumption that the knowledge base is consistent; this hypothesis, in practice, is compatible with some of the main intended uses of module extraction, such as importing selected parts of already validated knowledge bases.
- parallel implementation of module extraction.

We prove the correctness of the new algorithms and evaluate the effectiveness of all optimizations experimentally. Code and data can be found on: goo.gl/2UUgrr.

defeasible axioms that is reported to be of 8%, while the median ratio is only 1.5%.

4.1 Preliminary Experimental Analysis

This section introduces NMReasoner, a prototypical implementation of $\mathscr{DL}^{\mathrm{N}}$ based on existing classical reasoners. A preliminary experimental performance analysis of this prototype is included; it uses test cases with realistic size and the optimization techniques supported by the underlying, classical reasoning engine. For the purpose, synthetic test cases have been automatically generated in a principled way, as explained in Section 4.1.2. Currently, no "real" nonmonotonic DL knowledge bases exist, since mainstream DL technology does not support nonmonotonic inferences, and the available implementations of nononotonic DLs can only handle knowledge bases with moderate size.

4.1.1 NMReasoner

According to the theoretical framework, the engine consists of two modules. The first one, hereafter called *translation module*, constructs the classical knowledge base $\mathscr{KB}^{\Sigma}$ corresponding to the given $\mathscr{DL}^{\mathrm{N}}$ knowledge base $\mathscr{KB} = \mathscr{S} \cup \mathscr{D}$. The second module computes nonmonotonic subsumptions (sometimes called *queries* in the following). In NMReasoner, Σ is the set of all normality concepts that occur either in $\mathscr{KB}$ or in a given set of queries Q. Moreover, if no priority relation over DIs is provided in input (encoded in an appropriate file), then specificity (3.1) is applied by default. Both modules call an external classical reasoner for classification. For knowledge bases belonging to the $\mathscr{EL}$ family of description logics we chose ELK [Kazakov et al., 2012, Kazakov et al., 2014, Kazakov and Klinov, 2013], a particularly efficient, specialized engine. In this section we use a version of NMReasoner that adopts no optimization technique besides those natively supported by the underlying classical reasoner.

4.1.2 The Test Case Generator

We pursued two different approaches: (i) injecting fully synthetic random defeasible inclusions in a given real world ontology; (ii) transforming a random set of strong concept inclusions of a real world ontology into defeasible inclusions. Both approaches have been applied to a version of the Gene Ontology[3] (GO for short) published in 2006 and Fly Anatomy (FLY)[4], that has been extensively used in many performance experiments

[3] http://www.geneontology.org

[4] One of the largest ontologies listed at the OBO Foundry websites http://www.obofoundry.org/.

[Baader et al., 2006, Delaitre and Kazakov, 2009, Mendez and Suntisrivaraporn, 2009, Glimm et al., 2012, Kazakov, 2009, Sertkaya, 2011, Tsarkov et al., 2007].

Both ontologies are suitable for our purposes because of its size and domain (that fit our application scenarios): they are large biomedical ontologies with GO featuring 20465 atomic concepts and 28896 concept inclusions and FLY 7797 atomic concepts and 19137 concept inclusions. They can be encoded in $\mathscr{EL}$ ++, whose nonmonotonic version $\mathscr{EL}$ $++^{\mathrm{N}}$ (as proved in Section 3.1.3) enjoys tractable inference problems.

Under approach (i), that is, random DI injection, given a classical TBox $\mathscr{S}$, the set of DIs $\mathscr{D}$ is generated as follows. First, the size of $\mathscr{D}$ is determined by a parameter *Synthetic-DI-rate* as the ratio between the number of DIs and the number of CIs in $\mathscr{S}$. Then, iteratively, two atomic concepts A and B, and optionally a role R are randomly chosen from the signature of $\mathscr{S}$, and either $A \sqsubseteq_n B$ or $A \sqsubseteq_n \exists R.B$ is added to $\mathscr{D}$. The generator makes sure that no duplicates are generated, and that for each new DI δ, $\mathsf{pre}(\delta) \sqsubseteq \mathsf{con}(\delta)$ is not classically entailed by $\mathscr{S}$.

Under approach (ii), given a classical TBox $\mathscr{S}$, a set of concept inclusions $S' \subseteq \mathscr{S}$ is randomly chosen and turned into $\mathscr{D}$. The size of S' (and $\mathscr{D}$) is determined by a parameter *CI-to-DI-rate* that specifies the ratio $|S'|/|\mathscr{S}|$. Then all inclusions $C_1 \sqsubseteq C_2 \in S'$ are removed from $\mathscr{S}$ and the corresponding DIs $C_1 \sqsubseteq_n C_2$ are added to $\mathscr{D}$ (here C_1 and C_2 may be compound concepts, in general). The priority relation over $\mathscr{D}$ is specificity, as determined by the logical consequences of $\mathscr{S}$ before removing the inclusions in S', thus preserving as much as possible the semantics of the original relations encoded in the ontology. The intended effect is a progressive transformation of classical knowledge bases into purely defeasible knowledge bases, like those extensively adopted in the literature on preferential and rational closures.

Under both approaches, in order to increase the probability of overriding (and hence nonmonotonic behavior), some additional inconsistencies between DI conclusions can be injected in the ontology. To do that, DI pairs (δ_1, δ_2) are randomly selected from $\mathscr{D}$. For each such pair, two arbitrary concepts C_1 and C_2 are picked from the superclasses of $\mathsf{con}(\delta_1)$ and $\mathsf{con}(\delta_2)$ (respectively), and a new disjointness axiom $C_1 \sqcap C_2 \sqsubseteq \bot$ is added to $\mathscr{S}$. The test case generator makes sure that none of C_1, C_2, and $\mathscr{S}$ are made inconsistent, by checking that the following conditions are satisfied in the extended knowledge base: (i) $\mathsf{con}(\delta_2) \not\sqsubseteq C_1$; (ii) $\mathsf{con}(\delta_1) \not\sqsubseteq C_2$; (iii) $C_2 \not\sqsubseteq C_1$ and (iv) $C_1 \not\sqsubseteq C_2$. Note that we intentionally refrain from asserting $\mathsf{con}(\delta_1) \sqcap \mathsf{con}(\delta_2) \sqsubseteq \bot$ directly, so that the reasoning involved in checking whether a DI is overridden is generally nontrivial. The generation of disjointness axioms is controlled by parameter *DA-rate*, that specifies the ratio between the number of new disjointness axioms and $|\mathscr{S}|$.

Parameter	Meaning
Synthetic-DI-rate	percentage of fully synthetic DIs w.r.t. the number of CIs
CI-to-DI-rate	percentage of CIs to be transformed DIs
DA-rate	percentage of disjointness axioms w.r.t. the number of CIs
I-rate	percentage of new distinct individuals
ABox-rate	percentage of assertions w.r.t. the number of CIs
R-rate	percentage of role assertions w.r.t. the number of all assertions
NC-rate	percentage of DIs with normality concept within the scope of quantifiers

Figure 4.1. The main parameters of the test-case generator

The above test sets can be extended by adding random ABoxes to the nonmonotonic versions of GO and FLY. To do that, a number of new distinct individuals is introduced guided by a parameter *I-rate* as a ratio of $|S|$. The size of the ABox $\mathscr{A}$ is then determined by a parameter *ABox-rate* as the ratio $|\mathscr{A}|/|\mathscr{S}|$. Finally, the amount of role assertions in the ABox is controlled by a parameter *R-rate* specifying the ratio of role assertions to $|\mathscr{A}|$. Then, iteratively, two individuals a and b, two atomic concepts A and B, and a role R are randomly chosen from the signature of $\mathscr{S}$ and $\mathscr{A}$, and the assertions $A(a)$, $B(b)$ and $R(a,b)$ are added to $\mathscr{A}$. The generator makes sure that no duplicates are generated, A, B, and $\mathscr{S} \cup \mathscr{A}$ are still classically consistent, and that each new assertion is not already entailed by $\mathscr{S} \cup \mathscr{A}$. When the required amount of role assertions is reached, the generation keep on introducing random class assertions only until the desired total number of assertions is injected.

The above test sets are N-free. A new set of experiments can be generated by randomly introducing normality concepts in DIs, within the scope of quantifiers.[5] Specifically, $\exists R.C$ is transformed into $\exists R.\mathrm{N}\, C$. Their amount is controlled by a parameter *NC-rate* specifying the ratio of modified DIs to DIs with quantifiers in $\mathscr{D}$.

We estimate that the values of $|\Sigma|$ considered for the generation (in the range between 50 and 250) are larger than what should be expected in practice, given the specific role of explicit normality concepts, cf. footnote 5. Such values are also much larger than in N-free experiments, where $|\Sigma|$ is bounded by the query size.

[5] So far, all the application examples that are not N-free satisfy this restriction, as apparently the only purpose of explicit normality concepts is restricting default role ranges to normal individuals, cf. Example 3.1.36 and the nomonotonic design pattern in Section 3.1.6.

4.1.3 Experimental Results: Test Case Structure

A first analysis has been aimed at checking that synthetic test cases are not trivial. For this purpose we inspected the structure of the DIs that *actually apply* to each normality concept NC, which means that the DI's left-hand side subsumes C. In particular, we measure the height of the priority hierarchy of applicable DIs, and how many applicable DIs are overridden; the former quantity is related to the potential levels of overriding, while the latter is more directly related to the nonmonotonic behavior that actually occurs.

For all figures, every single reported value is obtained as the average over ten different non monotonic ontologies and fifty different queries on each ontology, each of which involved the construction of a different translation $\mathscr{K}\mathscr{B}^{\Sigma}$ (as Σ depends on the query).

Figure 4.2 and Figure 4.4 are devoted to generation approach (i). As the DA rate grows, the figures report the percentage of overridden applicable DIs, plus the average and maximum height of the priority hierarchy of applicable DIs. Figure 4.3 and Figure 4.5 report the same values for generation approach (ii).

These figures are reasonable, given GO's and FLY's structure. The length of the longest path in GO's classification (i.e. the maximum possible hierarchy depth) is 15, and the average length 3.66. In the experiments concerning DIs obtained from strong CIs (see Figure 4.3) the average (resp. maximal) depth of the applicable DIs hierarchy range between 41% and 55% of the average (resp. 26,6% and 40% of the maximal) path length in the original ontology, coherently with the random placement of normality concepts in the hierarchy.

For fully synthetic DIs (Figure 4.2) these values are lower and vary between 34% and 42% of the average length, and between 20% and 33,3% of the maximum length.

As expected, in Figure 4.3, each increment of the DA-rate causes an increment of the percentage of overridden DIs. This relation is less evident in Figure 4.2, probably due to the further randomness introduced by synthetic DI generation.

Similar considerations can be made for FLY based on Figure 4.4 and Figure 4.5. The length of the longest path in FLY's classification is 16, and the average length 2.39. The depth of the DI priority hierarchy (which is related to the interference between different DIs and, indirectly, the number of exception levels) ranges between 20% and 50% of the depth of FLY's taxonomy. In particular, there is a reasonable amount of overriding: the percentage of overridden DIs ranges from 34.44% to 66.68%, depending on the amount of disjointness axioms (hence DI conflicts) contained in the KB.

DA-rate	Overridden/Applicable DIs	Avg. Appl. DI Hierarchy Depth	Max Appl. DI Hierarchy Depth
5%	83,55%	1,34	4
10%	84,13%	1,4	4
15%	91,17%	1,25	3
20%	91,47%	1,3	4
25%	87,64%	1,5	4
30%	88,33%	1,44	5

Figure 4.2. Values characterizing the experiments with variable DA-rate (Synthetic-DI-rate=15%) in GO

DA-rate	Overridden/Applicable DIs	Avg. Appl. DI Hierarchy Depth	Max Appl. DI Hierarchy Depth
5%	61,86%	2,03	6
10%	70,71%	1,96	6
15%	68,45%	2,01	5
20%	75,63%	1,71	5
25%	80,28%	1,54	4
30%	81,27%	1,55	6

Figure 4.3. Values characterizing the experiments with variable DA-rate (CI-to-DI-rate=15%) in GO

DA-rate	Overridden/Applicable DIs	Avg. Appl. DI Hierarchy Depth	Max Appl. DI Hierarchy Depth
5%	34,44%	6,26	13
10%	42,6%	6,6	12
15%	39,1%	6,55	11
20%	46,4%	6,44	12
25%	37,53%	6,49	12
30%	48,11%	6,38	12

Figure 4.4. Values characterizing the experiments with variable DA-rate (Synthetic-DI-rate=15%) in FLY

DA-rate	Overridden/Applicable DIs	Avg. Appl. DI Hierarchy Depth	Max Appl. DI Hierarchy Depth
5%	37,95%	1,17	5
10%	55,56%	0,74	3
15%	57,03%	0,81	4
20%	57,32%	0,95	5
25%	65,98%	0,83	3
30%	66,68%	0,9	4

Figure 4.5. Values characterizing the experiments with variable DA-rate (CI-to-DI-rate=15%) in FLY

CI-to-DI	05%	10%	15%	20%	25%
GO	12.35	24.12	34.47	41.96	49.92
FLY	4.22	7.97	11.95	14.42	17.46

Figure 4.6. Impact of $|\mathscr{D}|$ on performance (sec) – DA rate = 15%

4.1.4 Experimental Setup

The experiments were performed on an Intel i7-2630QM 2GHz machine with 18GB RAM and Ubuntu 12.04.2 LTS. NMReasoner was run on Java 1.8 with the options *-Xms12G -Xmx12G -Xss4G* to set the available RAM to 12GB and the stack memory space to 4GB.

All test cases are modifications of GO and FLY according to approach (i) or (ii). Like in the previous section, every single reported value is obtained as the average execution time over ten different non monotonic ontologies and fifty different queries on each ontology, each of which involved the construction of a different translation $\mathscr{KB}^{\Sigma}$. The varying parameters are: CI-to-DI-rate, Synthetic-DI-rate, DA-rate, ABox-rate, I-rate, R-rate and NC-rate.

4.1.5 Experimental Results: Performance Analysis

Figures 4.7 and 4.6 report the execution time of the translation stage as the amount of DIs grows. In Figure 4.6, CI-to-DI-rate (the percentage of CIs that are transformed into DIs) ranges from 5% to 25%, Synthetic-DI-rate is fixed to 0% (no fully synthetic DIs are generated) and DA-rate to 15%. In Figure 4.7, Synthetic-DI-rate (the percentage

Synthetic-DI	05%	10%	15%	20%	25%
GO	13.15	27.77	37.47	46.11	57.14
FLY	4.86	9.86	14.75	19.86	24.27

Figure 4.7. Impact of $|\mathscr{D}|$ on performance (sec) – DA rate = 15%

DA	05%	10%	15%	20%	25%	30%
GO	27.38	27.52	34.47	38.57	36.21	42.37
FLY	10.31	11.46	11.95	12.13	12.65	13.79

Figure 4.8. Impact of DAs on performance (sec) – CI-to-DI-rate = 15%

of fully synthetic DIs) ranges from 5% to 25%, CI-to-DI-rate is fixed to 0% and DA-rate to 15%. In both cases, translation time increases linearly with the size of $\mathscr{D}$, in accordance to the linear increase in the number of classification problems that must be solved to compute $\mathscr{K}\mathscr{B}^{\Sigma}$. Translation is slightly faster over the test cases produced with approach (i), probably because of the less complex structure of applicable defaults (cf. Figures 4.2, 4.3, 4.4 and 4.5). Values of standard deviation do not exceed 2.36% in Figure 4.6 and respectively 3.74% in Figure 4.7.

Figures 4.8 and 4.9 show the impact of disjointness axioms on the performance of the translation phase. In Figure 4.8, CI-to-DI-rate is fixed to 15% and Synthetic-DI-rate to 0%, while in the Figure 4.9 these values are switched. Translation time is obviously affected by the additional strong disjointness axioms added to the ontology. However, execution time grows less steeply and with less uniform derivative. This is even more evident considering the standard deviation that ranges within 6.91% in the upper part of Tables 4.8 and 4.9 and respectively 1.34% in the lower. As in the previous graphs, fully synthetic DIs yield slightly longer execution times than approach (ii).

The experimental results reported in Figures 4.10 and 4.11 confirm the negative

DA	05%	10%	15%	20%	25%	30%
GO	33.34	35.63	37.47	42.05	43.03	47.03
FLY	12.93	14.01	14.76	15.94	16.60	17.69

Figure 4.9. Impact of DAs on performance (sec) – Synthetic-DI-rate = 15%

impact of nonempty ABoxes on the performance of the translation phase. Adding random assertions to the nonmonotonic versions of FLY and GO make the translation time for CI-to-DI-rate = 15% and DA-rate = 15% raise more than 3 (and up to 10) times (cf. the central column of Figure 4.8). As expected, effectiveness decreases when the ABox is more "interconnected", i.e. when the amount of role assertions increases (role assertions tend to introduce more dependencies, because pairs of individuals are always involved).

ABox size	role assrt.	individuals ∼ 5000	∼ 10000	∼ 20000
∼5000	10%	95.99	104.90	97.33
	20%	99.70	123.41	109.67
	30%	112.75	123.60	132.67
∼10000	10%	123.98	145.46	153.94
	20%	126.24	139.45	152.48
	30%	129.74	145.25	166.24
∼20000	10%	133320	126.59	124.55
	20%	135079	128.50	126.55
	30%	138244	129.33	129.02

Figure 4.10. Impact of ABox on performance (sec) in GO – CI-to-DI-rate = 15% DA-rate = 15%.

ABox size	role assrt.	individuals ∼ 2000	∼ 4000	∼ 8000
∼2000	10%	27.32	30.46	30.68
	20%	29.55	31.45	32.15
	30%	28.29	31.96	32.99
∼4000	10%	35.42	45.03	49.83
	20%	37.47	45.24	52.22
	30%	41.49	51.37	92.09
∼8000	10%	47.01	65.11	71.49
	20%	48.39	68.39	74.93
	30%	60.57	99.65	92.58

Figure 4.11. Impact of ABox on performance (sec) in FLY – CI-to-DI-rate = 15% DA-rate = 15%.

$\|\Sigma\|$	50	100	150	200	250
CI-to-DI					
GO	>30 min.	>30 min.	>30 min.	>30 min.	>30 min.
FLY	>30 min.	>30 min.	>30 min.	>30 min.	>30 min.
Synthetic-DI					
GO	>30 min.	>30 min.	>30 min.	>30 min.	>30 min.
FLY	>30 min.	>30 min.	>30 min.	>30 min.	>30 min.

Figure 4.12. Non N-free tests. Impact of normal roles ranges (sec) – DI rate = 25% DA rate = 15%.

The above test sets are N-free. The response times of the naive algorithm under priority (3.1) for increasing values of $|\Sigma|$ (that is directly related to the amount of normality concepts occurring in $\mathscr{K}\mathscr{B}$) are listed in Table 4.12 . Unfortunately, in all cases, the naive algorithm exceeded 30 min. timeout.

The substantial similarity between the results for generation approach (i) and those for approach (ii) suggests that our test generation methods do not introduce any significant bias as far as translation phase scalability is concerned.

The overhead of the second phase (subsumption checking) is negligible, because the translation phase constructs a classification of $\mathscr{K}\mathscr{B}^{\Sigma}$ as a byproduct, and subsumption checking consists of a simple search in the classification graph. In all tests, query evaluation time do not exceed 4 milliseconds. Detailed data are not reported, as the effects of increasing the number of DIs is dominated by statistical fluctuations.

As a term of comparison, a single classification of the original GO takes approximately 0.4 seconds. The current translation time is significantly higher due to the large number of classifications required for computing $\mathscr{K}\mathscr{B}^{\Sigma}$. Note, however, that translation time is compatible with off-line pre-computation of $\mathscr{K}\mathscr{B}^{\Sigma}$ for a suitable Σ, covering the normality concepts that are expected to occur in the queries.

Incremental reasoning algorithms are quite effective in reducing translation time. We tried ELK's incremental reasoning facility [Kazakov and Klinov, 2013], supported since distribution 0.4.0. Without incremental reasoning, the translation time for CI-to-DI-rate = 15% and DA-rate = 15% raises more than 5 times. The preliminary results obtained in this section for the performance of $\mathscr{DL}^{\mathrm{N}}$ inference are promising; still, as defeasible inclusions approach 25% of the KB, an ABox is added and for non N-free KBs query response time slows down enough to call for improvements. In the rest of the chapter, we study optimization techniques to improve DLN query response time.

4.2 Improving Module Extraction for Nonmonotonic and Classical DLs

Module extraction algorithms, e.g. [Sattler et al., 2009, Grau et al., 2008, Martin-Recuerda and Walther, 2014], can quickly select a subset of a given Description Logic (DL) knowledge base that suffices to answer any query formulated in a given signature of interest.

Roughly speaking, the problem of module extraction can be expressed as follows: given a reference vocabulary Sig, a module is a (possibly minimal) subset $\mathscr{M} \subseteq \mathscr{K}\mathscr{B}$ that is relevant for Sig in the sense that it preserves the consequences of $\mathscr{K}\mathscr{B}$ that contain only terms in Sig.

The interest in module extraction techniques is motivated by several ontology engineering needs. We are interested in modularization as an optimization technique for querying large ontologies: the query is evaluated on a (hopefully much smaller) module of the ontology that preserves the query result (as well as any inference whose signature is contained in the query's signature).

A $\mathscr{K}\mathscr{B}$ is said to be a conservative extension (CE) of $\mathscr{K}\mathscr{B}'$ if all consequences of $\mathscr{K}\mathscr{B}$ that can be expressed over Sig are also consequences of $\mathscr{K}\mathscr{B}'$. This logic-based approach is theoretically sound and provides a desirable guarantee: reusing only terms from Sig, it is not possible to distinguish between querying $\mathscr{K}\mathscr{B}'$ and $\mathscr{K}\mathscr{B}$. However, the problem of deciding whether two knowledge bases entail the same axioms over a given signature is usually harder than standard reasoning tasks. Consequently deciding whether $\mathscr{K}\mathscr{B}'$ is a CE of $\mathscr{K}\mathscr{B}$ (for Sig) is computationally expensive in general. For example, $DL\text{–}Lite_{horn}$ complexity grows from PTIME to coNP-TIME-complete [Kontchakov et al., 2008]; for $\mathscr{ALC}$, complexity is one exponential harder [Ghilardi et al., 2006], while for $\mathscr{ALCQIO}$ the problem becomes even undecidable [Lutz et al., 2007].

In order to achieve a practical solution, a syntactic approximation has been adopted in [Sattler et al., 2009, Grau et al., 2008]. The corrisponding algorithm $\top\bot^*$-Mod($Sig, \mathscr{K}\mathscr{B}$) is defined in [Sattler et al., 2009, Def. 4] and reported in Algorithm 1 below. It is based on the property of $\bot$-locality and $\top$-locality of single axioms (line 15). An axiom is local w.r.t. Sig if the substitution of all non-Sig terms with $\bot$ (resp. $\top$) turns it into a tautology.

The module extractor identifies a subset $\mathscr{M} \subseteq \mathscr{K}\mathscr{B}$ of the knowledge base and a signature Sig (containing all symbols of interest) such that all axioms in $\mathscr{K}\mathscr{B} \setminus \mathscr{M}$ are local w.r.t. Sig. This guarantees that every model of $\mathscr{M}$ can be extended to a

Algorithm 1: $\top\bot^*$-Mod($Sig, \mathcal{KB}$)

Input: Ontology $\mathcal{KB}$, signature Sig
Output: $\top\bot^*$-module $\mathscr{M}$ of $\mathcal{KB}$ w.r.t. Sig

```
   // main
 1 begin
 2     M := KB
 3     repeat
 4         M' := M
 5         M := ⊤-Mod(⊥-Mod(M, Sig), Sig)
 6     until M ≠ M'
 7     return M
 8 end
 9 function x-Mod(KB,Sig)  //   x ∈ {⊥, ⊤}
10 begin
11     M := ∅,   T := KB
12     repeat
13         changed = false
14         forall α ∈ T do
15             if  α is not x-local w.r.t. Sig ∪ M~  then
16                 M := M ∪ {α}
17                 T := T \ {α}
18                 changed = true
19     until changed = false
20     return M
21 end
```

model of $\mathcal{KB}$ by setting each non-Sig term to either $\bot$ or $\top$. Consequently, all queries formulated with symbols in Sig can be answered using only $\mathscr{M}$, instead of the entire $\mathcal{KB}$.

The function x-Mod($Sig, \mathcal{KB}$) (lines 9-20), where x stands for $\top$ or $\bot$, describes the procedure for constructing modules of a knowledge base $\mathcal{KB}$ for each notion of locality. Starting with an empty set of axioms (line 11), iteratively, the axioms α that are non-local are added to the module (line 16) and, in order to preserve soundness, the signature against which locality is checked is extended with the terms in α (line 15). Iteration stops when a fix-point is reached.

Modules based on a single syntactic locality can be further shrinked by iteratively nesting $\top$-extraction into $\bot$-extraction, thus obtaining $\top\bot^*$-Mod($Sig, \mathcal{KB}$) modules; the resulting algorithm is shown in lines 2-7 of Algorithm 1.

