TECHNISCHE UNIVERSITÄT DRESDEN

Dresden University of Technology
Institute for Theoretical Computer Science
Chair for Automata Theory

LTCS–Report

Expressive Non-Monotonic Description Logics Based on Circumscription

Piero Bonatti, Carsten Lutz, Frank Wolter

LTCS-Report 05-06

Lehrstuhl für Automatentheorie
Institut für Theoretische Informatik
TU Dresden
http://lat.inf.tu-dresden.de

Hans-Grundig-Str. 25
01062 Dresden
Germany

Expressive Non-Monotonic Description Logics Based on Circumscription

Piero Bonatti
Dept. of Physics
U. of Naples
Italy

Carsten Lutz
Inst. for Theoretical CS
TU Dresden
Germany

Frank Wolter
Dept. of CS
U. of Liverpool
UK

Abstract

Recent applications of description logics (DLs) strongly suggest the integration of non-monotonic features into DLs, with particular attention to defeasible inheritance. However, the existing non-monotonic extensions of DLs are usually based on default logic or autoepistemic logic, and have to be seriously restricted in expressive power to preserve the decidability of reasoning. In particular, such DLs allow the modelling of defeasible inheritance only in a very restricted form, where non-monotonic reasoning is limited to individuals that are explicitly identified by constants in the knowledge base. In this paper, we consider non-monotonic extensions of expressive DLs based on circumscription. We prove that reasoning in such DLs is decidable even without the usual, strong restrictions in expressive power. We pinpoint the exact computational complexity of reasoning as complete for NP^{NExp} and NExp^{NP}, depending on whether or not the number of minimized and fixed predicates is assumed to be bounded by a constant. These results assume that only concept names (and no role names) can be minimized and fixed during minimization. On the other hand, we show that fixing role names during minimization makes reasoning undecidable.

1 Introduction

Early KR formalisms such as semantic networks and frames usually included a wealth of features in order to provide as powerful representational capabilities as possible [27, 22]. Most notably, such formalisms admitted a structured representation of classes and objects similar to modern description logics (DLs), but also mechanisms for defeasible inheritance and other features nowadays provided by non-monotonic logics (NMLs). When the theory of KR was developed further, these all-embracing approaches were largely given up due to semantic and computational problems. The subsequent focussing on more specialized formalisms caused DLs and NMLs to develop into two independent subfields of KR. Consequently, modern description logics such as $\mathcal{SHIQ}$ fail to include any non-monotonic features [15].

Since the birth of DLs as a subfield of KR, there has been a continuous interest in the (re-)integration of non-monotonic features into description logics. Due to the advent of several new application areas, this interest has recently reached new peaks.

For example, DLs are nowadays a popular tool for the formalization of biomedical ontologies such as GALEN [29] and SNOMED [9]. As argued by Rector et al. in [28, 32], such ontologies have to support defeasible inheritance to represent knowledge such as "in humans, the heart is usually located on the left-hand side of the body; in humans with situs inversus, the heart is located on the right-hand side of the body". Another recent application of DLs is their use as an ontology language for the semantic web [3], and the feedback of DL users from this field reveals substantial interest in the typical nonmonotonic features of object oriented languages such as default attributes, defeasible inheritance, and overriding.

Many different approaches to adding non-monotonic features to DLs have been proposed, but none of them is fully convincing for modelling defeasible inheritance [1, 2, 34, 11, 10, 25, 16, 13, 31]. The main problem is taming the computational power that arises when combining the expressiveness of DLs and NMLs: it is nontrivial to identify a non-monotonic DL that enjoys the expressive power of modern DLs, admits non-monotonic reasoning without severe restrictions, and is decidable. For example, the non-monotonic DL proposed in [25] includes a mechanism for default reasoning, but has to impose severe restrictions on DL expressiveness to keep reasoning decidable. Another approach to non-monotonic DLs consists in including the (auto)epistemic operator "K" [11, 10]. However, in all known decidability results concerning such DLs, operator K can be used in a non-monotonic way only in queries, but not in the knowledge base. This is a serious limitation since it precludes the modelling of defeasible inheritance. The approaches [1, 2, 34, 16] are based on default logic [30] and share a common restriction: default rules can be applied to an individual only if it has a *name*, that is, it is denoted by an individual constant occurring in the knowledge base. Since the models of DL knowledge bases usually include a large number of implicit (nameless) individuals enforced via existential restrictions, the limitation of default rule application to named individuals is highly restrictive. Finally, the approaches described in [13, 31] aim at extending DLs with non-monotonic rules that, however, apply only to named individuals.

In view of the computational problems affecting non-monotonic DLs based on default logic or autoepistemic logic, it is surprising that circumscription [20] has never been investigated in the context of DLs. After all, circumscription is known to be slightly less expressive than the other major formalizations of non-monotonicity [5]. In this paper, *we advocate the use of circumscription to obtain non-monotonic extensions of expressive DLs that are decidable and impose no serious restrictions on expressive power*. In particular, we show how to obtain a family of DLs that allow to model defeasible inheritance without the limitation to named individuals.

The central tool for knowledge representation in our family of non-monotonic DLs are *circumscribed knowledge bases (cKBs)*. Like standard DL knowledge bases, a cKB comprises a TBox for representing terminological knowledge and an ABox for representing knowledge about individuals. Additionally, a cKB is equipped with a *circumscription pattern* that lists predicates (i.e., concept and role names) to be minimized: in models of the cKB, the extension of minimized predicates is required to be minimal w.r.t. set inclusion. Following McCarthy [21], the minimized predicates will often be "abnormality predicates" identifying instances that are not typical for their class. Circumscription patterns can require other predicates to be fixed during minimization, or allow them to vary freely. Moreover, circumscription patterns allow to express preferences between minimized predicates in terms of a partial ordering. As argued in [2], this is of great importance to ensure a smooth interplay between defeasible inheritance and DL subsumption.

The main contribution of this paper is a detailed analysis of the computational properties of non-monotonic DLs based on circumscription. We show that, in the expressive DLs $\mathcal{ALCIO}$ and $\mathcal{ALCQO}$, satisfiability and subsumption w.r.t. circumscribed knowledge bases are decidable if only concept names (and no role names) are minimized and fixed. More precisely, we prove that satisfiability in both DLs w.r.t. such *concept-circumscribed* knowledge bases is NExp^{NP}-complete. In contrast, reasoning becomes undecidable if role names are allowed to be fixed during minimization. The undecidability result already applies to the basic propositionally-closed DL $\mathcal{ALC}$, and even if TBoxes are empty. We also give a finer-grained analysis of the complexity of reasoning w.r.t. concept-circumscribed KBs: when imposing a constant bound on the number of minimized and fixed concept names, the complexity of satisfiability drops to NP^{NExp}-completeness. All lower complexity bounds apply to the description logic $\mathcal{ALC}$.

It is interesting to note that our results are somewhat unusual from the perspective of NMLs. First, the *arity* of predicates has an impact on decidability: fixing concept names (unary predicates) does not impair decidability, whereas fixing a single role name (binary predicate) leads to a strong undecidability result. Second, the *number* of predicates that are minimized or fixed (bounded vs. unbounded) affects the computational complexity of reasoning. Although (as we briefly argue) a similar effect can be observed in propositional logic with circumscription, this has, to the best of our knowledge, never been explicitly noted.

2 Description Logics and Circumscription

In DLs, *concepts* are inductively defined with the help of a set of *constructors*, starting with a set $\mathsf{N_C}$ of *concept names*, a set $\mathsf{N_R}$ of *role names*, and (possibly) a set $\mathsf{N_I}$ of *individual names* (all countably infinite). We use the term *predicates* to refer to elements of $\mathsf{N_C} \cup \mathsf{N_R}$. The concepts of the expressive DL $\mathcal{ALCQIO}$ are formed using the constructors shown in Figure 1. There, the inverse role constructor is the only role constructor, whereas the remaining six constructors are concept constructors. In Figure 1 and throughout this paper, we use $\#S$ to denote the cardinality of a set S, a and b to denote individual names, r and s to denote roles (i.e., role names and inverses thereof), A, B to denote concept names, and C, D to denote (possibly complex) concepts. As usual, we use $\top$ as abbreviation for an arbitrary (but fixed) propositional tautology, $\bot$ for $\neg\top$, $\rightarrow$ and $\leftrightarrow$ for the usual Boolean abbreviations, $\exists r.C$ (*existential restriction*) for $(\geqslant 1\ r\ C)$, and $\forall r.C$ (*universal restriction*) for $(\leqslant 0\ r\ \neg C)$.

In this paper, we will not be concerned with $\mathcal{ALCQIO}$ itself, but with several of its fragments. The basic such fragment allows only for negation, conjunction, disjunction, and universal and existential restrictions, and is called $\mathcal{ALC}$. The availability

Name	Syntax	Semantics
inverse role	r^-	$(r^\mathcal{I})^\smile = \{(d,e) \mid (e,d) \in r^\mathcal{I}\}$
nominal	$\{a\}$	$\{a^\mathcal{I}\}$
negation	$\neg C$	$\Delta^\mathcal{I} \setminus C^\mathcal{I}$
conjunction	$C \sqcap D$	$C^\mathcal{I} \cap D^\mathcal{I}$
disjunction	$C \sqcup D$	$C^\mathcal{I} \cup D^\mathcal{I}$
at-least restriction	$(\geqslant n\ r\ C)$	$\{d \in \Delta^\mathcal{I} \mid \#\{e \in C^\mathcal{I} \mid (d,e) \in r^\mathcal{I}\} \geq n\}$
at-most restriction	$(\leqslant n\ r\ C)$	$\{d \in \Delta^\mathcal{I} \mid \#\{e \in C^\mathcal{I} \mid (d,e) \in r^\mathcal{I}\} \leq n\}$

Figure 1: Syntax and semantics of $\mathcal{ALCQIO}$.

of additional constructors is indicated by concatenation of a corresponding letter: $\mathcal{Q}$ stands for number restrictions, $\mathcal{I}$ stands for inverse roles, and $\mathcal{O}$ for nominals. This explains the name $\mathcal{ALCQIO}$, and also allows us to refer to fragments such as $\mathcal{ALCIO}$, $\mathcal{ALCQO}$, and $\mathcal{ALCQI}$.

The semantics of $\mathcal{ALCQIO}$-concepts is defined in terms of an *interpretation* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$. The *domain* $\Delta^\mathcal{I}$ is a non-empty set of individuals and the *interpretation function* $\cdot^\mathcal{I}$ maps each concept name $A \in \mathsf{N_C}$ to a subset $A^\mathcal{I}$ of $\Delta^\mathcal{I}$, each role name $r \in \mathsf{N_R}$ to a binary relation $r^\mathcal{I}$ on $\Delta^\mathcal{I}$, and each individual name $a \in \mathsf{N_I}$ to an individual $a^\mathcal{I} \in \Delta^\mathcal{I}$. The extension of $\cdot^\mathcal{I}$ to inverse roles and arbitrary concepts is inductively defined as shown in the third column of Figure 1. An interpretation $\mathcal{I}$ is called a *model* of a concept C if $C^\mathcal{I} \neq \emptyset$. If $\mathcal{I}$ is a model of C, we also say that C is *satisfied* by $\mathcal{I}$.

A *TBox* is a finite set of *general concept implications (GCIs)* $C \sqsubseteq D$ where C and D are concepts. As usual, we use $C \doteq D$ as an abbreviation for $C \sqsubseteq D$ and $D \sqsubseteq C$. An *ABox* is a finite set of *concept assertions* $C(a)$ and *role assertions* $r(a,b)$, where a, b are individual names, r is a role name, and C is a concept. An interpretation $\mathcal{I}$ *satisfies* (i) a GCI $C \sqsubseteq D$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$, (ii) an assertion $C(a)$ if $a^\mathcal{I} \in C^\mathcal{I}$, and (iii) an assertion $r(a,b)$ if $(a^\mathcal{I}, b^\mathcal{I}) \in r^\mathcal{I}$. Then, $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$ if it satisfies all implications in $\mathcal{T}$, and a *model* of an ABox $\mathcal{A}$ if it satisfies all assertions in $\mathcal{A}$.

Circumscription with Partial Priority Ordering

Circumscription is a logical approach suitable for modelling what *normally* or *typically* holds, and thus admits the modeling of defeasible inheritance [21, 18]. The idea is to define, in a standard first-order language, both domain knowledge and so-called *abnormality predicates* that describe what does not fit the normality criteria of the application domain. To capture the intuition that abnormality is exceptional, inference is not based on the set of all models of the resulting theory as in classical logic, but rather restricted to those models where the extension of the abnormality predicates is *minimal*. Intuitively, this means that reasoning is done only on models that are "as normal as possible".

