Grounded Circumscription in Description Logics

Stathis Delivorias

MASTER THESIS
April 2015

Supervised by Professor Sebastian Rudolph

Fakultät Informatik
Technische Universität Dresden

Declaration of Authorship

Hereby, I certify that this dissertation is the result of my own work and is written by me. Any help that I have received in my research work has been acknowledged. In addition, I certify that I have not used any auxiliary sources and literature except those cited in the thesis.

Author :	Stathis Delivorias
Matriculation number :	3930352
Date of submission :	09.04.2015
Signature :	

Abstract

Circumscription is a paradigm of non-monotonic logic meant to formalize the common sense understanding that among competing theories that represent phenomena equally well, the one with the fewest assumptions should be selected. Description Logics are knowledge representation formalisms designed to describe and reason about qualitative properties and aspects of a system. In this essay we aim to fuse Description Logics, which traditionally are monotonic, with a restricted version of Circumscription, called Grounded Circumscription. The work is based on a 2011 publication by K. Sengupta, A.A. Krisnadhi and P. Hitzler, which throughout this study we will refer to as "the original paper". The paper introduces the main idea of grounded circumscription along with algorithms for certain decision problems. We have optimized and modified these ideas. The optimization was our initial aim, in particular we wanted (and largely achieved) to transfer a big part of the reasoning to standard Description Logics, for which tools and results already exist. But in the process we uncovered some insufficiencies in the original paper, hence we have modified the main definition to one that is more effective and seems more intuitive.

Acknowledgment

I am very greatful to professor Sebastian Rudolph, for his support and his confidence in this project. His remarks were decisive in making this work coherent and comprehensive.

I would like to thank my friends and family for bearing with my absence and supporting me.

Some credit must also go to Socrates. I have been travelling around encountering ideals in the platonic universe for quite some time now.

Contents

Chapter 1

Introduction

Circumscription is a paradigm of non-monotonic logic introduced by John McCarthy in 1980 [4]. The main idea is to formalize the common sense understanding that among competing theories that predict equally well, the one with the fewest assumptions should be selected. This is basically an application of the principle known as "Occam's razor" to logic. It is also similar to the closed world assumption, where what is not known to be true is taken to be false. In its original first-order logic formulation, circumscription minimizes the extension of some predicates, where the extension of a predicate is the set of tuples of values the predicate is true on.

Description Logics (DLs) are knowledge representation formalisms designed to describe and reason about qualitative properties and conceptual aspects of a system [1, 6]. Ontology languages based on DLs have been widely adopted in a large class of application areas. One of the most prominent applications of DLs is to provide the underlying logical basis of the web ontology language OWL 2, which is the current recommendation of the World Wide Web Consortium (W3C) [11, 2]. Therein DLs are used to represent the intended meaning of Web resources and establish powerful reasoning tools, so as to facilitate machine understandability of Web pages. From a more scholarly perspective, DLs are decidable fragments of first order logic.

Description Logics traditionally operate within the monotonic realm, namely the addition of more assertions to a knowledge base does not negate previously inferred information. But in many prevalent application domains, such as common sense reasoning, this property does not hold. Conclusions might

need to be revised in the light of new information. Hence it is quite intriguing to try to develop a DL framework where reasoning would be non-monotonic.

In this essay we aim to fuse Description Logics, with a restricted version of Circumscription, called Grounded Circumscription. The work is based on a 2011 publication by K. Sengupta, A.A. Krisnadhi and P. Hitzler, which throughout this thesis we will refer to as "the original paper" [9]. In ground circumscription, some of the predicates in our language (which in DL can only be unary or binary) are chosen to be grounded and minimized. Grounded means that their interpretations must include only named individuals, i.e. elements of the domain that correspond to one of the constants that appear in our knowledge base. Moreover those predicates are minimized in the sense that we accept only models which assign as few individuals as possible to them, so that there cannot be a model whose extensions of these predicates are subsets of the respective extensions in the minimal model.

In the original paper, the main idea of grounded circumscription is given along with algorithms for certain decision problems. We have optimized and modified these ideas. The optimization was our initial aim, in particular we wanted (and largely achieved) to transfer a big part of the reasoning to standard DLs, for which there already exist tools and available results. But in the process we uncovered some insufficiencies in the original paper, to the discussion of which we devote a special chapter in this study. Hence we have modified the main definition to one that is more effective and seems more intuitive.

After introducing the particular DL formalism and terminology that we work on (Chapter 2), we specify the basic notions (Chapter 3) and proceed to present an algorithm for satisfiability (Chapter 4) which is predominantly in the monotonic sphere. The next section (Chapter 5) is introducing important notions which are put to use in the algorithm for entailment of facts (Chapter 6). Following are supplementary results that further develop the theory of grounded circumscription in DLs (Chapter 7) and the discussion on the differences and improvements in comparison with the original paper (Chapter 8). Finally we give an overview of the contribution of this endeavor and discuss prospects of further research (Chapter 9).

Notation

In the following, $Part(X)$ symbolizes the set of all partitions of a set X, $\mathscr{P}(X)$ symbolizes the power set of X, Σ^* symbolizes the set of all finite words built from an alphabet Σ and of course we denote $\{0, 1\}$ with $\mathbb{Z}_2$, as the quotient ring of the ring of integers modulo the ideal of even numbers.

Chapter 2

Preliminaries

In this chapter we give a brief introduction to our formalism and the main terminology and ideas around it.

2.1 The Choice of Language

The choice of DL formalism is $\mathcal{ALCO}$, since it has been developed to a sufficient extent and where the trade-off between expressivity and complexity seems optimal. Also with our approach, the ability to embed non-monotonicity is preserved when expanding the language, hence choosing a rather simple language is an advantage. In the original paper, decidability of ground circumscription is proven for rather complex languages which feature concept products, role hierarchies and role disjunctions. And then independently, algorithms which apply only to $\mathcal{ALC}$ are given. In contrast, our work is entirely based on $\mathcal{ALCO}$ but it can trivially be extended to any more complex formalism, provided that it is decidable.

It is important to stress out here the significance of the inclusion of nominals in our language. Many of the results are largely obtained using nominals. But this is not a downturn since nominals do not increase the complexity of $\mathcal{ALC}$, contrary to other concept constructors.

2.2 $\mathcal{ALCO}$ Syntax

Description Logics emerge from the need to have an expressive formalism, in the style of first-order logic (FOL), but which is decidable, hence allowing for automated reasoning. As a result most of them are decidable fragments of FOL, with a more concise notation, meant to represent the expressive power of the characteristics remaining in the language after its reduction.

Let N_C, N_r and N_I be mutually disjoint sets of *concept-, role-* and *individual names*, respectively. Concepts C in $\mathcal{ALCO}$ are built using the grammar rule:

$$C ::= \top \mid A \mid \{a\} \mid \neg C \mid C \sqcap C \mid \exists r.C$$

where $A \in N_C$, $r \in N_r$, and $a \in N_I$. The symbols $\top$ ("truth"), $\{a\}$ ("nominal"), $\neg$ ("negation"), $\sqcap$ ("conjunction") and $\exists$ ("existential restriction") are the logical operators of the Description Logic $\mathcal{ALCO}$.

We employ the usual abbreviations

$$\begin{aligned} \bot &= \neg\top && \text{"falsehood"}, \\ C \sqcup D &= \neg(\neg C \sqcap \neg D) && \text{"disjunction"}, \\ \forall r.C &= \neg\exists r.\neg C && \text{"universal restriction"}. \end{aligned}$$

An expression of the form $C \sqsubseteq D$, where C and D are concepts, is called a *concept inclusion*. A finite set of concept inclusions is a *TBox*. An expression of the form $C(a)$, where C is a concept and $a \in N_I$, is called a *concept assertion*. For $r \in N_R$ and $a, b \in N_I$, an expression of the form $r(a, b)$ is called a *role assertion*. A finite set of concept and role assertions is called an *ABox*.

A pair $K = (\mathcal{T}, \mathcal{A})$ consisting of an TBox $\mathcal{T}$ and an ABox $\mathcal{A}$ is called a *knowledge base* (abbreviated frequently as KB). For ease of presentation, in this study we will usually understand a knowledge base as a single set of axioms, which would formally be expressed as $K = \mathcal{T} \cup \mathcal{A}$. We will not refer to Aboxes and Tboxes individually, rather we wil handle the knowledge base as a whole.

2.3 $\mathcal{ALCO}$ Semantics

An *interpretation* is a pair $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$, where Δ is a non-empty *domain* and $\cdot^{\mathcal{I}}$ is a function that maps every $a \in N_I$ to $a^{\mathcal{I}} \in \Delta$, every $A \in N_C$ to $A^{\mathcal{I}} \subseteq \Delta$, and every $r \in N_r$ to $r^{\mathcal{I}} \subseteq \Delta \times \Delta$. The mapping $\cdot^{\mathcal{I}}$ is naturally extended to all concepts by setting

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

An interpretation $\mathcal{I}$ *satisfies*

- a concept C if $C^{\mathcal{I}} \neq \emptyset$,
- a concept inclusion $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$,
- a concept assertion $C(a)$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and
- a role assertion $r(a, b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$.

We say that $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$ or an ABox $\mathcal{A}$ if it satisfies every concept inclusion in $\mathcal{T}$ or every assertion in $\mathcal{A}$, respectively. $\mathcal{I}$ is a model of a knowledge base $K = (\mathcal{T}, \mathcal{A})$ if $\mathcal{I}$ is a model of both $\mathcal{T}$ and $\mathcal{A}$.

- If there exists a model of a knowledge base K, then K is a *satisfiable* KB.

- If every model of K satisfies $C(a)$, we say that $C(a)$ is *entailed* by K.

