Technische Universität Dresden

International Masters Programme in Computational Logic
Institute for Theoretical Computer Science
Computer Science Department

Master's Thesis

OPTIMIZATION AND IMPLEMENTATION OF SUBSUMPTION ALGORITHMS FOR THE DESCRIPTION LOGIC $\mathcal{EL}$ WITH CYCLIC TBOXES AND GENERAL CONCEPT INCLUSION AXIOMS

Boontawee Suntisrivaraporn
meng@tcs.inf.tu-dresden.de

14 December 2004

Overseeing Professor : Prof. Dr. Franz Baader
Supervisor : Dr. Carsten Lutz

TECHNISCHE UNIVERSITÄT DRESDEN

Author:	**Boontawee Suntisrivaraporn**
Matrikel-Nr.:	**2981034**
Title:	**Optimization and Implementation of Subsumption Algorithms for the Description Logic $\mathcal{EL}$ with Cyclic TBoxes and General Concept Inclusion Axioms**
Degree:	**Master of Science**
Date of submission:	**14 December 2004**

Erklärung
Hiermit erkläre ich, die vorliegende Arbeit selbständig verfaßt und keine anderen als die angegebenen Literaturhilfsmittel verwendet zu haben.

Declaration
Hereby I certify that the thesis has been written by me. Any help that I have received in my research work has been acknowledged. Additionally, I certify that I have not used any auxiliary sources and literature except those cited in the thesis.

Signature of Author

Abstract

The subsumption problem in the description logic (DL) $\mathcal{EL}$ has been shown to be polynomial regardless of whether cyclic or acyclic TBoxes are used. Recently, it was shown that the problem remains tractable even when admitting general concept inclusion (GCI) axioms. Motivated by its nice complexity and sufficient expressiveness for some applications, we propose three decision procedures for computing subsumption in the DL $\mathcal{EL}$ whose run-time is bounded by a low-degree polynomial. The three decision procedures are for three terminological settings in $\mathcal{EL}$: TBoxes with greatest fixpoint semantics ($\mathcal{EL}^{gfp}$), TBoxes with descriptive semantics ($\mathcal{EL}^{desc}$), and terminologies with GCIs ($\mathcal{EL}^{gci}$).

For subsumption w.r.t. TBoxes, i.e., $\mathcal{EL}^{gfp}$ and $\mathcal{EL}^{desc}$, we use a characterization through simulations on so-called $\mathcal{EL}$-description graphs—the syntactically normalized representation of $\mathcal{EL}$-TBoxes. With an efficient algorithm for computing simulations on graphs, we show that $\mathcal{EL}^{gfp}$-subsumption can be decided in time cubic in the size of the input TBox. We decide $\mathcal{EL}^{desc}$-subsumption by reducing the simulation problem on graphs to the satisfiability problem of Horn formulae. Then, we apply a linear-time Horn-SAT algorithm to our Horn formulae. This approach yields a quartic-time decision procedure for $\mathcal{EL}^{desc}$-subsumption. Concerning terminologies with GCIs, we employ a different normalization and characterize subsumption through so-called implication sets. We show that $\mathcal{EL}^{gci}$-subsumption can be decided in time cubic in the size of the input terminology, by translating the implication sets into a Horn formula and exploiting the linear-time Horn-SAT algorithm similarly to $\mathcal{EL}^{desc}$.

Besides, we implement these decision procedures in the Common LISP language and evaluate their efficiency using the Gene Ontology as a benchmark. The implementation can be used as terminological reasoners that classify ontologies represented in $\mathcal{EL}$-TBoxes.

Acknowledgements

Many thanks should first go to Carsten Lutz, without whom this thesis would not have been accomplished in time. With the very interesting lecture of Logic-based Knowledge Representation given by him, I was for the first time introduced to the world of Description Logics. He has not only given me advice regularly on my research work but also developed my scientific research expertise in the field. Additionally, I am thankful for his patience with proofreading this thesis and for loads of useful criticisms.

I would like to thank Prof. Franz Baader for his support and trust in my ability to achieve the thesis. Many thanks also go to Prof. Andrei Voronkov, who trusts me and keeps pushing me from a distance to get the best opportunity I should. Many other people in the group have contributed their time towards this thesis, for which I am grateful. Thanks to Sebastian and Anni for constructive suggestions and fruitful discussions, especially when Common LISP is concerned. Thanks also go to our secretary Frau Kerstin Achtruth for her kindness and helpfulness with bureaucratic problems.

Not to mention, I appreciate the opportunity that German Academic Exchange Service (DAAD) and Siemens AG has granted me through an honored scholarship. This scholarship has enabled me to fulfill my will to complete the Master's degree in Germany.

Most importantly, I am grateful to my parents, sisters and Bookie for always giving me love and support virtually whenever I need.

Contents

Chapter 1

Introduction

Knowledge Representation (KR) is an indispensable subfield of Artificial Intelligence which focuses on the design of formalisms that are both epistemologically and computationally adequate for expressing knowledge about a particular domain. Since the 1960's, KR has been concerned with the idea that the knowledge should be represented by describing classes of objects and relationships among them. Early KR formalisms such as "semantic networks" [12] and "frame systems" [13] have been developed based on this idea. However, such formalisms were initially tailored towards specific applications, and their associated reasoning tools were strongly dependent on the implementation strategies. This is due to a major drawback of these early systems, namely the lack of formal semantics. A fundamental step towards a logic-based characterization of those early systems has been accomplished through the work on the KL-ONE system [14], which is usually regarded as the origin of the research on *Description Logics*.

Description Logics

Description Logics (DLs)—also known as terminological logics, conceptual languages, or KL-ONE-like languages—are a family of logic-based knowledge representation designed to represent and reason about conceptual knowledge. DLs collect many ideas stemming from semantic networks and frame-based systems and provide a logical basis for interpreting objects, classes of objects (or *concepts*), and relationships between objects (or *roles*). A specific DL is essentially characterized by the set of *constructors* that it provides to build more complex *concept descriptions* out of atomic *concept names* and *role names*. The expressive power of a DL depends mainly on its constructors, and so does its complexity. As a rule, the more constructors a DL provides, the higher in expressive power and complexity it is likely to be.

Besides a conceptual language, DLs can be equipped with a terminological component called a *TBox*. TBoxes comprise terminological statements that assert

a piece of knowledge about a particular domain. There are two kinds of them: a *concept definition* and a *general concept inclusion or GCI*. The former defines concept names as abbreviations for complex concept descriptions ($A \equiv D$) while the latter asserts the inclusion between two concept descriptions ($C \sqsubseteq D$). Unlike concept definitions, GCIs asserts that anything interpreted to be in one concept (a class of objects) must necessarily also be in another concept, but not vice versa. In the present paper we are interested in two kinds of TBoxes, namely with and without GCIs. For the rest of this paper, a TBox with GCIs is called *general*, and a *non-general* TBox (sometimes simply TBox) refers to a terminological box without GCIs. Non-general TBoxes may contain cyclic dependencies (or terminological cycles), i.e., a defined concept may refer to itself either directly or indirectly. In this case, the TBoxes are said to be *cyclic*, otherwise *acyclic*. In the case of general TBoxes, cyclicity does not concern, since it is unclear how cyclic dependencies should look like in a GCI. In fact, there are no defined concepts in general TBoxes, on which terminological cycles are defined.

The first thorough investigation of cyclic TBoxes in DLs is due to Nebel [7], who introduced three distinct semantics for such terminologies: *least fixpoint* (lfp) semantics, *greatest fixpoint* (gfp) semantics, and *descriptive* semantics. Whilst gfp-semantics (lfp-semantics) considers only the models that interpret the defined concepts as large (small) as possible, descriptive semantics considers all models. For acyclic TBoxes, the three semantics coincide. Since terminological cycles are undefined in TBoxes with GCIs, it would be nonsensical to apply fixpoint semantics. So only descriptive semantics makes sense and is considered for general TBoxes.

Motivation for $\mathcal{EL}$

Early DLs—for example, the basic DL $\mathcal{FL}_0$, which allows for the top-concept ($\top$), conjunction ($C \sqcap D$) and value restrictions ($\forall r.C$) only—allowed the use of value restrictions ($\forall r.C$) but not of existential restrictions ($\exists r.C$). Consequently, such DLs could express, for example, a parent whose children are all male using value restriction $\forall$has_child.Male, but not a parent with a son using existential restriction $\exists$has_child.Male. The main reason for having value restriction but not existential restriction in those early DLs was that, when formulating the logical status of property arcs in semantic networks (slots in frame-based systems, respectively), it was determined that arcs (slots, respectively) should be comprehended as value restrictions (see, e.g., [7]). Later, when more expressive DLs allowing for full negation were considered, existential restrictions came in as the dual of value restrictions.

As a quintessential example, $\mathcal{ALC}$ (Attributive Language with Complements) [15] is the smallest propositionally closed Description Logic, which provides for existential restriction, value restriction and all Boolean operators: top-concept

($\top$), negation ($\neg C$), conjunction ($C \sqcap D$) and disjunction ($C \sqcup D$). Intuitively, the $\mathcal{ALC}$-concept description

$$\mathsf{Male} \sqcap \exists \mathsf{has\_child}.\mathsf{Male} \sqcap \exists \mathsf{has\_child}.\neg \mathsf{Male} \sqcap \forall \mathsf{has\_child}.(\mathsf{Scientist} \sqcup \mathsf{Governor})$$

represents 'a father of at least two children, a son and a daughter, and all his children are either a scientist or governor.' Here, Male, Scientist and Governor are concept names while has_child is a sole role name. The concept description is constructed out of these concept and role names.

Thus, for historical reasons, DLs with existential but not value restriction are somewhat unexplored. In this thesis, we concentrate on the DL $\mathcal{EL}$, which allows for the top-concept, conjunction, and existential restrictions only. Obviously, this logic is comparable to the inexpressive DL $\mathcal{FL}_0$, but with existential restriction in place of value restriction, $\mathcal{EL}$ can sufficiently express some notions that $\mathcal{FL}_0$ cannot. For instance,

$$\exists \mathsf{has\_child}.\mathsf{Human}$$

denotes the notion of parent, and

$$\mathsf{Animal} \sqcap \mathsf{WarmBlooded} \sqcap \exists \mathsf{feeds\_broods\_with}.\mathsf{Milk}$$

represents mammal—the animal Class of Mammalia. If equipped with cyclic terminologies and an appropriate semantics, $\mathcal{EL}$ can be used to express, e.g., the notion of 'nodes on an infinite path' by the following cyclic concept definition:

$$\mathsf{InfNode} \equiv \mathsf{Node} \sqcap \exists \mathsf{edge}.\mathsf{InfNode}.$$

It should be noted that there are indeed applications where the small DL $\mathcal{EL}$ appears to be adequate. In fact, the Gene Ontology [10] can epistemologically sufficiently be represented in $\mathcal{EL}$ with an acyclic TBox (see Chapter 7). Another motivation to consider the DL $\mathcal{EL}$ is in the domain of medical terminologies: the widely used medical terminology SNOMED [23] corresponds to an $\mathcal{EL}$-TBox [24], and the GALEN [25] medical terminology, in which GCIs are used extensively [26], can be represented by a general $\mathcal{EL}$-TBox.

Reasoning and Complexity

The most important inference problems in DLs are *satisfiability* and *subsumption* of concept descriptions. A concept description is said to be *satisfiable* if it is consistent, i.e., no contradictions occur in it. However, in DLs with no negations such as $\mathcal{FL}_0$ and $\mathcal{EL}$, satisfiability is uninteresting, as all concept descriptions in such logics are satisfiable. With regards to concept subsumption, we aim at a determination of subconcept-superconcept relationship. A subconcept is always subsumed by its superconcepts, i.e., superconcepts are more general while

subconcepts are more specific. Semantically, a subconcept is always interpreted as a subset of superconcepts in all models of interest. The *subsumption hierarchy* of a TBox is a graph of which nodes are all concepts in question (i.e., all concepts present in the TBox) and edges are the subconcept-superconcept relationship.

In [2, 3], subsumption w.r.t. cyclic TBoxes in $\mathcal{FL}_0$ was characterized with the help of finite automata, which provided PSPACE decision procedures for subsumption in $\mathcal{FL}_0$ with cyclic TBoxes for the three types of semantics introduced by Nebel. Additionally, it was shown that subsumption is PSPACE-hard [20]. The PSPACE results for $\mathcal{FL}_0$ with cyclic TBoxes were extended by Küsters [6] to the DL $\mathcal{ALN}$, which extends $\mathcal{FL}_0$ by atomic negation and number restrictions. These hardness results immediately imply PSPACE-hardness for subsumption in $\mathcal{FL}_0$ and $\mathcal{ALN}$ w.r.t. terminologies with GCIs.

Terminological cycles were also considered in more expressive conceptual languages like $\mathcal{ALC}$, which extends $\mathcal{FL}_0$ by full negation. The complexity of the subsumption problem in this logic with cyclic TBoxes is EXPTIME-complete. This was accomplished under the fact that the DL $\mathcal{ALC}$ is a syntactic variant of the multi-modal logic K and a reduction of $\mathcal{ALC}$ with cyclic TBoxes to the modal μ-calculus [17, 18]. Again, on account of the generality of terminologies with GCIs, subsumption in $\mathcal{ALC}$ w.r.t. general TBoxes is EXPTIME-hard. Moreover, the EXPTIME-hardness result has been unveiled also for very expressive DLs such as $\mathcal{ALCNR}$ [21] and $\mathcal{SHIQ}$ [22].

Despite these very general results of subsumption in expressive DLs with cyclic TBoxes, there is still a good reason to explore cyclic terminologies in less expressive DLs, especially sub-Boolean logics. Definitely, a lower complexity is anticipated when sacrificing expressive power. For DLs with value restrictions, this expectation is not gratified, for even in the inexpressive DL $\mathcal{FL}_0$, subsumption turns from NP-complete to PSPACE-complete if cyclic dependencies are allowed in TBoxes. Even though this complexity class is better than EXPTIME-completeness for $\mathcal{ALC}$, it still means from the practical point of view that the subsumption algorithm may need exponential time. In comparison to $\mathcal{FL}_0$, the complexity class of subsumption problem in $\mathcal{EL}$ remains unchanged when allowing terminological cycles. In fact, subsumption in $\mathcal{EL}$ can be decided in polynomial time w.r.t. the three types of semantics introduced by Nebel [1], regardless terminological cyclicity. It has been investigated by Brandt [5] that by admitting GCIs and so-called *simple role inclusion* axioms, subsumption in $\mathcal{ELH}$ remains traceable. Hence, polynomial time result holds also for subsumption in $\mathcal{EL}$ with general TBoxes.

Polynomial-time Subsumption Algorithms

As mentioned earlier regarding cyclic TBoxes, there are three relevant semantics: gfp-semantics, lfp-semantics, and descriptive semantics. It has been shown in

[1], however, that for the DL $\mathcal{EL}$ lfp-semantics is trivial and thus uninteresting. Indeed, defined concepts with terminological cycles are always interpreted as the empty set w.r.t. lfp-semantics and thus can be removed from the cyclic TBox. The remaining TBox turns out to be acyclic on which the three semantics in question coincide. Concerning general TBoxes, only descriptive semantics is meant and is considered as an extension of TBoxes with descriptive semantics. Putting together, in the present paper we consider the subsumption problem in the DL $\mathcal{EL}$ w.r.t. three different combinations of terminologies and semantics: *(i)* TBoxes and greatest fixpoint semantics, *(ii)* TBoxes and descriptive semantics, and *(iii)* terminologies with GCIs and (descriptive) semantics.

We propose three algorithms for computing the subsumption hierarchy in $\mathcal{EL}$ w.r.t. the three terminologies and semantics mentioned above. Subsumption in $\mathcal{EL}$ w.r.t. (non-general) TBoxes can be characterized through graph simulation [1]. We pursue this direction for the first two algorithms. A quadratic-time algorithm for computing similarity [8] is exploited in the first algorithm, giving us a cubic time decision procedure. The second algorithm reduces graph simulation into the Horn-SAT and applies a linear-time algorithm for testing satisfiability of Horn formulae [9]. With this approach we can compute subsumption in $\mathcal{EL}$ w.r.t. TBoxes and descriptive semantics in quartic (bi-quadratic) time as the worst case. For general TBoxes, we introduce a new straightforward normal form and compute subsumption by means of implication sets. This algorithm appears to be more optimal than the second algorithm, as it needs only cubic time in the size of the input despite its generality.

Even in the case of acyclic terminologies, e.g., the Gene Ontology, our *cubic* and *quartic* subsumption algorithms improve the usual approach that first unfolds the TBox, since unfolding can potentially take an exponential number of steps.

The first two algorithms ($\mathcal{EL}$-TBoxes w.r.t. descriptive and gfp-semantics) are implemented in the Common LISP, Allegro CL. The implementations are evaluated using the Gene Ontology [10] as a benchmark. Successfully, the Gene Ontology can be classified with over a hundred thousand subsumption outcomes.

The following chapters of this paper are organized as follows:

The description logic $\mathcal{EL}$ is introduced in Chapter 2, beginning with the syntax and semantics of its concept language. Then, two kinds of terminological formalisms: $\mathcal{EL}$-TBoxes and general $\mathcal{EL}$-TBoxes, together with their (descriptive) semantics are introduced. For $\mathcal{EL}$-TBoxes without GCIs, we additionally define *fixpoint semantics*. In this chapter, we also give formal definitions of subsumption between two $\mathcal{EL}$-concept descriptions.

In Chapter 3, we define the notion of normalized $\mathcal{EL}$-TBoxes and $\mathcal{EL}$-description graphs. We show how such normalized $\mathcal{EL}$-TBoxes can be translated into $\mathcal{EL}$-

description graphs, which will be used for the characterization of subsumption w.r.t. $\mathcal{EL}$-TBoxes in succeeding chapters.

In Chapter 4, we give a definition of *graph simulation*, which is used to characterize subsumption in $\mathcal{EL}$ w.r.t. greatest fixpoint semantics. Then, we present an efficient algorithm for computing such a simulation. Finally, we show that the subsumption problem w.r.t. greatest fixpoint semantics can be computed in *cubic* time.

Chapter 5 is dedicated to a characterization of and an algorithm for subsumption in $\mathcal{EL}$ w.r.t. descriptive semantics. We reduce the subsumption problem to the satisfiability problem of *Horn formulae*. With the help of a linear-time Horn-SAT algorithm [9], we show in the end of this chapter that the subsumption problem w.r.t. descriptive semantics can be computed in *quartic* time.

For general $\mathcal{EL}$-TBoxes, we only consider descriptive semantics. A characterization of and an algorithm for subsumption w.r.t. general $\mathcal{EL}$-TBoxes are presented in Chapter 6. We start by introducing *GCI-normalized $\mathcal{EL}$-TBoxes* in Section 6.1. Then, *implication sets*—a characterization of subsumption w.r.t. GCI-normalized $\mathcal{EL}$-TBoxes—is presented in Section 6.2. Section 6.3 devotes to an efficient algorithm for computing implication sets by applying a linear-time Horn-SAT algorithm [9]. We also prove in this section that the subsumption problem w.r.t. general TBoxes can be computed in *cubic* time.

The Gene Ontology is used as a benchmark for our subsumption algorithms. We discuss the Gene Ontology and some facts on it in Section 7.1. The translation of the Gene Ontology into $\mathcal{EL}$-TBoxes is also given in this section. We perform a number of experiments, and the results are shown in Section 7.2.

Conclusion and further works are discussed in the Chapter 8.

Chapter 2

The Description Logic $\mathcal{EL}$

In this chapter, we formally define the syntax and semantics of the description logic $\mathcal{EL}$, as well as two kinds of terminological formalisms. Then, we introduce the standard inference problems in $\mathcal{EL}$ with respect to the two different terminological formalisms.

We start with introducing the syntax of $\mathcal{EL}$-concept descriptions.