Since description logics are fragments of first-order logic, circumscription can be readily applied. Using $\mathcal{ALC}$ syntax, we can assert that mammals normally inhabitate land, and that whales do not live on land:

$$\begin{array}{rcl} \mathtt{Mammal} & \sqsubseteq & \exists\mathtt{habitat.Land} \sqcup \mathtt{Ab}_{\mathtt{Mammal}} \\ \mathtt{Whale} & \sqsubseteq & \mathtt{Mammal} \sqcap \neg\exists\mathtt{habitat.Land} \end{array}$$

The upper inclusion states that any mammal not inhabitating land is an abnormal mammal, thus satisfying the abnormality predicate $\mathtt{Ab}_{\mathtt{Mammal}}$. When applying circumscription to the above TBox, we should thus only consider models where the extension of $\mathtt{Ab}_{\mathtt{Mammal}}$ is minimal. However, there is more than one way of defining such preferred models. The reason is that there are essentially two options to treat the remaining predicates during minimization of the abnormality predicate: we may either fix their extensions or let them vary freely. It should not come as a surprise that this decision may have a strong impact on the result of reasoning. In general, varying more predicates means that more subsumptions become derivable. For example, consider the above TBox. Even if all non-minimized predicates are fixed, we get the following subsumptions:

$$\begin{array}{rcl} \mathtt{Whale} & \sqsubseteq & \mathtt{Ab}_{\mathtt{Mammal}} \\ \mathtt{Ab}_{\mathtt{Mammal}} & \doteq & \mathtt{Mammal} \sqcap \neg\exists\mathtt{habitat.Land}. \end{array} \tag{$\dagger$}$$

If it is considered *very* unlikely for a mammal not to live on land, then one would expect that only those mammals do not live on land for which this was explicitly stated: whales. Consequently, the following subsumption should be derivable:

$$\mathtt{Whale} \doteq \mathtt{Ab}_{\mathtt{Mammal}}. \tag{$\ddagger$}$$

The way to achieve this is to let the role `habitat` and the concept name `Land` vary freely, and to fix only `Mammal` and `Whale`. The result is that both (†) and (‡) are derivable.

We can go even further and consider whales abnormal to such a degree that we do not believe they exist unless there is evidence that they do. Then we should, additionally, let `Whale` vary freely. The result is that (†) and (‡) can still be derived, and additionally we have $\mathtt{Whale} \doteq \mathtt{Ab}_{\mathtt{Mammal}} \doteq \bot$. We can then use an ABox to add evidence that whales exist, e.g. through the assertion `Whale(mobydick)`. As expected, the result of this change is that

$$\mathtt{Whale} \doteq \mathtt{Ab}_{\mathtt{Mammal}} \doteq \{\mathtt{mobydick}\}.$$

Evidence for the existence of another, anonymous whale could be generated by adding the ABox assertion `Male(mobydick)` and the TBox statement

$$\mathtt{Whale} \sqsubseteq \exists\mathtt{mother}.(\mathtt{Whale} \sqcap \neg\mathtt{Male})$$

with `mother` and `Male` varying freely. In general, it depends on the application which combination of fixed and varying predicates is appropriate. Therefore, the formalisms proposed in this paper leave the freedom to the user to choose the predicates that are minimized, fixed, and varying.

It has been convincingly argued by Baader and Hollunder in [2] that there is an interplay between subsumption and abnormality predicates that should be addressed in non-monotonic DLs. Consider, for example, the following TBox:

$$\begin{array}{rcl}
\mathtt{User} & \sqsubseteq & \neg\exists\mathtt{hasAccessTo.ConfidentialFile} \sqcup \mathtt{Ab}_{\mathtt{User}} \\
\mathtt{Staff} & \sqsubseteq & \mathtt{User} \\
\mathtt{Staff} & \sqsubseteq & \exists\mathtt{hasAccessTo.ConfidentialFile} \sqcup \mathtt{Ab}_{\mathtt{Staff}} \\
\mathtt{BlacklistedStaff} & \sqsubseteq & \mathtt{Staff} \sqcap \neg\exists\mathtt{hasAccessTo.ConfidentialFile}
\end{array}$$

To get models that are "as normal as possible", as a first attempt we could minimize the two abnormality predicates $\mathtt{Ab}_{\mathtt{User}}$ and $\mathtt{Ab}_{\mathtt{Staff}}$ in parallel. Assume that `hasAccessTo` and `ConfidentialFile` are varying, and `User`, `Staff`, and `BlacklistedStaff` are fixed. Then, the result of parallel minimization is that staff members may or may not have access to confidential files, with equal preference. In the first case, they are abnormal users, and in the second case, they are abnormal staff. However, one may argue that the first option should be preferred: since $\mathtt{Staff} \sqsubseteq \mathtt{User}$ (but not the other way round), the normality information for staff is more specific than the normality information for users and should have higher priority.

In the generalization of circumscription used in this paper, the user can specify priorities between minimized predicates. Normally, these priorities will reflect the subsumption hierarchy (as computed w.r.t. the class of *all* models). Since the subsumption hierarchy is a partial order, the priorities between minimized predicates are assumed to form a partial order, too. This is similar to partially ordered priorities on default rules as proposed by Brewka [7], and more general than standard prioritized circumscription which assumes a total ordering [21, 17]. More information can be found in [2].

To define DLs with circumscription, we start by introducing circumscription patters. Such a pattern describes how individual predicates are treated during minimization.

Definition 1 (Circumscription pattern, $<_{\mathsf{CP}}$) *A* circumscription pattern *is a tuple* $\mathsf{CP} = (\prec, M, F, V)$ *where* $\prec$ *is a strict partial order over* M*, and* M*,* F*, and* V *are subsets of* $\mathsf{N_C} \cup \mathsf{N_R}$*, the* minimized*,* fixed*, and* varying *predicates, respectively. By* $\preceq$*, we denote the reflexive closure of* $\prec$*. Define a preference relation* $<_{\mathsf{CP}}$ *on interpretations by setting* $\mathcal{I} <_{\mathsf{CP}} \mathcal{J}$ *iff the following conditions hold:*

1. $\Delta^{\mathcal{I}} = \Delta^{\mathcal{J}}$ *and, for all* $a \in \mathsf{N_I}$, $a^{\mathcal{I}} = a^{\mathcal{J}}$,
2. *for all* $p \in F$, $p^{\mathcal{I}} = p^{\mathcal{J}}$,
3. *for all* $p \in M$*, if* $p^{\mathcal{I}} \not\subseteq p^{\mathcal{J}}$ *then there exists* $q \in M$, $q \prec p$*, such that* $q^{\mathcal{I}} \subset q^{\mathcal{J}}$,
4. *there exists* $p \in M$ *such that* $p^{\mathcal{I}} \subset p^{\mathcal{J}}$ *and for all* $q \in M$ *such that* $q \prec p$, $q^{\mathcal{I}} = q^{\mathcal{J}}$.

When $M \cup F \subseteq \mathsf{N_C}$ *(i.e., the minimized and fixed predicates are all concepts) we call* $(\prec, M, F, V)$ *a* concept circumscription pattern. △

We use the term *concept circumscription* if only concept circumscription patterns are admitted. Based on circumscription patterns, we can define circumscribed DL knowledge bases and their models.

Definition 2 (Circumscribed KB) *A* circumscribed knowledge base (cKB) *is an expression* $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A})$*, where* $\mathcal{T}$ *is a TBox,* $\mathcal{A}$ *an ABox, and* $\mathsf{CP} = (\prec, M, F, V)$ *a circumscription pattern such that* M, F, V *partition the predicates used in* $\mathcal{T}$ *and* $\mathcal{A}$*. An interpretation* $\mathcal{I}$ *is a* model *of* $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A})$ *if it is a model of* $\mathcal{T}$ *and* $\mathcal{A}$ *and there exists no model* $\mathcal{I}'$ *of* $\mathcal{T}$ *and* $\mathcal{A}$ *such that* $\mathcal{I}' <_{\mathsf{CP}} \mathcal{I}$*.* △

A cKB $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A})$ is called a *concept-circumscribed knowledge base (KB)* if CP is a concept cicrcumscription pattern. The main reasoning tasks of description logics are defined with respect to circumscribed knowledge bases in the expected way.

Definition 3 (Reasoning problems)

- *A concept* C *is* satisfiable *w.r.t. a cKB* $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A})$ *if some model* $\mathcal{I}$ *of* $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A})$ *satisfies* $C^{\mathcal{I}} \neq \emptyset$*.*
- *A concept* C *is* subsumed by *a concept* D *w.r.t. a cKB* $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A})$ *(written* $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A}) \models C \sqsubseteq D$*) if* $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ *for all models* $\mathcal{I}$ *of* $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A})$*.*
- *An individual name* a *is an* instance of *a concept* C *w.r.t. a cKB* $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A})$ *(written* $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A}) \models C(a)$*) if* $a^{\mathcal{I}} \in C^{\mathcal{I}}$ *for all models* $\mathcal{I}$ *of* $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A})$*.*

△

These reasoning problems can be polynomially reduced to one another: first, C is satisfiable w.r.t. $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A})$ iff $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A}) \not\models C \sqsubseteq \bot$, and $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A}) \models C \sqsubseteq D$ iff $C \sqcap \neg D$ is unsatisfiable w.r.t. $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A})$. And second, C is satisfiable w.r.t. $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A})$ iff $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A}) \not\models \neg C(a)$, where a is an individual name not appearing in $\mathcal{T}$ and $\mathcal{A}$; conversely, we have $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A}) \models C(a)$ iff $A \sqcap \neg C$ is unsatisfiable w.r.t. $\mathsf{Circ}_{\mathsf{CP}'}(\mathcal{T},\mathcal{A} \cup \{A(a)\})$, where A is a concept name not occurring in $\mathcal{T}$ and $\mathcal{A}$, and CP' is obtained from CP by adding A to M (and leaving $\prec$ as it is). In this paper, we use satisfiability w.r.t. cKBs as the basic reasoning problem.

Note that partially ordered circumscription becomes standard parallel circumscription if the empty relation is used for $\prec$. Technically, partially ordered circumscription lies in between prioritized circumscription [21, 17] and *nested circumscription* [19]. It extends prioritized circumscription by admitting *partial* orders and, compared to nested circumscription, has the advantage of being technically simpler while still offering sufficient expressive power to address the interaction between subsumption and circumscription in DLs.

It is folklore in circumscription that there is a close connection between minimized concepts and fixed concepts: using TBoxes, the latter can be simulated by the former. Let C_0 be a concept and $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A})$ a concept-circumscribed KB with $\mathsf{CP} = (\prec, M, F, V)$ and $F = \{A_1, \ldots, A_k\}$. Define a new pattern $\mathsf{CP}' = (\prec, M', \emptyset, V)$ with

- $M' = M \cup \{A_1, \ldots, A_k, A'_1, \ldots, A'_k\}$, $A'_1, \ldots, A'_k$ concept names not occurring in C_0, M, F, $\mathcal{T}$, and $\mathcal{A}$;
- $\mathcal{T}' = \mathcal{T} \cup \{A'_i \doteq \neg A_i \mid 1 \leq i \leq k\}$.

It is not difficult to see that C_0 is satisfiable w.r.t. $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$ iff it is satisfiable w.r.t. $\mathsf{Circ}_{\mathsf{CP}'}(\mathcal{T}', \mathcal{A})$. Thus, we get the following.

Lemma 4 *Satisfiability w.r.t. concept-circumscribed KBs can be polynomially reduced to satisfiability w.r.t. concept-circumscribed KBs that have no fixed predicates.*

Also in the case of general cKBs, fixed concept names can be simulated by minimized concept names. However, such a simulation cannot be done for role names since Boolean operators on roles are not avaliable in standard DLs such as $\mathcal{ALCQIO}$.

3 Upper Bounds

The main contribution of this paper is to show that there are many description logics with circumscription that are decidable, and to perform a detailed analysis of the computational complexity of such logics. In particular, we will show that $\mathcal{ALCIO}$ and $\mathcal{ALCQO}$ with concept circumscription are decidable. We prepare the decidability proof for these logics by showing that if a concept is satisfiable w.r.t. a concept-circumscribed KB, then it is satisfiable in a model of bounded size. We use $|C|$ to denote the length of the concept C, i.e., the number of symbols needed to write C. The *size* $|\mathcal{T}|$ of a TBox $\mathcal{T}$ is $\sum_{C \sqsubseteq D \in \mathcal{T}} |C| + |D|$, and the *size* $|\mathcal{A}|$ of an ABox $\mathcal{A}$ is the sum of the sizes of all assertions in $\mathcal{A}$, where the size of each role assertion is 1 and the size of concept assertions $C(a)$ is $|C|$.

Lemma 5 *Let C_0 be a concept, $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$ a concept-circumscribed KB, and $n := |C_0| + |\mathcal{T}| + |\mathcal{A}|$. If C_0 is satisfiable w.r.t. $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$, then the following holds:*

(i) If $\mathcal{T}$, $\mathcal{A}$ and C_0 are formulated in $\mathcal{ALCIO}$, then C_0 is satisfied in a model $\mathcal{I}$ of $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$ with $\#\Delta^{\mathcal{I}} \leq 2^{2n}$.

