

**TECHNISCHE
UNIVERSITÄT
DRESDEN**

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

Master Thesis

NON-STANDARD INFERENCE FOR EXPLAINING
SUBSUMPTION
IN THE DESCRIPTION LOGIC \mathcal{EL} WITH GENERAL CONCEPT
INCLUSIONS AND COMPLEX ROLE INCLUSIONS

By
Huang Changsheng

Professor: Prof. Franz Baader
Supervisor: Boontawee Suntisrivaraporn

Technische Universität Dresden
Department Of
Computer Science

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled “**Non-standard Inference for Explaining Subsumption in the Description Logic \mathcal{EL} with General Concept Inclusions and Complex Role Inclusions**” by **Huang Changsheng** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 01.03.2007

TECHNISCHE UNIVERSITÄT DRESDEN

Date: **01.03.2007**

Author: **Huang Changsheng**

Matrikel-Nr: **2992270**

Title: **Non-standard Inference for Explaining Subsumption
in the Description Logic \mathcal{EL} with General Concept
Inclusions and Complex Role Inclusions**

Department: **Computational Logic**

Degree: **M.Sc.** Convocation: **March** Year: **2007**

Permission is herewith granted to Technische Universität Dresden to circulate and to make copies for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions.

Signature of Author

THE AUTHOR RESERVES OTHER PUBLICATION RIGHTS, AND NEITHER THE THESIS NOR EXTENSIVE EXTRACTS FROM IT MAY BE PRINTED OR OTHERWISE REPRODUCED WITHOUT THE AUTHOR'S WRITTEN PERMISSION.

THE AUTHOR ATTESTS THAT PERMISSION HAS BEEN OBTAINED FOR THE USE OF ANY COPYRIGHTED MATERIAL APPEARING IN THIS THESIS (OTHER THAN BRIEF EXCERPTS REQUIRING ONLY PROPER ACKNOWLEDGEMENT IN SCHOLARLY WRITING) AND THAT ALL SUCH USE IS CLEARLY ACKNOWLEDGED.

To my parents, grandmother and Echo

Abstract

Ontologies [11] are now ubiquitous and many of them are currently being ported into logical formalisms, most notably description logic (DL) [2]. It is inevitable that such migration might introduce inconsistencies – both in terms of logical and ontological – which could be far from obvious. This motivates the recent research topic of explanation of DL-based ontologies. Explanation comes in two flavors: pinpointing [5, 21, 15] which addresses the source of inconsistencies found in the ontology and debugging [14] which recovers the ontology into a consistent state. Since the latter often requires information from the former, we consider axiom pinpointing as essential for both flavors of the explanation problem. Much of the research in this area is focusing on expressive DLs, in which standard reasoning alone is already highly intractable. In this paper, we investigate this problem in a tractable extension of \mathcal{EL} which is useful in life science applications.

We have discovered that pinpointing is inherently intractable – despite the tractable logic considered – if all information is required. This is due to the combinatorial blow-up of possible sets of axioms. We develop a labelled algorithm for axiom pinpointing based on the \mathcal{EL} subsumption algorithm and the known labelling technique used in tableau algorithm. For implementation purposes, we restrict this algorithm to computation of only partial information, for which polynomial-time algorithm can be obtained. We have experimented this approach on GALEN [18] and found that even partial information can already help ease the way an ontology is being debugged.

Acknowledgements

I would like to thank Boontawee (Meng) Suntisrivaraporn, my supervisor, for his many suggestions during this thesis. Without his patience and constant support I could not have done this thesis in time.

Professor Franz Baader gave the lecture “Logic Based Knowledge Representation” and hereby brought me into the field of description logics, I appreciate his guidance and proposal during this thesis. I had the pleasure of meeting other people in the group. They are wonderful people. Thanks to Rafael for sharing and providing with me his knowledge of complexity, to Liu for giving his advice. I am grateful to Dr Barbara Morawska, with who I have happily worked on my project and seminar. Thanks for the kindness of secretary Frau Achtruth too.

Of course, I wish to thank the following: as the first reader of my thesis report, Fine and Margaret always encourage and pray for me. Another reader Lin helps me printing the first draft. I do not forget my friends in Germany. Thanks all of you to bring me joy in my life.

Most importantly, I am grateful to my families, for their love and financial support. Without them this work would never have come into existence. I must mention Echo, thanks for your love and support during my study in Germany. Particularly, I would like to dedicate this work to my grandmother.

Dresden Germany
Feb 28, 2007

Huang Changsheng

Table of Contents

Abstract	v
Acknowledgements	vi
Table of Contents	vii
Introduction	1
1 The Description Logic \mathcal{EL}	6
1.1 \mathcal{EL} Concept Language	6
1.2 \mathcal{EL}^+ Terminology	8
1.3 Reasoning problem in \mathcal{EL}^+	9
2 Subsumption of \mathcal{EL} with GCIs and CRIs	12
2.1 Normalization of \mathcal{EL}^+ Ontology	12
2.2 Classification of \mathcal{EL}^+ Ontology	16
2.3 Soundness and Completeness	18
3 Explaining Subsumption in \mathcal{EL}^+ by Axiom Pinpointing	20
3.1 The Labelled Algorithm	22
3.2 Soundness and Completeness	25
3.3 Minimal Explanation w.r.t Original Ontology	32
3.4 Complexity	35
4 A Tractable Restriction of Axiom Pinpointing in \mathcal{EL}^+	39
4.1 Computing One Explanation	39
4.2 A First Implementation	45
4.3 Complexity	47
5 Experiments on GALEN	49
6 Conclusion and Future Work	53

Introduction

Description Logics (DLs) [2] are a successful family of logic-based knowledge representation (KR) languages, which can be used to represent the conceptual knowledge of an application domain in a structured and formally well-understood way. As an increasingly large amount of applications in the field of KR, DLs play a more important role, from the foundation of logical ontology language which are used in several areas such as databases, the semantic web, biomedical ontologies, and natural language processing. DLs assume some classes of objects (*concept names*) and some binary relationships between objects (*role names*), then use these classes and relationships to describe properties of “objects” in a domain of knowledge base. With the help of concept and role names, DLs allow to build complex concept descriptions via a set of *constructors*. The language for building descriptions is characteristic of each DL, and different DLs are distinguished by their description languages. For example, a typical description logic language \mathcal{ALC} (Attributive Language with Complements) is built from the constructors value restrictions (\forall), existential restrictions (\exists), and all Boolean operators such as conjunctions (\sqcap), disjunctions (\sqcup), negation (\neg), and the top concept (\top). Usually, the stronger the expressive power of a description logic, the higher its complexity.

DLs consist of two components, a *TBox* and an *ABox*. A TBox stands for *terminology box* which declares general properties of concepts relevant in the domain, while an ABox contains *assertions* about individuals which can be related via roles and can be an instance of of concepts. The most basic form of terminological declarations is a *concept definition*

$A \equiv C$, in which the concept name A was defined by the concept description C . In addition, the TBox may contain a generic form of terminological axioms or so-called *general concept inclusions* (GCIs) denoted by $C \sqsubseteq D$, where C and D are arbitrary concept descriptions. GCI represents a universally true implication. If a TBox consists of finite GCIs, we call this TBox *general* differing from *non-general* TBox or TBox without GCIs. In fact, a concept definition $A \equiv C$ could be expressed by means of two GCIs, that means the defined concept and the concept description are mutually inclusive such as $A \sqsubseteq C$ and $C \sqsubseteq A$, but not vice versa. As a result, we allow the concept definitions in our knowledge base when talking about general TBox. From the application point of view, besides concept axioms, axioms statements concerning roles are widely used in the biomedical domain, for instance, the Galen Medical Knowledge Base (GALEN)[18]. A prominent form of role axioms is called *complex role inclusion* or CRI ($r_1 \circ r_2 \circ \dots \circ r_n \sqsubseteq s$), where r_i, s are role name and “ \circ ” denote the composition of the binary relation from the semantic point of view. In this paper, we take into account the description logic \mathcal{EL} which is built from existential restrictions, conjunctions, and the top concept. With regards to terminological formalisms, we consider \mathcal{EL}^+ terminology, which has \mathcal{EL} as its concept language and allows for GCIs and CRIs as well.

A DL based system not only stores terminologies and assertions, but also offers services to reason about them. Typical inference problems in DLs are to determine whether a concept description is satisfiable, or whether one concept description is more general than the other. The latter is known as subsumption problem of two concept descriptions. Computing the concept subsumption targets a determination of subconcept-superconcept relationship. In [7], by means of the so-called *implication set* (IS), it above all computes all subsumers of every concept name in an \mathcal{EL} general TBox, and the decision of subsumption between two concept names simply boils down to a look-up operation of the subsumer in the IS. This approach provides a polynomial decision procedure for the language at hand, and was

later extended to the more powerful language \mathcal{EL}^{++} . A refined algorithm based on [1] has been proposed for an efficient implementation in [3] and empirical success of this refined algorithm has been witnessed by the scalability of the CEL reasoner [4].

Though standard DL reasoning can be used to make implicit consequence explicit, it does not explain the reasons for a given consequence. This kind of support becomes significantly necessary as the size of the DL knowledge base grows. A case in point is the debugging of OWL in the application Semantic Web [6]. Finding the origin of errors is an extremely difficult task even for the experts. In this case, following the chain of evidence or information that is responsible for inconsistency is very important. On the other hand, when reasoning upon an enormous terminology, for instance, the Galen Medical Knowledge Base with approximately 4,370 numbers of axioms, explanation provides a more readily evidence for a given consequence. In all circumstances, it needs to pin down to the axioms that are pertinent to a certain consequence. At the end, one could retrieve the related axioms leading to the result. We utilize a technique that could retain this kind of information when reasoning. The approach adopted here is known as *pinpointing*, which was first introduced in [5]. Axiom pinpointing is a first step towards providing an explanation. Given a DL knowledge base and a logical consequence, it computes minimal (maximal) subsets of the KB from which the consequence follows (does not follow). Another utility of pinpointing could be found in [21], in that survey, the aim is to provide a complete algorithm for computing the so-called *Minimal Unsatisfiability-Preserving Sub-TBoxes* (MUPS) in \mathcal{ALC} , that is to identify the relevant axioms in charge of an unsatisfiable concept. Recently, pinpointing in tableaux has been investigated in [15]. In this paper, we sketch a pinpointing algorithm for computing the *minimal explanations* of subsumption in the description logic \mathcal{EL}^+ . Semantically, minimality is w.r.t set inclusion, i.e the consequence would not hold anymore if an item is removed from it. Since no concept descriptions are unsatisfiable because of the absence of negation in \mathcal{EL} , we will only consider concept subsumption. In the spirit

of naming scheme in [21], we are interested in *Minimal Ontology Preserving Subsumption* (MOPS) instead of MUPS or *minimal explanation sets*. In the following, we use “MOPS” and “minimal explanation set” interchangeably.

Despite the limited expressive power, \mathcal{EL} was used in large biomedical ontologies such as the Systematized Nomenclature of Medicine (SNOMED), the Gene Ontology. From the complexity point of view, subsumption in \mathcal{EL} can be decided in polynomial time even admitting GCIs and CRIs [1], and the polynomial time reasoner CEL was successfully fulfilled. In this thesis, We will first propose a pinpointing algorithm and analyse that the complexity of pinpointing in \mathcal{EL}^+ is **NP-hard**. In order to retain it to tractable, we modify this pinpointing algorithm a little and compute only one minimal explanation with the help of *brute-force* or *black box* reasoning approach and CEL reasoner. This tractable axiom pinpointing algorithm was implemented in Common LISP, Allegro CL [12]. Since large portion of GALEN can be expressed in \mathcal{EL}^+ , we will evaluate our implementation using the GALEN ontology.

This thesis is organized as follows:

Chapter 1 introduces the relevant definitions of the description logic \mathcal{EL} , starting with the syntax and semantics of its concept language. Then the extended language \mathcal{EL}^+ of \mathcal{EL} and its terminology is introduced. We also give a brief introduction to the inference problems in \mathcal{EL}^+ .

In Chapter 2, we introduce the normal form and normalization rule [23] for \mathcal{EL}^+ terminologies. Then we review the subsumption algorithm through the computation of implication sets, and show the soundness and completeness [1] of this algorithm.

Subsequently, we extend the algorithm mentioned in Chapter 2 and propose a labelled algorithm that could pinpoint relevant axioms in Chapter 3. The proof of the soundness and completeness of the labelled algorithm was reduced to the unlabelled algorithm via

a so-called ϵ -projection. Since this labelled algorithm processes on a normalized ontology, and the computed explanation sets are in normal form as well. When use reverse mapping from normal from to original form, it leads to non-minimality. In order to obtain the explanations w.r.t the original axioms, we could use brute force approach. Finally we prove that the complexity of pinpoint is **NP-complete**.

In Chapter 4, we restrict the labelled algorithm so that it computes only one explanation for each subsumption relationships. This algorithm terminates in polynomial time in the size of the input ontology. However, it cannot produce truly minimal explanation sets. To obtain minimal ones, we exploit the *black box* approach with the help of the CEL reasoner. At the end, we implement a refined version of the algorithm in Common Lisp.

The experiment of the tractable pinpointing algorithm on GALEN will be discussed in Chapter 5. We present the experimental results on the computation time and the degree of the non-minimality, e.g how many explanations are not minimal and how many axioms are unnecessary for a given non-minimal explanation.

Finally, conclusion and future work are discussed in Chapter 6.

Chapter 1

The Description Logic \mathcal{EL}

In this chapter we give a formal introduction into the description logic. Since this thesis concerns the description logic language \mathcal{EL} , we start with introducing the syntax and semantics of \mathcal{EL} , then the definition of \mathcal{EL}^+ terminology which is an extension of \mathcal{EL} . Finally, we will also present the main inference problem in this logic, i.e, subsumption. The solution for this inference problem will be discussed in the next chapter. In the first place, let us take a look at the syntax and semantics of \mathcal{EL} .

1.1 \mathcal{EL} Concept Language

Description Logics (DLs) are an important family of formalisms used to represent and reason about ontologies in the field of knowledge presentations. It assumes some *concepts* and binary relation *roles* and then uses these concepts and relations to describe properties of objects in a domain of a knowledge base. *Concept descriptions* or *complex concepts* are inductively defined with the help of a set of constructors, starting with a set N_{con}^{\top} of *concept names* including \top and a set N_{role} of *role names*. The description logic \mathcal{EL} allows only for the constructors of conjunctions (\sqcap), existential restrictions (\exists) and \top concept.

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

- each concept name $A \in N_{\text{con}}^{\top}$ is an \mathcal{EL} -concept description

- if C and D are \mathcal{EL} -concept descriptions and $r \in N_{\text{role}}$, then the conjunction $C \sqcap D$ and the existential restriction $\exists r.C$ are also \mathcal{EL} -concept descriptions.

◇

Concept names from N_{con} as well as \top referred to as N_{con}^\top and are called *atomic* concepts. Other concepts are called *non-atomic* or *complex*.

For instance, in the \mathcal{EL} concept description

$$\text{Person} \sqcap \exists \text{attends. Course}$$

Here 'Person' and 'Course' are concept names and 'attends' is a role name. Intuitively this complex concept describes the class of persons who attend a course. We can describe a more specific concept such as

$$\text{Person} \sqcap \exists \text{attends. (Course} \sqcap \exists \text{has_topics. DL)}$$

which describes those who attends the course with the topic description logic. The set theoretic semantics of the \mathcal{EL} concept descriptions can be defined as follows:

Definition 2. (*Semantics*) An *interpretation* \mathcal{I} is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set and the interpretation function $\cdot^{\mathcal{I}}$ maps

- each concept name $A \in N_{\text{con}}^\top$ to a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$;
- each role name $r \in N_{\text{role}}$ 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:

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

◇

Generally, there are two types of terminological formalisms in \mathcal{EL} , the first one is \mathcal{EL} -TBox and the second one is general TBox. In this paper, we pay attention to the latter one.

1.2 \mathcal{EL}^+ Terminology

In DLs, terminology presents a hierarchical structure built to provide an intensional representation of the domain of interest. Usually, a basic form of declaration in a terminology is a *concept definition* such as $A \equiv C$ which defines a *concept name* A by a *concept description* C . A finite set of concept definitions with unique left hand sides is called TBox (or Terminology).