- If there exists a model of a knowledge base K that satisfies C, then C is *satisfiable* with respect to K.
- If every model of K satisfies $C \sqsubseteq D$, then C is *subsumed* by D with respect to K.

2.4 Reasoning & Complexity

Description Logics satisfy requirements emerging from a great variety of practical modeling scenarios. Their basic modeling features concern specifying and querying knowledge.

The main reasoning tasks in our formalism are, given a knowledge base K, concepts C, D and an individual a, the following:

- **Concept Satisfiability** Is C satisfiable with respect to K?
- **Concept Subsumption** Is C subsumed by D with respect to K?
- **KB Satisfiability** Is there a model of K?
- **Instance Checking** Is $C(a)$ entailed by K?

We will focus only on the two last, as the two first can be encoded as subcases of the third. In particular the satisfiability of the knowledge base $K \cup \{\top \sqsubseteq \exists r.C\}$, where r does not appear in K, is equivalent to the satisfiability of C with respect to K. Furthermore, the subsumption of C from D wrt K is equivalent to the unsatisfiability of the concept $\neg D \sqcap C$.

$\mathcal{ALCO}$ is a rather simple language, expressive enough to be efficient in several applications, but allowing effective reasoning as it has the finite model property, i.e. when a knowledge base is satisfiable, it has a finite model [3]. It features many modeling capabilities usually found in knowledge representation languages. Complexity of reasoning tasks in $\mathcal{ALCO}$ is in the worst cases EXPTIME-complete [8, 7].

A great deal of valuable tools have been developed for reasoning within DLs, most of which support $\mathcal{ALCO}$ [10, 5].

Chapter 3

Fundamental Notions

In this chapter we formally define the basic notions of ground circumscription. The definition of minimality is reestablished in solid grounds, which can prove a useful framework for further development of this theory.

3.1 Ground Extension

A central notion in this study is that of ground extension of a predicate with respect to a certain interpretation, which is the set of individual names or pairs of individual names (depending on whether the predicate is a concept or a role), whose interpretations belong to the interpretation of this predicate. Given a knowledge base K, the set of indivual names that appear in K are symbolized $Ind(K)$.

Definition 1 Let K be an $\mathcal{ALCO}$ knowledge base and $\mathcal{I}$ an interpretation. The *ground extension* wrt $\mathcal{I}$ of a predicate $W \in N_C \cup N_r$, is the following set:

$$Ext^{\mathcal{I}}(W) := \begin{cases} \{a \in Ind(K) | a^{\mathcal{I}} \in W^{\mathcal{I}}\} & \text{if } W \in N_C \\ \{(a,b) \in Ind^2(K) | (a^{\mathcal{I}}, b^{\mathcal{I}}) \in W^{\mathcal{I}}\} & \text{if } W \in N_r \end{cases}$$

where of course $Ind^2(K) = Ind(K) \times Ind(K)$. ⊣

The key role that ground extension plays, is evident by its frequent presence throughout the rest of this work. $Ext^{\mathcal{I}}(\cdot)$ can be naturally extended to be

applicable to any concept description: if C is a concept, then $Ext^{\mathcal{I}}(C) := \{a \in Ind(K) | a^{\mathcal{I}} \in C^{\mathcal{I}}\}$. Since nominals are valid concept constructors in our language, we can ultimately view $Ext^{\mathcal{I}}(C)$ as a concept description, provided that C is a concept as well. One more important property of ground extension, which is easy to verify, is that it is monotonic with respect to set inclusion, i.e if $A^{\mathcal{I}} \subseteq B^{\mathcal{I}}$ then $Ext^{\mathcal{I}}(A) \subseteq Ext^{\mathcal{I}}(B)$.

3.2 Minimality

The main idea in grounded circumscription is to select some predicates (concept and role names), and demand that for every model their interpretation is grounded, i.e. it includes only named individuals, and that it is minimized, in the sense that there cannot be an interpretation that assigns fewer individuals to those predicates and still is a model of our given knowledge base.

Definition 2 A *GC-$\mathcal{ALCO}$-KB* is a pair (K, M) where K is an $\mathcal{ALCO}$ knowledge base and $M \subseteq N_C \cup N_r$. For every $W \in M$ we say that W is *closed* wrt K. If $\mathcal{I}$ and $\mathcal{J}$ are models of K then the *"smaller than"* relation is defined in the following way:

$\mathcal{I} \prec_M \mathcal{J}$ if

i) $Ext^{\mathcal{I}}(\{a\}) = Ext^{\mathcal{J}}(\{a\})$ for every $a \in Ind(K)$,

ii) $Ext^{\mathcal{I}}(W) \subseteq Ext^{\mathcal{J}}(W)$ for every $W \in M$ and

iii) there is a $W \in M$ such that $Ext^{\mathcal{I}}(W) \subset Ext^{\mathcal{J}}(W)$. ⊣

Definition 3 A model $\mathcal{I}$ of K is called *grounded* wrt M if

i) $C^{\mathcal{I}} \subseteq \{b^{\mathcal{I}} | b \in Ind(K)\}$ for every $C \in M \cap N_C$.

ii) $r^{\mathcal{I}} \subseteq \{(a^{\mathcal{I}}, b^{\mathcal{I}}) | a, b \in Ind(K)\}$ for every $r \in M \cap N_r$. ⊣

The following lemma indicates a property of the "smaller than" relation which is essential in reaching the satisfiability result of the next chapter in an immediate way and almost without touching non-monotonic territory.

Lemma 1 The relation $\prec_M$ is well-founded on the class of grounded models of a knowledge base K wrt to M.

Proof: Let $Gmod$ be the class of all grounded models of K. We define *ground cardinality* of a model $\mathcal{I}$ to be the sum of the cardinalities of the ground extensions of the predicates specified in M. In particular, we have the function

$$gc : Gmod \rightarrow \mathbb{N} \text{ with } gc(\mathcal{I}) := \sum_{W \in M} |Ext^{\mathcal{I}}(W)|.$$

M is assumed to be a finite set as of course is $Ind(K)$, hence in a grounded model the ground extension of every predicate in M is a finite set, thus the above function is well-defined. Let $\mathcal{I}_1, \mathcal{I}_2 \in Gmod$ and $\mathcal{I}_1 \prec_M \mathcal{I}_2$. Then the third condition in Definition 2 ensures that $gc(\mathcal{I}_1) < gc(\mathcal{I}_2)$. Let $B \subseteq Gmod$ with $B \neq \emptyset$, so B is an arbitrary set of grounded models of K. Let $n = \min\{gc(\mathcal{J}) | \mathcal{J} \in B\}$. So there is at least one $\mathcal{J} \in B$ such that $gc(\mathcal{J}) = n$. This $\mathcal{J}$ is a minimal element of B wrt the relation $\prec_M$. Hence $\prec_M$ is a well founded relation on $Gmod$. ♣

Definition 4 An interpretation $\mathcal{I}$ is a *GC-model* of (K, M) if it is a grounded model of K wrt M and $\mathcal{I}$ is minimal wrt M, i.e. there is no grounded model $\mathcal{J}$ of K such that $\mathcal{J} \prec_M \mathcal{I}$. (K, M) is *satisfiable* if it has a GC-model. A statement ϕ is a *logical consequence* of (K, M) if every GC-model of (K, M) satisfies ϕ. We then say that (K, M) *entails* ϕ. ⊣

Note that ϕ in the above definition could be a GCI, a concept assertion or a role assertion. Henceforth we will frequently substitute the term GC-model, with *minimal grounded model* or simply *minimal model*. Definitions 2, 3 and 4 are taken from the original paper. Although 3 and 4 are actually identical with the ones given in the original paper, Definition 2 differs substantially. In particular, the first condition for the minimality relation in the original paper requires that two interpretations have equal domains in order for them to be comparable. Firstly, that is counter-intuitive. When we say that a model has fewer assumptions than another model, this does not imply any similarity of their domains, it only requires that those predicates which are of importance to us are somehow smaller. What is more, by this imposition the search space

for minimal models becomes infinite and then the complexity of some tasks blows up, even if they remain within decidable waters.

Chapter 8 contains a more thorough discussion about the results presented in the original paper and the motives behind our decision to reformulate the definition of minimality.

Chapter 4

Satisfiability of a GC Knowledge Base

We present now a direct and complexitywise cheap way of determining whether a GC-$\mathcal{ALCO}$ knowledge base is satisfiable. To this end we first enhance the KB with axioms that ensure grounding of the closed predicates and then we take advantage of Lemma 1, which effectively says that if a grounded model exists, then a minimal grounded model must exist as well.

Definition 5 Let (K, M) be a *GC-$\mathcal{ALCO}$-KB*, where $M \cap N_C = \{A_1, ..., A_n\}$ and $M \cap N_r = \{r_1, ..., r_m\}$. We define K_M as the $\mathcal{ALCO}-KB$ which consists of all the axioms that are included in K as well as the following ones:

- $P \equiv \{x | x \in Ind(K)\}$ where P is a fresh concept name,
- $A_i \sqsubseteq P$ for every $i \in \{1, ..., n\}$,
- $\exists r_j.\top \sqsubseteq P$ for every $j \in \{1, ..., m\}$,
- $\top \sqsubseteq \forall r_j.P$ for every $j \in \{1, ..., m\}$.

K_M is then called a *grounded* $\mathcal{ALCO}$ knowledge base. ⊣

We do not give an explicit algorithm for determining satisfiability of a GC knowledge base. That is because we show that this decision problem is equivalent to the satisfiability checking of a (standard) $\mathcal{ALCO}$ knowledge base.

To solve the reasoning tasks of ground circuscription with the use of the already developed monotonic DL reasoning tools was our aim, and as the next proposition shows, in this case it is proven to be achieved quite ideally.

Proposition 1 Let (K, M) be a *GC-$\mathcal{ALCO}$-KB*. (K, M) is satisfiable if and only if the $\mathcal{ALCO}$ knowledge base K_M is satisfiable.