Definition 1 (Syntax of $\mathcal{EL}$-concept descriptions). Let N_{con} and N_{role} be disjoint sets of concept names and role names. The set of *$\mathcal{EL}$-concept descriptions* is defined inductively as follows:

- each concept name $A \in \mathsf{N}_{con}$ is an $\mathcal{EL}$-concept description;
- if C, D are $\mathcal{EL}$-concept descriptions and $r \in \mathsf{N}_{role}$, then the top-concept $\top$, conjunction $C \sqcap D$, and existential restriction $\exists r.C$ are also $\mathcal{EL}$-concept descriptions.

The top-concept and concept names are called *atomic* denoted by $\mathsf{N}_{con}^{\top}$, while conjunction and existential restriction are called *non-atomic* or *complex*. $\Diamond$

For example, let Human and has_child be a concept name and a role name, respectively. The complex concept description

$$\mathsf{Human} \sqcap \exists \mathsf{has\_child}.\mathsf{Human}$$

literally describes the notion of "parent". Analogously,

$$\mathsf{Human} \sqcap \exists \mathsf{has\_child}.\exists \mathsf{has\_child}.\mathsf{Human}$$

describes the notion of "grandparent". Formally, we define the semantics of concept descriptions in terms of interpretations.

Definition 2 (Semantics of $\mathcal{EL}$-concept descriptions). An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of the non-empty *interpretation domain* $\Delta^{\mathcal{I}}$ and the *interpretation function* $\cdot^{\mathcal{I}}$, which maps each concept name $A \in \mathsf{N}_{con}$ to a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and each role name $r \in \mathsf{N}_{role}$ to a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

The extension of $\cdot^{\mathcal{I}}$ to arbitrary concept descriptions is defined inductively as follows:

$$\begin{aligned} \top^{\mathcal{I}} &:= \Delta^{\mathcal{I}} \\ (C \sqcap D)^{\mathcal{I}} &:= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\ (\exists r.C)^{\mathcal{I}} &:= \left\{ x \in \Delta^{\mathcal{I}} \;\middle|\; \exists y : (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}} \right\}. \end{aligned}$$

$\Diamond$

Next, we introduce 2 kinds of terminological formalisms for the description logic $\mathcal{EL}$: namely *$\mathcal{EL}$-TBoxes* and *general $\mathcal{EL}$-TBoxes*.

2.1 Terminology (TBox)

Definition 3 (Syntax of $\mathcal{EL}$-TBoxes). If $A \in \mathsf{N}_{con}$ and D is an $\mathcal{EL}$-concept description, then $A \equiv D$ is a *concept definition*. An *$\mathcal{EL}$-TBox* $\mathcal{T}$ is a finite set of $\mathcal{EL}$-concept definitions, which must not contain *multiple definitions*, i.e., there cannot be two distinct concept descriptions D_1 and D_2 such that both $A \equiv D_1$ and $A \equiv D_2$ belong to $\mathcal{T}$.

Cyclic definitions (or terminological cycles) are a set of concept definitions $\{A_1 \equiv D_1, \ldots, A_n \equiv D_n\} \subseteq \mathcal{T}$ for $n \geq 1$ such that

- D_i contains A_{i+1} for $1 \leq i \leq n$, and
- D_n contains A_1.

An $\mathcal{EL}$-TBox is *acyclic* if it contains no cyclic definitions, otherwise the $\mathcal{EL}$-TBox is said to be *cyclic*.

Concept names occurring on the left-hand side of a definition are called *defined concepts* and denoted by N_{def}. All other concept names are called *primitive concepts* and denoted by N_{prim}. $\Diamond$

For example, let us consider the following concept definitions.

$$\begin{aligned} \mathsf{Parent} &\equiv \mathsf{Human} \sqcap \exists \mathsf{has\_child}.\mathsf{Human} \\ \mathsf{German} &\equiv \mathsf{Human} \sqcap \exists \mathsf{has\_father}.\mathsf{German} \sqcap \exists \mathsf{has\_mother}.\mathsf{German} \qquad (*) \\ \mathsf{GermanParent} &\equiv \mathsf{German} \sqcap \exists \mathsf{has\_child}.\mathsf{Human} \end{aligned}$$

The second definition contains a cyclic dependency through the defined concept German. But since there are no multiple definitions, the set of these three concept

definitions is a TBox. Adding the new definition $\mathsf{GermanParent} \equiv \mathsf{Parent} \sqcap \mathsf{German}$ to the existing TBox seems natural. However, due to the multiple definitions, the set of these four concept definitions is not admissible as a TBox in the sense defined above.

The semantics of TBoxes is defined using models. The natural semantics for acyclic TBoxes is *descriptive semantics*. We also consider this kind of semantics for cyclic TBoxes.

Definition 4 (Descriptive semantics of $\mathcal{EL}$-TBoxes). An interpretation $\mathcal{I}$ *satisfies* a concept definition $A \equiv D$ if $A^{\mathcal{I}} = D^{\mathcal{I}}$. $\mathcal{I}$ is a *model* of an $\mathcal{EL}$-TBox $\mathcal{T}$ if it satisfies all concept definitions in $\mathcal{T}$. $\Diamond$

For TBoxes with terminological cycles, there are other kinds of semantics. They are called *fixpoint semantics* by Nebel [7]. Before we can give a formal definition of the fixpoint semantics, we need to introduce some notation.

Definition 5 (Primitive interpretation). Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox over the role names N_{role}, the primitive concepts N_{prim}, and the defined concepts N_{def}.

- A *primitive interpretation* $\mathcal{J}$ for $\mathcal{T}$ is given by a non-empty interpretation domain $\Delta^{\mathcal{J}}$ and an interpretation function $\cdot^{\mathcal{J}}$ that maps each primitive concept $P \in \mathsf{N}_{prim}$ to a subset $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{J}}$ and each role name $r \in \mathsf{N}_{role}$ to a binary relation $r^{\mathcal{J}} \subseteq \Delta^{\mathcal{J}} \times \Delta^{\mathcal{J}}$.

- An interpretation $\mathcal{I}$ is *based on* (an *extension* of) a primitive interpretation $\mathcal{J}$ iff $\mathcal{I}$ has the same interpretation domain as $\mathcal{J}$ and, the interpretation functions $\cdot^{\mathcal{I}}$ and $\cdot^{\mathcal{J}}$ coincide on N_{prim} and N_{role}.

$\Diamond$

Obviously, a primitive interpretation differs from an interpretation in that it does not interpret the defined concepts. For a fixed primitive interpretation $\mathcal{J}$, an interpretation $\mathcal{I}$ based on $\mathcal{J}$ is determined only by the interpretations of the concepts in N_{def}.

Definition 6 (Fixpoint model). Let $\mathcal{J}$ be a primitive interpretation for an $\mathcal{EL}$-TBox $\mathcal{T}$, and $Ext_{\mathcal{J}}$ the set of all extensions of $\mathcal{J}$. Then the mapping $\mathcal{T}_{\mathcal{J}} : Ext_{\mathcal{J}} \to Ext_{\mathcal{J}}$ maps the extension $\mathcal{I}$ of $\mathcal{J}$ to the extension $\mathcal{T}_{\mathcal{J}}(\mathcal{I})$ of $\mathcal{J}$ defined by setting

$$A^{\mathcal{T}_{\mathcal{J}}(\mathcal{I})} := (\mathcal{T}(A))^{\mathcal{I}} \text{ for each defined concept } A,$$

where $\mathcal{T}(A)$ denotes the concept description C if $A \equiv C \in \mathcal{T}$. An interpretation $\mathcal{I}$ is a *model* of $\mathcal{T}$ iff $\mathcal{I}$ is a fixpoint of $\mathcal{T}_{\mathcal{J}}$ with $\mathcal{J}$ the restriction of $\mathcal{J}$ to a primitive interpretation. $\mathcal{I}$ is a greatest (least) fixpoint model of $\mathcal{T}$ if $A^{\mathcal{I}} \supseteq A^{\mathcal{I}'}$ ($A^{\mathcal{I}} \subseteq A^{\mathcal{I}'}$) for every defined concept A and every fixpoint $\mathcal{I}'$ of $\mathcal{T}_{\mathcal{J}}$. $\Diamond$

Intuitively, greatest (least) fixpoint models interpret those defined concepts as large (small) as possible for a given primitive interpretation.

We are now ready to define the fixpoint semantics of TBoxes.

Definition 7 (Fixpoint semantics of $\mathcal{EL}$-TBoxes). Greatest fixpoint semantics (gfp-semantics) considers only greatest fixpoint models as admissible models, whereas least fixpoint semantics (lfp-semantics) considers only least fixpoint models as admissible models. ◊

In the DL $\mathcal{EL}$, least fixpoint semantics is uninteresting since it does not make any sense (see e.g., [1]). In fact, all defined concepts in terminological cycles are unsatisfiable w.r.t. lfp-semantics and thus can be removed from the TBox. The remaining TBox turns out to be acyclic, on which descriptive, gfp- and lfp-semantics coincide [7].

In the following, we illustrate the intuitive idea of descriptive and gfp-semantics and their contrast by giving a few examples.

Example 8. Let the following be the only concept definition in the TBox:

$$\mathsf{TopEntrepreneur} \equiv \mathsf{Entrepreneur} \sqcap \mathsf{Rich} \sqcap \exists \mathsf{deals\_with}.\mathsf{TopEntrepreneur}.$$

Intuition: 'top-entrepreneurs' is defined as 'entrepreneur who are rich and deal business with at least one top-entrepreneur.'

Now consider the following primitive interpretation $\mathcal{J}$:

$$\begin{aligned} \Delta^{\mathcal{J}} &:= \{\mathsf{MATT}, \mathsf{ANNA}, \mathsf{BOB}\}, \\ \mathsf{Entrepreneur}^{\mathcal{J}} &:= \{\mathsf{MATT}, \mathsf{ANNA}, \mathsf{BOB}\}, \\ \mathsf{Rich}^{\mathcal{J}} &:= \{\mathsf{MATT}, \mathsf{ANNA}\}, \\ \mathsf{deals\_with}^{\mathcal{J}} &:= \{(\mathsf{MATT}, \mathsf{ANNA}), (\mathsf{ANNA}, \mathsf{MATT}), (\mathsf{BOB}, \mathsf{MATT})\}. \end{aligned}$$

Obviously, BOB is not a top-entrepreneur since he is not rich, while MATT and ANNA could be. Convincingly, MATT and ANNA should be classified as top-entrepreneurs, as they are rich entrepreneurs and deal business with each other. Our claim is that there are 2 interpretations based on $\mathcal{J}$: $\mathcal{I}^{\mathcal{J}}_{gfp}$ interprets the defined concept TopEntrepreneur to the set $\{\mathsf{MATT}, \mathsf{ANNA}\}$, and $\mathcal{I}^{\mathcal{J}}_{\emptyset}$ assigns the defined concept to the empty set. Both interpretations are fixpoints of $\mathcal{T}_{\mathcal{J}}$ and therefore admissible as models of the TBox w.r.t. descriptive semantics, whereas only $\mathcal{I}^{\mathcal{J}}_{gfp}$ is the unique greatest fixpoint model based on $\mathcal{J}$ and hence admissible w.r.t. gfp-semantics. Straightforwardly, only the greatest fixpoint model captures the intuition underlying the definition of TopEntrepreneur correctly.

Note that in case of least fixpoint semantics, only $\mathcal{I}^{\mathcal{J}}_{\emptyset}$—which is the least fixpoint model based on $\mathcal{J}$—is admissible. Since the defined concept is assigned to the empty set and thus can be removed from the TBox, the remaining TBox is acyclic (actually, no definition remains). As noted earlier, this is meaningless. ⊣

This example suggests that, for some applications, the greatest fixpoint semantics turns out to be more suitable than descriptive semantics. It should be noted, however, that in other cases descriptive semantics appears to be more appropriate. For instance, consider the definitions

$$\begin{aligned} \mathsf{German} &\equiv \mathsf{Human} \sqcap \exists \mathsf{has\_father.German} \sqcap \exists \mathsf{has\_mother.German}, \\ \mathsf{Thai} &\equiv \mathsf{Human} \sqcap \exists \mathsf{has\_father.Thai} \sqcap \exists \mathsf{has\_mother.Thai}. \end{aligned}$$

With gfp-semantics, the defined concepts German and Thai must always be interpreted as the same set which clearly does not reflect the intuition. In fact, any objects on cyclic or infinite has_father- and has_mother-paths are always be classified as both German and Thai w.r.t. gfp-semantics. Whereas this is not the case for descriptive semantics.

The standard terminological inference problems, i.e., inference problems with TBoxes, are *satisfiability* of an $\mathcal{EL}$-concept description and *subsumption* of two $\mathcal{EL}$-concept descriptions. The former problem concerns whether a concept description is free of contradictions, while the latter concerns whether one concept is a subconcept of the other one. For logics without negation, concept satisfiability is uninteresting since there are no unsatisfiable concept descriptions w.r.t. descriptive semantics. With respect to gfp-semantics, this is also the case since gfp-semantics admits only gfp-models, which interpret defined concepts, as well as concept descriptions containing defined concepts, largest possible. In addition to concept subsumption, we consider *concept equivalence* as an abbreviation of concept subsumption in both direction.

Definition 9 (Subsumption). Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox and C, D arbitrary $\mathcal{EL}$-concept descriptions. Then,

- C is *subsumed* by D w.r.t. descriptive semantics and $\mathcal{T}$ ($C \sqsubseteq_{\mathcal{T}} D$) iff $C^{\mathcal{T}} \subseteq D^{\mathcal{I}}$ holds for all models $\mathcal{I}$ of $\mathcal{T}$.
- C is *subsumed* by D w.r.t. gfp-semantics and $\mathcal{T}$ ($C \sqsubseteq_{gfp,\mathcal{T}} D$) iff $C^{\mathcal{T}} \subseteq D^{\mathcal{I}}$ holds for all gfp-models $\mathcal{I}$ of $\mathcal{T}$.

C and D are *equivalent* w.r.t. descriptive semantics and $\mathcal{T}$ iff they subsume each other, i.e., $C \equiv_{\mathcal{T}} D$ iff $C \sqsubseteq_{\mathcal{T}} D$ and $D \sqsubseteq_{\mathcal{T}} C$.[1] The *equivalence* w.r.t. gfp-semantics is defined in a similar fashion to the equivalence w.r.t. descriptive semantics and denoted by $\equiv_{gfp,\mathcal{T}}$. ◊

By introducing new concept definitions in the TBox, subsumption of two arbitrary concept definitions can be reduced to the same problem of two defined concepts.

[1]Please note that $C \equiv D$ denotes a concept definition in a TBox, whereas $C \equiv_{\mathcal{T}} D$ denotes concept equivalence w.r.t. descriptive semantics and $\mathcal{T}$.

Formally, let $\mathcal{T}$ be an $\mathcal{EL}$-TBox with defined concepts N_{def}, and let C, D be $\mathcal{EL}$-concept definitions. Then, the subsumption $C \sqsubseteq_{\mathcal{T}} D$ ($C \sqsubseteq_{gfp,\mathcal{T}} D$, respectively) can be reduced to the subsumption $A \sqsubseteq_{\mathcal{T}} B$ ($A \sqsubseteq_{gfp,\mathcal{T}} B$, respectively) by introducing in $\mathcal{T}$ the following concept definitions:

$$A \equiv C \quad \text{and} \quad B \equiv D$$

where A, B new concept names in N_{def}. We can therefore restrict our attention to subsumption of two defined concepts w.r.t. descriptive and gfp-semantics.

2.2 Terminology with GCIs (General TBox)

In the previous section, we have introduced $\mathcal{EL}$-terminologies, which are sets of concept definitions ($A \equiv D$). In this section, we define a new terminological formalism which is a generalization of TBoxes with descriptive semantics.

Definition 10 (Syntax of general $\mathcal{EL}$-TBoxes). If C and D are $\mathcal{EL}$-concept descriptions, then $C \sqsubseteq D$ is a *general concept inclusion* or *GCI*.[2] A *general $\mathcal{EL}$-TBox* (or simply a general TBox) $\mathcal{T}$ is a finite set of general concept inclusions. ◊

Since we explicitly allow arbitrary concept descriptions on both sides of a GCI, it is not clear which concept names are defined and which are primitive. Therefore, for general $\mathcal{EL}$-TBoxes, we do not distinguish defined concepts from primitive ones. All concept names occurring in a general $\mathcal{EL}$-TBox $\mathcal{T}$ are denoted by N_{con}, as role names by N_{role} as before. We supplementally define a concept definition as an abbreviation of two GCIs, i.e., the concept definition $A \equiv D$ is an abbreviation of $A \sqsubseteq D$ and $D \sqsubseteq A$.

Unlike TBoxes, it does not make much sense to consider general TBoxes in respect of fixpoint semantics. In fact, fixpoint semantics of TBoxes is determined by the interpretations of defined concepts, provided a primitive interpretation. Since, as mentioned above, we have neither defined concepts nor primitive ones in general TBoxes, we will only consider them w.r.t. descriptive semantics. Like TBoxes, the semantics of general TBoxes is defined through interpretations.

Definition 11 (Semantics of general $\mathcal{EL}$-TBoxes). An interpretation $\mathcal{I}$ *satisfies* a GCI $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. $\mathcal{I}$ is a *model* of a general $\mathcal{EL}$-TBox $\mathcal{T}$ if it satisfies all GCIs in $\mathcal{T}$. ◊

In the example on page 8, the TBox contains a single concept definition ($*$) of German

$$\mathsf{German} \equiv \mathsf{Human} \sqcap \exists \mathsf{has\_father}.\mathsf{German} \sqcap \exists \mathsf{has\_mother}.\mathsf{German},$$

[2]Please note that $C \sqsubseteq D$ denotes a GCI in a general $\mathcal{EL}$-TBox, whereas $C \sqsubseteq_{\mathcal{T}} D$ denotes concept subsumption w.r.t. descriptive semantics and $\mathcal{T}$.

whose intuitive meaning is destined for German citizens. Nevertheless, the concept definition does not encode the notion of "citizenship" correctly. German-born are always citizens of Germany, but at times people receive their German citizenship without having both German father and mother. This situation suggests that concept definitions ($\equiv$) are too strong, and it is sometimes more appropriate to use GCIs ($\sqsubseteq$). The refined TBox should contain the GCI

$$\mathsf{Human} \sqcap \exists \mathsf{has\_father.German} \sqcap \exists \mathsf{has\_mother.German} \sqsubseteq \mathsf{German}$$

in place of the concept definition ($*$).

In order to gain a clearer idea of general TBoxes, let us consider the following example. This example is a simplified version of a similar one in [4], where the simple role inclusion constraint is omitted.

Example 12. The following general TBox shows a simplified piece of terminology in a medical knowledge-base.[3]

$$\begin{aligned} \mathsf{Pericardium} &\sqsubseteq \mathsf{Tissue} \sqcap \exists \mathsf{cont\_in.Heart} \\ \mathsf{Pericarditis} &\sqsubseteq \mathsf{Inflammation} \sqcap \exists \mathsf{has\_loc.Pericardium} \\ \mathsf{Inflammation} &\sqsubseteq \mathsf{Disease} \sqcap \exists \mathsf{acts\_on.Tissue} \\ \mathsf{Disease} \sqcap \exists \mathsf{has\_loc.} \exists \mathsf{cont\_in.Heart} &\sqsubseteq \mathsf{Heartdisease} \sqcap \exists \mathsf{is\_state.NeedsTreatment} \end{aligned}$$

The TBox contains four GCIs, asserting that pericardium is tissue contained in the heart, that pericarditis is an inflammation located in the pericardium, that an inflammation is a disease and acts on tissue, and that a disease located in the heart is a heart disease and requires treatment. Without going into detail, one can naturally check that pericarditis would be classified as a heart disease requiring treatment. $\dashv$

Similar to the previous section, we are interested in subsumption of two $\mathcal{EL}$-concept descriptions with respect to general TBoxes and descriptive semantics.