In contrast to concept definitions, another form of terminological axiom is so-called *general concept inclusion* (GCI) axioms denoted by $C \sqsubseteq D$, where C, D are arbitrary concept descriptions. Besides concept definitions and GCIs, there is a widely utilization of role axioms which are known collectively as *complex role inclusions* or CRIs $r_1 \circ r_2 \cdots \circ r_n \sqsubseteq s$, where r_i and s are role names for $1 \leq i \leq n$. Role inclusions play an important role in some realistic ontologies, especially in the biomedical domain. Next, we will formally define the syntax of the \mathcal{EL}^+ terminology.

Definition 3. (*Syntax*) Let C and D are \mathcal{EL} -concept descriptions and r_i, s are role names for $1 \leq i \leq n$. Then $C \sqsubseteq D$ is a *general concept inclusion* (GCI) and $r_1 \circ r_2 \cdots \circ r_n \sqsubseteq s$ is a *complex role inclusion* (CRI). The \mathcal{EL}^+ terminology or ontology \mathcal{O} is a finite set of GCIs and CRIs. \diamond

Usually, we call that a \mathcal{EL} TBox is general if it contain only a finite set of GCIs. An \mathcal{EL}^+ ontology can be regarded as a general \mathcal{EL} TBox admitting CRIs. \mathcal{EL}^+ ontology contains universally true implication GCI, both C and D here can be arbitrary concept descriptions. In order to differentiate from \mathcal{EL} TBoxes, we henceforth use \mathcal{O} to denote \mathcal{EL}^+ ontology. By the notation $CN_{\mathcal{O}}$, we denote it as the set of all concept names occurring in \mathcal{O} . Similarly, $RN_{\mathcal{O}}$ refers to all role names appearing in \mathcal{O} . Concept names and \top are expressed by $CN_{\mathcal{O}}^{\top}$, i.e $CN_{\mathcal{O}}^{\top} = CN_{\mathcal{O}} \cup \{\top\}$. We allow for concept definitions $A \equiv C$ since a concept definition can be expressed by means of two GCIs, viz. $A \sqsubseteq C$ and $C \sqsubseteq A$. The features provided by \mathcal{EL}^+ are essential and widely used in the context of medical ontologies such as Galen Medical Knowledge Base (GALEN) [18], Systematized Nomenclature of Medicine (SNOMED) [16]

and Gene Ontology (GO) [20]. In particular, role inclusions are practically very useful in these medical ontologies. Most notably, they generalize transitive role axioms ($r \circ r \sqsubseteq r$), role hierarchies ($r \sqsubseteq s$), and so-called right-identities on roles ($r \circ s \sqsubseteq r$).

Now we can define the semantics of the \mathcal{EL}^+ terminology. Usually there are two approaches to interpret terminology formalisms, namely fixpoint or descriptive semantics by Nebel [13]. In this paper, we take into consideration the *descriptive semantics*.

Definition 4. (*Semantics*) An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ satisfies a general concept inclusion $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, and \mathcal{I} satisfies a role inclusion $r_1 \circ r_2 \cdots \circ r_n \sqsubseteq s$ if $r_1^{\mathcal{I}} \circ r_2^{\mathcal{I}} \cdots \circ r_n^{\mathcal{I}} \subseteq s^{\mathcal{I}}$, where “ \circ ” denotes composition of binary relation from semantics point of view. \mathcal{I} is a *model* of a \mathcal{EL}^+ ontology \mathcal{O} if it satisfies all GCIs and CRIs in \mathcal{O} . \diamond

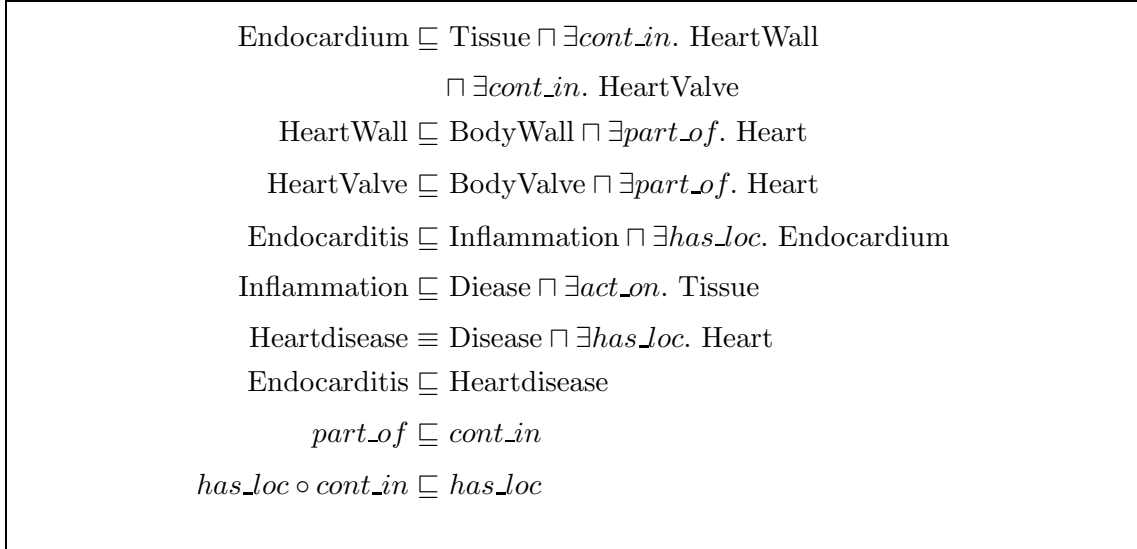
1.3 Reasoning problem in \mathcal{EL}^+

For terminological reasoning w.r.t an ontology, we distinguish *satisfiability* from *subsumption*. Satisfiability of a concept description concerns whether the concept description is free from contradiction, while subsumption concerns whether given two concept descriptions, one concept description is a subconcept of the other. Satisfiability problem is uninteresting in \mathcal{EL}^+ , since without negation, there is no unsatisfiable concept description. We define the subsumption problem as follows:

Definition 5. (*subsumption*) Let \mathcal{O} be an \mathcal{EL}^+ terminology and let C, D be arbitrary \mathcal{EL}^+ concept description. Then C is *subsumed* by D w.r.t \mathcal{O} ($C \sqsubseteq_{\mathcal{O}} D$)¹ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models \mathcal{I} of \mathcal{O} . \diamond

C and D are *equivalent* w.r.t \mathcal{O} iff they subsume each other. i.e $C \equiv_{\mathcal{O}} D$ iff $C \sqsubseteq_{\mathcal{O}} D$ and $D \sqsubseteq_{\mathcal{O}} C$. In fact, the decision of subsumption problem can be reduced to the computation of the subconcept-superconcept relationships. In [1], it computes so called *implication sets*

¹ $C \sqsubseteq D$ is an axiom with the form of GCI in a terminology, while $C \sqsubseteq_{\mathcal{O}} D$ is subsumption problem w.r.t \mathcal{O} .

Figure 1.1: Example \mathcal{EL}^+ ontology

which are the set of subsumers of a concept names, to decide $A \sqsubseteq B$ just to check if B is in the implication set of A . In this way, it has been investigated that reasoning in \mathcal{EL} and its extension remains tractable and the polynomial reasoner CEL is based on the algorithm [1] for the extension of \mathcal{EL}^+ . Subsumption of two concept descriptions can be reduced to the subsumption of two concept names. i.e, for concept descriptions C, D and $C \sqsubseteq_{\mathcal{O}} D$ can be reduced to $A \sqsubseteq_{\mathcal{O}} B$ for the new concept name A, B by introducing the following GCIs

$$A \sqsubseteq C \quad D \sqsubseteq B$$

Let's have a look an example:

Example 6. As an example of what can be expressed with \mathcal{EL}^+ , consider the example in Figure 1.1, this \mathcal{EL}^+ ontology \mathcal{O} contains six GCIs, two CRIs and one concept definition. For example, the Endocardium is Tissue and is contained in the HeartWall and HeartValve. Also we can say that Endocardium is Tissue and is composed of HeardWall and HeartValve on account of *cont_in* which is a super role of *part_of*. In this example, the axioms for role are role hierarchy ($\text{part_of} \sqsubseteq \text{cont_in}$) and right-identities on roles ($\text{has_loc} \circ \text{cont_in} \sqsubseteq \text{has_loc}$) respectively. Based on this ontology, it is not hard to discern that the concept Endocarditis

is subsumed by concept Heartdisease. (i.e. Endocarditis $\sqsubseteq_{\mathcal{O}}$ Heartdisease). \dashv

We say that the ontology \mathcal{O} is an explanation or an ontology preserving the subsumption Endocarditis \sqsubseteq Heartdisease. In fact, not all axioms in \mathcal{O} are indispensable for this subsumption. We are more interesting in seeking the minimal sub-ontologies that preserve the subsumption consequence of a given terminology from application point of view. As motivated in the introduction, we are interested in explaining subsumption in \mathcal{EL}^+ . We now give a definition of minimal explanation.

Definition 7. (*MOPS*) Let \mathcal{O} be an \mathcal{EL}^+ ontology over $CN_{\mathcal{O}}^{\top}$ and $RN_{\mathcal{O}}$ and A, B concept names from $CN_{\mathcal{O}}^{\top}$. Then a *Minimal Ontology Preserving Subsumption* (MOPS) or *minimal explanation* for subsumption $A \sqsubseteq B$ is the minimal sub-ontology \mathcal{O}' of \mathcal{O} such that $A \sqsubseteq_{\mathcal{O}'} B$ and for all $\mathcal{O}'' \subset \mathcal{O}'$, $A \not\sqsubseteq_{\mathcal{O}''} B$. \diamond

Intuitively, minimal explanation contains relevant axioms required for the subsumption in question to hold, i.e., subsumption does not hold any more if an axiom is taken away from it. In the example 6, a possible minimal explanation set is $\mathcal{O}'_1 = \{\text{Endocarditis} \sqsubseteq \text{Heartdisease}\}$ for the subsumption Endocarditis \sqsubseteq Heartdisease.

In this paper, we will denote the size of an ontology \mathcal{O} by $|\mathcal{O}|$ i.e. for the total number of occurrence of role names and concept names in \mathcal{O} . In the following, we will present the polynomial time decision procedure of the subsumption problem in \mathcal{EL}^+ .

Chapter 2

Subsumption of \mathcal{EL} with GCIs and CRIs

In terms of terminological inference services, a common method is tableaux-based algorithm which tries to generate a finite structure representative (possibly infinite) models. In tableaux algorithm, we reduce the satisfiability of a GCI $A \sqsubseteq B$ to the satisfiability problem of the concept description $\neg A \sqcup D$ which is treated as tautology since no negation in \mathcal{EL} . From complexity point of view, in [8], it has been shown that use of standard tableaux algorithm deciding consistency of a general \mathcal{ALC} TBox, which extends \mathcal{EL} by value restrictions (\forall), disjunctions (\sqcup), and negations (\neg), takes exponentially many steps in the worst case, even for the sublanguage \mathcal{EL} . Hence, a new approach is required to render tractability. In this chapter, we will bring in a polynomial algorithm for the concept subsumption. This algorithm takes an ontology in normal form as input and computes subsumption between all pairs of concept names in the ontology. First we will formally address the normalization of an \mathcal{EL}^+ ontology.

2.1 Normalization of \mathcal{EL}^+ Ontology

Definition 8. (\mathcal{EL}^+ Normal Form) Let \mathcal{O} be a \mathcal{EL}^+ ontology over $CN_{\mathcal{O}}$ and $RN_{\mathcal{O}}$. Then \mathcal{O} is in *normal form* iff \mathcal{O} contains only GCIs and CRIs of the following forms:

1. all general concept inclusions have one of the following forms:

$$A \sqsubseteq B$$

$$A_1 \sqcap A_2 \sqsubseteq B$$

$$A \sqsubseteq \exists r.B$$

$$\exists r.A \sqsubseteq B$$

where A, A_1, A_2, B represent concept names from $CN_{\mathcal{O}}$ or \top , r is role name from $RN_{\mathcal{O}}$.

2. all complex role inclusions have one of the following forms:

$$r \sqsubseteq s$$

$$r_1 \circ r_2 \sqsubseteq s$$

where r, r_1, r_2, s are role names from $RN_{\mathcal{O}}$.

◇

Any arbitrary \mathcal{EL}^+ ontology can be normalized by exhaustively applying the normalization rules. Here we will introduce a small extension of the normalization rules first proposed in [23] as the following:

Definition 9. Let \mathcal{O} be a \mathcal{EL}^+ ontology over $CN_{\mathcal{O}}^{\top}$ and $RN_{\mathcal{O}}$, the normalization rule are

defined as:

$$\begin{aligned}
\mathbf{NF1} \quad \hat{r} \circ s \sqsubseteq t &\longrightarrow \{\hat{r} \sqsubseteq u, u \circ s \sqsubseteq t\} \\
\mathbf{NF2} \quad C \doteq D &\longrightarrow \{C \sqsubseteq D, D \sqsubseteq C\} \\
\mathbf{NF3} \quad \hat{C} \sqcap D \sqsubseteq E &\longrightarrow \{\hat{C} \sqsubseteq A, A \sqcap D \sqsubseteq E\} \\
\mathbf{NF4} \quad C \sqcap \hat{D} \sqsubseteq E &\longrightarrow \{\hat{D} \sqsubseteq A, C \sqcap A \sqsubseteq E\} \\
\mathbf{NF5} \quad \exists r. \hat{C} \sqsubseteq D &\longrightarrow \{\hat{C} \sqsubseteq A, \exists r. A \sqsubseteq D\} \\
\mathbf{NF6} \quad \hat{C} \sqsubseteq \hat{D} &\longrightarrow \{\hat{C} \sqsubseteq A, A \sqsubseteq \hat{D}\} \\
\mathbf{NF7} \quad B \sqsubseteq \exists r. \hat{D} &\longrightarrow \{B \sqsubseteq \exists r. A, A \sqsubseteq \hat{D}\} \\
\mathbf{NF8} \quad B \sqsubseteq D \sqcap E &\longrightarrow \{B \sqsubseteq D, B \sqsubseteq E\}
\end{aligned}$$

where \hat{r} denotes a role concatenation of more than one role and u is a new role name not occurring in \mathcal{O} , \hat{C} , \hat{D} denote concept descriptions (complex concepts), A is a new concept name, B is a concept name, r is a role name and C, D, E could be any arbitrary concept descriptions. \diamond

Given a \mathcal{EL}^+ ontology \mathcal{O} , we apply the normalization rule $G \longrightarrow S$ to \mathcal{O} and change \mathcal{O} to $(\mathcal{O} \setminus \{G\} \cup S)$, the normalized \mathcal{EL}^+ ontology $norm(\mathcal{O})$ could be obtained by applying rules **NF1** to **NF5** back and forth (phase 1), and after that, exhaustively applying rule **NF6** to **NF8** (phase 2). The normalized \mathcal{EL}^+ ontology can be computed in linear time in the size of \mathcal{O} .

Lemma 10. *Let \mathcal{O} be an \mathcal{EL}^+ ontology. The normalized \mathcal{EL}^+ ontology $norm(\mathcal{O})$ can be computed in linear time in the size of \mathcal{O} , the resulting ontology $norm(\mathcal{O})$ is of linear size in the size of \mathcal{O} .*

PROOF. The exhaustive application of **NF1** and **NF2** gave rise to the size of \mathcal{O} is increased only linearly, and these two rules will never be applicable as the consequence of Rules **NF3** to **NF8**. Next, we may restrict our attention to Rules **NF3** to **NF8**. A single application of one

of the Rules **NF3** to **NF5** in the first phase increased the size of \mathcal{O} only by a constant. Rules **NF3** and **NF4** are applicable at most once for each occurrence of " \sqcap ", and each application introduces a new concept name and split one GCI into two. Likewise, Rule **NF5** is applicable at most once for each occurrence of " \exists " and generates a new concept and split one GCI into two. Therefore, the application in Phase 1 takes linear time and produces a ontology \mathcal{O}' of size linear in the size of \mathcal{O} .

