Generalized Closed World Reasoning in Description Logics with Extended Domain Closure

Oliver Gries

Hamburg University of Technology
21079 Hamburg, Germany

Abstract. Generalized closed world reasoning allows for the assumption of a specified maximal set of negated atomic assertions retaining the consistency of an indefinite knowledge base. In this paper, a generalized closed world assumption (GCWA) is specified for the description logic $\mathcal{ALCN}$ and all of its sublanguages, provided that the terminological component is eliminable. In certain situations, by applying the GCWA, queries are often answered as intended by users. Opposed to queries with an epistemic operator $\mathbf{K}$, querying with this approach provides the information which of the corresponding assertions can consistently be assumed to be true. Further, the GCWA can be applied locally.

1 Introduction

The set of assertions logically entailed by a knowledge base usually contains less elements than expected by users. In description logics (DLs), this is the case for queries involving value restrictions $\forall R.C$ (resp. negated existential restrictions), at-most restrictions $(\leq n\, R.C)$ (resp. negated at-least restrictions) or atomic negation $\neg A$ resp. $\neg R$. For example, $\forall \mathit{hasChild}.\mathit{Female}(\mathit{john})$ is not entailed from $\mathit{KB} = \{\mathit{hasChild}(\mathit{john}, \mathit{mary}), \mathit{Female}(\mathit{mary})\}$, though often intended. Due to the open world assumption (OWA) of logical formalisms, an assertion is only entailed if it is satisfied in all models of a knowledge base. An approach to solve this problem is the *closed world assumption* (CWA) defined by Reiter in [2]. Under the CWA, a given knowledge base KB can be considered to be complete: If an atomic formula does not follow from KB, its negation is assumed to be true. There are extensions to the CWA which further reduce the models of a knowledge base to intended models: The *domain closure* assumes that all entities of the domain are denoted by the constants mentioned in a knowledge base and the *unique name assumption* (UNA) is the assumption that different constants denote different entities of the domain. While in relational databases the CWA and its extensions are presupposed, this is not the case for description logics.

In less expressive DLs equivalent to sets of Horn clauses such as $\mathcal{AL}_0$ knowledge bases [6], there is a rather simple method to apply the CWA with a more expressive query language similar to the algorithm mentioned in [10] for so-called *vivid* first-order knowledge bases.

Unfortunately, indefinite knowledge bases (augmenting e.g. $\mathcal{AL}_0$ with disjunction or existential quantification) do not remain consistent by applying the CWA. Given e.g. a knowledge base $KB = \{(\mathit{Intelligent} \sqcup \mathit{Rich})(\mathit{john})\}$, under CWA John is not intelligent as well as not rich, leading to an inconsistency.

A well-established approach to enable closed world reasoning and to avoid such inconsistencies is to extend the query language with the epistemic operator $\mathbf{K}$ [5–7]. With this operator it is possible to apply a concept closure resp. a role closure by using the expression $\neg\mathbf{K}A$ instead of $\neg A$, $\neg\mathbf{K}R$ instead of $\neg R$, $\forall\mathbf{K}R.C$ instead of $\forall R.C$ and $(\leq n\,\mathbf{K}R.C)$ instead of $(\leq n\,R.C)$. The specification of these expressions is similar to the application of a local CWA and therefore of particular importance for querying DL systems. However, there is no information about which assertions can consistently be assumed. Consider again $KB = \{(\mathit{Intelligent} \sqcup \mathit{Rich})(\mathit{john})\}$. Since $\mathit{Intelligent}(\mathit{john})$ and $\mathit{Rich}(\mathit{john})$ are not satisfied in all models of KB, $\neg\mathbf{K}\mathit{Intelligent}(\mathit{john})$ as well as $\neg\mathbf{K}\mathit{Rich}(\mathit{john})$ are entailed from KB though $KB \cup \{\neg\mathit{Intelligent}(\mathit{john}), \neg\mathit{Rich}(\mathit{john})\}$ would be inconsistent. In other words, by executing epistemic queries, there is also the problem of considering indefinite knowledge.

A suitable formalism to deal with disjunctive information is the generalized closed world assumption (GCWA) [1]. It is a refinement of the CWA retaining the consistency of an indefinite knowledge base by considering indefinite clauses logically entailed by this knowledge base. Such a formalism would provide an answer how to apply a closed world assumption in expressive DLs, but instead of assuming complete knowledge for the whole domain, in concrete applications there is often the need to consider open world reasoning with the additional possibility to close off parts of the domain assumed to be represented completely. Therefore, it should be possible to apply the GCWA locally.

The main contribution of this paper is the specification of a local GCWA for $\mathcal{ALCN}$ and all of its sublanguages, provided that the terminological component is eliminable, i.e., a specification of a method deciding whether $KB \models_{GCWA^\pi} \alpha$ holds, where $KB = \mathcal{T} \cup \mathcal{A}$ is an $\mathcal{ALCN}$ knowledge base (with the possibility to eliminate $\mathcal{T}$ at the beginning of the reasoning process), α is an $\mathcal{ALCN}$-assertion and $\models_{GCWA^\pi}$ is an entailment relation applying a GCWA only to specific parts of the domain. Since the GCWA is defined under consideration of clauses, all assertions involved in KB as well as the assertion α are transformed to a propositional clause form. For the transformation of role restrictions, a kind of domain closure is required. In order to deal with existential quantifiers, this domain closure is extended with new constants.