Proof: Let $M \cap N_C = \{A_1, ..., A_n\}$ and $M \cap N_r = \{r_1, ..., r_m\}$.

($\Rightarrow$:) Let (K, M) be satisfiable. Therefore it has a GC-model. Let $\mathcal{I}$ be a GC-model of (K, M), i.e. a minimal grounded model of K with respect to M. Let $\mathcal{I}'$ be an interpretation such that $P^{\mathcal{I}'} = \{b^{\mathcal{I}'} | b \in Ind(K)\}$ and $\mathcal{I}'$ agrees with $\mathcal{I}$ on all the rest. So then $\mathcal{I}'$ is GC-model of (K, M) and it satisfies all the axioms that are common in K and K_M. For the rest of the axioms in K_M we have:

- $P^{\mathcal{I}'} = \{b^{\mathcal{I}'} | b \in Ind(K)\}$ therefore $P \equiv \{x | x \in Ind(K)\}$ is satisfied.
- Because $\mathcal{I}'$ is a grounded model, for all $i \in \{1, ..., n\}$ it holds that $A_i^{\mathcal{I}'} \subseteq \{b^{\mathcal{I}} | b \in Ind(K)\}$ and so all axioms of the form $A_i \sqsubseteq P$ for every $i \in \{1, ..., n\}$ are satisfied.
- Similarly, for all $j \in \{1, ..., m\}$ it holds that $r_j^{\mathcal{I}'} \subseteq \{(a^{\mathcal{I}}, b^{\mathcal{I}}) | a, b \in Ind(K)\}$ so then $(x, y) \in r_j^{\mathcal{I}'}$ implies $x, y \in P^{\mathcal{I}'}$ thus all axioms of the form $\exists r_j.\top \sqsubseteq P$ and $\top \sqsubseteq \forall r_j.P$ are satisfied.

($\Leftarrow$:) Let K_M be satisfiable. Thus it has a model $\mathcal{J}$. We will show that there exists a minimal grounded model $\mathcal{I}$ for (K, M). The interpretation $\mathcal{J}$ is a grounded model for (K, M) wrt M. This is because

- for every $A_i \in M \cap N_C$ the axiom $A_i \sqsubseteq P$ of K_M implies $A_i^{\mathcal{J}} \subseteq \{b^{\mathcal{J}} | b \in Ind(K)\}$ and
- for every $r_j \in M \cap N_r$ the axioms $\exists r_j.\top \sqsubseteq P$ and $\top \sqsubseteq \forall r_j.P$ imply that for all $x \in \Delta^{\mathcal{J}}$, if there exists $y \in \Delta^{\mathcal{J}}$ such that $(x, y) \in r_j$, then $x \in P^{\mathcal{J}}$ and if $x \in \Delta^{\mathcal{J}}$ then for all $y \in \Delta^{\mathcal{J}}$ such that $(x, y) \in r_j$ holds that $y \in P^{\mathcal{J}}$. So then if $(x, y) \in r_j$, it holds that $x, y \in P^{\mathcal{J}} = Ind(K)$.

Since the *"smaller than"* relation is well-founded (as shown in Lemma 1), the existence of any grounded model of K wrt to M, implies the existence of a minimal grounded model as well. Hence (K, M) must have a GC-model. ♣

Corollary 1 Every model of K_M is a grounded model of K wrt to M. ⊣

One observation worth mentioning here is that grounding, although defined at a semantic level, can be internalized in the syntax and expressed as a particular class of knowledge bases. And through this grounding, a localization of the non-monotonicity is achieved, such that for the principal task of deciding satisfiability, we do not even need to expand reasoning beyond the already known algorithms that exist for standard DLs.

Chapter 5

Towards an Instance Checking Algorithm

For the task of determining whether or not a concept assertion (also refered to as 'fact') is entailed by a GC-$\mathcal{ALCO}$ knowledge base, knowing that a minimal model exists is not enough. We have to be able to find this model or at least to negate the possibility of a grounded model being minimal. We tried two approaches. The first one, which has a similar strategy with the algorithms presented in the original paper, was to find GC-models by looking for smaller models of arbitrary grounded models. In this direction are the results presented in Chapter 7. However what seems to be more efficient is a bottom-up approach, where the grounded models found first are definitely minimal.

5.1 Specification of the Configuration Space

The idea of defining independently what is essentially the search space of our algorithm, a space of possible choices of extensions to the closed predicates, is inspired by the original paper, where a similar set is specified. However, and this is one more clue which points to the divergence between the intended meaning of ground circumscription and what was initially defined, in the original paper the domain and the possible interpretations of the individuals over it are not taken into consideration when defining this space. Having improved the definition, we still need to add a dimension to the search space which will

correspond to the possible interpretations of the individual names.

Given an interpretation $\mathcal{I}$, the *individual allocation* of $\mathcal{I}$ is the set $\mathsf{AL}(\mathcal{I}) \in Part(Ind(K))$ such that every $X \in \mathsf{AL}(\mathcal{I})$ has the property: for every $a \in X$ and $b \in Ind(K)$ holds that $a^{\mathcal{I}} = b^{\mathcal{I}}$ if and only if $b \in X$.

Suppose that $I \in Part(Ind(K))$ and $a, b \in Ind(K)$. We call a and b *I-invariant* and write $a \simeq_I b$ if there is an $X \in I$ such that $a, b \in X$. A set $Z \subseteq Ind(K)$ is called *I-complete* if $a \in Z$ and $a \simeq_I b$ imply $b \in Z$. Similarly a set $V \in Ind^2(K)$ is called *I-complete* if $(a, b) \in V$ and $a \simeq_I a'$ and $b \simeq_I b'$ imply $(a', b') \in V$. For the sake of conciseness in the next definition, we define the following sets:

$$Cmp_I(K) = \{X \subseteq Ind(K) | X \text{ is } I\text{-complete}\}$$
$$Cmp_I^2(K) = \{Y \subseteq Ind^2(K) | Y \text{ is } I\text{-complete}\}$$

We can now employ the above notions to specify the search space of our algorithm:

Definition 6 Let (K, M) be a GC-$\mathcal{ALCO}$-KB, where $M \cap N_C = \{A_1, ..., A_n\}$ and $M \cap N_r = \{r_1, ..., r_m\}$. Then the set

$$\mathcal{G}_{(K,M)} = \Big\{(X_1, ..., X_n, Y_1, ..., Y_m, I) \Big| X_i \subseteq Cmp_I(K), Y_j \subseteq Cmp_I^2(K), I \in Part(Ind(K))\Big\}$$

is called *configuration space* of (K, M). ⊣

$\mathcal{G}_{(K,M)}$ is obviously a finite set. Every grounded model $\mathcal{I}$ of K wrt to M, corresponds to a point in the configuration space. In particular we will call the tuple

$$\Big(Ext^{\mathcal{I}}(A_1), ..., Ext^{\mathcal{I}}(A_n), Ext^{\mathcal{I}}(r_1), ..., Ext^{\mathcal{I}}(r_m), \mathsf{AL}(\mathcal{I})\Big)$$

the *assignment* of $\mathcal{I}$.

Let $G_1, G_2 \in \mathcal{G}_{(K,M)}$ with $G_1 = (Z_1, ..., Z_{n+m}, I)$ and $G_2 = (V_1, ..., V_{n+m}, I)$. We say that G_1 is *smaller than* G_2 and we write $G_1 \prec G_2$ if it holds that $Z_i \subseteq V_i$ for all $i \in \{1, ..., n+m\}$ and there exists $i \in \{1, ..., n+m\}$ such that $Z_i \subset V_i$. The following result then holds trivially:

Lemma 2 Let $\mathcal{I}, \mathcal{J}$ be grounded models of a knowledge base K wrt M and let $G_1, G_2 \in \mathcal{G}_{(K,M)}$ be their respective assignments. Then it holds that $\mathcal{I} \prec_M \mathcal{J}$ if and only if $G_1 \prec G_2$. ⊣

5.2 Binary Encoding & Linear Order

Let $\mathcal{G}_{(K,M)}$ be the configuration space of a GC-$\mathcal{ALCO}$-KB. Let $Ind(K) = \{a_1, ..., a_\mu\}$. For the purposes of the algorithm presented in the next section, we want to order $\mathcal{G}_{(K,M)}$ linearly. We achieve that by using a binary encoding for every $G \in \mathcal{G}_{(K,M)}$ and the lexicographical order. We first introduce an encoding $s : Part(Ind(K)) \to \mathbb{Z}_2^*$. Every partition of $Ind(K)$ can be specified by indicating which couples of individual names (that appear in the knowledge base) belong to the same block of the partition. This is easily percieved with the following visualization:

	a_1	a_2	a_3	...	a_μ
a_1	-	$s_{(1,2)}$	$s_{(1,3)}$	...	$s_{(1,\mu)}$
a_2	-	-	$s_{(2,3)}$	...	$s_{(2,\mu)}$
$\vdots$	-	-	-	$\ddots$	$\vdots$
$a_{\mu-1}$	-	-	-	-	$s_{(\mu-1,\mu)}$
a_μ	-	-	-	-	-

In accordance with the above table we define

$$s(I) = s_{(1,2)}s_{(1,3)}...s_{(1,\mu)}s_{(2,3)}s_{(2,4)}...s_{(2,\mu)}...s_{(\mu-1,\mu)}$$

where $s_{(i,j)} = 1$ if there exists $Z \in I$ with $a_i, a_j \in Z$, otherwise $s_{(i,j)} = 0$. We can now proceed to define the complete binary encoding of the points of the configuration space.