Rule **NF6** is applicable at most once for each GCI in \mathcal{O}' and leads to the splitting of two GCIs of linear size. Rule **NF7** and Rule **NF8** are applicable at most once for each occurrence of \exists and \sqcap on the right hand side of \mathcal{O}' respectively, in both cases split one GCI into two, which increase the size of \mathcal{O}' by a constant. Therefore, the application of phase 2 yields an ontology of the size linear in the size of \mathcal{O} . ■

Example 11. Let us look over the terminology from Example 6. Only the sixth GCIs needs to be normalized w.r.t phase one. In the first place, rule **NF2** applies to the sixth GCI

$$\text{Heartdisease} \sqsubseteq \text{Disease} \sqcap \exists \text{has_loc. Heart}$$

$$\text{Disease} \sqcap \exists \text{has_loc. Heart} \sqsubseteq \text{Heartdisease}$$

for the second axiom the **NF4** generates the following two GCIs as

$$\text{Disease} \sqcap X_1 \sqsubseteq \text{Heartdisease}$$

$$\exists \text{has_loc. Heart} \sqsubseteq X_1$$

which X_1 is the new introducing variable and these axioms are already in normal form. Then apply **NF6** to **NF8** exhaustively for the rest of ontology, we will get a complete normalized

ontology looks as:

$$\begin{array}{ll}
\text{Disease} \sqcap X_1 \sqsubseteq \text{Heartdisease} & \exists \text{has_loc. Heart} \sqsubseteq X_1 \\
\text{Endocardium} \sqsubseteq \text{Tissue} & \text{Endocardium} \sqsubseteq \exists \text{cont_in. HeartWall} \\
\text{HeartWall} \sqsubseteq \text{BodyWall} & \text{HeartWall} \sqsubseteq \exists \text{part_of. Heart} \\
\text{HeartValve} \sqsubseteq \text{BodyValve} & \text{HeartValve} \sqsubseteq \exists \text{part_of. Heart} \\
\text{Endocarditis} \sqsubseteq \text{Inflammation} & \text{Endocarditis} \sqsubseteq \exists \text{has_loc. Endocardium} \\
\text{Inflammation} \sqsubseteq \text{Disease} & \text{Inflammation} \sqsubseteq \exists \text{act_on. Tissue} \\
\text{Heartdisease} \sqsubseteq \text{Disease} & \text{Heartdisease} \sqsubseteq \exists \text{has_loc. Heart} \\
\text{Endocarditis} \sqsubseteq \text{CriticalDisease} & \text{Endocardium} \sqsubseteq \exists \text{cont_in. HeartValve} \\
\text{has_loc} \circ \text{cont_in} \sqsubseteq \text{has_loc} & \text{part_of} \sqsubseteq \text{cont_in}
\end{array}$$

⊢

It is important to say that an \mathcal{EL}^+ ontology \mathcal{O} and its normal form $\text{norm}(\mathcal{O})$ are equivalent with respect to concept subsumption.

Proposition 12. *Let \mathcal{O} be an \mathcal{EL}^+ ontology over $CN_{\mathcal{O}}^{\top}$ and $RN_{\mathcal{O}}$, for $A, B \in CN_{\mathcal{O}}^{\top}$, $A \sqsubseteq_{\mathcal{O}} B$ iff $A \sqsubseteq_{\text{norm}(\mathcal{O})} B$.*

2.2 Classification of \mathcal{EL}^+ Ontology

The first polynomial time algorithm for classification in \mathcal{EL} in presence of GCIs and role hierarchies was proposed in [7], and this algorithm was further extended the much more powerful DL \mathcal{EL}^{++} in [1]. We will restrict the algorithm from [1] to \mathcal{EL}^+ .

Somewhere along the way, the strategy for our algorithm is, for every concept name $A \in CN_{\mathcal{O}}^{\top}$ to compute a set of concept names $S(A)$ which are super concept names or subsumers of A . Similarly, for every role r we want to represent by $R(r)$. When decide the subsumption $A \sqsubseteq B$, only check if B is in the implication set $S(A)$. In the following, we

Completion Rules	
CR1	If $A \in S(X)$, $g = A \sqsubseteq B \in \mathcal{O}$, and $B \notin S(X)$ then $S(X) := S(X) \cup \{B\}$
CR2	If $A_1, A_2 \in S(X)$, $g = A_1 \sqcap A_2 \sqsubseteq B \in \mathcal{O}$, and $B \notin S(X)$ then $S(X) := S(X) \cup \{B\}$
CR3	If $A \in S(X)$, $g = A \sqsubseteq \exists r.B \in \mathcal{O}$, and $(X, B) \notin R(r)$ then $R(r) := R(r) \cup \{(X, B)\}$
CR4	If $(X, Y) \in R(r)$, $A \in S(Y)$, $g = \exists r.A \sqsubseteq B \in \mathcal{O}$, and $B \notin S(X)$ then $S(X) := S(X) \cup \{B\}$
CR5	If $(X, Y) \in R(r)$, $g = r \sqsubseteq s \in \mathcal{O}$, and $(X, Y) \notin R(s)$ then $R(s) := R(s) \cup \{(X, Y)\}$
CR6	If $(X, Y) \in R(r)$, $(Y, Z) \in R(s)$, $g = r \circ s \sqsubseteq t \in \mathcal{O}$, and $(X, Z) \notin R(t)$ then $R(t) := R(t) \cup \{(X, Z)\}$

Table 2.1: Subsumption algorithm for \mathcal{EL}^+

only take into consideration of the \mathcal{EL}^+ ontology, which consists of finite set of GCIs and CRIs. Without loss of generality we assume that the input ontology \mathcal{O} is in normal form, the algorithm computes

- a mapping S assigning to each element of $CN_{\mathcal{O}}^{\top}$ a subset of $CN_{\mathcal{O}}^{\top}$ and
- a mapping R assigning to each element of $RN_{\mathcal{O}}$ a binary relation on $CN_{\mathcal{O}}^{\top}$.

Starting by $S(A) = \{A, \top\}$ and $R(r) = \emptyset$ for all $A \in CN_{\mathcal{O}}^{\top}$, $r \in RN_{\mathcal{O}}$, then computes S , R exhaustively by application of the completion rule shown in Table 2.1. Intuitively, the mappings disclose the implicit subsumption relationships in the sense $B \in S(A)$ implies $A \sqsubseteq_{\mathcal{O}} B$ and similarly, $(A, B) \in R(r)$ implies $A \sqsubseteq_{\mathcal{O}} \exists r.B$. This algorithm not only computes the subsumption between two given concept names, but it rather *classifies* \mathcal{O} , i.e., it concurrently computes the subsumption relationships between *all* pairs of concept names in \mathcal{O} .

Lemma 13. *Let \mathcal{O} be an \mathcal{EL}^+ ontology in normal form over $CN_{\mathcal{O}}^{\top}$ and $RN_{\mathcal{O}}$, the subsumption algorithm shown in Table 2.1 terminates polynomially in the size of \mathcal{O} .*

PROOF. Since \mathcal{O} is a finite set of GCIs and CRIs, it is easy to check that both $CN_{\mathcal{O}}^{\top}$ and $RN_{\mathcal{O}}$ are linear in $|\mathcal{O}|$. The application of each completion rules either adds a concept name in $CN_{\mathcal{O}}^{\top}$ to $S(A)$ for some $A \in CN_{\mathcal{O}}^{\top}$ or adds a pair $(A, B) \in CN_{\mathcal{O}}^{\top} \times CN_{\mathcal{O}}^{\top}$ to $R(r)$ for some $r \in RN_{\mathcal{O}}$. Since no rule removes element from S and R , the total number of rule application is polynomial. It is easy checked that each rule application can be performed in polynomial time. \blacksquare

2.3 Soundness and Completeness

In this section we will study the soundness and completeness of subsumption algorithm in \mathcal{EL} . Here we only present a proof sketch of soundness for reference in later chapters. The full proofs of soundness and completeness can be found in [1].

Assume that our algorithm is applied to an ontology \mathcal{O} in normal form. The execution of the algorithm will lead to a sequence of S_0, \dots, S_m and R_0, \dots, R_m . To prove the correctness of algorithm, it is sufficient to prove the following claim:

claim: Given concept name $A, B \in CN_{\mathcal{O}}^{\top}$ and role name $r \in RN_{\mathcal{O}}$, for all $n \in \mathbb{N}$, models \mathcal{I} of \mathcal{O} , and $x \in A^{\mathcal{I}}$, the following holds:

- (a) if $B \in S_n(A)$, then $x \in B^{\mathcal{I}}$;
- (b) if $(A, B) \in R_n(r)$ then there is a $y \in \Delta^{\mathcal{I}}$ with $(x, y) \in r^{\mathcal{I}}$ and $y \in B^{\mathcal{I}}$.

We prove it by induction on n .

Induction start:

- (a) $n = 0$ then $S_0(A) = \{A, \top\}$, $x \in A^{\mathcal{I}}$ implies $x \in A^{\mathcal{I}}$ and $x \in \top^{\mathcal{I}}$, we are done.
- (b) $n = 0$ then $R_0(r) = \emptyset$ for all $r \in RN_{\mathcal{O}}$, trivial.

Induction step:

For (a), we assume that $B \in S_n(A) \setminus S_{n-1}(A)$ (for otherwise we are done by IH), we make a case distinction according to the rule that was used to add B to S_n .

- CR1** If **CR1** could be applicable, there must be $B' \in S_{n-1}(A)$ and a concept inclusion $g = B' \sqsubseteq B \in \mathcal{O}$. By point(a) of IH, $x \in A^{\mathcal{I}}$ implies $x \in B'^{\mathcal{I}}$, by concept inclusion g , $x \in B'^{\mathcal{I}}$ implies $x \in B^{\mathcal{I}}$.
- CR2** If **CR2** could be applicable, there must be $B_1, B_2 \in S_{n-1}(A)$ and a concept inclusion $g = B_1 \sqcap B_2 \sqsubseteq B \in \mathcal{O}$. By point(a) of IH, $x \in A^{\mathcal{I}}$ implies $x \in B_1^{\mathcal{I}}$ and $x \in B_2^{\mathcal{I}}$, by concept inclusion g , implies $x \in B^{\mathcal{I}}$.
- CR4** If **CR4** could be applicable, there exist concept names $X, Y \in CN_{\mathcal{O}}^{\top}$ and role name $r \in RN_{\mathcal{O}}$, such that $(X, Y) \in R_{n-1}(r)$, and $A \in S_{n-1}(Y)$ and a concept inclusion $g = \exists r.A \sqsubseteq B \in \mathcal{O}$. By point(b) of IH, there is a $y \in \Delta^{\mathcal{I}}$ with $(x, y) \in r^{\mathcal{I}}$ and $y \in Y^{\mathcal{I}}$. By point(a) of IH, $y \in Y^{\mathcal{I}}$ implies $y \in A^{\mathcal{I}}$. Then $x \in (\exists r.A)^{\mathcal{I}}$ yields $x \in B^{\mathcal{I}}$ by g .

For (b), we assume $(X, Y) \in R_n(r) \setminus R_{n-1}(r)$ (for otherwise we are done by IH), we make a case distinction according to the rule that add the (X, Y) to $R_n(r)$:

- CR3** If **CR3** could be applied, there must be $A \in S_{n-1}(X)$ and a concept inclusion $g = A \sqsubseteq \exists r.B \in \mathcal{O}$. By point(a) of IH, $x \in X^{\mathcal{I}}$ implies $x \in A^{\mathcal{I}}$, by concept inclusion g , there must exist a $y \in \Delta^{\mathcal{I}}$ with $(x, y) \in r^{\mathcal{I}}$ and $y \in B^{\mathcal{I}}$.
- CR5** By the precondition of **CR5**, there exists $(X, Y) \in R_{n-1}(r)$ and a role inclusion $g = r \sqsubseteq s \in \mathcal{O}$. By point(b) of IH and $r \sqsubseteq s$, $x \in X^{\mathcal{I}}$ implies there is a $y \in \Delta^{\mathcal{I}}$ with $(x, y) \in s^{\mathcal{I}}$ and $y \in Y^{\mathcal{I}}$ and are done.
- CR6** The applicability of **CR6** conjecture there exists $(X, Y) \in R_{n-1}(r)$ and $(Y, Z) \in R_{n-1}(s)$ and a role inclusion $g = r \circ s \sqsubseteq t$. By point(2) of IH, there is a $y \in \Delta^{\mathcal{I}}$ with $(x, y) \in r^{\mathcal{I}}$ and $y \in Y^{\mathcal{I}}$, and there is a $z \in \Delta^{\mathcal{I}}$ with $(y, z) \in s^{\mathcal{I}}$ and $z \in Z^{\mathcal{I}}$. By g we have $(x, z) \in t^{\mathcal{I}}$ and are done.

Theorem 14. *Subsumption of \mathcal{EL}^+ ontology \mathcal{O} can be decided in polynomial time.*

In order to show decidability in polynomial time it suffices to show 2 phases, (i) \mathcal{O} can be normalized in polynomial time; (ii) the subsumption algorithm on \mathcal{O} terminates in polynomial time. Both of these two phases have been shown.

Chapter 3

Explaining Subsumption in \mathcal{EL}^+ by Axiom Pinpointing

In the previous chapter, we have already shown the known result that subsumption in \mathcal{EL}^+ can be decided in polynomial time. In fact, there already existed such a polynomial time reasoner CEL [4] for \mathcal{EL}^+ based ontology. On one hand, the standard DL reasoner CEL can efficiently reason about for \mathcal{EL}^+ ontologies, however, it does not give the reason for the inferred consequences. As the size of ontology grows, this support becomes increasingly important. For instance, inquiring subsumption relationship between Endocarditis and Heartdisease relative to the GALEN ontology, CEL could correctly answer positively. Nevertheless, if the user wants to know more which axioms are responsible for this subsumption, there is nothing that CEL could do. By Definition 7 in Chapter 1, we introduce MOPS or minimal explanation which is a minimal sub-ontology supporting the subsumption. The minimal explanation exactly renders the reason why a given subsumption holds. As a result, axioms in a minimal explanation can be used to explain the subsumption in question. Therefore, pinpointing the *relevant axioms* is the first step towards explanation.

Example 15. Let's consider the example from the Chapter 1 again. The ontology \mathcal{O} with axiom tagging as shown in Figure 3.1:

For simplicity, we will tag the axioms in the terminology by natural numbers. As we have

0 :	Endocardium	\sqsubseteq	Tissue \sqcap $\exists cont_in$.HeartWall \sqcap $\exists cont_in$.HeartValve
1 :	HeartWall	\sqsubseteq	BodyWall \sqcap $\exists part_of$. Heart
2 :	HeartValve	\sqsubseteq	BodyValve \sqcap $\exists part_of$. Heart
3 :	Endocarditis	\sqsubseteq	Inflammation \sqcap $\exists has_loc$. Endocardium
4 :	Inflammation	\sqsubseteq	Disease \sqcap $\exists acts_on$. Tissue
5 :	Heartdisease	\equiv	Disease \sqcap $\exists has_loc$. Heart
6 :	Endocarditis	\sqsubseteq	Heartdisease
7 :	<i>part_of</i>	\sqsubseteq	<i>cont_in</i>
8 :	<i>has_loc</i> \circ <i>cont_in</i>	\sqsubseteq	<i>has_loc</i>

Figure 3.1: \mathcal{EL}^+ ontology with labelling

already seen that $\text{Endocarditis} \sqsubseteq_{\mathcal{O}} \text{Heartdisease}$, we can say \mathcal{O} is an explanation for this subsumption but not minimal. Furthermore, sub-ontologies $\{0, 1, 3, 4, 5, 7, 8\}$, $\{0, 2, 3, 4, 5, 7, 8\}$ and $\{6\}$ support this subsumption as well. All of these sub-ontologies are smaller than \mathcal{O} w.r.t set inclusion and indeed minimal in the sense that i) they are pairwise incomparable; ii) any proper subsets of them do not entail the subsumption at hand. \dashv

In the light of the awareness of explanation, we extend the \mathcal{EL}^+ subsumption algorithm in the previous chapter with labelling technique which had used in [5, 21]. As mentioned in the previous chapter, since the computation of implication sets is decidable, there is an obvious “brute-force” solution which tests the subsumption of two concepts with respect to all subsets of input ontology, and only give as output those minimal subsets from which the test is positive. In the following we describe a more strategic method of finding these MOPSs. The method is a combination of the polynomial \mathcal{EL}^+ algorithm [1] and the labelling techniques [5, 21], as we shall see below.

3.1 The Labelled Algorithm

We will extend the original algorithm by pinpointing the axioms so that we can trace back the relevant axioms (GCIs and CRIs) that are responsible for the inferred subsumption relationships. This extension is no longer tractable due to multiple (exponentially many) occurrences of concept in a subsumer set. First, we define the labelled concept name.

