Local Closed World Semantics: Grounded Circumscription for OWL

Kunal Sengupta, Adila Krisnadhi, and Pascal Hitzler

Wright State University, Dayton OH 45435, USA
{kunal,adila,pascal}@knoesis.org

Abstract. We present a new approach to adding closed world reasoning to the Web Ontology Language OWL. It transcends previous work on circumscriptive description logics which had the drawback of yielding an undecidable logic unless severe restrictions were imposed. In particular, it was not possible, in general, to apply local closure to roles.
In this paper, we provide a new approach, called *grounded circumscription,* which is applicable to $\mathcal{SROIQ}$ and other description logics around OWL without these restrictions. We show that the resulting language is decidable, and we derive an upper complexity bound. We also provide a decision procedure in the form of a tableaux algorithm.

1 Introduction

The semantics of the Web Ontology Language OWL [8] (which is based on the description logic $\mathcal{SROIQ}$ [9]) adheres to the Open World Assumption (OWA): statements which are *not* logical consequences of a given knowledge base are not necessarily considered false. The OWA is a reasonable assumption to make in the World Wide Web context (and thus for Semantic Web applications). However, situations naturally arise where it would be preferable to use the Closed World Assumption (CWA), that is, statements which are *not* logical consequences of a given knowledge base are considered false. Such situations include, for example, when data is being retrieved from a database, or when data can be considered *complete* with respect to the application at hand (see, e.g., [6,23]).

As a consequence, efforts have been made to combine OWA and CWA modeling for the Semantic Web, and knowledge representation languages which have both OWA and CWA modeling features are said to adhere to the *Local Closed World Assumption* (LCWA). Most of these combinations are derived from non-monotonic logics which have been studied in logic programming [10] or on first-order predicate logic [19,20,24]. Furthermore, many of them are of a *hybrid* nature, meaning that they achieve the LCWA by combining, e.g., description logics with (logic programming) rules. Please see [14, Section 4].

On the other hand, there are not that many approaches which provide a seamless (non-hybrid) integration of OWA and CWA, and each of them has its drawbacks. This is despite the fact that the modeling task, from the perspective of the application developer, seems rather simple: Users would want to specify, simply, that individuals in the extension of a predicate should be exactly those which are *necessarily required* to be in it, i.e., extensions should be *minimized*. Thus, what is needed for applications is a simple, intuitive approach to closed world modeling which caters for the above intuition, and is also sound, complete and computationally feasible.

Among the primary approaches to non-monotonic reasoning, there is one approach which employs the minimization idea in a very straightforward and intuitively simple manner, namely *circumscription* [19]. However, a naive transfer of the circumscription approach to description logics, which was done in [2,3,6,7], appears to have three primary drawbacks.

1. The approach is undedicable for expressive description logics (e.g., for the description logic $\mathcal{SROIQ}$) unless awkward restrictions are put in place. More precisely, it is not possible to have non-empty TBoxes plus minimization of roles if decidability is to be retained.
2. Extensions of minimized predicates can still contain elements which are not named individuals (or pairs of such, for roles) in the knowledge base, which is not intuitive for modeling (see also [6]).
3. Complexity of the approach is very high.

The undecidability issue (point 1) hinges, in a sense, also on point 2 above. In this paper, we provide a modified approach to circumscription for description logics, which we call *grounded circumscription*, that remedies both points 1 and 2.[1] Our idea is simple yet effective: we modify the circumscription approach from [2,3,6,7] by adding the additional requirement that extensions of minimized predicates may only contain named individuals (or pairs of such, for roles). In a sense, this can be understood as porting a desirable feature from (hybrid) MNKF description logics [5,12,13,21] to the circumscription approach. In another (but related) sense, it can also be understood as employing the idea of DL-safety [22], respectively of DL-safe variables [17] or nominal schemas [4,15,16].

The paper is a substantial extension of the workshop paper [14] and will be structured as follows. In Section 2, we introduce the semantics of grounded circumscription. In Section 3, we show that the resulting language is decidable. Next, we provide a tableaux calculus in Section 4 to reason with grounded circumscription. We conclude with a discussion of further work in Section 5.

2 Local Closed World Reasoning with Grounded Circumscription

In this section we describe LCW reasoning with grounded circumscription (GC) and also revisit the syntax and semantics of the Description Logic $\mathcal{ALC}$ and extend it with GC. Some results in this paper also apply to many other description logics besides $\mathcal{ALC}$, and we will point this out in each case.

[1] We are not yet addressing the complexity issue; this will be done in future work.

2.1 The Description Logic $\mathcal{ALC}$

Let $\mathsf{N}_C, \mathsf{N}_R$ and N_I be countably infinite sets of concept names, role names and individual names, respectively. The set of $\mathcal{ALC}$ *concepts* is the smallest set that is created using the following grammar where $A \in \mathsf{N}_\mathsf{C}$ denotes an atomic concept, $R \in \mathsf{N}_\mathsf{R}$ is a role name and C, D are concepts.

$$C \longrightarrow \top \mid \bot \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C$$

An $\mathcal{ALC}$ *TBox* is a finite set of axioms of the form $C \sqsubseteq D$, called *general concept inclusion* (GCI) *axioms*, where C and D are concepts. An $\mathcal{ALC}$ *ABox* is a finite set of axioms of the form $C(a)$ and $R(a, b)$, which are called *concept* and *role assertion axioms*, where C is a concept, R is a role and a, b are individual names. An $\mathcal{ALC}$ *knowledge base* is a union of an $\mathcal{ALC}$ ABox and an $\mathcal{ALC}$ TBox

The semantics is defined in terms of interpretations $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a non-empty set called the *domain* of interpretation and $\cdot^\mathcal{I}$ is an interpretation function which maps each individual name to an element of the domain $\Delta^\mathcal{I}$ and interprets concepts and roles as follows.

$$\begin{aligned}
&\top^\mathcal{I} = \Delta^\mathcal{I}\,, \quad \bot^\mathcal{I} = \emptyset\,, \quad A^\mathcal{I} \subseteq \Delta^\mathcal{I}\,, \quad R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I} \\
&(\neg C)^\mathcal{I} = \Delta^\mathcal{I} \setminus C^\mathcal{I}\,, \qquad (C_1 \sqcap C_2)^\mathcal{I} = C_1^\mathcal{I} \cap C_2^\mathcal{I}\,, \qquad (C_1 \sqcup C_2)^\mathcal{I} = C_1^\mathcal{I} \cup C_2^\mathcal{I} \\
&(\forall r.C)^\mathcal{I} = \{x \in \Delta^\mathcal{I} \mid (x, y) \in r^\mathcal{I} \text{ implies } y \in C^\mathcal{I}\} \\
&(\exists r.C)^\mathcal{I} = \{x \in \Delta^\mathcal{I} \mid \text{there is some } y \text{ with } (x, y) \in r^\mathcal{I} \text{ and } y \in C^\mathcal{I}\}
\end{aligned}$$

An interpretation $\mathcal{I}$ *satisfies* (is a *model* of) a GCI $C \sqsubseteq D$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$, a concept assertion $C(a)$ if $a^\mathcal{I} \in C^\mathcal{I}$, a role assertion $R(a, b)$ if $(a^\mathcal{I}, b^\mathcal{I}) \in R^\mathcal{I}$. We say $\mathcal{I}$ *satisfies* (is a *model* of) a knowledge base K if it satisfies every axiom in K. K is *satisfiable* if such a model $\mathcal{I}$ exists.