The method presented provides a foundation to apply closed world reasoning to all further expressive DLs having the finite model property. At the best of the authors knowledge, until now there has not been a concrete specification of a closed world assumption for DLs composed of basic constructors such as disjunction, existential quantification or non-atomic negation. The approach is related to formalisms introducing circumscription to DLs [3, 4], but while [3] does only consider a very restricted form of $\mathcal{ALE}$, [4] provides a model-theoretic definition of circumscription as well as complexity results rather than a concrete decision procedure. Since the non-monotonic framework presented in [8], similar to [5–7], is based on the extension of DLs with epistemic operators, it encounters the same problematic of considering indefinite knowledge.

The approach presented here is also appropriate for users not experienced with non-monotonic DLs: Besides the DL KB and the query specified as usual, it only requires the determination of the parts of the domain to be closed.

The paper is structured as follows: In Sect. 2, the syntax and semantics of $\mathcal{ALCN}$ is presented. Problems with the CWA are mentioned in Sect. 3 leading to the consideration of the GCWA (Sect. 4). In Sect. 5, the need for an extended domain closure is explained. Sect. 6 shows how $\mathcal{ALCN}$-assertions can be transformed to a propositional clause form with respect to this domain closure and proposes an algorithm for deciding whether an $\mathcal{ALCN}$-assertion is entailed from an $\mathcal{ALCN}$ knowledge base under consideration of a local GCWA. The paper concludes in Sect. 7 with a summary and a discussion of the results.

2 The Description Logic $\mathcal{ALCN}$

The vocabulary of description-logic languages consists of *concepts*, *roles* and *constants*. Based on *atomic concepts* A and *atomic roles* R, $\mathcal{ALCN}$ concept descriptions C, D are defined according to the syntax rule

$$C,\, D \;\longrightarrow \top \mid \bot \mid A \mid \neg C \mid (C \sqcap D) \mid (C \sqcup D) \mid \forall R.C \mid \exists R.C \mid (\geq n\, R) \mid (\leq n\, R).$$

Descriptions are constructed with the logical constants $\top$ and $\bot$, complements $\neg C$, conjunctions $(C \sqcap D)$, disjunctions $(C \sqcup D)$, value restrictions $\forall R.C$, existential restrictions $\exists R.C$ and number restrictions $(\geq n\, R)$ and $(\leq n\, R)$.

The semantics is defined with interpretations $\mathcal{I} = (\triangle^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\triangle^{\mathcal{I}}$ is a non-empty set of all objects considered in $\mathcal{I}$ (called the domain of $\mathcal{I}$) and $\cdot^{\mathcal{I}}$ is an interpretation function which maps constants a to objects of the domain, $a^{\mathcal{I}} \in \triangle^{\mathcal{I}}$, atomic concepts A to subsets of the domain, $A^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}}$, and atomic roles R to subsets of the cartesian product of the domain, $R^{\mathcal{I}} \subseteq \triangle^{\mathcal{I}} \times \triangle^{\mathcal{I}}$. The extension of $\cdot^{\mathcal{I}}$ to arbitrary $\mathcal{ALCN}$ concept descriptions is defined as follows, where *card* is a function assigning to a set its cardinality:

$$\begin{aligned}
\top^{\mathcal{I}} &= \triangle^{\mathcal{I}} & (\forall R.C)^{\mathcal{I}} &= \{u \in \triangle^{\mathcal{I}} \mid (\forall v)\, [(u,v) \in R^{\mathcal{I}} \rightarrow v \in C^{\mathcal{I}}]\} \\
\bot^{\mathcal{I}} &= \emptyset & (\exists R.C)^{\mathcal{I}} &= \{u \in \triangle^{\mathcal{I}} \mid (\exists v)\, [(u,v) \in R^{\mathcal{I}} \wedge v \in C^{\mathcal{I}}]\} \\
(\neg C)^{\mathcal{I}} &= \triangle^{\mathcal{I}} \setminus C^{\mathcal{I}} & (\geq n\, R)^{\mathcal{I}} &= \{u \in \triangle^{\mathcal{I}} \mid card\{v \mid (u,v) \in R^{\mathcal{I}}\} \geq n\} \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} & (\leq n\, R)^{\mathcal{I}} &= \{u \in \triangle^{\mathcal{I}} \mid card\{v \mid (u,v) \in R^{\mathcal{I}}\} \leq n\} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} & &
\end{aligned}$$

An $\mathcal{ALCN}$ knowledge base $KB = \mathcal{T} \cup \mathcal{A}$ is comprised of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. $\mathcal{T}$ consists of a set of axioms $C \sqsubseteq D$ and $C \equiv D$ called *concept inclusions* resp. *concept equivalences* and $\mathcal{A}$ consists of a set of *concept assertions* $C(a)$ and *role assertions* $R(a,b)$, where a and b are constants. An interpretation $\mathcal{I}$ *satisfies* $C \sqsubseteq D$ resp. $C \equiv D$ resp. $C(a)$ resp. $R(a,b)$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ resp. $C^{\mathcal{I}} = D^{\mathcal{I}}$ resp. $a^{\mathcal{I}} \in C^{\mathcal{I}}$ resp. $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$. If an interpretation $\mathcal{I}$ satisfies all axioms of $\mathcal{T}$ resp. all assertions of $\mathcal{A}$ it is called a *model* of $\mathcal{T}$ resp. $\mathcal{A}$. If it satisfies both $\mathcal{T}$ and $\mathcal{A}$ it is called a model of KB. Finally, if there is a model of KB, then KB is called *satisfiable*.