Definition 16. Let \mathcal{O} be an \mathcal{EL}^+ ontology over $CN_{\mathcal{O}}^{\top}$ and $RN_{\mathcal{O}}$. Then the set of labelled concept names (labelled pair of concept names) $CN_{\mathcal{O}^*}^{\top}$ ($RN_{\mathcal{O}^*}$) is defined as

$$CN_{\mathcal{O}^*}^{\top} = \{A^{label} \mid A \in CN_{\mathcal{O}}^{\top} \text{ and } label \subseteq \mathcal{O}\}$$

$$RN_{\mathcal{O}^*} = \{(A, B)^{label} \mid (A, B) \in RN_{\mathcal{O}} \text{ and } label \subseteq \mathcal{O}\}$$

◇

Now, we could extend the mapping S and R as follows:

- a mapping S assigning to each element of $CN_{\mathcal{O}}^{\top}$ a set of labelled concept names A^{label} and
- a mapping R assigning to each element of $RN_{\mathcal{O}}$ a set of labelled pairs of concept names $(A, B)^{label}$.

The intuition is that these mappings make implicit subsumption relationships explicit and at the same time collect necessary explaining information for those inferred subsumption relationships, i.e. minimal sets of axioms from which the subsumption follows. In fact, the computed labels S are the MOPSS or minimal explanations. Axiom pinpointing for subsumption is carried out incrementally while subsumption is being computed. After a successful termination of the computation, labels are MOPSS in the sense that

- $B^{label} \in S(A)$ implies $A \sqsubseteq_{label} B$, and for all $l \subset label$, $A \not\sqsubseteq_l B$ and
- $(A, B)^{label} \in R(r)$ implies $A \sqsubseteq_{label} \exists r.B$, and for all $l \subset label$, $A \not\sqsubseteq_l \exists r.B$

Extended Completion Rules	
R1	If $A^{l_1} \in S(X)$, $g = A \sqsubseteq B \in \mathcal{O}$, and $B^l \notin S(X)$ for all $l \subseteq l_1 \cup \{g\}$ then $S(X) := S(X) \setminus \bigcup_{l_1 \cup \{g\} \subset l} B^l \cup \{B^{l_1 \cup \{g\}}\}$
R2	If $A_1^{l_1}, A_2^{l_2} \in S(X)$, $g = A_1 \sqcap A_2 \sqsubseteq B \in \mathcal{O}$, and $B^l \notin S(X)$ for all $l \subseteq l_1 \cup \{g\}$ then $S(X) := S(X) \setminus \bigcup_{l_1 \cup l_2 \cup \{g\} \subset l} B^l \cup \{B^{l_1 \cup l_2 \cup \{g\}}\}$
R3	If $A^{l_1} \in S(X)$, $g = A \sqsubseteq \exists r.B \in \mathcal{O}$, and $(X, B)^l \notin R(r)$ for all $l \subseteq l_1 \cup \{g\}$ then $R(r) := R(r) \setminus \bigcup_{l_1 \cup \{g\} \subset l} (X, B)^l \cup \{(X, B)^{l_1 \cup \{g\}}\}$
R4	If $(X, Y)^{l_1} \in R(r)$, $A^{l_2} \in S(Y)$, $g = \exists r.A \sqsubseteq B \in \mathcal{O}$, and $B^l \notin S(X)$ for all $l \subseteq l_1 \cup l_2 \cup \{g\}$ then $S(X) := S(X) \setminus \bigcup_{l_1 \cup l_2 \cup \{g\} \subset l} B^l \cup \{B^{l_1 \cup l_2 \cup \{g\}}\}$
R5	If $(X, Y)^{l_1} \in R(r)$, $g = r \sqsubseteq s \in \mathcal{O}$, and $(X, Y)^l \notin R(s)$ for all $l \subseteq l_1 \cup \{g\}$ then $R(s) := R(s) \setminus \bigcup_{l_1 \cup \{g\} \subset l} (X, Y)^l \cup \{(X, Y)^{l_1 \cup \{g\}}\}$
R6	If $(X, Y)^{l_1} \in R(r)$, $(Y, Z)^{l_2} \in R(s)$, $g = r \circ s \sqsubseteq t \in \mathcal{O}$, and $(X, Z)^l \notin R(t)$ for all $l \subseteq l_1 \cup l_2 \cup \{g\}$ then $R(t) := R(t) \setminus \bigcup_{l_1 \cup l_2 \cup \{g\} \subset l} (X, Y)^l \cup \{(X, Z)^{l_1 \cup l_2 \cup \{g\}}\}$

Table 3.1: labelled subsumption algorithm for \mathcal{EL}^+

The algorithm works very much like before. The mappings are initialized by setting $S(A) = \{A^\emptyset, \top^\emptyset\}$ and $R(r) = \emptyset$ for all concept names and role names in the ontology, then it saturates both mappings by exhaustively applying the label-extended completion rules. The modified completion rules are shown in Table 3.1. The main differences are that here labels also determine applicability of rules and that the same pair of concept name can occur more than once under different labels. There are three possibilities of applying the extended completion rules in terms of labels. Consider **R1** for instance. In the first place, if there is no such concept name B presenting in the implication set $S(X)$, just directly add this concept name with current label $B^{l_1 \cup \{g\}}$ to $S(X)$. The other possible scenario is the follows. There exists a concept name B with a label l , l and $l_1 \cup \{g\}$ are set incomparable, i.e., neither is subset of the other. In this case, **R1** applies adjoining $B^{l_1 \cup \{g\}}$ in $S(X)$, though it already contains B^l , because we could explain a subsumption in different way. Thirdly, there exists a B^l and the new label is a proper subset of the existing one such like $l_1 \cup \{g\} \subset l$,

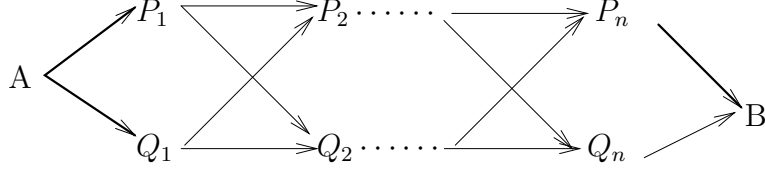


Figure 3.2: Example of exponential many explanations

this means the latest computing label is smaller than the one in the previous computation. Then replace this old labelled concept by the new one. In Table 3.1, g appearing in all rules denotes the tag of the axiom under consideration.

It is self-evident that this algorithm always terminate. In fact, this algorithm compute a set of labelled concept name $S(A)$ and a set of labelled pair of concept name $R(r)$. First, the concept names is finite, and the input ontology is a finite set of axioms. For the same concept name with different, incomparable labels, if the rule is applicable, the computed explanation sets always smaller than the one computed in the previous. As a result there are only a finite number of concept name accompanying explanation sets. Nevertheless, the tractability can not be obtained any more, as there could exponentially numbers of explanations.

Example 17. Consider the following ontology \mathcal{O} formulated in a sublogic of \mathcal{EL}^+ without existential restrictions.

$$\begin{aligned} \mathcal{O} := & \{P_i \sqsubseteq P_{i+1} \sqcap Q_{i+1} \mid 1 \leq i \leq n-1\} \cup \{Q_i \sqsubseteq P_{i+1} \sqcap Q_{i+1} \mid 1 \leq i \leq n-1\} \\ & \cup \{A \sqsubseteq P_1, A \sqsubseteq Q_1, P_n \sqsubseteq B, Q_n \sqsubseteq B\} \end{aligned}$$

For ease of presentation, we can view this ontology as an edge-directed graph as depicted in Figure 3.2. While nodes in the graph correspond to concepts, edges correspond to GCIs in normal form, i.e., $P_i \longrightarrow P_{i+1}$ represents the GCI $P_i \sqsubseteq P_{i+1}$. In this example, it is trivial to infer $A \sqsubseteq B$. From A there are two outgoing edges to P_1 and Q_1 , and every node P_i, Q_i ($1 \leq i \leq n-1$) there are two outgoing edges to their successors as well. Finally, we could

have 2^n different paths from A to B . These paths correspond to sets of axioms responsible for $A \sqsubseteq B$; therefore, there exists 2^n minimal explanations. \dashv

For the purpose of simplicity, in the following when we mention a particular implication set or role relation, which is clear from the context, we denote these by S and R respectively. In addition, we will emphasize the difference of the implication sets and role relations of the extended algorithm, and those of the original algorithm with the quantifier *labelled* (corresponding completion rule denoting by **R1** \cdots **R6**) and *unlabelled* (corresponding completion rule denoting by **CR1** \cdots **CR6**). For concept name, the labelled (unlabelled) implication set is a set of $CN_{\mathcal{O}^*}^\top$ ($CN_{\mathcal{O}}^\top$), for role name, the labelled (unlabelled) implication set is a set of $RN_{\mathcal{O}^*}$ (pair of concept names).

Definition 18. Let S be a labelled implication set, R a labelled role relation, and $\epsilon \subseteq \mathcal{O}$ a set of axioms. The ϵ -projections of S and R (for short, $\epsilon(S)$ and $\epsilon(R)$) are defined as follows:

$$\begin{aligned}\epsilon(S) &:= \{A|A^{exp} \in S \text{ and } exp \subseteq \epsilon\} \\ \epsilon(R) &:= \{(A, B)|(A, B)^{exp} \in R \text{ and } exp \subseteq \epsilon\}\end{aligned}$$

◇

Intuitively, $\epsilon(S(X))$ is a restriction of the labelled implication set to those subsumers of X w.r.t ϵ .

3.2 Soundness and Completeness

By introducing the following two lemmas, we show how application of a rule of the labelled algorithm to a labelled implication set S (respectively, role relation R) corresponds to application of a rule of the unlabelled algorithm to $\epsilon(S)$ (respectively, $\epsilon(R)$). Then we will prove the soundness of the labelled algorithm by reducing it to the unlabelled algorithm.

Lemma 19. *Let S, S' be labelled implication sets such that S' is obtained from S by application of the labelled completion rule **R1**, **R2**, or **R4** w.r.t. \mathcal{O} . Then we either have $\epsilon(S') = \epsilon(S)$, or $\epsilon(S')$ is obtained from $\epsilon(S)$ by application of the corresponding unlabelled completion rules **CR1**, **CR2**, **CR4**, w.r.t. ϵ .*

PROOF. We prove this lemma distinguishing with the labelled completion rule **R1**, **R2** and **R4**.

R1 Assume that **R1** is applied to $S = S(X)$ with $A^{exp} \in S(X)$. After this rule application, we obtain $S' = S'(X)$ from $S(X)$ by adding a new element $B^{exp'}$ with $exp' = exp \cup \{g\}$ and $g = A \sqsubseteq B \in \mathcal{O}$. If unlabelled completion rule **CR1** applicable to $\epsilon(S(X))$, there must be $A \in \epsilon(S(X))$ such that $A^{exp} \in S(X)$ with $exp \subseteq \epsilon$, a GCI $g = A \sqsubseteq B \in \epsilon$ and $B \notin \epsilon(S(X))$.

There are two possibilities for ϵ . First, consider the case where $exp' \not\subseteq \epsilon$. In this case, we have $\epsilon(S'(X)) = \epsilon(S(X))$. In fact, if $exp \not\subseteq \epsilon$, then by definition, A is not present in $\epsilon(S(X))$, unlabelled completion rule **CR1** w.r.t ϵ is not applicable, and this implying $\epsilon(S'(X)) = \epsilon(S(X))$. If $exp \subseteq \epsilon$, then $g \notin \epsilon$. As for the previous case, we have $\epsilon(S'(X)) = \epsilon(S(X))$, since B is not in $\epsilon(S'(X))$. Second, consider the case where $exp \subseteq exp' \subseteq \epsilon$. Since $S'(X)$ is obtained by extending $S(X)$ by $B^{exp'}$, we also know that B is contained in $\epsilon(S'(X))$. If there already exists $B^{exp''}$, in the case of $exp' \subset exp''$, we will remove $B^{exp''}$ by $B^{exp'}$ in S' , but we still have $\epsilon(S'(X)) = \epsilon(S(X))$ (otherwise if $exp'' \subseteq exp'$, **R1** is invalid for $S(X)$). Under other circumstances, $\epsilon(S'(X))$ is obtained from $\epsilon(S(X))$ by adding B as a result of the application of the unlabelled **CR1** rules.

R2 Assume that **R2** is applied to $S = S(X)$ with $A_1^{exp_1}, A_2^{exp_2} \in S(X)$. After this rule application, we obtain $S' = S'(X)$ from $S(X)$ by adding new element $B^{exp'}$ with $exp' = exp_1 \cup exp_2 \cup \{g\}$ and $g = A_1 \sqcap A_2 \sqsubseteq B \in \mathcal{O}$. If unlabelled completion rule **CR2** applied to $\epsilon(S(X))$, there must be $A_1 \in \epsilon(S(X))$ such that $A_1^{exp_1} \in S(X)$ with $exp_1 \subseteq \epsilon$ and $A_2 \in \epsilon(S(X))$ such that $A_2^{exp_2} \in S(X)$ with $exp_2 \subseteq \epsilon$, a GCI

$g = A_1 \sqcap A_2 \sqsubseteq B \in \epsilon$ and $B \notin \epsilon(S(X))$.

Similarly, there are two cases. In the case $exp' \not\subseteq \epsilon$, we have $\epsilon(S'(X)) = \epsilon(S(X))$. Since if $exp_1 \not\subseteq \epsilon$, by definition, A_1 is not present in $\epsilon(S(X))$, that implies $\epsilon(S'(X)) = \epsilon(S(X))$, if $exp_2 \not\subseteq \epsilon$, by definition, A_2 is not present in $\epsilon(S(X))$, that implies $\epsilon(S'(X)) = \epsilon(S(X))$, if both $exp_1, exp_2 \subseteq \epsilon$ then $g \not\subseteq \epsilon$, by definition, B is not present in $\epsilon(S(X))$, implies $\epsilon(S'(X)) = \epsilon(S(X))$. In the second case, when $exp' \subseteq \epsilon$, then we know that A_1, A_2 are in $\epsilon(S(X))$, and g is an axiom in ϵ . Since $S'(X)$ is obtained by extending $S(X)$ by $B^{exp'}$, we also know that B is contained in $\epsilon(S'(X))$. If there already exists $B^{exp''}$, in the situation of $exp' \subset exp''$, we replace exp'' by exp' in $S'(X)$ and $\epsilon(S'(X)) = \epsilon(S(X))$ still holds (otherwise if $exp'' \subseteq exp'$, **R2** is not applicable to $S(X)$). Otherwise, $\epsilon(S'(X))$ is obtained from $\epsilon(S(X))$ by adding B as a result of the application of the unlabelled **CR2** rule.

R4 Assume that **R4** is applied to $R = R(r)$ with $(X, Y)^{exp_1} \in R(r)$ and $S = S(X)$ with $A^{exp_2} \in S(Y)$. After this rule application, we get $S' = S'(X)$ from $S(X)$ by adding a new element $B^{exp'}$ with $exp' = exp_1 \cup exp_2 \cup \{g\}$ and $g = \exists r.A \sqsubseteq B \in \mathcal{O}$. If unlabelled completion rule **CR4** applied to $\epsilon(S(X))$, there must exist $(X, Y) \in \epsilon(R(r))$ with $(X, Y)^{exp_1} \in R(r)$ and $exp_1 \in \epsilon$, $A \in \epsilon(S(Y))$ with $A^{exp_2} \in S(Y)$ and $exp_2 \in \epsilon$, a GCI $g = \exists r.A \sqsubseteq B \in \epsilon$ and $B \notin \epsilon(S(X))$.