In general, module extractors are not correct under nonmonotonic semantics, because x-locality ($x \in \{\bot, \top\}$) is insensitive to the dependencies between predicates introduced by nonmonotonic inference.

4.2.1 Module Extraction for $\mathscr{DL}^{\mathrm{N}}$

The naive construction of $\mathscr{KB}^{\Sigma}$ evaluated in Section 4.1.5 must process all the axioms in $\mathscr{KB}^{\Sigma}_{all} = \mathscr{KB}^{\Sigma}_{0} \cup \{\delta^{\mathrm{N}C} \mid \delta \in \mathscr{D},\ \mathrm{N}C \in \Sigma\}$. Here we optimize $\mathscr{DL}^{\mathrm{N}}$ inference by quickly discarding some of the irrelevant axioms in $\mathscr{KB}^{\Sigma}_{all}$ using the notions of $\top\bot^*$-module and locality [Martin-Recuerda and Walther, 2014, Grau et al., 2008, Sattler et al., 2009], that we extend to DIs as follows.

Definition 4.2.1 (Module, locality) A $\top\bot^*$-*substitution* for $\mathscr{KB}$ and a signature Sig is a substitution σ over $\widetilde{\mathscr{KB}} \setminus Sig$[6] that maps each concept name on $\top$ or $\bot$, and each role name on the universal role or the empty role. A strong axiom α is σ-*local* iff $\sigma(\alpha)$ is a tautology. A DI $C \sqsubseteq_n D$ is σ-local iff $C \sqsubseteq D$ is σ-local. A set of axioms is σ-local if all of its members are. A *(syntactic) module* of $\mathscr{KB}$ with respect to Sig is a set $\mathscr{M} \subseteq \mathscr{KB}$ such that $\mathscr{KB} \setminus \mathscr{M}$ is σ-local for some $\top\bot^*$-substitution σ for $\mathscr{KB}$ and $\widetilde{\mathscr{M}} \cup Sig$.

Let $\mathrm{Mod}_{\mathrm{DI}}(Sig, \mathscr{KB})$ be the variant of the algorithm $\top\bot^*$-Mod$(Sig, \mathscr{KB})$ where the locality condition in line 15 is replaced by the one in Def. 4.2.1 (that applies to DIs as well). Using the original correctness argument for $\top\bot^*$-Mod$(Sig, \mathscr{KB})$, it is easy to see that $\mathrm{Mod}_{\mathrm{DI}}(Sig, \mathscr{KB})$ is a syntactic module of $\mathscr{KB}$ w.r.t. Sig according to Def. 4.2.1.
If $\mathscr{KB}$ contains no DIs, then Def. 4.2.1 is a rephrasing of standard syntactic notions of modules and locality[7], so

$$\text{for all queries } \alpha \text{ such that } \widetilde{\alpha} \subseteq Sig,\ \mathscr{M} \models \alpha \text{ iff } \mathscr{KB} \models \alpha. \tag{4.1}$$

However, proving that $\top\bot^*$-Mod$_{\mathrm{DI}}(Sig, \mathscr{KB})$ is correct for *full* $\mathscr{DL}^{\mathrm{N}}$ is far from obvious: removing axioms from $\mathscr{KB}$ using module extractors is incorrect under most nonmonotonic semantics (including circumscription, default logic and autoepistemic logic). The reason is that nonmonotonic inferences are more powerful than classical

[5] For efficiency, this test is approximated by a matching with a small set of templates.
[6] Both $\widetilde{X}$ and sig(X) denote the signature of X.
[7] In particular, our minimal modules correspond to the $\top\bot^*$-modules of [Sattler et al., 2009].

inferences, and the syntactic locality criterions illustrated above fail to capture some of the dependencies between different symbols.

Example 4.2.2 Given the knowledge base $\{\top \sqsubseteq A \sqcup B\}$ and $Sig = \{A\}$, the module extractor returns an empty module (because by setting $B = \top$ the only axiom in the KB becomes a tautology). The circumscription of this KB, assuming that both A and B are minimized, does not entail $A \sqsubseteq \bot$, while the circumscription of the empty module entails it. ∎

Example 4.2.3 Under the stable model semantics, the normal logic program

$$\begin{aligned} p &\leftarrow \neg p, q \\ q &\leftarrow \neg r \\ r &\leftarrow \neg q \end{aligned}$$

entails r, both credulously and skeptically. The module extractor, given $Sig = \{r\}$, removes the first rule (that becomes a tautology by setting $p =$true). The module does not entail r skeptically anymore, and erroneously entails q credulously. Analogues of this example apply to default and autoepistemic logic, using the usual translations. They can be adapted to the extension of DL based on MKNF [Donini et al., 2002] and default DL [Baader and Hollunder, 1995b]. ∎

Now we illustrate the correct way of applying $\top\bot^*$-$\mathrm{Mod}_{\mathrm{DI}}$ to a $\mathscr{DL}^{\mathrm{N}}$ $\mathscr{KB} = \mathscr{S} \cup \mathscr{D}$ and a query α (subsumption or assertion). Let Σ be the union of $\widetilde{\alpha}$ and the set of normality concepts occurring in $\mathscr{KB}$. Let

$$\mathscr{M}_0 = \mathrm{Mod}_{\mathrm{DI}}(\Sigma, \mathscr{KB} \cup \mathrm{N}\Sigma)\,,$$

where $\mathrm{N}\Sigma$ abbreviates $\{\mathrm{N}C \sqsubseteq C \mid \mathrm{N}C \in \Sigma\}$. Let

$$\mathscr{M} = (\mathscr{KB}_0^{\Sigma} \cap \mathscr{M}_0) \cup \{\delta^{\mathrm{N}C} \mid \delta \in \mathscr{D} \cap \mathscr{M}_0,\ \mathrm{N}C \in \Sigma\}.$$

Example 4.2.4 Let $\mathscr{KB}$ be the knowledge base:

$$A \sqsubseteq B \qquad (4.2) \qquad\qquad B \sqcap C \sqsubseteq A \qquad (4.4)$$

$$A \sqsubseteq_n D \sqcap E \qquad (4.3) \qquad\qquad F \sqsubseteq_n A \qquad (4.5)$$

and α the query $\mathrm{N}A \sqsubseteq D$. $\mathscr{M}_0$ is calculated as follows: first, since no normality concept occurs in $\mathscr{KB}$, Σ is equal to the signature $\widetilde{\alpha} = \{\mathrm{N}A, D\}$.

Algorithm 1 calls first the function $\bot$-Mod($\mathcal{KB} \cup \mathrm{N}\Sigma, \Sigma$). Notice that by replacing C and F with $\bot$, axioms (4.4) and (4.5) becomes a tautology. Consequently, it is easy to see that the returned knowledge base is $\mathcal{KB}' = \{(4.2), (4.3), \mathrm{N}A \sqsubseteq A\}$.

Then, $\top$-Mod is called on $\mathcal{KB}'$ and Σ. Now, replacing B with $\top$ makes $A \sqsubseteq B$ a tautology, so the resulting knowledge base is $\mathcal{KB}'' = \{(4.3), \mathrm{N}A \sqsubseteq A\}$. It is easy to see that a fix point is reached and hence $\mathcal{KB}''$ is returned. ∎

We shall prove that $(\mathcal{KB} \cap \mathcal{M}_0)^\Sigma$ can be used in place of $\mathcal{KB}^\Sigma$ to answer query α. This saves the cost of processing $\mathcal{KB}^\Sigma_{all} \setminus \mathcal{M}$, that usually is even larger than $\mathcal{KB} \setminus \mathcal{M}_0$ because for each DI $\delta \notin \mathcal{M}_0$, all its translations $\delta^{\mathrm{N}C}$ ($\mathrm{N}C \in \Sigma$) are removed from $\mathcal{M}$.

Lemma 4.2.5 *$\mathcal{M}$ is a module of $\mathcal{KB}^\Sigma_{all}$ w.r.t. Σ.*

Proof. Since $\mathrm{Mod}_{\mathrm{DI}}(\cdot,\cdot)$ returns modules, $(\mathcal{KB} \cup \mathrm{N}\Sigma) \setminus \mathcal{M}_0$ is σ-local, for some $\top\bot^*$-substitution σ for $\mathcal{KB} \cup \mathrm{N}\Sigma$ and $\widetilde{\mathcal{M}_0} \cup \Sigma$. So, for all axioms β in $\mathcal{KB}^\Sigma_0 \setminus \mathcal{M}$, β is σ-local (as $\mathcal{KB}^\Sigma_0 \setminus \mathcal{M} \subseteq (\mathcal{KB} \cup \mathrm{N}\Sigma) \setminus \mathcal{M}_0$). Moreover, for all $\beta = \delta^{\mathrm{N}D} \in \{\delta^{\mathrm{N}C} \mid \delta \in \mathcal{D},\ \mathrm{N}C \in \Sigma\} \setminus \mathcal{M}$, it must hold $\delta \in \mathcal{D} \setminus \mathcal{M}_0$ (by construction of $\mathcal{M}$), and hence δ is σ-local. Now note that if $\sigma(E \sqsubseteq F)$ is a tautology, then also $\sigma(\mathrm{N}D \sqcap E \sqsubseteq F)$ is a tautology, therefore the σ-locality of δ implies the σ-locality of $\delta^{\mathrm{N}D}$. It follows that all β in $\mathcal{KB}^\Sigma_{all} \setminus \mathcal{M}$ are σ-local.

Finally, note that $\mathrm{sig}(\mathcal{KB}^\Sigma_{all}) = \mathrm{sig}(\mathcal{KB} \cup \mathrm{N}\Sigma)$ and $\widetilde{\mathcal{M}} \cup \Sigma = \widetilde{\mathcal{M}_0} \cup \Sigma$, therefore σ is also a $\top\bot^*$-substitution for $\mathcal{KB}^\Sigma_{all}$ and $\mathcal{M} \cup \Sigma$. It follows immediately that $\mathcal{M}$ is a module of $\mathcal{KB}^\Sigma_{all}$ w.r.t. Σ. ∎

Lemma 4.2.6 *If $\mathcal{M}$ is a module of $\mathcal{KB}$ w.r.t. a signature Sig and $\mathcal{KB}' \subseteq \mathcal{KB}$, then $\mathcal{KB}' \cap \mathcal{M}$ is a module of $\mathcal{KB}'$ w.r.t. Sig.*

Proof. If $\mathcal{M}$ is a module of $\mathcal{KB}$ w.r.t. Sig, then $\mathcal{KB} \setminus \mathcal{M}$ is σ-local for some $\top\bot^*$-substitution σ for $\mathcal{KB}$ and $\widetilde{\mathcal{M}} \cup Sig$. Let σ' be the restriction of σ to the symbols in

$$\widetilde{\mathcal{KB}'} \setminus (\widetilde{\mathcal{M}} \cup Sig) = \widetilde{\mathcal{KB}'} \setminus (\mathrm{sig}(\mathcal{KB}' \cap \mathcal{M}) \cup Sig).$$

Clearly, σ' is a $\top\bot^*$-substitution for $\mathcal{KB}'$ and $\widetilde{\mathcal{KB}'} \setminus (\mathrm{sig}(\mathcal{KB}' \cap \mathcal{M}) \cup Sig)$. Moreover, for all $\beta \in \mathcal{KB}' \setminus \mathcal{M}$, $\sigma(\beta) = \sigma'(\beta)$, by construction, so $\mathcal{KB}' \setminus \mathcal{M}$ is σ'-local. Then $\mathcal{KB}' \cap \mathcal{M}$ is a syntactic module of $\mathcal{KB}'$ w.r.t. Sig. ∎

The relationship between $(\mathcal{KB} \cap \mathcal{M}_0)^\Sigma$ and $\mathcal{KB}^\Sigma$ is:

Lemma 4.2.7 $\mathcal{KB}^\Sigma \cap \mathcal{M} \subseteq (\mathcal{KB} \cap \mathcal{M}_0)^\Sigma \subseteq \mathcal{KB}^\Sigma$.

Proof. It suffices to prove by induction that for all $i = 0, 1, \ldots, |\mathscr{D}|$,

$$\mathscr{K}\mathscr{B}_i^{\Sigma} \cap \mathscr{M} \subseteq (\mathscr{K}\mathscr{B} \cap \mathscr{M}_0)_i^{\Sigma} \subseteq \mathscr{K}\mathscr{B}_i^{\Sigma}\,.$$

Here the sets $(\mathscr{K}\mathscr{B} \cap \mathscr{M}_0)_i^{\Sigma}$ are defined by replacing $\mathscr{S}$ with $\mathscr{S} \cap \mathscr{M}_0$, and $\mathscr{K}\mathscr{B}$ with $\mathscr{K}\mathscr{B} \cap \mathscr{M}_0$ in (3.24) and (3.25), while δ_i ranges over *all* the DIs in $\mathscr{K}\mathscr{B}$, not just $\mathscr{D} \cap \mathscr{M}_0$. This formulation facilitates the comparison with $\mathscr{K}\mathscr{B}^{\Sigma}$. By the condition in (3.25) , for all $\delta_i \notin \mathscr{M}_0$, $(\mathscr{K}\mathscr{B} \cap \mathscr{M}_0)_i^{\Sigma} = (\mathscr{K}\mathscr{B} \cap \mathscr{M}_0)_{i-1}^{\Sigma}$, so this def. is equivalent to building $(\mathscr{K}\mathscr{B} \cap \mathscr{M}_0)^{\Sigma}$ using only $\mathscr{D} \cap \mathscr{M}_0$.

Base case $(i = 0)$:

$$\begin{aligned} \underline{\mathscr{K}\mathscr{B}_0^{\Sigma} \cap \mathscr{M}} &= (\mathscr{S} \cup \mathrm{N}\Sigma) \cap \mathscr{M} \subseteq (\mathscr{S} \cap \mathscr{M}) \cup \mathrm{N}\Sigma = \\ &= (\mathscr{S} \cap \mathscr{M}_0) \cup \mathrm{N}\Sigma = \underline{(\mathscr{K}\mathscr{B} \cap \mathscr{M}_0)_0^{\Sigma}} \subseteq \\ &\quad \mathscr{S} \cup \mathrm{N}\Sigma = \underline{\mathscr{K}\mathscr{B}_0^{\Sigma}}\,. \end{aligned}$$

Induction step $(i > 0)$: By induction hypothesis (IH)

$$\mathscr{K}\mathscr{B}_{i-1}^{\Sigma} \cap \mathscr{M} \subseteq (\mathscr{K}\mathscr{B} \cap \mathscr{M}_0)_{i-1}^{\Sigma} \subseteq \mathscr{K}\mathscr{B}_{i-1}^{\Sigma}\,.$$

First suppose that $\delta_i \notin \mathscr{M}_0$ (and hence for all NC, $\delta_i^{\mathrm{NC}} \notin \mathscr{M}$). Then $\mathscr{K}\mathscr{B}_i^{\Sigma} \cap \mathscr{M} = \mathscr{K}\mathscr{B}_{i-1}^{\Sigma} \cap \mathscr{M}$, $(\mathscr{K}\mathscr{B} \cap \mathscr{M}_0)_i^{\Sigma} = (\mathscr{K}\mathscr{B} \cap \mathscr{M}_0)_{i-1}^{\Sigma}$, and (by def.) $\mathscr{K}\mathscr{B}_{i-1}^{\Sigma} \subseteq \mathscr{K}\mathscr{B}_i^{\Sigma}$. The Lemma follows by IH.

Next assume that $\delta_i \in \mathscr{M}_0$ and let NC be any normality concept in Σ. Note that $\delta_i^{\mathrm{NC}} \in \mathscr{M}$. By IH,

$$(\mathscr{K}\mathscr{B}_{i-1}^{\Sigma} \cap \mathscr{M}) \downarrow_{\prec\delta_i} \cup\{\delta_i^{\mathrm{NC}}\} \subseteq (\mathscr{K}\mathscr{B} \cap \mathscr{M}_0)_{i-1}^{\Sigma} \downarrow_{\prec\delta_i} \cup\{\delta_i^{\mathrm{NC}}\} \subseteq$$

$$\subseteq \mathscr{K}\mathscr{B}_{i-1}^{\Sigma} \downarrow_{\prec\delta_i} \cup\{\delta_i^{\mathrm{NC}}\}\,.$$

The leftmost term equals $(\mathscr{K}\mathscr{B}_{i-1}^{\Sigma} \downarrow_{\prec\delta_i} \cup\{\delta_i^{\mathrm{NC}}\}) \cap \mathscr{M} \subseteq \mathscr{K}\mathscr{B}_{all}^{\Sigma}$, so by Lemmas 4.2.12 and 4.2.9 and (4.6), the leftmost term entails $\mathrm{NC} \sqsubseteq \bot$ iff the rightmost does. It follows that the middle term $(\mathscr{K}\mathscr{B} \cap \mathscr{M}_0)_{i-1}^{\Sigma} \downarrow_{\prec\delta_i} \cup\{\delta_i^{\mathrm{NC}}\}$ entails $\mathrm{NC} \sqsubseteq \bot$ iff the other two terms do. Then, δ_i^{NC} is added to $(\mathscr{K}\mathscr{B} \cap \mathscr{M}_0)_i^{\Sigma}$ iff δ_i^{NC} belongs to $\mathscr{K}\mathscr{B}_i^{\Sigma} \cap \mathscr{M}$ and $\mathscr{K}\mathscr{B}_i^{\Sigma}$, and the Lemma follows using the IH. ∎

As a consequence, the modularized construction is correct:

Theorem 4.2.8 $(\mathscr{K}\mathscr{B} \cap \mathscr{M}_0)^{\Sigma} \models \alpha$ *iff* $\mathscr{K}\mathscr{B}^{\Sigma} \models \alpha$.

CI-to-DI	05%	10%	15%	20%	25%
GO					
naive	12.35	24.12	34.47	41.96	49.92
mod	00.26	00.28	00.29	00.32	00.34
FLY					
naive	4.22	7.97	11.95	14.42	17.46
mod	00.13	00.15	00.17	00.19	00.21

Table 4.2. Impact of $|\mathscr{D}|$ on performance (sec) – DA rate = 15%

Synthetic-DI	05%	10%	15%	20%	25%
GO					
naive	13.15	27.77	37.47	46.11	57.14
mod	0.48	0.83	1.47	2.76	4.66
FLY					
naive	4.86	9.86	14.75	19.86	24.27
mod	0.40	1.19	2.51	4.61	7.25

Table 4.3. Impact of $|\mathscr{D}|$ on performance (sec) – DA rate = 15%

Proof. By Lemmas 4.2.12 and 4.2.9, and (4.6), $\mathscr{K}\mathscr{B}^{\Sigma} \models \alpha$ iff $\mathscr{K}\mathscr{B}^{\Sigma} \cap \mathscr{M} \models \alpha$. The Theorem then follows by Lemma 4.2.7. ∎

Experimental Analysis

In the following we analyse the performance of the module extraction for $\mathscr{DL}^{\mathrm{N}}$ described in Section 4.2.1 according the experimental setup described in Section 4.1.4. The test suites adopted are those introduced in Section 4.1.2 because they have been proved to be nontrivial w.r.t. a number of structural parameters, including non classical features like exception levels and the amount of overriding.

Tables 4.2 and 4.3 show the impact of the number of DIs on response time for the two test suites, as DI rate ranges from 5% to 25%. The method Mod is slightly less effective in the second suite probably due to some random defaults connecting unrelated parts of the ontology, thereby hindering module extraction. In particular, Mod is on average approximately 87 times faster (max. speedup 125) in the first suite, and 28 times faster in the second (max. speedup 35). The additional conflicts induced by injected disjointness axioms have moderate effects on response time (cf. Table 4.4 and 4.5).

DA	05%	10%	15%	20%	25%	30%
GO						
naive	27.38	27.52	34.47	38.57	36.21	42.37
mod	00.28	00.29	00.29	00.30	00.31	00.32
FLY						
naive	10.31	11.46	11.95	12.13	12.65	13.79
mod	00.16	00.17	00.17	00.18	00.19	00.20

Table 4.4. Impact of DAs on performance (sec) – CI-to-DI-rate = 15%

DA	05%	10%	15%	20%	25%	30%
GO						
naive	33.34	35.63	37.47	42.05	43.03	47.03
mod	1.26	1.35	1.47	1.51	1.60	1.66
FLY						
naive	12.93	14.01	14.76	15.94	16.60	17.69
mod	2.23	2.50	2.52	2.61	2.82	2.98

Table 4.5. Impact of DAs on performance (sec) – Synthetic-DI-rate = 15%

Mod's average response time across both test suites is 0.89 sec. for Gene Ontology and 1.41 sec. for Fly Anatomy, and the longest Mod response time has been 1.66 sec. and 2.98 sec. respectively. As a term of comparison, a single classification of the original GO and FLY takes approximately 0.4 and 0.3 seconds.

The module extractor Mod has been assessed on the nonmonotonic versions of FLY and GO extended with random ABoxes as well. As expected, Mod is more effective when the ABox is less "interconnected": role assertions tend to introduce more dependencies, because they involve pairs of individuals; so Mod tends to be less effective as the percentage of role assertions increases. Accordingly, effectiveness increases as the ratio between the number of individuals and the number of assertions increases. The experimental results reported in Figures 4.13 and 4.14 confirm the above intuitions.

The above test sets are N-free. We carried out a third set of experiments on the non N-free suites as well. The response times of the naive algorithm and Mod are listed in Table 4.6 and 4.7 for increasing values of $|\Sigma|$ (that is directly related to the amount of normality concepts occurring in $\mathscr{K}\mathscr{B}$). In all cases, the naive algorithm exceeded the timeout. While the first test suite (CI-to-DI), Mod remains below 1 minute in all but

ABox size	role assrt.	individuals ∼ 5000	∼ 10000	∼ 20000
∼5000	10%	82% (17.10)	85% (15.32)	85% (14.27)
	20%	78% (21.87)	84% (20.26)	82% (20.19)
	30%	79% (23.40)	77% (2911)	76% (31.47)
∼10000	10%	76% (29.60)	72% (41.08)	76% (36.93)
	20%	65% (43.97)	60% (56.32)	68% (49.14)
	30%	61% (50.58)	59% (60.13)	60% (67.29)
∼20000	10%	41% (78.78)	40% (75.94)	39% (75.88)
	20%	40% (80.80)	39% (77.20)	34% (83.76)
	30%	30% (97.03)	37% (81.21)	31% (89.29)

Figure 4.13. Assessment of the module extractor in GO with non-empty ABox. The numbers in parentheses near the speedups are the average reasoning times using Mod (in sec.)

$\|\Sigma\|$	50	100	150	200	250
CI-to-DI					
naive	>30 min.	>30 min.	>30 min.	>30 min.	>30 min.
mod	2.70	8.59	16.95	28.16	42.04
Synthetic-DI					
naive	>30 min.	>30 min.	>30 min.	>30 min.	>30 min.
mod	186.5	414.5	696.6	1011.7	1411.8

Table 4.6. Impact of normal roles values (sec) on Gene Ontology – DI rate = 25% DA rate = 15%

one case for both ontologies; in the second suite (Synthetic-DI) it ranges between 186 seconds until exceeding 30 minutes. The reason of the higher computation times in the second suite is that the extracted modules are significantly larger, probably due to the random dependencies between concept names introduced by fully synthetic DIs.

$\mathscr{DL}^{\mathrm{N}}$'s module extractor proved to be very effective in speeding up nonmonotonic reasoning. There are still some limitations, though. First, the underlying classical $\bot$-module extractors in practice are not very effective in the presence of nonempty ABoxes; this affects also the performance of the more general $\top\bot^*$-module extraction approach (see the next section for a definition). This phenomenon is amplified in the nonmonotonic description logic $\mathscr{DL}^{\mathrm{N}}$, where reasoning requires repeated incremental classifications of the knowledge base. Furthermore, $\mathscr{DL}^{\mathrm{N}}$'s module extractor proved

ABox size	role assrt.	individuals ∼ 2000	∼ 4000	∼ 8000
∼2000	10%	80% (05.49)	83% (05.28)	83% (05.32)
	20%	79% (06.34)	79% (06.75)	78% (06.94)
	30%	75% (07.20)	76% (07.75)	75% (08.08)
∼4000	10%	63% (13.19)	69% (13.94)	70% (14.93)
	20%	63% (13.93)	67% (14.77)	70% (15.86)
	30%	67% (13.49)	71% (15.10)	81% (17.16)
∼8000	10%	68% (14.96)	69% (24.89)	53% (32.71)
	20%	67% (15.98)	67% (25.09)	42% (35.05)
	30%	72% (17.16)	71% (26.34)	34% (36.48)

Figure 4.14. Assessment of the module extractor in FLY with non-empty ABox. The numbers in parentheses near the speedups are the average reasoning times using Mod (in sec.)