The negation normal form of a concept C, denoted by $\mathsf{NNF}(C)$, is obtained by pushing the negation symbols inward, as usual, such that negation appears only in front of atomic concepts, e.g., $\mathsf{NNF}(\neg(C \sqcup D)) = \neg C \sqcap \neg D$.

Throughout the paper, we will often talk about $\mathcal{L}$ knowledge bases ($\mathcal{L}$-KBs for short), where $\mathcal{L}$ is some decidable description logic. When we do this, then this indicates that the result does not only hold for $\mathcal{L}$ being $\mathcal{ALC}$, but rather for many decidable description logics around OWL. We will point out restrictions in each case. For general background on various description logics, as well as for established names (like $\mathcal{ALC}$) for different description logics, see [1,9].

Besides widely known DL constructors, we will also make use of Boolean role constructors (in limited form), which can be added to many description logics without loss of decidability or even of complexity [25,28]. We also make limited use of the *concept product*, written $C \times D$ with C, D concepts in $\mathcal{L}$, which allows a role to be constructed from the Cartesian product of two concepts, and which can actually be eliminated in the presence of Boolean role constructors [16,26]. In terms of interpretations $\mathcal{I}$, concept products are characterized by the equation $(C \times D)^\mathcal{I} = \{(x, y) \mid x \in C^\mathcal{I}, y \in D^\mathcal{I}\}$.

2.2 Grounded Circumscription

We now describe a very simple way for ontology engineers to model local closed world aspects in their ontologies: simply use a description logic (DL) knowledge base (KB) as usual, and augment it with *meta*-information which states that some predicates (concept names or role names) are *closed*. Semantically, those predicates are considered minimized, i.e., their extensions contain only what is absolutely required, and furthermore only contain *known* (or *named*) individuals, i.e., individuals which are explicitly mentioned in the KB. In the case of concept names, the idea of restricting their extensions only to known individuals is similar to the notion of nominal schema [4,16] (and thus, DL-safe rules [17,22]) and also the notion of DBox [27], while the minimization idea is borrowed from circumscription [19], one of the primary approaches to non-monotonic reasoning.

In the earlier efforts to carry over circumscription to DLs [2,3,6,7], circumscription is realized by the notion of *circumscription pattern*. A circumscription pattern consists of three disjoint sets of predicates (i.e., concept names and role names) which are called *minimized*, *fixed* and *varying* predicates, and a preference relation on interpretations.[2] The preference relation allows us to pick *minimal* models as the *preferred* models with respect to set inclusion of the extensions of the minimized predicates.

Our formalism here is inspired by one of the approaches described by Makinson in [18], namely restricting the set of valuations to get more logical consequences than what we can get as classical consequences. Intuitively, this approach is a simpler version of the circumscription formalism for DLs as presented in [3,7] in the sense that we restrict our attention only to models in which the extension of minimized predicates may only contain known individuals from the KB. Furthermore, the predicates (concept names and role names) in KB are partitioned into two disjoint sets of minimized and non-minimized predicates, i.e., no predicate is considered fixed.[3] The non-minimized predicates would be viewed as varying in the more general circumscription formalism mentioned above.

The non-monotonic feature of the formalism is given by restricting models of an $\mathcal{L}$-KB such that the extension of closed predicates may only contain individuals (or pairs of them) which are explicitly occurring in the KB, plus a minimization of the extensions of these predicates. We define a function Ind that maps each $\mathcal{L}$-KB to the set of individual names it contains, i.e., given an $\mathcal{L}$-KB K, $\mathsf{Ind}(K) = \{b \in \mathsf{N}_I \mid b \text{ occurs in } K\}$. Among all possible models of K that are obtained by the aforementioned restriction to $\mathsf{Ind}(K)$, we then select a model that is minimal w.r.t. concept inclusion or role inclusion, in accordance with the following definition.

[2] There is also a notion of *prioritization* which we will not use, mainly because we are not convinced yet that it is a desirable modeling feature for local closed world reasoning for the Semantic Web.

[3] Fixed predicates can be simulated in the original circumscriptive DL approach if negation is available, i.e., for fixed concept names, concept negation is required, while for fixed role names, role negation is required. The latter can be added to expressive DLs without jeopardizing decidability [16,28].

Definition 1. *A* GC-$\mathcal{L}$-KB *is a pair* (K, M) *where* K *is an* $\mathcal{L}$*-KB and* $M \subseteq \mathsf{N}_C \cup \mathsf{N}_r$*. For every concept name and role name* $W \in M$*, we say that* W *is* closed with respect to K*. For any two models* $\mathcal{I}$ *and* $\mathcal{J}$ *of* K*, we furthermore say that* $\mathcal{I}$ *is* smaller than *(or* preferred over*)* $\mathcal{J}$ w.r.t. M*, written* $\mathcal{I} \prec_M \mathcal{J}$*, iff all of the following hold: (i)* $\Delta^{\mathcal{I}} = \Delta^{\mathcal{J}}$ *and* $a^{\mathcal{I}} = a^{\mathcal{J}}$ *for every* $a \in \mathsf{N}_I$*; (ii)* $W^{\mathcal{I}} \subseteq W^{\mathcal{J}}$ *for every* $W \in M$*; and (iii) there exists a* $W \in M$ *such that* $W^{\mathcal{I}} \subset W^{\mathcal{J}}$

The following notion will be helpful.

Definition 2 (grounded model). *Given a GC-*$\mathcal{L}$*-KB* (K, M)*, a model* $\mathcal{I}$ *of* K *is called a* grounded model *w.r.t* M *if all of the following hold:*
(1) $C^{\mathcal{I}} \subseteq \{b^{\mathcal{I}} \mid b \in \mathsf{Ind}(K)\}$ *for each concept* $C \in M$*; and*
(2) $R^{\mathcal{I}} \subseteq \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid a, b \in \mathsf{Ind}(K)\}$ *for each role* $R \in M$

We now define models and logical consequence of GC-$\mathcal{L}$-KBs as follows.