Definition 13 (Subsumption). Let $\mathcal{T}$ be a general $\mathcal{EL}$-TBox and C, D arbitrary $\mathcal{EL}$-concept descriptions. Then, C is *subsumed* by D w.r.t. $\mathcal{T}$ ($C \sqsubseteq_{gci,\mathcal{T}} D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{T}$.

C and D are *equivalent* w.r.t. $\mathcal{T}$ ($C \equiv_{gci,\mathcal{T}} D$) iff they mutually subsume each other, i.e., $C \sqsubseteq_{gci,\mathcal{T}} D$ and $D \sqsubseteq_{gci,\mathcal{T}} C$. $\Diamond$

Subsumption of two arbitrary concept descriptions can be reduced to subsumption of two concept names. Precisely, let $\mathcal{T}$ be a general TBox over concept names N_{con} and role names N_{role}, and let C, D be arbitrary $\mathcal{EL}$-concept descriptions.

[3] The example is only supposed to show the features of $\mathcal{EL}$ with general TBoxes and by no means claims to be correct nor adequate from a medical point of view.

Then, the subsumption $C \sqsubseteq_{\mathcal{T}} D$ can be reduced to the subsumption $A \sqsubseteq_{\mathcal{T}} B$ by introducing in $\mathcal{T}$ the following GCIs:

$$C \sqsubseteq A \quad \text{and} \quad B \sqsubseteq D$$

where A, B are *new* concept names in N_{con}. In the following, we will consider, without loss of generality, subsumption of two concept names w.r.t. general TBoxes.

As discussed in this chapter, there are three subsumption problems of interest in the DL $\mathcal{EL}$, two problems w.r.t. TBoxes and one w.r.t. general TBoxes. For the sake of conciseness and consistency, we will use the following abbreviations throughout this paper:

- $\mathcal{EL}^{gfp}$ **:** the logic $\mathcal{EL}$ and TBoxes interpreted w.r.t. gfp-semantics,
- $\mathcal{EL}^{desc}$ **:** the logic $\mathcal{EL}$ and TBoxes interpreted w.r.t. descriptive semantics, and
- $\mathcal{EL}^{gci}$ **:** the logic $\mathcal{EL}$ and general TBoxes equipped with GCIs.

We denote by $\mathcal{EL}^{gfp}$-subsumption the subsumption problem of two concepts in DL $\mathcal{EL}$ w.r.t. TBoxes and gfp-semantics. $\mathcal{EL}^{desc}$-subsumption and $\mathcal{EL}^{gci}$-subsumption denote the corresponding subsumption problems in an obvious way.

Chapter 3

Normalized $\mathcal{EL}$-TBoxes and $\mathcal{EL}$-Description Graphs

Subsumption problems in $\mathcal{EL}^{gfp}$ and in $\mathcal{EL}^{desc}$ can be characterized through graph simulation. The characterizations of both problems share the same idea of normalized $\mathcal{EL}$-TBoxes and $\mathcal{EL}$-description graphs, which are defined below in this chapter. We will also show that $\mathcal{EL}$-TBoxes can be translated into $\mathcal{EL}$-description graphs, which can be understood as a preprocessing step for the algorithms presented in Chapter 4 and 5 for $\mathcal{EL}^{gfp}$- and $\mathcal{EL}^{desc}$-subsumption, respectively. But before we can do such a translation of TBoxes, we must first normalize them.

3.1 Normalized $\mathcal{EL}$-TBoxes

Definition 14 (Normalized $\mathcal{EL}$-TBoxes). Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox with defined concepts N_{def}, primitive concepts N_{prim}, and role names N_{role}. Then, $\mathcal{T}$ is *normalized* (or *in normal form*) iff $A \equiv D \in \mathcal{T}$ implies that D is of the form

$$P_1 \sqcap \cdots \sqcap P_m \sqcap \exists r_1.B_1 \sqcap \cdots \sqcap \exists r_l.B_l,$$

with $m, l \geq 0,\ P_1, \cdots, P_m \in \mathsf{N}_{prim},\ r_1, \cdots, r_l \in \mathsf{N}_{role}$, and $B_1, \cdots, B_l \in \mathsf{N}_{def}$. If $m = l = 0$, then $D = \top$. $\Diamond$

Next, we will show how $\mathcal{EL}$-TBoxes can be converted into normal form. The same $\mathcal{EL}$-TBox may be normalized to two different normalized $\mathcal{EL}$-TBoxes, depending on the semantics used. We will see that this is due to different treatments of cyclic dependencies w.r.t. gfp- and descriptive semantics. First, we illustrate the normalization process by a typical example. Then, a formal definition of normalization rules is given.

Example 15. Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox comprising only the following concept definitions:

$$\begin{aligned} A_1 &\equiv P_1 \sqcap A_2 \sqcap \exists r_1.\exists r_2.A_3, \\ A_2 &\equiv P_2 \sqcap A_3 \sqcap \exists r_2.\exists r_1.A_1, \\ A_3 &\equiv P_3 \sqcap A_2 \sqcap \exists r_1.(P_1 \sqcap P_2). \end{aligned}$$

By introducing auxiliary definitions for complex concepts nested in existential restrictions, we obtain the new $\mathcal{EL}$-TBox $\mathcal{T}'$:

$$\begin{aligned} A_1 &\equiv P_1 \sqcap A_2 \sqcap \exists r_1.B_1, \\ B_1 &\equiv \exists r_2.A_3, \\ A_2 &\equiv P_2 \sqcap A_3 \sqcap \exists r_2.B_2, \\ B_2 &\equiv \exists r_1.A_1, \\ A_3 &\equiv P_3 \sqcap A_2 \sqcap \exists r_1.B_3, \\ B_3 &\equiv P_1 \sqcap P_2. \end{aligned}$$

Due to the occurrences of defined concepts in the top-level conjunction of the definitions of A_1, A_2 and A_3, none of them are yet normalized.

Let us first detect cyclic dependencies in the top-level conjunction of the definitions in $\mathcal{T}'$. Obviously, there is such a top-level cycle through the defined concepts A_2 and A_3. The occurrence of A_3 in the top-level conjunction of the definition of A_2 implies that A_2 is subsumed by A_3. By the same argument for the definition of A_3, it also holds that A_3 is subsumed by A_2. Hence, the defined concepts A_2 and A_3 are equivalent.[1] Moreover, both A_2 and A_3 are subsumed by $P_2 \sqcap P_3 \sqcap \exists r_2.B_2 \sqcap \exists r_1.B_3$. We can therefore replace the definitions of A_2 and A_3 by the general concept inclusions (GCIs)

$$\begin{aligned} A_2 &\sqsubseteq P_2 \sqcap P_3 \sqcap \exists r_2.B_2 \sqcap \exists r_1.B_3 \text{ and} \\ A_3 &\sqsubseteq P_2 \sqcap P_3 \sqcap \exists r_2.B_2 \sqcap \exists r_1.B_3, \end{aligned}$$

respectively. We now have the following terminology with two GCIs:

$$\begin{aligned} A_1 &\equiv P_1 \sqcap A_2 \sqcap \exists r_1.B_1, \\ B_1 &\equiv \exists r_2.A_3, \\ A_2 &\sqsubseteq P_2 \sqcap P_3 \sqcap \exists r_2.B_2 \sqcap \exists r_1.B_3, \\ B_2 &\equiv \exists r_1.A_1, \\ A_3 &\sqsubseteq P_2 \sqcap P_3 \sqcap \exists r_2.B_2 \sqcap \exists r_1.B_3, \\ B_3 &\equiv P_1 \sqcap P_2. \end{aligned}$$

[1] They are interpreted by the same set in all models of the TBox $\mathcal{T}'$, i.e., $A_2 \equiv_{\mathcal{T}'} A_3$ as well as $A_2 \equiv_{gfp,\mathcal{T}'} A_3$.

This is plainly not a TBox by definition. In order to convert this terminology back into a TBox, we must remove the two GCIs. How to do this depends on the semantics used for cyclic definitions.

If we apply *gfp-semantics*, then the GCIs can respectively be replaced by the definitions

$$\begin{aligned} A_2 &\equiv P_2 \sqcap P_3 \sqcap \exists r_2.B_2 \sqcap \exists r_1.B_3 \text{ and} \\ A_3 &\equiv P_2 \sqcap P_3 \sqcap \exists r_2.B_2 \sqcap \exists r_1.B_3, \end{aligned}$$

Indeed, these are the largest possible interpretations of A_2 and correspondingly A_3 that the the GCIs allow. Let $\mathcal{T}'_{gfp}$ denote the $\mathcal{EL}$-TBox obtained in this way.

On the contrary, if *descriptive semantics* is considered for the TBox, then we introduce a new primitive concept P and replace the GCIs by the definitions

$$\begin{aligned} A_2 &\equiv P \sqcap P_2 \sqcap P_3 \sqcap \exists r_2.B_2 \sqcap \exists r_1.B_3 \text{ and} \\ A_3 &\equiv P \sqcap P_2 \sqcap P_3 \sqcap \exists r_2.B_2 \sqcap \exists r_1.B_3, \end{aligned}$$

The auxiliary primitive concept P here allows A_2 and A_3 to be interpreted arbitrarily as long as the GCIs are satisfied. Let $\mathcal{T}'_{des}$ denote the $\mathcal{EL}$-TBox obtained in this way.

Neither $\mathcal{T}'_{gfp}$ nor $\mathcal{T}'_{des}$ is already in normal form since the definition A_1 still refers to A_2 on the top-level. However, we can now simply replace the top-level A_2 in the definition of A_1 by its defining concept description. Ultimately, we end up with two normalized $\mathcal{EL}$-TBoxes. With respect to gfp-semantics, we thus obtain the normalized $\mathcal{EL}$-TBox $\mathcal{T}_{gfp}$:

$$\begin{aligned} A_1 &\equiv P_1 \sqcap P_2 \sqcap P_3 \sqcap \exists r_2.B_2 \sqcap \exists r_1.B_3 \sqcap \exists r_1.B_1, \\ B_1 &\equiv \exists r_2.A_3, \\ A_2 &\equiv P_2 \sqcap P_3 \sqcap \exists r_2.B_2 \sqcap \exists r_1.B_3, \\ B_2 &\equiv \exists r_1.A_1, \\ A_3 &\equiv P_2 \sqcap P_3 \sqcap \exists r_2.B_2 \sqcap \exists r_1.B_3, \\ B_3 &\equiv P_1 \sqcap P_2; \end{aligned}$$

and w.r.t. descriptive semantics, we obtain the normalized $\mathcal{EL}$-TBox $\mathcal{T}_{desc}$:

$$\begin{aligned} A_1 &\equiv P_1 \sqcap P \sqcap P_2 \sqcap P_3 \sqcap \exists r_2.B_2 \sqcap \exists r_1.B_3 \sqcap \exists r_1.B_1, \\ B_1 &\equiv \exists r_2.A_3, \\ A_2 &\equiv P \sqcap P_2 \sqcap P_3 \sqcap \exists r_2.B_2 \sqcap \exists r_1.B_3, \\ B_2 &\equiv \exists r_1.A_1, \\ A_3 &\equiv P \sqcap P_2 \sqcap P_3 \sqcap \exists r_2.B_2 \sqcap \exists r_1.B_3, \\ B_3 &\equiv P_1 \sqcap P_2. \end{aligned}$$

⊣

The normalization process illustrated in the above example consists of 3 essential steps, which are

1. the introduction of auxiliary defined concepts, together with their concept definitions, for non-defined subconcepts of existential restrictions,

2. the detection and removal of cyclic dependencies of defined concepts in the top-level conjunction of concept definitions,

3. the substitution of defined concepts that occur in the top-level conjunction of a concept definition with their defining concept descriptions.

As shown in the example, the second step is carried out differently w.r.t. different semantics, whereas the first and third steps are identical regardless what semantics is being considered.

We generalize the 3 steps of the normalization process to arbitrary $\mathcal{EL}$-TBoxes by means of *$\mathcal{EL}$-normalization rules*.

Definition 16 (Normalization rules). Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox, N_{def} the defined concepts of $\mathcal{T}$, N_{prim} the primitive concepts of $\mathcal{T}$, and N_{role} the roles of $\mathcal{T}$. The *normalization rule*s are defined modulo commutativity of conjunction as follows:

$$\textsf{NF1} \qquad \{\ A \equiv \exists r.\hat{C} \sqcap C\ \} \quad\longrightarrow\quad \{\ A \equiv \exists r.B \sqcap C\ ,\ B \equiv \hat{C}\},\ \text{with } B \text{ a new concept name}$$

$$\textsf{NF2}^{gfp} \qquad \left\{ \begin{array}{c} A_1 \equiv A_2 \sqcap C_1, \\ A_2 \equiv A_3 \sqcap C_2, \\ \vdots \\ A_l \equiv A_1 \sqcap C_l \end{array} \right\} \quad\longrightarrow\quad \left\{ \begin{array}{l} A_i \equiv C_1 \sqcap \ldots \sqcap C_l \\ \text{for } 1 \leq i \leq l \end{array} \right\}$$

$$\textsf{NF2}^{desc} \qquad \left\{ \begin{array}{c} A_1 \equiv A_2 \sqcap C_1, \\ A_2 \equiv A_3 \sqcap C_2, \\ \vdots \\ A_l \equiv A_1 \sqcap C_l \end{array} \right\} \quad\longrightarrow\quad \left\{ \begin{array}{l} A_i \equiv P \sqcap C_1 \sqcap \ldots \sqcap C_l \\ \text{for } 1 \leq i \leq l \end{array} \right\},\ \text{with } P \text{ a new concept name}$$

$$\textsf{NF3} \qquad \{\ A \equiv A' \sqcap C\ ,\ A' \equiv \bar{C}\ \} \quad\longrightarrow\quad \{\ A \equiv \bar{C} \sqcap C\ ,\ A' \equiv \bar{C}\ \}$$

where A, A', A_i denote concept names, C and C_i any concept descriptions (possibly complex concepts or $\top$), $\hat{C}$ a non-defined concept (either primitive or complex concept), and $\bar{C}$ a concept definition with no defined concepts occurring in the top-level conjunction. Multiple occurrences of conjuncts but one are immediately

eliminated after each application of Rules **NF2**^{gfp}, **NF2**^{desc} and **NF3** (idempotency of $\sqcap$).

Applying a rule $\mathcal{S}^{Left} \longrightarrow \mathcal{S}^{Right}$ to $\mathcal{T}$ changes $\mathcal{T}$ to $(\mathcal{T} \backslash \mathcal{S}^{Left}) \cup \mathcal{S}^{Right}$. The normalized $\mathcal{EL}$-TBox of $\mathcal{T}$ w.r.t. gfp-semantics—denoted by $norm_{gfp}(\mathcal{T})$—is defined by exhaustively applying Rules **NF1**; then **NF2**^{gfp}; and finally, **NF3**. The normalized $\mathcal{EL}$-TBox of $\mathcal{T}$ w.r.t. descriptive semantics—denoted by $norm_{desc}(\mathcal{T})$—is defined analogously but with **NF2**^{desc} instead of **NF2**^{gfp}. $\Diamond$

We write $|\mathcal{T}|$ to denote the size of an $\mathcal{EL}$-TBox $\mathcal{T}$, i.e., the total number of *all occurrences* of concept names and role names in $\mathcal{T}$. Note that it is crucial to consider idempotency of conjunction right after each application of the last two rules. This is to avoid an exponential blow-up as illustrated in the following example.

Example 17. Consider the following $\mathcal{EL}$-TBox $\mathcal{T}$ with $n \geq 2$ concept definitions:

$$\begin{array}{rcl} A_1 & \equiv & P_1 \sqcap P_2, \\ A_2 & \equiv & A_1 \sqcap A_1, \\ & \vdots & \\ A_n & \equiv & A_{n-1} \sqcap A_{n-1}, \end{array}$$

where P_1, P_2 and A_i are concept names. Without considering the idempotency of $\sqcap$, the size of the normalized TBox will obviously be exponential in the size of the original one with many occurrences of P_1 and P_2. Indeed, the size of $norm_{gfp}(\mathcal{T})$ as well as $norm_{desc}(\mathcal{T})$ precisely equates to that of $\mathcal{T}$. $\dashv$

With this $\mathcal{EL}$-TBox reduction, the size of normalized $\mathcal{EL}$-TBoxes may be blown up quadratically in the size of the original ones. This is a consequence of exhaustive application of Rules **NF2**^{gfp}, **NF2**^{desc} or **NF3**. We demonstrate this by a couple of examples as follows.

Example 18. Let $\mathcal{T}_1$ be an $\mathcal{EL}$-TBox containing the following concept definitions:

$$\begin{array}{rcl} A_1 & \equiv & A_2 \sqcap P_1, \\ A_2 & \equiv & A_3 \sqcap P_2, \\ & \vdots & \\ A_n & \equiv & A_1 \sqcap P_n, \end{array}$$

with A_i defined concepts and P_i primitive concepts for $1 \leq i \leq n$ and $n \geq 1$. $\mathcal{T}_1$ is already normalized w.r.t. Rule **NF1**, and its size is linear in n (i.e., $|\mathcal{T}_1| = 3 \cdot n$). By exhaustive application of Rule **NF2**^{gfp}, we obtain the normalized $\mathcal{EL}$-TBox

$norm_{gfp}(\mathcal{T}_1)$ as follows:

$$\begin{aligned} A_1 &\equiv P_1 \sqcap P_2 \sqcap \cdots \sqcap P_n, \\ A_2 &\equiv P_1 \sqcap P_2 \sqcap \cdots \sqcap P_n, \\ &\vdots \\ A_n &\equiv P_1 \sqcap P_2 \sqcap \cdots \sqcap P_n. \end{aligned}$$

This is obviously quadratic in the size of $\mathcal{T}_1$ (i.e., $|norm_{gfp}(\mathcal{T}_1)| = n \cdot (n+1)$).

Now let us consider another $\mathcal{EL}$-TBox $\mathcal{T}_2$ containing the following concept definitions:

$$\begin{aligned} A_1 &\equiv B, \\ A_2 &\equiv B, \\ &\vdots \\ A_n &\equiv B, \text{ and} \\ B &\equiv \exists r.A_1 \sqcap \exists r.A_2 \sqcap \cdots \sqcap \exists r.A_n, \end{aligned}$$

with r a role name and A_i, B defined concepts for $1 \leq i \leq n$ and $n \geq 1$. The size of $\mathcal{T}_2$ is linear in n (i.e., $|\mathcal{T}_2| = 4 \cdot n + 1$). $\mathcal{T}_2$ is also normalized w.r.t. Rule **NF1**, and since it contains no cycles, Rule $\mathbf{NF2}^{gfp}$ is not applicable. We can however apply the last normalization rule n times, once to each definition of A_i. As a result, we obtain the normalized $\mathcal{EL}$-TBox $norm_{gfp}(\mathcal{T}_2)$ as follows:

$$\begin{aligned} A_1 &\equiv \exists r.A_1 \sqcap \exists r.A_2 \sqcap \cdots \sqcap \exists r.A_n, \\ A_2 &\equiv \exists r.A_1 \sqcap \exists r.A_2 \sqcap \cdots \sqcap \exists r.A_n, \\ &\vdots \\ A_n &\equiv \exists r.A_1 \sqcap \exists r.A_2 \sqcap \cdots \sqcap \exists r.A_n, \\ B &\equiv \exists r.A_1 \sqcap \exists r.A_2 \sqcap \cdots \sqcap \exists r.A_n, \end{aligned}$$

which is quadratic in n (i.e., $|norm_{gfp}(\mathcal{T}_2)| = (n+1) \cdot (2 \cdot n + 1)$).

The normalization of $\mathcal{T}_2$ w.r.t. descriptive semantics yields the same normalized $\mathcal{EL}$-TBox as w.r.t. gfp-semantics, i.e., $norm_{gfp}(\mathcal{T}_2) = norm_{desc}(\mathcal{T}_2)$. Nevertheless, this is not the case for $\mathcal{T}_1$. $norm_{desc}(\mathcal{T}_1)$ is slightly bigger than $norm_{gfp}(\mathcal{T}_1)$, due to the introduction of a new primitive concept. ⊣

Having seen an example of quadratic blow-up, we now want to show that the normalization will not be worse than this in general.