$\|\Sigma\|$	50	100	150	200	250
CI-to-DI					
naive	>30 min.	>30 min.	>30 min.	>30 min.	>30 min.
mod	10.44	23.44	42.77	64.57	88.05
Synthetic-DI					
naive	>30 min.	>30 min.	>30 min.	>30 min.	>30 min.
mod	288.0	619.6	1020.0	1478.2	>30 min.

Table 4.7. Impact of normal roles values (sec) on Fly Anatomy – DI rate = 25% DA rate = 15%

to be less effective for KBs that contain many explicit occurrences of the normality concepts.

4.2.2 Iterated Module Extraction

In the previous section, it has been proved that all the queries whose signature is contained in Σ can be correctly answered with the translation $(\mathscr{K}\mathscr{B} \cap \mathscr{M}_0)^{\Sigma}$, where $\mathscr{M}_0$ is a $\top\bot^*$-module of $\mathscr{K}\mathscr{B} \cup \{\mathrm{N}C \sqsubseteq C \mid \mathrm{N}C \in \Sigma\}$ w.r.t. Σ. Note that, by definition, Σ contains all the $\mathrm{N}C \in \mathrm{sig}(\mathscr{K}\mathscr{B})$, as prescribed by the definition of $\mathscr{K}\mathscr{B}^{\Sigma}$. Here we reduce the size of Σ and the number of normality concepts to be processed by iterating module extraction, and progressively discarding the normality concepts that turn out

to be irrelevant to the given query.

First we recall the generalization to $\mathscr{DL}^{\mathrm{N}}$ of modules and locality. Basically, DIs are treated like classical inclusions (cf. Definition 4.2.1).

Let $\mathrm{Mod}_{\mathrm{DI}}(\mathit{Sig}, \mathscr{K}\mathscr{B})$ be the variant of $\top\bot^*$-Mod$(\mathit{Sig}, \mathscr{K}\mathscr{B})$ where the x-locality test in line 15 is replaced with the corresponding σ-locality condition of Def. 4.2.1 (so as to cover DIs). As argued, $\mathrm{Mod}_{\mathrm{DI}}(\mathit{Sig}, \mathscr{K}\mathscr{B})$ returns a syntactic module of $\mathscr{K}\mathscr{B}$ w.r.t. Sig according to Def. 4.2.1. Moreover, if $\mathscr{K}\mathscr{B}$ contains no DIs (i.e. it is a classical knowledge base), then for all queries α such that $\widetilde{\alpha} \subseteq \mathit{Sig}$,

$$\mathrm{Mod}_{\mathrm{DI}}(\mathit{Sig}, \mathscr{K}\mathscr{B}) \models \alpha \text{ iff } \mathscr{K}\mathscr{B} \models \alpha. \tag{4.6}$$

We are finally ready to define the iterated module extractor. Given a query α (which may be a subsumption or an assertion), and a canonical knowledge base $\mathscr{K}\mathscr{B} = \mathscr{S} \cup \mathscr{D}$, $\mathrm{Nsig}(\alpha, \mathscr{K}\mathscr{B})$ means the union of $\widetilde{\alpha}$ and all the normality concepts occurring in $\mathscr{K}\mathscr{B}$. Formally, we have

$$\mathrm{Nsig}(\alpha, \mathscr{K}\mathscr{B}) = \widetilde{\alpha} \cup \{\mathrm{N}C \mid \mathrm{N}C \text{ occurs in } \mathscr{D}\}\,,$$

Moreover, for a certain signature Sig, let

$$\mathit{Ax}(\mathit{Sig}) = \{\mathrm{N}C \sqsubseteq C \mid \mathrm{N}C \in \mathit{Sig}\}\,.$$

Now, define the sequence of modules $\mathscr{M}[0], \mathscr{M}[1], \ldots$ by letting

$$\mathscr{M}[0] = \mathscr{K}\mathscr{B} \cup \mathit{Ax}(\mathrm{Nsig}(\alpha, \mathscr{K}\mathscr{B}))$$

and, for $i > 0$

$$\mathscr{M}[i+1] = \mathrm{Mod}_{\mathrm{DI}}(\mathrm{Nsig}(\alpha, \mathscr{M}[i]), \mathscr{M}[i])\,.$$

Finally, let $\mathscr{M}^* = \bigcap_i \mathscr{M}[i]$ and $\mathit{Sig}^* = \bigcap_i \mathrm{Nsig}(\alpha, \mathscr{M}[i])$. We shall prove that $(\mathscr{K}\mathscr{B} \cap \mathscr{M}^*)^{\mathit{Sig}^*}$ can be used in place of $\mathscr{K}\mathscr{B}^{\Sigma}$ to answer query α. This saves the cost of processing $\mathscr{K}\mathscr{B}^{\Sigma}_{all} \setminus \mathscr{M}_1$, where

$$\mathscr{M}_1 = (\mathscr{K}\mathscr{B}^{\Sigma}_0 \cap \mathscr{M}^*) \cup \{\delta^{\mathrm{N}C} \mid \delta \in \mathscr{D} \cap \mathscr{M}^*,\ \mathrm{N}C \in \mathit{Sig}^*\} \tag{4.7}$$

A comparison of the new module $(\mathscr{K}\mathscr{B} \cap \mathscr{M}^*)^{\mathit{Sig}^*}$ with the previous one, $(\mathscr{K}\mathscr{B} \cap \mathscr{M}_0)^{\Sigma}$, shows that not only $\mathscr{M}^*$ is generally smaller than $\mathscr{M}_0$ (i.e. the translation applies to a smaller knowledge base); also the set of normality concepts to be processed is smaller, since $\mathit{Sig}^* \subseteq \Sigma$. So the translation's size reduction can be quadratic, in the best cases.

Now we have to prove the correctness of iterated module extraction. Some auxiliary lemmas are needed.

Lemma 4.2.9 *If $\mathscr{T}$ is a module of $\mathscr{K}\mathscr{B}$ w.r.t. a signature Σ and $\mathscr{K}\mathscr{B}' \subseteq \mathscr{K}\mathscr{B}$, then $\mathscr{K}\mathscr{B}' \cap \mathscr{T}$ is a module of $\mathscr{K}\mathscr{B}'$ w.r.t. Σ.*

Proof. If $\mathscr{T}$ is a module of $\mathscr{K}\mathscr{B}$ w.r.t. Σ, then $\mathscr{K}\mathscr{B} \setminus \mathscr{T}$ is σ-local for some $\top\bot^*$-substitution σ for $\mathscr{K}\mathscr{B}$ and $\widetilde{\mathscr{T}} \cup \Sigma$. Let σ' be the restriction of σ to the symbols in

$$\widetilde{\mathscr{K}\mathscr{B}}' \setminus (\widetilde{\mathscr{T}} \cup \Sigma) = \widetilde{\mathscr{K}\mathscr{B}'} \setminus (\mathrm{sig}(\mathscr{K}\mathscr{B}' \cap \mathscr{T}) \cup \Sigma).$$

Clearly, σ' is a $\top\bot^*$-substitution for $\mathscr{K}\mathscr{B}'$ and $\widetilde{\mathscr{K}\mathscr{B}}' \setminus (\mathrm{sig}(\mathscr{K}\mathscr{B}' \cap \mathscr{T}) \cup \Sigma)$. Moreover, for all $\beta \in \mathscr{K}\mathscr{B}' \setminus \mathscr{T}$, $\sigma(\beta) = \sigma'(\beta)$, by construction, so $\mathscr{K}\mathscr{B}' \setminus \mathscr{T}$ is σ'-local. Then $\mathscr{K}\mathscr{B}' \cap \mathscr{T}$ is a syntactic module of $\mathscr{K}\mathscr{B}'$ w.r.t. Σ. ∎

Lemma 4.2.10 *If $\mathscr{T}$ is a module of $\mathscr{K}\mathscr{B}$ w.r.t. a signature Σ and $\Sigma' \subseteq \Sigma$, then $\mathscr{T}$ is a module of $\mathscr{K}\mathscr{B}$ w.r.t. Σ', too.*

Proof. Let σ be a $\top\bot^*$-substitution for $\mathscr{K}\mathscr{B}$ for which $\mathscr{K}\mathscr{B} \setminus \mathscr{T}$ is σ-local w.r.t $\widetilde{\mathscr{T}} \cup \Sigma$. Let σ' be the extension of σ to the symbols in $\widetilde{\mathscr{K}\mathscr{B}} \setminus (\widetilde{\mathscr{T}} \cup \Sigma')$. Note that σ' differs form σ on the symbols in $\Sigma \setminus \Sigma'$, that can either be set to $\top$ or $\bot$. Clearly, σ' is a $\top\bot^*$-substitution for $\mathscr{K}\mathscr{B}$ and $\widetilde{\mathscr{K}\mathscr{B}} \setminus (\mathrm{sig}(\mathscr{K}\mathscr{B} \cap \mathscr{T}) \cup \Sigma')$. Now, consider all $\beta \in \mathscr{K}\mathscr{B} \setminus \mathscr{T}$. If $\widetilde{\beta} \cap (\Sigma \setminus \Sigma') = \emptyset$, $\sigma(\beta) = \sigma'(\beta)$, by construction, so β is σ'-local. Otherwise, $\widetilde{\beta} \cap (\Sigma \setminus \Sigma') \neq \emptyset$ and β is σ-local w.r.t $\widetilde{\mathscr{T}} \cup \Sigma$. By definition of locality, $\sigma(\beta)$ is a tautology for all possible interpretations of the symbols in Σ. In particular, β is still a tautology interpreting the symbols in $\Sigma \setminus \Sigma'$ as either $\top$ or $\bot$, so β is σ'-local. Then $\mathscr{K}\mathscr{B}' \cap \mathscr{M}$ is a syntactic module of $\mathscr{K}\mathscr{B}'$ w.r.t. Σ. ∎

Lemma 4.2.11 *$\mathscr{M}^*$ is a module of $\mathscr{K}\mathscr{B} \cup Ax(\Sigma)$ w.r.t. Sig^*.*

Proof. Observe that, being $\mathscr{K}\mathscr{B}$ finite, there exist an $n \geq 0$ such that for all $m > n$, $\mathscr{M}[m] = \mathscr{M}[n]$. Furthermore, for all $i = 0, ..., n$ we have $\mathscr{M}[i+1] \subseteq \mathscr{M}[i]$, by construction (recall that the locality notion is monotonic). By definition of $\mathrm{Nsig}(\alpha, \cdot)$ we also have that $\widetilde{\alpha} \subseteq \mathrm{Nsig}(\alpha, \mathscr{M}[i+1]) \subseteq \mathrm{Nsig}(\alpha, \mathscr{M}[i])$. It follows that

$$\mathscr{M}^* = \bigcap_i \mathscr{M}[i] = \mathscr{M}[n] \tag{4.8}$$

$$Sig^* = \bigcap_i \mathrm{Nsig}(\alpha, \mathscr{M}[i]) = \mathrm{Nsig}(\alpha, \mathscr{M}[n]). \tag{4.9}$$

Claim 1. For all $i > 0$, $\mathscr{M}[i]$ is a module of $\mathscr{M}[0]$ w.r.t. $\mathrm{Nsig}(\alpha, \mathscr{M}[i])$.

Proof of Claim 1. We prove the claim by induction.
Base case ($i = 1$):

$$\mathscr{M}[1] = \mathrm{Mod}_{\mathrm{DI}}(\mathrm{Nsig}(\alpha, \mathscr{M}[1]), \mathscr{M}[0])$$

By construction $\mathscr{M}[1] = \mathrm{Mod}_{\mathrm{DI}}(\mathrm{Nsig}(\alpha, \mathscr{M}[0]), \mathscr{M}[0])$. Since $\mathrm{Mod}_{\mathrm{DI}}(\cdot,\cdot)$ returns modules and $\mathrm{Nsig}(\alpha, \mathscr{M}[1]) \subseteq \mathrm{Nsig}(\alpha, \mathscr{M}[0])$, the case follows from Lemma 4.2.10.

Induction step ($i > 1$): By induction hypothesis (IH)

$$\mathscr{M}[i] = \mathrm{Mod}_{\mathrm{DI}}(\mathrm{Nsig}(\alpha, \mathscr{M}[i]), \mathscr{M}[0]).$$

If $\mathscr{M}[i]$ is a module of $\mathscr{M}[0]$ w.r.t. $\mathrm{Nsig}(\alpha, \mathscr{M}[i])$, then $\mathscr{M}[0] \setminus \mathscr{M}[i]$ is σ-local for some $\top\bot^*$-substitution σ for $\mathscr{M}[0]$ and $\mathrm{Nsig}(\alpha, \mathscr{M}[i]) \cup \widetilde{\mathscr{M}[i]} = \widetilde{\mathscr{M}[i]} \cup \widetilde{\alpha}$. $\mathscr{M}[i+1]$ is a module of $\mathscr{M}[i]$ w.r.t. $\mathrm{Nsig}(\alpha, \mathscr{M}[i])$, by construction, so $\mathscr{M}[i] \setminus \mathscr{M}[i+1]$ is σ'-local for some $\top\bot^*$-substitution σ' for $\mathscr{M}[i]$ and $\mathrm{Nsig}(\alpha, \mathscr{M}[i]) \cup \widetilde{\mathscr{M}[i+1]} = \widetilde{\mathscr{M}[i+1]} \cup \widetilde{\alpha}$. Note also, that the containment relation between the modules implies:

$$(\mathscr{M}[0] \setminus \mathscr{M}[i]) \cup (\mathscr{M}[i] \setminus \mathscr{M}[i+1]) = \mathscr{M}[0] \setminus \mathscr{M}[i+1]$$

Let $\bar{\sigma}$ be a substitution such that: *(i)* for all $C, R \notin \widetilde{\mathscr{M}[i]}$, $\bar{\sigma}(C) = \sigma(C)$ (resp. $\bar{\sigma}(R) = \sigma(R)$); *(ii)* for all $C, R \in \mathrm{sig}(\mathscr{M}[i] \setminus \mathscr{M}[i+1])$, $\bar{\sigma}(C) = \sigma'(C)$ (resp. $\bar{\sigma}(R) = \sigma'(R)$. Note, that the substitution $\bar{\sigma}$ is well defined as σ and σ' domains have no elements in common (σ' is restricted to the symbols in $\mathscr{M}[i] \setminus \{\mathscr{M}[i+1] \cup \alpha\}$, while σ to $\mathscr{M}[0] \setminus \{\mathscr{M}[i] \cup \alpha\}$), so their composition always exists. It is easy to see that for all $\beta \in (\mathscr{M}[0] \setminus \mathscr{M}[i]) \cup (\mathscr{M}[i] \setminus \mathscr{M}[i+1])$, β is $\bar{\sigma}$-local for $\mathscr{M}[0]$ and $\mathrm{Nsig}(\alpha, \mathscr{M}[i+1]) \cup \widetilde{\mathscr{M}[i+1]}$. The claim follows using the IH.
The lemma is a direct consequence of (4.8), (4.9) and Claim 1. ∎

Lemma 4.2.12 *$\mathscr{M}_1$ is a module of $\mathscr{K}\mathscr{B}^{\Sigma}_{all}$ w.r.t. Sig^*.*

Proof. By Lemma 4.2.11, $(\mathscr{K}\mathscr{B} \cup Ax(\Sigma)) \setminus \mathscr{M}^*$ is σ-local, for some $\top\bot^*$-substitution σ for $\mathscr{K}\mathscr{B} \cup Ax(\Sigma)$ and $\widetilde{\mathscr{M}^*} \cup Sig^*$. So, for all axioms β in $\mathscr{K}\mathscr{B}^{\Sigma}_0 \setminus \mathscr{M}_1$, β is σ-local (as $\mathscr{K}\mathscr{B}^{\Sigma}_0 \setminus \mathscr{M}_1 \subseteq (\mathscr{K}\mathscr{B} \cup Ax(\Sigma)) \setminus \mathscr{M}^*$). Now, let $\beta = \delta^{\mathrm{N}D} \notin \mathscr{M}_1$, where $\delta \in \mathscr{D}$ and $\mathrm{N}D \in \Sigma$. We have to consider two cases (w.r.t. the def. of $\mathscr{M}_1$, see (4.7)):

1. $\delta \in \mathscr{D} \setminus \mathscr{M}^*$, and hence δ is σ-local. Note that if $\sigma(E \sqsubseteq F)$ is a tautology, then also $\sigma(\mathrm{N}D \sqcap E \sqsubseteq F)$ is a tautology, therefore the σ-locality of δ implies the σ-locality of all $\delta^{\mathrm{N}D}$.
2. $\delta \in \mathscr{D} \cap \mathscr{M}^*$ and $\mathrm{N}D \in \Sigma \setminus Sig^*$. It is easy to see that $(\Sigma \setminus Sig^*) \cap (\widetilde{\mathscr{M}^*} \cup Sig^*) = \emptyset$ follows by construction, so $\sigma(\mathrm{N}D) = \bot$[8] and $\delta^{\mathrm{N}D}$ is σ-local.

[8] Algorithm 1 assigns first $\bot$ to the symbols not occurring in Sig^*.

It follows that all β in $\mathcal{KB}^{\Sigma}_{all} \setminus \mathcal{M}_1$ are σ-local. Finally, note that $\mathrm{sig}(\mathcal{KB}^{\Sigma}_{all}) = \mathrm{sig}(\mathcal{KB} \cup Ax(\Sigma))$ and $\widetilde{\mathcal{M}_1} \cup Sig^* = \widetilde{\mathcal{M}^*} \cup Sig^*$, therefore σ is also a $\top\bot^*$-substitution for $\mathcal{KB}^{\Sigma}_{all}$ and $\widetilde{\mathcal{M}_1} \cup Sig^*$. It follows immediately that $\mathcal{M}_1$ is a module of $\mathcal{KB}^{\Sigma}_{all}$ w.r.t. Sig^*. ∎

The relationship between $(\mathcal{KB} \cap \mathcal{M}^*)^{Sig^*}$ and $\mathcal{KB}^{\Sigma}$ is:

Lemma 4.2.13 $\mathcal{KB}^{\Sigma} \cap \mathcal{M}_1 \subseteq (\mathcal{KB} \cap \mathcal{M}^*)^{Sig^*} \subseteq \mathcal{KB}^{\Sigma}$.

Proof. Redefine the steps $(\mathcal{KB} \cap \mathcal{M}^*)^{Sig^*}_i$ of the classical translation by replacing $\mathcal{S}$ with $\mathcal{S} \cap \mathcal{M}^*$, and $\mathcal{KB}$ with $\mathcal{KB} \cap \mathcal{M}^*$ in (3.24) and (3.25), while δ_i ranges over *all* the DIs in $\mathcal{KB}$, not just $\mathcal{D} \cap \mathcal{M}^*$. This formulation is equivalent to the original one and facilitates the comparison with $\mathcal{KB}^{\Sigma}$.

Now it suffices to prove by induction that for all $i = 0, 1, \ldots, |\mathcal{D}|$,

$$\mathcal{KB}^{\Sigma}_i \cap \mathcal{M}_1 \subseteq (\mathcal{KB} \cap \mathcal{M}^*)^{Sig^*}_i \subseteq \mathcal{KB}^{\Sigma}_i .$$

Base case ($i = 0$): Similar to the corresponding case of Lemma 4.2.7.
Induction step ($i > 0$): By induction hypothesis (IH)

$$\mathcal{KB}^{\Sigma}_{i-1} \cap \mathcal{M}_1 \subseteq (\mathcal{KB} \cap \mathcal{M}^*)^{Sig^*}_{i-1} \subseteq \mathcal{KB}^{\Sigma}_{i-1} .$$

First suppose that $\delta_i \notin \mathcal{M}^*$ (and hence for all NC, $\delta_i^{\mathrm{NC}} \notin \mathcal{M}_1$). Then $\mathcal{KB}^{\Sigma}_i \cap \mathcal{M}_1 = \mathcal{KB}^{\Sigma}_{i-1} \cap \mathcal{M}_1$, $(\mathcal{KB} \cap \mathcal{M}^*)^{Sig^*}_i = (\mathcal{KB} \cap \mathcal{M}^*)^{Sig^*}_{i-1}$, and (by def.) $\mathcal{KB}^{\Sigma}_{i-1} \subseteq \mathcal{KB}^{\Sigma}_i$. The Lemma follows by IH.

Next assume that $\delta_i \in \mathcal{M}^*$ and let $\mathcal{F}_i = \{\delta_i^{\mathrm{ND}} \mid \mathrm{ND} \in Sig^*\}$ and $\mathcal{G}_i = \{\delta_i^{\mathrm{ND}} \mid \mathrm{ND} \in \Sigma \setminus Sig^*\}$. By construction, $\mathcal{KB}^{\Sigma}_i = \mathcal{KB}^{\Sigma}_{i-1} \cup (\mathcal{KB}^{\Sigma}_i \cap \mathcal{F}_i) \cup (\mathcal{KB}^{\Sigma}_i \cap \mathcal{G}_i)$. Note that $\mathcal{G}_i \cap \mathcal{M}_1 = \emptyset$ and $\mathcal{F}_i \subseteq \mathcal{M}_1$, by def. of $\mathcal{M}_1$. Consequently, $\mathcal{KB}^{\Sigma}_i \cap \mathcal{M}_1 = (\mathcal{KB}^{\Sigma}_{i-1} \cap \mathcal{M}_1) \cup (\mathcal{KB}^{\Sigma}_i \cap \mathcal{F}_i)$. Now, consider $(\mathcal{KB} \cap \mathcal{M}^*)^{Sig^*}_i = (\mathcal{KB} \cap \mathcal{M}^*)^{Sig^*}_{i-1} \cup ((\mathcal{KB} \cap \mathcal{M}^*)^{Sig^*}_i \cap \mathcal{F}_i) \cup ((\mathcal{KB} \cap \mathcal{M}^*)^{Sig^*}_i \cap \mathcal{G}_i)$. It is easy to see that also $\mathcal{G}_i \cap (\mathcal{M}^*)^{Sig^*}_i = \emptyset$ (in particular, for all $\mathrm{ND} \in \Sigma \setminus Sig^*$ we have $\sigma(\mathrm{ND}) = \bot$ and δ^{ND} is σ-local), therefore $(\mathcal{KB} \cap \mathcal{M}^*)^{Sig^*}_i = (\mathcal{KB} \cap \mathcal{M}^*)^{Sig^*}_{i-1} \cup ((\mathcal{KB} \cap \mathcal{M}^*)^{Sig^*}_i \cap \mathcal{F}_i)$. By IH for each $\delta^{\mathrm{ND}} \in \mathcal{F}_i$ we have:

$$\begin{aligned} (\mathcal{KB}^{\Sigma}_{i-1} \cap \mathcal{M}_1) \downarrow_{\prec\delta_i} \cup \{\delta_i^{\mathrm{ND}}\} \subseteq \\ \subseteq (\mathcal{KB} \cap \mathcal{M}^*)^{Sig^*}_{i-1} \downarrow_{\prec\delta_i} \cup \{\delta_i^{\mathrm{ND}}\} \subseteq \\ \subseteq \mathcal{KB}^{\Sigma}_{i-1} \downarrow_{\prec\delta_i} \cup \{\delta_i^{\mathrm{ND}}\} . \end{aligned}$$

The leftmost term equals $(\mathscr{K}\mathscr{B}^{\Sigma}_{i-1}\downarrow_{\prec\delta_i}\cup\{\delta_i^{\mathrm{N}D}\})\cap\mathscr{M}_1\subseteq\mathscr{K}\mathscr{B}^{\Sigma}_{all}$, so by Lemmas 4.2.12 and 4.2.9 and (4.6), the leftmost term entails $\mathrm{N}D\sqsubseteq\bot$ iff the rightmost does. It follows that the middle term $(\mathscr{K}\mathscr{B}\cap\mathscr{M}^*)^{Sig^*}_{i-1}\downarrow_{\prec\delta_i}\cup\{\delta_i^{\mathrm{N}D}\}$ entails $\mathrm{N}D\sqsubseteq\bot$ iff the other two terms do. Then, $(\mathscr{K}\mathscr{B}\cap\mathscr{M}^*)^{Sig^*}_{i}\cap\mathscr{F}_i=\mathscr{K}\mathscr{B}^{\Sigma}_{i}\cap\mathscr{F}_i$. The Lemma follows using the IH. ∎

As a consequence, the modularized construction is correct:

Theorem 4.2.14 *If* $\mathrm{sig}(\alpha)\subseteq Sig^*$*, then* $(\mathscr{K}\mathscr{B}\cap\mathscr{M}^*)^{Sig^*}\models\alpha$ *iff* $\mathscr{K}\mathscr{B}^{\Sigma}\models\alpha$.

Proof. By Lemmas 4.2.12 and 4.2.9, $\mathscr{K}\mathscr{B}^{\Sigma}\cap\mathscr{M}_1$ is a module of $\mathscr{K}\mathscr{B}^{\Sigma}$ w.r.t. Sig^*. Then $\mathscr{K}\mathscr{B}^{\Sigma}\cap\mathscr{M}_1\models\alpha$ iff $\mathscr{K}\mathscr{B}^{\Sigma}\models\alpha$. The theorem then follows from Lemma 4.2.7 and the monotonicity of $\models$. ∎

Experimental Analysis

In the following we analyse the performance of the iterated module extraction for $\mathscr{DL}^{\mathrm{N}}$ described in Section 4.2.2 according to the experimental setup described in Section 4.1.4. Note that the first tests reported in Tables 4.2, 4.3, 4.4 and 4.5, and Figures 4.13 and 4.14 are N-free (normality concepts occur only in the queries), therefore the iterated module extractor (here denoted by mod*) yields the same result as Mod, since mod* immediately reaches a fix-point.