(ii) If $\mathcal{T}$, $\mathcal{A}$ and C_0 are formulated in $\mathcal{ALCQO}$ and m is the maximal parameter occuring in a number restriction in $\mathcal{T}$, $\mathcal{A}$, or C_0, then C_0 is satisfied in a model $\mathcal{I}$ of $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$ with $\#\Delta^{\mathcal{I}} \leq 2^{2n} \times (m+1) \times n$.

Proof. Let CP, $\mathcal{T}$, $\mathcal{A}$, and C_0 be as in the lemma. We may assume that $\mathcal{A} = \emptyset$ as every assertion $C(a)$ can be expressed as an implication $\{a\} \sqsubseteq C$, and every assertion $r(a, b)$ can be expressed as $\{a\} \sqsubseteq \exists r.\{b\}$. Denote by $\mathsf{cl}(C, \mathcal{T})$ the smallest set of concepts that contains all subconcepts of C, all subconcepts of concepts appearing in $\mathcal{T}$, and is closed under single negations.

Let $\mathcal{I}$ be a common model of C_0 and $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$, and let $d_0 \in C_0^{\mathcal{I}}$. Define an equivalence relation "$\sim$" on $\Delta^{\mathcal{I}}$ by setting $d \sim d'$ iff

$$\{C \in \mathsf{cl}(C_0, \mathcal{T}) \mid d \in C^{\mathcal{I}}\} = \{C \in \mathsf{cl}(C_0, \mathcal{T}) \mid d' \in C^{\mathcal{I}}\}.$$

We use $[d]$ to denote the equivalence class of $d \in \Delta^{\mathcal{I}}$ w.r.t. the "$\sim$" relation. Pick from each equivalence class $[d]$ exactly one member and denote the resulting subset of $\Delta^{\mathcal{I}}$ by Δ'.

We first prove Point (i). Thus, assume that $\mathcal{T}$ and C_0 are formulated in $\mathcal{ALCIO}$. We define a new interpretation $\mathcal{J}$ as follows:

$$\begin{array}{rcl}
\Delta^{\mathcal{J}} & := & \Delta' \\
A^{\mathcal{J}} & := & \{d \in \Delta' \mid d \in A^{\mathcal{I}}\} \\
r^{\mathcal{J}} & := & \{(d_1, d_2) \in \Delta' \times \Delta' \mid \exists d_1' \in [d_1], d_2' \in [d_2] : (d_1', d_2') \in r^{\mathcal{I}}\} \\
a^{\mathcal{J}} & := & d \in \Delta' \text{ if } a^{\mathcal{I}} \in [d].
\end{array}$$

The following claim is easily proved using induction on the structure of C.

Claim: For all $C \in \mathsf{cl}(C_0, \mathcal{T})$ and all $d \in \Delta^{\mathcal{I}}$, we have $d \in C^{\mathcal{I}}$ iff $d' \in C^{\mathcal{J}}$ for the element $d' \in [d]$ of $\Delta^{\mathcal{J}}$.

Thus, $\mathcal{J}$ is a model of $\mathcal{T}$ satisfying C_0. To show that $\mathcal{J}$ is a model of $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$, it thus remains to show that there is no model $\mathcal{J}'$ of $\mathcal{T}$ with $\mathcal{J}' <_{\mathsf{CP}} \mathcal{J}$. Assume to the contrary that there is such a $\mathcal{J}'$. We define an interpretation $\mathcal{I}'$ as follows:

$$\begin{array}{rcl}
\Delta^{\mathcal{I}'} & := & \Delta^{\mathcal{I}} \\
A^{\mathcal{I}'} & := & \bigcup\limits_{d \in A^{\mathcal{J}'}} [d] \\
r^{\mathcal{I}'} & := & \bigcup\limits_{(d_1, d_2) \in r^{\mathcal{J}'}} [d_1] \times [d_2] \\
a^{\mathcal{I}'} & := & a^{\mathcal{I}}.
\end{array}$$

It is a matter of routine to show the following:

Claim: For all concepts $C \in \mathsf{cl}(C_0, \mathcal{T})$ and all $d \in \Delta^{\mathcal{I}}$, we have $d \in C^{\mathcal{I}'}$ iff $d' \in C^{\mathcal{J}'}$ for the element $d' \in [d]$ from $\Delta^{\mathcal{J}}$.

It follows that $\mathcal{I}'$ is a model of $\mathcal{T}$. Observe that $A^{\mathcal{I}} \circ A^{\mathcal{I}'}$ iff $A^{\mathcal{J}} \circ A^{\mathcal{J}'}$ for each concept name A and $\circ \in \{\supseteq, \subseteq\}$. Therefore and since CP is a concept circumscription pattern, $\mathcal{I}' <_{\mathsf{CP}} \mathcal{I}$ follows from $\mathcal{J}' <_{\mathsf{CP}} \mathcal{J}$. We have derived a contradiction and conclude that $\mathcal{J}$ is a model of $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$. Thus we are done since the size of $\mathcal{J}$ is bounded by 2^{2n}.

Now for Point (ii). Pick, for each $d \in \Delta'$ and concept $(\geqslant n\ r\ C) \in \mathsf{cl}(C_0, \mathcal{T})$ such that $d \in (\geqslant n\ r\ C)^{\mathcal{I}}$, n elements from $\{d' \mid d' \in C^{\mathcal{I}}, (d, d') \in r^{\mathcal{I}}\}$. Also pick, for each concept $(\leqslant n\ r\ C) \in \mathsf{cl}(C_0, \mathcal{T})$ such that $d \in (\neg(\leqslant n\ r\ C))^{\mathcal{I}}$, $n+1$ elements from $\{d' \mid d' \in C^{\mathcal{I}}, (d, d') \in r^{\mathcal{I}}\}$. Denote by Δ'' the collection of the elements picked. Take for each $d \in \Delta'' \setminus \Delta'$ an element $d^s \in \Delta'$ such that $d \sim d^s$ and define an interpretation $\mathcal{J}$ by

$$\begin{array}{rcl}
\Delta^{\mathcal{J}} & := & \Delta' \cup \Delta'' \\
A^{\mathcal{J}} & := & \{d \in \Delta' \cup \Delta'' \mid d \in A^{\mathcal{I}}\} \\
r^{\mathcal{J}} & := & \{(d_1, d_2) \in \Delta' \times (\Delta' \cup \Delta'') \mid (d_1, d_2) \in r^{\mathcal{I}}\} \\
 & & \cup\{(d_1, d_2) \in (\Delta'' \setminus \Delta') \times (\Delta' \cup \Delta'') \mid (d_1^s, d_2) \in r^{\mathcal{I}}\} \\
a^{\mathcal{J}} & := & d \text{ if } a^{\mathcal{I}} \in [d].
\end{array}$$

The following claim is easily proved.

Claim: For all $C \in \mathsf{cl}(C_0, \mathcal{T})$, we have the following:

(i) for all $d, d' \in \Delta^{\mathcal{J}}$, if $d \sim d'$, then $d \in C^{\mathcal{J}}$ iff $d' \in C^{\mathcal{J}}$;

(ii) for all $d \in \Delta^{\mathcal{I}}$, we have $d \in C^{\mathcal{I}}$ iff $d' \in C^{\mathcal{J}}$ for an element $d' \in [d]$ of $\Delta^{\mathcal{J}}$.

Thus, $\mathcal{J}$ is a model of $\mathcal{T}$ satisfying C_0. To show that $\mathcal{J}$ is a model of $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$, it thus remains to show that there is no model $\mathcal{J}'$ of $\mathcal{T}$ with $\mathcal{J}' <_{\mathsf{CP}} \mathcal{J}$. Assume to the contrary that there is such a $\mathcal{J}'$. We define an interpretation $\mathcal{I}'$. To this end, take for each $d \in \Delta^{\mathcal{I}} \setminus \Delta^{\mathcal{J}}$ the $d^p \in \Delta'$ such that $d \sim d^p$. Now define $\mathcal{I}'$ as follows

$$
\begin{array}{rcl}
\Delta^{\mathcal{I}'} & := & \Delta^{\mathcal{I}} \\
A^{\mathcal{I}'} & := & A^{\mathcal{J}'} \cup \{d \in \Delta^{\mathcal{I}} \setminus \Delta^{\mathcal{J}} \mid d^p \in A^{\mathcal{J}'}\} \\
r^{\mathcal{I}'} & := & r^{\mathcal{J}'} \cup \{(d_1, d_2) \in (\Delta^{\mathcal{I}} \setminus \Delta^{\mathcal{J}}) \times \Delta^{\mathcal{I}} \mid (d_1^p, d_2) \in r^{\mathcal{J}'}\} \\
a^{\mathcal{I}'} & := & a^{\mathcal{I}}.
\end{array}
$$

Again, it is a matter of routine to show:

Claim: For all concepts $C \in \mathsf{cl}(C_0, \mathcal{T})$ and all $d \in \Delta^{\mathcal{I}}$, we have $d \in C^{\mathcal{I}'} \cap \Delta^{\mathcal{J}}$ iff $d \in C^{\mathcal{J}'}$ and $d \in C^{\mathcal{I}'} \cap (\Delta^{\mathcal{I}} \setminus \Delta^{\mathcal{J}})$ iff $d^p \in C^{\mathcal{J}'}$ for the element $d^p \in [d]$ from Δ'.

It follows that $\mathcal{I}'$ is a model for $\mathcal{T}$. Observe that $A^{\mathcal{I}} \circ A^{\mathcal{I}'}$ iff $A^{\mathcal{J}} \circ A^{\mathcal{J}'}$ for each concept name A and $\circ \in \{\supseteq, \subseteq\}$. Therefore and since CP is a concept circumscription pattern, $\mathcal{I}' <_{\mathsf{CP}} \mathcal{I}$ follows from $\mathcal{J}' <_{\mathsf{CP}} \mathcal{J}$. We have derived a contradiction and conclude that $\mathcal{J}$ is a model of $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$. Thus we are done since the size of $\mathcal{J}$ is clearly bounded by $2^{2n} \times (m+1) \times n$. ❑

It is interesting to note that the proof of Lemma 5 does not go through if role names are minmized or fixed.

Using the bounded model property just established, we can now prove decidability of reasoning in $\mathcal{ALCIO}$ and $\mathcal{ALCQO}$ with concept circumscription. More precisely, Lemma 5 suggests a non-deterministic decision procedure for satisfiability w.r.t. concept circumscription patterns: simply guess an interpretation of bounded size and then check whether it is a model. It turns out that this procedure shows containment of satisfiability in the complexity class NExp^{NP}, which contains those problems that can be solved by a non-deterministic exponentially time-bounded Turing machine that has access to an NP oracle. It is known that $\text{NExp} \subseteq \text{NExp}^{\text{NP}} \subseteq$ 2-Exp.

Theorem 6 *In $\mathcal{ALCIO}$ and $\mathcal{ALCQO}$, it is in* NExp^{NP} *to decide whether a concept is satisfiable w.r.t. a concept-circumscribed KB* $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$.

Proof. It is not hard to see that there exists an NP algorithm that takes as input a cKB $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$ and an interpretation $\mathcal{I}$, and checks whether $\mathcal{I}$ is *not* a model of $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$: the algorithm first verifies in polynomial time whether $\mathcal{I}$ is a model of $\mathcal{T}$ and $\mathcal{A}$, answering "yes" if this is not the case. Otherwise, the algorithm guesses an interpretation $\mathcal{J}$ that has the same domain as $\mathcal{I}$ and interpretes all object names in the same way, and then checks whether (i) $\mathcal{J}$ is a model of $\mathcal{T}$ and $\mathcal{A}$, and (ii) $\mathcal{J} <_{\mathsf{CP}} \mathcal{I}$.

It answers "yes" if both of the checks succeed, and "no" otherwise. Clearly, checking whether $\mathcal{J} <_{\mathsf{CP}} \mathcal{I}$ can be done in time polynomial w.r.t. the size of $\mathcal{J}$ and $\mathcal{I}$.

This NP algorithm may now be used as an oracle in a NExp-algorithm for deciding satisfiability of a concept C_0 w.r.t. a cKB $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$: by Lemma 5, it suffices to guess an interpretation of size 2^{4k} with $k = |C_0| + |\mathcal{T}| + |\mathcal{A}|$, [1] and then use the NP algorithm to check whether $\mathcal{I}$ is a model of $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$. This proves that concept satisfiability is in NExp^{NP}. ❏

By the reductions given in Section 2, Theorem 6 yields co-NExp^{NP} upper bounds for subsumption and the instance problem. We will show in Section 4 that these upper bounds are tight. However, since NExp^{NP} is a relatively large complexity class, it is a natural question whether we can impose restrictions on concept circumscription such that reasoning becomes simpler. In the following, we identify such a case by considering cKBs in which the number of minimized and fixed concept names is bounded by some constant. In this case, the complexity of satisfiability w.r.t. concept-circumscribed KBs drops to NP^{NExp}. For readers uninitiated to oracle complexity classes, we note that $\text{NExp} \subseteq \text{NP}^{\text{NExp}} \subseteq \text{NExp}^{\text{NP}} \subseteq$ 2-Exp, and that NP^{NExp} is believed to be much less powerful than NExp^{NP}, see for example [13].