Lemma 19. *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox. The normalized $\mathcal{EL}$-TBoxes $norm_{gfp}(\mathcal{T})$ and $norm_{desc}(\mathcal{T})$ w.r.t. gfp-semantics and descriptive semantics, respectively, can be computed in time quadratic in $|\mathcal{T}|$, and the resulting ontologies are of size quadratic in $|\mathcal{T}|$.*

Proof. Let us fix an $\mathcal{EL}$-TBox $\mathcal{T}$ with defined concept names N_{def}, primitive concept names N_{prim} and role names N_{role}. The size of the TBox increases only linearly by exhaustive application of Rule **NF1**. To be more precise, Rule **NF1** is applicable at most once for each occurrence of existential restriction in $\mathcal{T}$, and an application of Rule **NF1** increases the size of $\mathcal{T}$ only by a constant, introducing a new defined concept. Let $\mathcal{T}'$ denote the resulting TBox after this phase of rule applications, and N'_{def} (N'_{prim}) denote the defined (primitive) concept names in $\mathcal{T}'$. Clearly, N'_{prim} remains identical to N_{prim}. Rule $\mathbf{NF2}^{gfp}/\mathbf{NF2}^{desc}$ is applicable once for each cycle of defined concepts in the top-level conjunction in concept definitions, and this cycle is removed forever from the TBox. For Rule $\mathbf{NF2}^{desc}$, one primitive concept name is introduced for each application. Since the number of such cycles is bounded by $|\mathsf{N}'_{def}|$, Rule $\mathbf{NF2}^{gfp}/\mathbf{NF2}^{desc}$ can be applied only linearly many times. Let $\mathcal{T}''$ be the resulting TBox after this phase with defined concepts N''_{def} and primitive concepts N''_{prim}. With respect to gfp-semantics, neither primitive nor defined concept names are introduced in this phase, whereas only linearly many new primitive concepts are introduced in case of descriptive semantics. Thus, N''_{def} and N''_{prim} are bounded by $|\mathcal{T}'|$, implying $|\mathcal{T}|$. Rule **NF3** can be applied once for each defined concept in the top-level conjunction of a concept definition in $\mathcal{T}''$, and then the defined concept is removed from this definition. Additionally, since no new defined concepts are introduced in the top-level conjunction by this rule application, the number of such defined concepts decreases. Due to idempotency of $\mathbf{NF2}^{gfp}/\mathbf{NF2}^{desc}$, there can be at most $|\mathsf{N}''_{def}|$ defined concepts in each definition in $\mathcal{T}''$. Since there are $|\mathsf{N}''_{def}|$ definitions, this rule is applicable at most $|\mathsf{N}''_{def}|^2$ times. This is still quadratic in the size of the original TBox, i.e., $\mathcal{O}(|\mathcal{T}|^2)$.

Because of idempotency of conjunction of $\mathbf{NF2}^{gfp}/\mathbf{NF2}^{desc}$ and **NF3**, each concept definition in $norm_{gfp}(\mathcal{T})/norm_{desc}(\mathcal{T})$ may have at most linearly many conjuncts: in the worst case every primitive concepts ($|\mathsf{N}''_{prim}|$) and existential restrictions occurring in $\mathcal{T}''$ (bounded by $|\mathcal{T}|$). Moreover, there are only linearly many concept definitions ($|\mathsf{N}''_{def}|$) in these normalized TBoxes. Thus, the size of the normalized TBox $norm_{desc}(\mathcal{T})$, as well as $norm_{gfp}(\mathcal{T})$, is quadratic in $|\mathcal{T}|$. ❑

Since the first normalization rule only introduce auxiliary definition, subsumption between defined concepts in $\mathcal{T}$ is preserved. As already claimed in Example 15, all defined concepts in a cycle in the top-level conjunction are equivalent. Hence, the last two rules only replace concepts defined in the top-level conjunction with their equivalent definitions. This obviously preserves subsumption between defined concepts in $\mathcal{T}$. The following proposition shows that an $\mathcal{EL}$-TBox and its normal form are equivalent with respect to concept subsumption.

Proposition 20. *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox and A, B defined concepts in $\mathcal{T}$. Then, it holds that $A \sqsubseteq_{\mathcal{T}} B$ iff $A \sqsubseteq_{norm_{desc}(\mathcal{T})} B$, and $A \sqsubseteq_{gfp,\mathcal{T}} B$ iff $A \sqsubseteq_{gfp,norm_{gfp}(\mathcal{T})} B$.*

3.2 $\mathcal{EL}$-Description Graphs

Thus regardless of the complexity, we can now assume without loss of generality that all TBoxes are normalized (w.r.t. either gfp- or descriptive semantics). Normalized $\mathcal{EL}$-TBoxes can be viewed as graphs whose nodes are the defined concepts, which are labeled by sets of primitive concepts, and whose edges are given by the existential restrictions.

Definition 21 ($\mathcal{EL}$-description graphs). Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox in normal form over primitive concepts N_{prim} and defined concepts N_{def}. An *$\mathcal{EL}$-description graph* based on N_{prim}, N_{def} and N_{role} is a graph $\mathcal{G} = (V, E, L)$ where

- $V := \mathsf{N}_{def}$ is the set of nodes,
- $E \subseteq V \times \mathsf{N}_{role} \times V$ is the set of edges labeled by role names,
- $L : V \to 2^{\mathsf{N}_{prim}}$ is a function that labels nodes with sets of primitive concepts.

The *corresponding* $\mathcal{EL}$-description graph of $\mathcal{T}$, denoted by $\mathcal{G}_\mathcal{T}$, is a graph $\mathcal{G}_\mathcal{T} = (V_\mathcal{T}, E_\mathcal{T}, L_\mathcal{T})$ such that $V_\mathcal{T} = \mathsf{N}_{def}$ and if A is a defined concept with definition $P_1 \sqcap \cdots \sqcap P_m \sqcap \exists r_1.B_1 \sqcap \cdots \sqcap \exists r_l.B_l$ in $\mathcal{T}$, then

- $L_\mathcal{T}(A) = \{P_1, \cdots, P_m\}$, and
- A is the source of the edges $(A, r_1, B_1), \cdots, (A, r_l, B_l) \in E_\mathcal{T}$.

$\Diamond$

The translation from $\mathcal{EL}$-TBoxes to $\mathcal{EL}$-description graphs works in both directions, i.e., any $\mathcal{EL}$-description graph can also be viewed as an $\mathcal{EL}$-TBox. For example, the $\mathcal{EL}$-description graph of the normalized $\mathcal{EL}$-TBox w.r.t. gfp-semantics ($\mathcal{T}_{gfp}$) in Example 15 is depicted in Figure 3.1.

In the following, we write $|\mathcal{G}|$ to denote the size of an $\mathcal{EL}$-description graph $\mathcal{G} = (V, E, L)$, i.e., the summation of the cardinalities of nodes $|V|$ and edges $|E|$.

Lemma 22. *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox, not necessarily in normal form, and $\mathcal{G}_\mathcal{T} = (V_\mathcal{T}, E_\mathcal{T}, L_\mathcal{T})$ be the corresponding $\mathcal{EL}$-description graph of $\mathcal{T}$. Then,*

1. *the number of nodes $|V_\mathcal{T}|$ is linear in the size of $\mathcal{T}$, and the number of edges $|E_\mathcal{T}|$ is quadratic in the size of $\mathcal{T}$; and,*
2. *the size of $\mathcal{G}_\mathcal{T}$ is quadratic in the size of $\mathcal{T}$.*

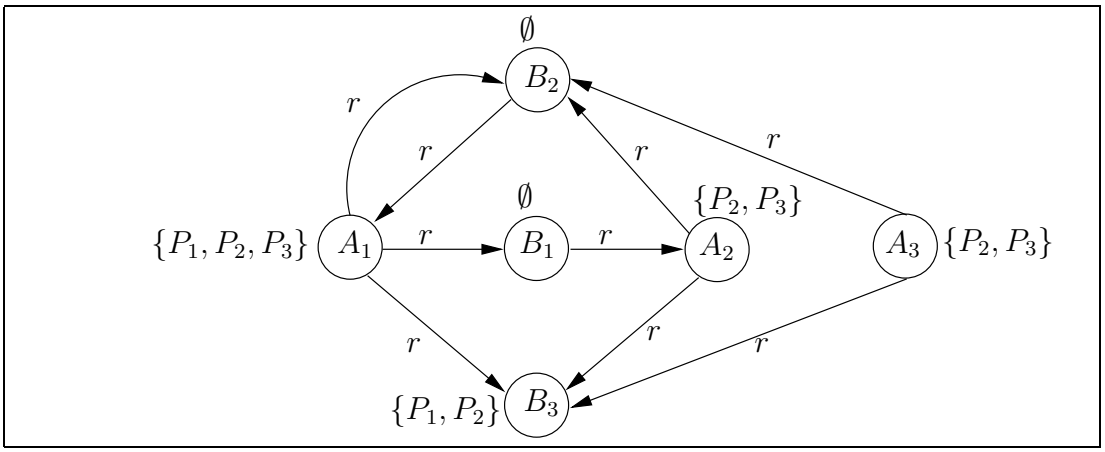

Figure 3.1: The $\mathcal{EL}$-description graph of the normalized $\mathcal{EL}$-TBox w.r.t. gfp-semantics $\mathcal{T}_{gfp}$ in Example 15

Proof. Since the size of $\mathcal{G}_\mathcal{T}$ is determined by the number of nodes and edges in $\mathcal{G}_\mathcal{T}$, *(2)* immediately follows from *(1)*.

Nodes in $\mathcal{G}_\mathcal{T}$ corresponds to defined concepts in the normalized $\mathcal{EL}$-TBox of $\mathcal{T}$. A new defined concept is introduced by each application of Normalization Rule **NF1** and not by the others. Since the rule is applied only linearly many times in the size of $\mathcal{T}$, there are a linear number of nodes in $\mathcal{G}_\mathcal{T}$.

An edge (A, r, B) in $\mathcal{G}_\mathcal{T}$ corresponds to the occurrence of the subconcept $\exists r.B$ in the top-level conjunction of the definition of A in the normalized $\mathcal{EL}$-TBox $\mathcal{T}'$ of $\mathcal{T}$. The number of occurrences of such existential restriction is bounded by the size of $\mathcal{T}'$, and by Lemma 19 the size of $\mathcal{T}'$ is quadratic in the size of $\mathcal{T}$. Thus, $|E_\mathcal{T}|$ is quadratic in the size of $\mathcal{T}$. ❑

Chapter 4

A Cubic-time Algorithm for $\mathcal{EL}^{gfp}$-Subsumption

In this section, we present a cubic-time algorithm for $\mathcal{EL}^{gfp}$-subsumption. This is attained with the help of the characterization of $\mathcal{EL}^{gfp}$-subsumption through so-called simulations of $\mathcal{EL}$-description graphs and an efficient algorithm for computing such simulations [8]. More precisely, we have shown how to translate $\mathcal{EL}$-TBoxes into $\mathcal{EL}$-description graphs w.r.t. gfp-semantics. Next, we will introduce the notion of a simulation between nodes of an $\mathcal{EL}$-description graph and show some useful properties of simulations. To put it another way, we reduce $\mathcal{EL}^{gfp}$-subsumption w.r.t. an $\mathcal{EL}$-TBox to the simulation problem on the corresponding $\mathcal{EL}$-description graph.

Simulations are binary relations between nodes of two $\mathcal{EL}$-description graphs that respect labels and edges in the sense defined below.

Definition 23 (Simulation). Let $\mathcal{G}_i = (V_i, E_i, L_i)$ (for $i = 1, 2$) be two $\mathcal{EL}$-description graphs. The binary relation $Z \subseteq V_1 \times V_2$ is a *simulation* from $\mathcal{G}_1$ to $\mathcal{G}_2$ iff

(S1) $(v_1, v_2) \in Z$ implies $L_1(v_1) \subseteq L_2(v_2)$; and

(S2) if $(v_1, v_2) \in Z$ and $(v_1, r, v_1') \in E_1$, then there exists a node v_2' such that $(v_1', v_2') \in Z$ and $(v_2, r, v_2') \in E_2$.

We write $Z : \mathcal{G}_1 \simeq \mathcal{G}_2$ to express that Z is a simulation from $\mathcal{G}_1$ to $\mathcal{G}_2$. $\Diamond$

It is not hard to convince ourselves that the set of all simulations from $\mathcal{G}_1$ to $\mathcal{G}_2$ is closed under set union, i.e., if $Z_1 : \mathcal{G}_1 \simeq \mathcal{G}_2$ and $Z_2 : \mathcal{G}_1 \simeq \mathcal{G}_2$ are simulations from $\mathcal{G}_1$ to $\mathcal{G}_2$, then so is their union $Z_1 \cup Z_2 : \mathcal{G}_1 \simeq \mathcal{G}_2$. As a result, there always exists *the greatest simulation* $\hat{Z}$ from $\mathcal{G}_1$ to $\mathcal{G}_2$ that is obtained by taking the union of all the simulations between those two graphs.

Definition 23 also covers the case where $\mathcal{G}_1 = \mathcal{G} = \mathcal{G}_2$. In this case, we write $Z : \mathcal{G} \simeq \mathcal{G}$ to express that Z is a simulation *on* $\mathcal{G}$. The identity relation on the nodes of $\mathcal{G}$ is a simulation on $\mathcal{G}$ and, consequently, is contained in the greatest simulation on $\mathcal{G}$.

In the following, let $\mathcal{T}$ be a normalized $\mathcal{EL}$-TBox w.r.t. gfp-semantics with primitive concepts N_{prim}, defined concepts N_{def}, and roles N_{role}. The characterization of $\mathcal{EL}^{gfp}$-subsumption through simulations on $\mathcal{EL}$-description graph has been shown in [1]. The result is stated by the following theorem.

Theorem 24. ***[Baader]*** *Let A, B be defined concepts in $\mathcal{T}$ and $\mathcal{G}_\mathcal{T}$ the corresponding $\mathcal{EL}$-description graph of $\mathcal{T}$. Then the following are equivalent:*

1. *$A \sqsubseteq_{gfp,\mathcal{T}} B$.*
2. *There is a simulation $Z : \mathcal{G}_\mathcal{T} \simeq \mathcal{G}_\mathcal{T}$ such that $(B, A) \in Z$.*

This theorem provides us with a means to answer $\mathcal{EL}^{gfp}$-subsumption queries w.r.t. an $\mathcal{EL}$-TBox $\mathcal{T}$ by computing a simulation on the $\mathcal{EL}$-description graph $\mathcal{G}_\mathcal{T}$. As mentioned earlier in this chapter, the greatest simulation on $\mathcal{G}_\mathcal{T}$ always exists and contains all the simulations on $\mathcal{G}_\mathcal{T}$. Consequently, in order to check Condition *(2)* of the above theorem, we may alternatively examine whether the greatest simulation $\hat{Z} : \mathcal{G}_\mathcal{T} \simeq \mathcal{G}_\mathcal{T}$ satisfies $(B, A) \in \hat{Z}$.

In [8], an efficient algorithm `EfficientSimilarity` for computing the greatest simulation of a graph is presented. Its time complexity is shown to be $\mathcal{O}(mn)$, where m is the number of edges and n is the number of nodes of the graph (assuming that $m \geq n$). The algorithm takes as an input a graph with unlabeled edges; in our framework, this corresponds to admitting only a single role name. Thus, this algorithm cannot be immediately applied to our $\mathcal{EL}$-description graphs, since the edges of $\mathcal{EL}$-description graphs are labeled with (potentially different) role names from N_{role}. To this end, we present a new algorithm—called $\mathcal{EL}^{gfp}$-`EfficientSimilarity`—which is a generalization of the `EfficientSimilarity` algorithm to take into account labeled edges. $\mathcal{EL}^{gfp}$-`EfficientSimilarity` is shown in Figure 4.1.

For each node v, the set $sim(v)$ contains those nodes that are candidates for simulating v. Initially, $sim(v)$ contains the nodes that satisfy Condition **S1** in Definition 23 and have an r-successor if v has for all $r \in \mathsf{N}_{role}$. We denote by $post(u, r)$ the set of all r-successors of u and similarly by $pre(u, r)$ the set of all r-predecessors of u. For a set U of nodes, we define $pre(U, r) := \bigcup_{u \in U} pre(u, r)$. We use a mapping $remove : V \times R \to 2^V$ as an auxiliary data structure. Throughout the WHILE loop, the edge condition of a simulation—i.e., Condition **S2** in Definition 23—is checked. The nodes in $remove(v, r)$ are to be removed from $sim(u)$

procedure $\mathcal{EL}^{gfp}$-`EfficientSimilarity`:

INPUT: an edge-labeled graph $\mathcal{G} = (V, E, L)$, where $E \subseteq (V \times R \times V)$ is a set of R-labeled edges.
OUTPUT: for each node $v \in V$, the simulator set $sim(v)$.

```
{ Initialization }
for all v ∈ V do
  sim(v) := { u ∈ V | L(v) ⊆ L(u) and
                      post(v,r) ≠ ∅ ⇒ post(u,r) ≠ ∅ for all r ∈ R };
  remove(v,r) := pre(V,r)\pre(sim(v),r); for all r ∈ R
  pre*(v) := { (u,r) | u ∈ pre(v,r) for all r ∈ R };
od;

{ Sharpening Loop }
while there is a node v ∈ V and an edge-label r ∈ R
      such that remove(v,r) ≠ ∅ do
  for all u ∈ pre(v,r) do
    for all w ∈ remove(v,r) do
      if w ∈ sim(u) then
        sim(u) := sim(u)\{w};
        for all (w',r') ∈ pre*(w) do
          if post(w',r') ∩ sim(u) = ∅ then
            remove(u,r') := remove(u,r') ∪ {w'};
          fi
        od
      fi
    od
  od;
  remove(v,r) := ∅;
od;
```

Figure 4.1: The algorithm for computing the greatest simulation of an $\mathcal{EL}$-description graph.

for $u \in pre(v, r)$. In this case, we say that $sim(u)$ is sharpened with respect to the edge $u \xrightarrow{r} v$. In addition to the data structures used in the algorithm `EfficientSimilarity`, we have an auxiliary mapping $pre^* : V \to 2^{V \times R}$, which maps each node v to the set of all tuples (u, r) if u is an r-predecessor of v.

With the set R singleton, i.e., the graph has no or a single edge-label, our algorithm is more or less identical to `EfficientSimilarity`. In fact, the mappings *pre*, *post* and *remove* with the lone edge-label boil down to the corresponding mappings in `EfficientSimilarity` without edge-label. The only difference is Condition **S1** of a simulation in Definition 23, where we use a label containment test in place of a label equivalence test. This is reflected the initialization of $sim(v)$ in the algorithm.

By applying the same arguments as in [8], it is not hard to prove that $\mathcal{EL}^{gfp}$-`EfficientSimilarity` can compute the simulator sets in time $\mathcal{O}(|V| \cdot |E|)$ assuming $|V| \leq |E|$. In case of loose graphs, i.e., $|V| \geq |E|$, the time complexity of our algorithm is bounded by $\mathcal{O}(|V|^2)$.

We are now ready to prove the following corollary.