The iterative module extractor mod* shows its benefits in the test suites, illustrated in Figures 4.15 and 4.16, where randomly selected DIs are modified by turning the concepts C in the scope of role restrictions into the corresponding normality concepts $\mathrm{N}C$[9]. In these test cases, the size of Σ may grow up to 250 normality concepts. Since for each DI δ the translation of $\mathscr{K}\mathscr{B}$ must process an inclusion $\delta^{\mathrm{N}C}$ for each $\mathrm{N}C\in\Sigma$, the computational cost of the naive translation grows significantly with $|\Sigma|$. The old module extractor, mod, includes all of Σ in the relevant signature, so the extracted modules are quite large and the computation time is considerably slower, compared to the N-free examples in Tables 4.2 and 4.3. Figure 4.12 shows also that with mod*, $\mathscr{DL}^{\mathrm{N}}$ reasoning is more than one order of magnitude faster than using mod. This remarkable result can be explained by the twofold benefits of reducing the size of Σ: First, whenever some $\mathrm{N}C$ is eliminated from Σ, mod* can further reduce the module in the next iterations. Moreover, as explained above, a smaller Σ reduces the number of inclusions $\delta^{\mathrm{N}C}$ processed in the translation (for each fixed DI δ included in the module).

[9]The explicit use of normality concepts in $\mathscr{K}\mathscr{B}$ is needed precisely to restrict role range to prototypical individuals, cf. Section 3.1.6.

$\|\Sigma\|$	50	100	150	200	250
GO – CI-to-DI					
mod	2.70	8.59	16.95	28.16	42.04
mod*	0.45	0.45	0.46	0.46	0.46
GO – Synthetic-DI					
mod	186.5	414.5	696.6	1011.7	1411.8
mod*	8.18	10.45	15.29	20.45	28.30

Figure 4.15. Non N-free tests. Impact of normal roles ranges (sec) – DI rate = 25% DA rate = 15%.

$\|\Sigma\|$	50	100	150	200	250
FLY – CI-to-DI					
mod	10.44	23.44	42.77	64.57	88.05
mod*	0.47	0.58	0.66	0.73	0.99
FLY – Synthetic-DI					
mod	288.0	619.6	1020.0	1478.2	-
mod*	22.4	34.0	47.8	63.5	83.5

Figure 4.16. Non N-free tests. Impact of normal roles ranges (sec) – DI rate = 25% DA rate = 15%.

For a discussion of additional benefits provided by mod*, e.g., by eliminating most normality concepts, it may enable the application of a new optimization technique, please refer to Section 4.3.

4.2.3 A Module Extractor for ABoxes

The module extractor based on $\bot$-locality ($\bot$-Mod, Algorithm 1) may be significantly less effective when applied to (classical) knowledge bases with nonempty ABoxes (cf. evidence has been provided in Section 4.2.1). The reason is that replacing an assertion's predicate with $\bot$ (e.g. replacing $A(c)$ with $\bot(c)$) always produces an inconsistent axiom, therefore the module's signature must contain all the predicate symbols occurring in the ABox. In turn, this weakness of $\bot$-Mod reduces the effectiveness of the overall module extractor $\top\bot^*$-Mod.

In this section we refine $\bot$-Mod to address this problem. We call the resulting module extractor *conditional* because it is correct under the mild assumption that $\mathscr{K}\mathscr{B}$ is consistent; if not, the fall-back solution is the original $\bot$-Mod.

The refined module extractor, called $\bot$-cMod, is recursively defined in terms of $\bot$-Mod. For all classical $\mathscr{K}\mathscr{B}$ with TBox $\mathscr{T}$ and ABox $\mathscr{A}$, and all signatures Sig, let:

$$\begin{aligned}
&\mathscr{M}_0 = \mathscr{A}_0 = \emptyset \\
&\mathscr{M}_{i+1} = \bot\text{-Mod}(\text{sig}(\mathscr{M}_i \cup \mathscr{A}_i) \cup Sig, \mathscr{T}) \\
&\mathscr{A}_{i+1} = \textstyle\bigcup_{j>0} \mathscr{A}_{i+1,j}, \text{ where} \\
&\mathscr{A}_{i+1,0} = \emptyset \\
&\mathscr{A}_{i+1,j+1} = \{\alpha \in \mathscr{A} \mid \text{sig}(\alpha) \cap \text{sig}(\mathscr{M}_{i+1} \cup \mathscr{A}_{i+1,j}) \neq \emptyset\}\,.
\end{aligned}$$

Finally, let $\bot\text{-cMod}(Sig, \mathscr{K}\mathscr{B}) = \bigcup_{i>0} \mathscr{M}_i \cup \mathscr{A}_i$.

Note that at each step i, $\bot$-Mod is applied only to the TBox, so the ABox does not prevent axioms from being removed from the current module $\mathscr{M}_i$. Then the ABox is inspected to extend the module and its signature with the assertions that are syntactically connected to $\mathscr{M}_i$ (which are gathered by the sequence $\langle \mathscr{A}_{i,j} \rangle_j$). Clearly the non-decreasing sequences $\langle \mathscr{M}_i \rangle_i$ and $\langle \mathscr{A}_i \rangle_i$ reach a fix-point after a finite number of steps, because $\mathscr{K}\mathscr{B}$ is finite.

Example 4.2.15 Consider the knowledge base $\mathscr{KB}_0$

$$\begin{array}{ll} A \sqsubseteq B & A(c),\ R(c,d) \\ B' \sqsubseteq C & B'(d) \\ E \sqsubseteq F & E(a)\,. \end{array}$$

Given $Sig = \{A\}$, we have

$$\begin{array}{ll} \mathscr{M}_1 = \{A \sqsubseteq B\} & \mathscr{A}_1 = \{A(c),\ R(c,d), B'(d)\} \\ \mathscr{M}_2 = \mathscr{M}_1 \cup \{B' \sqsubseteq C\} & \mathscr{A}_2 = \mathscr{A}_1 \\ \mathscr{M}_3 = \mathscr{M}_2 & \mathscr{A}_3 = \mathscr{A}_2\,. \end{array}$$

Then $\bot$-cMod$(Sig, \mathscr{KB}_0) = \mathscr{M}_3 \cup \mathscr{A}_3 = \mathscr{KB}_0 \setminus \{E \sqsubseteq F, E(a)\}$. Note that the standard module extractor $\bot$-Mod returns the entire $\mathscr{KB}_0$, because E cannot be consistently replaced with $\bot$ due to assertion $E(a)$. ∎

If $\mathscr{KB}$ is consistent and nominal-free, then $\bot$-cMod$(Sig, \mathscr{KB})$ enjoys the fundamental property of modules, that guarantees their completeness w.r.t. the queries that can be formulated with Sig:

Theorem 4.2.16 *Let $\mathscr{KB}$ be a consistent $\mathcal{SROIQ}$ knowledge base with no nominals. Then each model of $\bot$-cMod$(Sig, \mathscr{KB})$ can be extended to a model of $\mathscr{KB}$.*

Proof. Let $\mathscr{M} = \bot$-cMod$(Sig, \mathscr{KB})$, and let $\mathscr{M}_T$ and $\mathscr{M}_A$ denote $\mathscr{M}$'s TBox and ABox, respectively. We may assume w.l.o.g. that $\mathscr{KB}$ does not contain the universal role (cf. [Horrocks et al., 2006]).

Since $\mathscr{KB}$ is consistent, there exists a model $\mathscr{I} = \langle \Delta^{\mathscr{I}}, \cdot^{\mathscr{I}} \rangle$ of $\mathscr{KB} \setminus \mathscr{M}_A$. Note that $\mathscr{KB} \setminus \mathscr{M}_A$ contains none of the individual constants in $\mathscr{M}$, because (i) $\mathscr{KB}$ is nominal-free, and (ii) if some assertion α contains a constant $c \in \mathrm{sig}(\mathscr{M}_A)$, then $\alpha \in \mathscr{M}_A$, by definition of $\mathscr{A}_{i+1,j+1}$.

Next, let $\mathscr{J} = \langle \Delta^{\mathscr{J}}, \cdot^{\mathscr{J}} \rangle$ be a model of $\mathscr{M}$. We may assume w.l.o.g that the domains of $\mathscr{I}$ and $\mathscr{J}$ are disjoint ($\Delta^{\mathscr{I}} \cap \Delta^{\mathscr{J}} = \emptyset$).

Claim: for all $\alpha \in \mathscr{T} \setminus \mathscr{M}_T$ (where $\mathscr{T}$ is $\mathscr{KB}$'s TBox), α is $\bot$-local w.r.t. Sig.

To prove the claim, note that for some index k, $\mathscr{M} = \mathscr{M}_k \cup \mathscr{A}_k$, and $\mathscr{M}_k = \mathscr{M}_{k+1}$. By the properties of $\bot$-Mod, it follows that all the $\alpha \in \mathscr{T} \setminus \mathscr{M}_k$ are $\bot$-local w.r.t. a superset of Sig, hence w.r.t. Sig alone. Since $\mathscr{T} \setminus \mathscr{M}_k = \mathscr{T} \setminus \mathscr{M}_T$, the claim immediately follows.

By the claim, $\mathscr{J}$ can be extended to a model $\mathscr{J}'$ of $\mathscr{T} \cup \mathscr{M}_A$ simply by setting $N^{\mathscr{J}'} = \emptyset$ for all concept and role names N in $\mathrm{sig}(\mathscr{K}\mathscr{B}) \setminus Sig$.

Finally, it is easy to verify that the union $\mathscr{K}$ of $\mathscr{I}$ and $\mathscr{J}'$ is a model of $\mathscr{K}\mathscr{B}$, where $\mathscr{K}$ is defined as follows:

$$\begin{aligned}
\Delta^{\mathscr{K}} &= \Delta^{\mathscr{I}} \cup \Delta^{\mathscr{J}'} \\
N^{\mathscr{K}} &= N^{\mathscr{I}} \cup N^{\mathscr{J}'} \quad \text{(for all predicate names } N\text{)} \\
a^{\mathscr{K}} &= \begin{cases} a^{\mathscr{J}'} & \text{if individual constant } a \text{ occurs in } \mathscr{M} \\ a^{\mathscr{I}} & \text{otherwise.} \end{cases}
\end{aligned}$$

∎

The correctness of $\bot$-cMod immediately follows:

Corollary 4.2.17 *Let $\mathscr{K}\mathscr{B}$ be a consistent $\mathcal{SROIQ}$ knowledge base with no nominals. Then for all subsumptions and assertions α such that* $\mathrm{sig}(\alpha) \subseteq Sig$,

$$\mathscr{K}\mathscr{B} \models \alpha \text{ if and only if } \bot\text{-}cMod(Sig, \mathscr{K}\mathscr{B}) \models \alpha\,.$$

The same idea can be applied to *some* $\mathscr{K}\mathscr{B}$ with nominals. The prerequisite is that the individuals in the module and those in the rest of the knowledge base should be logically unrelated. In order to make this test practically feasible, it is approximated by means of syntactic locality.

Theorem 4.2.18 *Let $\mathscr{K}\mathscr{B}$ be a consistent $\mathcal{SROIQ}$ knowledge base, and let $\mathscr{M} = \bot\text{-}cMod(Sig, \mathscr{K}\mathscr{B})$. If no individual constant occurs both in $\mathscr{M}$ and in $\mathscr{M}' = \top\bot^*\text{-}Mod(\emptyset, \mathscr{K}\mathscr{B} \setminus \mathscr{M}_A)$ (where $\mathscr{M}_A$ is $\mathscr{M}$'s ABox), then each model of $\bot\text{-}cMod(Sig, \mathscr{K}\mathscr{B})$ can be extended to a model of $\mathscr{K}\mathscr{B}$, and*

$$\mathscr{K}\mathscr{B} \models \alpha \text{ if and only if } \bot\text{-}cMod(Sig, \mathscr{K}\mathscr{B}) \models \alpha\,.$$

Proof. Let $\mathscr{T}$ and $\mathscr{A}$ denote $\mathscr{K}\mathscr{B}$'s TBox and ABox, respectively. Let $\mathscr{J}$ be a model of $\mathscr{M}$ and $\mathscr{J}'$ its extension to $\mathscr{T} \cup \mathscr{M}_A$ (cf. proof of Theorem 4.2.16).

Let $\mathscr{I}$ be a model of $\mathscr{M}'$ (we assume w.l.o.g. that $\Delta^{\mathscr{I}} \cap \Delta^{\mathscr{J}} = \emptyset$), and let σ be a $\top\bot^*$-substitution for $\mathscr{K}\mathscr{B}$ and the signature of $\mathscr{M}'$ that makes all axioms in $(\mathscr{K}\mathscr{B} \setminus \mathscr{M}_A) \setminus \mathscr{M}'$

σ-local (σ exists by the properties of $\top\bot^*$-Mod). Extend $\mathscr{I}$ to $\mathscr{I}'$ as follows:

$$N^{\mathscr{I}'} = \begin{cases} N^{\mathscr{I}} & \text{if predicate } N \text{ occurs in } \mathscr{M}' \\ \sigma(N)^{\mathscr{I}} & \text{otherwise} \end{cases}$$

$$a^{\mathscr{I}'} = \begin{cases} a^{\mathscr{I}} & \text{if constant } a \text{ occurs in } \mathscr{M}' \\ x & \text{if } a \text{ occurs neither in } \mathscr{M} \text{ nor in } \mathscr{M}' \end{cases}$$

where x is any member of $\Delta^{\mathscr{I}}$. Finally, let $\mathscr{K}$ be the union of $\mathscr{I}'$ and $\mathscr{J}'$ (cf. proof of Theorem 4.2.16). Since $\mathscr{M}$ and $\mathscr{M}'$ have no individual constants in common, $\mathscr{K}$ is well defined. The reader may easily verify that $\mathscr{K}$ is a model of $\mathscr{K}\mathscr{B}$. This proves that each model $\mathscr{J}$ of $\mathscr{M}$ can be extended to a model $\mathscr{K}$ of $\mathscr{K}\mathscr{B}$; it follows that $\mathscr{K}\mathscr{B} \models \alpha$ if and only if $\bot$-cMod$(Sig, \mathscr{K}\mathscr{B}) \models \alpha$. ∎

Experimental Analysis

The conditional module extractor $\bot$-cMod for nonempty ABoxes has been assessed on the test suites obtained by adding random ABoxes to the nonmonotonic versions of FLY and GO. As expected, $\bot$-cMod is more effective when the ABox is less "interconnected": if a same individual occurs in many assertions, then introducing in a module any of the predicates occurring in those assertions causes the other predicates to be included, too, by the definition of the sequences $\langle A_{i,j}\rangle_j$. Accordingly, effectiveness increases as the ratio between the number of individuals and the number of assertions increases. Moreover, role assertions tend to introduce more dependencies, because they involve pairs of individuals; so $\bot$-cMod tends to be less effective as the percentage of role assertions increases. The experimental results reported in Figures 4.17 and 4.18 confirm the above intuitions.

Figures 4.19 and 4.20 provide further evidence in this sense in terms of the corresponding reduction in the extracted module sizes when $\bot$-cMod replaces $\bot$-Mod.

We have also compared $\bot$-cMod and OWL API's $\bot$-Mod over 16 classical KBs from Oxford's OBO repository. The initial signature has been selected with the following methods: (i) the symbols occurring in one randomly selected assertion; (ii) those in a randomly selected inclusion; (iii) n concepts, randomly selected from those occurring in the KB, for $n = 2, 5, 10$. On average, the size of the modules extracted alternating $\bot$-cMod and $\top$-Mod is 15% of the size of the modules extracted by the classical $\top\bot^*$-Mod.

Speedup of mod when ⊥-cMod replaces ⊥-Mod

ABox size	role assrt.	individuals ∼ 5000	∼ 10000	∼ 20000
∼5000	10%	69% (05.25)	77% (03.58)	78% (03.10)
	20%	63% (08.01)	76% (04.92)	76% (04.82)
	30%	62% (08.85)	77% (06.56)	80% (06.34)
∼10000	10%	33% (19.86)	71% (12.03)	70% (11.03)
	20%	30% (30.93)	66% (19.33)	68% (15.71)
	30%	31% (35.08)	65% (21.12)	71% (19.36)
∼20000	10%	19% (60.68)	20% (66.68)	24% (57.54)
	20%	16% (68.06)	18% (63.64)	15% (71.31)
	30%	4% (93.18)	09% (74.01)	11% (79.45)

Figure 4.17. Assessment of the conditional module extractor in GO. The numbers in parentheses near the speedups are the average reasoning times using ⊥-cMod (in sec.)

4.3 Optimistic Computation

The construction of $\mathcal{K}\mathcal{B}^{\Sigma}$ repeats the concept consistency check (3.25) over knowledge bases $(\mathcal{K}\mathcal{B}^{\Sigma}_{i-1}\downarrow_{\prec\delta_i} \cup \{\delta_i^{\mathrm{NC}}\})$ that share a (possibly large) common part $\mathcal{K}\mathcal{B}^{\Sigma}_{0}$, so incremental reasoning mechanisms help by avoiding multiple computations of the consequences of $\mathcal{K}\mathcal{B}^{\Sigma}_{0}$. On the contrary, the set of δ_j^{NC} may change significantly at each step due to the filtering $\downarrow_{\prec\delta_i}$. This operation requires many axiom deletions, which are less efficient than monotonically increasing changes. The optimistic algorithm introduced here (Algorithm 2) computes a knowledge base $\mathcal{K}\mathcal{B}^{*}$ equivalent to $\mathcal{K}\mathcal{B}^{\Sigma}$ in a way that tends to reduce the number of deletions.

Phase 1 optimistically assumes that the DIs with the same priority as δ_i^{NC} do not contribute to entailing $\mathrm{N}C \sqsubseteq \bot$ in (3.25), so they are not filtered with $\downarrow_{\delta_i}$ in line 7.Phase 2 checks whether the DIs discarded during Phase 1 are actually overridden by applying $\downarrow_{\delta_i}$ (lines 14 and 21). DIs are processed in non-increasing priority order as much as possible (cf. line 7) so as to exploit monotonic incremental classifications.

The following theorem shows the correctness of Algorithm 2 in case the normality concepts do not occur in $\mathcal{K}\mathcal{B}$, but only in the queries. In Section 3.1 we have called such knowledge bases *N-free*. It is worth noting that the optimistic method is not generally correct when $\mathcal{K}\mathcal{B}$ is not N-free and $|\Sigma| > 1$, yet it may still be applicable after the module extractor if the latter removes all normality concepts from $\mathcal{K}\mathcal{B}$.

Theorem 4.3.1 *If $\mathcal{K}\mathcal{B}$ is N-free, then Algorithm 2's output is equivalent to $\mathcal{K}\mathcal{B}^{\Sigma}$.*

Speedup of mod when ⊥-cMod replaces ⊥-Mod

ABox size	role assrt.	individuals ∼ 2000	∼ 4000	∼ 8000
∼2000	10%	51% (02.71)	61% (02.06)	63% (01.98)
	20%	44% (03.52)	53% (03.19)	52% (03.32)
	30%	37% (04.56)	45% (04.24)	48% (04.19)
∼4000	10%	31% (09.11)	57% (06.06)	69% (04.67)
	20%	33% (09.33)	46% (08.04)	54% (07.22)
	30%	29% (09.53)	35% (09.80)	43% (09.80)
∼8000	10%	02% (13.70)	25% (18.73)	53% (15.43)
	20%	00% (16.00)	16% (20.97)	42% (20.39)
	30%	02% (16.81)	11% (23.40)	34% (24.02)

Figure 4.18. Assessment of the conditional module extractor in FLY. The numbers in parentheses near the speedups are the average reasoning times using ⊥-cMod (in sec.)

Proof. First assume that Σ is a singleton $\{\mathrm{N}C\}$. We start by proving some invariants of lines 6-10.

Claim 1: $\mathscr{K}\mathscr{B}^{\Sigma} \models \Pi$.

Claim 2: If, for some $j < i$, $\delta_j^{\mathrm{N}C} \in \mathscr{K}\mathscr{B}^{\Sigma} \setminus \Pi$, then $\mathscr{K}\mathscr{B}^{\Sigma} \models \mathrm{N}C \sqsubseteq \bot$.

We prove these two claims simultaneously. Both claims hold vacuously at the first execution of line 6. Next, assume by induction hypothesis that they hold at line 6 in some iteration; we have to prove that they still hold at the next iteration. There are two possibilities: First suppose that for some $j < i$, $\delta_j^{\mathrm{N}C} \in \mathscr{K}\mathscr{B}^{\Sigma} \setminus \Pi$. By Claim 2, $\mathscr{K}\mathscr{B}^{\Sigma} \models \mathrm{N}C \sqsubseteq \bot$. This immediately implies that Claim 2 holds also at the next iteration. Moreover, it implies Claim 1 because all members of Π have an occurrence of $\mathrm{N}C$ in the left-hand side.

We are left the case in which

$$\text{for all } j < i, \text{ if } \delta_j^{\mathrm{N}C} \in \mathscr{K}\mathscr{B}^{\Sigma} \text{ then } \delta_j^{\mathrm{N}C} \in \Pi. \tag{4.10}$$

If the condition in line 7 is true, then Π is not changed, so Claim 1 must hold at the next iteration. Otherwise, by (4.10),

$$\mathscr{K}\mathscr{B}_0^{\Sigma} \cup \Pi' \supseteq \mathscr{K}\mathscr{B}_{i-1}^{\Sigma}\downarrow_{\delta_i} \cup \{\delta_i^{\mathrm{N}C}\}, \tag{4.11}$$

and hence $\mathrm{N}C \sqsubseteq \bot$ is not provable in (3.25), either. It follows that $\delta_i^{\mathrm{N}C}$ belongs to both Π (by line 8) and $\mathscr{K}\mathscr{B}^{\Sigma}$ (by (3.25)). This proves Claim 1 for iteration i.

Module size reduction when $\bot$-cMod replaces $\bot$-Mod

ABox size	role assrt.	individuals ∼ 2000	∼ 4000	∼ 8000
∼2000	10%	32.14%	42.54%	40.94%
	20%	19.33%	24.88%	21.77%
	30%	16.20%	10.08%	9.03%
∼4000	10%	12.23%	16.52%	20.53%
	20%	9.80%	14.03%	15.20%
	30%	6.63%	13.52%	16.84%
∼8000	10%	8.67%	8.49%	8.60%
	20%	7.98%	7.48%	7.49%
	30%	3.11%	4.48%	5.37%

Figure 4.19. Assessment of the conditional module extractor in GO. The numbers represent the average module size reduction using $\bot$-cMod (in sec.)

Concerning Claim 2, first suppose that the condition in line 7 is true; then either Claim 2 remains vacuously satisfied, or $\delta_i^{\mathrm{NC}} \in \mathcal{K}\mathcal{B}^{\Sigma} \setminus \Pi$. The latter (plus the def. of $\mathcal{K}\mathcal{B}^{\Sigma}$ and the ind. hyp. of Claim 1) implies that $\mathcal{K}\mathcal{B}_0^{\Sigma} \cup \Pi'$ is entailed by $\mathcal{K}\mathcal{B}^{\Sigma}$. It follows that $\mathcal{K}\mathcal{B}^{\Sigma} \models \mathrm{N}C \sqsubseteq \bot$ as well, which proves Claim 2 in this case. Finally, if the condition in line 7 is false, then at line 8 $\delta_i^{\mathrm{NC}} \in \Pi$. Together with (4.10), this implies that Claim 2 holds vacuously.

Claim 3: If $\mathcal{K}\mathcal{B}^{\Sigma} \models \mathrm{N}C \sqsubseteq \bot$ then $\mathcal{K}\mathcal{B}^{*} \models \mathrm{N}C \sqsubseteq \bot$.

Suppose not (we shall derive a contradiction). The assumption and Claim 1 imply that there must be some $\delta_k^{\mathrm{NC}} \in \mathcal{K}\mathcal{B}^{\Sigma} \setminus \mathcal{K}\mathcal{B}^{*}$. Let δ_i^{NC} be the one with minimal k. Using minimality, it can be proved that $\mathcal{K}\mathcal{B}_0^{\Sigma} \cup \Pi \downarrow_{\delta_i} = \mathcal{K}\mathcal{B}_{i-1}^{\Sigma} \downarrow_{\delta_i}$, so the concept consistency tests in lines 14 and 21 (the latter instantiated with $j = i$ and $D = C$) are equivalent to the one in (3.25). But then $\delta_i^{\mathrm{NC}} \in \mathcal{K}\mathcal{B}^{\Sigma}$ iff $\delta_i^{\mathrm{NC}} \in \mathcal{K}\mathcal{B}^{*}$, which contradicts the assumption, so Claim 3 is proved.