Similarly as what have shown above, when the case $exp' \not\subseteq \epsilon$, trivially implies $\epsilon(S'(X)) = \epsilon(S(X))$. We assume that $exp' \subseteq \epsilon$. Then we know that $A \in \epsilon(S(X))$ and $(X, Y) \in \epsilon(R(r))$, and g is an axiom in ϵ . Since $S'(X)$ is obtained by extending $S(X)$ of $B^{exp'}$, we also know that B contained in $\epsilon(S'(X))$. If there already exists $B^{exp''}$, for the case $exp' \subset exp''$, replace $B^{exp''}$ by $B^{exp'}$ and still hold $\epsilon(S'(X)) = \epsilon(S(X))$ (if $exp'' \subseteq exp'$, **R4** is invalid for $S(X)$). Otherwise, $\epsilon(S'(X))$ is obtained from $\epsilon(S(X))$ by adding B as a result of the application of the unlabelled **CR4** rule. ■

Lemma 20. *Let R, R' be labelled role relations such that R' is obtained from R by application of the completion rule **R3**, **R5**, or **R6**. Then we either have $\epsilon(R') = \epsilon(R)$, or $\epsilon(R')$ is*

obtained from $\epsilon(R)$ by application of the corresponding unlabelled completion rules **CR3**, **CR5**, **CR6**, w.r.t, ϵ .

PROOF. The proof is similar to the Lemma 19, we distinguish the cases w.r.t **R3**, **R5** and **R6**.

R3 Assume that **R3** is applied to $R = R(r)$ with $A^{exp} \in S(X)$. After this rule application, we obtain $R' = R'(r)$ from $R(r)$ by adding a new binary relation $(X, B)^{exp'}$ with $exp' = exp \cup \{g\}$ and $g = A \sqsubseteq \exists r. B \in \mathcal{O}$. If the unlabelled completion **CR3** applicable, there should be a $A \in \epsilon(S(A))$ with $A^{exp} \in S(A)$ and $exp \subseteq \epsilon$, a GCI $g = A \sqsubseteq \exists r. B \in \epsilon$ and $(X, B) \notin \epsilon(R(r))$.

There are two possibilities for ϵ . In the case where $exp' \not\subseteq \epsilon$. Then we have $exp \not\subseteq \epsilon$, by definition of ϵ projection, A is not present in $\epsilon(S(X))$, nothing changed for $\epsilon(R(r))$, implies $\epsilon(R'(r)) = \epsilon(R(r))$. If $exp \subseteq \epsilon$, then $g \not\subseteq \epsilon$, not satisfied the precondition of **R3**, **R3** is not applicable, implies $\epsilon(R'(r)) = \epsilon(R(r))$. In the case where $exp' \subseteq \epsilon$. Since $R'(r)$ is obtained by extending $R(r)$ with $(X, B)^{exp'}$, we also know (X, B) is contained in $\epsilon(R'(r))$. If there already exists $(X, B)^{exp''}$, in the case of $exp' \subset exp''$, replace $(X, B)^{exp''}$ by $(X, B)^{exp'}$ and $\epsilon(R'(r)) = \epsilon(R(r))$ still holds (if $exp'' \subseteq exp'$, unfit the precondition of **R3**,). Otherwise, $\epsilon(R'(r))$ is obtained from $\epsilon(R(r))$ by adding (X, B) as a result of the application of the unlabelled **CR3** rule.

R5 Assume that **R5** is applied to $R = R(s)$ with $(X, Y)^{exp} \in R(r)$. After application of this rule, we got $R' = R'(s)$ from $R(s)$ by adding a new binary relation $(X, Y)^{exp'}$ with $exp' = exp \cup \{g\}$ and $g = r \sqsubseteq s \in \mathcal{O}$. If **CR5** applicable $\epsilon(R(s))$, there exists $(X, Y) \in \epsilon(R(r))$ with $(X, Y)^{exp} \in R(r)$ and $exp \subseteq \epsilon$, a GCI $g = r \sqsubseteq r \in \epsilon$ and $(X, Y) \notin \epsilon(R(s))$.

For the case where $exp' \not\subseteq \epsilon$, if $exp \not\subseteq \epsilon$, by definition of ϵ , (X, Y) is not present in $\epsilon(R(r))$, implies $\epsilon(R'(r)) = \epsilon(R(r))$, if $g \not\subseteq \epsilon$, **CR5** is not applicable w.r.t ϵ , nothing and we have $\epsilon(R'(r)) = \epsilon(R(r))$. Considering the case where $exp' \subseteq \epsilon$. Since $R'(s)$ is obtained by extending $R(s)$ of $(X, Y)^{exp'}$, we also know that (X, Y) is included in $\epsilon(R'(s))$. If there another $(X, Y)^{exp''}$, if $exp' \subset exp''$, still give rise to $\epsilon(R'(r)) =$

$\epsilon(R(r))$ (for otherwise, if $exp'' \not\subseteq exp'$, **R5** is not applicable). Otherwise, $\epsilon(R'(s))$ is obtained from $\epsilon(R(s))$ by adding (X, Y) as a result of the application of the unlabelled **CR5** rule.

R6 Assume that **R6** is applied to $R = R(t)$ with $(X, Y)^{exp_1} \in R(r)$, $(Y, Z)^{exp_2} \in R(s)$. After the rule application, we have $R' = R'(t)$ from $R(t)$ by adding new binary relation $(X, Z)^{exp'}$ with $exp' \subseteq exp_1 \cup exp_2 \cup \{g\}$ and $g = r \circ s \sqsubseteq t \in \mathcal{O}$. If **CR6** is applicable, there exists $(X, Y) \in \epsilon(R(r))$ and $(Y, Z) \in \epsilon(R(s))$, a GCI $g = r \circ s \sqsubseteq t \in \epsilon$ and $(X, Z) \notin \epsilon(R(t))$.

Similarly, the case where $exp' \not\subseteq \epsilon$ trivially implies $\epsilon(R'(t)) = \epsilon(R(t))$. Assume $exp' \subseteq \epsilon$, then we know $(X, Y) \in \epsilon(R(s))$ and $(Y, Z) \in \epsilon(R(t))$, because of $R'(t)$ is generated from $R(t)$ by adding $(X, Z)^{exp'}$, we know (X, Z) is also contained in $\epsilon(R'(t))$. If there already exists $(X, Z)^{exp''}$ for some $exp'' \not\subseteq exp'$, if $exp' \subset exp''$, similar to the previous case, we have $\epsilon(R'(t)) = \epsilon(R(t))$. Otherwise, $\epsilon(R'(t))$ is obtained from $\epsilon(R(t))$ by adding (X, Z) as a result of the application of the unlabelled **CR6** rule.

■

Lemma 21. (Completeness) *Let S and R be a labelled implication set and a labelled role relation, respectively, to which none of the completion rules of the labeled classification algorithm in Table 3.1 applies. Then none of the original completion rules of the unlabeled classification algorithm in Table 2.1 applies to $\epsilon(S)$ and $\epsilon(R)$, respectively.*

PROOF. We will distinguish the case for every completion rule.

CR1 For an element $A \in \epsilon(S)$, and a GCI $g = A \sqsubseteq B \in \epsilon$ (For otherwise, ϵ does not contain g , this (unlabeled)rule does not apply). We show that this (unlabeled)rule cannot be applied. Since A is present in $\epsilon(S)$, by definition of ϵ , there must be a label $exp \subseteq \epsilon$ with $A^{exp} \in S$. Completeness of S implies that the (labeled)rule **R1** is not applicable to A^{exp} in S . Since the first two preconditions of **R1** are fulfilled, this can only mean that there exists an element $B^{exp''}$ in S for some $exp'' \subseteq exp \cup \{g\}$. However, $exp \subseteq \epsilon$

and $g \in \epsilon$ implies $exp'' \subseteq \epsilon$. By definition, we know that B is present in $\epsilon(S)$, which shows that the (unlabeled) rule **CR1** is not applicable to A in $\epsilon(S)$.

CR2 There are two element A_1, A_2 in $\epsilon(S)$, and a GCI $g = A_1 \sqcap A_2 \sqsubseteq B \in \epsilon$. For both A_1 and A_2 are present in $\epsilon(S)$, there must be two labels $exp_1 \subseteq \epsilon, exp_2 \subseteq \epsilon$ with $A_1^{exp_1} \in S$ and $A_2^{exp_2} \in S$. Completeness of S implies that the (labeled)rule **R2** is unavailable to $A_1^{exp_1}, A_2^{exp_2}$ in S . Since the first two preconditions of **R2** are fulfilled, this means that there exists an element $B^{exp''}$ in S for some $exp'' \subseteq exp_1 \cup exp_2 \cup \{g\}$. But $exp_1 \subseteq \epsilon, exp_2 \subseteq \epsilon$ and $g \in \epsilon$ implies that $exp'' \in \epsilon$. By definition, we know that B is present in $\epsilon(S)$, which shows that the (unlabeled)rule **CR2** is not applicable A_1 and A_2 in $\epsilon(S)$.

CR4 There is binary relation $(X, Y) \in \epsilon(R(r))$ and an element $A \in \epsilon(S(Y))$, and a GCI $g = \exists r.A \sqsubseteq B \in \epsilon$, we show that **CR4** cannot be applied to $\epsilon(S(X))$. Since (X, Y) is present in $\epsilon(R(r))$, there must be a label $exp_1 \subseteq \epsilon$ with $(X, Y)^{exp_1} \in R(r)$, similarly, A is present in $S(Y)$, there must be a label $exp_2 \subseteq \epsilon$ with $A^{exp_2} \in S(Y)$. Completeness of S implies that the (label)rule **R4** is not applicable. Since the first two preconditions of **R1** are satisfied, only if the possibility that there exists an element $B^{exp''}$ in $S(X)$ with $exp'' \subseteq exp_1 \cup exp_2 \cup \{g\}$. $exp_1 \subseteq \epsilon, exp_2 \subseteq \epsilon$ and $g \in \epsilon$ implies $exp'' \subseteq \epsilon$, then we know that B is present in $\epsilon(S(X))$, which shows that the (unlabeled)rule **CR4** is not applicable to B in $\epsilon(S(X))$.

CR3 There is an element $A \in \epsilon(S(X))$ and a GCI $g = A \sqsubseteq \exists r.B \in \epsilon$. Since A is present in $\epsilon(S(X))$, there must be a label $exp_1 \subseteq \epsilon$ with $A^{exp_1} \in S(X)$. Completeness of S implies that the (label)rule **R3** is not applicable. The satisfiability of the first two preconditions conjecture that there exists a binary relation $(X, B)^{exp''}$ in $R(r)$ with $exp'' \subseteq exp_1 \cup \{g\}$. Since both $exp_1 \subseteq \epsilon$ and $g \in \epsilon$, we obtain $exp'' \subseteq \epsilon$. By definition of $\epsilon(R)$, we know that (X, B) is included in $\epsilon(R(r))$, which show that the (unlabelled)rule **CR3** cannot be applied to (X, B) for $\epsilon(R(r))$.

CR5 There is an binary relation $(X, Y) \in \epsilon(R(r))$ and a CGI $g = r \sqsubseteq s \in \epsilon$. Since (X, Y)

is present in $\epsilon(R(r))$, there must be a label $exp \subseteq \epsilon$ with $(X, Y)^{exp} \in R(r)$. Since the (label)rule **R5** is not applicable. The first two preconditions of **R5** are fulfilled, this means there exists an binary relation $(X, Y)^{exp''}$ in $R(s)$ with $exp'' \subseteq exp \cup \{g\}$. We can obtain $exp'' \subseteq \epsilon$ by $exp \subseteq \epsilon$ and $g \in \epsilon$, from the definition, we know that (X, Y) are in $\epsilon(R(s))$, it means the (unlabelled)rule **CR5** is not applicable to (X, Y) for $\epsilon(R(s))$.

CR6 There is (X, Y) in $\epsilon(R(r))$, (Y, Z) in $\epsilon(R(s))$ and a GCI $g = r \circ s \sqsubseteq t \in \epsilon$. Since (X, Y) is present in $\epsilon(R(r))$, there must be a label $exp_1 \subseteq \epsilon$ with $(X, Y)^{exp_1} \in R(r)$, similarly, there must be a label $exp_2 \subseteq \epsilon$ with $(Y, Z)^{exp_2} \in R(s)$. We know the (label)rule **R6** is not applicable, but the precondition of first two of **R6** is fulfilled, this mean that there exists an binary relation $(X, Z)^{exp''}$ in $R(t)$ with $exp'' \subseteq exp_1 \cup exp_2 \cup \{g\}$. Since $exp_1 \subseteq \epsilon$, $exp_2 \subseteq \epsilon$ and $g \in \epsilon$, implies $exp'' \subseteq \epsilon$. From the definition, we know that (X, Z) is present in $\epsilon(R(t))$ and the (unlabelled)rule **CR6** is not applicable to (X, Z) for $\epsilon(R(t))$. ■

Now let us prove the soundness of the extended subsumption algorithm for \mathcal{EL}^+ .

Theorem 22. (Soundness) *Given a normalized ontology \mathcal{O} , let S be the mapping obtained after the application of the rules in Table 3.1 for \mathcal{O} has terminated, and let A, B be concept names occurring in \mathcal{O} . Then, the following holds:*

1. *If $B^{exp} \in S(A)$, then $A \sqsubseteq_{exp} B$.*
2. *If $A \sqsubseteq_{\mathcal{O}'} B$ for $\mathcal{O}' \subseteq \mathcal{O}$, then there is a $B^{exp} \in S(A)$ with $exp \subseteq \mathcal{O}'$.*
3. *If $\{B^{exp_1}, \dots, B^{exp_k}\} \subseteq S(A)$, then all exp_i are “ \subseteq ”-incomparable.*

PROOF. Lemma 19 and Lemma 20 show that, for each application of a (labeled) completion rule, there is a corresponding application of a (unlabeled) completion rule w.r.t a subset

$\epsilon \subseteq \mathcal{O}$. Assume that B^{exp} is in $S(A)$, we set $\epsilon := exp$. Since $exp' \subseteq \epsilon$ for all internal labels exp' relevant to get B^{exp} in $S(A)$, we know that there is a corresponding sequence of rule applications in the original classification algorithm w.r.t. ϵ , adding B to $\epsilon(S(A))$. By the soundness of the unlabeled algorithm, it follows that $A \sqsubseteq_{exp} B$.

For point 2, we will show the contrapositive. Given a subset $\mathcal{O}' \subseteq \mathcal{O}$, we assume that there is no such B^{exp} in $S(A)$ with $exp \subseteq \mathcal{O}'$ after the termination of the (labeled) classification algorithm. By Lemma 21, the original classification algorithm has also terminated with no more (unlabeled) rules applicable to $\epsilon(S)$ and $\epsilon(R)$. In particular, $\mathcal{O}'(S(A))$ is complete. By definition, $B^{exp} \notin S(A)$ implies $B \notin \mathcal{O}'(S(A))$. This together with the completeness of the unlabeled classification algorithm shows that $A \not\sqsubseteq_{\mathcal{O}'} B$.

For point 3, it suffices to show that each application of rule **R1**, **R2**, **R4** (respectively **R3**, **R5** or **R6**) adds a new entry B^{exp} to $S(X)$ (respectively, $(A, B)^{exp}$ to $R(r)$) such that exp is neither subset nor superset of exp' for all $B^{exp'} \in S(X)$ ($(A, B)^{exp'} \in R(r)$).

We are done if $S(X)$ ($R(r)$) does not contain such an entry, otherwise let's fix an arbitrary exp' labeling a corresponding entry. It is immediate by the definition of the extended completion rules that $exp' \not\subseteq exp$. It remains to be shown that $exp \not\subseteq exp'$, fortunately, it is immediate by the definition of the labeled completion rules as well, since for every completion rule, if there is a exp' such that $exp \subset exp'$, in order to keep the minimality, we replace the exp' by the relative small explanation exp and $exp \not\subseteq exp'$. ■

3.3 Minimal Explanation w.r.t Original Ontology

In the previous sections, we have already shown the labelled subsumption algorithm for \mathcal{EL}^+ . At the first glance, this algorithm yields the classifications for every concept name A with an implication set $S(A) = \{B^{exp_1}, B^{exp_2}, C^{exp_3}, \dots\}$, in which B and C are the super concept names of A , and the labels exp_1, exp_2, exp_3 are the minimal explanations of corresponding subsumption relationships. Nevertheless, this algorithm runs on a normalized \mathcal{EL}^+ ontology. As a result, the computed explanations are relative to axioms in normal form as well. From the application stand point, these explanations are not immediately informative. On one

Original ontology	GCI normalized ontology
a: $A \sqsubseteq B \sqcap C$	0: $A \sqsubseteq B$ (a,d)
b: $B \sqsubseteq C$	1: $A \sqsubseteq C$ (a,d)
c: $A \sqcap C \sqsubseteq D$	2: $B \sqsubseteq C$ (b)
d: $A \sqsubseteq B \sqcap C \sqcap E$	3: $A \sqcap C \sqsubseteq D$ (c)
	4: $A \sqsubseteq E$ (d)

Table 3.2: reverse mapping causes non-minimality

hand, most realistic ontologies are not in our normal form. On the other hand, the user is not likely to be able to comprehend the explanations equipped with normalized axioms which consist of newly introduced concept names. The explanations relative to axioms from the original ontology would be make more sense.