Corollary 25. *Subsumption between concepts in the description logic $\mathcal{EL}$ w.r.t. a TBox $\mathcal{T}$ and greatest fixpoint semantics can be computed in cubic time in the size of $\mathcal{T}$, i.e., $\mathcal{O}(|\mathcal{T}|^3)$.*

Proof. Let $\mathcal{G}_\mathcal{T} = (V_\mathcal{T}, E_\mathcal{T}, L_\mathcal{T})$ be the corresponding $\mathcal{EL}$-description graph of $\mathcal{T}$. By Theorem 24, it suffices to show the complexity of $\mathcal{EL}^{gfp}$-`EfficientSimilarity` algorithm with respect to $\mathcal{G}_\mathcal{T}$. By Lemma 22, $|V_\mathcal{T}|$ is bounded by $|\mathcal{T}|$ whereas $|E_\mathcal{T}|$ is bounded by $|\mathcal{T}|^2$. If $|V| \geq |E|$, then the complexity of $\mathcal{EL}^{gfp}$-`Efficient-Similarity` is $\mathcal{O}(|V_\mathcal{T}| \cdot |E_\mathcal{T}|)$, i.e., $\mathcal{O}(|\mathcal{T}|^3)$. Otherwise, the complexity is bounded by $\mathcal{O}(|V|^2)$, i.e., $\mathcal{O}(|\mathcal{T}|^2)$. ❏

Chapter 5

A Quartic-time Algorithm for $\mathcal{EL}^{desc}$-Subsumption

Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox and $\mathcal{G}_\mathcal{T}$ the corresponding $\mathcal{EL}$-description graph. Since every gfp-model of $\mathcal{T}$ is also a model of $\mathcal{T}$, $A \sqsubseteq_\mathcal{T} B$ implies $A \sqsubseteq_{gfp,\mathcal{T}} B$. Consequently, $A \sqsubseteq_\mathcal{T} B$ implies that there is a simulation $Z : \mathcal{G}_\mathcal{T} \simeq \mathcal{G}_\mathcal{T}$ such that $(B, A) \in Z$. In order for the implication in the other direction to hold, an additional property on the simulation Z must be satisfied. It has been shown in [1] that this property is *synchronization*. In this chapter, we summarize the characterization of $\mathcal{EL}^{desc}$-Subsumption and present an essential corollary. Next, we introduce the notion of $\mathcal{EL}$-description formulae and develop a reduction from $\mathcal{EL}^{desc}$-subsumption to the satisfiability problem of Horn formulae (Horn-SAT). We then use a linear-time algorithm for Horn-SAT [9] to decide $\mathcal{EL}^{desc}$-subsumption. Ultimately, we also show that, for a given $\mathcal{EL}$-TBox $\mathcal{T}$, the whole algorithm for $\mathcal{EL}^{desc}$-subsumption takes quadratic time in the size of the corresponding $\mathcal{EL}$-description graph $\mathcal{G}_\mathcal{T}$. That is $\mathcal{O}(|\mathcal{T}|^4)$, since, as shown in Chapter 3, the size of $\mathcal{G}_\mathcal{T}$ is quadratic in the size of $\mathcal{T}$.

Similar to Chapter 4, we assume without loss of generality that $\mathcal{EL}$-TBoxes are normalized w.r.t. descriptive semantics. In the following, let $\mathcal{T}$ be a normalized $\mathcal{EL}$-TBox and $\mathcal{G}_\mathcal{T}$ the corresponding $\mathcal{EL}$-description graph of $\mathcal{T}$. Subsumption w.r.t. $\mathcal{T}$ and descriptive semantics can be characterized through synchronized simulation on $\mathcal{G}_\mathcal{T}$ similarly to Theorem 24, where *synchronized simulation* is considered instead of the normal *simulation*. The exact definition of *synchronized simulation* relation is omitted in this paper (please refer to [1] for this).

In order to decide the existence of a synchronized simulation that relates the defined concepts in question, an appropriate synchronized simulation $\mathcal{Y}_\mathcal{T} : \mathcal{G}_\mathcal{T} \simeq \mathcal{G}_\mathcal{T}$ has been introduced in [1] as follows:

Definition 26. Let $\mathcal{T}$ be a normalized $\mathcal{EL}$-TBox and $\mathcal{G}_\mathcal{T}$ the corresponding $\mathcal{EL}$-description graph. The *synchronized simulation* relation $\mathcal{Y}_\mathcal{T}$ is defined as $\bigcup_{n\geq 0} Y_n$,

where the relations Y_n are defined by induction on n: Y_0 is the identity on the nodes of $\mathcal{G}_\mathcal{T}$. If Y_{n-1} is already defined, then

$$Y_n := Y_{n-1} \cup \{\ (A,B) \mid \begin{array}{ll} (1) & L_\mathcal{T}(A) \subseteq L_\mathcal{T}(B), \\ (2) & (A, r_1, A_1), \ldots, (A, r_l, A_l) \text{ are all the edges} \\ & \text{in } \mathcal{G}_\mathcal{T} \text{ with source } A, \text{ and} \\ (3) & \text{there are edges } (B, r_1, B_1), \ldots, (B, r_l, B_l) \\ & \text{in } \mathcal{G}_\mathcal{T} \text{ such that } (A_1, B_1) \in Y_{n-1}, \ldots, \\ & (A_l, B_l) \in Y_{n-1}. \ \} \end{array}$$

$\Diamond$

The following corollary immediately follows from Theorem 29 and Proposition 36 in [1].

Theorem 27. ***[Baader]*** *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox, A, B be defined concepts in $\mathcal{T}$, $\mathcal{G}_\mathcal{T}$ the corresponding $\mathcal{EL}$-description graph of $\mathcal{T}$, and $\mathcal{Y}_\mathcal{T}$ the synchronized simulation relation for $\mathcal{T}$. Then, the following are equivalent:*

- $A \sqsubseteq_\mathcal{T} B$.
- $(B, A) \in \mathcal{Y}_\mathcal{T}$.

This provides us with a way to answer subsumption queries in $\mathcal{EL}$ w.r.t. descriptive semantics by constructing the corresponding synchronized simulation relation $\mathcal{Y}_\mathcal{T}$ and looking up in this relation the specific pair of defined concepts in question.

In the following we present a method to compute the relation $\mathcal{Y}_\mathcal{T}$ using a linear-time Horn-SAT algorithm. Before doing this, we must introduce some notation.

Definition 28 (Syntax of Horn formula). Let $\mathcal{P}$ be a set of *propositional letter*s. Then,

- a *literal* is either a propositional letter P from $\mathcal{P}$ (a *positive* literal) or the negation $\neg P$ of a propositional letter P from $\mathcal{P}$ (a *negative* literal);
- a *Horn clause* is a disjunction of literals, with at most one positive literal; and,
- a *Horn formula* is a conjunction (set) of Horn clauses.

A Horn clause with one positive literal $P \vee \neg P_1 \vee \ldots \vee \neg P_l$ where $l \geq 0$ and $\{P, P_1, \ldots, P_l\} \subseteq \mathcal{P}$ can be written as an implication $P \leftarrow P_1 \wedge \ldots \wedge P_l$. We call P the *head* of the clause and $P_1 \wedge \ldots \wedge P_l$ the *body* of the clause. Horn clauses with an empty body (no negative literals) are called *fact*s. $\Diamond$

It should be noted that the set of all Horn formulae is a subset of *Propositional Logic*. Therefore, the semantics of Horn formulae is defined based on that of Propositional Logic, i.e., *truth assignments*. A truth assignment $\mathcal{A}$ assigns to each propositional letter $P \in \mathcal{P}$ a truth value, either True or False. This is denoted by $P^{\mathcal{A}}$. $\mathcal{A}$ can inductively be extended to arbitrary propositional formulae in the obvious way. If $\mathcal{H}$ is a Horn formula, then an assignment $\mathcal{A}$ is a *model* of $\mathcal{H}$ if $\mathcal{H}^{\mathcal{A}} = \mathsf{True}$, written $\mathcal{A} \models \mathcal{H}$. If $\mathcal{A}$ is not a model of $\mathcal{H}$, we write $\mathcal{A} \not\models \mathcal{H}$. $\mathcal{H}$ is *satisfiable* if it has at least one model; otherwise, $\mathcal{H}$ is said to be *unsatisfiable*.

Definition 29 (Logical consequence). Let $\mathcal{H}$ be a Horn formula (i.e., a set of Horn clauses) and P a propositional letter. Then, P is *valid* in or a *(logical) consequence* of $\mathcal{H}$, denoted by $\mathcal{H} \models P$ if, and only if

$$\mathcal{A} \models \mathcal{H} \text{ implies } P^{\mathcal{A}} = \mathsf{True} \text{ for every model } \mathcal{A} \text{ of } \mathcal{H}.$$

We write $\mathcal{H} \not\models P$ if P is not a consequence of $\mathcal{H}$. $\Diamond$

Since Horn formulae enjoy the so-called *model intersection property* (see e.g., [19]):

$$\text{If } \mathcal{A}_1 \text{ and } \mathcal{A}_2 \text{ are models of } \mathcal{H}, \text{ then so is } \mathcal{A}_1 \cap \mathcal{A}_2,$$

it holds that for each Horn formula $\mathcal{H}$ there is a *least* model $\mathcal{M}_{\mathcal{H}}$, i.e., the intersection of all its models. Moreover, in order to check whether a propositional letter P logically follows from a Horn formulae $\mathcal{H}$ it suffices to consider only the least model $\mathcal{M}_{\mathcal{H}}$ of $\mathcal{H}$.

Proposition 30. *Let $\mathcal{H}$ be a Horn formula, $\mathcal{M}_{\mathcal{H}}$ the least model of $\mathcal{H}$, and P a propositional letter. Then, $\mathcal{H} \models P$ iff $\mathcal{M}_{\mathcal{H}} \models P$.*

The least model can be computed—inductively on the length of derivation—by means of a *meaning function* $f \colon \mathcal{P} \to \mathcal{P}$. Let $\mathcal{H}$ be a Horn formula. The meaning function $f_{\mathcal{H}}$ of $\mathcal{H}$ is defined inductively as follows:

$$\begin{array}{lcll} f_{\mathcal{H}} \uparrow 0 & := & f_{\mathcal{H}}(\emptyset) & = \{P \mid P \leftarrow \in \mathcal{H}\} \\ f_{\mathcal{H}} \uparrow (n+1) & := & f_{\mathcal{H}}(f_{\mathcal{H}} \uparrow n) & = f_{\mathcal{H}} \uparrow n \cup \{P \mid P \leftarrow P_1 \wedge \ldots \wedge P_l \in \mathcal{H} \\ & & & \qquad \text{and } \{P_1, \ldots, P_l\} \subseteq f_{\mathcal{H}} \uparrow n\} \end{array}$$

The least model $\mathcal{H}$ is defined as the least fixpoint of $f_{\mathcal{H}}$:

$$\mathcal{M}_{\mathcal{H}} := lfp(f_{\mathcal{H}}) = \bigcup_{n \geq 0} (f_{\mathcal{H}} \uparrow n).$$

First, we transform the relation $\mathcal{Y}_{\mathcal{T}}$ into a Horn formula. Then, we give a characterization of $\mathcal{EL}^{desc}$-subsumption through satisfiability of Horn formulae.

Definition 31 ($\mathcal{EL}$-description formulae). Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox, $\mathcal{G}_\mathcal{T} = (V_\mathcal{T}, E_\mathcal{T}, L_\mathcal{T})$ the corresponding $\mathcal{EL}$-graph, and $\mathcal{Y}_\mathcal{T}$ the corresponding synchronized simulation relation. The corresponding *$\mathcal{EL}$-description formula* of $\mathcal{T}$ denoted by $\mathcal{H}_\mathcal{T}$ is the smallest set of Horn clauses containing propositional letters of the following forms:

- $P_{A,B}$: if $A, B \in V_\mathcal{T}$, and
- $P_{(A,r,A'),B}$: if $A, B \in V_\mathcal{T}$ and $(A, r, A') \in E_\mathcal{T}$;

and containing the following Horn clauses:

(H1)	$P_{A,A} \longleftarrow$	for all nodes A in $V_\mathcal{T}$,
(H2)	$P_{(A,r,A'),B} \longleftarrow P_{A',B'}$	for all edges (A, r, A') and (B, r, B') in $E_\mathcal{T}$,
(H3)	$P_{A,B} \longleftarrow \bigwedge_{(A,r,A') \in E_\mathcal{T}} P_{(A,r,A'),B}$	for all nodes A, B in $V_\mathcal{T}$ with $L_\mathcal{T}(A) \subseteq L_\mathcal{T}(B)$

We call an H3 clause $H \longleftarrow \mathbf{B}$ (i.e., with the head H and the body $\mathbf{B}$) the *supporting clause* for H. ◊

Intuitively, the propositional letter $P_{A,B}$ encodes the fact that $(A, B) \in \mathcal{Y}_\mathcal{T}$, and the propositional letter $P_{(A,r,A'),B}$ says whether or not the pair (A, B) respects the condition (2) and (3) of $\mathcal{Y}_\mathcal{T}$ w.r.t. the edge (A, r, A') in $\mathcal{G}_\mathcal{T}$, i.e., there is an edge $(B, r, B') \in \mathcal{G}_\mathcal{T}$ for some $B' \in V_\mathcal{T}$ such that $(A, B) \in \mathcal{Y}_\mathcal{T}$.

While H1 encodes the identity relation on the nodes of $\mathcal{G}_\mathcal{T}$ (Y_0), H2 and H3 encode the construction of Y_n, provided that Y_{n-1} has already been computed. In fact, the existence of the following H3 Horn clause

$$P_{A,B} \longleftarrow \bigwedge_{(A,r,A') \in E_\mathcal{T}} P_{(A,r,A'),B}$$

in $\mathcal{H}_\mathcal{T}$ implies that Condition (1) of the construction of Y_n for some $n > 0$ is satisfied. Condition (2) and (3) of the same construction step are satisfied iff the body of this Horn clause logically follows from $\mathcal{H}_\mathcal{T}$ (i.e., $\mathcal{H}_\mathcal{T} \models P_{(A,r,A'),B}$ for all (A, r, A') in $E_\mathcal{T}$).

Before we state this characterization formally, let us consider an example illustrating the transformation of an $\mathcal{EL}$-description graph into its corresponding $\mathcal{EL}$-description formula.

Example 32. Let $\mathcal{G}_\mathcal{T} = (V_\mathcal{T}, E_\mathcal{T}, L_\mathcal{T})$ be an $\mathcal{EL}$-description graph for an $\mathcal{EL}$-TBox $\mathcal{T}$, which has at least the nodes and edges shown in Figure 5.1. Let $\mathcal{H}_\mathcal{T}$ be the corresponding $\mathcal{EL}$-description formula of $\mathcal{T}$. Assume that the labels of nodes

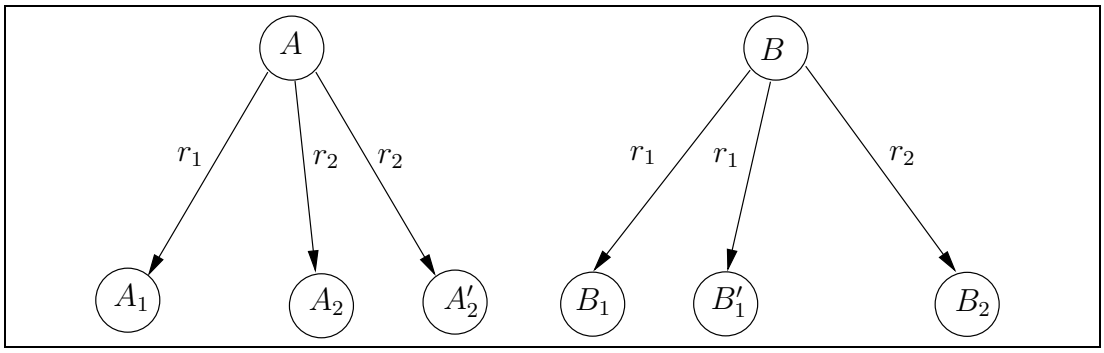

Figure 5.1: $\mathcal{EL}$-graph fragments for Example 32

are irrelevant, i.e., the labels of all nodes are identical. With respect to the nodes A and B, $\mathcal{H}_\mathcal{T}$ contains at least the following Horn clauses:

$$
\begin{aligned}
P_{A,A} &\longleftarrow \\
P_{B,B} &\longleftarrow \\
\\
P_{(A,r_1,A_1),B} &\longleftarrow P_{A_1,B_1} \\
P_{(A,r_1,A_1),B} &\longleftarrow P_{A_1,B'_1} \\
P_{(A,r_2,A_2),B} &\longleftarrow P_{A_2,B_2} \\
P_{(A,r_2,A'_2),B} &\longleftarrow P_{A'_2,B_2} \\
P_{(B,r_1,B_1),A} &\longleftarrow P_{B_1,A_1} \\
P_{(B,r_1,B'_1),A} &\longleftarrow P_{B'_1,A_1} \\
P_{(B,r_2,B_2),A} &\longleftarrow P_{B_2,A_2} \\
P_{(B,r_2,B'_2),A} &\longleftarrow P_{B_2,A'_2} \\
\\
P_{A,B} &\longleftarrow P_{(A,r_1,A_1),B} \wedge P_{(A,r_2,A_2),B} \wedge P_{(A,r_2,A'_2),B} \\
P_{B,A} &\longleftarrow P_{(B,r_1,B_1),A} \wedge P_{(B,r_1,B'_1),A} \wedge P_{(B,r_2,B_2),A}
\end{aligned}
$$

⊣

Theorem 33. *Let $\mathcal{T}$ be a normalized $\mathcal{EL}$-TBox, $\mathcal{Y}_\mathcal{T}$ the corresponding synchronized simulation relation of $\mathcal{T}$, and $\mathcal{H}_\mathcal{T}$ the corresponding $\mathcal{EL}$-description formula of $\mathcal{T}$. If A and B are defined concepts in $\mathcal{T}$, then the following are equivalent:*

1. $(A, B) \in \mathcal{Y}_\mathcal{T}$
2. $\mathcal{H}_\mathcal{T} \models P_{A,B}$.

Proof.

"(1)⇒(2)" $(A, B) \in \mathcal{Y}_\mathcal{T}$ implies that there exists an $n \geq 0$ such that $(A, B) \in Y_n$. Hence, it suffices to show that $(A, B) \in Y_n$ implies $\mathcal{H}_\mathcal{T} \models P_{A,B}$ for all $n \geq 0$. We prove this by induction on n.

$n = 0$ **:** Y_0 is the identity on the nodes of $\mathcal{G}_\mathcal{T}$; thus, $A = B$. H1 contains a fact $P_{A,B}$ implying that $\mathcal{H}_\mathcal{T} \models P_{A,B}$.

$n > 0$ **:** $(A, B) \in Y_n$ iff the pair (A, B) satisfies Conditions (1) to (3) in Definition 26. By definition of $\mathcal{H}_\mathcal{T}$, Condition (1) implies the existence of an H3 clause

$$P_{A,B} \longleftarrow P_{(A,r_1,A_1),B} \wedge \ldots \wedge P_{(A,r_l,A_l),B},$$

where $(A, r_1, A_1), \ldots, (A, r_l, A_l)$ are all out-going edges from A. Conditions (2) and (3) ensure for each out-going edge (A, r_i, A_i) from A that there exists an out-going edge (B, r_i, B_i) from B such that $(A_i, B_i) \in Y_{n-1}$. So, $\mathcal{H}_\mathcal{T}$ must contains an H2 clause $P_{(A,r_i,A_i),B} \longleftarrow P_{A_i,B_i}$. By I.H., it holds that $\mathcal{H}_\mathcal{T} \models P_{A_i,B_i}$ implying $\mathcal{H}_\mathcal{T} \models P_{(A,r_i,A_i),B}$. Since all conjuncts in the body of the above H3 clause are logical consequences of $\mathcal{H}_\mathcal{T}$, so is the head $P_{A,B}$. That is, $\mathcal{H}_\mathcal{T} \models P_{A,B}$.

"(2)⇒(1)" With Proposition 30 it suffices to prove that $\mathcal{M}_{\mathcal{H}_\mathcal{T}} \models P_{A,B}$ implies $(A, B) \in \mathcal{Y}_\mathcal{T}$. We prove this by induction on the meaning function of $\mathcal{M}_{\mathcal{H}_\mathcal{T}}$:

$P_{A,B} \in f_{\mathcal{H}_\mathcal{T}} \uparrow 0$ **:** Thus, there must be a fact $P_{A,B} \leftarrow$ in $\mathcal{H}_\mathcal{T}$, and this can only be if $A = B$. Then, by the definition of $\mathcal{Y}_\mathcal{T}$, $(A, B) \in Y_0 \subseteq \mathcal{Y}_\mathcal{T}$.