Definition 3. *Let* (K, M) *be a GC-*$\mathcal{L}$*-KB. An interpretation* $\mathcal{I}$ *is a* GC-model *of* (K, M) *if it is a grounded model of* K *w.r.t.* M *and* $\mathcal{I}$ *is minimal w.r.t.* M*, i.e., there is no model* $\mathcal{J}$ *of* K *with* $\mathcal{J} \prec_M \mathcal{I}$*. A statement (GCI, concept assertion, or role assertion)* α *is a* logical consequence *(a* GC-inference*) of* (K, M) *if every GC-model of* (K, M) *satisfies* α*. Finally, a GC-*$\mathcal{L}$*-KB is said to be* GC-satisfiable *if it has a GC-model.*

Note that every GC-model is also a grounded model. Moreover, in comparison with the more general circumscription formalism for DLs as presented in [3,7], every GC-model of a KB is also a circumscriptive model,[4] hence every circumscriptive inference is also a valid GC-inference.

To give an example, consider the knowledge base K consisting of the axioms

$$\mathtt{hasAuthor}(\mathtt{paper1}, \mathtt{author1}) \qquad \mathtt{hasAuthor}(\mathtt{paper1}, \mathtt{author2})$$
$$\mathtt{hasAuthor}(\mathtt{paper2}, \mathtt{author3}) \qquad \top \sqsubseteq \forall \mathtt{hasAuthor}.\mathtt{Author}$$

Consider the following (ABox) statements: $\neg\mathtt{hasAuthor}(\mathtt{paper1}, \mathtt{author3})$ and $(\leq 2\ \mathtt{hasAuthor}.\mathtt{Author})(\mathtt{paper1})$.[5] Neither of them is a logical consequence of K under classical DL semantics. However, if we assume that we have complete information on authorship relevant to the application under consideration, then it would be reasonable to *close* parts of the knowledge base in the sense of the LCWA. In the original approach to circumscriptive DLs, we could close the concept name `Author`, but to no avail. But if we close `hasAuthor`, we obtain $(\leq 2\ \mathtt{hasAuthor}.\mathtt{Author})(\mathtt{paper1})$ as a logical consequence. In addition, if we adopt the Unique Name Assumption (UNA), $\neg\mathtt{hasAuthor}(\mathtt{paper1}, \mathtt{author3})$ is also a logical consequence of K. Even without UNA, we can still obtain this as a logical consequence if we add the following axioms to K, which essentially

[4] This can be seen, e.g., by a straightforward proof by contradiction.
[5] The semantics is $(\leq n\ R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid |\{y \mid (x, y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}| \leq n\}$; this *qualified number restriction* is not part of $\mathcal{ALC}$, though it makes a very good example without depending on the UNA.

forces the UNA:[6] $\mathtt{A}_1(\mathtt{author1}); \mathtt{A}_2(\mathtt{author2}); \mathtt{A}_3(\mathtt{author3}); \mathtt{A}_i \sqcap \mathtt{A}_j \sqsubseteq \bot$ for all $i \neq j$. With regard to this example, note that the closure of roles in the original circumscriptive DL approach leads to undecidability [3]. The GC-semantics, in contrast, is decidable even under role closure (see Section 3 below), and also yields the desired inferences.

3 Decidability of Grounded Circumscription

As noted earlier, circumscription in many expressive DLs is undecidable [3]. Undecidability even extends to the basic DL $\mathcal{ALC}$ when non-empty TBoxes are considered and roles are allowed as minimized predicates. Such a bleak outlook would greatly discourage useful application of circumscription, despite the fact that there is a clear need of such a formalism to model LCWA.

Our formalism aims to fill this gap by offering a simpler approach to circumscription in DLs that is decidable provided that the underlying DL is also decidable. The decidability result is obtained due to the imposed restriction of minimized predicates to known individuals in the KB as specified in Definition 3. Let $\mathcal{L}$ be any standard DL. We consider the reasoning task of *GC-KB satisfiability*: "given a GC-$\mathcal{L}$-KB (K, M), does (K, M) have a GC-model?" and show in the following that this is decidable.

Assume that $\mathcal{L}$ is any DL featuring nominals, concept disjunction, concept products, role hierarchies and role disjunctions. We show that GC-KB satisfiability in $\mathcal{L}$ is decidable if satisfiability in $\mathcal{L}$ is decidable.

Let (K, M) be a GC-$\mathcal{L}$-KB. We assume that $M = M_A \cup M_r$ where $M_A = \{A_1, \ldots, A_n\}$ is the set of minimized concept names and $M_r = \{r_1, \ldots, r_m\}$ is the set of minimized role names. Now define a family of $(n+m)$-tuples as

$$\mathcal{G}_{(K,M)} = \{(X_1, \ldots, X_n, Y_1, \ldots, Y_m) \mid X_i \subseteq \mathsf{Ind}(K), Y_j \subseteq \mathsf{Ind}(K) \times \mathsf{Ind}(K)\}$$

with $1 \leq i \leq n, 1 \leq j \leq m$. Note that there are

$$\left(2^{|\mathsf{Ind}(K)|}\right)^n \cdot \left(2^{|\mathsf{Ind}(K)|^2}\right)^m = 2^{n \cdot |\mathsf{Ind}(K)| + m \cdot |\mathsf{Ind}(K)|^2} \tag{1}$$

of such tuples; in particular note that $\mathcal{G}_{(K,M)}$ is a finite set.

Now, given (K, M) and some $G = (X_1, \ldots, X_n, Y_1, \ldots, Y_m) \in \mathcal{G}_{(K,M)}$, let K_G be the $\mathcal{L}$-KB consisting of all axioms in K together with all of the following axioms, where the A_i and r_j are all the predicates in M—note that we require role disjunction and concept products for this.

$$A_i \equiv \bigsqcup \{a\} \quad \text{for every } a \in X_i \text{ and } i = 1, \ldots, n$$

$$r_j \equiv \bigsqcup (\{a\} \times \{b\}) \quad \text{for every pair } (a, b) \in Y_j \text{ and } j = 1, \ldots, m$$

Then the following result clearly holds.

[6] The UNA can be enforced in an $\mathcal{ALC}$ KB by adding ABox statements $A_i(a_i)$, where a_i are all individuals and A_i are new concept names, to the knowledge base, together with all disjointness axioms of the form $A_i \sqcap A_j \sqsubseteq \bot$ for all $i \neq j$.

Lemma 1. *Let (K, M) be a GC-$\mathcal{L}$-KB. If K has a grounded model $\mathcal{I}$ w.r.t. M, then there exists $G \in \mathcal{G}_{(K,M)}$ such that K_G has a (classical) model $\mathcal{J}$ which coincides with $\mathcal{I}$ on all minimized predicates. Likewise, if there exists $G \in \mathcal{G}_{(K,M)}$ such that K_G has a (classical) model $\mathcal{J}$, then K has a grounded model $\mathcal{I}$ which coincides with $\mathcal{J}$ on all minimized predicates.*

Now consider the set

$$\mathcal{G}'_{(K,M)} = \{G \in \mathcal{G}_{(K,M)} \mid K_G \text{ has a (classical) model}\},$$

and note that this set is finite and computable in finite time since $\mathcal{G}_{(K,M)}$ is finite and $\mathcal{L}$ is decidable. Furthermore, consider $\mathcal{G}'_{(K,M)}$ to be ordered by the pointwise ordering $\prec$ induced by $\subseteq$. Note that the pointwise ordering of the finite set $\mathcal{G}'_{(K,M)}$ is also computable in finite time.