In the following, the set of all predicate names (i.e. atomic concepts A and atomic roles R) appearing in a given KB is denoted with PR and the set of all assertions $A(a)$ and $R(a,b)$ constructable from PR and the constants of KB with AS. In order to get a basis for the application of a GCWA to DLs, it is assumed that $\mathcal{T}$ is acyclic and does only contain concept inclusions and concept equivalences of the form $A \sqsubseteq C$ resp. $A \equiv C$. Consequently, it is possible to eliminate $\mathcal{T}$ at the beginning of any reasoning process [9] and KB is reduced to an ABox $\mathcal{A}_{\mathcal{T}}$, expanded with respect to $\mathcal{T}$ such that $KB = \mathcal{A}_{\mathcal{T}}$.

A reasoning service relevant for this investigation is the *assertion check*, symbolically $KB \models_R \alpha$, deciding if an assertion α is entailed from an $\mathcal{ALCN}$ knowledge base KB with respect to an entailment relation $\models_R$ ($KB \models \alpha$, if α is satisfied in all models of KB). This service provides the basis for querying DL knowledge bases. If $\alpha = C(a)$, it is called *instance check* and if $\alpha = R(a,b)$, it is called *relation check*. Additionally, by a slight extension of the query language, it is possible to decide *negated relation checks* $KB \models_R \neg R(a,b)$. We will also consider the case in which α is a propositional expression (note that $\models_R$ is not restricted to relate (sets of) formulas of the same representation language).

3 Problems with the Closed World Assumption (CWA)

The CWA has been introduced by Reiter in [2]. Under the CWA, for all atomic assertions that are not derivable from a given DL knowledge base, this knowledge base is augmented *implicitly* with corresponding negated atomic assertions:

Definition 1. *Let KB be a DL knowledge base. Then*

$$CWA(KB) = \{\neg p \mid p \in AS \text{ and } KB \not\models p\}$$

is the closed world assumption of KB. The entailment relation $\models_{cwa}$ considering this assumption for any assertion α is defined by

$$KB \models_{cwa} \alpha \;\text{ iff }\; KB^{+} \models \alpha, \text{ with } KB^{+} = KB \cup CWA(KB).$$

With the CWA, there are no gaps in the knowledge base, since for each atomic assertion $p \in AS$ either $KB \models_{cwa} p$ or $KB \models_{cwa} \neg p$ is successful. The same holds for arbitrary boolean assertions, but not for assertions with quantifiers until a domain closure axiom $DCA(KB) = (\forall x)\ [x \doteq c_1 \vee ... \vee x \doteq c_m]$ has been applied [10]. With $DCA(KB)$, it is assumed that all objects of the domain are denoted by the constants $c_1, ..., c_m$ mentioned in a knowledge base KB. There is no appropriate representation of $DCA(KB)$ for a DL knowledge base KB. However, this axiom can easily be applied by an appropriate decision procedure.

Definition 2. *Let p_i be an atomic assertion $A(a)$ or $R(a, b)$.*

1. *A clause $L_1 \vee ... \vee L_n$ is a disjunction of literals L_i of the form p_i or $\neg p_i$.*[1]
2. *A Horn clause is a clause with at most one positive literal.*
3. *An indefinite clause is a positive clause $p_1 \vee ... \vee p_n$ with $n > 1$.*

In the context of first-order logic, Reiter [2] proved that the CWA retains the consistency of *Horn knowledge bases*, i.e., knowledge bases only consisting of Horn clauses. The following theorem is a direct consequence of [1, Theorem 3]:

Theorem 1 (Inconsistency). *Let KB be a DL knowledge base. Then KB^+ is inconsistent if and only if there are positive ground literals $L_1, ..., L_n$ with $KB \models L_1 \vee ... \vee L_n$ and $KB \not\models L_i$ for all $i = 1, ..., n$.*

As a consequence, DL KBs with concept disjunction generally do not remain consistent by applying the CWA: If $KB = \{(\mathit{Intelligent} \sqcup \mathit{Rich})(\mathit{john})\}$, then $KB \not\models \mathit{Intelligent}(\mathit{john})$, $KB \not\models \mathit{Rich}(\mathit{john})$ and $KB \models \mathit{Intelligent}(\mathit{john}) \vee \mathit{Rich}(\mathit{john})$.

Consider $\exists R.\top(a) \in KB$. It is known that the constant a is related over R to an unknown role filler ω, usually referred to as a *Skolem constant*. Minker [1] analyses the case in which ω in each interpretation is *identified* with one of the constants $c_1, ..., c_m$ mentioned in KB (cf. domain closure). Then, by specifying $\exists R.\top(a)$ it holds that $KB \models R(a, c_1) \vee ... \vee R(a, c_m)$. Following to this implicit disjunction and referring to Theorem 1, DL knowledge bases with existential quantification (in $\mathcal{ALCN}$ existential restrictions $\exists R.C$ and at-least restrictions $(\geq n\, R)$) in general are also inconsistent with the CWA.