This section concerns the following problem: Given minimal explanations w.r.t normalized axioms, find minimal explanations w.r.t original axioms. Firstly, we tag the original axioms. It is not difficult to keep these tags under surveillance when normalization. The normalized axioms are mapped to the generating original axioms. This mapping may not be functional since multiple original axioms may have generated the same axiom in normal form. Since these axioms could be mapped back to original axioms we could obtain the minimal explanations w.r.t original form as well.

Example 23. Let us look at a simple example as in Table 3.2. For the sake of clarity and simplicity, we will tag the original axioms with letters (i.e. a, b, c, ...) and normalized ontology with natural number (i.e. 0, 1, 2 ...) respectively. The letters behind each normalized axiom are reverse reference to the respective original axioms that generate it. Note that in principle there can be multiple original axioms that generate the same normalized one, hence a set of here. These letters are read disjunctively, i.e. any of them can generate this normalized axiom. The complete implication set of A computed by the extended algorithm is

$$S(A) = \{A^\emptyset, B^{\{0\}}, C^{\{1\}}, C^{\{0,2\}}, D^{\{1,3\}}, D^{\{0,2,3\}}, E^{\{4\}}, \top^\emptyset\}$$

The axiom tags in S are from the normalized ontology. For example, consider both $D^{\{0,2,3\}}$ and $D^{\{1,3\}}$ in $S(A)$. Since either $A \sqsubseteq_{\{0,2,3\}} D$ or $A \sqsubseteq_{\{1,3\}} D$, both $\{0, 2, 3\}$ and $\{1, 3\}$ are normalized MOPSs for subsumption $A \sqsubseteq D$. When using the original ontology to explain this subsumption, we could obtain four original explanations by reverse mapping to original ontology, either $A \sqsubseteq_{\{a,b,c\}} D$ and $A \sqsubseteq_{\{b,d,c\}} D$ from the minimal normalized explanation set $\{0, 2, 3\}$, or $A \sqsubseteq_{\{a,c\}} D$ and $A \sqsubseteq_{\{d,c\}} D$ from the normalized MOPS $\{1, 3\}$. When checking the minimality of these explanations, the explanations $\{a, b, c\}$, $\{b, d, c\}$ are not minimal w.r.t the original ontology since we can remove the axiom $b (B \sqsubseteq C)$ and this subsumption still holds. We can thus conclude from the example above that direct reverse mapping from normalized axioms to original axioms results in the non-minimality explanations w.r.t original axioms. \dashv

In the labeled algorithm, we know which facts contribute to a particular subsumption by MOPSs. In fact, we could view our tags as Boolean propositional variables, and regard explanations as “monotonic” Boolean formula built from these variables, that is, propositional formula built from the variables by using conjunction and disjunction only. Given a minimal explanation in normal form with tags ψ_1, \dots, ψ_n , the subsumption hold if the Boolean formula $\psi_1 \wedge \dots \wedge \psi_n$ evaluates to “true”. Since the same normalized axiom in the minimal explanation may come more than one way from original ontology, we also obtain disjunctions in tags. In order to compute the minimal explanation w.r.t the original ontology, we viewed the tags of original axioms as $\phi_{i,1}, \dots, \phi_{i,k_i}$ by reverse mapping from a fact ψ_i in the minimal explanation in normal form and these propositional variables was read disjunctively. Thus, the tags in the original axioms can be used to describe which of the original axioms in the input ontology \mathcal{O} are responsible for a subsumption problem. In fact, MOPS w.r.t original ontology directly correspond to minimal valuations satisfying the formula $\bigwedge_{i=1}^n \bigvee_{j=1}^{k_i} \phi_{i,j}$. In the following, we will probe how hard it is to decide the existence of an explanation of a certain size w.r.t the original ontology.

3.4 Complexity

It is known that the problem of finding minimal valuations that satisfies a monotonic Boolean formula is **NP-complete**. Since ϕ is in conjunctive normal form, this is just the well-known problem of finding minimal hitting sets by Reiter [10, 19]. In this section, we will prove that to decide the existence of an explanation w.r.t original ontology whose size is at most n ($n \in \mathbb{N}$) for which subsumption holds in \mathcal{EL}^+ is **NP-complete**.

Hardness is shown by a reduction from the problem of deciding the existence of a minimal valuation of size $\leq n$ satisfying a monotonic Boolean formula. Next, we define the monotonic Boolean formula and the valuation satisfying a monotonic Boolean formula.

Definition 24. Let \mathbf{P} be a set of propositional Boolean variables. A monotonic Boolean formula ϕ is built from \mathbf{P} and used conjunction and disjunction only. ϕ is in *conjunctive normal form* (CNF) iff it is of the form $(P_{11} \vee \dots \vee P_{1l_1}) \wedge \dots \wedge (P_{k1} \vee \dots \vee P_{kl_k})$, where $P_{ij} \in \mathbf{P}$. \diamond

Definition 25. Let ϕ be a monotonic Boolean formula over a set of proposition variables \mathbf{P} . A valuation satisfying ϕ is a subset $\mathbf{VAL} \subseteq \mathbf{P}$ that evaluates ϕ to “true” ($\mathbf{VAL} \models \phi$). A valuation \mathbf{V} is *minimal* if $\mathbf{V} \models \phi$ and $\mathbf{W} \not\models \phi$ for all $\mathbf{W} \subset \mathbf{V}$. \diamond

Example 26. As an example, consider the following Boolean monotonic formula ϕ over Boolean propositional variable $\mathbf{P} = \{P_1, P_2, P_3, P_4, P_5\}$

$$\phi = (P_1 \vee P_2) \wedge (P_3 \vee P_4 \vee P_2) \wedge P_5$$

Both $\{P_1, P_2, P_5\}$ and $\{P_1, P_3, P_5\}$ satisfy ϕ , but $\{P_1, P_2, P_5\}$ is not minimal since by removing P_1 , ϕ is still evaluated to “true”. In this example, the minimal valuations satisfying ϕ are $\{P_1, P_3, P_5\}$, $\{P_1, P_4, P_5\}$ and $\{P_2, P_5\}$ \dashv

Given a monotonic Boolean formula ϕ , we can now construct an \mathcal{EL}^+ terminology \mathcal{T} such that the existence of minimal explanations corresponds to that of minimal valuations.

Definition 27. Let $\phi = (P_{11} \vee P_{12} \vee \dots \vee P_{1l_1}) \wedge \dots \wedge (P_{k1} \vee P_{k2} \vee \dots \vee P_{kl_k})$ be a monotonic Boolean formula in conjunctive normal form, consisting of k disjunctive clauses over \mathbf{P} . We define \mathcal{EL}^+ terminology $\mathcal{T} = \hat{\mathcal{O}}_\phi \cup \mathcal{O}_\phi$ as follows:

- $\hat{\mathcal{O}}_\phi = \{P_{ij} \sqsubseteq Q_i \mid 1 \leq i \leq k \text{ and } 1 \leq j \leq l_i\} \cup \{Q_1 \sqcap \dots \sqcap Q_k \sqsubseteq B\}$ where P_{ij} is the j -th propositional variable appearing in the disjunctive clause i , and $B, Q_i \notin \mathbf{P}$.
- $\mathcal{O}_\phi = \{A \sqsubseteq P\}$ where $P \in \mathbf{P}$ and $A \notin \mathbf{P}$.

◇

The terminology \mathcal{T} consists of two parts $\hat{\mathcal{O}}_\phi$ and \mathcal{O}_ϕ . Both of which have size polynomial of the size of ϕ . To reduce the problem of minimal valuation satisfying a monotonic Boolean formula, we first show the following equivalence.

Lemma 28. *Let ϕ be a monotonic Boolean formula in conjunctive normal form, and A, B be fresh concept names not occurring propositional letter in ϕ , then the following are equivalent:*

1. *there is a minimal valuation of size at most n satisfying the formula ϕ .*
2. *there is a sub TBox $\mathcal{O}'_\phi \subseteq \mathcal{O}_\phi$ of size at most n such that $A \sqsubseteq_{\hat{\mathcal{O}}_\phi \cup \mathcal{O}'_\phi} B$.*

PROOF.

(1 \Rightarrow 2) According to the assumption, $\phi = (P_{11} \vee \dots \vee P_{1l_1}) \wedge \dots \wedge (P_{k1} \vee \dots \vee P_{kl_k})$ over \mathbf{P} , and a minimal valuation \mathbf{VAL} satisfying ϕ and $|\mathbf{VAL}| \leq n$. We define $\mathcal{O}'_\phi = \{A \sqsubseteq P \mid P \in \mathbf{VAL}\}$. Since $\mathbf{VAL} \subseteq \mathbf{P}$, $\mathcal{O}'_\phi \subseteq \mathcal{O}_\phi$, and $|\mathbf{VAL}|$ is at most n , then \mathcal{O}'_ϕ of size at most n . \mathbf{VAL} is the minimal valuation satisfying ϕ , by definition 25, there at least one P_{ij} was presented for every disjunctive clause i of ϕ where $1 \leq i \leq k$. Then $A \sqsubseteq Q_i$ for $1 \leq i \leq k$. By the GCI $Q_1 \sqcap \dots \sqcap Q_k \sqsubseteq B$ it is trivial $A \sqsubseteq_{\hat{\mathcal{O}}_\phi \cup \mathcal{O}'_\phi} B$.

(2 \Rightarrow 1) From the assumption, $A \sqsubseteq_{\hat{\mathcal{O}}_\phi \cup \mathcal{O}'_\phi} B$ and $|\mathcal{O}'_\phi| \leq n$. We define $\mathbf{VAL} = \{P \mid A \sqsubseteq P \in \mathcal{O}'_\phi\}$. Since \mathcal{O}'_ϕ is of size at most n , \mathbf{VAL} is also of size at most n . $A \sqsubseteq B$ then $A \sqsubseteq Q_i$ for all $1 \leq i \leq k$. \mathbf{VAL} satisfies every disjunctive clause i of ϕ and satisfies ϕ .

■

This theorem shows that the problem of finding sub TBoxes of size smaller or equal to n is **NP-hard**, in the presence of an irrefutable TBox, i.e., $\hat{\mathcal{O}}_\phi$. We will get rid of the irrefutable part and obtain the same hardness result.

Lemma 29. *There is a TBox $\mathcal{O}' \subseteq \hat{\mathcal{O}}_\phi \cup \mathcal{O}_\phi$ of the size at most $n+k+1$ such that $A \sqsubseteq_{\mathcal{O}'} B$ iff there is a sub TBox $\mathcal{O}'_\phi \subseteq \mathcal{O}_\phi$ of size at most n such that $A \sqsubseteq_{\hat{\mathcal{O}}_\phi \cup \mathcal{O}'_\phi} B$.*

PROOF. We will prove by both directions.

(\Rightarrow) According to the assumption, since $A \sqsubseteq_{\mathcal{O}'} B$, then the GCI $Q_1 \sqcap \dots \sqcap Q_k \sqsubseteq B$ must be present in \mathcal{O}' , otherwise, the interpretation containing all concept names appearing in $\hat{\mathcal{O}}_\phi$ and \mathcal{O}_ϕ except B would be a model for \mathcal{O}' but $A \not\sqsubseteq B$, in which contradicting hypothesis. For instance, given a model I such that $A^I = \Delta^{\mathcal{I}}$, $P_{ij}^I = \Delta^{\mathcal{I}}$, $Q_i^I = \Delta^{\mathcal{I}}$ and $B^I = \emptyset$, since B not in \mathcal{O}' , I is a model of \mathcal{O}' but not $A \sqsubseteq B$. Similarly, it must be the case that $A \sqsubseteq_{\mathcal{O}'} Q_i$ for each $1 \leq i \leq k$ as well. Thus, for every $1 \leq i \leq k$ there is a GCI $P_{ij} \sqsubseteq Q_i$ in \mathcal{O}' . From the above, it implies that $|\mathcal{O}' \cap \hat{\mathcal{O}}_\phi| \geq k+1$. Now we can construct such sub TBox \mathcal{O}'_ϕ . Define $\mathcal{O}'_\phi = \mathcal{O}' \setminus \hat{\mathcal{O}}_\phi$, then, if \mathcal{O}' is of size at most $n+k+1$, \mathcal{O}'_ϕ is of size at most n , and $\mathcal{O}'_\phi \subseteq \mathcal{O}_\phi$. Furthermore, $\mathcal{O}' \subseteq \hat{\mathcal{O}}_\phi \cup (\mathcal{O}' \setminus \hat{\mathcal{O}}_\phi) = \hat{\mathcal{O}}_\phi \cup \mathcal{O}'_\phi$, and hence, since $A \sqsubseteq_{\mathcal{O}'} B$, it holds also that $A \sqsubseteq_{\hat{\mathcal{O}}_\phi \cup \mathcal{O}'_\phi} B$.

(\Leftarrow) By assumption, there is a sub TBox \mathcal{O}'_ϕ of \mathcal{O}_ϕ such that $A \sqsubseteq_{\hat{\mathcal{O}}_\phi \cup \mathcal{O}'_\phi} B$, we now going to construct a \mathcal{O}' such that $\mathcal{O}' \subseteq \hat{\mathcal{O}}_\phi \cup \mathcal{O}'_\phi$ and $A \sqsubseteq_{\mathcal{O}'} B$. In order obtain $A \sqsubseteq B$, the GCI $Q_1 \sqcap \dots \sqcap Q_k \sqsubseteq B$ must be present in \mathcal{O}' , and for every $1 \leq i \leq k$, it is enough to have with just one GCI $P_{ij} \sqsubseteq Q_i$ for every i with $A \sqsubseteq P_{ij} \in \mathcal{O}'_\phi$ to get the subsumption. Let define the smallest set $\hat{\mathcal{O}}'$ such that 1) $P \sqsubseteq Q_i \in \hat{\mathcal{O}}'$ if $A \sqsubseteq P \in \mathcal{O}'_\phi$ and 2) $P' \sqsubseteq Q_i \notin \hat{\mathcal{O}}'$ if $P \sqsubseteq Q_i \in \hat{\mathcal{O}}'$ and $P' \sqsubseteq Q_i$. This $\hat{\mathcal{O}}'$ guarantee there only one P accompany for every Q_i where $1 \leq i \leq k$. Now, we can define $\mathcal{O}' = \mathcal{O}'_\phi \cup \hat{\mathcal{O}}' \cup \{Q_1 \sqcap \dots \sqcap Q_k \sqsubseteq B\}$, if \mathcal{O}'_ϕ is of size at most n , \mathcal{O}' is of size at most $n+k+1$. It is trivial to see that $\mathcal{O}' \subseteq \hat{\mathcal{O}}_\phi \cup \mathcal{O}'_\phi$, since $\hat{\mathcal{O}}' \cup \{Q_1 \sqcap \dots \sqcap Q_k \sqsubseteq B\} \subseteq \hat{\mathcal{O}}_\phi$, $\mathcal{O}'_\phi \subseteq \mathcal{O}_\phi$, and trivially $A \sqsubseteq_{\mathcal{O}'} B$.



With the help of Lemma 28 and Lemma 29, we can prove the Theorem 30 that the hardness of the problem does not depend on the fact that a part of the TBox is irrefutable. Since finding the minimal valuation that make a monotonic Boolean formula “true” is **NP-hard**, each of the solution of the latter problem must also be **NP-hard**.

Theorem 30. *The problem of deciding the existence of a sub TBox of size at most n for which subsumption holds in \mathcal{EL}^+ is **NP-hard**.*

It is not hard to see that the decision problem is in NP, given that subsumption is decidable in polynomial time [1]. In fact, we can i) guess a subontology $\mathcal{O}' \subseteq \mathcal{O}$ with $|\mathcal{O}'| \leq n$ in time polynomial and then ii) check in polynomial time whether \mathcal{O}' entails the subsumption in question. Hence, the NP completeness is obtained.