To prove the NP^{NExp} upper bound, we first introduce counting formulas as a common generalization of TBoxes and ABoxes.

Definition 7 (Counting Formula) *A* counting formula ϕ *is a Boolean combination of GCIs, ABox assertions $C(a)$, and* cardinality assertions

$$(C = n) \textit{ and } (C \leq n),$$

where C is a concept and n a non-negative integer. We use $\wedge$, $\vee$, $\neg$ and $\rightarrow$ to denote the Boolean operators of counting formulas. An interpretation $\mathcal{I}$ satisfies *a cardinality assertion $(C = n)$ if $\#C^{\mathcal{I}} = n$, and $(C < n)$ if $\#C^{\mathcal{I}} < n$. The satisfaction relation $\mathcal{I} \models \phi$ between models $\mathcal{I}$ and counting formulas ϕ is defined in the obvious way.* △

In the following, we assume that the integers occurring in cardinality assertions are coded in binary. The NP^{NExp} algorithm to be devised will use an algorithm for satisfiability of (non-circumscribed) counting formulas as an oracle. Therefore, we should first determine the computational complexity of the latter. It follows from [35] that, in $\mathcal{ALC}$, satisfiability of counting formulas is NExp-hard. A matching upper bound for the DLs $\mathcal{ALCIO}$ and $\mathcal{ALCQO}$ is obtained from the facts that (i) there is a polynomial translation of counting formulas formulated in these languages into C2, the two-variable fragment of first-order logic extended with counting quantifiers [14, 24], and (ii) satisfiability in C2 is in NExp even if the numbers in counting quantifiers are coded in binary [26].

Theorem 8 (Tobies, Pratt) *In $\mathcal{ALC}$, $\mathcal{ALCIO}$ and $\mathcal{ALCQO}$, satisfiability of counting formulas is* NExp*-complete even if numbers in number restrictions are coded in binary.*

[1]The bound 2^{4k} clearly dominates the two bounds given in Parts (i) and (ii) of Lemma 5.

We now establish the improved upper bound.

Theorem 9 *Let n be a constant. In $\mathcal{ALCIO}$ and $\mathcal{ALCQO}$, it is in* $\mathrm{NP}^{\mathrm{NExp}}$ *to decide satisfiability w.r.t. concept-circumscribed KBs* $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A})$*, where* $\mathsf{CP} = (\prec, M, F, V)$ *is such that* $|M| \leq n$ *and* $|F| \leq n$.

Proof. Assume that we want to decide satisfiability of the concept C_0 w.r.t. the cKB $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A})$, where $\mathsf{CP} = (\prec, M, F, V)$ with $|M| \leq n$ and $|F| \leq n$. By Lemma 4, we may assume that $F = \emptyset$ (we may have to increase the constant n appropriately). We may assume w.l.o.g. that the cardinality of M is exactly n. Thus, let $M = \{A_0, \ldots, A_n\}$. By Lemma 5, C_0 is satisfiable w.r.t. $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A})$ iff there exists a model of C_0 and $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T},\mathcal{A})$ of size 2^{4k}, with $k = |C_0| + |\mathcal{T}| + |\mathcal{A}|$. Consider, for all $S \subseteq M$, the concept

$$C_S := \bigsqcap_{A \in S} A \sqcap \bigsqcap_{A \in \{A_1,\ldots,A_n\} \setminus S} \neg A.$$

As n is fixed, the number 2^n of such concepts is fixed as well. Clearly, the sets $C_S^{\mathcal{I}}$, $S \subseteq M$, form a partition of the domain $\Delta^{\mathcal{I}}$ of any model $\mathcal{I}$. Introduce, for each concept name B and role name r in $\mathcal{T} \cup \mathcal{A}$, a fresh concept name B' and a fresh role name r', respectively. For a concept C, denote by C' the result of replacing in C each concept name B and role name r with B' and r', respectively. The primed versions $\mathcal{A}'$ and $\mathcal{T}'$ of $\mathcal{A}$ and $\mathcal{T}$ are defined analogously. Denote by N the set of individual names in $\mathcal{T} \cup \mathcal{A} \cup \{C_0\}$.

The NExp-oracle we are going to use in our algorithm checks whether a counting formula ϕ is satisfiable or not. Now, the $\mathrm{NP}^{\mathrm{NExp}}$-algorithm is as follows (we use $C \sqsubset D$ as an abbreviation for the counting formula $(C \sqsubseteq D) \wedge \neg(D \sqsubseteq C)$):

1. Guess
   - a sequence $(n_S \mid S \subseteq M)$ of numbers $n_S \leq 2^{4k}$ coded in binary;
   - for each individual name $a \in N$, exactly one set $S_a \subseteq M$;
   - a subset E of $N \times N$.
2. By calling the oracle, check whether the counting formula ϕ_1 is satisfiable, where ϕ_1 is the conjunction over
   - $\mathcal{T} \cup \mathcal{A} \cup \{\neg(C_0 = 0)\}$;
   - $(C_S = n_S)$, for all $S \subseteq M$;
   - $C_{S_a}(a)$, for each $a \in N$;
   - $\{(\{a\} \sqsubseteq \{b\}) \mid (a,b) \in E\} \cup \{\neg(\{a\} \sqsubseteq \{b\}) \mid (a,b) \in N - E\}$.
3. By calling the oracle, check whether the counting formula ϕ_2 is satisfiable, where ϕ_2 is the conjunction over
   - $\mathcal{T}' \cup \mathcal{A}'$;
   - $(C_S = n_S)$, for all $S \subseteq M$ (note that we use the unprimed versions);
   - $C_{S_a}(a)$, for each individual name $a \in N$ (we use the unprimed versions);

- $\{(\{a\} \sqsubseteq \{b\}) \mid (a,b) \in E\} \cup \{\neg(\{a\} \sqsubseteq \{b\}) \mid (a,b) \in N - E\}$;
- for all $A \in M$,
$$\neg(A' \sqsubseteq A) \rightarrow \bigvee_{B \in M, B \prec A} (B' \sqsubset B);$$
- and, finally,
$$\bigvee_{A \in M} ((A' \sqsubset A) \wedge \bigwedge_{B \in M, B \prec A} (B = B')).$$

4. The algorithm states that C_0 is satisfiable in a model of $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$ if, and only if, ϕ_1 is satisfiable and ϕ_2 is not satisfiable.

Using the condition that n is fixed, is is clear that this is a $\mathrm{NP}^{\mathrm{NExp}}$-algorithm. It remains to show correctness and completeness.

Suppose that there exists a model of $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$ satisfying C_0. Then there is such a model $\mathcal{I}$ of size bounded by 2^{4k}. Let the algorithm guess

- the numbers $n_S = |C_S^{\mathcal{I}}|$, $S \subseteq M$,
- the sets S_a such that $a^{\mathcal{I}} \in C_{S_a}^{\mathcal{I}}$,
- the set $E = \{(a,b),(b,a) \mid a^{\mathcal{I}} = b^{\mathcal{I}}, a, b \in N\}$.

Clearly, ϕ_1 is satisfied in $\mathcal{I}$. It remains to show that ϕ_2 is unsatisfiable. But suppose there exists a model $\mathcal{J}$ satisfying ϕ_2. By the conjuncts under Item 2, 3, and 4 of the definitions of ϕ_1 and ϕ_2, we may assume that

- $\Delta^{\mathcal{I}} = \Delta^{\mathcal{J}}$;
- $A^{\mathcal{I}} = A^{\mathcal{J}}$ for all $A \in M$;
- $a^{\mathcal{I}} = a^{\mathcal{J}}$ for all individual names a.

Moreover, as no unprimed role names occur in ϕ_2 and the only unprimed concept names in ϕ_2 are those in M, we may assume that the interpretation of all unprimed concept and role names in $\mathcal{I}$ and $\mathcal{J}$ coincide. Thus, $\mathcal{J}$ is a model of $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$ satisfying C_0. But now define a model $\mathcal{J}'$ with domain $\Delta^{\mathcal{J}}$ by setting

- $a^{\mathcal{J}'} = a^{\mathcal{J}}$, for all individual names a;
- $r^{\mathcal{J}'} = (r')^{\mathcal{J}}$, for all role names r;
- $A^{\mathcal{J}'} = (A')^{\mathcal{J}}$, for all concept names A.

Then, by the conjunct under Item 1 of the definition of ϕ_2, $\mathcal{J}'$ is a model for $\mathcal{A} \cup \mathcal{T}$. By Items 5 and 6 of the definition of ϕ_2, $\mathcal{J}' <_{\mathsf{CP}} \mathcal{J}$, and we have derived a contradiction.

Conversely, suppose the algorithm says that there exists a model of $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$ satisfying C_0. Then take a model $\mathcal{I}$ for ϕ_1. By the conjunct under Item 1 of ϕ_1, $\mathcal{I}$ is a model for $\mathcal{T} \cup \mathcal{A}$ satisfying C_0. It follows from the unsatisfiability of ϕ_2 that $\mathcal{I}$ is a model for $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$. ❑

As an immediate corollary, we obtain co-$\mathrm{NP}^{\mathrm{NExp}}$ upper bounds for subsumption and the instance problem. A similar drop of complexity occurs in propositional logic, where satisfiability w.r.t. circumscribed theories is complete for $\mathrm{NP}^{\mathrm{NP}}$ and it is not difficult to see that bounding the minimized and fixed predicates allows to find a P^{NP} algorithm. To the best of our knowledge, this has never been explicitly observed before.

4 Lower Bounds

We show that the upper bounds given in Section 3 are tight. As usual, the lower bounds are established by reduction of a suitable problem that is complete for the complexity class under consideration. Thus, we are given an input x of the chosen problem, construct a cKB and a concept from x, and show that the concept is satisfiable w.r.t. the cKB iff x is a yes-instance of the problem. To achieve a gentle presentation of the reductions, it is convenient to split up the constructed cKB into independent parts. We first establish a general lemma facilitating such a splitting. A concept C is *simultaneously satisfiable w.r.t. cKBs* $\mathsf{Circ}_{\mathsf{CP}_1}(\mathcal{T}_1, \mathcal{A}_1), \dots, \mathsf{Circ}_{\mathsf{CP}_k}(\mathcal{T}_k, \mathcal{A}_k)$ if there exists an interpretation $\mathcal{I}$ that is a model of all the cKBs and satisfies $C^{\mathcal{I}} \neq \emptyset$. The following lemma says that simultaneous satisfiability coincides with separate satisfiability if there are no shared role names in the two cKBs.

Lemma 10 *Let* $\mathsf{Circ}_{\mathsf{CP}_1}(\mathcal{T}_1, \mathcal{A}_1), \dots \mathsf{Circ}_{\mathsf{CP}_k}(\mathcal{T}_k, \mathcal{A}_k)$ *be concept-circumscribed cKBs formulated in* $\mathcal{ALC}$ *such that* $\mathsf{Circ}_{\mathsf{CP}_i}(\mathcal{T}_i, \mathcal{A}_i)$ *and* $\mathsf{Circ}_{\mathsf{CP}_j}(\mathcal{T}_j, \mathcal{A}_j)$ *have no shared role names, for all* $1 \leq i < j \leq k$*. Then, simultaneous satisfiability w.r.t.* $\mathsf{Circ}_{\mathsf{CP}_1}(\mathcal{T}_1, \mathcal{A}_1), \dots \mathsf{Circ}_{\mathsf{CP}_k}(\mathcal{T}_k, \mathcal{A}_k)$*, can be polynomially reduced to satisfiability w.r.t. a single concept-circumscribed KB* $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$ *such that the cardinality of each component of* CP *is the sum of cardinalities of the corresponding components of* $\mathsf{CP}_1, \dots, \mathsf{CP}_k$*.*

Proof. We only give a proof for the case $k = 2$. A generalization is straightforward. Let C be an $\mathcal{ALC}$ concept and let $\mathsf{Circ}_{\mathsf{CP}_1}(\mathcal{T}_1, \mathcal{A}_1)$, $\mathsf{Circ}_{\mathsf{CP}_2}(\mathcal{T}_2, \mathcal{A}_2)$ be two concept-circumscribed KBs formulated in $\mathcal{ALC}$ that have no shared role names. Moreover, let $A_0, \dots, A_{k-1}$ be the concept names used in both cKBs, $\mathcal{R}$ the role names used in the two cKBs, and $\mathsf{CP}_i = (\prec_i, M_i, F_i, V_i)$ for $i \in \{1, 2\}$. We obtain a new TBox $\mathcal{T}_2'$ from $\mathcal{T}_2$ by replacing each concept name A_i, $i < k$, with a new concept name A_i'. Let $\mathcal{A}_2'$ be obtained from $\mathcal{A}_2$ and $\mathsf{CP}_2' = (\prec_2', M_2', F_2', V_2')$ from CP_2 in an analogous way. Define a TBox $\mathcal{T}^*$ as follows, where P is a new concept name:

$$\begin{aligned} A_i \sqcap \neg A_i' &\sqsubseteq P \text{ for all } i < k \\ \neg A_i \sqcap A_i' &\sqsubseteq P \text{ for all } i < k \\ P &\sqsubseteq \forall r.P \text{ for all } r \in \mathcal{R} \\ \exists r.P &\sqsubseteq P \text{ for all } r \in \mathcal{R} \end{aligned}$$

Now set:

$$\begin{array}{rcl} \mathcal{T} & := & \mathcal{T}_1 \cup \mathcal{T}_2' \cup \mathcal{T}^* \\ \mathcal{A} & := & \mathcal{A}_1 \cup \mathcal{A}_2' \\ \prec & := & \prec_1 \cup \prec_2' \\ M & := & M_1 \cup M_2' \\ F & := & F_1 \cup F_2' \\ V & := & V_1 \cup V_2' \\ \mathsf{CP} & := & (\prec, M, F, V) \end{array}$$

It remains to show the following:

Claim. C is simultaneously satisfiable w.r.t. $\mathsf{Circ}_{\mathsf{CP}_1}(\mathcal{T}_1, \mathcal{A}_1)$ and $\mathsf{Circ}_{\mathsf{CP}_2}(\mathcal{T}_2, \mathcal{A}_2)$ iff $C \sqcap \neg P$ is satisfiable w.r.t. $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$.