$P_{A,B} \in f_{\mathcal{H}_\mathcal{T}} \uparrow (n + 1)$ **:** Thus, there must be a supporting clause (H3) of the form

$$P_{A,B} \longleftarrow P_{(A,r_1,A_1),B} \wedge \ldots \wedge P_{(A,r_l,A_l),B} \qquad (*)$$

where $(A, r_1, A_1), \ldots, (A, r_l, A_l)$ are all out-going edges from A such that $\{P_{(A,r_1,A_1),B}, \ldots, P_{(A,r_l,A_l),B}\} \subseteq f_{\mathcal{H}_\mathcal{T}} \uparrow n$. Let us now concentrate on each propositional letter $P_{(A,r_i,A_i),B}$ with $1 \leq i \leq l$. Since $P_{(A,r_i,A_i),B}$ is in $f_{\mathcal{H}_\mathcal{T}} \uparrow n$, $\mathcal{H}_\mathcal{T}$ must contains an H2 clause of the form

$$P_{(A,r_i,A_i),B} \longleftarrow P_{A_i,B_i}$$

such that P_{A_i,B_i} is in $f_{\mathcal{H}_\mathcal{T}} \uparrow (n - 1)$, and thus also in $f_{\mathcal{H}_\mathcal{T}} \uparrow n$. Hence there must exist an out-going edge (B, r_i, B_i) from B. By I.H., $(A_i, B_i) \in \mathcal{Y}_\mathcal{T}$. The existence of $(*)$ implies that Condition (1) of Definition 26 is satisfied for the pair (A, B). Since each P_{A_i,B_i} corresponds to a part of Condition (2) and (3) of Definition 26 w.r.t. the edge (A, r_i, A_i) and there exists the corresponding edge (B, r_i, B_i) such that $(A_i, B_i) \in \mathcal{Y}_\mathcal{T}$, this concludes $(A, B) \in \mathcal{Y}_\mathcal{T}$.

❑

In logics with negation, the problem of logical consequence can be reduced to the satisfiability problem, i.e., $\mathcal{H} \models P$ is equivalent to that $\mathcal{H} \cup \neg P$ is unsatisfiable. Consequently, Theorem 27 together with Theorem 33 reduces the problem of $\mathcal{EL}^{desc}$-subsumption to the satisfiability problem of Horn formula.

We use this reduction to devise an algorithm for $\mathcal{EL}^{desc}$-subsumption with the help of a linear-time algorithm for Horn-SAT presented in [9]. For a Horn formula $\mathcal{H}$, we write $|\mathcal{H}|$ to denote its size, i.e., the number of all occurrences of propositional letters in $\mathcal{H}$. We prove that $\mathcal{EL}^{desc}$-subsumption can be decided in time quadratic in the size of the $\mathcal{EL}$-description graph, i.e., $\mathcal{O}(|\mathcal{G}_\mathcal{T}|^2)$. Due to the quadratic size of $\mathcal{G}_\mathcal{T}$ with respect to the size of $\mathcal{T}$, the algorithm will need the time $\mathcal{O}(|\mathcal{T}|^4)$. Since Horn-SAT is decidable in linear time, it suffices to show that $|\mathcal{H}_\mathcal{T}|$ is quadratic in $|\mathcal{G}_\mathcal{T}|$.

Lemma 34. *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox and $\mathcal{H}_\mathcal{T}$ the corresponding $\mathcal{EL}$-description formula of $\mathcal{T}$. Then $|\mathcal{H}|$ is bounded by $\mathcal{O}(|\mathcal{T}|^4)$.*

Proof. Let $\mathcal{G}_\mathcal{T} = (V_\mathcal{T}, E_\mathcal{T}, L_\mathcal{T})$ be the corresponding $\mathcal{EL}$-graph of $\mathcal{T}$. We have shown in Lemma 22 that the size of $\mathcal{G}_\mathcal{T}$ is quadratic in the size of $\mathcal{T}$. Thus, it suffices to show that $|\mathcal{H}_\mathcal{T}|$ is quadratic in $|\mathcal{G}_\mathcal{T}|$. We make a case distinction according to the the syntactic form of implications in $\mathcal{H}_\mathcal{T}$. In the following, we use $\#outedges_X$ to denote the number of out-going edges from node X.

H1 : There are $|V_\mathcal{T}|$ such implications, each with only a single occurrence of a propositional letter. Hence, the total size of H1 clauses $|\mathsf{H1}|$ is bounded by $|\mathcal{G}_\mathcal{T}|$.

H2 : Since each clause has exactly 2 occurrences of propositional letters, only the number of such clauses is relevant. By definition, the number of H2 clauses is bounded by $|E_\mathcal{T}|^2$, which is smaller than $|\mathcal{G}_\mathcal{T}|^2$.

H3 : For a fixed node B, there are at most $|V_\mathcal{T}|$ clauses of this form, one for each node A. The size of each such clause is $\#outedges_A + 1$. So, the total size of H3 clauses $|\mathsf{H3}|$ is $\sum_B \sum_A \#outedges_A + 1$, which is $|V_\mathcal{T}| \cdot (|E_\mathcal{T}| + |V_\mathcal{T}|)$. This is bounded by $|\mathcal{G}_\mathcal{T}|^2$.

Since $\mathcal{H}_\mathcal{T}$ contains only Horn clauses of these forms, its size is bounded by $\mathcal{O}(|\mathcal{G}_\mathcal{T}|^2)$, i.e., quadratic in the size of $\mathcal{G}_\mathcal{T}$. ❏

Corollary 35. *Subsumption between concepts in the description logic $\mathcal{EL}$ w.r.t. a TBox $\mathcal{T}$ and descriptive semantics can be computed in quartic time in the size of $\mathcal{T}$, i.e., $\mathcal{O}(|\mathcal{T}|^4)$.*

Chapter 6

A Cubic-time Algorithm for $\mathcal{EL}^{gci}$-Subsumption

In the previous two chapters, we have seen characterizations of and algorithms for subsumption w.r.t. $\mathcal{EL}$-TBoxes. The aim of this chapter is to present a characterization of and an efficient algorithm for subsumption w.r.t. *general* $\mathcal{EL}$-TBoxes (i.e., sets of general concept inclusions: GCIs). It has been shown in [4] that $\mathcal{EL}^{gci}$-subsumption can be decided in polynomial time. A natural question is what is the exact degree of the polynomial.

As mentioned in Chapter 2, both $\mathcal{EL}^{gci}$ and $\mathcal{EL}^{desc}$ use descriptive semantics, but $\mathcal{EL}^{gci}$ is more expressive than $\mathcal{EL}^{desc}$. In other words, an $\mathcal{EL}^{desc}$-TBox can be translated to an $\mathcal{EL}^{gci}$-TBox, but not vice versa in general. This means that we can compute $\mathcal{EL}^{desc}$-subsumption using the algorithm presented in this chapter instead of that in Chapter 5. The fact that we can decide $\mathcal{EL}^{gci}$-subsumption—and thus also $\mathcal{EL}^{desc}$-subsumption—in time cubic in the size of TBoxes shows that the algorithm in Chapter 5 is not optimal. The main reason is due to the normalization in Section 3.1, which potentially leads to quadratic blowup already in normalization phase.

Since we explicitly allow for GCIs in $\mathcal{EL}^{gci}$, as discussed in Section 2.2 the boundary between primitive and defined concept names vanishes. As a result, we can neither use the old notion of normalized $\mathcal{EL}$-TBoxes, nor can we characterize $\mathcal{EL}^{gci}$-subsumption through simulation on $\mathcal{EL}$-description graphs.[1] For this reason together with the quadratic blowup of normalization in Section 3.1, we will present a fresh notion of normalization and a characterization of $\mathcal{EL}^{gci}$-subsumption. Similar to Chapter 5, the algorithm for $\mathcal{EL}^{gci}$-subsumption applies the technique of translating general $\mathcal{EL}$-TBoxes to Horn formulae. Then, the linear-time algorithm for Horn-SAT [9] is exploited.

[1] $\mathcal{EL}$-description graphs have *defined concept*s as nodes and *primitive concept*s as nodes' labels.

We extend the notion $|\mathcal{T}|$ to general $\mathcal{EL}$-TBoxes with the same meaning, i.e., the total number of occurrences of role and concept names in $\mathcal{T}$. The presented algorithm needs time cubic in the size of the general $\mathcal{EL}$-TBox $\mathcal{T}$, i.e., $\mathcal{O}(|\mathcal{T}|^3)$.

6.1 $\mathcal{EL}^{gci}$ Normalization

Definition 36 (GCI-normalized $\mathcal{EL}$-TBox). Let $\mathcal{T}$ be a general $\mathcal{EL}$-TBox over N_{con} and N_{role}. $\mathcal{T}$ is *GCI-normalized* (or in *GCI-normal form*) iff $\mathcal{T}$ contains only GCIs of the following forms:

GCI1	A	$\sqsubseteq$	B
GCI2	$A_1 \sqcap A_2$	$\sqsubseteq$	B
GCI3	A	$\sqsubseteq$	$\exists r.B$
GCI4	$\exists r.A$	$\sqsubseteq$	B

where A, A_1, A_2, and B are concept names from N_{con} or the top concept $\top$, and r is a role name from N_{role}. ◊

A general $\mathcal{EL}$-TBox can be transformed into GCI-normal form by exhaustively applying the following normalization rules.

Definition 37 (GCI-normalization rules). Let $\mathcal{T}$ be a general $\mathcal{EL}$-TBox over concept names N_{con} and role names N_{role}. The *GCI-normalization rules* are defined as follows:

NF1	$C \doteq D$	$\longrightarrow$	$\{\ C \sqsubseteq D, D \sqsubseteq C\ \}$
NF2	$\hat{C} \sqcap D \sqsubseteq E$	$\longrightarrow$	$\{\ \hat{C} \sqsubseteq A, A \sqcap D \sqsubseteq E\ \}$
NF3	$C \sqcap \hat{D} \sqsubseteq E$	$\longrightarrow$	$\{\ \hat{D} \sqsubseteq A, C \sqcap A \sqsubseteq E\ \}$
NF4	$\exists r.\hat{C} \sqsubseteq D$	$\longrightarrow$	$\{\ \hat{C} \sqsubseteq A, \exists r.A \sqsubseteq D\ \}$
NF5	$\hat{C} \sqsubseteq \hat{D}$	$\longrightarrow$	$\{\ \hat{C} \sqsubseteq A, A \sqsubseteq \hat{D}\ \}$
NF6	$B \sqsubseteq \exists r.\hat{C}$	$\longrightarrow$	$\{\ B \sqsubseteq \exists r.A, A \sqsubseteq \hat{C}\ \}$
NF7	$B \sqsubseteq C \sqcap D$	$\longrightarrow$	$\{\ B \sqsubseteq C, B \sqsubseteq D\ \}$

where r denotes a role name, B a concept name, and A a *new* concept name. Additionally, let $\hat{C}, \hat{D}$ denote non-atomic concept descriptions (i.e., complex) and C, D, E any concept descriptions (possibly complex).

Applying a rule $G \longrightarrow \mathcal{S}$ to $\mathcal{T}$ changes $\mathcal{T}$ to $(\mathcal{T} \backslash \{G\}) \cup \mathcal{S}$. The *GCI-normalized TBox*, denoted by $norm_{gci}(\mathcal{T})$, is defined by exhaustively applying Rules **NF1** to **NF4** (Phase 1); and after that, exhaustively applying Rules **NF5** to **NF7** (Phase 2). ◊

Lemma 38. *Let $\mathcal{T}$ be a general TBox. The GCI-normalized TBox $norm_{gci}(\mathcal{T})$ of $\mathcal{T}$ can be computed in linear time in the size of $\mathcal{T}$. The resulting ontology $norm_{gci}(\mathcal{T})$ is of linear size in the size of $\mathcal{T}$.*

Proof. The size of $\mathcal{T}$ is increased only linearly by exhaustive application of Rule **NF1**. Since this rule will never become applicable as a consequence of the remaining Rules, we may restrict our attention to Rules **NF2** to **NF7**. Rules **NF2** and **NF3** are each applicable at most once for each occurrence of "$\sqcap$" on the left-hand side of a GCI in $\mathcal{T}$. Similarly, the number of application of Rule **NF4** is bounded by the occurrences of "$\exists$" on the left-hand side of a GCI in $\mathcal{T}$. A single application of one of the rules in Phase 1—**NF2** to **NF4**—increases the size of $\mathcal{T}$ only by a constant, introducing a new concept name and splitting one GCI to two. Therefore, exhaustive application of the rules in Phase 1 takes linear time and produces a TBox $\mathcal{T}'$ of size linear in the size of $\mathcal{T}$.

Rule **NF5** is applicable at most once for each GCI in $\mathcal{T}'$ and results in two split GCIs of linear size. Analogous to Phase 1, **NF6** (**NF7** respectively) is applicable once for each occurrence of "$\exists$" ("$\sqcap$" respectively) in $\mathcal{T}'$. A single application of Rule **NF6** increases the size of $\mathcal{T}'$ only by a constant. This holds also for Rule **NF7**, since the left-hand side of the GCIs in this rule is a concept name (i.e., of constant size). Thus, exhaustive application of the rules in Phase 2—**NF5** to **NF7**—yields a TBox of size linear in the size of $\mathcal{T}$. ❑

It is crucial that we divide GCI-normalization rules into two phases and exhaustively apply rules in Phase 1 first. If we allow the rules to be applied arbitrarily, the size of the resulting GCI-normalized TBox may blow up quadratically in the original TBox's size. The following example illustrates this situation.

Example 39. Let us consider a general $\mathcal{EL}$-TBox $\mathcal{T}$ with the sole GCI:

$$A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B_1 \sqcap \cdots \sqcap B_n \quad (*)$$

where A_i, B_i are concept names for an $n \geq 3$ and $1 \leq i \leq n$. Note that $\mathcal{T}$ is not yet normalized w.r.t. Phase 1 because the left-hand side of $(*)$ is an n-ary conjunction with $n \geq 3$. In other words, the rules in Phase 1, e.g., Rule **NF2** or **NF3**, is applicable.

By exhaustive application of Rule **NF7** to $(*)$, we obtain a general TBox $\mathcal{T}'$ as follows:

$$\begin{aligned} A_1 \sqcap \cdots \sqcap A_n &\sqsubseteq B_1 \\ A_1 \sqcap \cdots \sqcap A_n &\sqsubseteq B_2 \\ &\vdots \\ A_1 \sqcap \cdots \sqcap A_n &\sqsubseteq B_n \end{aligned}$$

Though $\mathcal{T}'$ is not yet in GCI-normal form, its size is already quadratic in n, i.e., $\mathcal{O}(n^2)$. The reason is that Rule **NF7** replicates for each application the left-hand

side of the GCI which is $A_1 \sqcap \cdots \sqcap A_n$ in this case. This quadratic blowup will not happen if we exhaustively apply the rules in Phase 1 before. ⊣

Now, let us consider Example 12 back on Page 13. We will illustrate the GCI-normalization process step by step by applying the GCI-normalization rules to the medical ontology.

Example 40. Let $\mathcal{T}$ be the general TBox presented in Example 12. $\mathcal{T}$ comprises 4 GCIs and no concept definitions. Therefore, normalization Rule **NF1** is never applied. The first three GCIs are already normalized with respect to Phase 1. Rule **NF3** applies once to the last GCI

$$\mathsf{Disease} \sqcap \exists \mathsf{has\_loc}.\exists \mathsf{cont\_in}.\mathsf{Heart} \sqsubseteq \mathsf{Heartdisease} \sqcap \exists \mathsf{is\_state}.\mathsf{NeedsTreatment}$$

splitting it up into two GCIs:

$$\left\{ \begin{array}{r} \mathsf{Disease} \sqcap \mathsf{A}_1 \sqsubseteq \mathsf{Heartdisease} \sqcap \exists \mathsf{is\_state}.\mathsf{NeedsTreatment} \\ \exists \mathsf{has\_loc}.\exists \mathsf{cont\_in}.\mathsf{Heart} \sqsubseteq \mathsf{A}_1 \end{array} \right\}$$

The first of which is then normalized with respect to Phase 1, while the second is not. Applying Rule **NF4** to it results again in two GCIs. Let $\mathcal{T}'$ denote the normalized TBox with respect to Phase 1; $\mathcal{T}'$ has 6 GCIs as follows:

$$\begin{array}{rl} \mathsf{Pericardium} & \sqsubseteq \mathsf{Tissue} \sqcap \exists \mathsf{cont\_in}.\mathsf{Heart} \\ \mathsf{Pericarditis} & \sqsubseteq \mathsf{Inflammation} \sqcap \exists \mathsf{has\_loc}.\mathsf{Pericardium} \\ \mathsf{Inflammation} & \sqsubseteq \mathsf{Disease} \sqcap \exists \mathsf{acts\_on}.\mathsf{Tissue} \\ \mathsf{Disease} \sqcap \mathsf{A}_1 & \sqsubseteq \mathsf{Heartdisease} \sqcap \exists \mathsf{is\_state}.\mathsf{NeedsTreatment} \\ \exists \mathsf{has\_loc}.\mathsf{A}_2 & \sqsubseteq \mathsf{A}_1 \\ \exists \mathsf{cont\_in}.\mathsf{Heart} & \sqsubseteq \mathsf{A}_2 \end{array}$$

Now, Rule **NF5** is applicable only to the fourth GCI changing it to

$$\left\{ \begin{array}{rl} \mathsf{Disease} \sqcap \mathsf{A}_1 & \sqsubseteq \mathsf{A}_3 \\ \mathsf{A}_3 & \sqsubseteq \mathsf{Heartdisease} \sqcap \exists \mathsf{is\_state}.\mathsf{NeedsTreatment} \end{array} \right\}$$

The first GCI of the above set, as well as the last two GCIs of $\mathcal{T}'$, is in GCI-normal form. The rest will also be GCI-normalized after an application of Rule **NF7** once for each GCI. The final general TBox $\mathcal{T}''$ in GCI-normal form is composed of 11

GCIs as follows:

$$\begin{aligned}
\mathsf{Pericardium} &\sqsubseteq \mathsf{Tissue}\\
\mathsf{Pericardium} &\sqsubseteq \exists \mathsf{cont_in.Heart}\\
\mathsf{Pericarditis} &\sqsubseteq \mathsf{Inflammation}\\
\mathsf{Pericarditis} &\sqsubseteq \exists \mathsf{has_loc.Pericardium}\\
\mathsf{Inflammation} &\sqsubseteq \mathsf{Disease}\\
\mathsf{Inflammation} &\sqsubseteq \exists \mathsf{acts_on.Tissue}\\
\mathsf{Disease} \sqcap \mathsf{A}_1 &\sqsubseteq \mathsf{A}_3\\
\mathsf{A}_3 &\sqsubseteq \mathsf{Heartdisease}\\
\mathsf{A}_3 &\sqsubseteq \exists \mathsf{is_state.NeedsTreatment}\\
\exists \mathsf{has_loc.A}_2 &\sqsubseteq \mathsf{A}_1\\
\exists \mathsf{cont_in.Heart} &\sqsubseteq \mathsf{A}_2
\end{aligned}$$

⊣

It is not hard to see that all GCI-normalization rules preserve concept subsumption. Thus, a general $\mathcal{EL}$-TBox and its GCI-normal form are equivalent with respect to concept subsumption.