4 Generalized Closed World Assumption (GCWA)

The GCWA [1] is a generalization of the CWA which is guaranteed to retain the consistency of an indefinite knowledge base by considering indefinite clauses logically entailed by this knowledge base:

Definition 3. *Let KB be a DL knowledge base and PK the set of all positive clauses constructable from AS. Then GCWA(KB) =*

$$\{\neg p \mid p \in AS,\ K \in PK,\ KB \not\models p,\ \text{and if } KB \models p \vee K,\ \text{then } KB \models K\}$$

is the generalized closed world assumption of KB. The entailment relation $\models_{gcwa}$ considering this assumption for any assertion α is defined by

$$KB \models_{gcwa} \alpha \ \text{ iff } \ KB^* \models \alpha \text{ with } KB^* = KB \cup GCWA(KB).$$

Minker suggests that a literal of an indefinite clause can consistently be assumed to be false if there is a subclause without this literal. For example, in the context of propositional logic, $KB = \{p_1, p_1 \vee p_2, p_3 \vee p_4\}$ can consistently be augmented with $\{\neg p_2\}$ (while $KB^+ = KB \cup \{\neg p_2, \neg p_3, \neg p_4\}$ is inconsistent).

A consequence of retaining the consistency is that indefinite knowledge bases under the GCWA are not assumed to be complete: There can be assertions α with $KB \not\models_{gcwa} \alpha$ and $KB \not\models_{gcwa} \neg\alpha$. Another property of the GCWA is its relation to the CWA: If KB^+ is consistent, then $KB^* = KB^+$.

[1] In this investigation, a clause is always considered to be ground, i.e., only containing variable-free literals.

5 Extended Domain Closure

Since the domain closure axiom solely considers constants $c_1, ..., c_m$ mentioned in a knowledge base, Skolem constants ω can be identified only with these constants such that the interpretation of ω is $\omega \doteq c_1 \vee ... \vee \omega \doteq c_m$. A consequence of this is that the set of models of a given knowledge base is possibly reduced to unintended models, resulting in unexpected inferences: Suppose $KB_1 = \{Human(john), \exists hasChild.\top(john)\}$ and $DCA(KB_1) = (\forall x)[x \doteq john]$. Then there is only one "absurd" model in which John is a human and a child of himself such that $KB_1 \cup DCA(KB_1) \models hasChild(john, john)$. In addition, it is possible to construct a knowledge base that is inconsistent with the domain closure: If $KB_2 = \{Human(john), Human(susy), (\geq 3\ hasChild)(john)\}$ is augmented with its domain closure axiom, it is assumed that there are only the two objects denoted by *john* and *susy*, but the restriction $(\geq 3\ hasChild)(john)$ requires that there are at least three objects.

To avoid these problems, in this article it is proposed that a Skolem constant ω can also be identified with a new constant η. Under consideration of such a constant, every Skolem constant ω_k can be interpreted as follows:

$$\omega_k \doteq c_1 \vee ... \vee \omega_k \doteq c_m \vee \omega_k \doteq \eta_k \qquad (1)$$

Given an $\mathcal{ALCN}$ knowledge base KB, it is not obvious to determine all Skolem constants associated to existentially quantified assertions in KB or even all implicit assertions related to these Skolem constants. A foundation to solve this problem is to eliminate $\mathcal{T}$ such that $KB = \mathcal{A}_{\mathcal{T}}$ and to apply the tableau-based satisfiability algorithm for $\mathcal{ALCN}$ e.g. presented in [9].[2] If KB is satisfiable, let $\mathcal{S}(KB)$ be the full expanded tableau of KB, i.e., the union of all assertions of all complete and clash-free ABoxes. For example, it is known that $KB_3 = \{\forall P.\exists P.\top(a), \exists R.\top(a), \exists R.\exists P.A(a)\}$ is satisfiable and $\mathcal{S}(KB_3) = \{\forall P.\exists P.\top(a), \exists R.\top(a), \exists R.\exists P.A(a), R(a,\omega_1), R(a,\omega_2), \exists P.A(\omega_2), P(\omega_2,\omega_3), A(\omega_3)\}$. Regarding $\exists P.\top$ in scope of the value restriction, no Skolem constant is introduced, since there is no information of a filler of a with respect to the atomic role P. Cases in which Skolem constants such as ω_2 are identified with a are not considered in this context in order to avoid the introduction of infinitely many Skolem constants.

For querying purposes, there is the need to capture new constants η associated to Skolem constants. This is done by augmenting initially empty sets $NEW(T, v)$ with η_k for every role assertion $T(v, \omega_k) \in \mathcal{S}(KB)$. In the case that v itself is a Skolem constant, for each constant p the Skolem constant is identifiable with due to (1), the set $NEW(T, p)$ has to be augmented with the corresponding η_k. Concerning KB_3 e.g. $NEW(R, a) = \{\eta_1, \eta_2\}$, $NEW(P, a) = \{\eta_3\}$ and $NEW(P, \eta_2) = \{\eta_3\}$. If a role assertion $T(v, \omega_k)$ is not contained in all ABoxes of the tableau of $\mathcal{S}(KB)$, then there are models $\mathcal{I}$ of KB in which ω_k need not be considered. However, since there is an interest in all possibilities, sets $NEW(T, v)$ in this case are constructed as above.