Let $\sigma : \mathscr{P}\Big(Ind(K)\Big) \cup \mathscr{P}\Big(Ind^2(K)\Big) \cup Part\Big(Ind(K)\Big) \to \mathbb{Z}_2^*$, with

$$\sigma(X) = \begin{cases} z_1 z_2 \dots z_\mu & \text{if } X \subseteq Ind(K) \text{ where } z_\kappa = \begin{cases} 1 & \text{if } a_\kappa \in X \\ 0 & \text{if } a_\kappa \notin X \end{cases} \\ z_1 z_2 \dots z_{\mu^2} & \text{if } X \subseteq Ind^2(K) \text{ where } z_{\kappa\mu+\lambda} = \begin{cases} 1 & \text{if } (a_\kappa, a_\lambda) \in X \\ 0 & \text{if } (a_\kappa, a_\lambda) \notin X \end{cases} \\ s(X) & \text{if } X \in Part(Ind(K)) \end{cases}$$

We can view words over $\mathbb{Z}_2$ as natural numbers encoded in the binary system. If $w_1, w_2 \in \mathbb{Z}_2^*$ are words of the same length, we write $w_1 < w_2$ if this relation

holds for the respective natural numbers. We can now define a total order '$<$' on $\mathcal{G}_{(K,M)}$.

Definition 7 Let $G_1 = (Z_1, ..., Z_k)$ and $G_2 = (V_1, ..., V_k)$ be two points in the configuration space of a *GC-$\mathcal{ALCO}$-KB* (K, M). G_1 *precedes* G_2 and we write $G_1 < G_2$ if there exists an $i \leq k$ such that $\sigma(Z_i) < \sigma(V_i)$ and for all $j < i$ holds $\sigma(Z_j) = \sigma(V_j)$. $\dashv$

For efficiency purposes, it is important here that the order defined above induces the partial order of minimality, so that the algorithm will discover the minimal model early on and discard searching in large sections of the configuration space. The following lemma ensures us that this is indeed the case.

Lemma 3 Let $\mathcal{G}_{(K,M)}$ be the configuration space of a *GC-$\mathcal{ALCO}$-KB*. For every $G_1, G_2 \in \mathcal{G}_{(K,M)}$ holds that $G_1 \prec G_2$ implies $G_1 < G_2$.

Proof: Let $G_1 = (Z_1, ..., Z_k)$ and $G_2 = (V_1, ..., V_k)$ with $G_1 \prec G_2$. By definition we have that $Z_k = V_k$. Because $Z_j \subseteq V_j$, we get that for every $j \in \{1, ..., k-1\}$, either $\sigma(Z_j) = \sigma(V_j)$ or $\sigma(Z_j) < \sigma(V_j)$. And because there is at least one $i \in \{1, ..., k-1\}$ such that $Z_i \subset V_i$, which implies $\sigma(Z_i) < \sigma(V_i)$, by chosing the first one we fulfill the conditions of the above definition, hence $G_1 < G_2$. ♣

5.3 Navigation within the Configuration Space

It is critical, given a point in the configuration space, to be able to construct a grounded model with such an assignment, if one exists. This is accomplished by adding the axioms specified in the next definition.

Definition 8 Let (K, M) be a *GC-$\mathcal{ALCO}$-KB*, where $M \cap N_C = \{A_1, ..., A_n\}$ and $M \cap N_r = \{r_1, ..., r_m\}$. Let $G = (X_1, ..., X_n, Y_1, ..., Y_m, I)$ be a point in the configuration space of (K, M). We define K_G as the $\mathcal{ALCO}-KB$ which consists of all the axioms that are included in K_M as well as the following ones:

- $\{a\} \equiv \{b\}$ for all $a, b \in Ind(K)$ with $a \simeq_I b$,
- $\{a\} \sqsubseteq \neg\{b\}$ for all $a, b \in Ind(K)$ with $a \not\simeq_I b$,
- $A_i \equiv X_i$ for every $i \in \{1, ..., n\}$,
- $R_{(a,j)} \equiv \{c \in Ind(K) | (a, c) \in Y_j\}$ for every $a \in Ind(K)$ and $j \in \{1, ..., m\}$,
- $N_{(a,j)} \equiv \{c \in Ind(K) | (c, a) \in Y_j\}$ for every $a \in Ind(K)$ and $j \in \{1, ..., m\}$,
- $\{a\} \sqsubseteq \forall r_j.R_{(a,j)}$ for every $a \in Ind(K)$ and $j \in \{1, ..., m\}$,
- $\exists r_j.\{a\} \equiv N_{(a,j)}$ for every $a \in Ind(K)$ and $j \in \{1, ..., m\}$.

K_G is then called a *pointwise restriction* of (K, M). $\dashv$

Lemma 4 Let K_G be a pointwise restriction of a *GC*-$\mathcal{ALCO}$ knowledge base (K, M). The following statements hold:

i) If $\mathcal{I}$ is a model of K_G, then G is the assignment of $\mathcal{I}$.

ii) If there exists a model of K_M with assignment G, then K_G is satisfiable.

Proof: i) From the first two axioms we get that $a^{\mathcal{I}} = b^{\mathcal{I}}$ if and only if $a \simeq_I b$, which entails $\mathsf{AL}(\mathcal{I}) = I$. Then we observe that because for all $i \in \{1, ..., n\}$, X_i are I-complete, it holds that $Ext^{\mathcal{I}}(X_i) = X_i$. Hence from the third axiom above, we have that $A_i^{\mathcal{I}} = X_i^{\mathcal{I}} \Rightarrow Ext^{\mathcal{I}}(A_i) = Ext^{\mathcal{I}}(X_i) \Rightarrow Ext^{\mathcal{I}}(A_i) = X_i$.

Let $(a, b) \in Ext^{\mathcal{I}}(r_j)$ for some $j \in \{1, ..., m\}$. Then $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r_j^{\mathcal{I}}$ and by the axiom $\{a\} \sqsubseteq \forall r_j.R_{(a,j)}$ we get that $b^{\mathcal{I}} \in \{c \in Ind(K) | (a, c) \in Y_i\}^{\mathcal{I}}$. So there exists a $c \in Ind(K)$ such that $b^{\mathcal{I}} = c^{\mathcal{I}}$ and $(a, c) \in Y_j$. But Y_j is I-complete and so since $b \simeq_I c$ we conclude that $(a, b) \in Y_j$. Now let $(a, b) \in Y_j$. Then $a \in N_{(b,j)} \Rightarrow a^{\mathcal{I}} \in (N_{(b,j)})^{\mathcal{I}}$ and from the last axiom we get that $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r_j^{\mathcal{I}}$, so $(a, b) \in Ext^{\mathcal{I}}(r_j)$. As a result we get that $Y_j = Ext^{\mathcal{I}}(r_j)$.

To conclude, we have shown that all X_i and Y_j are equal with the ground extensions of the respective predicates and also that I is the individual allocation of $\mathcal{I}$. Therefore G is the assignment of $\mathcal{I}$.

ii) Let $\mathcal{I}'$ be a model of K_M with assignment G. We construct $\mathcal{I}$ by interpreting everything as in $\mathcal{I}'$, with the addition of

$R^{\mathcal{I}}_{(a,j)} = \{c \in Ind(K) | (a,c) \in Y_j\}^{\mathcal{I}}$ for every $a \in Ind(K)$ and $j \in \{1,...,m\}$,
$N^{\mathcal{I}}_{(a,j)} = \{c \in Ind(K) | (c,a) \in Y_j\}^{\mathcal{I}}$ for every $a \in Ind(K)$ and $j \in \{1,...,m\}$,

So for every $i \in \{1,...,n\}$ and $j \in \{1,...,m\}$ we have that $X_i = Ext^{\mathcal{I}}(A_i)$ and $Y_j = Ext^{\mathcal{J}}(r_j)$. Moreover, we also have that $\mathsf{AL}(\mathcal{I}) = I$ so $a^{\mathcal{I}} = b^{\mathcal{I}}$ if and only if $a \simeq_I b$ hence the first two axioms hold in $\mathcal{I}$. Because A_i is closed wrt K, we get that $A_i^{\mathcal{I}} = (Ext^{\mathcal{I}}(A_i))^{\mathcal{I}}$, thus $\mathcal{I}$ satisfies the third axiom as well. That the next two axioms hold in $\mathcal{I}$ is an immediate consequence of its definition. Consequently, because r_j is closed wrt K, we get that for every $a \in Ind(K)$, all $y \in \Delta^{\mathcal{I}}$ with $(a^{\mathcal{I}}, y) \in r_j^{\mathcal{I}}$ correspond to named individuals, hence they belong to the set $\{c \in Ind(K) | (a^{\mathcal{I}}, c^{\mathcal{I}}) \in r_j^{\mathcal{I}}\} = \{c \in Ind(K) | (a,c) \in Ext^{\mathcal{I}}(r_j)\}$ so the axiom $\{a\} \sqsubseteq \forall r_j.R_{(a,j)}$ is satisfied. Similarly

$$(\exists r_j.\{a\})^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | \exists y \in \Delta^{\mathcal{I}} : (x,y) \in r_j^{\mathcal{I}} \text{ and } y = a^{\mathcal{I}}\}$$

but since r_j is closed wrt K

$$(\exists r_j.\{a\})^{\mathcal{I}} = \{c^{\mathcal{I}} \in Ind(K) | (c^{\mathcal{I}}, a^{\mathcal{I}}) \in r_j^{\mathcal{I}}\} = N^{\mathcal{I}}_{(a,j)}$$

so the last axiom holds as well and thus $\mathcal{I}$ is a model of K_G. ♣

Chapter 6

Instance Checking for a GC Knowledge Base

In this chapter we specify an algortihm for deciding whether or not a GC knowledge base entails an assertion. Furthermore, we give an example and discuss complexity and possibility of further use and development.