Lemma 2. *Let (K, M) be a GC-$\mathcal{L}$-KB and let*

$$\mathcal{G}''_{(K,M)} = \{G \in \mathcal{G}'_{(K,M)} \mid G \text{ is minimal in } (\mathcal{G}'_{(K,M)}, \prec)\}.$$

Then (K, M) has a GC-model if and only if $\mathcal{G}''_{(K,M)}$ is non-empty.

Proof. This follows immediately from Lemma 1 together with the following observation: Whenever K has two grounded models $\mathcal{I}$ and $\mathcal{J}$ such that $\mathcal{I}$ is smaller than $\mathcal{J}$, then there exist $G_{\mathcal{I}}, G_{\mathcal{J}} \in \mathcal{G}'_{(K,M)}$ with $G_{\mathcal{I}} \prec G_{\mathcal{J}}$ such that $K_{G_{\mathcal{I}}}$ and $K_{G_{\mathcal{J}}}$ have (classical) models $\mathcal{I}'$ and $\mathcal{J}'$, respectively, which coincide with $\mathcal{I}$, respectively, $\mathcal{J}$, on the minimized predicates.

Theorem 1. *GC-KB-satisfiability is decidable.*

Proof. This follows from Lemma 2 since the set $\mathcal{G}''_{(K,M)}$, for any given GC-KB (K, M), can be computed in finite time, i.e., it can be decided in finite time whether $\mathcal{G}''_{(K,M)}$ is empty.

Some remarks on complexity are as follows. Assume that the problem of deciding KB satisfiability in $\mathcal{L}$ is in the complexity class C. Observe from equation (1) that there are exponentially many possible choices of the $(n+m)$-tuples in $\mathcal{G}_{(K,M)}$ (in the size of the input knowledge base). Computation of $\mathcal{G}'_{(K,M)}$ is thus in $\textsc{Exp}^{\mathrm{C}}$, and subsequent computation of $\mathcal{G}''_{(K,M)}$ is also in $\textsc{Exp}$. We thus obtain the following upper bound.

Proposition 1. *The problem of finding a GC-model (if one exists) of a given GC-$\mathcal{L}$-KB is in $\textsc{Exp}^{C}$, where C is the complexity class of $\mathcal{L}$. Likewise, GC-$\mathcal{L}$-KB satisfiability is in $\textsc{Exp}^{C}$.*

4 Algorithms for Grounded Circumscriptive Reasoning

We now present algorithms for reasoning with grounded circumscription. We start with a tableaux algorithm to decide knowledge base GC-satisfiability and then discuss how to extend it to other reasoning tasks. For simplicity of presentation, we only consider GC-KB-satisfiability in $\mathcal{ALC}$, but the procedure should be adaptable to other DLs. Inspiration for the algorithm comes from [7,11].

4.1 Decision Procedure for GC-Satisfiability in $\mathcal{ALC}$

The algorithm is a tableaux procedure as usual where the expansion rules are defined to be compatible with the semantics of the language, and for easier reference, we call the resulting algorithm *Tableau1*. It starts with an initial graph F_i constructed using the ABox of a given GC-$\mathcal{ALC}$-KB (K, M), such that all known individuals are represented as nodes along with their labels that consist of the concepts that contain them in the ABox. Additionally, links are added for all role assertions using labels that consist of the roles in the ABox assertion axioms. We call this set of nodes and labels the *initial graph*. The creation of the initial graph F_i is described in terms of the following steps called the initialization process:

- create a node a, for each individual a that appears in at least one assertion of the form $C(a)$ in K (we call these nodes *nominal nodes*),
- add C to $\mathcal{L}(a)$, for each assertion of the form $C(a)$ or $R(a, b)$ in K,
- add R to $\mathcal{L}(a, b)$, for each assertion of the form $R(a, b)$ in K,
- initialize a set $T := \{\mathsf{NNF}(\neg C \sqcup D) \mid C \sqsubseteq D \in K\}$.

The algorithm begins with the initial graph F_i along with the sets T and M, and proceeds by non-deterministically applying the rules defined in Table 1, a process which can be understood as creating a candidate model for the knowledge base. The $\longrightarrow_{TBox}$, $\longrightarrow_{\sqcap}$, $\longrightarrow_{\exists}$ and $\longrightarrow_{\forall}$ rules are deterministic rules, whereas the $\longrightarrow_{\sqcup}$, $\longrightarrow_{GC_C}$ and $\longrightarrow_{GC_R}$ rules are non-deterministic rules, as they provide a choice, with each choice leading to possibly a different graph. The algorithm differs from the usual tableaux algorithm for $\mathcal{ALC}$, as it provides extra $\longrightarrow_{GC_C}$ and $\longrightarrow_{GC_R}$ non-deterministic rules, such that the candidate models are in fact *grounded* candidate models as defined in Definition 2. The rules are applied until a clash is detected or until none of the rules is applicable. A graph is said to contain an *inconsistency clash* when one of the node labels contains both C and $\neg C$, or it contains $\bot$, and it is called *inconsistency-clash-free* if it does not contain an inconsistency clash. The algorithm by application of the rules upon termination generates a so-called *completion graph*. A notion of blocking is required to ensure termination, and we define it as follows.

Definition 4 (Blocking). *A non-nominal node x is* blocked

1. *if it has a* blocked *ancestor; or*
2. *if it has a non-nominal ancestor x' such that $\mathcal{L}(x) \subseteq \mathcal{L}(x')$ and the path between x' and x consists only of non-nominal nodes.*

In the second case, we say that x is directly blocked *by the node x'. Note that any non-nominal successor node of x is also blocked.*

For a GC-$\mathcal{ALC}$-KB (K, M), the tableau expansion rules when applied exhaustively, generate a completion graph which consists of nodes, edges and their labels, each node x of the graph is labeled with a set of (complex or atomic) concepts and each edge (x, y) is labeled with a set of roles.

Lemma 3 (termination). *Given any GC-$\mathcal{ALC}$-KB (K, M), the tableaux procedure for (K, M) terminates.*

Table 1. Tableau1 expansion rules for GC-$\mathcal{ALC}$-KBs (K, M). The first five rules are taken directly from the $\mathcal{ALC}$ tableaux algorithm. Input: F_i, T and M.