[2] The modified algorithm needing only polynomial space is not considered here.

Definition 4 (Extended Domain Closure). *Let KB be a satisfiable DL knowledge base,* $c_1, ..., c_m$ *all constants mentioned in KB and* $\{\eta_1, ..., \eta_p\}$ *the union of all sets* $NEW(T, v)$*. Then*

$$DCA^{+}(KB) = (\forall x)[x \doteq c_1 \vee ... \vee x \doteq c_m \vee x \doteq \eta_1 \vee ... \vee x \doteq \eta_p]$$

is the extended domain closure axiom.

Regarding KB_1, under consideration of $DCA^{+}(KB_1) = (\forall x)[x \doteq john \vee x \doteq \eta_1]$, $hasChild(john, john)$ is not entailed from KB_1, and referring to KB_2, under extended domain closure for each ω_k associated to $(\geq 3\, hasChild)(john)$ a new constant η_k is introduced such that KB_2 remains consistent.

Definition 5 (Extended Unique Name Assumption). *Let KB be a DL knowledge base,* $c_1, ..., c_m$ *all constants mentioned in KB and* $\eta_1, ..., \eta_p$ *all new constants to be considered from* $DCA^{+}(KB)$*. Then*

$$\begin{aligned} UNA^{+}(KB) = &\{(c_i \not\doteq c_j) \mid 1 \leq i < j \leq m\} \cup \\ &\{(c_i \not\doteq \eta_k) \mid 1 \leq i \leq m,\ 1 \leq k \leq p\} \cup \\ &\{(\eta_k \not\doteq \eta_l) \mid 1 \leq k < l \leq p,\ \eta_k \text{ and } \eta_l \text{ are associated to the} \\ &\qquad \text{same at-least restriction}\} \end{aligned}$$

is the extended unique name assumption.

Note that by this definition, new constants η_k denote new (not necessarily existent) objects, i.e., different from all objects denoted by constants mentioned in *KB*. Therefore, by considering an extended domain closure and a corresponding extended unique name assumption, an assertion $\exists hasChild.\top(john)$ is interpreted as usually intended, namely to represent the information that John does have a child either known or not known to a knowledge base.

6 Local GCWA in $\mathcal{ALCN}$

6.1 Propositional Clause Form of $\mathcal{ALCN}$-Assertions

Arbitrary $\mathcal{ALCN}$ concept assertions in negation normal form (NNF)[3] can be transformed to a propositional clause form. The transformation of boolean assertions is depicted in Table 1, where u is a constant belonging to $DCA^{+}(KB)$.

Table 1. Transformation of boolean assertions to a clause form

$\top(u)$	$\Rightarrow$	$\top$	$\neg A(u)$	$\Rightarrow$	$\neg A(u)$
$\bot(u)$	$\Rightarrow$	$\bot$	$(C_1 \sqcap ... \sqcap C_n)(u)$	$\Rightarrow$	$C_1(u) \wedge ... \wedge C_n(u)$
$A(u)$	$\Rightarrow$	$A(u)$	$(C_1 \sqcup ... \sqcup C_n)(u)$	$\Rightarrow$	$C_1(u) \vee ... \vee C_n(u)$

The transformation of assertions with role restrictions requires a form of domain closure. For *all-quantified assertions* $\forall R.C(u)$ and $(\leq n\, R)(u)$, all constants

[3] In NNF, negation occurs only in front of atomic concepts and roles.

belonging to $DCA^+(KB)$ are considered while for *existential assertions* $\exists R.C(u)$ and $(\geq n\, R)(u)$, it suffices to consider the constants $c_1, ..., c_m$ and one resp. n new constants η_k. However, new constants associated to a query have to correspond to new constants considered for KB. Therefore, the transformation of existential assertions with respect to a role R and a constant u, besides all constants $c_1, ..., c_m$ mentioned in KB is applied with exactly the new constants $\eta_1, ..., \eta_r$ contained in $NEW(R, u)$. If all constants to be considered are determined, transformations of existential- and all-quantified assertions to a propositional (dual) clause form are based on known correspondences of $\mathcal{ALCN}$ to first-order logic (cf. [9, p. 54]) and are depicted in Table 2 resp. Table 3. Note that transformations of assertions with qualified role restrictions are defined recursively with respect to C.

Table 2. Transformation of existential assertions to a propositional (dual) clause form

$$\exists R.C(u) \Rightarrow_{NEW(R,u)} [R(u,c_1) \wedge C(c_1)] \vee ... \vee [R(u,c_m) \wedge C(c_m)] \vee [R(u,\eta_1) \wedge C(\eta_1)] \vee ... \vee [R(u,\eta_r) \wedge C(\eta_r)]$$

$$(\geq n\, R)(u) \Rightarrow_{NEW(R,u)} [R(u,c_{i_1}) \vee ... \vee R(u,c_{i_{|N|-n+1}})] \wedge \vdots \wedge [R(u,\eta_{k_1}) \vee ... \vee R(u,\eta_{k_{|N|-n+1}})]$$