Claim 4: If $\mathcal{K}\mathcal{B}^{\Sigma} \models \mathrm{N}C \sqsubseteq \bot$ then $\mathcal{K}\mathcal{B}^{*} \equiv \mathcal{K}\mathcal{B}^{\Sigma}$.

Note that $\mathcal{K}\mathcal{B}^{*} \subseteq \mathcal{K}\mathcal{B}_0^{\Sigma} \cup \Pi \cup \{\mathrm{N}C \sqsubseteq \bot\}$ (cf. lines 11, 15, and 22). Clearly, $\mathcal{K}\mathcal{B}^{\Sigma} \models \mathcal{K}\mathcal{B}_0^{\Sigma} \cup \Pi \cup \{\mathrm{N}C \sqsubseteq \bot\}$ (by def., Claim 1 and the assumption), so $\mathcal{K}\mathcal{B}^{\Sigma} \models \mathcal{K}\mathcal{B}^{*}$. We are left to prove $\mathcal{K}\mathcal{B}^{*} \models \mathcal{K}\mathcal{B}^{\Sigma}$. By Claim 3, $\mathcal{K}\mathcal{B}^{*} \models \mathrm{N}C \sqsubseteq \bot$, and this inclusion in turn entails all $\delta_i^{\mathrm{NC}} \in \mathcal{K}\mathcal{B}^{\Sigma}$ (cf. (3.24)). The other members of $\mathcal{K}\mathcal{B}^{\Sigma}$ are those in $\mathcal{K}\mathcal{B}_0^{\Sigma}$, by definition, and $\mathcal{K}\mathcal{B}^{*} \supseteq \mathcal{K}\mathcal{B}_0^{\Sigma}$ (line 11). It follows that $\mathcal{K}\mathcal{B}^{*} \models \mathcal{K}\mathcal{B}^{\Sigma}$, which completes the proof of Claim 4.

Claim 5: If $\mathcal{K}\mathcal{B}^{\Sigma} \not\models \mathrm{N}C \sqsubseteq \bot$ then $\mathcal{K}\mathcal{B}^{*} \equiv \mathcal{K}\mathcal{B}^{\Sigma}$.

Module size reduction when $\bot$-cMod replaces $\bot$-Mod

ABox size	role assrt.	individuals ∼ 2000	∼ 4000	∼ 8000
∼2000	10%	38.12%	46.58%	48.45%
	20%	22.83%	27.52%	26.94%
	30%	6.92%	7.49%	9.68%
∼4000	10%	15.66%	36.28%	43.92%
	20%	9.33%	21.38%	26.68%
	30%	1.33%	6.73%	8.46%
∼8000	10%	1.29%	10.49%	29.98%
	20%	0.04%	7.29%	18.70%
	30%	0.00%	1.43%	5.89%

Figure 4.20. Assessment of the conditional module extractor in FLY. The numbers represent the average module size reduction using $\bot$-cMod (in sec.)

Suppose that $\mathscr{K}\mathscr{B}^{\Sigma} \not\models \mathrm{N}C \sqsubseteq \bot$. Then, by the contrapositive of Claim 2 and Claim 1, $\mathscr{K}\mathscr{B}_0^{\Sigma} \cup \Pi \equiv \mathscr{K}\mathscr{B}^{\Sigma}$, which further implies that the concept consistency tests in lines 14 and 21 are equivalent to the corresponding test in (3.25). Then it can be proved that if any of these tests were true, then also $\mathscr{K}\mathscr{B}^{\Sigma} \models \mathrm{N}C \sqsubseteq \bot$ because, by Π's construction, in that case δ_i must be in conflict with some other DI with the same priority. However, $\mathscr{K}\mathscr{B}^{\Sigma} \models \mathrm{N}C \sqsubseteq \bot$ contradicts the assumption. It follows that all tests in lines 14 and 21 are false, so $\mathscr{K}\mathscr{B}^{*} = \mathscr{K}\mathscr{B}_0^{\Sigma} \cup \Pi$, and we have already argued that this knowledge base is equivalent to $\mathscr{K}\mathscr{B}^{\Sigma}$. This completes the proof for $|\Sigma| = 1$.

For $|\Sigma| > 1$, note that the test in lines 7 and 14 (resp. 21) do not depend on any $\delta_k^{\mathrm{N}E}$ such that $E \neq C$ (resp. $E \neq D$). Indeed, by $\bot$-*locality*, all such $\delta_k^{\mathrm{N}E}$ are local w.r.t. the signature of $\mathscr{K}\mathscr{B} \cup \{\mathrm{N}C \sqsubseteq \bot\}$, so they can be removed without changing the result of the concept consistency test [Sattler et al., 2009]. However, after their removal, the concept consistency tests correspond to the ones for the singleton case $\Sigma = \{\mathrm{N}C\}$, which we have already proved correct. ∎

4.3.1 Experimental Analysis

For each parameter setting, we report the execution time of: (*i*) the naive $\mathscr{D}\mathscr{L}^{\mathrm{N}}$ reasoner; (*ii*) the optimistic method introduced in Sec. 4.3 (Opt); (*iii*) the module extraction method of Sec. 4.2 (Mod) using the module extraction facility of the OWL API; (*iv*) the sequential execution of Mod and Opt, i.e. Algorithm 2 is applied to

Algorithm 2: Optimistic-Method

Input: $\mathscr{K}\mathscr{B} = \mathscr{S} \cup \mathscr{D}$, Σ
Output: a knowledge base $\mathscr{K}\mathscr{B}^*$ such that $\mathscr{K}\mathscr{B}^* \equiv \mathscr{K}\mathscr{B}^{\Sigma}$

```
// Phase 1
compute a linearization δ_1, ..., δ_|𝒟| of 𝒟
Π := ∅  // Π collects the prototypes
Δ := ∅  // ordered list of all discarded δ_i^NC
for i = 1, 2, ..., |𝒟| do
    for NC ∈ Σ do
        Π' := Π ∪ {δ_i^NC}
        if 𝒦𝓑_0^Σ ∪ Π' ⊭ NC ⊑ ⊥ then
            Π := Π'
        else
            append δ_i^NC to Δ
// Phase 2
𝒦𝓑* = 𝒦𝓑_0^Σ ∪ Π
while Δ ≠ ∅ do
    extract from Δ its first element δ_i^NC
    if (𝒦𝓑_0^Σ ∪ Π)↓_{≺δ_i} ∪{δ_i^NC} ⊭ NC ⊑ ⊥ then
        𝒦𝓑* := 𝒦𝓑* ∪ {NC ⊑ ⊥}
        extract all δ_k^NE with E = C from Δ
    else
        // δ_i^NC is actually overridden
        let δ := δ_i
        while Δ contains some δ_j^ND such that δ ≺ δ_j do
            extract from Δ the first such δ_j^ND
            if (𝒦𝓑_0^Σ ∪ Π)↓_{≺δ_j} ∪{δ_j^ND} ⊭ ND ⊑ ⊥ then
                𝒦𝓑* := 𝒦𝓑* ∪ {ND ⊑ ⊥}
                extract all δ_k^NE with E = D from Δ
                let δ := δ_j
```

CI-to-DI	naive	opt	mod	mod+opt
Gene Ontology				
05%	12.35	04.99	00.26	00.24
10%	24.12	09.49	00.28	00.26
15%	34.47	14.28	00.29	00.29
20%	41.96	19.70	00.32	00.31
25%	49.92	24.97	00.34	00.33
Fly Anatomy				
05%	4.22	1.90	00.13	00.12
10%	7.97	3.78	00.15	00.14
15%	11.95	5.59	00.17	00.16
20%	14.42	7.34	00.19	00.18
25%	17.46	9.18	00.21	00.20

Table 4.8. Impact of $|\mathscr{D}|$ on performance (sec) – DA rate = 15%

Synthetic-DI	naive	opt	mod	mod+opt
Gene Ontology				
05%	13.15	05.17	0.48	00.42
10%	27.77	09.85	0.83	00.64
15%	37.47	15.54	1.47	00.98
20%	46.11	21.16	2.76	01.54
25%	57.14	27.57	4.66	02.46
Fly Anatomy				
05%	4.86	02.01	0.40	0.35
10%	9.86	03.97	1.19	0.67
15%	14.75	06.01	2.51	1.18
20%	19.86	08.59	4.61	2.15
25%	24.27	10.80	7.25	3.34

Table 4.9. Impact of $|\mathscr{D}|$ on performance (sec) – DA rate = 15%

DA	naive	opt	mod	mod+opt
Gene Ontology				
05%	27.38	13.29	00.28	00.27
10%	27.52	13.79	00.29	00.28
15%	34.47	14.28	00.29	00.29
20%	38.57	14.99	00.30	00.29
25%	36.21	15.64	00.31	00.30
30%	42.37	16.22	00.32	00.31
Fly Anatomy				
05%	10.31	05.02	00.16	00.15
10%	11.46	05.38	00.17	00.16
15%	11.95	05.60	00.17	00.17
20%	12.13	05.85	00.18	00.17
25%	12.65	06.20	00.19	00.18
30%	13.79	06.51	00.20	00.19

Table 4.10. Impact of DAs on performance (sec) – CI-to-DI-rate = 15%

DA	naive	opt	mod	mod+opt
Gene Ontology				
05%	33.34	13.74	1.26	00.86
10%	35.63	14.73	1.35	00.92
15%	37.47	15.54	1.47	00.98
20%	42.05	16.68	1.51	01.00
25%	43.03	16.86	1.60	01.06
30%	47.03	17.88	1.66	01.08
Fly Anatomy				
05%	12.93	05.55	2.23	1.07
10%	14.01	05.84	2.50	1.14
15%	14.76	06.01	2.52	1.18
20%	15.94	06.30	2.61	1.22
25%	16.60	06.59	2.82	1.30
30%	17.69	07.01	2.98	1.36

Table 4.11. Impact of DAs on performance (sec) – Synthetic-DI-rate = 15%

$\mathscr{KB} \cap \mathscr{M}_0$. This combined method is correct by Theorem 4.3.1 and Theorem 4.2.14. Table 4.8 and 4.9 shows the impact of the number of DIs on response time for the two test suites, as DI rate ranges from 5% to 25%. The methods Mod and Mod+Opt are slightly less effective in the second suite probably because random defaults connect unrelated parts of the ontology, thereby hindering module extraction. In both suites, Opt's speedup factor (w.r.t. the naive method) is about two. On average, the combined method yields a further 23% improvement over Mod alone; the maximum reduction is 54% (2^{nd} suite, Synthetic-DI-rate=25%, DA-rate=15%). The additional conflicts induced by injected disjointness axioms have moderate effects on response time (cf. Table 4.10 and 4.11). Mod+Opt's average response time across both test suites is 0.7 sec., and the longest Mod+Opt response time has been 3.34 sec. Similar results have been obtained on the test suites containing non-empty ABox. In all cases, the speedups of Mod and Mod-Opt remain well above one order of magnitude.

Note, that in presence of non-N-free $\mathscr{DL}^{\text{N}}$ knowledge bases Opt and Mod+Opt are not applicable, in general. However, an additional benefit of mod* is that, by eliminating most normality concepts, it may enable the application of the optimistic method Opt, that is correct only when $|\Sigma| = 1$. Of course, the number of cases in which Opt becomes applicable significantly depends on the $\mathscr{KB}$ structure. The combination of mod* and opt has the following performance:

GO – CI-to-DI Opt is enabled in nearly all examples; the average speedup (when opt is applicable) is 5%;

GO – Synthetic-DI

$|\Sigma|$=50: Opt enabled in 20% cases; speedup 38.33%;
$|\Sigma|$=100: Opt enabled in 10% cases; speedup 38.36%;
$|\Sigma|$=150: Opt enabled in 10% cases; speedup 34.61%;

FLY – CI-to-DI Opt enabled in 22% cases; speedup 7%;

FLY – Synthetic-DI Opt is never enabled.

It appears that in FLY concepts are more densely interconnected, which hinders the removal of the normality predicates that do not occur in the query, hence the limited applicability of Opt. The new random connections introduced by the completely random DIs contained in the Synthetic-DI suite produce a similar effect.

4.4 Summary

The module-based and optimistic optimizations introduced in this chapter are sound and complete. In our experiments, the combined method (when applicable) and the module-based method make $\mathscr{DL}^{\mathrm{N}}$ reasoning at least one order of magnitude faster (and up to ∼780 times faster in some case).

The iterated module extractor, mod*, brings major speedups that make $\mathscr{DL}^{\mathrm{N}}$ reasoning with a large number of explicit normality concepts feasible in practice.

The conditional module extractor for nonempty ABoxes, cMod, is very effective when the ABox assertions are loosely interconnected, with speedups up to ∼ 75%. In the current random $\mathscr{DL}^{\mathrm{N}}$ testbed, the advantages of cMod tend to disappear when there are approximately 4 assertions per individual.

The conditional module extractor can be applied also to classical knowledge bases; on an excerpt of the OBO repository, so far, the average reduction of module size is promising (85%).

The query response times obtained with the N-free test suites or NC-rates up to 10% (that in our opinion exceed what should be expected in practice, given the specific role of explicit normality concepts) are compatible with real time $\mathscr{DL}^{\mathrm{N}}$ reasoning. Only the random dependencies introduced by synthetic DIs, combined with numerous restrictions of role ranges to normal individuals, can raise response time up to 83.5 seconds; in most of the other cases, computation time remains below 30 seconds. This is the first time such performance is reached over nonmonotonic KBs of this size: more than 20K concept names and over 30K inclusions.[10] Our approach brings technology closer to practical nonmonotonic reasoning with very large knowledge bases.

[10] Good results have been obtained also for KBs with ∼5200 inclusions under rational closure semantics [Casini et al., 2013a, Casini et al., 2014].

Chapter 5

Optimizing the Construction of Secure Knowledge Base Views

In recent years, Semantic Web technologies have become increasingly used to encode sensible knowledge on individuals, companies and public organizations. The most popular access control method protect sensitive data from unauthorized disclosure via direct accesses. However, they fail to prevent indirect data disclosure that may occur when sensitive information can be inferred from non-sensitive data and metadata [Farkas and Jajodia, 2002]. Despite the difficulties to develop techniques to detect potential inference vulnerabilities, no system can guarantee confidentiality of data without them. As reasoning techniques make it possible to extract implicit information, any access control method that does not deal with inference fails to ensure privacy [Abel et al., 2007, Flouris et al., 2010]. In particular, various sources of background knowledge can be exploited to reconstruct secrets. Background knowledge can be knowledge of the domain of interest, e.g. auxiliary ontologies, as well as meta knowledge about which kind of information the knowledge base is expected to represent. For instance, suppose a hospital allows to know whether a patient has been hospitalized but omits to reveal where, if she is in the infective disease ward. Since a hospital's $\mathscr{KB}$ is expected to have complete knowledge about which patients are in which ward, from the fact that John has been admitted to the hospital and yet he does not appear to be located in any ward, a user can reconstruct he is affected by some infection. In general,

meta knowledge helps in preventing *attacks to complete knowledge* and *attacks to the signature*.

To tackle the vulnerabilities arising from these scenarios, in Section 3.2 we provide a fully generic formalization of background knowledge and metaknowledge, a confidentiality model which neutralizes the inference-based attacks that exploit such knowledge, and – since the user's metaknowledge is not directly possessed by the knowledge engineer – a rule-based methodology to safely approximate it.

Regarding complexity issues, for the confidentiality model has been shown that by using Horn rules to encode the user's meta knowledge, if the underlying DL is tractable, then the filtering secure function is tractable too.[1] Although such promising theoretical properties suggest that the framework can be practically used, they are still to be assessed experimentally. In this chapter, we present SOVGen, a first prototype suited for a concrete e-health scenario. In particular, extensional data is encoded in realistic electronic health records conforming to the international standard HL7 v.3 - CDA Rel.2 We approximate the user's background knowledge with the SNOMED-CT ontology, together with an ontology establishing the mapping between SNOMED-CT concepts and ICD-9CM[2] and LOINC[3] codes that occur in the records. The user's meta knowledge, on the other hand, consists of *(i)* bridge metarules that permit to identify SNOMED-CT concepts starting from the specific encoding of the records required by CDA, as well as *(ii)* metarules that establish relationships between medications, diseases, medical procedures, etc.

In Section 5.1.1 we will describe the abstract algorithm underlying SOVGen. Section 5.2 and 5.3 introduce related optimizations. Sections 5.1.2 and 5.4 describe the experimental settings and performance analysis, respectively. Section 5.5 concludes the chapter.

5.1 Preliminary Experimental Analysis

This section introduces SOVGen, a prototypical implementation of the confidentiality model based on existing classical reasoners. A preliminary experimental performance analysis of this prototype is included; it uses test cases with realistic size and the optimization techniques supported by the underlying, classical reasoning engine. For the

[1] Non-Horn metarules can be safely approximated with Horn metarules; the price to pay is a loss of *cooperativeness*, i.e. a reduction of the information available to the user.

[2] http://www.who.int/classifications/icd/en/

[3] https://loinc.org/

purpose, synthetic test cases specifically designed to simulate the employment of the confidentiality model in a concrete e-health scenario have been generated in a principled way, as explained in Section 5.1.2.

5.1.1 SOVGen Abstract Algorithm

In this section we give a brief description of the abstract algorithm underlying SOVGen, the prototypical implementation of the confidentiality model illustrated in Section 3.2 based on Horn metarules.

By standard logic programming techniques, a minimal $K \subseteq PAX$ satisfying the set of metarules and the constraints K^+ can be obtained with the following polynomial construction:

$$K_0 = K^+, \quad K_{i+1} = K_i \cup \bigcup \{\, head(r) \mid r \in ground_{K_i}(MR) \wedge body(r) \subseteq Cn(K_i) \,\}.$$

It can be proved that the sequence limit $K_{|PAX|}$ satisfies $\langle K^+, K^- \rangle$ as well if $K_{|PAX|}$ does not entail an axiom in K^-. Then, for all $s \in S$, s activates the censor iff s is a consequence of $K_{|PAX|} \cup BK$. For further details refer to Section 3.2.

Algorithm 3 takes as input a knowledge base $\mathscr{KB}$, a set of secrets S, a set of metarules MR and the user's background knowledge BK. The output is the set of axioms that constitute a secure view for the user.

The sets M_M and M_G constitute a partition of MR based on the metarules' type (ground or containing metavariables). Iterating over the axioms $\alpha \in PAX$ (*lines 6-23*), at each step K collects all the axioms of PAX that does not contribute to the entailment of secrets. The repeat-until loop (*lines 9-16*) computes the deductive closure K' of K under the set of metarules MR[4]. In particular, for each ground metarule (*lines 10-12*) we evaluate a conjunctive query (encoded in *line 11*) in order to check if m body is satisfied by the current K'. Similarly, for each metarule containing metavariables (*lines 13-15*), we obtain all possible bindings for the metavariables in the body of m by means of a conjunctive query evaluation (*line 14*). The sequence of steps described above is iterated until a fix-point is reached (*line 16*). At this point the condition $Cn(K') \cap K^- = \emptyset$ is verified (*line 17*). It is now possible to determine the value of the censor for α. We first check that no secret is entailed from the minimal K (*line 18*) enriched with BK. Finally, we can safely include α in the view only if it is entailed by $\mathscr{KB}$ (*line 20*). Otherwise, the set K^- is updated (*line 23*).

[4]The result of Proposition 3.2.7 guarantees that considering only the minimal PKB is sound.

Algorithm 3:

Data: $\mathscr{KB}, S, MR, BK$.

$K_i^+, K_i^- \leftarrow \emptyset$;
$M_M \leftarrow \{r_i | r_i \in MR$ and r_i metarule containing metavariables$\}$;
$M_G \leftarrow \{r_i | r_i \in MR$ and r_i ground metarule$\}$;
$PAX \leftarrow \mathscr{KB} \cup \bigcup_{r \in ground_{\mathscr{KB}}(MR)} head(r)$;
$K \leftarrow BK$;
forall $\alpha \in PAX$ **do**
    $K' \leftarrow K \cup \{\alpha\}$;
    $M'_G \leftarrow M_G$;
    repeat
        forall $m \in M'_G$ **do**
            if $K' \models body(m)$ **then**
                $K' \leftarrow K' \cup \{head(m)\}$;
        forall $m \in M_M$ **do**
            forall $(a_0, \ldots, a_n) \mid K' \models body(m, [X_0/a_0, \ldots, X_n/a_n])$ **do**
                $K' \leftarrow K' \cup \{head(m, [X_0/a_0, \ldots, X_n/a_n])\}$;
    until *No element is added to* K';
    if $\{\beta \in K^- \mid K' \models \beta\} = \emptyset$ **then**
        if $\{s \in S \mid K' \cup BK \models s\} = \emptyset$ **then**
            if $\mathscr{KB} \models \alpha$ **then**
                $K^+ \leftarrow K^+ \cup \{\alpha\}$;
                $K \leftarrow K'$;
            else
                $K^- \leftarrow K^- \cup \{\alpha\}$;
return K_i^+

5.1.2 Experimental Settings

In this section we present synthetic test cases which have been specifically designed to simulate the employment of SOVGen in a e-health scenario as part of the *SmartHealt 2.0 Project*[5]. In particular, each test case represents the encoding of sensitive data in a CDA-compliant electronic health record. Clinical Document Architecture (CDA) is an international standard for information exchange, based on the Health Level 7 Reference

[5]The main goal of SmartHealt 2.0 Project is promoting innovation in the National Health System through the introduction of a new model of digital Healthcare.

Information Model[6] (HL7 RIM).

According to the theoretical framework each test case comprises four different components: the ontology $\mathscr{K}\mathscr{B}$ that contains confidential data to be protected; an ontology MR encoding the user metaknowledge with a set of metarules; a set S of secrets; a series of ontologies representing the user's background knowledge BK.

KB generation

KB is generated as a set of assertions instantiating the PS ontology. PS encodes a patient summary clinical document following the HL7 Implementation Guide for CDA Rel.2 – Level 3: Patient Summary. As it can be seen in Figure 5.1, PS currently provide a support for encoding information about (i) history of assumed medications; (ii) clinical problem list including diagnosis, diagnostic hypothesis and clinical findings; (iii) history of a family member disease; (iv) list of the procedures the patient has undergone; (v) list of relevant diagnostic tests and laboratory data. Note that, according to the CDA standards a disease in the PS ontology is represented by a ICD-9CM[7] code, pharmaceutical products and procedures are represented by a SNOMED CT codes, while diagnostic tests and laboratory data by LOINC[8] codes. For example, <code code="64572001" codeSystemName="SNOMED CT"/> stands for an instance of the SNOMED CT concept *Disease* ($SCT\_64572001$). The type of sections to be generated are randomly chosen among those mentioned above. A disease (resp. product, procedure, test) code to associate to the entries is chosen as a random leaf of the corresponding *Disease* (resp. Pharmaceutical/biologic product, Procedure by site, Measurement procedure, Imaging) concept of the SNOMED CT ontology. In case a disease code is needed, the ICD-9CM code corresponding to the SNOMED CT one is retrieved and the equivalence is added to a background knowledge ontology named EQIV-RL.

Metarule generation

The knowledge encoded in KB gives rise to several possible types of metarules. Bridge metarules associate a ICD-9CM/SNOMED CT/LOINC code to the concept in the respective ontology. For instance,

$$\mathrm{CD}(C),\ \mathrm{dtpCode}(C, 64572001),\ \mathrm{dtpCodeSystem}(C, \text{SNOMED-CT}) \Rightarrow \mathrm{SCT\_64572001}(C)$$

[6] http://www.hl7.org/
[7] International Classification of Diseases, 9th Revision, Clinical Modification
[8] A universal code system for tests, measurements, and observations.

makes it possible to derive that a code instance C is in fact an instance of the *Disease* concept in SNOMED CT.

The second type of metarules concerns the pharmaceutical products. The presence of a drug in the history of medication use implies that the patient suffers (certainly or with a great probability) from a specific pathology or has undertaken a specific procedure. Consider the following example of metarule which says that the presence of a medicine with active ingredient Phenytoin ($SCT\_40556005$) indicates that the patient suffers from some kind of Epilepsy ($SCT\_84757006$):

$$\begin{aligned}&\text{Patient}(P),\ \text{SubstanceAdministration}(SA),\ \text{Consumable}(C),\ \text{hasConsumable}(SA, C),\\ &\ \text{ManufacturedProduct}(MP),\ \text{hasManufacturedProduct}(C, MP),\ \text{Material}(M),\\ &\ \text{hasManufacturedMaterial}(MP, M),\ \text{SCT\_40556005}(CD),\ \text{hasCode}(M, CD)\\ &\Rightarrow \exists\text{suffer.SCT\_84757006}(P)\end{aligned}$$

The third type of metarules concerns the problems section. In particular the presence of a diagnosis (resp. diagnostic hypothesis) indicates that the patient suffer (resp. possibly suffer) a certain pathology.

$$\begin{aligned}&Patient(P), Section(S), hasCode(S, L-11450-4), Entry(E), hasEntry(S, E),\\ &Act(A), hasAct(E, A), EntryRelationship(ER), hasEntryRelationship(A, ER),\\ &Observation(O), hasObservation(ER, O), SCT_6 4572001(CD), hasCode(O, CD),\\ &SCT_x yz(V), hasValue(O, V)\\ &\Rightarrow suffer(P, V)\end{aligned}$$

Other types of metarules apply to the family history – e.g. a patient could be subject to a family members' disease – and the procedures section. For instance, the metarule

$$\text{Patient}(P),\ \text{Procedure}(I),\ \text{SCT\_77465005}(C),\ \text{hasCode}(I, C) \Rightarrow \text{subject}(P, C)$$

allows to entail that the presence of an organ transplantation (SCT_77465005) in the procedure section indicates that the patient is subject to transplantation.