$\longrightarrow_{TBox}$:	**if** $C \in T$ and $C \notin \mathcal{L}(x)$ **then** $\mathcal{L}(x) := \mathcal{L}(x) \cup \{C\}$
$\longrightarrow_{\sqcap}$:	**if** $C_1 \sqcap C_2 \in \mathcal{L}(x), x$ is not blocked, and $\{C_1, C_2\} \not\subseteq \mathcal{L}(x)$ **then** $\mathcal{L}(x) := \mathcal{L}(x) \cup \{C_1, C_2\}$
$\longrightarrow_{\sqcup}$:	**if** $C_1 \sqcup C_2 \in \mathcal{L}(x), x$ is not blocked, and $\{C_1, C_2\} \cap \mathcal{L}(x) = \emptyset$ **then** $\mathcal{L}(x) := \mathcal{L}(x) \cup \{C_1\}$ or $\mathcal{L}(x) := \mathcal{L}(x) \cup \{C_2\}$
$\longrightarrow_{\exists}$:	**if** $\exists R.C \in \mathcal{L}(x), x$ is not blocked, and x has no R-successor y with $C \in \mathcal{L}(y)$ **then** add a new node y with $\mathcal{L}(y) := \{C\}$ and $\mathcal{L}(x, y) := \{R\}$
$\longrightarrow_{\forall}$:	**if** $\forall R.C \in \mathcal{L}(x), x$ is not blocked, and x has an R-successor y with $C \notin \mathcal{L}(y)$ **then** $\mathcal{L}(y) := \mathcal{L}(y) \cup \{C\}$
$\longrightarrow_{GC_C}$:	**if** $C \in \mathcal{L}(x), C \in M, x \notin \mathsf{Ind}(K)$ and x is not blocked **then** for some $a \in \mathsf{Ind}(K)$ do 1. $\mathcal{L}(a) := \mathcal{L}(a) \cup \mathcal{L}(x)$, 2. if x has a predecessor y, then $\mathcal{L}(y, a) := \mathcal{L}(y, a) \cup \mathcal{L}(y, x)$, 3. remove x and all incoming edges to x in the completion graph
$\longrightarrow_{GC_R}$:	**if** $R \in \mathcal{L}(x, y), R \in M$ and y is not blocked. **then** initialize variables $x' := x$ and $y' := y$, and do 1. if $x \notin \mathsf{Ind}(K)$ then for some $a \in \mathsf{Ind}(K), \mathcal{L}(a) := \mathcal{L}(a) \cup \mathcal{L}(x)$, $x' := a$. 2. if $y \notin \mathsf{Ind}(K)$ for some $b \in \mathsf{Ind}(K), \mathcal{L}(b) := \mathcal{L}(b) \cup \mathcal{L}(y)$ and $y' := b$ 3. if $x' = a$ and x has a predecessor z, then $\mathcal{L}(z, a) := \mathcal{L}(z, a) \cup \mathcal{L}(z, x)$. 4. $\mathcal{L}(x', y') := \mathcal{L}(x', y') \cup \{R\}$ 5. if $x' = a$ remove x and all incoming edges to x and if $y' = b$ remove y and all incoming edges to y from the completion graph.

Proof. First note that node labels can only consist of axioms from K in NNF or of subconcepts of axioms from K in NNF. Thus, there is only a finite set of possible node labels, and thus there is a global bound, say $m \in \mathbb{N}$, on the cardinality of node labels.

Now note the following. (1) The number of times any rule can be applied to a node is finite, since the labels trigger the rules and the size of labels is bounded by m. (2) The outdegree of each node is bounded by the number of possible elements of node labels of the form $\exists R.C$, since only the $\longrightarrow_{\exists}$ rule generates new nodes. Thus the outdegree is also bounded by m. Further, infinite non-looping paths cannot occur since there are at most 2^m possible different labels, and so the blocking condition from Definition 4 implies that some node along such a path would be blocked, contradicting the assumption that the path would be infinite. (3) While the $\longrightarrow_{GC_C}$ rule and the $\longrightarrow_{GC_R}$ rule delete nodes, they can only change labels of nominal nodes by possibly adding elements to nominal node labels. Since the number of possible elements of node labels is bounded by m, at some stage application of the $\longrightarrow_{GC_C}$ rule or the $\longrightarrow_{GC_R}$ rule will no longer add anything to nominal node labels, and then no new applications of rules can be enabled by this process.

From (1), (2) we obtain a global bound on the size of the completion graphs which can be generated by the algorithm, and from (3) we see that infinite loops due to deletion and recreation of nodes cannot occur. Thus, the algorithm necessarily terminates.

Before we show that the tableaux calculus is sound and complete, we define a function called read function which will be needed for clarity of the proof and verification of minimality of the models.

Definition 5 (read function). *Given an inconsistency-clash-free completion graph F, we define a read function* r *which maps the graph to an interpretation* $\mathsf{r}(F) = \mathcal{I}$ *in the following manner. The interpretation domain* $\Delta^{\mathcal{I}}$ *contains all the non-blocked nodes in the completion graph. Further, for each atomic concept* A*, we set* $A^{\mathcal{I}}$ *to be the set of all non-blocked nodes* x *for which* $A \in \mathcal{L}(x)$*. For each role name* R*, we set* $R^{\mathcal{I}}$ *to be the set of pairs* (x, y) *which satisfy any of the following conditions:*

- $R \in \mathcal{L}(x, y)$ *and* y *is not blocked; or*
- x *is an immediate* R*-predecessor of some node* z*, and* y *directly blocks* z

The mapping just defined is then lifted to complex concept descriptions as usual.

The second condition is due to the well-known technique of unraveling (see, e.g., [11]): while disregarding blocked nodes, an incoming edge from an immediate R-predecessor x of the blocked node z is considered to be replaced by an edge from the predecessor to the node y which directly blocks z. This accounts for the intuition that a path ending in a blocked node stands for an infinite but repetitive path in the model.

Lemma 4 (soundness). *If the expansion rules are applied to a GC-$\mathcal{ALC}$-KB* (K, M)*, such that they result in an inconsistency-clash-free completion graph* F*, then* K *has a grounded model* $\mathcal{I} = \mathsf{r}(F)$*. Furthermore, the extension* $A^{\mathcal{I}}$ *of each concept* $A \in M$ *under* $\mathcal{I}$ *coincides with the set* $\{x \mid x \in A^{\mathsf{r}(F)}\}$*, the extension* $R^{\mathcal{I}}$ *of each role* $R \in M$ *under* $\mathcal{I}$ *coincides with the set* $\{(x, y) \mid (x, y) \in R^{\mathsf{r}(F)}\}$*, and both these sets can be read off directly from the labels of the completion graph.*

Proof. From the inconsistency-clash-free completion graph F, we create an interpretation $\mathcal{I} = \mathsf{r}(F)$ where r is the read function defined in Definition 5. Since the completion graph is free of inconsistency clashes, and the first five expansion rules from Table 1 follow the definition of a model from Section 2, the resulting interpretation is indeed a model of K.[7] Moreover, the $\longrightarrow_{GC_C}$ and $\longrightarrow_{GC_R}$ rules ensure that the extensions of minimized predicates contain only (pairs of) known

[7] This can be proven formally by structural induction on formulas as in [11].

individuals. Hence, $\mathsf{r}(F) = \mathcal{I}$ is a grounded model of K w.r.t M, and Definition 5 shows how the desired extensions can be read off from the completion graph.