Theorem 31. *The problem of deciding the existence of a sub TBox of size at most n for which subsumption holds in \mathcal{EL}^+ is **NP-complete**.*

Chapter 4

A Tractable Restriction of Axiom Pinpointing in \mathcal{EL}^+

We have proved that the extended algorithm presented in the previous chapter is inherently intractable. It computes all MOPSs which are exponentially many in the worst case as demonstrated by Example 17. In some applications, however, not all explanations are required. Now we introduce a revised labelled algorithm by strengthening the preconditions of the completion rules, so that the algorithm computes and retains only one explanation set for each subsumption relationship, which in turn reduces the complexity. In this chapter, we first introduce the modified completion rules for tractable explanation algorithm and then present a refined version based on the queue techniques used in the CEL reasoner [3] of this algorithm. Then we discuss a first unoptimized implementation, and finally we show that one explanation can be computed in polynomial time.

4.1 Computing One Explanation

In order to obtain one explanation for each subsumption relationship, we modify the preconditions of the completion rules from the ones given in Chapter 3 (see Table 4.1). Compared to the standard labelled algorithm, the preconditions here are stronger in the sense that they admit less number of rule applications. In fact, the number of rule application is bounded by a polynomial of the input size as we shall see later in this chapter.

Completion Rules	
R1	If $A^{l_1} \in S(X)$, $g = A \sqsubseteq B \in \mathcal{O}$, and $B^l \notin S(X)$ for some l then $S(X) := S(X) \cup \{B^{l_1 \cup \{g\}}\}$
R2	If $A_1^{l_1}, A_2^{l_2} \in S(X)$, $g = A_1 \sqcap A_2 \sqsubseteq B \in \mathcal{O}$, and $B^l \notin S(X)$ for some l then $S(X) := S(X) \cup \{B^{l_1 \cup l_2 \cup \{g\}}\}$
R3	If $A^{l_1} \in S(X)$, $g = A \sqsubseteq \exists r.B \in \mathcal{O}$, and $(X, B)^l \notin R(r)$ for some l then $R(r) := R(r) \cup \{(X, B)^{l_1 \cup \{g\}}\}$
R4	If $(X, Y)^{l_1} \in R(r)$, $A^{l_2} \in S(Y)$, $g = \exists r.A \sqsubseteq B \in \mathcal{O}$, and $B^l \notin S(X)$ for some l then $S(X) := S(X) \cup \{B^{l_1 \cup l_2 \cup \{g\}}\}$
R5	If $(X, Y)^{l_1} \in R(r)$, $g = r \sqsubseteq s \in \mathcal{O}$, and $(X, Y)^l \notin R(s)$ for some l then $R(s) := R(s) \cup \{(X, Y)^{l_1 \cup \{g\}}\}$
R6	If $(X, Y)^{l_1} \in R(r)$, $(Y, Z)^{l_2} \in R(s)$, $g = r \circ s \sqsubseteq t \in \mathcal{O}$, and $(X, Z)^l \notin R(t)$ for some l then $R(t) := R(t) \cup \{(X, Z)^{l_1 \cup l_2 \cup \{g\}}\}$

Table 4.1: Revised completion rules (tractable version)

When it comes to implementation, one of the most important aspects is to develop a good strategy for finding the next applicable rule. In [3], the authors adopted an effective and efficient approach inspired by the linear-time algorithm for checking satisfiability of propositional Horn formulas [9]. The main idea of this method is to use queues and putting next applicable rules in those queues, then to process the queues until they are empty. In this thesis, we follow the main tenets of this approach; however, the detailed mechanism has to be enhanced a little in order to cater for axiom tags, which lie in the heart of axiom pinpointing.

Our strategy is to employ a set of queues, i.e., one queue for each concept name in the ontology, to guide the application of completion rule. In order to capture relevant axioms, we put the axiom to queue. For the purpose of simplicity, we define a mapping $Tag : \mathcal{O} \rightarrow \mathcal{N}$ which maps each axiom to a unique natural number. So the queue entries are effectively natural numbers. The possible entries of the queue are of the form

$$Tag(A \sqsubseteq B), Tag(A \sqcap A' \sqsubseteq B), Tag(A \sqsubseteq \exists r.B) \text{ and } Tag(\exists r.A \sqsubseteq B)$$

$Exp(A, B)$ is the abbreviation for the computed explanation for $A \sqsubseteq B$ which is a set of numbers denoting a subset of \mathcal{O} . We initialize $Exp(A, A) = \emptyset$ and $Exp(A, \top) = \emptyset$ for $A \in CN_{\mathcal{O}}^{\top}$. The entry $Tag(A \sqsubseteq B) \in queue(X)$ means that A is in $S(X)$ and there must exist an $Exp(X, A)$, B has to be added to $S(X)$ and at the same time $Exp(X, B)$ has to be set the union of $Tag(A \sqsubseteq B)$ and $Exp(X, A)$. $Tag(A \sqcap A' \sqsubseteq B) \in queue(X)$ means that B have to be processed w.r.t $S(X)$ if $S(X)$ contain both A' and A . Since $A, A' \in S(X)$ there must the explanation for $Exp(X, A)$ and $Exp(X, A')$, then $Exp(X, A)$, $Exp(X, A')$ together with $Tag(A \sqcap A' \sqsubseteq B)$ form the value for $Exp(X, B)$. For $Tag(A \sqsubseteq \exists r.B) \in queue(X)$, it indicates $S(X)$ contained A , and there exist an explanation $Exp(X, A)$, then (X, B) has to be added to $R(r)$ and $Exp(X, \exists r.B)$ has to be set the union of $Exp(X, A)$ and $Tag(A \sqsubseteq \exists r.B)$. Similarly, $Tag(\exists r.A \sqsubseteq B) \in queue(X)$, then $(X, A) \in R(r)$, and an explanation $Exp(X, \exists r.A)$, B has to be added to $S(X)$ and both $Exp(X, \exists r.A)$ and $Tag(\exists r.A \sqsubseteq B)$ has to be set for $Exp(X, B)$.

We view the input ontology \mathcal{O} (assumed to be in normal form w.l.o.g) as a mapping $\hat{\mathcal{O}}$ which maps to a set of queue entries every concept name and existential restrictions occurred on left hand side of GCIs in \mathcal{O} . More precisely, given $A \in CN_{\mathcal{O}}^{\top}$, $\hat{\mathcal{O}}(A)$ is the minimal set of queue entries such that:

- if $A \sqsubseteq B \in \mathcal{O}$, then $Tag(A \sqsubseteq B) \in \hat{\mathcal{O}}(A)$;
- if $A \sqcap A' \sqsubseteq B \in \mathcal{O}$ or $A' \sqcap A \sqsubseteq B \in \mathcal{O}$, then $Tag(A \sqcap A' \sqsubseteq B)$ or $Tag(A' \sqcap A \sqsubseteq B) \in \hat{\mathcal{O}}(A)$, respectively; and
- if $A \sqsubseteq \exists r.B \in \mathcal{O}$, then $Tag(A \sqsubseteq \exists r.B) \in \hat{\mathcal{O}}(A)$

In a similar manner, $\hat{\mathcal{O}}(\exists r.A)$ is the minimal set of queue entries such that, if $\exists r.A \sqsubseteq B \in \mathcal{O}$, then $Tag(\exists r.A \sqsubseteq B) \in \hat{\mathcal{O}}(\exists r.A)$.

Figure 4.1 depicts how to process each queue entry. The procedure $process(A, tag)$ performs on concept name A and the axiom specified by tag from $queue(A)$ one by one. Like that in [3], $queue(A)$ is initialized with $\hat{\mathcal{O}}(A) \cup \hat{\mathcal{O}}(\top)$ since $S(A)$ is initialized with $\{A, \top\}$.

```

procedure process( $A, tag$ )
 $tag = Tag(Y \sqsubseteq X)$  where  $Y \sqsubseteq X \in \mathcal{O}$ 
begin
  if  $Y = A_1 \sqcap A_2$  and  $\{A_1, A_2\} \subseteq S(A)$  and  $X \notin S(A)$  then
     $S(A) := S(A) \cup \{X\}$ ;
     $Exp(A, X) := Exp(A, A_1) \cup Exp(A, A_2) \cup \{tag\}$ ;
     $queue(A) := queue(A) \cup \hat{\mathcal{O}}(X)$ ;
    for all concept name  $B$  and role name  $r$  with  $(B, A) \in R(r)$  do
      if  $Exp(B, \exists r. X) = \emptyset$  then
         $Exp(B, \exists r. X) := Exp(A, X) \cup Exp(B, \exists r. A)$ ;
         $queue(B) := queue(B) \cup \hat{\mathcal{O}}(\exists r. X)$ ;
  if  $Y$  is a concept name or exist restriction and  $X \notin S(A)$  then
     $S(A) := S(A) \cup \{X\}$ ;
     $Exp(A, X) := Exp(A, Y) \cup \{tag\}$ ;
     $queue(A) := queue(A) \cup \hat{\mathcal{O}}(X)$ ;
    for all concept name  $B$  and role name  $r$  with  $(B, A) \in R(r)$  do
      if  $Exp(B, \exists r. X) = \emptyset$  then
         $Exp(B, \exists r. X) := Exp(A, X) \cup Exp(B, \exists r. A)$ ;
         $queue(B) := queue(B) \cup \hat{\mathcal{O}}(\exists r. X)$ ;
  if  $Y$  is a concept name and  $X = \exists r. B$  and  $(A, B) \notin R(r)$  then
    process-new-edge( $A, r, B, Exp(A, Y) \cup \{tag\}$ );
end;

procedure process-new-edge( $A, r, B, Exp$ )
begin
   $R(r) = R(r) \cup \{(A, B)\}$ ;
  if  $Exp(A, \exists r. B) := \emptyset$  then
     $Exp(A, \exists r. B) := Exp$ ;
  for all  $B'$  in  $S(B)$  do
    if  $Exp(A, \exists r. B') = \emptyset$ 
       $Exp(A, \exists r. B') = Exp(B, B') \cup Exp$ ;
     $queue(A) := queue(A) \cup \hat{\mathcal{O}}(\exists r. B')$ ;
  for all RI axioms  $Tag(i)$  with  $r$  on the left hand sides do
    if  $Tag(i) = r \sqsubseteq s \in \mathcal{O}$  then
      process-new-edge( $A, s, B, \{Exp, i\}$ );
    if  $Tag(i) = r \circ s \sqsubseteq t \in \mathcal{O}$  then
      for all  $(B, B') \in R(s)$  and  $(A, B') \notin R(t)$  do
        process-new-edge( $A, t, B', \{Exp, i, Exp(B, \exists s. B')\}$ );
    if  $Tag(i) = s \circ r \sqsubseteq t \in \mathcal{O}$  then
      for all  $(A', A) \in S(s)$  and  $(A', B) \notin R(t)$  do
        process-new-edge( $A', t, B, \{Exp, i, Exp(A', \exists s. A)\}$ );
end;

```

Figure 4.1: Processing the queue entries

In addition, $R(r) = \emptyset$ for $r \in RN_{\mathcal{O}}$, $Exp(A, A) = \emptyset$ and $Exp(A, \top) = \emptyset$ for all $A \in CN_{\mathcal{O}}^{\top}$. The procedure is invoked repeatedly until all queues are empty. Since this algorithm computes the subsumer sets and the explanation sets at the same time, we discuss these two facets in terms of our implementation. a) For subsumer set: the first top-most *if-clause* with regards to the axiom $A_1 \sqcap A_2 \sqsubseteq X$, it applies Rule **R2** and part of Rule **R4**. The second top-most *if-clause* with the axiom $A \sqsubseteq X$ or $\exists r.A \sqsubseteq X$ correspond to Rule **R1** and part of Rule **R4**. The third top-most *if-clause* with the axiom $Y \sqsubseteq \exists r.B$ implement **R3**, **R5**, **R6** and part of **R4**. The task is carried over to a sub-procedure `process-new-edge`. b) For explanation set: the top-most *if-clause* applies Rule **R2** and part of Rule **R4**. An important thing is the awareness of the computation of explanation set for **R4** because the implementation for Rule **R4** is split. Generally, if (A, B) is in $R(r)$, we have $Exp(A, \exists r.B)$. However, only in Rule **R3** it is possible to add a pair of concept name to R . In **R4**, according to the precondition $(X, Y)^{l_1} \in R(r)$ and $A^{l_2} \in S(Y)$, there must be an explanation for $Exp(X, \exists r.A) = l_1 \cup l_2$, if (X, A) not in $R(r)$, we could not obtain this explanation, but $Exp(X, \exists r.A)$ maybe useful for the later computation of explanation. So we need to keep tract of this intermediate explanation in **R4** and that is what the second *if-clause* of the top-most *if-clause* done. Similarly, the second top-most *if-clause* applies **R1** and **R4**, the third top-most implement **R3**, **R5**, **R6** and part of **R4** as well.

The algorithm shown above computes only a single explanation for each subsumption relationship. Nevertheless, this algorithm can not guarantee that the computed explanation is minimal. Let's look at a snapshot of the computation on the GALEN ontology.

Example 32. In a moment of the execution GALEN, the algorithm processing the concept name X1581 with the axiom $X406 \sqcap \text{Topology} \sqsubseteq X110$ where `Topology` is a concept name from GALEN and X406, X110, X1581 are new concept names introduced by the normalization rule. Before this point, $X406, \text{Topology} \in S(X1581)$ and $X110 \notin S(X1581)$, these assertions satisfy the preconditions of **R2**, and the application assert X110 to be added the

implication set of X1581 with the following explanation as its label.

$$\begin{aligned}
 Exp(X1581, X110) = & \{X1581 \sqsubseteq Topology, \\
 & TrulyHollow \sqsubseteq Hollow, \\
 & X1581 \sqsubseteq \exists hasState.TrulyHollow, \\
 & \exists hasState.Hollow \sqsubseteq X406, \\
 & X406 \sqcap Topology \sqsubseteq X110\},
 \end{aligned}$$

Also, a concept name TrulyHollowBodyStructure had also been related via the role *hasTopology* to the concept X1581, i.e, TrulyHollowBodyStructure $\sqsubseteq \exists hasTopology.X1581$. The computed label was:

$$\begin{aligned}
 Exp(TrulyHollowBodyStructure, \exists hasTopology.X1581) = \\
 \{ & TrulyHollowBodyStructure \sqsubseteq BodyStructure \\
 & \quad BodyStructure \sqsubseteq SolidStructure \\
 & \quad X216 \sqcap SolidStructure \sqsubseteq TrulyHollowStructure \\
 & \quad TrulyHollowStructure \sqsubseteq \exists hasTopology.X1581 \\
 & TrulyHollowBodyStructure \sqsubseteq \exists hasTopology.X1503 \\
 & \quad X1503 \sqsubseteq \exists hasState.TrulyHollow \\
 & \quad \exists hasState.TrulyHollow \sqsubseteq X777 \\
 & \quad X1503 \sqsubseteq Topology \\
 & \quad X777 \sqcap Topology \sqsubseteq X675 \\
 & \quad \exists hasTopology.X675 \sqsubseteq X226\}
 \end{aligned}$$

It can be easily verified that the explanations for the corresponding subsumption relationships are minimal. Since the value $Exp(TrulyHollowBodyStructure, \exists hasTopology.X110)$

had not been set before, the algorithm would assign to it the union of the previous two explanations

$$\begin{aligned} &Exp(\text{TruelyHollowBodyStructure}, \exists \text{hasTopology.X110}) = \\ &Exp(\text{X1581}, \text{X110}) \cup Exp(\text{TruelyHollowBodyStructure}, \exists \text{hasTopology.X1581}) \end{aligned}$$

Obviously, the explanation computed in this way is not minimal since the following *proper subset* also entails the subsumption in question.

$$\begin{aligned} \{ \text{X1503} &\sqsubseteq \text{Topology} \\ \text{X1503} &\sqsubseteq \exists \text{hasStateTruelyHollow} \\ \text{TruelyHollow} &\sqsubseteq \text{Hollow} \\ \exists \text{hasState.Hollow} &\sqsubseteq \text{X406} \\ \text{X406} \sqcap \text{Topology} &\sqsubseteq \text{X110} \\ \text{TruelyHollowBodyStructure} &\sqsubseteq \exists \text{hasTopology.X1503} \} \end{aligned}$$

⊣

In the example above, the reason that caused non-minimality lies in the selection of concept names, i.e., which concept name should be processed first. If the concept name `TruelyHollow` had been processed before `X1503`, we could have obtained a minimal explanation. This algorithm performs on every concept name in an arbitrary order. It provides no strategy nor heuristic search to choose the optimum one. On the contrary, the computed explanation was compared to previous explanation and fetch the smaller one in each step in the general labelled subsumption algorithm discussed in Chapter 3.