Table 3. Transformation of all-quantified assertions to a propositional clause form

$$\forall R.C(u) \Rightarrow_{DCA^+} [\neg R(u,c_1) \vee C(c_1)] \wedge ... \wedge [\neg R(u,c_m) \vee C(c_m)] \wedge [\neg R(u,\eta_1) \vee C(\eta_1)] \wedge ... \wedge [\neg R(u,\eta_p) \vee C(\eta_p)]$$

$$(\leq n\, R)(u) \Rightarrow_{DCA^+} [\neg R(u,c_{i_1}) \vee ... \vee \neg R(u,c_{i_{n+1}})] \wedge \vdots \wedge [\neg R(u,\eta_{k_1}) \vee ... \vee \neg R(u,\eta_{k_{n+1}}) \vee EQ_\eta]$$

For the dual clause form of $\exists R.C(u)$ there is an equivalent clause form such that for each $\mathcal{ALCN}$ knowledge base $KB = \mathcal{A}_\mathcal{T}$ there is a corresponding set of clauses. Regarding $(\geq n\, R)(u)$, $|N|$ is the cardinality of $\{c_1, ..., c_m, \eta_1, ..., \eta_r\}$. In the context of a query or in the case that Skolem constants associated to an expression in scope of a value restriction are not introduced (see Sect. 5), it is possible that $n > |N|$ such that $(\geq n\, R)(u)$ is equivalent to $\bot(u)$. However, in these cases the problems mentioned for KB_1 and KB_2 in Sect. 5 cannot occur. Due to $UNA^+(KB)$, axioms $v_1 \not\doteq v_2$ with arbitrary constants v_1, v_2 need not be considered by transforming $(\geq n\, R)(u)$. Similarly, due to $UNA^+(KB)$, only equalities $\eta_k \doteq \eta_l$ involving pairs of new constants not associated to the same at-least restriction have to be considered for the transformation of $(\leq n\, R)(u)$ (abbreviated with EQ_η). Furthermore, if there are less than $n+1$ constants belonging to $DCA^+(KB)$, then $(\leq n\, R)(u)$ is equivalent to $\top(u)$.

6.2 Decision Procedure

Definition 6. *Let KB be an $\mathcal{ALCN}$ knowledge base, U the set of all constants belonging to $DCA^{+}(KB)$ and $\pi(PR) \subseteq PR$ the set of predicates assumed to be specified complete. Then $\models_{GCWA^{\pi}}$ for any assertion α is defined by*

$$KB \models_{GCWA^{\pi}} \alpha \text{ iff } KB \cup \pi(GCWA(KB)) \cup DCA^{+}(KB) \cup UNA^{+}(KB) \models \alpha,$$

where $\pi(GCWA(KB)) = \{\neg F(u) \in GCWA(KB) \mid u \in U \text{ and } F \in \pi(PR)\}$.

Besides the extended domain closure and its corresponding unique name assumption, with $\models_{GCWA^{\pi}}$ a GCWA is applied locally, i.e., only for predicates $F \in \pi(PR)$. The *AssertALCN* algorithm deciding whether $KB \models_{GCWA^{\pi}} \alpha$ is depicted in Table 4.

Table 4. *AssertALCN* algorithm specifying a local GCWA

Algorithm $AssertALCN(KB, \alpha, \pi(PR))$
input $\mathcal{ALCN}$ knowledge base $KB = \mathcal{T} \cup \mathcal{A}$, $\mathcal{ALCN}$-assertion α, set $\pi(PR)$ of predicates assumed to be specified complete
output TRUE if $KB \models_{GCWA^{\pi}} \alpha$, else FALSE

1. Expand $\mathcal{A}$ w.r.t. $\mathcal{T}$ such that $KB = \mathcal{A}_{\mathcal{T}}$
2. If there is an $\mathcal{ALCN}$ tableau proof for $KB \models_{una^{+}} \bot(a)$ for the constant a, evaluate to TRUE and stop
3. Determine all sets $NEW(T, v)$ containing new constants η_k for atomic roles T and constants v appearing in the full expanded tableau of 2.
4. Transform $NNF(KB)$ under consideration of sets $NEW(T, v)$ in a propositional clause form $\mathcal{C}l(KB)$
5. Compute the set $\mathcal{R}es(\mathcal{C}l(KB))$ by adding to $\mathcal{C}l(KB)$ all resolvents of $\mathcal{C}l(KB)$
6. Transform $NNF(\alpha)$ under consideration of sets $NEW(T, v)$ in a propositional clause form $\mathcal{C}l(\alpha)$
7. If there is a clause $K \in \mathcal{C}l(\alpha)$ neither containing $\top$ nor literals L and $\neg L$ and further not fulfilling any of the constraints

   (a) K or a subclause of K is in $\mathcal{R}es(\mathcal{C}l(KB))$
   (b) K contains a negative Literal $\neg A(a)$ resp. $\neg R(a, b)$ with $A \in \pi(PR)$ resp. $R \in \pi(PR)$ s.t. for all positive clauses of $\mathcal{R}es(\mathcal{C}l(KB))$ containing $A(a)$ resp. $R(a, b)$ there is a subclause in $\mathcal{R}es(\mathcal{C}l(KB))$ without $A(a)$ resp. $R(a, b)$