"if". Assume that $C \sqcap \neg P$ is satisfiable w.r.t. $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$, and let $\mathcal{I}$ be a model witnessing this. We may w.l.o.g. assume that $\mathcal{I}$ is connected. By construction of $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$, $\mathcal{I}$ is a model of C, $\mathcal{T}_1$, and $\mathcal{A}_1$. To show that C is satisfiable w.r.t. $\mathsf{Circ}_{\mathsf{CP}_1}(\mathcal{T}_1, \mathcal{A}_1)$, we prove that $\mathcal{I}$ is a model of $\mathsf{Circ}_{\mathsf{CP}_1}(\mathcal{T}_1, \mathcal{A}_1)$. Assume to the contrary that this is not the case. Then there exists a model $\mathcal{J}$ of $\mathcal{T}_1$ and $\mathcal{A}_1$ such that $\mathcal{J} <_{\mathsf{CP}_1} \mathcal{I}$. Define a model $\mathcal{J}'$ as follows:

- $\Delta^{\mathcal{J}'} = \Delta^{\mathcal{J}}$;
- all predicates used in $\mathcal{T}_1$ and $\mathcal{A}_1$ are interpreted as in $\mathcal{J}$;
- all predicates used in $\mathcal{T}_2'$ and $\mathcal{A}_2'$ are interpreted as in $\mathcal{I}$.
- $P^{\mathcal{J}} := \begin{cases} \Delta^{\mathcal{I}} & \text{if } ((A_i \sqcap \neg A_i') \sqcup (\neg A_i \sqcap A_i'))^{\mathcal{J}} \neq \emptyset \text{ for some } i < k \\ \emptyset & \text{otherwise.} \end{cases}$

It is readily checked that $\mathcal{J}'$ is a model of $\mathcal{T}$ and $\mathcal{A}$, and that $\mathcal{J}' <_{\mathsf{CP}} \mathcal{I}$. Thus, we have derived a contradiction to the fact that $\mathcal{I}$ is a model of $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$.

Since $\mathcal{I}$ is connected and satisfies $\neg P$ and $\mathcal{T}^*$, we have that $A_i^{\mathcal{I}} = (A_i')^{\mathcal{I}}$ for all $i < k$. Therefore, $\mathcal{I}$ is also a model of $\mathcal{T}_2$ and $\mathcal{A}_2$. It remains to be shown that $\mathcal{I}$ is a model of $\mathsf{Circ}_{\mathsf{CP}_2}(\mathcal{T}_2, \mathcal{A}_2)$, which can be done analogously to the case of $\mathsf{Circ}_{\mathsf{CP}_1}(\mathcal{T}_1, \mathcal{A}_1)$.

"only if". Assume that C is simultaneously satisfiable w.r.t. $\mathsf{Circ}_{\mathsf{CP}_1}(\mathcal{T}_1, \mathcal{A}_1)$ and $\mathsf{Circ}_{\mathsf{CP}_2}(\mathcal{T}_2, \mathcal{A}_2)$. Then there exists a model $\mathcal{I}$ of C that is a model of $\mathsf{Circ}_{\mathsf{CP}_1}(\mathcal{T}_1, \mathcal{A}_1)$ and $\mathsf{Circ}_{\mathsf{CP}_2}(\mathcal{T}_2, \mathcal{A}_2)$. We modify $\mathcal{I}$ to a new model $\mathcal{I}'$ by setting

- $(A_i')^{\mathcal{I}'} := A_i^{\mathcal{I}}$ for all $i < k$;
- $P^{\mathcal{I}'} := \emptyset$.

It should be clear that $\mathcal{I}'$ is a model of $C \sqcap \neg P$, $\mathcal{T}$, and $\mathcal{A}$. It remains to show that $\mathcal{I}'$ is also model of $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$. To do this, we first show the following:

(a) $\mathcal{I}'$ is a model of $\mathsf{Circ}_{\mathsf{CP}_1}(\mathcal{T}_1, \mathcal{A}_1)$. This is the case since any model $\mathcal{J}$ of $\mathcal{T}_1$ and $\mathcal{A}_1$ with $\mathcal{J} <_{\mathsf{CP}_1} \mathcal{I}'$ satisfies $\mathcal{J} <_{\mathsf{CP}_1} \mathcal{I}$. Thus, the existence of such a model contradicts the fact that $\mathcal{I}$ is a model of $\mathsf{Circ}_{\mathsf{CP}_1}(\mathcal{T}_1, \mathcal{A}_1)$.

(b) $\mathcal{I}'$ is a model of $\mathsf{Circ}_{\mathsf{CP}'_2}(\mathcal{T}'_2, \mathcal{A}'_2)$. Assume to the contrary that there is a model $\mathcal{J}$ of $\mathcal{T}'_2$ and $\mathcal{A}'_2$ with $\mathcal{J} <_{\mathsf{CP}'_2} \mathcal{I}'$. Convert $\mathcal{J}$ into an interpretation $\mathcal{J}^*$ by setting $A_i^{\mathcal{J}^*} := (A'_i)^{\mathcal{J}}$ for all $i < k$. Then, $\mathcal{J}^*$ is a model of $\mathcal{T}_2$ and $\mathcal{A}_2$ and satisfies $\mathcal{J}^* <_{\mathsf{CP}_2} \mathcal{I}$. This is a contradiction to the fact that $\mathcal{I}$ is a model of $\mathsf{Circ}_{\mathsf{CP}_2}(\mathcal{T}_2, \mathcal{A}_2)$.

Now, assume to the contrary of what remains to be shown that there is a model $\mathcal{J}'$ of $\mathcal{T}$ and $\mathcal{A}$ with $\mathcal{J}' <_{\mathsf{CP}} \mathcal{I}'$. By definition of CP, $\mathcal{J}' <_{\mathsf{CP}} \mathcal{I}'$ implies that we have $\mathcal{J}' <_{\mathsf{CP}_1} \mathcal{I}'$ or $\mathcal{J}' <_{\mathsf{CP}'_2} \mathcal{I}'$. Since $\mathcal{J}'$ clearly satisfies $\mathcal{T}_1$, $\mathcal{A}_1$, $\mathcal{T}'_2$, and $\mathcal{A}'_2$, we obtain a contradiction to (a) and (b). ❑

We start our study of lower complexity bounds by proving a matching lower bound for Theorem 6: we show that, in $\mathcal{ALC}$, satisfiability w.r.t. concept-circumscribed KBs is NExp^{NP}-hard. Therefore, satisfiability w.r.t. concept-circumscribed KBs is NExp^{NP}-complete in $\mathcal{ALC}$, $\mathcal{ALCIO}$, and $\mathcal{ALCQO}$. The proof is by reduction of a succinct version of the problem co-CERT3COL [12]. Let us first introduce the regular (non-succinct) version of this problem:

Instance of size n: an undirected graph G on the vertices $\{0, 1, \ldots, n-1\}$ such that every edge is labelled with a disjunction of two literals over the Boolean variables $\{V_{i,j} \mid i, j < n\}$.

Yes-Instance of size n: an instance G of size n such that, for some truth value assignment t to the Boolean variables, the graph $t(G)$ obtained from G by including only those edges whose label evaluates to true under t is not 3-colorable.

As shown in [33], co-CERT3COL is complete for NP^{NP}. To obtain a problem complete for NExp^{NP}, Eiter et al. use the complexity upgrade technique: by encoding the input in a succinct form using Boolean circuits, the complexity is raised by one exponential to NExp^{NP} [12]. More precisely, the succinct version co-CERT3COL$_S$ of co-CERT3COL is obtained by representing the input graph G with nodes $\{0, \ldots, 2^n - 1\}$ as $4n + 3$ Boolean circuits with $2n$ inputs (and one output) each. The Boolean circuits are named c_E, $c_S^{(1)}$, $c_S^{(2)}$, and $c_j^{(i)}$, with $i \in \{1, 2, 3, 4\}$ and $j < n$. For all circuits, the $2n$ inputs are the bits of the binary representation of two nodes of the graph. The purpose of the circuits is as follows:

- circuit c_E outputs 1 if there is an edge between the two input nodes, and 0 otherwise;

- if there is an edge between the input nodes, circuit $c_S^{(1)}$ outputs 1 if the first literal in the disjunction labelling this edge is positive, and 0 otherwise; the circuit $c_S^{(2)}$ does the same for the second literal;

- if there is an edge between the input nodes, the circuits $c_j^{(i)}$ compute the labelling $V_{k_1,k_2} \vee V_{k_3,k_4}$ of this edge between the input nodes by generating the numbers $k_1, \ldots, k_4$: the circuit $c_j^{(i)}$ outputs the j-th bit of k_i.

Now for the reduction of co-CERT3COL$_S$ to satisfiability of concept-circumscribed KBs. Let

$$G = (n, c_E, c_S^{(1)}, c_S^{(2)}, \{c_j^{(i)}\}_{i\in\{1,..,4\},j<n})$$

be the (succinct representation of the) input graph with 2^n nodes. We will construct two TBoxes $\mathcal{T}_G$ and $\mathcal{T}'_G$, circumscription patterns CP and CP', and a concept C_G such that C_G is simultaneously satisfiable w.r.t. $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}_G, \emptyset)$ and $\mathsf{Circ}_{\mathsf{CP}'}(\mathcal{T}'_G, \emptyset)$ iff G is a yes-instance of co-CERT3COL$_S$. By Lemma 10, we then obtain a reduction to (non-simultaneous) satisfiability w.r.t. concept-circumscribed cKBs. Intuitively, the purpose of the first TBox $\mathcal{T}_G$ is to fix a truth assignment t for the variables $\{V_{i,j} \mid i,j < n\}$, and to construct (an isomorphic image of) the graph $t(G)$ obtained from G by including only those edges whose label evaluates to true under t. Then, the purpose of $\mathcal{T}'_G$ is to make sure that $t(G)$ is not 3-colorable.

When formulating the reduction TBoxes, we use several binary counters for counting modulo 2^n (the number of nodes in the input graph). The main counters X and Y use concept names $X_0, \dots, X_{n-1}$ and $Y_0, \dots, Y_{n-1}$ as their bits, respectively. Additionally, we introduce concept names $K_0^{(i)}, \dots, K_{n-1}^{(i)}$, $i \in \{1,2,3,4\}$ that binarily encode numbers from the range $0, \dots, 2^n - 1$, but are never incremented as a counter. The main part of the TBox $\mathcal{T}_G$ can be found in Figure 2, where the following abbreviations are used: first, $\forall r.(K^{(i)} = X)$ is a concept expressing that, for all its instances x, the values of $X_0, \dots, X_{n-1}$ at all r-successors agree with the values of $K_0^{(i)}, \dots, K_{n-1}^{(i)}$ at x. And second, $\forall r.(X{+}{+})$ is an abbreviation for the well-known concept stating that the value of the counter $X_0, \dots, X_{n-1}$ is incremented when going to r-successors:

$$\bigsqcap_{k=0..n-1} \big(\bigsqcap_{j=0..k-1} X_j \big) \rightarrow \big((X_k \rightarrow \forall r.\neg X_k) \sqcap (\neg X_k \rightarrow \forall r.X_k) \big)$$

$$\bigsqcap_{k=0..n-1} \big(\bigsqcup_{j=0..k-1} \neg X_j \big) \rightarrow \big((X_k \rightarrow \forall r.X_k) \sqcap (\neg X_k \rightarrow \forall r.\neg X_k) \big)$$

The intuitions of $\mathcal{T}_G$ are as follows: Lines (1) to (3) ensure that, for each possible value of the counters X and Y, there is at least one domain element in $\mathsf{Val}^{\mathcal{I}}$ with this counter value. We will minimize Val to ensure that there is *exactly* one domain element in $\mathsf{Val}^{\mathcal{I}}$ for each possible value i of X and j of Y. Intuitively, these domain elements are used to store informaion about the variables V_{ij} and the (potential) edges (i,j). Concerning the variables, each element of $\mathsf{Val}^{\mathcal{I}}$ with $X = i$ and $Y = j$ corresponds to the variable $V_{i,j}$ of co-3CERTCOL$_S$ and determines a truth value for this variable via the concept name Tr. Thus, the elements of $\mathsf{Val}^{\mathcal{I}}$ jointly describe a truth assignment for the variables of co-3CERTCOL$_S$. Line (4) introduces Edge as another name for Val. We do this to distinguish the use of the elements of Val as variables and as edges. Intuitively, an element of $d \in \mathsf{Edge}^{\mathcal{I}}$ with $X = i$ and $Y = j$ corresponds to the (potential) edge between the nodes i and j To explain this more properly, we must first discuss the part of $\mathcal{T}_G$ that is missing in Figure 2.