4.2 A First Implementation

We have mentioned that the algorithm computing one explanation leads to non-minimality, and the explanation set we talked so far are relative to the normalized ontology, i.e., all

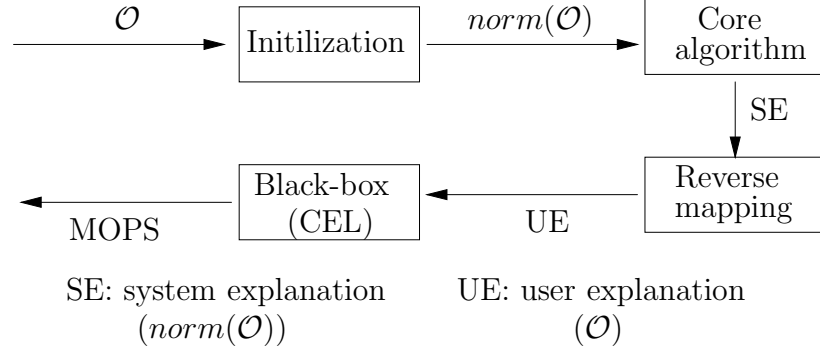


Figure 4.2: workflow of the first implementation

the axioms are in normal form. It is apparent that the explanations are not minimal as well when talking about the original terminology. In fact, a minimal explanation relative to $norm(\mathcal{O})$ could give rise to a non-minimal explanation relative to \mathcal{O} . This means that even the algorithm provides a minimal explanation, the reverse mapping could result in non-minimality of explanations w.r.t \mathcal{O} . In Example 23, given the subsumption $A \sqsubseteq D$ and its MOPS $\{A \sqsubseteq B, B \sqsubseteq C, A \sqcap C \sqsubseteq D\}$, one of its original explanation by reverse-mapping this MOPS to original terminology is $\{A \sqsubseteq B \sqcap C, B \sqsubseteq C, A \sqcap C \sqsubseteq D\}$. Obviously, this original explanation is not minimal for subsumption $A \sqsubseteq D$.

In order to compute the minimal explanations w.r.t the original ontology, we use the *black box* approach with the help of the CEL reasoner in our implementation. The black box method sweeps through every axiom in the computed explanations, and thereby tests if the subsumption in question is still satisfied when the axiom under consideration is absent. We implement this idea in the Common Lisp, Allegro CL [12]. Since it is the first implementation of the tractable approach to axiom pinpointing, we do not consider efficiency when choosing the data structures. Figure 4.2 displays four major modules of computation in our implementation. In the first phase, the input ontology is transformed into normal form and the data structures, such as S , R , $\hat{\mathcal{O}}$, queues etc, are initialized. In the implementation, we use hash table to store normalized axioms. The *core algorithm* is the tractable

version of our axiom pinpointing algorithm presented in section 4.1. We call the explanation sets computed by the core algorithm *system explanations* (SE), An SE is a (possible) non-minimal explanation w.r.t $norm(\mathcal{O})$. An SE is then reverse mapped to a so-called *user explanations*(UE) which is a (possible) non-minimal explanation w.r.t \mathcal{O} . Finally, the algorithm yields a MOPS w.r.t \mathcal{O} through the black-box reasoning. Note that the whole input ontology \mathcal{O} can be considered a non-minimal explanation indeed, and thus only the black box approach with the CEL reasoner is in principle sufficient to compute MOPS. However, this way is rather brute-force and might not work in practice though the orache reasoner is efficient. We will see the experiment of this implementation on GALEN in the next chapter, the black box phrase occupies most of the computation time in terms of the whole computation.

4.3 Complexity

In order to show termination in polynomial time, it suffices to show that each of the four phases displayed in Figure 4.2 needs polynomial time. For the normalization phase, it has been proven in Lemma 10 that the transformation to normal form is linear in the size of \mathcal{O} .

Corollary 33. *Let \mathcal{O} is a normalized \mathcal{EL}^+ ontology over $CN_{\mathcal{O}}^{\top}$ and $RN_{\mathcal{O}}$, the revised subsumption algorithm shown in Table 4.1 terminates in polynomial time in $|\mathcal{O}|$.*

PROOF. The only difference compared to the unlabelled subsumption algorithm is that, in the revised labelled algorithm, each element in S or R is labelled with a set of tags. Due to the stronger preconditions, $\{B^{l_1}, B^{l_2}\} \subseteq S(A)$ implies that $l_1 = l_2$ for the implication set of concept name $A \in CN_{\mathcal{O}}^{\top}$, and for all labels, i.e., $label$ in $A^{label} \in S(B)$ are also only of polynomial size of \mathcal{O} . The application of each complete rule of revised labelled algorithm adds a concept name in $CN_{\mathcal{O}}^{\top}$ to S or a binary relation (A, B) to R with a subset of \mathcal{O} as its labels. No rule removes elements from S and R . The total number of application can be performed in polynomial time of $|\mathcal{O}|$. ■

It is easily seen that the reverse mapping can be computed in polynomial time, since it directly maps each normalized axiom to the generating original axiom. In fact, the reverse mapping is linear in the size of $|\mathcal{O}|$. As for the black-box reasoning phase, i) CEL is a polynomial time reasoner, ii) There is only one explanation for each subsumption and the number of subsumption relationships is bounded by $|CN_{\mathcal{O}}^{\top}|^2$, so the black box reasoning with CEL can be done in polynomial time.

Theorem 34. *One minimal explanation for subsumption in \mathcal{EL}^+ can be decided in polynomial time.*

Chapter 5

Experiments on GALEN

We implemented the tractable pinpointing algorithm from the previous chapter and evaluated it on GALEN. In fact, we utilize a fragment of the GALEN ontology. Large parts of GALEN medical knowledge base can be expressed in \mathcal{EL} with GCIs and transitive roles [17]. Since the full ontology is beyond the capacity of \mathcal{EL}^+ , we use a stripped-down version in which inverse role axioms are dropped and functional roles are treated as if they were normal ones. This stripped-down version can be expressed by \mathcal{EL}^+ , and we present the experimental results of the implementation on the GALEN ontology in this chapter.

First, we briefly introduce the structure and size of GALEN used here. The ontology is formulated in the *KRSS* syntax¹. Table 5.1 lists some axioms that are available in the GALEN ontology corresponding to the syntax of the description logic \mathcal{EL}^+ . Here, CN ranges over \mathcal{EL}^+ concept name. C, C_1 and C_2 are \mathcal{EL}^+ concept descriptions. RN, RN_1 and RN_2 correspond to role names. For instance:

(**define-concept** ViralInfection (**AND** Infection
(**SOME** *hasCausalAgent* Virus)))

This concept definition defines the concept name “ViralInfection” by the complex concept description “Infection $\sqcap \exists hasCausalAgent. Virus$ ” in terms of \mathcal{EL}^+ syntax. We could merge

¹<http://dl.kr.org/krss-spec.ps>

	\mathcal{EL}^+ syntax	GALEN syntax
Conjunctions	$C_1 \sqcap C_2$	(AND $C_1 C_2$)
Existential restrictions	$\exists RN. C$	(SOME $RN C$)
Primitive concept definition	$CN \sqsubseteq C$	(define-primitive-concept $CN C$)
Concept definition	$CN \equiv C$	(define-concept $CN C$)
General concept inclusion	$C_1 \sqsubseteq C_2$	(implies $C_1 C_2$)
Role transitivity axiom	$RN \circ RN \sqsubseteq RN$	(define-primitive-role RN : transitive t)
Role hierarchies	$RN_1 \sqsubseteq RN_2$	(define-primitive-role RN_1 : parents (RN_2))

Table 5.1: \mathcal{EL}^+ syntax versus GALEN syntax

different properties of a role name into one axiom, such as

(**define-primitive-role** *hasFunctionalComponent* : **parents** (*FunctionalAttribute*)
: **transitive** t)

which represents both the transitivity and role hierarchy.

The GALEN ontology used in this experiment has a total number of 4,367 axioms, and 699 of them are concept definitions, 2,041 are primitive concept definition, 1,214 are GCIs and 413 are role axioms. It has 2,748 concept names and 413 role names.

We will display the evaluation results of the implementation on the GALEN ontology. The experiments were performed on a standard PC with 2.8 GHz Pentium 4 processor and 512 MB of memory. Table 5.2 displays the elapse times spent on each computation stage as shown in Figure 4.2. From the result, we can see the real time of the core algorithm paltrily occupies in the whole computation, while the first phase for normalization and initialization procedure was relatively more time-consuming. The initialization includes normalization, initializing data structure, such as labelled implication set S , R , queues etc.

The application of the tractable algorithm on the Galen ontology produced 27,973 subconcept-superconcept relationships, all accompanied by a single explanation. As we have seen before, there are two sources of non-minimality, the core algorithm and the reverse mapping from system explanation to user explanation. Since our implementation only output the user explanation, it is interesting to find out the system explanation as

Time(mm:ss)	Real Time	Percentage
Initialization	07:27	40.6%
Core algorithm	01:35	8.7%
Reverse mapping	01:51	10.2%
Black Box	07:22	40.5%
Total	18:01	100%

Table 5.2: Time Consumption of running one-explanation algorithm on GALEN

well. Table 5.3 shows how many system explanations (in normal form) and how many user explanation (in original form) are not minimal respectively. The *degree* of non-minimality denotes, if an explanation is not minimal, the number of redundant or unnecessary axioms. The *#non-minimal* explanations displays the total number of non-minimal explanations for the computed subconcept-superconcept relationships of concept names ². For instance, there are totally 1,065 system explanations which are not minimal, and 604 of which, about 56.7% have 7 unnecessary axioms. Although the tractable algorithm is not able to generate a minimal explanation for each subsumption relationship, the results show that the degree of non-minimality is not high, and even acceptable. Approximately 3.8% of the system explanations and 4.3% of the user explanations which cover very little for whole 27,894 explanations are not minimal. Furthermore for user explanations, there are 723, which occupy half of the non-minimal explanation, with only 1 redundant axiom. This level of non-minimality is almost negligible to the use of such a system. That is to say, though not minimal, the computed set of axioms do explain why subsumption holds. We also discern the number of non-minimal user explanations is more than the non-minimal system explanations because of the reverse mapping.

²In fact, the subconcept-superconcept relationships include subsumption relationships between concept name and value restriction, however, we only provide non-minimal explanations for subsumption between concept names.

Table 5.3: Degree of the Non-minimality

	#non-minimal (%)	Degree of non-minimality										
		1	2	3	4	6	7	8	9	10	11	13
SE	1065 (3.8%)						604	393	5	4	58	1
UE	1224 (4.3%)	723	420	19	17	1	44					

SE: system explanation **UE**: user explanation
 #non-minimal: the number of non-minimal explanation

Chapter 6

Conclusion and Future Work

In this thesis, we have developed two labelled algorithms for non-standard inference of explaining subsumption in the description logic \mathcal{EL} with GCIs and CRIs. In general, minimal explanation is not unique, and there may be exponentially many explanations in the worst case. We have investigated that the corresponding decision problem of explanation in \mathcal{EL}^+ is NP-complete. We restricted the labelled algorithm so that one explanation for each subsumption can be computed in polynomial time. This algorithm can not guarantee that all computed explanations are minimal. From the experiments on the GALEN ontology, the non-minimal explanations occupy a little in whole computed explanations and the degree of non-minimality, i.e., the number of redundant axioms, is not high and even acceptable. That to say, despite not minimal, the computed set of axioms do explain why subsumption holds. Generally the non-minimality comes from two factors, the core algorithm and the reverse mapping from system explanations to user explanations. The core algorithm provides a polynomial time procedure to compute a single explanation for each subsumption relationship. Though not every explanation is minimal, the minimal one can be computed in polynomial time with the help of the black-box approach using the CEL reasoner as an oracle.

Since the standard labelled algorithm potentially requires exponential time, it might not be immediately enticing to directly implement it. However, there are circumstances when

all conceivable explanations for a certain logical consequence are necessary. For example, when the logical consequence in question is not desired, and the ontology developer needs to dispensed with it. By removing all axioms found in a single explanation, it is not guaranteed that the consequence is dismissed. A stepwise computation with user interaction might be a tractable solution – the user is provided with a single explanation, and upon request, the next explanation (if any) is computed incrementally. This way, users could be satisfied both in terms of affordable computation times and sufficient explanation output.

Another interesting work for future would be to optimize the algorithm and implementation. Since we do not mainly consider the efficiency when choosing data structures, it is worthwhile to revise eligible data structures and use the more efficient ones. Last but not least is to extend \mathcal{EL}^+ to the more expressive language \mathcal{EL}^{++} , which introduces among other things nominals and concrete domains.

Bibliography

- [1] F. Baader, S. Brandt, and C. Lutz. Pushing the \mathcal{EL} envelope. LTCS-Report LTCS-05-01, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology, Germany, 2005. See <http://lat.inf.tu-dresden.de/research/reports.html>.
- [2] F. Baader, D. Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
- [3] F. Baader, C. Lutz, and B. Suntisrivaraporn. Is tractable reasoning in extensions of the description logic \mathcal{EL} useful in practice? In *Proceedings of the Methods for Modalities Workshop (M4M-05)*, Berlin, Germany, 2005.
- [4] F. Baader, C. Lutz, and B. Suntisrivaraporn. CEL—a polynomial-time reasoner for life science ontologies. In U. Furbach and N. Shankar, editors, *Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR'06)*, volume 4130 of *Lecture Notes in Artificial Intelligence*, pages 287–291. Springer-Verlag, 2006.
- [5] Franz Baader and Bernhard Hollunder. Embedding defaults into terminological knowledge representation formalisms. Technical Report RR-93-20, Germany, 1993.
- [6] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

-
- [7] S. Brandt. Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else? In R. López de Mantáras and L. Saitta, editors, *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-2004)*, pages 298–302. IOS Press, 2004.
- [8] S. Brandt. Reasoning in \mathcal{ELH} w.r.t. general concept inclusion axioms. LTCS-Report LTCS-04-03, 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>.
- [9] William F. Dowling and Jean H. Gallier. Linear-time algorithms for testing the satisfiability of propositional horn formulae. *J. Log. Program.*, 1(3):267–284, 1984.
- [10] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Mathematical sciences series. Freeman, 1979.
- [11] N. Guarino and R. Poli. Formal ontology in conceptual analysis and knowledge representation, 1995.
- [12] Allegro Common LISP. Franz Inc
See <http://www.franz.com>.
- [13] B. Nebel. Terminological cycles: Semantics and computational properties. In J. F. Sowa, editor, *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, pages 331–361. Morgan Kaufmann Publishers, San Mateo (CA), USA, 1991.
- [14] Bijan Parsia, Evren Sirin, and Aditya Kalyanpur. Debugging owl ontologies.
- [15] Rafael Penaloza. Pinpointing in tableaux. LTCS-Report LTCS-06-05, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology, Germany, 2006. See <http://lat.inf.tu-dresden.de/research/reports.html>.

-
- [16] 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.
- [17] A. Rector and I. Horrocks. Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions, 1997.
- [18] A. Rector, W. Nowlan, and A. Glowinski. Goals for concept representation in the galen project, 1993.
- [19] R. Reiter. A theory of diagnosis from first principles. In M. L. Ginsberg, editor, *Readings in Nonmonotonic Reasoning*, pages 352–371. Kaufmann, Los Altos, CA, 1987.
- [20] B. Smith, J. Williams, and S. Schulze-Kremer. The ontology of the gene ontology, 2003.
- [21] S.Schlobach and R.Cornet. Non-standard reasoning services for the debugging of description logical terminologies. In *In Proceedings of the International Joint Conference on Artificial Intelligence - IJCAI'03*, Acapulco, Mexico, 2003. Morgan Kaufmann.
- [22] B. Suntisrivaraporn. Cel manual, a polynomial-time classifier for the description logic \mathcal{EL}^+ . See <http://lat.inf.tu-dresden.de/systems/cel/>.
- [23] B. Suntisrivaraporn. Optimization and implementation of subsumption algorithms for the description logic \mathcal{EL} with cyclic tboxes and general concept inclusion axioms. Master thesis, TU Dresden, Germany, 2005.