   evaluate to FALSE and stop, else evaluate to TRUE and stop

It is assumed that the reader is familiar with the propositional resolution calculus. After the elimination of $\mathcal{T}$ (1.), $KB = \mathcal{A}_{\mathcal{T}}$ is checked for satisfiability (2.). $GCWA(KB)$ as well as $DCA^{+}(KB)$ need not be considered for this check, since these assumptions are known to retain the consistency of KB. But KB is inconsistent with $UNA^+(KB)$, if $(\leq n\,R)(v)$, $R(v, v_1), ..., R(v, v_{n+1})$ and $v_i \not\approx v_j$ for all pairs (v_i, v_j) are in $\mathcal{S}(KB)$. Therefore the satisfiability check is evaluated with the entailment relation $\models_{una^+}$ considering $UNA^+(KB)$. There is no information whether $c_i \not\approx \omega_k$, and all axioms $\omega_k \not\approx \omega_l$ with pairs of Skolem constants associated to the same at-least restriction are introduced by the calculus [9] such that $\mathcal{S}(KB)$ is augmented only with axioms $c_i \not\approx c_j$ for pairs of constants mentioned in KB. If there is no tableau proof, the full expanded tableau contains all relevant assertions $T(v, \omega_k)$ to determine all sets $NEW(T, v)$ (3.). All assertions of KB (in NNF) are then transformed to a propositional clause form w.r.t. sets $NEW(T, v)$ (resp. $DCA^+(KB)$ obtained by these sets) (4.). Role assertions $R(a, b) \in KB$ are equivalent to the clause $R(a, b)$. The set $\mathcal{C}l(KB)$ is augmented with all its resolvents to $\mathcal{R}es(\mathcal{C}l(KB))$ (5.). The transformation of $NNF(\alpha)$ to $\mathcal{C}l(\alpha)$ (6.) is analogue to (4.). $AssertALCN$ then evaluates to TRUE, if for each non-tautological clause $K \in \mathcal{C}l(\alpha)$ either K or a subclause of K is in $\mathcal{R}es(\mathcal{C}l(KB))$ or if K contains $\neg A(a)$ resp. $\neg R(a, b)$ with $A \in \pi(PR)$ resp. $R \in \pi(PR)$ such that for all positive clauses of $\mathcal{R}es(\mathcal{C}l(KB))$ with $A(a)$ resp. $R(a, b)$, there is a subclause without $A(a)$ resp. $R(a, b)$ in $\mathcal{R}es(\mathcal{C}l(KB))$.

The soundness and completeness of $AssertALCN$ is based on the soundness and completeness of the well-known propositional resolution calculus. Given a set of clauses $\mathcal{C}l(KB)$, a set of clauses $\mathcal{C}l(\alpha)$ is proved with this calculus, symbolically $\mathcal{C}l(KB) \vdash_{Res} \mathcal{C}l(\alpha)$, if an empty clause can be derived from $\mathcal{C}l(KB) \cup \neg\mathcal{C}l(\alpha)$, where $\neg\mathcal{C}l(\alpha)$ has been transformed to clause form.

Theorem 2. *Let K be a non-tautological clause. Then K or a subclause of K is in $\mathcal{R}es(\mathcal{C}l(KB))$ if and only if $\mathcal{C}l(KB) \vdash_{Res} K$.*

Proof. (*sketch*) $\mathcal{R}es(\mathcal{C}l(KB))$ contains all implicit clauses of $\mathcal{C}l(KB)$. All further resolvents are obtained by the consideration of $\neg K = \neg L_1 \wedge ... \wedge \neg L_n$. "$\Rightarrow$": Consider $Cl = L_1 \vee ... \vee L_k \in \mathcal{R}es(\mathcal{C}l(KB))$ with $k \leq n$. It is obvious that the k-th resolvent of Cl and $\neg K$ is the empty clause. "$\Leftarrow$": Consider $\{\} \not\vdash_{Res} K$ (K is non-tautological), $K' = L'_1 \vee ... \vee L'_m \in \mathcal{R}es(\mathcal{C}l(KB))$ and there is no subclause of K' in $\mathcal{R}es(\mathcal{C}l(KB))$. If the negation of only k literals L'_i is contained in $\neg K$, then the k-th resolvent of K' and $\neg K$ consists of all literals L'_j whose negation is not contained in $\neg K$ and there are no further resolvents of $\mathcal{R}es(\mathcal{C}l(KB))$ and $\neg K$. Therefore all $\neg L'_i$ must be contained in $\neg K$ and m must be lower than or equal than n in order to derive the empty clause. □

Since $\mathcal{C}l(KB) \vdash_{Res} K_1 \wedge ... \wedge K_n$ iff $\mathcal{C}l(KB) \vdash_{Res} K_1, ..., \mathcal{C}l(KB) \vdash_{Res} K_n$, all clauses $K_1, ..., K_n$ are independent of each other. Consequently, Theorem 2 also holds for a set of clauses $\mathcal{C}l(\alpha)$ such that decisions obtained by executing step 7 (a) are sound and complete. Finally, step 7 (b) exactly corresponds to the consideration of $\pi(GCWA(KB))$. $AssertALCN$ can further be simplified by deleting all superclauses of $\mathcal{R}es(\mathcal{C}l(KB))$. Then, in step 7 (b) it suffices to check if there are positive clauses in $\mathcal{R}es(\mathcal{C}l(KB))$ containing $A(a)$ resp. $R(a, b)$.