Lemma 5 (completeness). *If a GC-$\mathcal{ALC}$-KB (K, M) has a grounded model $\mathcal{I}$, then the expansion rules can be applied to the initial graph F_i of (K, M) in such a way that they lead to an inconsistency-clash-free completion graph F, and such that the following hold.*

- $\Delta^{\mathsf{r}(F)} \subseteq \Delta^{\mathcal{I}}$
- $a^{\mathsf{r}(F)} = a^{\mathcal{I}}$ *for every nominal node a*
- $W^{\mathsf{r}(F)} \subseteq W^{\mathcal{I}}$ *for every $W \in M$*
- *the extensions, under $\mathsf{r}(F)$, of the closed concept and role names can be read off from F as in the statement of Lemma 4.*

Proof. Given a grounded model $\mathcal{I}$ for K w.r.t M, we can apply the completion rules to F_i in such a way that they result in an inconsistency-clash-free completion graph F. To do this we only have to ascertain that, for any nodes x and y in the graph, the conditions $\mathcal{L}(x) \subseteq \{C \mid \pi(x) \in C^{\mathcal{I}}\}$ and $\mathcal{L}(x, y) \subseteq \{R \mid (\pi(x), \pi(y)) \in R^{\mathcal{I}}\}$ are satisfied, where π is mapping from nodes to $\Delta^{\mathcal{I}}$. This construction is very similar to the one in [11, Lemma 6], to which we refer for details of the argument.

The remainder of the statement follows from the fact that the two conditions just given are satisfied, and from the reading-off process specified in Lemma 4.

We have provided an algorithm that generates a set of completion graphs and each inconsistency-clash-free completion graph represents a grounded model. In fact (K, M) is GC-satisfiable if at least one of the completion graphs is inconsistency-clash-free.

Theorem 2. *Let (K, M) be a GC-$\mathcal{ALC}$-KB. Then (K, M) has a grounded model if and only if it is GC-satisfiable.*

Proof. The *if* part of the proof is trivial.

We prove the *only if* part. For any grounded model $\mathcal{I}$, let $|M_{\mathcal{I}}|$ denote the sum of the cardinalities of all extensions of all the minimized predicates in M, and note that, for any two grounded models $\mathcal{I}$ and $\mathcal{J}$ of K w.r.t. M, we have $|M_{\mathcal{J}}| < |M_{\mathcal{I}}|$ whenever $\mathcal{J} \prec_M \mathcal{I}$. Hence, for any grounded model $\mathcal{I}$ of K w.r.t. M which is not a GC-model of (K, M), there is a grounded model $\mathcal{J}$ of K w.r.t. M with $\mathcal{J} \prec_M \mathcal{I}$ and $|M_{\mathcal{J}}| < |M_{\mathcal{I}}|$. Since $|M_{\mathcal{I}}| > 0$ for all grounded models $\mathcal{I}$ (and because $\prec_M$ is transitive), we obtain that, given some grounded model $\mathcal{I}$, it is not possible that there is an infinite descending chain of grounded models preferred over $\mathcal{I}$. Consequently, there must be some grounded model $\mathcal{J}$ of K w.r.t. M which is minimal w.r.t. $|M_{\mathcal{J}}|$ among all models which are preferred over $\mathcal{I}$. This model $\mathcal{J}$ must be a GC-model, since otherwise it would not be minimal.

The following is a direct consequence of Lemmas 3, 4, 5, and Theorem 2.

Theorem 3. *The tableaux algorithm Tableau1 presented above is a decision procedure to determine GC-satisfiability of GC-$\mathcal{ALC}$-KBs.*

4.2 Inference Problems beyond GC-Satisfiability

Unlike in other description logics, common reasoning tasks such as concept satisfiability or instance checking cannot be readily reduced to GC-satisfiability checking.[8] To cover other inference tasks, we need to extend the previously described algorithm. To do this, we first describe a tableaux algorithm *Tableau2* which is a modification of Tableau1, as follows. All computations are done with respect to an input GC-$\mathcal{ALC}$-KB (K, M).

(i) Initialization of Tableau2 is done on the basis of a inconsistency-clash-free completion graph F, as follows. We create a finite set of nodes which is exactly the domain $\Delta^{\mathcal{I}}$ of a grounded model $\mathcal{I} = \mathsf{r}(F)$. We distinguish between two different kinds of nodes, the *$\mathcal{I}$-nominal nodes*, which are nodes corresponding to some $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ where a is an individual name, and the remaining nodes which we call *variable nodes.* For initialization, we furthermore add all information from the ABox of K to the graph and create the set T from K, as in the initialization of Tableau1.

(ii) We modify the $\longrightarrow_{\exists}$ rule as follows.

$$\begin{array}{ll} \longrightarrow_{\exists}: & \textbf{if } \exists R.C \in \mathcal{L}(x), \text{ and } x \text{ has no R-successor } y \text{ with } C \in \mathcal{L}(y) \\ & \textbf{then } \text{ select an existing node } y \text{ and} \\ & \qquad \text{set } \mathcal{L}(y) := \{C\} \text{ and } \mathcal{L}(x, y) := \{R\} \end{array}$$

The above change in the $\longrightarrow_{\exists}$-rule enables us to restrict the graph to contain only the nodes it was initialized with, which means new nodes are not created.

(iii) We retain all other completion rules, however we dispose of blocking.

(iv) We retain the notion of inconsistency clash, and add a new notion of *preference clash* as follows. A graph F' obtained during the graph construction performed by Tableau2 is said to *contain a preference clash with $\mathcal{I}$* if at least one of the following holds.
  - $W^{\mathsf{r}(F')} = W^{\mathcal{I}}$ for each predicate $W \in M$
  - $W^{\mathsf{r}(F')} \cap \{a^{\mathcal{I}} \mid a \text{ an individual }\} \not\subseteq W^{\mathcal{I}}$ for some concept name $W \in M$
  - $W^{\mathsf{r}(F')} \cap \{(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid a, b \text{ individuals }\} \not\subseteq W^{\mathcal{I}}$ for some role name $W \in M$

Proposition 2. *Tableau2 always terminates. If it terminates by constructing an inconsistency- and preference-clash-free completion graph F', then $\mathsf{r}(F')$ is preferred over $\mathcal{I}$, i.e., it shows that $\mathcal{I}$ is not a GC-model. If no such graph F' is found, then $\mathcal{I}$ has been verified to be a GC-model.*

Proof. Termination is obvious due to the fact that no new nodes are created, i.e., the algorithm will eventually run out of choices for applying completion rules.