6.1 The Algorithm

Let (K, M) be a GC-$\mathcal{ALCO}$-KB, where $M \cap N_C = \{A_1, ..., A_n\}$ and $M \cap N_r = \{r_1, ..., r_m\}$. We want to check if an assertion $B(a)$ is a logical conseqence of (K, M). Such a reasoning task is commonly refered to as *instance checking*, hence the title of this section. If $a \notin Ind(K)$ the answer is trivial, so for the rest it assumed that $a \in Ind(K)$. We split the decision procedure in two cases, the first of which will prove to be solvable in a much more simple way, by only once calling the "oracle" $\mathcal{ALCO}$ reasoner. In the following, given a knowledge base K_0, we use the notation $K_0^+ := K_0 \cup \{\neg B(a)\}$ to refer to K_0 augmented with the negation of the assertion we are checking for entailment.

Case 1: $B \in M$

Proposition 2 K_M^+ is unsatisfiable if and only if (K, M) entails $B(a)$.

Proof: We use contraposition for proving both directions of the above equivalence.

($\Rightarrow$:) Let $\mathcal{I}$ be a GC-model of (K, M) such that $\mathcal{I}$ satisfies $\neg B(a)$. Then it is obviously also a model of K_M. And since $a^{\mathcal{I}} \notin B^{\mathcal{I}}$, $\mathcal{I}$ is also a model of K_M^+.

($\Leftarrow$:) Let $\mathcal{I}$ be a model of K_M^+. It is thus also a model of the smaller KB K_M. But $\mathcal{I}$ satisfies $\neg B(a)$ so if it is minimal wrt M, we have found a counterexample for entailment of $B(a)$ from (K, M). If $\mathcal{I}$ is not minimal, then as we have shown in Proposition 1, there exists a GC-model $\mathcal{J}$ of (K, M) such that $\mathcal{J} \prec_M \mathcal{I}$. Since $B \in M$, according to condition ii) of Definition 2 we have $Ext^{\mathcal{J}}(B) \subseteq Ext^{\mathcal{I}}(B)$, hence $a \notin B^{\mathcal{I}}$ implies $a \notin B^{\mathcal{J}}$. So $\mathcal{J}$ satisfies $\neg B(a)$, thus (K, M) does not entail $B(a)$. ♣

Case 2: $B \notin M$

We want to determine if every GC-model of (K, M) entails $B(a)$. To achieve that, we navigate bottom up in the configuration space which is essentially the space of possible individual allocations and ground extensions to the predicates in M. Let $\mathcal{G}_{(K,M)} = \{G_1, ..., G_\lambda\}$, where $G_1 < G_2 < ... < G_\lambda$.

IC Algorithm:

```
1] Initiate Stack := G_(K,M).
2] for i = 1 to λ
3]     If G_i ∈ Stack:
4]         Check K_{G_i} for satisfiability.
5]             If YES:
6]                 Check K^+_{G_i} for satisfiability.
7]                     If YES return FALSE.
8]                     Else remove all G_j ≻ G_i from Stack.
9] return TRUE.
```

That the above algorithm terminates is obvious, because there is only one loop. Moreover the command in line 9, outside of the loop, guarantees that it will return either TRUE or FALSE.

Proposition 3 The IC algorithm returns TRUE if and only if (K, M) entails $B(a)$.

Proof: Once again we prove the contrapositive.

($\Rightarrow$:) Suppose (K, M) does not entail $B(a)$. So there is at least one GC-model $\mathcal{I}$ of (K, M) where $a \notin B^{\mathcal{I}}$. Let G be the assignment of $\mathcal{I}$. We first show by cotradiction that K_G^+ must be one of the knowledge bases checked for satisfiability. If it was not, G must have been one of the elements removed from the `Stack` at line 8 of the algorithm. Then there would exist a $G' \prec G$ such that $K_{G'}$ is satisfiable, say with $\mathcal{I}'$ being its model. But $\mathcal{I}'$ is then also a model of K_M, hence a grounded model of K wrt M and $\mathcal{I}' \prec_M \mathcal{I}$. But that can not be, because $\mathcal{I}$ is minimal. Therefore we know that the IC algorithm will check K_G^+ for satisfiability. But K_G^+ is satisfiable, $\mathcal{I}$ being a model of it, so the algorithm will return FALSE.

($\Leftarrow$:) We assume that the IC algorithm returns FALSE. Hence for some $i \in \{1, ..., \lambda\}$, $K_{G_i}^+$ is checked for satisfiability and found to have a model $\mathcal{I}$. But $\mathcal{I}$ is also a model of K_M (because $K_M \subset K_{G_i}^+$). Suppose $\mathcal{I}$ is not a GC-model. Then there exists $\mathcal{I}' \prec_M \mathcal{I}$. Let G' be the assignment of $\mathcal{I}'$. Then $G' \prec G_i$ hence from Lemma 3 we get $G' < G_i$. So since G' precedes G_i, either $K_{G'}$ is checked for satisfiability by the algorithm, or it is not because there exists a $G'' < G'$ such that $K_{G''}$ is found to be satisfiable and G' is then deleted from the `Stack`. Both cases would result in a model with assignment smaller than G_i. But then G_i by the command in Line 8 must have been removed from the `Stack`. That is contradictory, hence $\mathcal{I}$ is a GC-model of (K, M). However $\mathcal{I}$ satisfies $\neg B(a)$, so (K, M) does not entail $B(a)$. ♣

To demonstrate how this whole procedure works, we give a simple example.

6.2 Example

Let K be the following knowledge base:

$$B(a),\ \ \neg B(b),\ \ r(b,c),\ \ \rho(a,b),\ \ \rho(a,c),\ \ \exists r.\neg A \sqsubseteq A$$

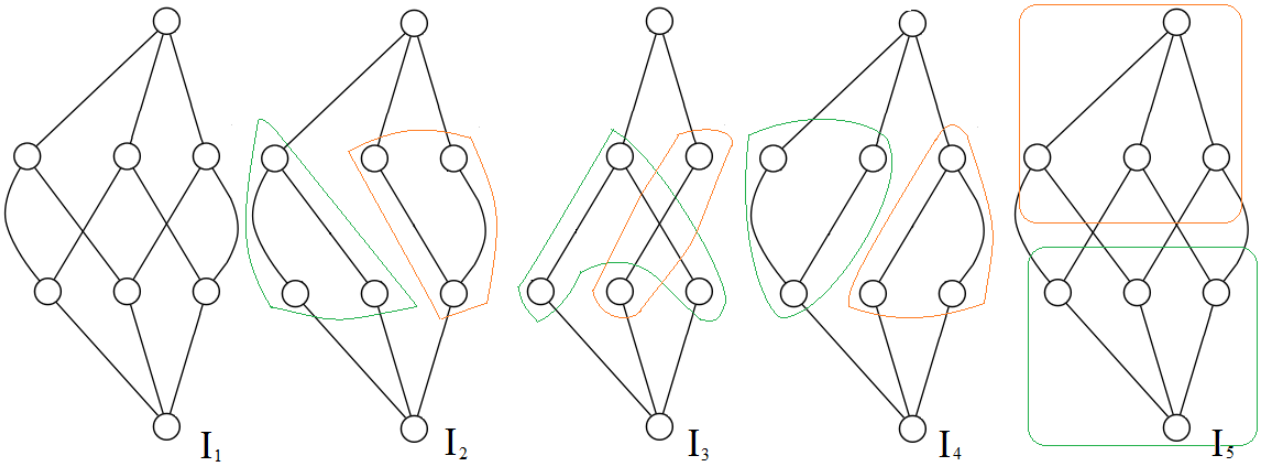

Figure 6.1: The configuration space of (K, M).

And let $M = \{A\}$. Then $Part(Ind(K)) = \{I_1, I_2, I_3, I_4, I_5\}$ where

$$
\begin{aligned}
I_1 &= \{\{a\}, \{b\}, \{c\}\} \\
I_2 &= \{\{a, b\}, \{c\}\} \\
I_3 &= \{\{a, c\}, \{b\}\} \\
I_4 &= \{\{a\}, \{b, c\}\} \\
I_5 &= \{\{a, b, c\}\}.
\end{aligned}
$$

Figure 6.1 is a visualization of our configuration space. Each possible individual allocation corresponds to a lattice of possible ground extensions for the closed predicates, which in our case consists of just A. The restriction of the search space to I-complete sets of possible extensions, with respect to an individual allocation I, is portrayed by the groupings of points in green and orange enclosure. Basically, what occurs there is a reduction to the search space, as what used to be three or four points contract to only one.