Note that the generation of MR is not completely random for a part of the metarules. In order to obtain a nontrivial reasoning, during the KB generation, together with the creation of a section' entry is also created one or more corresponding bridge metarules and a metarule corresponding to the section in question. A second part of metarules are constructed by randomly selecting appropriate SNOMED CT concepts as needed. The adoption of such approach guarantees that at least part of the metarules are actually fired during the secure ontology view generation. Furthermore, observe that there are

```
<clinicalDocument>
   <recordTarget>
      <patientRole>
         <patient> . . . </patient>
      </patientRole>
   </recordTarget>
   <structuredBody>
      <section> <code code='10160-0' codeSystemName='LOINC'/> <!-- HISTORY OF MEDICATION USE -->
         <entry> . . . </entry>
      </section>
      <section> <code code='11450-4' codeSystemName='LOINC'/> <!-- CLINICAL PROBLEM LIST -->
         <entry> . . . </entry>
      </section>
      <section> <code code='10157-6' codeSystemName="LOINC"/> <!-- FAMILY MEMBER DISEASES -->
         <entry> . . . </entry>
      </section>
      <section> <code code='47519-4' codeSystemName='LOINC'/> <!-- HISTORY OF PROCEDURES -->
         <entry> . . . </entry>
      </section>
      <section> <code code='30954-2' codeSystemName="LOINC"/> <!-- RELEVANT DIAGNOSTIC TESTS -->
         <entry> . . . </entry>
      </section>
   </structuredBody>
</clinicalDocument>
```

Figure 5.1. HL7 CDA Rel.2 Patient Summary

actually two levels of metarules, the bridge metarules constitute a precondition for the activation of the others.

Secrets generation

The ontology S is randomly generated as a set of assertions of the types:

$$\exists \text{suffer}.X(p) \quad \exists \text{possiblySuffer}.X(p) \quad \exists \text{possibleSubject}.X(p) \quad \exists \text{subject}.Y(p)$$

where X (resp. Y) is chosen as a random subconcept of the *Disease* (resp. *Procedure*) concept of the SNOMED CT ontology.

Background knowledge

The background knowledge BK is approximated by means of the PS, SNOMED-CT and the previously mentioned EQIV-RL ontologies.

5.1.3 Experimental Results: Performance Analysis

In this section we present a preliminary experimental performance analysis of a naive implementation of the framework. Scalability evaluations have been carried out

on synthetic test cases generated according the settings described in Section 5.1.2. The size of $\mathscr{K}\mathscr{B}$ is given by the parameter *KB-size* as the number of assertions occurring in the ontology. Then, the size of MR, *MR-rate*, is the ratio between the number of metarules and the number of assertions in $\mathscr{K}\mathscr{B}$. Finally, the size of S is determined by the parameter *S-rate* that specifies the ratio $|S|/|KB|$.

The experiments were performed on an Intel Core i7 2,5GHz laptop with 16GB RAM and OS X 10.10.1. SOVGen was run on Java 1.8 with the options *-Xms8G -Xmx8G -Xss4G* to set the available RAM to 8GB and the stack memory space to 4GB.

As expected, given the amount of background knowledge (consider that SNOMED-CT describes about 300K concepts), the computation time of the secure ontology views exceeded a 30 minutes time out in all executions. Thus, the use of suitable optimization techniques proves mandatory in order to achieve usability in practice.

5.2 Module Extraction for Background Knowledge

In recent years, OWL ontologies have been used in several countries to describe electronic patient records (EPR). Patients' data typically involves descriptions of human anatomy, medical conditions, drugs, and so on. These domains have been described in well-established reference ontologies such SNOMED-CT, GALEN, NCI, etc.

Such foreign medical ontologies are usually huge, so importing a whole ontology would make the consequences of the additional information costly to compute. In practice, therefore, one may need to extract a (ideally small) fragment $\mathscr{M}$ of the external medical ontology - a module - that includes only the relevant background information, i.e. describes just the concepts that are used in $\mathscr{K}\mathscr{B}$, $\mathscr{S}$ and MR.

We first recall the definition of modules in terms of locality:

Definition 5.2.1 (Locality-based Modules [Grau et al., 2008]) Let $\mathscr{M}$ and KB be ontologies, and *Sig* a signature. We say that $\mathscr{M}$ is a $\bot$-module ($\top$-module) for *Sig* in $\mathscr{K}\mathscr{B}$ if $\mathscr{K}\mathscr{B} \setminus \mathscr{M}$ is $\bot - local(\top - local) w.r.t. \widetilde{\mathscr{M}} \cup Sig$.

An important property of locality-based modules which determines their scope is the following: suppose that M_1 (M_2) is a $\bot$-module ($\top$-module) for a signature *Sig* in $\mathscr{K}\mathscr{B}$, then M_1 (M_2) will contain all superconcepts (subconcepts) in $\mathscr{K}\mathscr{B}$ of all concepts in *Sig*:

Proposition 5.2.2 ***(Scope of a Module [Grau et al., 2008])*** *Let $\mathscr{K}\mathscr{B}$ be a knowledge base, A and B be concept names in $\mathscr{K}\mathscr{B} \cup \{\top\} \cup \{\bot\}$, $\alpha := (A \sqsubseteq B)$, $\beta := (B \sqsubseteq A)$,*

and $\mathscr{M}_A \subseteq \mathscr{K}\mathscr{B}$ *with* $A \in Sig$. *If* $\mathscr{M}_A$ *is a* $\bot$*-module in* $\mathscr{K}\mathscr{B}$ *for* Sig, *then* $\mathscr{M}_A \models \alpha$ *iff* $\mathscr{K}\mathscr{B} \models \alpha$, *If* $\mathscr{M}_A$ *is a* $\top$*-module in* $\mathscr{K}\mathscr{B}$ *for* Sig, *then* $\mathscr{M}_A \models \beta$ *iff* $\mathscr{K}\mathscr{B} \models \beta$.

As we have already seen in Section 4.2 the extraction of $\top$-modules or $\bot$-modules may introduce symbols not in Sig that are potentially unnecessary. To make the module as small as possible, we nest the extraction of $\top$-module and $\bot$-module until a fix-point is reached. The module obtained at the end of this process still satisfies the module coverage guarantee of Sig in $\mathscr{K}\mathscr{B}$:

Proposition 5.2.3 ***([Sattler et al., 2009])*** *Let* $\mathscr{K}\mathscr{B}$ *be a knowledge base,* Sig *a signature in* $\mathscr{K}\mathscr{B}$, A *and* B *be concept names in* Sig, *with* $\{A, B\} \subseteq Sig$, $\alpha := (A \sqsubseteq B)$, *and* $\mathscr{M} \subseteq \mathscr{K}\mathscr{B}$. *If* $\mathscr{M}$ *is a* $\top\bot^*$*-module in* $\mathscr{K}\mathscr{B}$ *for* Sig, *then* $\mathscr{M} \models \alpha$ *iff* $\mathscr{K}\mathscr{B} \models \alpha$,

Further important characteristic which make $\top\bot^*$-modules suitable in our ontology reuse scenario is that they are minimal *self-contained*[9] and *depleting*[10].

Proposition 5.2.4 ***([Sattler et al., 2009])*** *Let* $\mathscr{K}\mathscr{B}$, $\mathscr{K}\mathscr{B}'$ *be knowledge base and* Sig, Sig' *be signatures.*

1. *If* $Sig \subseteq Sig'$, *then* $\top\bot^* - \mathrm{Mod}(Sig, \mathscr{K}\mathscr{B}) \subseteq \top\bot^* - \mathrm{Mod}(Sig', \mathscr{K}\mathscr{B})$.
2. *If* $\mathscr{K}\mathscr{B} \subseteq \mathscr{K}\mathscr{B}'$, *then* $\top\bot^* - \mathrm{Mod}(Sig, \mathscr{K}\mathscr{B}) \subseteq \top\bot^* - \mathrm{Mod}(, \mathscr{K}\mathscr{B}')$.

The above result easily implies the following proposition:

Proposition 5.2.5 *Let* $\mathscr{K}\mathscr{B}$, $\mathscr{K}\mathscr{B}'$ *be knowledge base and* Sig *a signature. Then* $\top\bot^* - \mathrm{Mod}(Sig, \mathscr{K}\mathscr{B}) \cup \top\bot^* - \mathrm{Mod}(Sig, \mathscr{K}\mathscr{B}') \subseteq \top\bot^* - \mathrm{Mod}(Sig, \mathscr{K}\mathscr{B} \cup \mathscr{K}\mathscr{B}')$. *Moreover, there exists* $\mathscr{K}\mathscr{B}$ *and* $\mathscr{K}\mathscr{B}'$ *s.t. the above inclusion is strict.*

Proof. Let $M = \top\bot^* - \mathrm{Mod}(Sig, \mathscr{K}\mathscr{B} \cup \mathscr{K}\mathscr{B}')$, $M_1 = \top\bot^* - \mathrm{Mod}(Sig, \mathscr{K}\mathscr{B})$ and $M_2 = \top\bot^* - \mathrm{Mod}(Sig, \mathscr{K}\mathscr{B}')$. The inclusion follows from $\mathscr{K}\mathscr{B} \subseteq \mathscr{K}\mathscr{B} \cup \mathscr{K}\mathscr{B}'$ and $\mathscr{K}\mathscr{B}' \subseteq \mathscr{K}\mathscr{B} \cup \mathscr{K}\mathscr{B}'$ applying Proposition 5.2.4 .

To prove the second part of the claim consider the following knowledge basses:

$\mathscr{K}\mathscr{B}_1 = \{A \sqsubseteq \exists R.C,\ C \sqsubseteq B\}$,

$\mathscr{K}\mathscr{B}_2 = \{C \sqsubseteq A\}$,

and the signature $Sig = \{\mathrm{A}, \mathrm{R}\}$. We have $\top\bot^* - \mathrm{Mod}(Sig, \mathscr{K}\mathscr{B}_1) = \{A \sqsubseteq \exists R.C\}$, $\top\bot^* - \mathrm{Mod}(Sig, \mathscr{K}\mathscr{B}_2) = \emptyset$, and $\top\bot^* - \mathrm{Mod}(Sig, \mathscr{K}\mathscr{B}_1 \cup \mathscr{K}\mathscr{B}_2) = \{A \sqsubseteq \exists R.C,\ C \sqsubseteq A\} \supset \top\bot^* - \mathrm{Mod}(Sig, \mathscr{K}\mathscr{B}_1) \cup \top\bot^* - \mathrm{Mod}(Sig, \mathscr{K}\mathscr{B}_2)$. ∎

[9] A module $M \subseteq \mathscr{K}\mathscr{B}$ is self-contained if $\mathscr{M}$ preserve all the entailments over $Sig \cup \mathscr{M}$, i.e., $\mathscr{M}$ is indistinguishable w.r.t. $Sig \cup \mathscr{M}$ from $\mathscr{K}\mathscr{B}$.

[10] A module $M \subseteq \mathscr{K}\mathscr{B}$ is depleting if $\mathscr{K}\mathscr{B} \setminus \mathscr{M}$ has no non-trivial entailments over Sig , i.e., if the set of axioms in $\mathscr{K}\mathscr{B} \setminus \mathscr{M}$ is indistinguishable w.r.t. Sig from the empty set.

As we have already discussed in Section 3.2.2 the background knowledge of a user can be estimated collecting as many public sources of formalized relevant knowledge such as ontologies and triple stores as possible. The larger the user background knowledge taken in consideration during the secure view computation is, the smaller is the probability that a secret can be leaked. Unfortunately, as the preliminary experimental analysis reported in Section 5.1.3 have proved, higher safety guarantees may cost the practical usability of the confidentiality model.

As a consequence, the presence of (very) large background knowledge bases (such as SNOMED-CT) makes it desirable to apply a process of modularization designed to reduce the time of secure view computation. In fact, many of the axioms in a large BK are reasonably expected to be irrelevant to the given view.

The correct way to use *locality-based module extractors* [Sattler et al., 2009, Grau et al., 2008] in order to make reasoning focus on relevant background knowledge only is described below. Assume that the user background knowledge comprises a set $BK_1, \ldots, BK_n$ of knowledge bases. We can extract the fragments relevant in the construction of a secure view by:

1. building the logical union BK of the axioms occurring in the different BK_i under the standard semantics
2. extracting a $\top\bot^*$-module from BK w.r.t. a signature $\Sigma = \widetilde{K}B \cup \widetilde{S} \cup \widetilde{MR}$

Performing *step 1* corresponds in practice to importing all the different $BK_1, \ldots, BK_n$ in a single background ontology BK. While it may seem more convenient, extracting separately a $\top\bot^* - \mathrm{Mod}(\Sigma, \mathscr{K}\mathscr{B}_i)$ from each BK_i, this may result in missing some relevant axioms as proved in Proposition 5.2.5. This may compromise the confidentiality of the generated view[11].

Experimental results (cf. Section 5.4) will show that the modules extracted are on average two or three orders of magnitude smaller than the initial BK which drastically improves performance.

5.3 Metarule Evaluation

The presence of technologies that permit native conjunctive query evaluation reveals fundamental to efficiently process users' metaknowledge. Unfortunately, the OWL rea-

[11] The incompleteness of the relevant background knowledge may prevent the detection of the violation of some secrets in *line 18* of Algorithm 3.

soners publicly available do not offer native support [12].

Straightforward evaluation of metarules in the presence of metavariables with an OWL reasoner would need to consider all possible ways of uniformly replacing metavariables by individual constants occurring in the ontology. Thus, as the ontology ABox grow, metarule evaluation can easily become unmanageable in terms of execution time.

Nowadays SPARQL[13], constitute a de facto standard when it comes to conjunctive query answering. It has been recently extended with the OWL Direct Semantics Entailment Regime in order to permit reasoning over OWL ontologies. Unfortunately, few tools provide support to this new semantics. An important tool with this feature is *Apache Jena Semantic Web Toolkit*[14]. A valid alternative to the consolidated SPARQL engines seems to be *OWL-BGP*[15], a relatively new framework for parsing SPARQL basic graph patterns (BGPs) to OWL object representation and their assessment under the OWL Direct Semantics Entailment Regime. *OWL-BGP* incorporates various optimization techniques [Kollia and Glimm, 2013] including query rewriting and a sophisticated cost-based model[16] for determining the order in which conjunctive query atoms are evaluated. As we will see in Section 5.4 the performance of the query evaluation module of SOVGen is unacceptable when Jena is used and not quite satisfactory when *OWL-BGP* is adopted[17].

As an alternative to the above frameworks for conjunctive query evaluation we propose an ad hoc module, called *Metarule Evaluation Engine (MEE)*, that aims to take advantage of the specific nature of the Horn metarules and incremental reasoning techniques of ELK [Kazakov et al., 2012, Kazakov et al., 2014, Kazakov and Klinov, 2013]. In particular, for each α in the enumeration of PAX, the incremental reasoner is expected to restrict reasoning to the new inferences triggered by α without repeating the inferences that involve only K^{+}_{i-1}.

In order to simplify our description of the procedure employed we first need to provide some formal definitions.

Definition 5.3.1 Let $\mathsf{N_C}$, $\mathsf{N_R}$ and $\mathsf{N_I}$ be the disjoint union of countably infinite sets of

[12] A partial exception of this rule is the Pellet reasoner. However, the Pellet's query engine – capable of answering only ABox queries – seems not to have been re-engineered for the last few years.

[13] http://www.w3.org/TR/sparql11-overview/

[14] http://jena.apache.org/

[15] https://code.google.com/p/owl-bgp/

[16] The cost calculation is based on information about instances of concepts and roles extrapolated from an abstract model built by reasoners that implement Tableaux reasoning algorithms.

[17] Note that evaluation of ground metarules results in *SPARQL ASK* query (*line 11* of Alg.3), while evaluation of metarules with metavariables in *SPARQL SELECT* query (*line 14* of Alg.3).

concept, role and individual names and $\mathsf{V_I}$ a set of individual variables (called metavariables). The set of individual terms consists of the disjoint union of $\mathsf{N_I}$ and $\mathsf{V_I}$.

An axiom template τ has the form $A(t_a)$, $R(t_a, t_b)$, where A is a concept name, R a role name, t_a and t_b individual terms.

An axiom α is a subsumption $A \sqsubseteq B$ with A and B are concept names or an axiom template that does not contain any metavarible.

A conjunctive query q is a finite set of axioms and axiom templates. If q contains no axiom template we call it ground.

Definition 5.3.2 (Mapping) A mapping is a (partial) function : $\mathsf{V_I} \rightarrow \mathsf{N_I}$. The domain of a mapping μ is denoted by $dom(\mu)$. We say that two mappings μ_1 and μ_2 are compatible if for all variables $x \in dom(\mu_1) \cap dom(\mu_2)$ we have that $\mu_1(x) = \mu_2(x)$. The join of two compatible mappings μ_1 and μ_2 is the mapping $\mu = \mu_1 \bowtie \mu_2$ defined as follows:

1. $\mu(x) = \mu_1(x) = \mu_2(x)$ if $x \in dom(\mu_1) \cap dom(\mu_2)$,
2. $\mu(x) = \mu_i(x)$ if $x \in dom(\mu_i)$ and $x \notin dom(\mu_j)$, with $i, j \in \{1, 2\}$ and $i \neq j$.

If $\mathscr{U}_1$ and $\mathscr{U}_2$ are two sets of mappings then with a slight abuse of notation we write

$$\mathscr{U}_1 \bowtie \mathscr{U}_2 = \{\mu_1 \bowtie \mu_2 \mid \mu_1 \in \mathscr{U}_1,\ \mu_2 \in \mathscr{U}_2 \text{ and } \mu_1 \text{ is compatible with } \mu_2\}.$$

Definition 5.3.3 Let $\mathscr{I}$ be a standard DL interpretation, $\mathscr{KB}$ a knowledge base and q a query. A mapping μ is an answer for a query q w.r.t. a knowledge base $\mathscr{KB}$, written $\mathscr{KB}, \mu \models q$, if for each $\mathscr{I}$ s.t. $\mathscr{I} \models \mathscr{KB}$, $\mathscr{I}$ satisfies all ground instantiations of the axiom templates of q obtained by replacing each $x \in dom(\mu)$ with $\mu(x)$. The set of answers to a query w.r.t. to $\mathscr{KB}$ is given by $ans(q, \mathscr{KB}) = \{\mu \mid \mathscr{KB}, \mu \models q\}$.

Note, that for a query $q = \{\tau_1, \tau_2\}$ $ans(q, \mathscr{KB}) = ans(\{\tau_1\}, \mathscr{KB}) \bowtie ans(\{\tau_2\}, \mathscr{KB})$ (follows directly from Definition 5.3.3).

Optimized algorithm for evaluation of a single metarule

Algorithm 4 describes the core of MEE, that is, the optimized procedure employed for the evaluation of a single metarule. More precisely, Algorithm 4 takes as input a knowledge base K' and a metarule m, and returns the set of mappings μ such that $K' \models \mu(body(m))$ (i.e. such that the corresponding instance of m fires).

Algorithm 4:

Data: $\mathscr{K}', m \in MR$
where $\mathscr{K}'$ is the same as in lines(11, 14) of Algorithm 3, and
$body(m) = \{\alpha_1, \ldots, \alpha_n, \tau_1, \ldots, \tau_m\}$.

`/* Evaluation of the ground part of the body. */`

1 **forall** $\alpha \in \{\alpha_1, \ldots, \alpha_n\}$ **do**
2 **if** $K' \models \alpha$ **then**
3 $body(m) \leftarrow body(m) \setminus \{\alpha\}$;
4 **else**
5 **return** $\emptyset$;

`/* Evaluation of the non ground part of the body. */`

6 $\mathscr{U}_1 = ans(\{\tau_1\}, K')$;
7 **forall** $i = \{2, \ldots, m\}$ **do**
8 $\mathscr{U}_i = \mathscr{U}_{i-1} \bowtie ans(\{\tau_i\}, K')$;
9 **if** $\mathscr{U}_i = \emptyset$ **then**
10 **return** $\emptyset$;
11 **return** $\mathscr{U}_m$;

We first illustrate in more detail how $ans(\cdot, \cdot)$ is computed in *lines 6, 8*. If $\tau = A(x)$ than it is possible to retrieve the solutions directly from the reasoner by using the instance retrieval reasoning service for A.

Unfortunately, there are no reasoning services that permit to retrieve the instances of a template $R(x, y)$ where both x and y are variables. In that case the candidate instances can be restricted to those compatible with the mappings computed for the previous template. Therefore in the actual implementation of *line 8* of Algorithm 4 the computation of $ans(\{\tau_i\}, K')$ is replaced with a computation of all $ans(\{\mu(\tau_i)\}, K')$ for all $\mu \in \mathscr{U}_{i-1}$.

Line 4 perform another optimization: it prevents the same α_i in the body of a metarule m to be re-evaluated whenever another metarule m' fires which could trigger m in turn (so m's body should be evaluated another time). This requires a change in Algorithm 3: if the current $\alpha \in PAX$ is discovered to entail a secret and must be retracted, then all the α_i that have been removed from $body(m)$ must be restored because they might not be derivable anymore after the retraction. In particular, an *else* clause with this purpose must be added to the *if* statement on *line 18* of Algorithm 3.

S-rate	05%	25%	50%	75%	100%
MEE	27	34	43	54	61
OWL-BGP	114	515	1619	2730	3805

Figure 5.2. View construction time with as the number of secrets increase (KB-size=200, MR-rate=10%)

Ordering euristics

The evaluation of metarules with metavariables, comprises a preprocessing step that partitions the axiom templates $\tau_1, \ldots, \tau_m$ in the metarules body in sets of *connected components*. Within a component, axiom templates share common metavariables, while there are no metavariables shared between atoms belonging to different *connected components*. Evaluating together templates belonging to non-related components increases unnecessarily the amount of intermediate results, whereas it is sufficient to combine the results for the single components. Furthermore, within each connected component, the evaluation is performed in a precise order. The templates τ_i of the type $A(x)$ are processed first in order to restrict as much as possible the compatible mappings for the metavariables occurring in the templates $R(x, y)$ whose answers are not directly computed by ELK's native methods (cf. the implementation of *line 8* of Algorithm 4, discussed in the previous paragraph).

Another optimization of this type concerns the order in which the axioms $\alpha \in PAX$ are evaluated in *line 6* of Algorithm 3. It addresses the fact that checking whether $\{\beta \in K^- \mid K' \models \beta\} = \emptyset$ (*line 17*) is time consuming, therefore it is convenient to keep K^- as small as possible. A simple analysis of the definition of the sequence $\langle K_i^+, K_i^- \rangle_{i \geq 0}$ (cf. Section 3.2.1) shows that the axioms $\alpha \in \mathscr{KB}$ cannot possibly enter K^-, so by processing those α first, we keep K^- empty until the first $\alpha \notin \mathscr{KB}$ is processed (consequently, in this first phase, the cost of the test in *line 17* is negligible).

5.4 Performance Analysis

In the following we present a performance analysis of a version of SOVGen that incorporates the optmization techniques introduced in Section 5.2 and 5.3. Scalability evaluations have been carried out according the experimental setup described in Section 5.1.3.

Given the amount of background knowledge (consider that SNOMED-CT describes

MR-rate	05%	10%	15%	20%	25%
MEE	23	34	41	48	52
OWL-BGP	420	515	661	769	858

Figure 5.3. View construction time as the number of metarules increase (KB-size=200, S-rate=25%)

KB-size	100	200	300	400	500
MEE	20	34	46	51	57
OWL-BGP	432	515	1063	1254	2136

Figure 5.4. View construction time as the size of KB increase (MR-rate=10%, S-rate=25%)

about 300K concepts) the use of module extraction techniques improves the computation time of two–three orders of magnitude at a cost of about 30 sec of overhead. As a concrete example, for *KB-size = 200*, *MR-rate = 20%* and *S-rate = 25%*, the secure view computation using the MEE module employs 35 sec if module extraction is used, 2198 sec otherwise.