Proposition 41. *Let $\mathcal{T}$ be a general $\mathcal{EL}$-TBox over concept names N_{con} and role names N_{role}. If $A, B \in \mathsf{N}_{def}$, then $A \sqsubseteq_{gci,\mathcal{T}} B$ iff $A \sqsubseteq_{gci,norm_{gci}(\mathcal{T})} B$.*

6.2 Implication Sets

In the following, we assume without loss of generality that general $\mathcal{EL}$-TBoxes are GCI-normalized. For the rest of this section, we fix a GCI-normalized $\mathcal{EL}$-TBox $\mathcal{T}$ over concept names N_{con} and role names N_{role}. To simplify notation, let $\mathsf{N}_{con}^{\top} := \mathsf{N}_{con} \cup \{\top\}$.

Our strategy is to compute for every concept $A \in \mathsf{N}_{con}^{\top}$ a set of concepts $\mathcal{S}_{\mathcal{T}}(A) \subseteq \mathsf{N}_{con}^{\top}$ with the following property: for all models $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of $\mathcal{T}$ and all individuals $x \in \Delta^{\mathcal{I}}$, if the concept A holds at x in $\mathcal{I}$ (i.e., $x \in A^{\mathcal{I}}$) then every concept in $\mathcal{S}_{\mathcal{T}}(A)$ also holds at x in $\mathcal{I}$. With the simple structure of GCIs in GCI-normalized TBoxes, we define such set as follows.

Definition 42 (Implication set). For every concept name A in $\mathsf{N}_{con}^{\top}$, the *implication set* $\mathcal{S}_{\mathcal{T}}(A)$ is defined by $\bigcup_{n \geq 0} S_n(A)$ where the sets S_n are defined inductively on n: $S_0(A) := \{A, \top\}$. If $S_n(B)$ is already defined for all concept names $B \in \mathsf{N}_{con}^{\top}$, then $S_{n+1}(A)$ is the result of *exhaustive* application of the extension rules in Figure 6.1. ◊

The set $S_n(A)$ is said to be *complete* if no extension rules are applicable. We now present the result showing that implication sets characterize $\mathcal{EL}^{gci}$-subsumption. For the full proof, please refer to [5].

IS1	If $A \in S_n(B)$, $A \sqsubseteq C \in \mathcal{T}$, and $C \notin S_n(B)$ then $S_{n+1}(B) := S_n(B) \cup \{C\}$
IS2	If $\{A_1, A_2\} \subseteq S_n(B)$, $A_1 \sqcap A_2 \sqsubseteq C \in \mathcal{T}$, and $C \notin S_n(B)$ then $S_{n+1}(B) := S_n(B) \cup \{C\}$
IS3	If $A_1 \in S_n(B)$, $A_1 \sqsubseteq \exists r.A_2 \in \mathcal{T}$, $A_3 \in S_n(A_2)$, $\exists r.A_3 \sqsubseteq C \in \mathcal{T}$, and $C \notin S_n(B)$ then $S_{n+1}(B) := S_n(B) \cup \{C\}$

Figure 6.1: Extension Rules for Implication Sets

Theorem 43. *Let $\mathcal{T}$ be a GCI-normalized $\mathcal{EL}$-TBox over concept names N_{con} and role names N_{role}, and A, B concept in $\mathsf{N}_{con}^{\top}$. Then, $A \in \mathcal{S}_{\mathcal{T}}(B)$ if and only if $B \sqsubseteq_{\mathcal{T}} A$.*

6.3 $\mathcal{EL}^{gci}$-Description Formulae

So far, we have a characterization of $\mathcal{EL}^{gci}$-subsumption through the notion of implication sets. In this section, we will show how the extension rules depicted in Figure 6.1 can be encoded using Horn formulae.

Definition 44 ($\mathcal{EL}^{gci}$-description formulae). Let $\mathcal{T}$ be a GCI-normalized $\mathcal{EL}$-TBox over concept names N_{con} and role names N_{role}. The corresponding $\mathcal{EL}^{gci}$-*description formula* of $\mathcal{T}$, denoted by $\mathcal{H}_{\mathcal{T}}$, is the smallest set of Horn clauses containing only propositional letters of the form

$$P_{\alpha,\beta}$$

where $\{\alpha, \beta\} \subseteq \mathsf{N}_{con}^{\top}$ and comprising the following Horn clauses:

(H0)	$P_{C,C}$	$\longleftarrow$		for all $C \in \mathsf{N}_{con}$
	$P_{\top,C}$	$\longleftarrow$		
(H1)	$P_{B,C}$	$\longleftarrow$	$P_{A,C}$	for all $C \in \mathsf{N}_{con}$ and for each GCI $A \sqsubseteq B \in \mathcal{T}$
(H2)	$P_{B,C}$	$\longleftarrow$	$P_{A_1,C} \wedge P_{A_2,C}$	for all $C \in \mathsf{N}_{con}$ and for each GCI $A_1 \sqcap A_2 \sqsubseteq B \in \mathcal{T}$
(H3)	$P_{B,C}$	$\longleftarrow$	$P_{A,C} \wedge P_{B_2,B_1}$	for all $C \in \mathsf{N}_{con}$ and for GCIs $\{A \sqsubseteq \exists r.B_1, \exists r.B_2 \sqsubseteq B\} \subseteq \mathcal{T}$.

We call a clause $H \longleftarrow \mathbf{B}$ (i.e., with head H and body $\mathbf{B}$) the *supporting clause* for H. $\Diamond$

Intuitively, the propositional letter $P_{A,B}$ encodes the fact that $A \in \mathcal{S}_\mathcal{T}(B)$, which in turns means that the concept B is subsumed by the concept A. The facts $\mathsf{H0}$

$$P_{C,C} \longleftarrow \quad \text{and} \quad P_{\top,C} \longleftarrow$$

encodes the initialization of the implication sets of the concept name C, i.e., $S_0(C)$, asserting that C is subsumed by the top-concept and the concept name C itself.

Horn clauses $\mathsf{H1}$, $\mathsf{H2}$ and $\mathsf{H3}$ correspond to the extension rules **IS1**, **IS2** and **IS3** respectively. The existence of a supporting clause $H \longleftarrow \mathbf{B}$ together with the validity of $\mathbf{B}$ ($\mathcal{H}_\mathcal{T} \models \mathbf{B}$) implies the applicability of the corresponding rule. Additionally, as the correspondence of post-condition of the rule application, the head H becomes valid in the $\mathcal{EL}^{gci}$-description formula, i.e., $\mathcal{H}_\mathcal{T} \models H$.

The following theorem formally states this characterization.

Theorem 45. *Let $\mathcal{T}$ be a GCI-normalized $\mathcal{EL}$-TBox over N_{con} and N_{role} and $\mathcal{H}_\mathcal{T}$ the corresponding $\mathcal{EL}^{gci}$-description formula of $\mathcal{T}$. If A and B are concept names in $\mathsf{N}_{con}^\top$, then the following are equivalent:*

1. $A \in \mathcal{S}_\mathcal{T}(B)$.
2. $\mathcal{H}_\mathcal{T} \models P_{A,B}$.

Proof.

"(1)⇒(2)" $A \in \mathcal{S}_\mathcal{T}(B)$ implies that there exists an $n \geq 0$ such that $A \in S_n$. Hence, it suffices to show that $A \in S_n(B)$ implies $\mathcal{H}_\mathcal{T} \models P_{A,B}$ for all $n \geq 0$. We prove this by induction on the minimal n with $A \in S_n(A)$.

- $n = 0$ **:** Since $S_0(B) := \{\top, B\}$, A is either $\top$ or B. It holds trivially by $\mathsf{H0}$ that $\mathcal{H}_\mathcal{T} \models P_{A,B}$.
- $n > 0$ **:** Due to the minimality condition on n, $A \in S_n(B)$ but $A \notin S_{n-1}(B)$. So, there must be an extension rule which is applicable to the implication set $S_{n-1}(B)$. As we have 3 such extension rules, we analyse them one by one as follows:
  - Applicability of Rule **IS1** implies that there exists $A' \in \mathsf{N}_{con}^\top$ such that $A' \in S_{n-1}(B)$ and $A' \sqsubseteq A \in \mathcal{T}$. By I.H., $\mathcal{H}_\mathcal{T} \models P_{A',B}$. Since $A' \sqsubseteq A \in \mathcal{T}$ and $B \in \mathsf{N}_{con}$, the definition of $\mathcal{H}_\mathcal{T}$ implies there is an $\mathsf{H1}$ clause $P_{A,B} \longleftarrow P_{A',B}$. These altogether yield $\mathcal{H}_\mathcal{T} \models P_{A,B}$.

- Applicability of Rule **IS2** implies that there exist A_1 and A_2 in $\mathsf{N}_{con}^{\top}$ such that $\{A_1, A_2\} \subseteq S_{n-1}(B)$ and $A_1 \sqcap A_2 \sqsubseteq A \in \mathcal{T}$. By I.H., $\mathcal{H}_{\mathcal{T}} \models P_{A_1,B}$ and $\mathcal{H}_{\mathcal{T}} \models P_{A_2,B}$, i.e., $\mathcal{H}_{\mathcal{T}} \models P_{A_1,B} \wedge P_{A_2,B}$. Since $A_1 \sqcap A_2 \sqsubseteq A \in \mathcal{T}$ and $B \in \mathsf{N}_{con}$, the definition of $\mathcal{H}_{\mathcal{T}}$ implies there is an $\mathsf{H2}$ clause $P_{A,B} \longleftarrow P_{A_1,B} \wedge P_{A_2,B}$. These altogether yield $\mathcal{H}_{\mathcal{T}} \models P_{A,B}$.
- Applicability of Rule **IS3** implies that there exist A_1, A_2, and A_3 in $\mathsf{N}_{con}^{\top}$ and $r \in \mathsf{N}_{role}$ such that $A_1 \in S_{n-1}(B)$ and $A_1 \sqsubseteq \exists r.A_2 \in \mathcal{T}$ and $A_3 \in S_{n-1}(A_2)$ and $\exists r.A_3 \sqsubseteq A \in \mathcal{T}$. By I.H., $\mathcal{H}_{\mathcal{T}} \models P_{A_1,B}$ and $\mathcal{H}_{\mathcal{T}} \models P_{A_3,A_2}$, i.e., $\mathcal{H}_{\mathcal{T}} \models P_{A_1,B} \wedge P_{A_3,A_2}$. Since $\{A_1 \sqsubseteq \exists r.A_2, \exists r.A_3 \sqsubseteq A\} \subseteq \mathcal{T}$ and $B \in \mathsf{N}_{con}$, the definition of $\mathcal{H}_{\mathcal{T}}$ implies there is an $\mathsf{H3}$ clause $P_{A,B} \longleftarrow P_{A_1,B} \wedge P_{A_3,A_2}$. These altogether yield $\mathcal{H}_{\mathcal{T}} \models P_{A,B}$.

"(2)⇒(1)" Let $\mathcal{H}_{\mathcal{T}} \models P_{A,B}$ and $\mathcal{M}_{\mathcal{H}_{\mathcal{T}}}$ be the least model of $\mathcal{H}_{\mathcal{T}}$. With the help of Proposition 30, it suffices to prove that $\mathcal{M}_{\mathcal{H}_{\mathcal{T}}} \models P_{A,B}$ implies $A \in \mathcal{S}_{\mathcal{T}}(B)$. We prove this by induction on the meaning function $f_{\mathcal{H}_{\mathcal{T}}}$ of $\mathcal{H}_{\mathcal{T}}$:

$P_{A,B} \in f_{\mathcal{H}_{\mathcal{T}}} \uparrow 0$ **:** This is the case only when $A = \top$ or $A = B$. Thus, this is trivial by the definition of $\mathcal{S}_{\mathcal{T}}$. Indeed, $\{\top, B\} = S_0(B) \subseteq \mathcal{S}_{\mathcal{T}}(B)$.

$P_{A,B} \in f_{\mathcal{H}_{\mathcal{T}}} \uparrow (n+1)$ **:** There must be a supporting clause $\mathsf{H} : H \longleftarrow \mathbf{B}$ such that $H := P_{A,B}$ and every conjunct in $\mathbf{B}$ holds in $f_{\mathcal{H}_{\mathcal{T}}} \uparrow n$ (implying $\mathcal{H}_{\mathcal{T}} \models \mathbf{B}$). According to the definition of $\mathcal{EL}^{gci}$-description formula, such a supporting clause falls into either $\mathsf{H1}$, $\mathsf{H2}$, or $\mathsf{H3}$. Together with the induction hypothesis, the existence of H and that $\mathcal{H}_{\mathcal{T}} \models \mathbf{B}$ implies the applicability condition of the corresponding extension rule. Hence, we can conclude that $A \in \mathcal{S}_{\mathcal{T}}(B)$.

❑

As we already mentioned, in Horn logic the consequence problem can be reduced to the satisfiability problem. Precisely, in order to check if $\mathcal{H} \models P$, we can instead verify that $\mathcal{H} \cup \neg P$ is unsatisfiable. Consequently, Theorem 45 together with Theorem 43 reduces the problem of $\mathcal{EL}^{gci}$-subsumption to the satisfiability problem of Horn formula.

Since the satisfiability problem of Horn formulae is decidable in time linear in the size of the Horn formula [9], the time complexity of $\mathcal{EL}^{gci}$-subsumption depends only on the size of the $\mathcal{EL}^{gci}$-description formula. In the following, we fix a general $\mathcal{EL}$-TBox $\mathcal{T}$. We prove that the size of the corresponding $\mathcal{EL}^{gci}$-description formulae $\mathcal{H}_{\mathcal{T}}$, denoted by $|\mathcal{H}_{\mathcal{T}}|$, is cubic in the size of $\mathcal{T}$. This result immediately implies that $\mathcal{EL}^{gci}$-subsumption is decidable in time cubic in the size $\mathcal{T}$, i.e., $\mathcal{O}(|\mathcal{T}|^3)$.

Lemma 46. *Let $\mathcal{T}$ be a GCI-normalized $\mathcal{EL}$-TBox over N_{con} and N_{role} and $\mathcal{H}_\mathcal{T}$ the corresponding $\mathcal{EL}^{gci}$-description formula of $\mathcal{T}$. The size of $\mathcal{H}_\mathcal{T}$ is cubic in the size of $\mathcal{T}$.*

Proof. Since $\mathcal{T}$ is in GCI-normal form, it comprises only GCIs of the forms presented in Definition 36. To simplify the proof, let

$$\begin{aligned}
\#c &:= \text{number of concept names in } \mathcal{T}\text{, i.e., } |\mathsf{N}_{con}|,\\
\#r &:= \text{number of role names in } \mathcal{T}\text{, i.e., } |\mathsf{N}_{role}|,\\
\#n_1 &:= \text{number of } \textsf{GCI1}\text{s in } \mathcal{T},\\
\#n_2 &:= \text{number of } \textsf{GCI2}\text{s in } \mathcal{T},\\
\#n_3 &:= \text{number of } \textsf{GCI3}\text{s in } \mathcal{T}\text{, and}\\
\#n_4 &:= \text{number of } \textsf{GCI4}\text{s in } \mathcal{T}.
\end{aligned}$$

It is easy to see that the size of $\mathcal{T}$ is linear in all these numbers (accurately, $|\mathcal{T}| = 2 \cdot \#n_1 + 3 \cdot \#n_2 + 3 \cdot \#n_3 + 3 \cdot \#n_4$). The size of the corresponding $\mathcal{EL}^{gci}$-description formula $|\mathcal{H}_\mathcal{T}|$ is $2 \cdot \#c + 2 \cdot \#n_1 \cdot \#c + 3 \cdot \#n_2 \cdot \#c + 3 \cdot \#n_3 \cdot \#n_4 \cdot \#c$, which is obviously bounded by $\mathcal{O}(|\mathcal{T}|^3)$. ❑

Corollary 47. *Subsumption between concepts in the description logic $\mathcal{EL}$ w.r.t. a general TBox $\mathcal{T}$ can be computed in time cubic in the size of $\mathcal{T}$, i.e., $\mathcal{O}(|\mathcal{T}|^3)$.*

Chapter 7

Experiments of $\mathcal{EL}$-Subsumptions on the Gene Ontology

In the preceding three chapters, we have formulated characterizations of and algorithms for subsumption in three small description logics based on $\mathcal{EL}$. The time complexity of these algorithms has been theoretically investigated. It is therefore interesting to know—and, of course, natural to explore—the behaviors of these algorithms in practice. In this chapter, we will present the results of experiments on the Gene Ontology.

7.1 The Gene Ontology

An *ontology* is a domain-specific vocabulary—usually used in a particular field such as biology and chemistry. The terms in an ontology are defined in a controlled manner and are linked to each other. *The Gene Ontology* (or GO, for short) is "controlled ontologies describe gene products in terms of their associated biological processes, cellular components and molecular functions in a species-independent manner" (definition from Gene Ontology™ Consortium [10]).

For example, let us consider the following GO concept with its concept definition:[1]

[1]Please note that the term 'concept definition' here is not the same as that in a DL sense. We write '$\mathcal{EL}$-concept definition' for the definition, i.e., $A \equiv D$ in the DL $\mathcal{EL}$.

```
[Term]
id:           GO:0000019
name:         regulation of mitotic recombination
namespace:    process
def:          ‘‘Any process that modulates the frequency,
              rate or extent of DNA recombination during
              mitosis.’’  [GO:curators]
is_a:         GO:0000018
relationship: part_of GO:0006312
```

This is the definition of GO concept GO:0000019, described in words through the def keyword and indicating relationships with other GO concepts through the keywords is_a and relationship. This GO concept definition can straightforwardly be translated into a general concept inclusion

$$\mathsf{GO\_0000019} \sqsubseteq \mathsf{GO\_0000018} \sqcap \exists \mathsf{part\_of}.\mathsf{GO\_0006312}$$

in the description logic $\mathcal{EL}$. The reason that we do not translate it into an $\mathcal{EL}$-concept definition is that this would be too strong. Intuitively, the $\mathcal{EL}$-concept description $\mathsf{GO\_0000018} \sqcap \mathsf{part\_of}.\mathsf{GO\_0006312}$ is not necessarily contained in GO_0000019, but it is known that the reverse containment holds.

The translation of the whole GO yields a general $\mathcal{EL}$-TBox, but not yet an $\mathcal{EL}$-TBox due to the existence of concept inclusions. In Chapter 3, we have shown how to get rid of this kind of concept inclusions w.r.t. gfp- and descriptive semantics. It is more natural to interpret GO w.r.t. descriptive semantics. Nevertheless, for the experimental benefit, we also take into account gfp-semantics. The corresponding $\mathcal{EL}$-TBox of GO w.r.t. gfp-semantics, denoted by $\mathcal{T}^{\mathsf{GO}}_{gfp}$, contains a definition

$$\mathsf{GO\_0000019} \equiv \mathsf{GO\_0000018} \sqcap \mathsf{part\_of}.\mathsf{GO\_0006312}$$

for the above GO concept definition. Similarly, the corresponding $\mathcal{EL}$-TBox of GO w.r.t. descriptive semantics, denoted by $\mathcal{T}^{\mathsf{GO}}_{desc}$, contains a definition

$$\mathsf{GO\_0000019} \equiv \mathsf{GO\_0000019'} \sqcap \mathsf{GO\_0000018} \sqcap \mathsf{part\_of}.\mathsf{GO\_0006312}$$

for the same GO concept definition, with $\mathsf{GO\_0000019'}$ a new primitive concept.

These $\mathcal{EL}$-TBoxes are to be used as inputs of the algorithms for $\mathcal{EL}^{gfp}$- and $\mathcal{EL}^{desc}$-subsumption. Before going to the experiments, we would like to discuss some facts about the Gene Ontology, its corresponding $\mathcal{EL}$-TBoxes, and its corresponding $\mathcal{EL}$-description graphs. In the following, we write $\mathcal{G}^{\mathsf{GO}}_{gfp}$ and $\mathcal{G}^{\mathsf{GO}}_{desc}$ to represent the corresponding $\mathcal{EL}$-description graphs of $\mathcal{T}^{\mathsf{GO}}_{gfp}$ and $\mathcal{T}^{\mathsf{GO}}_{desc}$, respectively.