Suppose that we want to check the assertion $\neg(A \sqcap \forall\rho.A)(a)$ for entailment. This basically means that not all individuals can be interpreted as members of the extension of A. Then the IC algorithm will look bottom-up for grounded models of K wrt M. If a model is found, then an augmented knowledge base will be built, consisting of the current pointwise restriction and the negation of the given assertion, which in our case is just $(A \sqcap \forall.\rho A)(a)$. In case this augmented KB is found to be satisfiable, the algorithm will halt, giv-

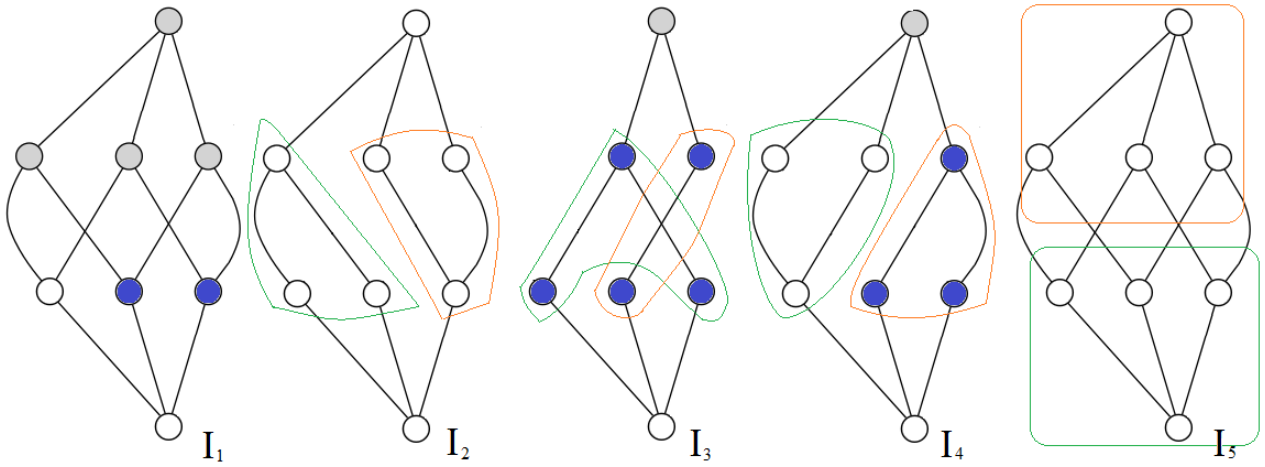

Figure 6.2: The distribution of GC-models of (K, M) in the configuration space.

ing FALSE as an answer. Otherwise it will remove from the `Stack` all points which are above, hence reducing further the remaining exploration. For this particular instance, (K, M) entails $\neg(A \sqcap \forall.\rho A)(a)$, so no minimal model which satisfies $(A \sqcap \forall.\rho A)(a)$ can be found, and so the algorithm will return TRUE.

Note that the entailment holds exactly because of the minimality, i.e. there are grounded models where all individuals belong to A. Figure 6.2 gives an account of the distribution of grounded models and GC-models of (K, M) over the configuration space. Points in white are those that do not correspond to any model of K_M, points in blue correspond to GC-models and points in grey to the rest of the grounded models. All the points in grey are exactly those that will be never "visited", i.e. at some step they will be removed from the `Stack`.

6.3 Discussion about Complexity

By the distributive character of the set $\mathcal{G}_{(K,M)}$ which serves as the initial `Stack`, we can see that our algorithm is in EXPSPACE. However it is possible that the worst case complexity is lower. Considering that at most exponentially many calls of the $\mathcal{ALCO}$ reasoner are needed, each of which requires EXPTIME, we get that the overall complexity is still EXPTIME.

Moreover, by the removal of points that results from the command in line

8, we conjecture that the average complexity is considerably reduced. That is because the algorithm, in accordance with the defined linear order, will try smaller points of the configuration space first and once a model is found, the algorithm will stop looking at the rest of the branch. In the prevous section's example, this advantage of the algorithm is not highlighted enough. Only 6 of the 22 points of the configuration space are removed in the process described in the above paragraph. This is a result of the simplicity of the particular case. But had it been for a much larger configuration space, models would probably be found on the lower positions of the lattices of the individual allocations, hence big parts of those lattices would directly be removed from the search space.

Of course there is room for optimization of this algorithm. Notably from the example we can see how two out of the five lattices should have been rejected from the start, since they represent individual allocations which are incompatible with the given knowledge base. More thoroughly, one could remove points which correspond to assignments which are not consistent with the axioms in the knowledge base. But in order to achieve this, a solid case-by-case analysis should be made, which would then involve to a greater degree and depend upon the specific choice of DL formalism.

On the other hand the results we have acquired so far are directly extendable to more complex languages. That follows from the fact that in none of the proofs presented in this study have we invoked the limitations of $\mathcal{ALCO}$. In effect, we have used the constructive capabilities of our language, in creating new knowledge bases that represent the notion of grounding and different points in the configuration space. But we have not appealed to any restrictions imposed by the specific syntax of $\mathcal{ALCO}$, with the exception of course of the property of decidabilty, which is implicit wherever a decision procedure is regarded.

Chapter 7

Auxiliary Theory

Minimality Check: A Non-standard Reasoning Task

In this chapter we present one of the more involved results we came up with, as far as the intricacy of the proof is concerned. It is a solution to the task of determining whether a specifc grounded model is minimal by calling the standard DL reasoner just once. It can be of use in devising algorithms for other reasoning problems in grounded circumscription, but also maybe in some optimized variant of the IC algorithm presented previously.

Definition 9 Let K_M be a grounded KB where $M \cap N_C = \{A_1, ..., A_n\}$ and $M \cap N_r = \{r_1, ..., r_m\}$ and let $\mathcal{I}$ be a model of K_M. We call *down-the-chain axioms* with respect to $\mathcal{I}$, the following set of GCIs:

I. $\{a\} \equiv Ext^{\mathcal{I}}(\{a\})$ for every $a \in Ind(K)$,

II. $Ext^{\mathcal{I}}(\neg\{a\}) \sqsubseteq \neg\{a\}$ for every $a \in Ind(K)$,

III. $A_i \sqsubseteq Ext^{\mathcal{I}}(A_i)$ for every $i \in \{1, ..., n\}$,

IV. $B_{(a,j)} \equiv \{c \in Ind(K) | (a^{\mathcal{I}}, c^{\mathcal{I}}) \in r_j^{\mathcal{I}}\}$ for every $a \in Ind(K)$ and $j \in \{1, ..., m\}$,

V. $\{a\} \sqsubseteq \forall r_j.B_{(a,j)}$ for every $a \in Ind(K)$ and $j \in \{1, ..., m\}$,

VI. $\top \sqsubseteq \exists r.\Big(\bigsqcup_{i\in\{1,\ldots,n\}}\Big(Ext^{\mathcal{I}}(A_i)\sqcap\neg A_i\Big)\Big)\sqcup\Big(\bigsqcup_{\substack{j\in\{1,\ldots,m\}\\ a\in Ind(K)}}\Big(\bigsqcup_{c\in B_{(a,j)}}\Big(\{a\}\sqcap\forall r_j.\neg\{c\}\Big)\Big)\Big)$,

where r is a fresh role, i.e. it does not appear in K. K_M augmented with the down-the-chain axioms with respect to a model is called a *confining* of K_M and symbolized $K_M^{\mathcal{I}-}$, where $\mathcal{I}$ is the respective model. ⊣

Notice that the number of axioms in each of the categories I-IV depends on M whereas V is one single axiom. The next lemma shows how we can find a smaller grounded model than a given one, if there exists one. Intuitively this is like going down in the lattice of possible grounded models, hence the terminology.

Lemma 5 Let (K, M) be a *GC-$\mathcal{ALCO}$-KB* and let $\mathcal{I}$ be a model of K_M. There exists a model $\mathcal{J}$ of K_M such that $\mathcal{J} \prec_M \mathcal{I}$ if and only if $K_M^{\mathcal{I}-}$ is satisfiable.

Proof: ($\Rightarrow$:) Let $\mathcal{J}$ be a model of $K_M^{\mathcal{I}-}$. $\mathcal{J}$ is then obviously also a model of K_M. Let $b \in Ext^{\mathcal{I}}(\{a\})$ for some $a \in Ind(K)$. Then $b^{\mathcal{J}} \in (Ext^{\mathcal{I}}(\{a\}))^{\mathcal{J}}$ and by axioms in I, we get that $b^{\mathcal{J}} \in \{a\}^{\mathcal{J}} \Rightarrow b^{\mathcal{J}} = a^{\mathcal{J}}$, hence $b \in Ext^{\mathcal{J}}(\{a\})$ and we have proven that $Ext^{\mathcal{I}}(\{a\}) \subseteq Ext^{\mathcal{J}}(\{a\})$. Conversely, if $b \in Ext^{\mathcal{J}}(\{a\})$ then $b^{\mathcal{J}} = a^{\mathcal{J}}$. If we assume that $b \notin Ext^{\mathcal{I}}(\{a\})$, we get that $b^{\mathcal{I}} \neq a^{\mathcal{I}}$ and so $b \in Ext^{\mathcal{I}}(\neg\{a\})$, thus $b^{\mathcal{J}} \in (Ext^{\mathcal{I}}(\neg\{a\}))^{\mathcal{J}}$. Then, from the axioms in category II we infer $b^{\mathcal{J}} \in \neg\{a\}^{\mathcal{J}} \Rightarrow b^{\mathcal{J}} \notin \{a^{\mathcal{J}}\} \Rightarrow b^{\mathcal{J}} \neq a^{\mathcal{J}}$ and we have reached a contradiction, so it must be that $b \in Ext^{\mathcal{I}}(\{a\})$. Hence $Ext^{\mathcal{J}}(\{a\}) \subseteq Ext^{\mathcal{I}}(\{a\})$ and conclusively $Ext^{\mathcal{I}}(\{a\}) = Ext^{\mathcal{J}}(\{a\})$, which fulfills the first condition of Definition 2.

Now let $A \in M \cap N_C$ and suppose that $b \in Ext^{\mathcal{J}}(A)$. Then $b^{\mathcal{J}} \in A^{\mathcal{J}}$, hence from the axioms of category III we get that $b^{\mathcal{J}} \in (Ext^{\mathcal{I}}(A))^{\mathcal{J}}$, but

$$Ext^{\mathcal{I}}(A) = \bigsqcup_{a\in Ext^{\mathcal{I}}(A)} \{a\}$$

and

$$\Big(Ext^{\mathcal{I}}(A)\Big)^{\mathcal{J}} = \bigcup_{a\in Ext^{\mathcal{I}}(A)} \{a\}^{\mathcal{J}}$$

hence there exists $a \in Ext^{\mathcal{I}}(A)$ such that $b^{\mathcal{J}} \in \{a\}^{\mathcal{J}}$, so $b \in Ext^{\mathcal{J}}(\{a\})$ and using the equality proven in the above paragraph we get $b \in Ext^{\mathcal{I}}(\{a\})$. But $\{a\}^{\mathcal{I}} \subseteq A^{\mathcal{I}}$ therefore due to the monotonicity of ground extension with respect to set inclusion we have that $b \in Ext^{\mathcal{I}}(A)$. Hence $Ext^{\mathcal{J}}(A) \subseteq Ext^{\mathcal{I}}(A)$.