[8] E.g., say we want to decide whether (K, M) GC-entails $C(a)$. We cannot do this, in general, by using the GC-satisfiability algorithm in the usual way, i.e., by adding $\neg C(a)$ to K with subsequent checking of its GC-satisfiability. This is because in general it does not hold that (K, M) does not GC-entail $C(a)$ if $(K \cup \neg C(a), M)$ is GC-satisfiable. This is due to the non-monotonic nature of circumscription.

Now assume that the algorithm terminates by finding an inconsistency- and preference-clash-free completion graph F'. We have to show that $\mathsf{r}(F')$ is preferred over $\mathcal{I}$, i.e., we need to verify the properties listed in Definition 1. $\Delta^{\mathcal{I}} = \Delta^{\mathsf{r}(F')}$ holds because we initiate the algorithm with nodes being elements from $\Delta^{\mathcal{I}}$ and no new nodes are created. In case nodes are lost due to the grounding rules of Tableau2, we can simply extend $\Delta^{\mathsf{r}(F')}$ with some additional elements which are not otherwise of relevance for the model. The condition $a^{\mathcal{I}} = a^{\mathsf{r}(F')}$ for every $a^{\mathcal{I}} \in \Delta^{\mathsf{r}(F')}$ holds because this is how the algorithm is initialized. The remaining two conditions hold due to the absence of a preference clash.

For the last statement of the proposition, note that Tableau2 will non-deterministically find an inconsistency- and preference-clash-free completion graph if such a graph exists. This can be seen in a similar way as done in the proof of Lemma 5.

We next use Tableau1 and Tableau2 together to create an algorithm which finds GC-models for (K, M) if they exists. We call this algorithm *GC-model finder.* The algorithm is specified as follows, on input (K, M).

1. Initialize and run Tableau1 on (K, M). If no inconsistency-clash-free completion graph is found, then (K, M) has no GC-model and the algorithm terminates. Otherwise let F be the resulting completion graph.
2. Initialize Tableau2 from F and run it. If no inconsistency- and preference-clash-free completion graph is found, then $\mathsf{r}(F)$ is a GC-model of (K, M) and the algorithm terminates with output $\mathsf{r}(F)$. Otherwise let F' be the resulting completion graph.
3. Set $F = F'$ and go to step 2.

The loop in steps 2 and 3 necessarily terminates, because whenever step 2 finds a completion graph F' as specified, then $\mathsf{r}(F')$ is preferred over $\mathsf{r}(F)$. As argued in the proof of Theorem 2, there are no infinite descending chains of grounded models w.r.t. the *preferred over* relation, so the loop necessarily terminates. The output $\mathsf{r}(F)$ of the GC-model finder is a GC-model of (K, M), and we call F a *GC-model graph* of (K, M) in this case.

Theorem 4. *On input a GC-$\mathcal{ALC}$-KB (K, M), the GC-model finder creates a GC-model $\mathcal{I}$ of (K, M) if such a model exists. Conversely, for every GC-model $\mathcal{J}$ of (K, M), there exist non-deterministic choices of rule applications in the GC-model finder such that they result in a model $\mathcal{I}$ which coincides with $\mathcal{J}$ on all extensions of minimized predicates.*

Proof. The first statement follows from Propositon 2 together with the explanations already given. The second statement follows due to Lemma 5, since Tableau1 can already create the sought GC-model $\mathcal{I}$.

We now consider the reasoning tasks usually known as instance checking, concept satisfiability and concept subsumption. We provide a convenient way to utilize the GC-model finder algorithm to solve these problems by use of another notion of clash called entailment clash. The following definition describes the inference tasks and provides the notion of entailment clash for each of them as well.

Definition 6. *For a GC-$\mathcal{ALC}$-KB (K, M).*

– Instance checking*: Given an atomic concept C and an individual a in (K, M), $(K, M) \models_{GC} C(a)$ if and only if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for all GC-models $\mathcal{I}$ of (K, M). For instance checking of $C(a)$, a GC-model graph F is said to contain an* entailment clash *if $C \in \mathcal{L}(a)$ in F.*
– Concept satisfiability*: Given an atomic concept C in (K, M), C is GC-satisfiable if and only if $C^{\mathcal{I}} \neq \emptyset$ for some GC-model of (K, M). For checking satisfiability of C, a GC-model graph F is said to contain an* entailment clash *if $C \in \mathcal{L}(x)$ for any node x in F.*
– Concept subsumption*: Given concepts C and D in (K, M), $(K, M) \models_{GC} C \sqsubseteq D$ if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ in (K, M). Subsumption can be reduced to concept satisfiability: GC-$\mathcal{ALC}$-KB$(K, M) \models_{GC} C \sqsubseteq D$ if and only if $C \sqcap \neg D$ is not GC-satisfiable.*

We use the following process to solve these inference problems:

To determine if $C(a)$ is entailed by a GC-$\mathcal{ALC}$-KB (K, M), we invoke the GC-model finder until we find a GC-model. If this non-deterministic procedure results in a GC-model graph which does not contain an entailment clash, then $(K, M) \not\models_{GC} C(a)$. If no such GC-model graph can be generated this way, then $(K, M) \models_{GC} C(a)$.

To determine if C is GC-satisfiable, we invoke the GC-model finder until we find a GC-model. If this non-deterministic procedure results in a GC-model graph which contains an entailment clash, then C is satisfiable. If no such GC-model graph can be generated this way, then C is unsatisfiable.

5 Conclusion

We have provided a new approach for incorporating the LCWA into description logics. Our approach, grounded circumscription, is a variant of circumscriptive description logics which avoids two major issues of the original approach: Extensions of minimized predicates can only contain named individuals, and we retain decidability even for very expressive description logics while we can allow for the minimization of roles. We have also provided a tableaux algorithm for reasoning with grounded circumscription.

While the contributions in this paper provide a novel and, in our opinion, very reasonable perspective on LCWA reasoning with description logics, there are obviously also many open questions. A primary theoretical task is to investigate the complexity of our approach. Of more practical relevance would be an implementation of our algorithm with a substantial evaluation to investigate its efficiency empirically. More work also needs to be done in carrying over the concrete algorithm to description logics which are more expressive than $\mathcal{ALC}$.

It also remains to investigate the added value and limitations in practice of modeling with grounded circumscription. This will also shed light onto the question whether fixed predicates and prioritization are required for applications.