It is easily seen that each Boolean circuit c with $2n$ inputs can be converted into a TBox $\mathcal{T}_c$ in the following sense: if the output of c upon input $b_0, \dots, b_{2n-1}$ is b, then, for all models $\mathcal{I}$ of $\mathcal{T}_c$ and all domain elements $x \in \Delta^{\mathcal{I}}$ such that the truth value of the concept names $X_0, \dots, X_{n-1}, Y_0, \dots, Y_{n-1}$ at x is described by $b_0, \dots, b_{n-1}$, the truth

$$
\begin{array}{rcll}
\top & \sqsubseteq & \exists \mathsf{aux}.\mathsf{Val} & (1)\\
\mathsf{Val} & \sqsubseteq & \exists \mathsf{nextx}.\top \sqcap \forall \mathsf{nextx}.\mathsf{Val} \sqcap \forall \mathsf{nextx}.(X{+}{+}) \sqcap \forall \mathsf{nextx}.(Y{=}Y) & (2)\\
\mathsf{Val} & \sqsubseteq & \exists \mathsf{nexty}.\top \sqcap \forall \mathsf{nexty}.\mathsf{Val} \sqcap \forall \mathsf{nexty}.(Y{+}{+}) \sqcap \forall \mathsf{nexty}.(X{=}X) & (3)\\
\\
\mathsf{Edge} & \doteq & \mathsf{Val} & (4)\\
\mathsf{Edge} & \sqsubseteq & \exists \mathsf{var1}.\top \sqcap \forall \mathsf{var1}.\mathsf{Val} \sqcap \forall \mathsf{var1}.(K^{(1)}{=}X) \sqcap \forall \mathsf{var1}.(K^{(2)}{=}Y) & (5)\\
\mathsf{Edge} & \sqsubseteq & \exists \mathsf{var2}.\top \sqcap \forall \mathsf{var2}.\mathsf{Val} \sqcap \forall \mathsf{var2}.(K^{(3)}{=}X) \sqcap \forall \mathsf{var2}.(K^{(4)}{=}Y) & (6)\\
\\
\mathsf{Edge} & \sqsubseteq & S_1 \rightarrow (\mathsf{Tr}_1 \leftrightarrow \forall \mathsf{var1}.\mathsf{Tr}) & (7)\\
\mathsf{Edge} & \sqsubseteq & \neg S_1 \rightarrow (\neg \mathsf{Tr}_1 \leftrightarrow \forall \mathsf{var1}.\mathsf{Tr}) & (8)\\
\mathsf{Edge} & \sqsubseteq & S_2 \rightarrow (\mathsf{Tr}_2 \leftrightarrow \forall \mathsf{var2}.\mathsf{Tr}) & (9)\\
\mathsf{Edge} & \sqsubseteq & \neg S_2 \rightarrow (\neg \mathsf{Tr}_2 \leftrightarrow \forall \mathsf{var1}.\mathsf{Tr}) & (10)\\
\mathsf{Edge} & \sqsubseteq & \mathsf{Elim} \leftrightarrow (\neg E \sqcup \neg(\mathsf{Tr}_1 \sqcup \mathsf{Tr}_2)) & (11)
\end{array}
$$

Figure 2: The TBox $\mathcal{T}_G$ (partly).

value of some concept name Out at x is described by b. By introducing one auxiliary concept name for every inner gate of c, the translation can be done such that the size of $\mathcal{T}_c$ is linear in the size of c. Now, the part of $\mathcal{T}_G$ not shown in Figure 2 is obtained by converting the Boolean circuits describing the graph G into a TBox in the described way. More precisely, this is done such that the following concept names are used as output:

- the translation of c_E uses the concept name E as output;
- the translation of $c_S^{(i)}$ uses the concept name S_i as output, for $i \in \{1, 2\}$;
- the translation of $c_j^{(i)}$ uses the concept name $K_j^{(i)}$ as output, for $i \in \{1, \ldots, 4\}$ and $j < n$.

Note that the evaluation of Boolean circuits takes place locally at every domain element. In principle, it suffices to evaluate the circuits only at instances of Edge: there, $X_0, \ldots, X_{n-1}$ describe the left-hand node of the corresponding edge, and $Y_0, \ldots, Y_{n-1}$ describe the right-hand node of the corresponding edge.

With this in mind, it is easy to see that Line (5) ensures the following: each element $d \in \mathsf{Edge}^{\mathcal{I}}$ representing an edge (i, j) is connected via the role var1 to the element of $\mathsf{Val}^{\mathcal{I}}$ that represents the variable in the first disjunct of the label of (i, j). Line (6) is analogous for the role var2 and the variable in the second disjunct of the edge label. Then, Lines (7) to (11) ensure that $d \in \mathsf{Edge}^{\mathcal{I}}$ is an instance of Elim iff the edge corresponding to d is not present in the graph $t(G)$ induced by the truth assignment t described by Val.

The TBox $\mathcal{T}'_G$ can be found in Figure 3. Here, $(X = i)$ stands for the concepts expressing that $X_0, \ldots, X_{n-1}$ are the binary encoding of the number i. As already said,

$$
\begin{array}{rcll}
\mathsf{Node} & \doteq & \mathsf{Val} \sqcap (Y = 0) & (12)\\
\mathsf{Node} & \sqsubseteq & R \sqcup B \sqcup G & (13)\\
\mathsf{Node} & \sqsubseteq & \neg(R \sqcap B) \sqcap \neg(R \sqcap G) \sqcap \neg(B \sqcap G) & (14)\\
\mathsf{Edge} & \sqsubseteq & \exists \mathsf{col1}.\top \sqcap \forall \mathsf{col1}.\mathsf{Node} \sqcap \forall \mathsf{col1}.(X{=}X) & (15)\\
\mathsf{Edge} & \sqsubseteq & \exists \mathsf{col2}.\top \sqcap \forall \mathsf{col2}.\mathsf{Node} \sqcap \forall \mathsf{col2}.(Y{=}X) & (16)\\
P & \sqsupseteq & \mathsf{Edge} \sqcap \neg \mathsf{Elim} \sqcap \exists \mathsf{col1}.R \sqcap \exists \mathsf{col2}.R & (17)\\
P & \sqsupseteq & \mathsf{Edge} \sqcap \neg \mathsf{Elim} \sqcap \exists \mathsf{col1}.G \sqcap \exists \mathsf{col2}.G & (18)\\
P & \sqsupseteq & \mathsf{Edge} \sqcap \neg \mathsf{Elim} \sqcap \exists \mathsf{col1}.B \sqcap \exists \mathsf{col2}.B & (19)
\end{array}
$$

Figure 3: The TBox $\mathcal{T}'_G$.

the purpose of $\mathcal{T}'_G$ is to ensure that the graph $t(G)$ induced by the truth assignment t described by Val does not have a 3-coloring. The strategy for ensuring this is as follows: we use the 2^n elements of $(\mathsf{Val} \sqcap (Y = 0))^{\mathcal{I}}$ to store the colors of the nodes. By Line (12), these elements are identified by the concept name Node, and there is a unique coloring due to Lines (13) and (14). Then, Line (15) ensures that each element $d \in \mathsf{Edge}^{\mathcal{I}}$ is connected via the role col1 to the element of $\mathsf{Node}^{\mathcal{I}}$ storing the color of the first node of the edge corresponding to d. Line (16) is analogous for the role col2 and the second node of the edge. Lines (17) to (19) guarantee that instances of Edge corresponding to problematic edges are instances of the concept name P. Here, an edge is problematic if it exists in the original graph, is not dropped by the current truth assignment, and the connected nodes have the same color. The idea is that P will be minimized with all concept names fixed except R, G, and B. Then, we have $P^{\mathcal{I}}$ non-empty iff there is no 3-coloring of $t(G)$. Please observe that fixing all concept names except R, G, B also means that the used roles are fixes on instances of Edge and Val.

Lemma 11 *G is a yes-instance of co-3CERTCOL$_S$ iff P is simultaneously satisfiable w.r.t.* $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}_G, \emptyset)$ *and* $\mathsf{Circ}_{\mathsf{CP}'}(\mathcal{T}'_G, \emptyset)$*, where*

- $\mathsf{CP} = (\prec, M, F, V)$ *with* $\prec = \emptyset$, $M = \{\mathsf{Val}\}$, $F = \emptyset$, *and V all remaining predicates in* $\mathcal{T}_G$*;*
- $\mathsf{CP}' = (\prec', M', F', V')$ *with* $\prec' = \emptyset$, $M' = \{P\}$, $F' = \emptyset$, *and V' the set of all remaining predicates used in* $\mathcal{T}'_G$*.*

Proof. "If". Suppose that P is simultaneously satisfiable w.r.t. $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}_G, \emptyset)$ and $\mathsf{Circ}_{\mathsf{CP}'}(\mathcal{T}'_G, \emptyset)$, and let $\mathcal{I}$ be a model of P and a model of both $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}_G, \emptyset)$ and $\mathsf{Circ}_{\mathsf{CP}'}(\mathcal{T}'_G, \emptyset)$. We have to show that G is a yes-instance of co-CERT3COL$_S$. We first note that, for all $i, j \in \{0, \ldots, n-1\}$, $\mathsf{Val}^{\mathcal{I}}$ contains exactly one element

$$x \in ((X = i) \sqcap (Y = j))^{\mathcal{I}}.$$

The reasons for this are as follows: (i) Lines (1)-(3) force $\mathsf{Val}^\mathcal{I}$ to contain at least one such element for each pair (i, j); (ii) since $\mathcal{I}$ is a model of $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}_G, \emptyset)$ and CP minimizes Val while varying all other predicates, there cannot be more than one such x in $\mathsf{Val}^\mathcal{I}$. In the following, we use x_{ij} to denote the unique element of $(\mathsf{Val} \sqcap (X = i) \sqcap (Y = j))^\mathcal{I}$.

Now suppose, to the contrary of what is to be shown, that G is not a yes-instance. Then, for all truth assignments t, the subgraph $t(G)$ is 3-colorable. In particular this holds for the assignment t defined by setting

$$t(V_{ij}) := \mathit{true} \text{ iff } x_{ij} \in \mathsf{Tr}^\mathcal{I}\,.$$

Let $c : \{0, \ldots, n-1\} \to \{R, G, B\}$ be a 3-coloring of $t(G)$ and construct an interpretation $\mathcal{J}$ as follows:

$$\begin{array}{rcl} \Delta^\mathcal{J} & = & \Delta^\mathcal{I} \\ r^\mathcal{J} & = & r^\mathcal{I} \text{ for all role names} \\ A^\mathcal{J} & = & A^\mathcal{I} \text{ for all concept names except } R, G, \text{ and } B \\ C^\mathcal{J} & = & \{x_{i0} \mid c(i) = C\} \text{ for } C = R, G, B \\ P^\mathcal{J} & = & \emptyset\,. \end{array}$$

Clearly, $\mathcal{J} <_{\mathsf{CP}'} \mathcal{I}$, because the minimized predicate P is non-empty in $\mathcal{I}$ and empty in $\mathcal{J}$. Thus, to obtain a contradiction, it suffices to show that $\mathcal{J}$ is a model of $\mathcal{T}'_G$.

Since $\mathcal{I}$ and $\mathcal{J}$ agree on all predicates but R, G, B, and P, Inclusions (12), (15), and (16) that do not mention these concepts must hold in $\mathcal{J}$. Line (12) implies $\mathsf{Node}^\mathcal{J} = \{x_{i0} \mid 0 \le i < n\}$, and hence $\mathcal{J}$ satisfies (13) by construction. Moreover, since c is a function, (14) is satisfied, too. The following claim is a consequence of the definition of the truth assignment t and the facts that (i) $\mathcal{I}$ is a model of $\mathcal{T}_G$ and (ii) $\mathcal{I}$ and $\mathcal{J}$ interpret the concept names Edge and Elim in the same way.