Example 1. Let $KB = \{\exists R.A(a), R(a,b), A(b)\}$ and suppose there is interest in deciding whether $KB \models_{GCWA^{\pi}} \forall R.A(a)$ for $\pi(PR) = \{R\}$. It is obvious that $KB \not\models_{una^+} \bot(a)$. The fact that a itself is mentioned in KB is independent of this. $DCA^+(KB) = (\forall x)[x \doteq a \vee x \doteq b]$, since the $\exists$-rule (cf. [9]) cannot be applied such that all sets $NEW(T,v)$ are empty. The transformation of $\exists R.A(a)$ is

$$\exists R.A(a) \Rightarrow_{NEW(R,a)} [R(a,a) \wedge A(a)] \vee [R(a,b) \wedge A(b)].$$

The clauses obtained by this dual clause form are redundant, since for each clause the subclause $R(a,b)$ or $A(b)$ is in $\mathcal{C}l(KB)$ such that $\mathcal{C}l(KB) = \mathcal{R}es(\mathcal{C}l(KB)) = \{R(a,b), A(b)\}$. The propositional clause form of $\forall R.A(a)$ is $\{\neg R(a,a) \vee A(a), \neg R(a,b) \vee A(b)\}$. The second clause follows from KB w.r.t. $\models_{GCWA^{\pi}}$, since $A(b)$ is in $\mathcal{R}es(\mathcal{C}l(KB))$ and the first clause, since $R \in \pi(PR)$ and there are no positive clauses with $R(a,a)$ in $\mathcal{R}es(\mathcal{C}l(KB))$. Consequently, it holds that $KB \models_{GCWA^{\pi}} \forall R.A(a)$.

Example 2. Suppose $KB = \{\forall R.A(a), \exists R.\top(a)\}$ with $NEW(R,a) = \{\eta_1\}$. There is interest in deciding whether $KB \models_{GCWA^{\pi}} \neg A(a)$ with $\pi(PR) = \{A, R\}$. The clause form of KB is $\{\neg R(a,a) \vee A(a), \neg R(a,\eta_1) \vee A(\eta_1), R(a,a) \vee R(a,\eta_1)\}$. The resolvents $A(a) \vee R(a,\eta_1)$ as well as $R(a,a) \vee A(\eta_1)$ can be derived. Since in $\mathcal{R}es(\mathcal{C}l(KB))$ there is the clause $A(a) \vee R(a,\eta_1)$, but not the clause $R(a,\eta_1)$, the instance check given above is not successful.

7 Conclusion

We proposed a method for applying a local closed world assumption for DLs including indefinite knowledge which is also appropriate for users not experienced with non-monotonic DLs. As mentioned in the introduction, under the OWA there are often answers to queries that are not intended by users. Referring to the approach presented, these answers are due to the presence of negative literals in the corresponding propositional clause form of the query. By applying the GCWA, a subset of those negative literals whose non-negative counterpart cannot be proven is assumed to be true. This subset in general is strict, since under the GCWA indefinite knowledge remains indefinite, i.e., an indefinite knowledge base under the GCWA is not assumed to contain complete knowledge. Following to this, there are still answers to queries that are not intended by a lot of users.

Further investigations could be adressed to the extension of this approach to DLs more expressive than $\mathcal{ALCN}$ as well as to DLs with no restrictions on the terminological component, i.e., allowing for general concept inclusions (GCIs). Though the set $\mathcal{R}es(\mathcal{C}l(KB))$ for each KB has to be determined only once, its computation is highly intractable such that an optimization of the proposed method is of particular importance.

Acknowledgements. Special thanks go to Carola Eschenbach from the University of Hamburg and Ralf Möller from the Hamburg University of Technology.

References

1. Minker, J.: On indefinite databases and the Closed World Assumption. In: LNCS 138, pages 292–308. Springer, 1982.
2. Reiter, R.: On Closed World Data Bases. In Gallaire, H., Minker, J. (eds.) Logic and Data Bases, pages 55–76. New York: Plenum, 1978.
3. Cadoli, M., Donini, F.M., Schaerf, M.: Closed World Reasoning in Hybrid Systems. In Proceedings of the 5th International Symposium on Methodologies for Intelligent Systems (ISMIS-90), pages 474–481. North Holland, 1990.
4. Bonatti, P., Lutz, C., Wolter, F.: Description logics with circumscription. In Knowledge Representation, pages 400-410, 2006.
5. Donini, F.M., Lenzerini, M., Nardi, D., Nutt, W., Schaerf, A.: Adding Epistemic Operators to Concept Languages. Principles of Knowledge Represenation and Reasoning, pages 342–353, Cambridge, 1992.
6. Schaerf, A.: Query Answering in Concept-Based Knowledge Representation Systems: Algorithms, Complexity and Semantic Issues. PhD thesis, Dipartimento di Informatica e Sistemistica, Universita di Roma "La Sapienza", 1994.
7. Grimm, S., Motik, B.: Closed-World Reasoning in the Semantic Web through Epistemic Operators. In: CEUR Proceedings of the OWL Experiences and Directions Workshop, Galway, Ireland, 2005.
8. Donini, F.M., Nardi, D., Rosati, R.: Description logics of minimal knowledge and negation as failure. ACM Transactions on Computational Logic, 3(2), pages 177-225, 2002.
9. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F., editors. The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, January 2003.
10. Brachman, R.J., Levesque, H.J.: Knowledge Representation and Reasoning. San Francisco, CA: Morgan Kaufmann, 2004.