Acknowledgements. This work was supported by the National Science Foundation under award 1017225 "III: Small: TROn—Tractable Reasoning with Ontologies," and by State of Ohio Research Incentive funding in the Kno.e.CoM project. Adila Krisnadhi acknowledges support by a Fulbright Indonesia Presidential Scholarship PhD Grant 2010.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, second edn. (2007)
2. Bonatti, P.A., Lutz, C., Wolter, F.: Expressive Non-Monotonic Description Logics Based on Circumscription. In: Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'06). pp. 400–410. AAAI Press (2009)
3. Bonatti, P.A., Lutz, C., Wolter, F.: The Complexity of Circumscription in Description Logic. Journal of Artificial Intelligence Research 35, 717–773 (2009)
4. Carral Martínez, D., Krisnadhi, A., Maier, F., Sengupta, K., Hitzler, P.: Reconciling OWL and rules. Tech. rep., Kno.e.sis Center, Wright State University, Dayton, Ohio, USA (2011), available from http://www.pascal-hitzler.de/
5. Donini, F.M., Nardi, D., Rosati, R.: Description Logics of Minimal Knowledge and Negation as Failure. ACM Trans. on Computational Logic (TOCL) 3(2), 177–225 (April 2002)
6. Grimm, S., Hitzler, P.: Semantic Matchmaking of Web Resources with Local Closed-World Reasoning. International Journal of Electronic Commerce 12(2), 89–126 (2007)
7. Grimm, S., Hitzler, P.: A Preferential Tableaux Calculus for Circumscriptive $\mathcal{ALCO}$. In: Polleres, A., Swift, T. (eds.) Proc. of the 3rd Int. Conference on Web Reasoning and Rule Systems (RR'09). Lecture Notes in Computer Science, vol. 5837, pp. 40–54. Springer Berlin (2009)
8. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S. (eds.): OWL 2 Web Ontology Language: Primer. W3C Recommendation 27 October 2009 (2009), available from http://www.w3.org/TR/owl2-primer/
9. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman & Hall/CRC (2009)
10. Hitzler, P., Seda, A.K.: Mathematical Aspects of Logic Programming Semantics. CRC Press (2010)
11. Horrocks, I., Sattler, U.: Ontology reasoning in the SHOQ(D) description logic. In: Proc. IJCAI-01. pp. 199–204 (2001)
12. Knorr, M., Alferes, J., Hitzler, P.: Local Closed-World Reasoning with Description Logics under the Well-founded Semantics. Artificial Intelligence 175(9–10), 1528–1554 (2011)
13. Knorr, M., Alferes, J.J., Hitzler, P.: A Coherent Well-founded Model for Hybrid MKNF Knowledge Bases. In: Ghallab, M., Spyropoulos, C.D., Fakotakis, N., Avouris, N.M. (eds.) Proceedings of the 18th European Conference on Artificial Intelligence,Patras, Greece, July 21-25, 2008. Frontiers in Artificial Intelligence and Applications, vol. 178, pp. 99–103. IOS Press, Amsterdam, The Netherlands (2008)
14. Krisnadhi, A., Sengupta, K., Hitzler, P.: Local closed world semantics: Keep it simple, stupid! In: Rosati, R., Rudolph, S., Zakharyaschev, M. (eds.) 2011 International Workshop on Description Logics. CEUR Workshop Proceedings, vol. 745. CEUR-WS.org (2011)

15. Krisnadhi, A., Maier, F., Hitzler, P.: OWL and Rules. In: Polleres, A., d'Amato, C., Arenas, M., Handschuh, S., Kroner, P., Ossowski, S., Patel-Schneider, P.F. (eds.) Reasoning Web. Semantic Technologies for the Web of Data - 7th International Summer School 2011, Galway, Ireland, August 23-27, 2011, Tutorial Lectures. Lecture Notes in Computer Science, vol. 6848, pp. 382–415. Springer, Heidelberg (2011)
16. Krötzsch, M., Maier, F., Krisnadhi, A.A., Hitzler, P.: A better uncle for OWL: Nominal schemas for integrating rules and ontologies. In: Sadagopan, S., Ramamritham, K., Kumar, A., Ravindra, M., Bertino, E., Kumar, R. (eds.) Proceedings of the 20th International World Wide Web Conference, WWW2011, Hyderabad, India, March/April 2011. pp. 645–654. ACM, New York (2011)
17. Krötzsch, M., Rudolph, S., Hitzler, P.: ELP: Tractable Rules for OWL 2. In: Sheth, A.P., et al. (eds.) Proceedings of the 7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. Lecture Notes in Computer Science, vol. 5318, pp. 649–664. Springer (2008)
18. Makinson, D.: Bridges from Classical to Nonomonotonic Logic, Texts in Computing, vol. 5. King's College Publications (2005)
19. McCarthy, J.: Circumscription – A Form of Non-Monotonic Reasoning. Artificial Intelligence 13(1–2), 27–39 (1980)
20. Moore, R.: Possible-worlds Semantics for Autoepistemic Logic. In: Proceedings of the 1984 Non-monotonic Reasoning Workshop. AAAI, Menlo Park, CA (1984)
21. Motik, B., Rosati, R.: Reconciling Description Logics and Rules. Journal of the ACM 57(5), 1–62 (2010)
22. Motik, B., Sattler, U., Studer, R.: Query Answering for OWL-DL with Rules. Journal of Web Semantics 3, 41–60 (July 2005)
23. Patel, C., Cimino, J.J., Dolby, J., Fokoue, A., Kalyanpur, A., Kershenbaum, A., Ma, L., Schonberg, E., Srinivas, K.: Matching Patient Records to Clinical Trials Using Ontologies. In: Aberer, K., Choi, K.S., Noy, N.F., Allemang, D., Lee, K.I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007. Lecture Notes in Computer Science, vol. 4825, pp. 816–829. Springer (2007)
24. Reiter, R.: A Logic for Default Reasoning. Artificial Intelligence 13, 81–132 (1980)
25. Rudolph, S., Krötzsch, M., Hitzler, P.: Cheap Boolean Role Constructors for Description Logics. In: Hölldobler, S., et al. (eds.) Proc. of the 11th European Conference on Logics in Artificial Intelligence (JELIA'08). Lecture Notes in Artificial Intelligence, vol. 5293, pp. 362–374. Springer-Verlag (2008)
26. Rudolph, S., Krötzsch, M., Hitzler, P.: All Elephants are Bigger than All Mice. In: Baader, F., Lutz, C., Motik, B. (eds.) Proceedings of the 21st International Workshop on Description Logics (DL2008), Dresden, Germany, May 13-16, 2008. CEUR Workshop Proceedings, vol. 353 (2008)
27. Seylan, I., Franconi, E., de Bruijn, J.: Effective query rewriting with ontologies over DBoxes. In: Boutilier, C. (ed.) IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009. pp. 923–925 (2009)
28. Tobies, S.: Complexity Results and Practical Algorithms for Logics in Knowledge Representation. Ph.D. thesis, RWTH Aachen, Germany (2001)