Claim 1: (i, j) is an edge of $t(G)$ iff $x_{ij} \in (\mathsf{Edge} \sqcap \neg\mathsf{Elim})^\mathcal{J}$.

Now, we prove that (17) to (19) are satisfied in $\mathcal{J}$. Let $C \in \{R, G, B\}$ and $x_{ij} \in (\mathsf{Edge} \sqcap \neg\mathsf{Elim})^\mathcal{J}$. By Claim 1, we get $c(i) \neq c(j)$ since c is a 3-coloring of $t(G)$. Thus, by construction of $\mathcal{J}$, x_{i0} and x_{j0} cannot belong to C together. Moreover, by (15) and (16), $\mathsf{col1}$ and $\mathsf{col2}$ connect x_{ij} precisely to x_{i0} and x_{j0}, respectively. Therefore, $x_{ij} \notin (\exists\mathsf{col1}.C \sqcap \exists\mathsf{col2}.C)^\mathcal{J}$. Since this holds for any $x_{ij} \in (\mathsf{Edge} \sqcap \neg\mathsf{Elim})^\mathcal{J}$, it follows that the right-hand sides of (17) to (19) are empty in $\mathcal{J}$. Thus, these implications are satisfied.

"Only if". Suppose that G is a yes-instance and let t be a truth assignment such that $t(G)$ is not 3-colorable. Let $c : \{0, \ldots, n-1\} \to \{R, G, B\}$ be a color assignment that minimizes (w.r.t. set inclusion) the set $\{(i, j) \mid c(i) = c(j)\}$. Define an interpretation $\mathcal{I}$ as follows:

$$\begin{array}{rcl} \Delta^\mathcal{I} & = & \{(i, j) \mid 0 \le i < 2^n,\ 0 \le j < 2^n\} \\ \mathsf{Val}^\mathcal{I} & = & \mathsf{Edge}^\mathcal{I} = \Delta^\mathcal{I} \\ \mathsf{Tr}^\mathcal{I} & = & \{(i, j) \mid t(V_{ij}) = \mathit{true}\} \end{array}$$

$$
\begin{array}{rcl}
\mathsf{Tr}_i^{\mathcal{I}} & = & \{(i,j) \mid t(V_{ij}) \leftrightarrow c_S^{(i)}(i,j)\} \quad (i = 1,2) \\
\mathsf{Elim}^{\mathcal{I}} & = & \{(i,j) \mid (i,j) \text{ is an edge of } t(G)\} \\
\mathsf{Node}^{\mathcal{I}} & = & \{(i,0) \mid 0 \leq i < 2^n\} \\
C^{\mathcal{I}} & = & \{(i,\emptyset) \mid c(i) = C\} \quad (C = R,G,B) \\
P^{\mathcal{I}} & = & \{(i,j) \mid (i,j) \text{ is an edge of } t(G) \text{ and } c(i) = c(j)\} \\
\mathsf{nextx}^{\mathcal{I}} & = & \{((i,j),(i+1 \bmod 2^n, j)) \mid 0 \leq i,j < 2^n\} \\
\mathsf{nexty}^{\mathcal{I}} & = & \{((i,j),(i,j+1 \bmod 2^n)) \mid 0 \leq i,j < 2^n - 1\} \\
\mathsf{col1}^{\mathcal{I}} & = & \{((i,j),(i,0)) \mid 0 \leq i < 2^n\} \\
\mathsf{col2}^{\mathcal{I}} & = & \{((i,j),(j,0)) \mid 0 \leq i < 2^n\} \\
\mathsf{var1}^{\mathcal{I}} & = & \{((i,j),(k,l)) \mid \text{the first variable in the label of } (i,j) \text{ is } V_{kl}\} \\
\mathsf{var2}^{\mathcal{I}} & = & \{((i,j),(k,l)) \mid \text{the second variable in the label of } (i,j) \text{ is } V_{kl}\}
\end{array}
$$

Moreover, the concept names $X_k^{\mathcal{I}}$ and $Y_k^{\mathcal{I}}$ are interpreted in such a way that $(i,j) \in ((X = i) \sqcap (Y = j))^{\mathcal{I}}$ holds for all $i,j < 2^n$. For each Boolean circuit c the corresponding output concept name $\mathsf{Out}_c^{\mathcal{I}}$ contains precisely those (i,j) such that $c(i,j)$ is *true*.

Since c is not a 3-coloring, P is satisfied in $\mathcal{I}$. Thus, it remains to show that $\mathcal{I}$ is a model of $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}_G, \emptyset)$ and $\mathsf{Circ}_{\mathsf{CP}'}(\mathcal{T}'_G, \emptyset)$. We start with the former. It is straightforward to see that $\mathcal{I}$ is a model of $\mathcal{T}_G$. To see that $\mathcal{I}$ is also a model of $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}_G, \emptyset)$, note that by inclusions (1)–(3), there must be at least one instance of Val in each of the (mutually disjoint) concepts $(X = i) \sqcap (Y = j)$. Since $\mathcal{I}$ has exactly one element for each such concept, the extension of Val is minimal in $\mathcal{I}$.

Now for $\mathsf{Circ}_{\mathsf{CP}'}(\mathcal{T}'_G, \emptyset)$. The reader may easily verify that $\mathcal{I}$ satisfies $\mathcal{T}'_G$ by construction. To prove that $\mathcal{I}$ is also a $<_{\mathsf{CP}'}$-minimal model of $\mathcal{T}'_G$, first note that if there existed a model $\mathcal{J} <_{\mathsf{CP}'} \mathcal{I}$, then $P^{\mathcal{J}} \subset P^{\mathcal{I}}$ would hold. Moreover, the minimization of $P^{\mathcal{J}}$ would make its extension equal to the disjunction of the right-hand sides of (17)–(19). As a consequence, to satisfy (17)–(19), we should have (i) $(\exists \mathsf{col1}.C \sqcap \exists \mathsf{col2}.C)^{\mathcal{J}} \subseteq (\exists \mathsf{col1}.C \sqcap \exists \mathsf{col2}.C)^{\mathcal{I}}$ for $C = R, G, B$, and (ii) for some color C,

$$(\exists \mathsf{col1}.C \sqcap \exists \mathsf{col2}.C)^{\mathcal{J}} \subset (\exists \mathsf{col1}.C \sqcap \exists \mathsf{col2}.C)^{\mathcal{I}}.$$

But then, the coloring c' defined by

$$c'(i) = C \text{ iff } (i,0) \in C^{\mathcal{J}} \quad (C = R,G,B)$$

would be such that

$$\{(i,j) \mid c'(i) = c'(j)\} \subset \{(i,j) \mid c(i) = c(j)\}.$$

This inclusion contradicts the minimality assumption on c. ❑

Since it is easily checked that the size of $\mathcal{T}_G$ and $\mathcal{T}'_G$ is polynomial in n, we get the following result.

Theorem 12 *In $\mathcal{ALC}$, satisfiability w.r.t. concept-circumscribed KBs is* $\mathrm{NExp}^{\mathrm{NP}}$-*hard.*

It is interesting to observe that the reduction works even if we assume ABoxes and preference relations to be empty. Corresponding lower bounds for subsumption and the instance problems follow from the reduction given in Section 2.

We now establish a matching lower bound for Theorem 9: we show that, in $\mathcal{ALC}$, satisfiability w.r.t. concept-circumscribed KBs is NP^{NExp}-hard even if only a constant number of predicates are allowed to be minimized and fixed. Recall that a (non-deterministic) *k-tape Turing machine* is described by a tuple

$$(Q, \Sigma, q_0, \Delta, q_{\mathsf{acc}}, q_{\mathsf{rej}}),$$

with Q a set of states, Σ a finite alphabet, $q_0 \in Q$ a starting state,

$$\Delta \subseteq Q \times \Sigma^k \times Q \times \Sigma^k \times \{L, R\}^k$$

a transition relation, and $q_{\mathsf{acc}}, q_{\mathsf{rej}} \in Q$ the accepting and rejecting states. For our purposes, an *oracle Turing machine* is a 2-tape Turing machine M that is, additionally, equipped with the following:

- a 1-tape Turing machine M' (the *oracle*) whose alphabet contains that of M,
- a query state $q_?$, and
- two answer states q_{yes} and q_{no}.

When M enters $q_?$, the oracle determines the next state of M: if the content of M's second tape is contained in the language accepted by the oracle, the next state is q_{yes}. Otherwise, it is q_{no}. During this transition, the head is not moved and no symbols are written. The state $q_?$ cannot occur as the left-most component of a tuple in M's transition relation.

Let $M = (Q, \Sigma, q_0, \Delta, q_{\mathsf{acc}}, q_{\mathsf{rej}}, M', q_?, q_{\mathsf{yes}}, q_{\mathsf{no}})$ be an oracle Turing machine such that the following holds:

- M solves an NP^{NExp}-complete problem;
- the time consumption of M is bounded by a polynomial p;
- the time consumption of $M' = (Q', \Sigma', q'_0, \Delta', q'_{\mathsf{acc}}, q'_{\mathsf{rej}})$ is bounded by $2^{q(n)}$, with q a polynomial.

Our NP^{NExp}-hardness proof uses a reduction of the word problem of M. Thus, let $w \in \Sigma^*$ be an input for M of length n, and let $m = p(n)$ and $m' = q(p(n))$. We will construct three TBoxes $\mathcal{T}_w$, $\mathcal{T}'_w$, and $\mathcal{T}''_w$, circumscription patterns CP, CP', and CP'', and a concept C such that M accepts w iff C is simultaneously satisfiable w.r.t. $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}_w, \emptyset)$, $\mathsf{Circ}_{\mathsf{CP}'}(\mathcal{T}'_w, \emptyset)$, and $\mathsf{Circ}_{\mathsf{CP}''}(\mathcal{T}''_w, \emptyset)$. Then, Lemma 10 yields a reduction to (non-simultaneous) satisfiability w.r.t. concept-circumscribed cKBs. Intuitively, the purpose of the first TBox $\mathcal{T}_w$ is to impose a basic structure on the domain, while $\mathcal{T}'_w$ describes computations of M, and $\mathcal{T}''_w$ describes computations of M'.

The details of $\mathcal{T}_w$ can be found in Figure 4, where we use the same abbreviations

$$
\begin{array}{rcll}
\top & \sqsubseteq & \exists \mathsf{aux}.\mathsf{NExp} & (20)\\
\mathsf{NExp} & \sqsubseteq & \exists r.\mathsf{NExp} \sqcap \exists u.\mathsf{NExp} & (21)\\
\mathsf{NExp} & \sqsubseteq & \forall r.(Y{=}Y) & (22)\\
\mathsf{NExp} & \sqsubseteq & \forall r.(X{+}{+}) & (23)\\
\mathsf{NExp} & \sqsubseteq & \forall u.(X{=}X) & (24)\\
\mathsf{NExp} & \sqsubseteq & \forall u.(Y{+}{+}) & (25)\\
\\
\top & \sqsubseteq & \displaystyle\bigsqcap_{i<m} \exists \mathsf{aux}.(\mathsf{Result} \sqcap R_i) & (26)\\
\mathsf{Result} & \sqsubseteq & \displaystyle\bigsqcap_{i<j<m} \neg(R_i \sqcap R_j) & (27)\\
\\
\top & \sqsubseteq & \exists \mathsf{aux}.\mathsf{NP} & (28)
\end{array}
$$

Figure 4: The TBox $\mathcal{T}_w$.

as in the previous reduction. The circumscription pattern for $\mathcal{T}_w$ is

$$\mathsf{CP} := (\emptyset, \{\mathsf{NExp}, \mathsf{Result}, \mathsf{NP}\}, \emptyset, V),$$

with V containing all remaining predicates used in $\mathcal{T}_w$. The purpose of Lines 20 to 25 is to ensure that, for each possible value (i, j) of the counters X and Y, there is at least one instance of NExp that satisfies $(X = i)$ and $(Y = j)$. By minimizing NExp, we thus enforce that NExp has *exactly* $2^{m'} \times 2^{m'}$ elements. These elements are interconnected via the roles r ("right") and u ("up"). Indeed, it is not difficult to see that the structure $(\mathsf{NExp}^{\mathcal{I}}, r^{\mathcal{I}}, u^{\mathcal{I}})$ is isomorphic to the $2^{m'} \times 2^{m'}$-torus in each model $\mathcal{I}$ of $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}_w, \emptyset)$. Later on, we use this grid to encode computations of the oracle machine M'.

Together with the minimization of Result, Lines 26 and 27 guarantee that there is exactly one instances of the concept $\mathsf{Result} \sqcap R_i$, for all $i < m$. Intuitively, if M makes a call to the oracle in the i-th step, then the result of this call will be stored in the (unique) instance of $\mathsf{Result} \sqcap R_i$: this instance will satisfy the concept name Rej iff M' rejected the input. Finally, Line 28 and the minimization of NP guarantee that there is exactly one instance of NP. This instance will be used to represent the computation of M.