- There are overall 17,736 GO concepts in the Gene Ontology.

- 933 GO concepts have no definitions, i.e., they are considered primitive. Besides, only 3 of these are not marked *obsolete* and appear as labels of nodes in the $\mathcal{EL}$-description graphs. They are GO_0005575, GO_0008150 and GO_0003674.
- 16,803 GO concepts have definitions and thus are considered defined concepts in $\mathcal{T}_{gfp}^{\mathsf{GO}}$ and $\mathcal{T}_{desc}^{\mathsf{GO}}$.
- $\mathcal{T}_{gfp}^{\mathsf{GO}}$ and $\mathcal{T}_{desc}^{\mathsf{GO}}$ have the same sets of defined concepts and role names, but $\mathcal{T}_{desc}^{\mathsf{GO}}$ contains considerably more primitive concepts than $\mathcal{T}_{gfp}^{\mathsf{GO}}$ does, i.e., a new primitive concept is introduced for each concept definition in $\mathcal{T}_{desc}^{\mathsf{GO}}$, but not in $\mathcal{T}_{gfp}^{\mathsf{GO}}$.
- GO has no terminological cycles. Consequently, the $\mathcal{EL}$-description graphs are acyclic.
- The $\mathcal{EL}$-description graphs—both $\mathcal{G}_{gfp}^{\mathsf{GO}}$ and $\mathcal{G}_{desc}^{\mathsf{GO}}$—have 16,803 nodes, 11,275 edges and a singleton edge label which is part_of.
- The maximum length of is_a-dependency chains in GO, i.e., the paths of GO concepts linked together by the is_a relationship, is 13.

7.2 The Experiments

In this thesis, we have developed three subsumption algorithms for the description logic $\mathcal{EL}$. Two of them have been implemented: the algorithms for subsumption w.r.t. non-general TBoxes (with descriptive and gfp-semantics). The reason that the $\mathcal{EL}^{gci}$ algorithm presented in Chapter 6 has not been implemented is that our aim is to classify the Gene Ontology, and GO can adequately be represented by an $\mathcal{EL}$-TBox as shown in the previous section. In this section, we present the results of our experiments using the $\mathcal{EL}^{gfp}$ and $\mathcal{EL}^{desc}$ algorithms with GO.

Testing Environment

Both $\mathcal{EL}^{gfp}$ and $\mathcal{EL}^{desc}$ algorithms are implemented in the Common LISP[2] language because it is well-suited to realize the data structure for our $\mathcal{EL}$-description graphs. Moreover, choosing LISP as the implementation language enables our algorithms to easily read in TBoxes in LISP-like syntax. This is compatible to highly optimized reasoners for expressive description logics like RACER [27] and FaCT [28].

The Gene Ontology is first translated into two $\mathcal{EL}$-TBoxes: $\mathcal{T}_{gfp}^{\mathsf{GO}}$ and $\mathcal{T}_{desc}^{\mathsf{GO}}$ as mentioned in previous section, both in LISP-like syntax. For instance, the

[2]We use Allegro© CL, which is a LISP language implementation from Franz Inc. [11]

corresponding $\mathcal{EL}$-concept definition for the GO concept GO_0000019 will look like

```
(DEFCONCEPT GO_0000019 (AND GO_0000018
                            (EXIST part_of GO_0006312)))
```

in $\mathcal{T}_{gfp}^{\mathsf{GO}}$ and

```
(DEFCONCEPT GO_0000019 (AND GO_0000019P
                            GO_0000018
                            (EXIST part_of GO_0006312)))
```

in $\mathcal{T}_{desc}^{\mathsf{GO}}$. We have generated a number of $\mathcal{EL}$-TBoxes with different numbers of GO concepts which will used as benchmarks. The benchmarks are measured on a standard PC with a 1.7GHz Pentium-4 processor and 512MB of memory, running Linux©RedHat as the operating system.

Results

As noted in the previous section, the Gene Ontology (GO) contains no cyclic dependencies, and consequently neither do its corresponding TBoxes. Since the gfp- and descriptive semantics of *acyclic* $\mathcal{EL}$-TBoxes coincide, we can apply either $\mathcal{EL}^{gfp}$ or $\mathcal{EL}^{desc}$ algorithm to $\mathcal{T}_{gfp}^{\mathsf{GO}}$ ($\mathcal{T}_{desc}^{\mathsf{GO}}$, respectively) to compute the (unique) subsumption hierarchy. We have seen that, in comparison with $\mathcal{EL}^{desc}$-subsumption, $\mathcal{EL}^{gfp}$-subsumption enjoys better complexity. Therefore, besides the typical experiments of $\mathcal{EL}^{gfp}$ and $\mathcal{EL}^{desc}$ on the Gene Ontology, we also perform an experiment of $\mathcal{EL}^{gfp}$ algorithm with $\mathcal{T}_{desc}^{\mathsf{GO}}$ as its input. Concisely, we carry out 3 experiments as follows:

1. the algorithm for $\mathcal{EL}^{gfp}$-subsumption with the input $\mathcal{T}_{gfp}^{\mathsf{GO}}$ ($\mathcal{EL}^{gfp} + \mathcal{T}_{gfp}^{\mathsf{GO}}$).
2. the algorithm for $\mathcal{EL}^{desc}$-subsumption with the input $\mathcal{T}_{desc}^{\mathsf{GO}}$ ($\mathcal{EL}^{desc} + \mathcal{T}_{desc}^{\mathsf{GO}}$).
3. the algorithm for $\mathcal{EL}^{gfp}$-subsumption with the input $\mathcal{T}_{desc}^{\mathsf{GO}}$ ($\mathcal{EL}^{gfp} + \mathcal{T}_{desc}^{\mathsf{GO}}$).

Figure 7.1 depicts the time required for each experiment with different numbers of concept definitions. Consider, for example, the curve of Experiment 3: by adding 200 more definitions from 400 to 600, the time rises *21.92* seconds; from 800 to 1000, it requires additional *30.79* second time; and adding the same amount of 200 definitions from 1800 to 2000 soars the time up about *53.88* seconds. To sum up, the time tends to increase faster with a large number of definitions (the size of input). In fact, the curves of all three experiments reflect that the time increases polynomially in the number of definitions. Experiment 3 gives the best result, since it needs less time than Experiment 1 and 2, and we will

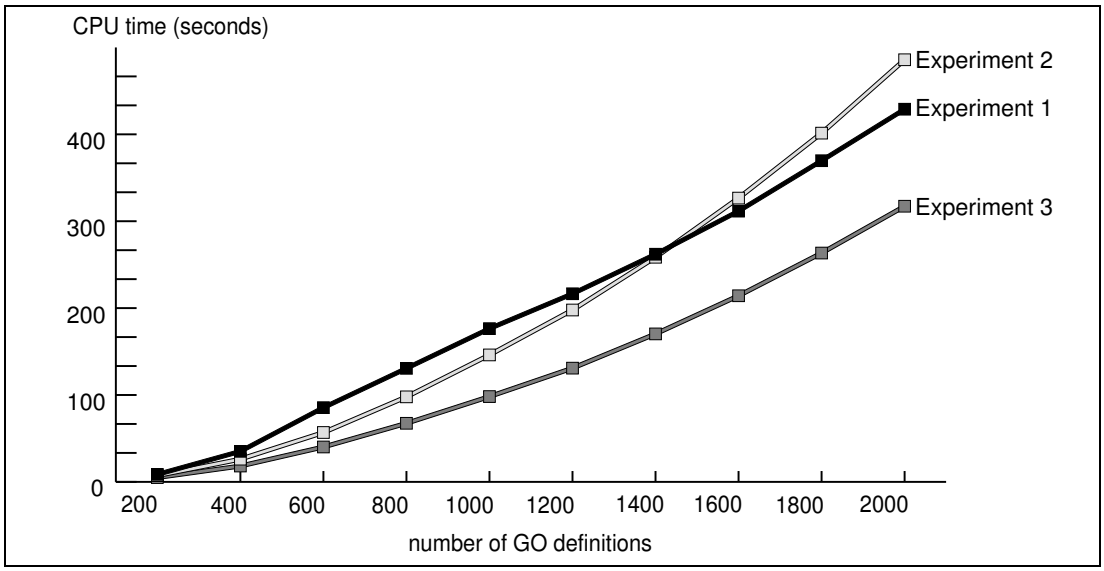

Figure 7.1: Experiment results of the $\mathcal{EL}$-subsumption algorithms on the Gene Ontology

see later in this chapter that only Experiment 3 is able to compute subsumption w.r.t. the whole Gene Ontology.

Let us now consider Experiment 2 in comparison with Experiment 1. With a small number of concept definitions, Experiment 2 takes less time than Experiment 1. However, due to the higher degree of the polynomial of the $\mathcal{EL}^{desc}$ algorithm, Experiment 2 will be worse with greater numbers of concept definitions. More accurately, with 1000 concept definitions, Experiment 2 takes *146.12* seconds, while Experiment 1 needs *176.41* seconds. Now if we input a bigger $\mathcal{EL}$-TBox, say 2000 concept definitions, Experiment 2 needs *485.48* seconds, whereas Experiment 1 only takes *428.91* seconds.

We now consider only descriptive semantics for the final result of the subsumption hierarchy on the whole Gene Ontology, since this reflects the intuition of the Gene Ontology more than gfp-semantics does. As said above, we can either apply the $\mathcal{EL}^{desc}$ algorithm or exploit the $\mathcal{EL}^{gfp}$ algorithm on $\mathcal{T}^{\mathsf{GO}}_{desc}$, as both are equivalent on acyclic $\mathcal{EL}$-TBoxes. Experiment 2 and 3 show that the $\mathcal{EL}^{gfp}$ algorithm is more efficient—in the sense that it runs faster for a given number of definitions, or that it achieves more definitions in the given period of time—at least on the Gene Ontology.

With Experiment 3, we are able to compute the subsumption hierarchy for the whole Gene Ontology. The experiment $\mathcal{EL}^{gfp} + \mathcal{T}^{\mathsf{GO}}_{desc}$ takes approximately 33 minutes, about 18 minutes of which contributes to CPU time. We divide the algorithm into 3 parts and analyse them, which are (i) the normalization part,

Time (mm:ss)	**CPU time**	**Real time**	
Normalization part	4:05	7:32	*23.08%*
Initialization part	8:41	16:09	*49.49%*
Sharpening part	4:49	8:57	*27.43%*
Total	17:35	32:38	

Figure 7.2: Elapse time spent on running $\mathcal{EL}^{gfp}$ algorithm with the input $\mathcal{T}^{\mathsf{GO}}_{desc}$

(ii) the initialization part of Procedure $\mathcal{EL}^{gfp}$-`EfficientSimilarity`, and (iii) the sharpening part of this procedure. The time consumption with respect to these three parts are illustrated in Figure 7.2. According to the table, about a quarter of the time is spent on each of Part (i) and (iii), whilst the rest of the time, about a half, is spent on Part (ii).

The algorithm yields the subsumption hierarchy on the Gene Ontology with 112,292 subsumption outcomes. For instance, GO concepts that subsume the example GO concept on Page 45—besides `GO_0000019` itself and `GO_0000018` which is stated explicitly in the definition through the `is_a` relationship—are `GO_0008150`, `GO_0019219`, `GO_0019222`, `GO_0050789`, `GO_0050791` and `GO_0051052`.

Chapter 8

Conclusion

We have proposed three algorithms to decide subsumption between concepts in the description logic $\mathcal{EL}$. With respect to (non-general) TBoxes, a characterization of $\mathcal{EL}$ subsumption through graph simulation has been proposed by Baader [1]. This characterization shows the tractability of subsumption in $\mathcal{EL}$ w.r.t. TBoxes and the three semantics introduced by Nebel [7]. We have shown further that with this approach, $\mathcal{EL}^{gfp}$-subsumption can be decided in time cubic in the size of the input TBox, whilst $\mathcal{EL}^{desc}$-subsumption can be decided in quartic time, i.e., $\mathcal{O}(|\mathcal{T}|^4)$. We have devised the algorithm $\mathcal{EL}^{gfp}$-`EfficientSimilarity` to compute the greatest simulation on the $\mathcal{EL}$-description graph, and thus subsumption w.r.t. gfp-semantics. This algorithm generalizes the "efficient similarity" algorithm from [8] by taking into account edge-labeled graphs. Concerning descriptive semantics, we reduce the subsumption problem w.r.t. TBoxes to the satisfiability problem of Horn formulae and apply the linear-time algorithm for Horn-SAT from [9] to our $\mathcal{EL}^{desc}$ algorithm.

In addition, we have proposed an efficient $\mathcal{EL}^{gci}$ algorithm to compute concept subsumption in $\mathcal{EL}$ with GCIs. It was shown by Brandt [4] that subsumption in $\mathcal{ELH}$—the description logic $\mathcal{EL}$ admitting GCIs and simple role inclusions—can be decided in polynomial time. Discarding simple role inclusions, we have proved that $\mathcal{EL}^{gci}$-subsumption can be computed in cubic time. We have introduced a new normal form for general TBoxes and proposed a linear-time normalization. The characterization of $\mathcal{EL}^{gci}$-subsumption through so-called implication sets is proposed by Brandt [4]. These implication sets can be translated into a Horn formula. Again, we exploit the linear-time algorithm for Horn-SAT [9] in our $\mathcal{EL}^{gci}$ algorithm. Like the first two algorithms, the $\mathcal{EL}^{gci}$ algorithm computes a subsumption hierarchy once and uses it to answer all subsequent subsumption queries.

The $\mathcal{EL}^{gfp}$ and $\mathcal{EL}^{desc}$ algorithms have been implemented in the Common LISP language and evaluated using the Gene Ontology [10] as a benchmark. With the $\mathcal{EL}^{gfp}$ algorithm, we are able to classify the whole Gene Ontology with

more than one hundred thousand subsumption outcomes.

As depicted in Chapter 2, both descriptive and gfp-semantics are significant in the description logic $\mathcal{EL}$, but their use are not always mutually exclusive. In other words, there may be an application that requires a portion of the terminology to be interpreted w.r.t. gfp-semantics, while the rest is interpreted w.r.t. descriptive semantics. Especially, this could mean the integration of GCIs and $\mathcal{EL}$-concept definitions w.r.t. gfp-semantics. Therefore, it might be interesting to combine the $\mathcal{EL}^{gfp}$ and $\mathcal{EL}^{gci}$ algorithms to compute subsumption w.r.t. such integrated terminologies.

Bibliography

[1] Franz Baader: Terminological cycles in a description logic with existential restriction. In Georg Gottlob and Toby Walsh, editors, *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 325–330. Morgan Kaufmann, 2003.

[2] Franz Baader: Terminological cycles in KL-ONE-based knowledge representation languages. *In Proceedings of the Eighth National Conference on Artificial Intelligence, AAAI-90*, pages 621–626, Boston (USA), 1990.

[3] Franz Baader: Using automata theory for characterizing the semantics of terminological cycles. *Annals of Mathematics and Artificial Intelligence*, 18(2–4):175–219, 1996.

[4] Sebastian Brandt: On Subsumption and Instance problem in $\mathcal{ELH}$ w.r.t. gerneral TBoxes. In *Proceedings of the 2004 International Workshop on Description Logics (DL2004)*, CEUR-WS, 2004.

[5] Sebastian Brandt: Subsumption and instance problem in $\mathcal{ELH}$ w.r.t. general TBoxes. LTCS-Report LTCS-04-04, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology, Germany, 2004. See *http://lat.inf.tu-dresden.de/research/reports.html*.

[6] R. Küsters. Characterizing the semantics of terminological cycles in $\mathcal{ALN}$ using finite automata. In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 499–510. Morgan Kaufmann, 1998.

[7] Bernhard Nebel. Terminological cycles: Semantics and computational properties. In John F. Sowa, editor, *Principles of Semantic Networks*, pages 331–361. Morgan Kaufmann, Los Altos, 1991.

[8] Monik R. Henzinger, Thomas A. Henzinger, and Peter W. Kopke: Computing simulations on finite and infinite graphs. In *36th Annual Symposium*

on Foundations of Computer Science, pages 453–462, Milwaukee, Wisconsin, 1995. IEEE Computer Society Press.

[9] William F. Dowling and Jean Gallier. Linear-time algorithms for testing the satisfiability of propositional horn formulae. *Journal of Logic Programming*, 1(3):267–284, 1984.

[10] Gene Ontology™ Consortium.
See *http://www.geneontology.org/GO.consortiumlist.html*.

[11] Allegro© Common LISP. Franz Inc.
See *http://www.franz.com/*.

[12] R. M. Quillian. Semantic Memory. In Minsky, editor, *Semantic Information Processing*, pages 216–270. MIT Press, 1968.

[13] M. L. Minsky. A framework for representing knowledge. In Winston, editor, *The Psychology of Computer Vision*, pages 211–277. McGraw-Hill, 1975.

[14] Brachman, Ronald J., and James G. Schmolze. An Overview of the KL-ONE Knowledge Representation System. *Cognitive Science* 9(2):171–216.

[15] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptive with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[16] A. Rector and I. Horrocks. Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In *Proceeding of the WS on Ontological Engineering, AAAI Spring Symposium (AAAI'97)*. AAAI Press, 1997.

[17] G. De Giacomo and M. Lenzerini. Concept language with number restrictions and fixpoints, and its relationship with μ-calculus. In *Proceedings of the 11th Europe Conference on Artificial Intelligence (ECAI'94)*, pages 411–415, 1994.

[18] Klaus Schild. Terminological cycles and the propositional μ-calculus. In J. Doyle, E. Sandewall, and P. Torasso, editors, *Proceedings of the 4th International Conference on the Principles of Knowledge Representation and Reasoning (KR'94)*, pages 509–520, Bonn (Germany), 1994. Morgan Kaufmann, Los Altos.

[19] J. W. Lloyd. Foundations of Logic Programming. Springer, Berlin, Heidelberg, 1987.

[20] Yevgeny Kazakov and Hans De Nivelle. Subsumption of concepts in $\mathcal{FL}_0$ for (cyclic) terminologies with respect to descriptive semantics is PSPACE-complete. In *Proceedings of the 2003 International Workshop on Description Logics (DL 2003)*, CEUR-WS, 2003.

[21] M. Buchheit, F. M. Donini, and A. Schaerf. Decidable reasoning in terminological knowledge presentation systems. *Journal of Artificial Intelligence Research*, 1, pages 109–138, 1993.

[22] Ian Horrocks, Ulrike Satler, and Stephan Tobies. Practical reasoning for expressive description logics. In Harald Ganzinger, David McAllester, and Andrei Voronkov, editors, *Proceedings of Automated Reasoning (LPAR'99)*, pages 161–180. Springer-Verlag, 1999.

[23] R. Cote, D. Rothwell, J. Palotay, R. Beckett, and L. Brochu. The systematized nomenclature of human and veterinary medicine. Technical report, SNOMED International, Northfield, IL, 1993.

[24] K. Spackman. Normal forms for description logic expressions of clinical concepts in SNOMED RT. *Journal of the American Medical Informatics Association*, Symposium Supplement, 2001.

[25] A. Rector, W. Nowlan, and A. Glowinski. Goals for concept representation in the GALEN project. In *Proceedings of the 17th annual Symposium on Computer Applications in Medical Care (SCAMC)*, Washington, USA, pages 414–418, 1993.

[26] A. Rector. Medical informatics. In Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation, and Applications*, pages 406–426, Cambridge University Press, 2003.

[27] Volker Haarslev and Ralf Müller. RACER System Description. In *Proceeding of the International Joint Conference on Automated Reasoning (IJCAR'2001)*, volume 2083 of LNAI, pages 701-706, Siena, Italy, 2001.

[28] I. Horrocks. The FaCT system. In H. de Swart, editor, *Automated Reasoning with Analytic Tableaux and Related Methods: International Conference Tableaux'98*, number 1397 in Lecture Notes in Artificial Intelligence, pages 307–312. SpringerVerlag, Berlin, May 1998.