Likewise, assume $\rho \in M \cap N_r$ and let $(a, b) \in Ext^{\mathcal{J}}(\rho)$, hence $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in \rho^{\mathcal{J}}$. But from the axioms IV we conclude that

$$a^{\mathcal{J}} \in \left\{ y \in \Delta^{\mathcal{J}} \middle| \forall z \in \Delta^{\mathcal{J}} : (y, z) \in \rho^{\mathcal{J}} \Rightarrow z \in B^{\mathcal{J}}_{(x,j)} \right\}$$

hence $b^{\mathcal{J}} \in B^{\mathcal{J}}_{(x,j)} = \{c \in Ind(K) | (a^{\mathcal{I}}, c^{\mathcal{I}}) \in \rho^{\mathcal{I}}\}^{\mathcal{J}}$, which means that there exists $c \in Ind(K)$ with $b^{\mathcal{J}} = c^{\mathcal{J}}$ and $(a^{\mathcal{I}}, c^{\mathcal{I}}) \in \rho^{\mathcal{I}}$. So $Ext^{\mathcal{J}}(\{b\}) = Ext^{\mathcal{J}}(\{c\})$, but as we proved earlier $Ext^{\mathcal{J}}(\{b\}) = Ext^{\mathcal{I}}(\{b\})$ thus $Ext^{\mathcal{I}}(\{b\}) = Ext^{\mathcal{I}}(\{c\})$ and $b^{\mathcal{I}} = c^{\mathcal{I}}$. Therefore $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in \rho^{\mathcal{I}}$ so $(a, b) \in Ext^{\mathcal{I}}(\rho)$. We have thence shown that $Ext^{\mathcal{J}}(\rho) \subseteq Ext^{\mathcal{I}}(\rho)$.

Conjoining the above results we can say that for every $W \in M$ it holds that $Ext^{\mathcal{J}}(W) \subseteq Ext^{\mathcal{I}}(W)$, so the second condition of Definition 2 is also met.

To prove the satisfaction of the last condition as well, we analyze axiom V. Role r here is purely auxiliary, invoked only to assure the satisfiability of the complex concept that follows. So for this concept inclusion to hold, for every element y of $\Delta^{\mathcal{J}}$ there must exist an r-neighbor x, such that either there is an $i \in \{1, ..., n\}$ with $x^{\mathcal{J}} \in (Ext^{\mathcal{I}}(A_i))^{\mathcal{J}} \setminus A_i^{\mathcal{J}}$, or there exist $j \in \{1, ..., m\}$, $a \in Ind(K)$ and $c \in B_{(a,j)}$ such that $a^{\mathcal{J}} = x$ and for every r_j-neighbor z of x it holds that $z \neq c^{\mathcal{J}}$. In the first case, from $x^{\mathcal{J}} \in (Ext^{\mathcal{I}}(A_i))^{\mathcal{J}} \setminus A_i^{\mathcal{J}}$, because $(Ext^{\mathcal{I}}(A_i))^{\mathcal{J}} = \bigcup_{a \in Ext^{\mathcal{I}}(A_i)} \{a\}^{\mathcal{J}}$, we get that there is an $a \in Ext^{\mathcal{I}}(A_i)$ such that $x^{\mathcal{J}} = a^{\mathcal{J}}$ but $a^{\mathcal{J}} \notin A_i^{\mathcal{J}}$, hence $a \notin Ext^{\mathcal{J}}(A_i)$, which implies that $Ext^{\mathcal{I}}(A_i) \subset Ext^{\mathcal{J}}(A_i)$. In the second case, from $c \in B_{(a,j)}$ we get that $(a, c) \in Ext^{\mathcal{I}}(r_j)$. And $(x, z) = (a^{\mathcal{J}}, c^{\mathcal{J}})$, so $(a^{\mathcal{J}}, c^{\mathcal{J}}) \notin r_j^{\mathcal{J}}$, hence $(a, c) \notin Ext^{\mathcal{J}}(r_j)$, thus we derive $Ext^{\mathcal{J}}(r_j) \subset Ext^{\mathcal{I}}(r_j)$. Therefore, conclusively we can say that there exists a $W \in M$ such that $Ext^{\mathcal{J}}(W) \subset Ext^{\mathcal{I}}(W)$.

Consequently all the conditions of the definition of the "smaller than" relation are fulfilled, thus $\mathcal{J} \prec_M \mathcal{I}$.

($\Leftarrow$:) Let $\mathcal{J}'$ be a model of K_M such that $\mathcal{J}' \prec_M \mathcal{I}$. Then there must be a $W \in M$ such that $Ext^{\mathcal{J}'}(W) \subset Ext^{\mathcal{I}}(W)$. Hence there exists $b \in Ind(K)$

such that either for some $A \in M \cup N_C$, $b^{\mathcal{I}} \in A^{\mathcal{I}}$ and $b^{\mathcal{J}'} \notin A^{\mathcal{J}'}$, or for some $\rho \in M \cup N_r$ there is a $d \in Ind(K)$ with $(b^{\mathcal{I}}, d^{\mathcal{I}}) \in \rho^{\mathcal{I}}$ and $(b^{\mathcal{J}'}, d^{\mathcal{J}'}) \notin \rho^{\mathcal{J}'}$. Let $\mathcal{J}$ be the interpretation which agrees with $\mathcal{J}'$ in everything except for the predicates $B_{(a,j)}$ and r, where the following hold:

$$B^{\mathcal{J}}_{(a,j)} = \{c \in Ind(K) | (a^{\mathcal{I}}, c^{\mathcal{I}}) \in r^{\mathcal{I}}_j\}^{\mathcal{J}}$$
$$r^{\mathcal{J}} = \{(x, b^{\mathcal{J}}) | x \in \Delta^{\mathcal{J}}\}$$

for all $a \in Ind(K)$ and $j \in \{1, ..., m\}$. Of course then $\mathcal{J}$ is also a model of K_M, smaller than $\mathcal{I}$. To demonstrate that $\mathcal{J}$ is a model of $K^{\mathcal{I}-}_M$ we go through all the axiom categories:

I) $Ext^{\mathcal{J}}(\{a\}) = Ext^{\mathcal{I}}(\{a\})$ so it suffices if $\{a\}^{\mathcal{J}} = (Ext^{\mathcal{J}}(\{a\}))^{\mathcal{J}}$ which is equivalent to $a^{\mathcal{J}} = \{c \in Ind(K) | a^{\mathcal{J}} = c^{\mathcal{J}}\}^{\mathcal{J}}$ which clearly holds.

II) We notice that $Ext^{\mathcal{I}}(\neg\{a\}) = Ind(K) \setminus Ext^{\mathcal{I}}(\{a\})$, hence $Ext^{\mathcal{I}}(\neg\{a\}) = Ext^{\mathcal{J}}(\neg\{a\})$. And we have $\{c^{\mathcal{J}} | c \in Ind(K), c^{\mathcal{J}} \neq a^{\mathcal{J}}\} \subseteq \{x \in \Delta^{\mathcal{J}} | x \neq a^{\mathcal{J}}\}$ so $(Ext^{\mathcal{J}}(\neg\{a\}))^{\mathcal{J}} \subseteq (\neg\{a\})^{\mathcal{J}}$ and thus the axiom holds.

III) Because A_i is closed wrt K, we get that $A_i^{\mathcal{J}} = (Ext^{\mathcal{J}}(A_i))^{\mathcal{J}}$, and from $(Ext^{\mathcal{J}}(A_i))^{\mathcal{J}} \subseteq (Ext^{\mathcal{I}}(A_i))^{\mathcal{J}}$ we derive that $A_i^{\mathcal{J}} \subseteq (Ext^{\mathcal{J}}(A_i))^{\mathcal{J}}$.

IV) This holds trivially by the definition of $\mathcal{J}$.

V) Because r_j is closed wrt K, we get that for every $a \in Ind(K)$, all $y \in \Delta^{\mathcal{J}}$ with $(a^{\mathcal{J}}, y) \in r_j^{\mathcal{J}}$ correspond to named individuals, hence they belong to the set $\{c \in Ind(K) | (a^{\mathcal{J}}, c^{\mathcal{J}}) \in r_j^{\mathcal{J}}\}$. But we have that $Ext^{\mathcal{J}}(r_j) \subseteq Ext^{\mathcal{I}}(r_j)$ so we have $\{c \in Ind(K) | (a^{\mathcal{J}}, c^{\mathcal{J}}) \in r_j^{\mathcal{J}}\} \subseteq B_{(a,j)}$. Therefore

$$a^{\mathcal{J}} \in \{x \in \Delta^{\mathcal{J}} | \text{ for all } y \in \Delta^{\mathcal{J}} \text{ with } (x, y) \in r_j^{\mathcal{J}} \text{ holds } y \in B^{\mathcal{J}}_{(a,j)}\}$$

and so the axioms hold.