In Figures 5.2–5.4 , the first (resp. second) row shows the experimental results obtained by using MEE (resp. OWL-BGP) to evaluate metarules – no result for Jena is reported as the execution time on all the test cases exceeded 1 hour time-out. Figure 5.2 reports the execution time as the amount of secrets grows. Both *MR-rate* and *KB-size* are fixed, respectively to 10% and 200 assertions. Note that, MEE outperforms OWL-BGP of 1–2 orders of magnitude. Figure 5.3 shows the impact of MR-rate on the performance of the secure view construction when *KB-size* is fixed to 200 and *S-size* to 10%. Here, MEE runs about 10 times faster than OWL-BGP. Finally, Figure 5.4 illustrate the way the execution time changes as the the size of $\mathscr{KB}$ increases. Again MEE is 10^2 faster than OWL-BGP.

The reason why a relatively simple query answering engine like MEE outperforms the others is that MEE does not restart reasoning from scratch at each step of the repeat-until loop. The sequence of calls submitted to the ELK reasoner, exploit the incremental classification facility of ELK. OWL-BGP, instead, re-computes its cost model at each step, which slows down significantly answer computation.

5.5 Summary

In Section 3.2 a confidentiality model has been introduced which adapts Controlled Query Evaluation to the context of Description Logics, and extends it by taking into account object-level and meta background knowledge. In this chapter, we have presented SOVGen, a first implementation of this methodology that has been specialized to deal with a concrete e-health application. In order to maximize performance, we have compared different reasoning tools and designed several optimization techniques. Then, we assessed SOVGen experimentally by using realistic electronic health records that refer to SNOMED-CT concepts, and Horn rules to represent meta knowledge. In particular, we observed that *(i)* module extraction reduces the secure-view computation time of several orders of magnitude, and *(ii)* the ad hoc (and relatively simple) answer computation method adopted by the MEE metarule evaluator – that intensively exploits ELK's incremental reasoning facility – outperforms the other query evaluation engines that rely on cost models. Whether these two approaches can be profitably combined is an interesting direction for further research.

Considering that secure views are constructed off-line – so that no overhead is placed on user queries – performance analysis shows that SOVGen is already compatible with practical use in this application scenario.

Chapter 6

Conclusions and Future Work

The increasing adoption of semantic technologies and the corresponding increasing complexity of application requirements are motivating extensions to the standard reasoning paradigms and services supported by such technologies. This thesis focuses on two of such extensions: nonmonotonic reasoning and inference-proof access control.

Concerning the former, we focused on the novel logic $\mathscr{DL}^{\mathrm{N}}$ because it solves several problems affecting previous approaches, and because it is flexible, that is, it is neutral with respect to the inferences that are not always desired and gives knowledge engineers the ability of switching those inferences on and off.

In Chapter 4 we introduced an implementation of $\mathscr{DL}^{\mathrm{N}}$ reasoning and several possible optimizations, that have been systematically assessed. Preliminary experimental scalability tests (cf. Section 4.1.5) on a semi-naive implementation of $\mathscr{DL}^{\mathrm{N}}$ (relying only on the optimization techniques of the underlying classical reasoner) yield promising results. Still, as defeasible inclusions rate grows, query response time slows down enough to call for improvements. In Chapter 4 we introduce module-based and optimistic optimizations that are sound and complete, where the latter applies only if the knowledge base is N-free. In particular:

- Many of the axioms in a large KB are expected to be irrelevant to the given query. In Section 4.2 we investigate the use of *module extractors* [Sattler et al., 2009, Grau et al., 2008] in $\mathscr{DL}^{\mathrm{N}}$ in order to focus reasoning on relevant axioms only. The approach is not trivial and requires an articulated correctness proof (module extractors are unsound for most nonmonotonic logics).

- In Section 4.3 we introduce a new algorithm for query answering, that is expected to exploit incremental reasoners at their best. Incremental reasoning is crucial as $\mathscr{DL}^{\mathrm{N}}$'s reasoning method iterates consistency tests on a set of KBs with large intersections. The main idea behind the *optimistic reasoning method* is to try to reduce the number of retractions (an expensive class of operations in incremental reasoning).

The experimental evaluation reported in Section 4.2.1 and 4.2.2 prove that module extraction is highly effective in speeding up reasoning in $\mathscr{DL}^{\mathrm{N}}$. Applying module extraction makes $\mathscr{DL}^{\mathrm{N}}$ reasoning at least one order of magnitude faster (and up to ∼780 times faster in some case). The optimistic reasoning method speedup factor (w.r.t. the naive implementation) is about two (cf. Section 4.3.1).

However, there are cases in which module extraction techniques should be improved. Currently, such methods are less effective when the knowledge base has nonempty ABoxes; this phenomenon is amplified in the nonmonotonic description logic $\mathscr{DL}^{\mathrm{N}}$, where reasoning requires repeated classifications of the knowledge base.

A new module extraction algorithm introduced in Section 4.2.3 constitutes a further contribute to the research on module extraction. The algorithm discards significantly more axioms in the presence of nonempty ABoxes and is not specific to $\mathscr{DL}^{\mathrm{N}}$ but applies also to classical DL reasoning. The method is correct under the assumption that the knowledge base is consistent; this hypothesis, in practice, is compatible with some of the main intended uses of module extraction, such as importing selected parts of already validated knowledge bases.

The experimental evaluation shows that the conditional module extractor for nonempty ABoxes, is very effective when the ABox assertions are loosely interconnected, with speedups up to $\sim 75\%$. In the current random $\mathscr{DL}^{\mathrm{N}}$ testbed, the advantages of cMod tend to disappear when there are approximately 4 assertions per individual. As far as the application to classical knowledge bases is concerned, on an excerpt of the OBO repository, the average reduction of module size is promising (85%).

The test case generator adoperated in the experimental validation of the $\mathscr{DL}^{\mathrm{N}}$ optimizations should be considered as a contribution of this thesis, as well. Its output has been analyzed in depth to verify that the synthetic ontologies it constructs and their classification are not trivial. The test case generator and the above validation criteria will hopefully be of help for other researchers in this field, where real nonmonotonic knowledge bases are not yet available.

To sum up, the query response times obtained in the extensive experimental analysis with N-free test suites or NC occurring in up to 10% of the DIs (that in our opinion

exceeds what should be expected in practice, given the specific role of explicit normality concepts) are compatible with real time $\mathscr{DL}^{\mathrm{N}}$ reasoning. Only the random dependencies introduced by synthetic DIs, combined with numerous restrictions of role ranges to normal individuals, can raise response time up to 83.5 seconds; in most of the other cases, computation time remains below 30 seconds. This is the first time a real-time performance is reached over nonmonotonic KBs of this size: more than 20K concept names and over 30K inclusions.

As a future work we plan to try different parallelization strategies, based on suitable reorderings of the operations executed by Algorithm 1. There are further possible optimizations for $\mathscr{DL}^{\mathrm{N}}$, such as caching the translations used for previous queries. An interesting open question is whether the sequences $\langle A_{i,j}\rangle_j$ can be refined so as to make the conditional module extractor more effective on highly interconnected ABoxes.

Another topic that deserves further attention is nonmonotonic conjunctive query answering (especially in the context of the DL-lite family). We expect the nice properties of subsumption and instance checking in $\mathscr{DL}^{\mathrm{N}}$ to carry over to this class of queries.

Last but not least, we are progressively extending the set of experiments by covering the missing cases and by widening the benchmark set, using real ontologies different from GO and FLY as well as completely synthetic ontologies.

Concerning secure ontology view construction, we adopted the confidentiality model introduced in [Bonatti and Sauro, 2013] that protects knowledge bases from attacks based on background knowledge as well as metaknowledge. We introduced an implementation of secure view construction, called SOVGen, in Chapter 5.

A preliminary performance assessment proves that a naive implementation of secure view construction is practically infeasible despite the theoretical tractability of the approach. The experiments have been conducted on test cases specifically designed to simulate the employment of the methodology in a concrete e-health scenario (cf. Section 5.1.2). In particular, each test case represents the encoding of sensitive data in a CDA-compliant electronic health record based on HL7 RIM and the user's background knowledge is in part approximated with the SNOMED-CT ontology.

In order to maximize performance, several optimization techniques have been designed:

- The presence of very large background knowledge bases suggests to apply a process of modularization designed to reduce the time of secure view computation. In fact, many of the axioms in a large BK, e.g. consider SNOMED CT, are reasonably expected to be irrelevant to a given view. The correct way to use

locality-based module extractors [Sattler et al., 2009, Grau et al., 2008] in order to make reasoning focus on relevant background knowledge only is illustrated in Section 5.2.

- Processing of the users' metaknowledge calls for technologies that permit native conjunctive query evaluation. As an alternative to the few available frameworks, in Section 5.3 we propose an ad hoc module that aims to take advantage of the specific nature of the (Horn) metarules and incremental reasoning techniques of the underlying classical reasoner [Kazakov et al., 2012, Kazakov et al., 2014, Kazakov and Klinov, 2013].

Experimental evaluation of the optimized framework implementation based on Horn metarules proved that: *(i)* module extraction reduces the secure-view computation time of several orders of magnitude, and *(ii)* the ad hoc (and relatively simple) answer computation method adopted by the proposed metarule evaluator – that intensively exploits ELK's incremental reasoning facility – outperforms the available query evaluation engines that rely on cost models (i.e. OWL-BGP and Jena). Whether these two approaches can be profitably combined is an interesting direction for further research. Considering that secure views are constructed off-line – so that no overhead is placed on user queries – performance analysis shows that SOVGen is already compatible with practical use in this application scenario.

As a future work, we aim to extend the system to general (meta)rules. A further interesting direction is to extend the test set with ABoxes of significant size in order to verify the efficiency of the conditional module extractor for nonempty ABoxes in the context of a second non standard reasoning service.

The above discussion shows that reasoning in $\mathscr{DL}^{\mathrm{N}}$ and secure view construction have in common two features: *(i)* they are applied to large or very large knowledge bases, and *(ii)* they involve repeated calls to a classical reasoner over knowledge bases that are related to each other: either they increase monotonically or they are rolled back to a previous state (e.g. when a defeasible inclusion is overridden or an axiom entails a secret). Point *(i)* is effectively addressed by means of module extraction techniques, while point *(ii)* needs efficient incremental reasoning algorithms. We have already pointed out some interesting directions to extend module extraction methods. Concerning incrementality, the available algorithms are designed for more general scenarios, especially as axiom retraction is concerned. In particular they are designed for arbitrary retractions, while our applications only require rollbacks, that could be implemented in more efficient way. Specialized implementations of rollbacks may speed up both $\mathscr{DL}^{\mathrm{N}}$ reasoning and

secure view construction, and as such they constitute an interesting direction for further research.

Bibliography

[Abel et al., 2007] Abel, F., Coi, J. L. D., Henze, N., Koesling, A. W., Krause, D., and Olmedilla, D. (2007). Enabling advanced and context-dependent access control in RDF stores. In Aberer et al., K., editor, *ISWC/ASWC*, volume 4825 of *LNCS*, pages 1–14. Springer.

[Artale et al., 2009] Artale, A., Calvanese, D., Kontchakov, R., and Zakharyaschev, M. (2009). The DL-lite family and relations. *J. Artif. Intell. Res. (JAIR)*, 36:1–69.

[Baader, 2003a] Baader, F. (2003a). Restricted role-value-maps in a description logic with existential restrictions and terminological cycles. In *In Proc. DL'03, http://CEUR-WS.org/Vol-81.*

[Baader, 2003b] Baader, F. (2003b). Terminological cycles in a description logic with existential restrictions. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, pages 325–330, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Baader et al., 2005a] Baader, F., Brandt, S., and Lutz, C. (2005a). Pushing the EL envelope. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence, IJCAI-05*, pages 364–369. Professional Book Center.

[Baader et al., 2005b] Baader, F., Brandt, S., and Lutz, C. (2005b). Pushing the EL envelope. LTCS-Report. Technical Report LTCS-05-01, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology, Germany.

[Baader et al., 2008] Baader, F., Brandt, S., and Lutz, C. (2008). Pushing the EL envelope further. In Clark, K. and Patel-Schneider, P. F., editors, *Proceedings of the*

Fourth OWLED Workshop on OWL: Experiences and Directions, Washington, DC, USA, 1-2 April 2008., volume 496 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Baader et al., 1996] Baader, F., Buchheit, M., and Hollander, B. (1996). Cardinality restrictions on concepts. *Artificial Intelligence*, 88(1):195 – 213.

[Baader et al., 2010] Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F. (2010). *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, New York, NY, USA, 2nd edition.

[Baader and Hollunder, 1995a] Baader, F. and Hollunder, B. (1995a). Embedding defaults into terminological knowledge representation formalisms. *J. Autom. Reasoning*, 14(1):149–180.

[Baader and Hollunder, 1995b] Baader, F. and Hollunder, B. (1995b). Priorities on defaults with prerequisites, and their application in treating specificity in terminological default logic. *J. Autom. Reasoning*, 15(1):41–68.

[Baader et al., 2009] Baader, F., Knechtel, M., and Peñaloza, R. (2009). A generic approach for large-scale ontological reasoning in the presence of access restrictions to the ontology's axioms. In *International Semantic Web Conference*, pages 49–64.

[Baader et al., 2006] Baader, F., Lutz, C., and Suntisrivaraporn, B. (2006). CEL - a polynomial-time reasoner for life science ontologies. In Furbach, U. and Shankar, N., editors, *IJCAR*, volume 4130 of *Lecture Notes in Computer Science*, pages 287–291. Springer.

[Baader and Narendran, 2001] Baader, F. and Narendran, P. (2001). Unification of concept terms in description logics. *Journal of Symbolic Computation*, 31(3):277 – 305.

[Baader and Peñaloza, 2010] Baader, F. and Peñaloza, R. (2010). Axiom pinpointing in general tableaux. *Journal of Logic and Computation*, 20(1):5–34. Special Issue: Tableaux and Analytic Proof Methods.

[Baader and Sattler, 2001] Baader, F. and Sattler, U. (2001). An overview of tableau algorithms for description logics. *Studia Logica*, 69(1):5–40.

[Bachmair and Ganzinger, 2001] Bachmair, L. and Ganzinger, H. (2001). Resolution theorem proving. In Robinson, J. A. and Voronkov, A., editors, *Handbook of Automated Reasoning (in 2 volumes)*, pages 19–99. Elsevier and MIT Press.

[Bao et al., 2007] Bao, J., Slutzki, G., and Honavar, V. (2007). Privacy-preserving reasoning on the Semantic Web. In *Web Intelligence*, pages 791–797. IEEE Computer Society.

[Bertino and Ferrari, 2002] Bertino, E. and Ferrari, E. (2002). Secure and selective dissemination of XML documents. *ACM Trans. Inf. Syst. Secur.*, 5(3):290–331.

[Biskup, 2016] Biskup, J. (2016). Selected results and related issues of confidentiality-preserving controlled interaction execution. In *Foundations of Information and Knowledge Systems - 9th International Symposium, FoIKS 2016, Linz, Austria, March 7-11, 2016. Proceedings*, pages 211–234.

[Biskup and Bonatti, 2001] Biskup, J. and Bonatti, P. A. (2001). Lying versus refusal for known potential secrets. *Data Knowl. Eng.*, 38(2):199–222.

[Biskup and Bonatti, 2004a] Biskup, J. and Bonatti, P. A. (2004a). Controlled query evaluation for enforcing confidentiality in complete information systems. *Int. J. Inf. Sec.*, 3(1):14–27.

[Biskup and Bonatti, 2004b] Biskup, J. and Bonatti, P. A. (2004b). Controlled query evaluation for known policies by combining lying and refusal. *Ann. Math. Artif. Intell.*, 40(1-2):37–62.

[Biskup and Bonatti, 2007] Biskup, J. and Bonatti, P. A. (2007). Controlled query evaluation with open queries for a decidable relational submodel. *Ann. Math. Artif. Intell.*, 50(1-2):39–77.

[Biskup et al., 2010] Biskup, J., Tadros, C., and Wiese, L. (2010). Towards controlled query evaluation for incomplete first-order databases. In Link, S. and Prade, H., editors, *Foundations of Information and Knowledge Systems, 6th International Symposium, FoIKS 2010, Sofia, Bulgaria, February 15-19, 2010. Proceedings*, volume 5956 of *Lecture Notes in Computer Science*, pages 230–247. Springer.

[Biskup and Weibert, 2008] Biskup, J. and Weibert, T. (2008). Keeping secrets in incomplete databases. *Int. J. Inf. Sec.*, 7(3):199–217.

[Blanco et al., 2008] Blanco, C., Lasheras, J., Valencia-García, R., Fernández-Medina, E., Álvarez, J. A. T., and Piattini, M. (2008). A systematic review and comparison of security ontologies. In *ARES*, pages 813–820. IEEE Computer Society.

[Bonatti, 2010] Bonatti, P. A. (2010). Datalog for security, privacy and trust. In de Moor, O., Gottlob, G., Furche, T., and Sellers, A. J., editors, *Datalog Reloaded - First International Workshop*, volume 6702 of *Lecture Notes in Computer Science*, pages 21–36. Springer.

[Bonatti et al., 2017] Bonatti, P. A., Faella, M., Petrova, I. M., and Sauro, L. (2017). A new semantics for overriding in description logics (extended abstract). In Sierra, C., editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 4975–4979. ijcai.org.

[Bonatti et al., 2009a] Bonatti, P. A., Faella, M., and Sauro, L. (2009a). Defeasible inclusions in low-complexity DLs: Preliminary notes. In Boutilier, C., editor, *IJCAI*, pages 696–701.

[Bonatti et al., 2010] Bonatti, P. A., Faella, M., and Sauro, L. (2010). EL with default attributes and overriding. In *Int. Semantic Web Conf. (ISWC 2010)*, volume 6496 of *LNCS*, pages 64–79. Springer.

[Bonatti et al., 2011a] Bonatti, P. A., Faella, M., and Sauro, L. (2011a). Adding default attributes to EL++. In Burgard, W. and Roth, D., editors, *AAAI*. AAAI Press.

[Bonatti et al., 2011b] Bonatti, P. A., Faella, M., and Sauro, L. (2011b). Defeasible inclusions in low-complexity DLs. *J. Artif. Intell. Res. (JAIR)*, 42:719–764.

[Bonatti et al., 2009b] Bonatti, P. A., Lutz, C., and Wolter, F. (2009b). The complexity of circumscription in DLs. *J. Artif. Intell. Res. (JAIR)*, 35:717–773.

[Bonatti et al., 2015a] Bonatti, P. A., Petrova, I. M., and Sauro, L. (2015a). A new semantics for overriding in description logics. *Artificial Intelligence*, 222:1–48. Available online: http://www.sciencedirect.com/science/article/pii/S0004370215000028.

[Bonatti et al., 2015b] Bonatti, P. A., Petrova, I. M., and Sauro, L. (2015b). Optimized construction of secure knowledge-base views. In Calvanese, D. and Konev, B., editors, *Proceedings of the 28th International Workshop on Description Logics, Athens,Greece, June 7-10, 2015.*, volume 1350 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Bonatti et al., 2015c] Bonatti, P. A., Petrova, I. M., and Sauro, L. (2015c). Optimizing the computation of overriding. In Arenas, M., Corcho, Ó., Simperl, E., Strohmaier, M., d'Aquin, M., Srinivas, K., Groth, P. T., Dumontier, M., Heflin, J., Thirunarayan, K., and Staab, S., editors, *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*, volume 9366 of *Lecture Notes in Computer Science*, pages 356–372. Springer.

[Bonatti et al., 2015d] Bonatti, P. A., Petrova, I. M., and Sauro, L. (2015d). Optimizing the computation of overriding. *CoRR*, abs/1507.04630.

[Bonatti and Samarati, 2003] Bonatti, P. A. and Samarati, P. (2003). Logics for authorization and security. In *Logics for Emerging Applications of Databases*, pages 277–323. Springer.

[Bonatti and Sauro, 2013] Bonatti, P. A. and Sauro, L. (2013). A confidentiality model for ontologies. In *International Semantic Web Conference (1)*, pages 17–32.

[Bonatti et al., 2014] Bonatti, P. A., Sauro, L., and Petrova, I. M. (2014). A mechanism for ontology confidentiality. In Giordano, L., Gliozzi, V., and Pozzato, G. L., editors, *Proceedings of the 29th Italian Conference on Computational Logic, Torino, Italy, June 16-18, 2014.*, volume 1195 of *CEUR Workshop Proceedings*, pages 147–161. CEUR-WS.org.

[Brandt, 2004] Brandt, S. (2004). Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else? In *Proceedings of the 16th European Conference on Artificial Intelligence*, ECAI'04, pages 298–302, Amsterdam, The Netherlands, The Netherlands. IOS Press.

[Britz et al., 2013] Britz, K., Casini, G., Meyer, T., Moodley, K., and Varzinczak, I. (2013). Ordered interpretations and entailment for defeasible description logics. Technical report, CAIR, CSIR Meraka and UKZN, South Africa.

[Britz et al., 2008] Britz, K., Heidema, J., and Meyer, T. A. (2008). Semantic preferential subsumption. In Brewka, G. and Lang, J., editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR 2008, Sydney, Australia, September 16-19, 2008*, pages 476–484. AAAI Press.

[Cadoli et al., 1990] Cadoli, M., Donini, F., and Schaerf, M. (1990). Closed world reasoning in hybrid systems. In *Proc. of ISMIS'90*, pages 474–481. Elsevier.

[Calvanese et al., 2005] Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., and Rosati, R. (2005). DL-Lite: Tractable description logics for ontologies. In *Proc. of AAAI 2005*, pages 602–607.

[Calvanese et al., 1999] Calvanese, D., De Giacomo, G., and Lenzerini, M. (1999). Reasoning in expressive description logics with fixpoints based on automata on infinite trees. In *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI 1999)*, pages 84–89.

[Carroll and Klyne, 2004] Carroll, J. and Klyne, G. (2004). Resource description framework (RDF): Concepts and abstract syntax. W3C recommendation, W3C. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

[Casini et al., 2014] Casini, G., Meyer, T., Moodley, K., and Nortje, R. (2014). Relevant closure: A new form of defeasible reasoning for description logics. In Fermé, E. and Leite, J., editors, *Logics in Artificial Intelligence - 14th European Conference, JELIA 2014, Funchal, Madeira, Portugal, September 24-26, 2014. Proceedings*, volume 8761 of *Lecture Notes in Computer Science*, pages 92–106. Springer.

[Casini et al., 2013a] Casini, G., Meyer, T., Moodley, K., and Varzinczak, I. J. (2013a). Towards practical defeasible reasoning for description logics. In Eiter, T., Glimm, B., Kazakov, Y., and Krötzsch, M., editors, *Informal Proceedings of the 26th International Workshop on Description Logics, Ulm, Germany, July 23 - 26, 2013*, volume 1014 of *CEUR Workshop Proceedings*, pages 587–599. CEUR-WS.org.

[Casini et al., 2013b] Casini, G., Meyer, T., Varzinczak, I. J., and Moodley, K. (2013b). Nonmonotonic reasoning in description logics: Rational closure for the abox. In Eiter, T., Glimm, B., Kazakov, Y., and Krötzsch, M., editors, *Informal Proceedings of the 26th International Workshop on Description Logics, Ulm, Germany, July 23 - 26, 2013*, volume 1014 of *CEUR Workshop Proceedings*, pages 600–615. CEUR-WS.org.

[Casini et al., 2015] Casini, G., Meyer, T. A., Moodley, K., Sattler, U., and Varzinczak, I. J. (2015). Introducing defeasibility into OWL ontologies. In Arenas, M., Corcho, Ó., Simperl, E., Strohmaier, M., d'Aquin, M., Srinivas, K., Groth, P. T., Dumontier, M., Heflin, J., Thirunarayan, K., and Staab, S., editors, *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II*, volume 9367 of *Lecture Notes in Computer Science*, pages 409–426. Springer.

[Casini and Straccia, 2010] Casini, G. and Straccia, U. (2010). Rational closure for defeasible description logics. In Janhunen, T. and Niemelä, I., editors, *JELIA*, volume 6341 of *Lecture Notes in Computer Science*, pages 77–90. Springer.

[Casini and Straccia, 2013] Casini, G. and Straccia, U. (2013). Defeasible inheritance-based description logics. *J. Artif. Intell. Res. (JAIR)*, 48:415–473.

[Chandra and Merlin, 1977] Chandra, A. K. and Merlin, P. M. (1977). Optimal implementation of conjunctive queries in relational data bases. In Hopcroft, J. E., Friedman, E. P., and Harrison, M. A., editors, *Proceedings of the 9th Annual ACM Symposium on Theory of Computing, May 4-6, 1977, Boulder, Colorado, USA*, pages 77–90. ACM.

[Chen and Stuckenschmidt, 2009] Chen, W. and Stuckenschmidt, H. (2009). A model-driven approach to enable access control for ontologies. In Hansen et al., H. R., editor,

Wirtschaftsinformatik, volume 246 of *books@ocg.at*, pages 663–672. Österreichische Computer Gesellschaft.

[Cuenca Grau, 2010] Cuenca Grau, B. (2010). Privacy in ontology-based information systems: A pending matter. *Semantic Web*, 1(1-2):137–141.

[Cuenca Grau and Horrocks, 2008] Cuenca Grau, B. and Horrocks, I. (2008). Privacy-preserving query answering in logic-based information systems. In Ghallab, M., Spyropoulos, C. D., Fakotakis, N., and Avouris, N. M., editors, *ECAI*, volume 178 of *Frontiers in Artificial Intelligence and Applications*, pages 40–44. IOS Press.

[Cuenca Grau et al., 2013] Cuenca Grau, B., Kharlamov, E., Kostylev, E. V., and Zheleznyakov, D. (2013). Controlled query evaluation over OWL 2 RL ontologies. In *Proceedings of the 12th International Semantic Web Conference - Part I*, ISWC '13, pages 49–65, New York, NY, USA. Springer-Verlag New York, Inc.

[Cuenca Grau and Motik, 2009] Cuenca Grau, B. and Motik, B. (2009). Importing ontologies with hidden content. In *Proceedings of the 22nd International Workshop on Description Logics (DL 2009), Oxford, UK, July 27-30, 2009*, volume 477 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Damiani et al., 2002] Damiani, E., De Capitani di Vimercati, S., Paraboschi, S., and Samarati, P. (2002). A fine-grained access control system for xml documents. *ACM Trans. Inf. Syst. Secur.*, 5(2):169–202.

[Damiani et al., 2000] Damiani, E., di Vimercati, S. D. C., Paraboschi, S., and Samarati, P. (2000). *Securing XML Documents*, pages 121–135. Springer Berlin Heidelberg, Berlin, Heidelberg.

[de Nivelle et al., 2000] de Nivelle, H., Schmidt, R., and Hustadt, U. (2000). Resolution-based methods for modal logics. *Logic Journal of the IGPL*, 8(3):265–292.

[Delaitre and Kazakov, 2009] Delaitre, V. and Kazakov, Y. (2009). Classifying ELH ontologies in SQL databases. In *Proceedings of the 5th International Workshop on OWL: Experiences and Directions (OWLED 2009), Chantilly, VA, United States, October 23-24, 2009*.

[Donini and Massacci, 2000] Donini, F. M. and Massacci, F. (2000). EXPTIME tableaux for ALC. *Artif. Intell.*, 124(1):87–138.

[Donini et al., 2002] Donini, F. M., Nardi, D., and Rosati, R. (2002). Description logics of minimal knowledge and negation as failure. *ACM Trans. Comput. Log.*, 3(2):177–225.

[Drabent et al., 2009] Drabent, W., Eiter, T., Ianni, G., Krennwallner, T., Lukasiewicz, T., and Maluszynski, J. (2009). Hybrid reasoning with rules and ontologies. In Bry, F. and Maluszynski, J., editors, *Semantic Techniques for the Web, The REWERSE Perspective*, volume 5500 of *Lecture Notes in Computer Science*, pages 1–49. Springer.

[Duc et al., 2012] Duc, C. L., Lamolle, M., and Curé, O. (2012). An expspace tableau-based algorithm for SHOIQ. In *Proceedings of the 2012 International Workshop on Description Logics, DL-2012, Rome, Italy, June 7-10, 2012*.

[Eiter et al., 2008] Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R., and Tompits, H. (2008). Combining answer set programming with description logics for the Semantic Web. *Artif. Intell.*, 172(12-13):1495–1539.

[Eiter et al., 2005] Eiter, T., Ianni, G., Schindlauer, R., and Tompits, H. (2005). A uniform integration of higher-order reasoning and external evaluations in Answer-Set Programming. In Kaelbling, L. P. and Saffiotti, A., editors, *IJCAI*, pages 90–96. Professional Book Center.

[Eiter and Lukasiewicz, 2000] Eiter, T. and Lukasiewicz, T. (2000). Default reasoning from conditional knowledge bases: Complexity and tractable cases. *Artif. Intell.*, 124(2):169–241.

[Eldora et al., 2011] Eldora, Knechtel, M., and Peñaloza, R. (2011). Correcting access restrictions to a consequence more flexibly. In Rosati, R., Rudolph, S., and Zakharyaschev, M., editors, *Description Logics*, volume 745 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Farkas and Jajodia, 2002] Farkas, C. and Jajodia, S. (2002). The inference problem: A survey. *SIGKDD Explor. Newsl.*, 4(2):6–11.

[Finin et al., 2008] Finin, T. W., Joshi, A., Kagal, L., Niu, J., Sandhu, R. S., Winsborough, W. H., and Thuraisingham, B. M. (2008). ROWLBAC: representing role based access control in OWL. In Ray, I. and Li, N., editors, *SACMAT*, pages 73–82. ACM.

[Flouris et al., 2010] Flouris, G., Fundulaki, I., Michou, M., and Antoniou, G. (2010). Controlling access to RDF graphs. In Berre, A.-J., Gómez-Pérez, A., Tutschku, K., and Fensel, D., editors, *FIS*, volume 6369 of *Lecture Notes in Computer Science*, pages 107–117. Springer.

[Fundulaki and Marx, 2004] Fundulaki, I. and Marx, M. (2004). Specifying access control policies for XML documents with XPath. In Jaeger, T. and Ferrari, E., editors, *SACMAT*, pages 61–69. ACM.

[Gabillon and Bruno, 2002] Gabillon, A. and Bruno, E. (2002). Regulating access to XML documents. In *Proceedings of the Fifteenth Annual Working Conference on Database and Application Security*, Das'01, pages 299–314, Norwell, MA, USA. Kluwer Academic Publishers.

[Geffner and Pearl, 1992] Geffner, H. and Pearl, J. (1992). Conditional entailment: Bridging two approaches to default reasoning. *Artif. Intell.*, 53(2-3):209–244.

[Ghilardi et al., 2006] Ghilardi, S., Lutz, C., and Wolter, F. (2006). Did I damage my ontology? A case for conservative extensions in description logics. In Doherty, P., Mylopoulos, J., and Welty, C., editors, *Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR'06)*, pages 187–197. AAAI Press.

[Giordano and Dupré, 2016] Giordano, L. and Dupré, D. T. (2016). Reasoning in a rational extension of SROEL. In Fiorentini, C. and Momigliano, A., editors, *Proceedings of the 31st Italian Conference on Computational Logic, Milano, Italy, June 20-22, 2016.*, volume 1645 of *CEUR Workshop Proceedings*, pages 53–68. CEUR-WS.org.

[Giordano et al., 2009a] Giordano, L., Gliozzi, V., Olivetti, N., and Pozzato, G. L. (2009a). Prototypical reasoning with low complexity description logics: Preliminary results. In Erdem, E., Lin, F., and Schaub, T., editors, *LPNMR*, volume 5753 of *Lecture Notes in Computer Science*, pages 430–436. Springer.

[Giordano et al., 2012] Giordano, L., Gliozzi, V., Olivetti, N., and Pozzato, G. L. (2012). Preferential low complexity description logics: Complexity results and proof methods. In Kazakov, Y., Lembo, D., and Wolter, F., editors, *Description Logics*, volume 846 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Giordano et al., 2013a] Giordano, L., Gliozzi, V., Olivetti, N., and Pozzato, G. L. (2013a). Minimal model semantics and rational closure in description logics. In Eiter, T., Glimm, B., Kazakov, Y., and Krötzsch, M., editors, *Informal Proceedings of the 26th International Workshop on Description Logics, Ulm, Germany, July 23 - 26, 2013*, volume 1014 of *CEUR Workshop Proceedings*, pages 168–180. CEUR-WS.org.

[Giordano et al., 2013b] Giordano, L., Gliozzi, V., Olivetti, N., and Pozzato, G. L. (2013b). A non-monotonic description logic for reasoning about typicality. *Artif. Intell.*, 195:165–202.

[Giordano et al., 2009b] Giordano, L., Olivetti, N., Gliozzi, V., and Pozzato, G. L. (2009b). ALC + T: a preferential extension of description logics. *Fundam. Inform.*, 96(3):341–372.

[Glimm et al., 2012] Glimm, B., Horrocks, I., Motik, B., Shearer, R., and Stoilos, G. (2012). A novel approach to ontology classification. *J. Web Sem.*, 14:84–101.

[Glimm et al., 2014] Glimm, B., Horrocks, I., Motik, B., Stoilos, G., and Wang, Z. (2014). Hermit: An owl 2 reasoner. *Journal of Automated Reasoning*, 53(3):245–269.

[Grau et al., 2008] Grau, B. C., Horrocks, I., Kazakov, Y., and Sattler, U. (2008). Modular reuse of ontologies: Theory and practice. *J. Artif. Intell. Res. (JAIR)*, 31:273–318.

[Grau et al., 2014] Grau, B. C., Kharlamov, E., Kostylev, E., and Zheleznyakov, D. (2014). Controlled query evaluation over lightweight ontologies. In *Proc. of the International Workshop on Description Logics (DL)*, pages 141–152.

[Grau et al., 2015] Grau, B. C., Kharlamov, E., Kostylev, E. V., and Zheleznyakov, D. (2015). Controlled query evaluation for Datalog and OWL 2 Profile ontologies. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 2883–2889. AAAI Press.

[Grau and Kostylev, 2016] Grau, B. C. and Kostylev, E. V. (2016). Logical foundations of privacy-preserving publishing of linked data. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 943–949. AAAI Press.

[Grau et al., 2012] Grau, B. C., Patel-Schneider, P., and Motik, B. (2012). OWL 2 web ontology language direct semantics (Second Edition). W3C recommendation, W3C. http://www.w3.org/TR/2012/REC-owl2-direct-semantics-20121211/.

[Grimm and Hitzler, 2009] Grimm, S. and Hitzler, P. (2009). A preferential tableaux calculus for circumscriptive ALCO. In Polleres, A. and Swift, T., editors, *RR*, volume 5837 of *Lecture Notes in Computer Science*, pages 40–54. Springer.

[Heflin, 2004] Heflin, J. (2004). OWL web ontology language use cases and requirements. W3C recommendation, W3C. http://www.w3.org/TR/2004/REC-webont-req-20040210/.

[Horridge et al., 2008] Horridge, M., Parsia, B., and Sattler, U. (2008). *Laconic and Precise Justifications in OWL*, pages 323–338. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Horrocks et al., 2006] Horrocks, I., Kutz, O., and Sattler, U. (2006). The even more irresistible SROIQ. In Doherty, P., Mylopoulos, J., and Welty, C. A., editors, *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006*, pages 57–67. AAAI Press.

[Horrocks and Sattler, 1999] Horrocks, I. and Sattler, U. (1999). A description logic with transitive and inverse roles and role hierarchies. *Journal of Logic and Computation*, 9(3):385–410.

[Hustadt et al., 2007] Hustadt, U., Motik, B., and Sattler, U. (2007). Reasoning in description logics by a reduction to disjunctive datalog. *J. Autom. Reason.*, 39(3):351–384.

[Hustadt et al., 2008] Hustadt, U., Motik, B., and Sattler, U. (2008). Deciding expressive description logics in the framework of resolution. *Information and Computation*, 206(5):579 – 601. Special Issue: The 17th International Conference on Concurrency Theory (CONCUR 2006).

[Kazakov, 2009] Kazakov, Y. (2009). Consequence-driven reasoning for Horn SHIQ Ontologies. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pages 2040–2045, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Kazakov and Klinov, 2013] Kazakov, Y. and Klinov, P. (2013). Incremental reasoning in EL+ without bookkeeping. In Eiter, T., Glimm, B., Kazakov, Y., and Krötzsch, M., editors, *Informal Proceedings of the 26th International Workshop on Description Logics, Ulm, Germany, July 23 - 26, 2013*, volume 1014 of *CEUR Workshop Proceedings*, pages 294–315. CEUR-WS.org.

[Kazakov and Klinov, 2014] Kazakov, Y. and Klinov, P. (2014). Bridging the gap between tableau and consequence-based reasoning. In *Informal Proceedings of the 27th International Workshop on Description Logics, Vienna, Austria, July 17-20, 2014.*, pages 579–590.

[Kazakov et al., 2012] Kazakov, Y., Krötzsch, M., and Simancik, F. (2012). ELK reasoner: Architecture and evaluation. In *Proceedings of the 1st International Workshop on OWL Reasoner Evaluation (ORE-2012), Manchester, UK, July 1st, 2012.*

[Kazakov et al., 2014] Kazakov, Y., Krötzsch, M., and Simancik, F. (2014). The incredible ELK - from polynomial procedures to efficient reasoning with ontologies. *J. Autom. Reasoning*, 53(1):1–61.

[Kazakov and Motik, 2008] Kazakov, Y. and Motik, B. (2008). A resolution-based decision procedure for $**shoiq**$. *J. Autom. Reason.*, 40(2-3):89–116.

[Knechtel and Stuckenschmidt, 2010] Knechtel, M. and Stuckenschmidt, H. (2010). Query-based access control for ontologies. In *Web Reasoning and Rule Systems - 4th Int. Conference, RR 2010.*, volume 6333 of *Lecture Notes in Computer Science*, pages 73–87. Springer.

[Kollia and Glimm, 2013] Kollia, I. and Glimm, B. (2013). Optimizing QL query answering over OWL ontologies. *J. Artif. Int. Res.*, 48(1):253–303.

[Kolovski et al., 2007] Kolovski, V., Hendler, J. A., and Parsia, B. (2007). Analyzing web access control policies. In Williamson, C. L., Zurko, M. E., Patel-Schneider, P. F., and Shenoy, P. J., editors, *WWW*, pages 677–686. ACM.

[Kontchakov et al., 2008] Kontchakov, R., Wolter, F., and Zakharyaschev, M. (2008). Can you tell the difference between DL-Lite ontologies? In Brewka, G. and Lang, J., editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR 2008, Sydney, Australia, September 16-19, 2008*, pages 285–295. AAAI Press.

[Kraus et al., 1990] Kraus, S., Lehmann, D. J., and Magidor, M. (1990). Nonmonotonic reasoning, preferential models and cumulative logics. *Artif. Intell.*, 44(1-2):167–207.

[Lehmann and Magidor, 1992] Lehmann, D. J. and Magidor, M. (1992). What does a conditional knowledge base entail? *Artif. Intell.*, 55(1):1–60.

[Lehmann, 2009] Lehmann, J. (2009). DL-Learner: Learning concepts in description logics. *J. Mach. Learn. Res.*, 10:2639–2642.

[Leone et al., 2006] Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., and Scarcello, F. (2006). The DLV system for knowledge representation and reasoning. *ACM Trans. Comput. Logic*, 7(3):499–562.

[Lukasiewicz, 2008] Lukasiewicz, T. (2008). Expressive probabilistic description logics. *Artif. Intell.*, 172(6-7):852–883.

[Lutz, 1999] Lutz, C. (1999). Complexity of terminological reasoning revisited. In *Proceedings of the 6th International Conference on Logic Programming and Automated Reasoning*, LPAR '99, pages 181–200, London, UK, UK. Springer-Verlag.

[Lutz, 2008] Lutz, C. (2008). The complexity of conjunctive query answering in expressive description logics. In Armando, A., Baumgartner, P., and Dowek, G., editors, *Proceedings of the 4th International Joint Conference on Automated Reasoning (IJCAR2008)*, number 5195 in LNAI, pages 179–193. Springer.

[Lutz and Sattler, 2000] Lutz, C. and Sattler, U. (2000). Mary likes all cats. In Baader, F. and Sattler, U., editors, *Proceedings of the 2000 International Workshop on Description Logics (DL2000)* , number 33 in CEUR-WS, pages 213–226, Aachen , Germany. RWTH Aachen.

[Lutz et al., 2009] Lutz, C., Toman, D., and Wolter, F. (2009). Conjunctive query answering in the description logic el using a relational database system. In *Proceedings*

of the 21st International Jont Conference on Artifical Intelligence, IJCAI'09, pages 2070–2075, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Lutz et al., 2007] Lutz, C., Walther, D., and Wolter, F. (2007). Conservative extensions in expressive description logics. In Veloso, M. M., editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 453–458.

[Martin-Recuerda and Walther, 2014] Martin-Recuerda, F. and Walther, D. (2014). Axiom dependency hypergraphs for fast modularisation and atomic decomposition. In Bienvenu, M., Ortiz, M., Rosati, R., and Simkus, M., editors, *Proceedings of the 27th International Workshop on Description Logics (DL'14)*, volume 1193 of *CEUR Workshop Proceedings*, pages 299–310.

[McCarthy, 1986] McCarthy, J. (1986). Applications of circumscription to formalizing common sense knowledge. *Artificial Intelligence*, 28:89–116.

[Mendez and Suntisrivaraporn, 2009] Mendez, J. and Suntisrivaraporn, B. (2009). Reintroducing cel as an owl 2 el reasoner. In Grau, B. C., Horrocks, I., Motik, B., and Sattler, U., editors, *Proceedings of the 2009 International Workshop on Description Logics (DL2009)*, volume 477 of *CEUR-WS*.

[Motik et al., 2012] Motik, B., Grau, B. C., Horrocks, I., Wu, Z., and Fokoue, A. (2012). OWL 2 web ontology language profiles (Second Edition). W3C recommendation, W3C. http://www.w3.org/TR/2012/REC-owl2-profiles-20121211/.

[Motik and Patel-Schneider, 2012] Motik, B. and Patel-Schneider, P. (2012). OWL 2 web ontology language mapping to RDF graphs (Second Edition). W3C recommendation, W3C. http://www.w3.org/TR/2012/REC-owl2-mapping-to-rdf-20121211/.

[Motik and Rosati, 2010] Motik, B. and Rosati, R. (2010). Reconciling description logics and rules. *J. ACM*, 57(5).

[Motik and Sattler, 2006] Motik, B. and Sattler, U. (2006). A comparison of reasoning techniques for querying large description logic aboxes. In *Proceedings of the 13th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning*, LPAR'06, pages 227–241, Berlin, Heidelberg. Springer-Verlag.

[Murata et al., 2003] Murata, M., Tozawa, A., Kudo, M., and Hada, S. (2003). XML access control using static analysis. In *Proceedings of the 10th ACM Conference on Computer and Communications Security, CCS 2003, Washington, DC, USA, October 27-30, 2003*, pages 73–84.

[Nebel, 1990] Nebel, B. (1990). Terminological reasoning is inherently intractable. *Artificial Intelligence*, 43(2):235 – 249.

[Nguyen, 2014] Nguyen, L. A. (2014). ExpTime tableaux with global state caching for the description logic SHIO. *Neurocomput.*, 146(C):249–263.

[Nguyen and Golinska-Pilarek, 2014] Nguyen, L. A. and Golinska-Pilarek, J. (2014). An ExpTime tableau method for dealing with nominals and qualified number restrictions in deciding the description logic SHOQ. *Fundam. Inform.*, 135:433–449.

[Parsia et al., 2012] Parsia, B., Patel-Schneider, P., and Motik, B. (2012). OWL 2 web ontology language XML serialization (Second Edition). W3C recommendation, W3C. http://www.w3.org/TR/2012/REC-owl2-xml-serialization-20121211/.

[Patel-Schneider and Horridge, 2012] Patel-Schneider, P. and Horridge, M. (2012). OWL 2 web ontology language manchester syntax (Second Edition). W3C note, W3C. http://www.w3.org/TR/2012/NOTE-owl2-manchester-syntax-20121211/.

[Patel-Schneider et al., 2012] Patel-Schneider, P., Parsia, B., and Motik, B. (2012). OWL 2 web ontology language structural specification and functional-style syntax (Second Edition). W3C recommendation, W3C. http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/.

[Raimond and Schreiber, 2014] Raimond, Y. and Schreiber, G. (2014). RDF 1.1 primer. W3C note, W3C. http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/.

[Rector, 2004] Rector, A. L. (2004). Defaults, context, and knowledge: Alternatives for OWL-indexed knowledge bases. In *Pacific Symposium on Biocomputing*, pages 226–237. World Scientific.

[Rudolph et al., 2012] Rudolph, S., Patel-Schneider, P., Parsia, B., Hitzler, P., and Krötzsch, M. (2012). OWL 2 web ontology language primer (second edition). W3C recommendation, W3C. http://www.w3.org/TR/2012/REC-owl2-primer-20121211/.

[Sandewall, 2010] Sandewall, E. (2010). Defeasible inheritance with doubt index and its axiomatic characterization. *Artif. Intell.*, 174(18):1431–1459.

[Sattler et al., 2009] Sattler, U., Schneider, T., and Zakharyaschev, M. (2009). Which kind of module should I extract? In *Proceedings of the 22nd International Workshop on Description Logics (DL 2009), Oxford, UK, July 27-30, 2009*.

[Schmidt-Schaubßand Smolka, 1991] Schmidt-Schaubß, M. and Smolka, G. (1991). Attributive concept descriptions with complements. *Artif. Intell.*, 48(1):1–26.

[Schneider, 2012] Schneider, M. (2012). OWL 2 web ontology language RDF-based semantics (Second Edition). W3C recommendation, W3C. http://www.w3.org/TR/2012/REC-owl2-rdf-based-semantics-20121211/.

[Sertkaya, 2011] Sertkaya, B. (2011). In the search of improvements to the EL+ classification algorithm. In *Proceedings of the 24th International Workshop on Description Logics (DL 2011), Barcelona, Spain, July 13-16, 2011.*

[Sicherman et al., 1983] Sicherman, G. L., De Jonge, W., and Van de Riet, R. P. (1983). Answering queries without revealing secrets. *ACM Trans. Database Syst.*, 8(1):41–59.

[Simancik et al., 2014] Simancik, F., Motik, B., and Horrocks, I. (2014). Consequence-based and fixed-parameter tractable reasoning in description logics. *Artif. Intell.*, 209:29–77.

[Simančík et al., 2011] Simančík, F., Kazakov, Y., and Horrocks, I. (2011). Consequence-based reasoning beyond Horn ontologies. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, pages 1093–1098. AAAI Press.

[Sirin et al., 2007] Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y. (2007). Pellet: A practical owl-dl reasoner. *Web Semant.*, 5(2):51–53.

[Sperberg-McQueen et al., 2012] Sperberg-McQueen, M., Peterson, D., Malhotra, A., Gao, S., Thompson, H., and Biron, P. V. (2012). W3C XML schema definition language (XSD) 1.1 Part 2: Datatypes. W3C recommendation, W3C. http://www.w3.org/TR/2012/REC-xmlschema11-2-20120405/.

[Stevens et al., 2007] Stevens, R., Aranguren, M. E., Wolstencroft, K., Sattler, U., Drummond, N., Horridge, M., and Rector, A. L. (2007). Using OWL to model biological knowledge. *International Journal of Man-Machine Studies*, 65(7):583–594.

[Stouppa and Studer, 2009] Stouppa, P. and Studer, T. (2009). Data privacy for knowledge bases. In Artëmov, S. N. and Nerode, A., editors, *LFCS*, volume 5407 of *LNCS*, pages 409–421. Springer.

[Tao et al., 2010] Tao, J., Slutzki, G., and Honavar, V. (2010). Secrecy-preserving query answering for instance checking in $\mathcal{EL}$. In *Web Reasoning and Rule Systems - 4th Int. Conference, RR 2010.*, volume 6333 of *Lecture Notes in Computer Science*, pages 195–203. Springer.

[Thomas, 1990] Thomas, W. (1990). Handbook of Theoretical Computer Science (Vol. B). chapter Automata on Infinite Objects, pages 133–191. MIT Press, Cambridge, MA, USA.

[Tsarkov and Horrocks, 2006] Tsarkov, D. and Horrocks, I. (2006). Fact++ description logic reasoner: System description. In *Proceedings of the Third International Joint Conference on Automated Reasoning*, IJCAR'06, pages 292–297, Berlin, Heidelberg. Springer-Verlag.

[Tsarkov et al., 2007] Tsarkov, D., Horrocks, I., and Patel-Schneider, P. F. (2007). Optimizing terminological reasoning for expressive description logics. *J. Autom. Reasoning*, 39(3):277–316.

[Uszok et al., 2003] Uszok, A., Bradshaw, J. M., Jeffers, R., Suri, N., Hayes, P. J., Breedy, M. R., Bunch, L., Johnson, M., Kulkarni, S., and Lott, J. (2003). KAoS policy and domain services: Towards a description-logic approach to policy representation, deconfliction, and enforcement. In *4th IEEE Int. Workshop on Policies for Distributed Systems and Networks (POLICY)*, pages 93–96. IEEE Computer Soc.

[Wishart et al., 2005] Wishart, R., Henricksen, K., and Indulska, J. (2005). Context obfuscation for privacy via ontological descriptions. In Strang, T. and Linnhoff-Popien, C., editors, *LoCA*, volume 3479 of *Lecture Notes in Computer Science*, pages 276–288. Springer.

[Woo and Lam, 1993] Woo, T. Y. C. and Lam, S. S. (1993). Authorizations in distributed systems: A new approach. *Journal of Computer Security*, 2(2-3):107–136.

[Zhang et al., 2009] Zhang, R., Artale, A., Giunchiglia, F., and Crispo, B. (2009). Using description logics in relation based access control. In *Proceedings of the 22nd International Workshop on Description Logics (DL 2009), Oxford, UK, July 27-30.*