VI) By our definition of $\mathcal{J}$, we just need to show that b satisfies the concept

$$\left(\bigsqcup_{i \in \{1,...,n\}} \left(Ext^{\mathcal{I}}(A_i) \sqcap \neg A_i \right) \right) \sqcup \left(\bigsqcup_{\substack{j \in \{1,...,m\} \\ a \in Ind(K)}} \left(\bigsqcup_{c \in B_{(a,j)}} \left(\{a\} \sqcap \forall r_j . \neg\{c\} \right) \right) \right)$$

We can see that one of the two disjuncts will be fulfilled depending on how b was chosen. If there is an $A \in M \cup N_C$ such that $b^{\mathcal{I}} \in A^{\mathcal{I}}$ and $b^{\mathcal{J}} \notin A^{\mathcal{J}}$, then $b^{\mathcal{J}} \in \Big(Ext^{\mathcal{I}}(A) \sqcap \neg A\Big)^{\mathcal{J}}$. Otherwise if for some $\rho \in M \cup N_r$ there is a $d \in Ind(K)$ with $(b^{\mathcal{I}}, d^{\mathcal{I}}) \in \rho^{\mathcal{I}}$ and $(b^{\mathcal{J}}, d^{\mathcal{J}}) \notin \rho^{\mathcal{J}}$ then we have $b^{\mathcal{J}} \in \Big(\{b\} \sqcap \forall \rho.\neg\{d\}\Big)^{\mathcal{J}}$, hence the concept is satisfied in any case.

We have thus shown that $\mathcal{J}$ is a model of $K_M^{\mathcal{I}-}$. ♣

For direct practical use, the above lemma is more conveniently expressed in the following form:

Corollary 2 *(Minimality Check)* Let (K, M) be a *GC-$\mathcal{ALCO}$-KB* and let $\mathcal{I}$ be a model of K_M. If $K_M^{\mathcal{I}-}$ is unsatisfiable, then $\mathcal{I}$ is a GC-model of (K, M).⊣

Chapter 8

Ground Circumscription in the Original Paper

In this chapter, we follow the definitions and algorithms specified in the original paper. Our purpose is to discuss the insufficiencies of the approach of the original paper and to defend and support our choice of modifying the definition of grounded circumscription. To this end, we also give a counterexample in order to indicate a flaw in what is presented to be an instance checking algorithm.

8.1 The Algorithm for Instance Checking

Firstly we see that *instance checking* at that paper is meant only for checking assertions $C(a)$ where C is a concept name, in contrast to the typical meaning of the term, in which C is an arbitrary concept. But even this restricted version of instance checking is not possible with the algorithms provided.

In the paper, two algorithms are given, the first one, *Tableau1*, given a GC-knowledge base, finds a grounded model and the second one, *Tableau2*, given a grounded model, finds a smaller model if there exists one. Then a third algorithm is defined by putting together *Tableau1* and *Tableau2*, in order to find a GC-model if there exists one. It is called *GC-model finder*. This algorithm does not actually produce all different kinds of GC-models, as is the claim in the publication upon which their *instance checking* algorithm is based.

As a counterexample, consider the followiing GC-knowledge base:

$$(K, M) = \Big(\{r(a, b), \exists r.\top \sqsubseteq A \sqcup \exists r.A, C \equiv \neg A \sqcup \forall r.\neg A\}, \{A\}\Big)$$

Given (K, M), the initial graph for the algorithm *Tableau1* would consist of two nodes, one for each individual appearing in K. Since *Tableau2* retains the same domain that is the output of *Tableau1*, every GC-model produced by the *GC-model finder* will have a domain of at least two. By the axioms in K it is easy to see that in that case, in order to minimize the extension of A, every GC-model $\mathcal{I}$ will have the property that exactly one of a,b will be mapped to elements belonging to $A^{\mathcal{I}}$. Hence $C(a)$ will hold. However, the following interpretation is also a GC-model:

$$\begin{aligned}
\Delta^{\mathcal{J}} &= \{0\} \\
a^{\mathcal{J}} &= 0 \\
b^{\mathcal{J}} &= 0 \\
A^{\mathcal{J}} &= \Delta^{\mathcal{J}} \\
C^{\mathcal{J}} &= \emptyset \\
r^{\mathcal{J}} &= \Delta^{\mathcal{J}} \times \Delta^{\mathcal{J}}
\end{aligned}$$

and $\mathcal{J}$ does not satisfy $C(a)$. Hence although (K, M) does not entail $C(a)$, the *instance checking* algorithm will answer positively in this circumstance.

8.2 The Definition of Minimality

Certainly, the problem spotted in the *GC-model finder* could be corrected if in the *Tableau1* algorithm, in a non-deterministic fashion, we allowed for the initial graph to consist of any number of nodes smaller than or equal to the number of individuals in the knowledge base, and then also non deterministically distributed the individuals over the nodes. Nevertheless, the nature of this omission is indicative of the imbalance between the intended meaning of ground circumscription and the definition of GC-model given in the paper.

Our definition of minimality, which subsumes the one in the original paper, is more intuitive in that it directly involves the assignment of individuals to

concepts and roles. This is anyway at the heart of the tableau method used by the authors of the original paper when providing algorithms for the reasoning tasks in ground circumscription. Apart from more intuitive, this is also the more realistic approach. Comparison between two models can only be done on the basis of the mapping of individuals to concepts and roles and requiring same domains as well is a non necessary specialization.

As supported in the following section, we have improved very much the reasoning in comparison to the original paper. Keeping the old definition would have hindered this development. In particular, had we not modified the definition, in order to devise a notion like the configuration space we would have needed to employ a model theoretical approach similar to filtration. That could possibly produce a more *language-dependent* theory, meaning that the results would be more conditional to $\mathcal{ALCO}$ and harder to generalize. As argued in Section 6.3, with our approach we have ensured that by just replacing our language with a more complex, but still decidable one, all the results maintain their validity.

Hence we believe that our reformulation of the notion of minimality is an upgrading to the previous work. It is more efficient in producing results by avoiding to interfere as much with the actual semantics, whilst capturing the essense of the idea of grounded circumscription in more satisfactory way.

8.3 Reasoning

In the original paper, satisfiability of a GC-knowledge base is proven decidable given that we work in a fairly complex language. We have shown that this result holds for a much simpler language and by only once calling a standard DL reasoner. This particular result can be obtained with the original definition of minimality as well.

However for the purpose of constructing an effective instance checking algorithm, our modified definition is very convenient. The confining of the search space of our algorithm, and the ability to define a linear order that induces the "smaller than" relation are advantages that result from our adoption of the new definition. Hence we can have a search strategy which we believe can prove to be very effective on average case (that remains to be seen with

testing).

In the original paper, the non-deterministic choices which must be made in order to obtain the set of all GC-models, resemble a brute force search, suggesting an unnecessary blow up to the complexity. Moreover, all algorithms are defined for $\mathcal{ALC}$ and because of their very specific nature they are not easily expanded. The fact that our algorithm and general theory are not restricted by the special features of $\mathcal{ALCO}$ is also a result of the flexibility of the new definition.

Chapter 9

Summary and Conclusions

We believe that we have considerably upgraded the foundational definition of grounded circumscription and have produced some first results which develop a strong basis for further research. Starting from a definition that is more accurate in incorporating the intuition behind grounded circuscription, we have an improved solution to the satisfiability task which does not require any very elaborate language. Moreover, we have provided an algorithm for instance checking, something that was only partially covered in the original paper (instance checking was defined narrowly and there was a minor mistake as well).

Apart from the algorithm itself, the theory provided gives a well-rounded understanding of the general potential of grounded circumscription, as redifined here. The configuration space can prove to be a useful notion for devising other non-standard reasoning algorithms. The down-the-chain axioms and minimality check as a sub-task could contribute to solving other reasoning tasks within ground circuscription as well.

As mentioned earlier, an advantage of our approach is that all our results hold if $\mathcal{ALCO}$ is replaced by a more complex language, as long as it is decidable. Certainly there is a lot of space for further development of grounded circumscription. It remains to be seen whether the IC algorithm can be sufficiently optimized. Additionally a more comprehensive account of the possible reasoning tasks should take place, perhaps resulting to more algorithms.

One of our main aims was to concentrate as much of the reasoning as possible to monotonic standard DLs. This is achieved, in our opinion to the

largest extent possible, as far as $\mathcal{ALCO}$ is considered. With this feature, our theory is implementation-friendly, and one main future objective is to create a reasoner for ground circumscription, which will of course be working on top of an efficient standard DL reasoner.

Bibliography

[1] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, New York, NY, USA, 2003.

[2] Pascal Hitzler, Markus Krötzsch, and Sebastian Rudolph. *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC, 1st edition, 2009.

[3] Carsten Lutz, Carlos Areces, Ian Horrocks, and Ulrike Sattler. Keys, nominals, and concrete domains. *Journal of Artificial Intelligence Research*, 23:667–726, 2004.

[4] John McCarthy. Circumscription — A Form of Non-Monotonic Reasoning. *Artificial Intelligence*, 13:27–39, 1980.

[5] Boris Motik, Bernardo Cuenca Grau, and Ulrike Sattler. Structured Objects in OWL: Representation and Reasoning, 2008.

[6] Sebastian Rudolph. Foundations of Description Logics. In *Proceedings of the 7th International Conference on Reasoning Web: Semantic Technologies for the Web of Data*, RW'11, pages 76–136, Berlin, Heidelberg, 2011. Springer-Verlag.

[7] Andrea Schaerf. Reasoning with individuals in concept languages. *Data and Knowledge Engineering*, 13(2):141–176, 1994.

[8] Manfred Schmidt-Schaußand Gert Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[9] K. Sengupta, A. Krisnadhi, and P. Hitzler. Local Closed World Semantics: Grounded Circumscription for OWL. In *The Semantic Web - ISWC 2011*, pages 617–632. Springer Berlin Heidelberg, 2011.

[10] Dmitry Tsarkov and Ian Horrocks. FaCT++ Description Logic Reasoner: System Description. In *Proceedings of the Third International Joint Conference on Automated Reasoning*, IJCAR'06, pages 292–297, Berlin, Heidelberg, 2006. Springer-Verlag.

[11] W3C OWL Working Group. *OWL 2 Web Ontology Language: Document Overview*, 2009. `http://www.w3.org/TR/owl2-overview/`.