The purpose of $\mathcal{T}'_w$ is to describe computations of M. We use the following concept names:

- For all $a \in \Sigma$, $i, j < m$, and $k \in \{1, 2\}$, we introduce a concept name $S_a^{i,j,k}$. Intuitively, $S_a^{i,j,k}$ expresses that a is the symbol in the j-th cell of the k-th tape in the i-th step of M's computation. We start our numbering of tape cells and steps with 0.

- For all $q \in Q$ and $i < m$, Q_q^i is a concept name expressing that M is in state q in the i-th step of the computation.

$$\mathsf{NP} \sqsubseteq \exists \mathsf{res}_i.(\mathsf{Result} \sqcap R_i) \sqcap \forall \mathsf{res}_i.(\mathsf{Result} \sqcap R_i) \quad (29)$$

$$\mathsf{NP} \sqsubseteq (Q^i_{q_?} \sqcap \exists \mathsf{res}_i.\mathsf{Rej}) \rightarrow Q^{i+1}_{q_{\mathsf{no}}} \quad (30)$$

$$\mathsf{NP} \sqsubseteq (Q^i_{q_?} \sqcap \exists \mathsf{res}_i.\neg\mathsf{Rej}) \rightarrow Q^{i+1}_{q_{\mathsf{yes}}} \quad (31)$$

$$\mathsf{NP} \sqsubseteq (Q^i_{q_?} \sqcap H^{i,k}_j) \rightarrow H^{i+1,k}_j \quad (k = 1, 2) \quad (32)$$

$$\mathsf{NP} \sqsubseteq \bigsqcap_{a \in \Sigma} \bigsqcap_{j<m} \big((Q^i_{q_?} \sqcap S^{i,j}_a) \rightarrow S^{i,j}_a\big) \quad (33)$$

Figure 5: The TBox $\mathcal{T}'_w$ (partly).

- For all $q \in Q$, $i, j < m$, and $k \in \{1, 2\}$, $H^{i,k}_j$ is a concept name expressing that the k-th head of M is on cell j in the i-th step of the computation.

In $\mathcal{T}'_w$, we describe computations of M employing the usual set of axioms: each tape cell contains exactly one alphabet symbol in each step, there is exactly one current state at each step, the transition table is obeyed, etc. We leave details to the reader and give, in Figure 5, only the part of $\mathcal{T}'_w$ that deals with the oracle. We assume that copies of Lines 29 to 33 are contained in $\mathcal{T}'_w$ for every $i < m$. The circumscription pattern is simply $\mathsf{CP}' := (\emptyset, \emptyset, \emptyset, V)$, with V the set of all predicates used in $\mathcal{T}'_w$. Line 29 ensures that the instance of NP can reach the (unique) instance of $\mathsf{Result} \sqcap R_i$ via the role res_i, for all $i < m$. Lines 30 and 31 deal with transitions of M in the query state: the result of the oracle call is looked up in the corresponding instance of Result. Finally, Lines 32 and 33 merely ensure that the head position and symbol under the head does not change when querying the oracle.

The purpose of $\mathcal{T}''_w$ is to describe computations of M'. As already noted, such computations are represented using the instances of NExp: the $2^{m'}$ instances satisfying $(X = i)$ represent the i-th configuration of M', for $i < 2^{m'}$. Here, the instance of $(Y = 0)$ represents the first tape cell and the instance of $(Y = 2^{m'})$ represents the last tape cell. Note that we may have to describe more than a single computation of M' as M may visit the state $q_?$ more than once. All these computations are "overlayed" in the NExp grid using different concept names for different computations. More precisely, we use the following concept names:

- For all $a \in \Sigma$ and $i < m$, a concept name S^i_a. If S^i_a is satisfied by some instance of NExp with $X = j$ and $Y = k$, then the i-th computation of M' has, in it's j-th step, label a on the k-th cell.

- For all $q \in Q$ and $i < m$, a concept name Q^i_q. The purpose of this concept name is two-fold: first, it represents the current state of M' in the i-th computation. And second, it indicates the head position in the i-th computation.

The behaviour of M' is again described via the usual axioms. Details are omitted. In Figure 6, we only show the GCIs of $\mathcal{T}''_w$ that deal with the interaction with M. Similarly to the case of $\mathcal{T}'_w$, we assume that $\mathcal{T}''_w$ contains a copy of Lines 34 to 39 for

$$\mathsf{NExp} \sqsubseteq (\neg(X = 2^{m'} - 1) \rightarrow \exists r'.\mathsf{NExp}) \sqcap (\neg(Y = 2^{m'} - 1) \rightarrow \exists u'.\mathsf{NExp}) \quad (34)$$

$$\mathsf{NExp} \sqsubseteq \forall r'.(Y{=}Y) \sqcap \forall r'.(X{+}{+}) \sqcap \forall u'.(X{=}X) \sqcap \forall u'.(Y{+}{+}) \quad (35)$$

$$\mathsf{NExp} \sqsubseteq \exists \mathsf{res}'_i.(\mathsf{Result} \sqcap R_i) \sqcap \forall \mathsf{res}'_i.(\mathsf{Result} \sqcap R_i) \quad (36)$$

$$\mathsf{NExp} \sqsubseteq \exists \mathsf{toNP}.\mathsf{NP} \sqcap \forall \mathsf{toNP}.\mathsf{NP} \quad (37)$$

$$\mathsf{NExp} \sqsubseteq \bigsqcap_{j<m} \bigsqcap_{a\in\Sigma} \Big(((X = 0) \sqcap (Y = j) \sqcap \forall \mathsf{toNP}.S_a^{i,j,2}) \rightarrow S_a^i \Big) \quad (38)$$

$$\mathsf{NExp} \sqsubseteq Q^i_{q'_{rej}} \rightarrow \forall \mathsf{res}'_i.\mathsf{Rej} \quad (39)$$

Figure 6: The TBox $\mathcal{T}''_w$ (partly).

all $i < m$. With $\mathcal{T}''_w$, we use the circumscription pattern $\mathsf{CP}'' := (\emptyset, \{\mathsf{Rej}\}, \emptyset, V')$, where V' contains all other predicates used in $\mathcal{T}''_w$.

The purpose of Lines 34 and 35 is to regenerate the grid structure of NExp using the roles r' und u'. This is necessary since the roles r and u are used in $\mathcal{T}_w$, and, with simultaneous satisfiability, the TBoxes cannot share any role names. Lines 36 and 37 ensure that every instance of NExp reaches (only) the instance of NP via the role toNP, and (only) the instance of $\mathsf{Result} \sqcap R_i$ via the role res'_i, for all $i < m$. Line 38 guarantees that the i-th computation of M' uses as its input the contents of the second tape of M, as it is at the i-th step of M. Finally, Line 39 ensures that, if the i-th computation of M is rejecting, then Rej is true in the instance of $\mathsf{Result} \sqcap R_i$.

Note that M is a non-deterministic machine and may have more than one computation. For storing Rej in $\mathsf{Result} \sqcap R_i$, we need to know that *all* these computations are rejecting. To deal with this issue, Rej is minimized with all other predicates varying: if there exists an accepting computation of M' on i-th input, then we can represent this computation in NExp and make Rej false in the instance of $\mathsf{Result} \sqcap R_i$. Hence, Rej holds at $\mathsf{Result} \sqcap R_i$ iff there exists no accepting computation. Note that we cannot fix the concept names $X_0, \ldots, X_{m-1}, Y_0, \ldots, Y_{m-1}$ while minimizing Rej since we would get an unbounded number of fixed concept names. Intuitively, the result is that the elements of NExp may change their position during minimization, and with them the roles r' and u'. However, this is not harmful since $\mathcal{T}_w$ and Lines 34 and 35 ensure that that $(\mathsf{NExp}^\mathcal{I}, (r')^\mathcal{I}, (u')^\mathcal{I})$ is always isomorphic to a grid, and (the omitted part of) $\mathcal{T}''_w$ ensures that the elements of NExp always encode computations of M'.

The proof of the following lemma is left to the reader.

Lemma 13 *M accepts w iff $\mathsf{NP} \sqcap \bigsqcup_{i<m} Q^i_{q_{\mathsf{acc}}}$ is simultaneously satisfiable w.r.t. $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}_w, \emptyset)$, $\mathsf{Circ}_{\mathsf{CP}'}(\mathcal{T}'_w, \emptyset)$, and $\mathsf{Circ}_{\mathsf{CP}''}(\mathcal{T}''_w, \emptyset)$.*

Since it is easily checked that the size of the constructed TBoxes is polynomial in n, we get the following result.

Theorem 14 *In $\mathcal{ALC}$, satisfiability w.r.t. simple cKBs is hard for $\mathrm{NP}^{\mathrm{NExp}}$.*

It is interesting to observe that the reduction even works if we assume ABoxes to be empty. Corresponding lower bounds for subsumption and the instance problems follow from the reduction given in Section 2.

5 Undecidability

In the preceeding sections, we have pinpointed the exact computational complexity of reasoning w.r.t. concept-circumscribed KBs. In particular, we have proved that reasoning w.r.t. such KBs is decidable. In this section, we extend the framework of concept-circumscribed KBs by allowing role names to be fixed during the minimization of concept names. Interestingly, it turns out that this seemingly harmless modification leads to undecidability of reasoning. More precisely, we prove that reasoning is undecidable already in $\mathcal{ALC}$, and that this holds even with empty TBoxes.

A circumscribed knowledge base $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$ is called *concept-minimizing* if $\mathsf{CP} = (\prec, M, F, V)$ with M a set of concept names. We prove that, in $\mathcal{ALC}$, the instance problem w.r.t. concept-minimizing cKBs is undecidable. By the reductions given in Section 2, this implies undecidability of the other reasoning problems as well. The proof is by a reduction of the semantic consequence problem of modal logic on transitive frames, which has been proved undecidable in [8].

A *frame* is a structure $\mathfrak{F} = (\Delta^{\mathfrak{F}}, \cdot^{\mathfrak{F}})$, where $\mathfrak{F}$ a non-empty domain, r a role name, and $r^{\mathfrak{F}} \subseteq \Delta^{\mathfrak{F}} \times \Delta^{\mathfrak{F}}$. A *pointed frame* is a pair $(\mathfrak{F}, d)$ such that $d \in \Delta^{\mathfrak{F}}$. For $\mathfrak{F} = (\Delta^{\mathfrak{F}}, r^{\mathfrak{F}})$ a frame, $d, e \in \Delta^{\mathfrak{F}}$, and $n \in \mathbb{N}$, we write $d(r^{\mathfrak{F}})^{\leq n}e$ iff there exists a sequence $d_0, \ldots, d_n \in \Delta^{\mathfrak{F}}$ with $d = d_0$, $e = d_n$, and $d_i \; r^{\mathfrak{F}} d_{i+1}$ for $i < n$. Moreover, $d \in \Delta^{\mathfrak{F}}$ is called a *root* of $\mathfrak{F}$ if for every $e \in \Delta^{\mathfrak{F}}$, there exists m such that $d(r^{\mathfrak{F}})^{\leq m}e$. An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is *based on* a frame $\mathfrak{F}$ iff $\Delta^{\mathfrak{F}} = \Delta^{\mathcal{I}}$ and $r^{\mathcal{I}} = r^{\mathfrak{F}}$. We say that a concept C is *valid* on $\mathfrak{F}$ and write $\mathfrak{F} \models C$ iff $C^{\mathcal{I}} = \Delta^{\mathcal{I}}$ for every interpretation $\mathcal{I}$ based on $\mathfrak{F}$, and $(\mathfrak{F}, d) \models C$ iff $d \in C^{\mathcal{I}}$ for every interpretation $\mathcal{I}$ based on $\mathfrak{F}$. The following theorem restates, in a DL formulation, the undecidability of the semantic consequence problem of modal logic on transitive frames.

Theorem 15 (Chagrov) *There exists an $\mathcal{ALC}$ concept E containing only the concept name A and the role r such that the following problem is undecidable: given an $\mathcal{ALC}$ concept D, does there exist a transitive frame $\mathfrak{F}$ such that $\mathfrak{F} \models E$ and $\mathfrak{F} \not\models D$.*

For convenience, we will use the following abbreviation: for $m \in \mathbb{N}$, we use $\forall^m r.C$ to denote C if $m = 0$, and $\forall^m r.C \sqcap \forall r.\forall^m r.C$ if $m > 0$. As usual, the role depth $\mathsf{rd}(C)$ of a concept C is defined as the nesting depth of the constructors $\exists r.D$ and $\forall r.D$ in C. The following lemma establishes a connection between the instance problem w.r.t. concept-minimizing cKBs and a bounded version of the semantic consequence problem (not yet on transitive frames). For the sake of readability, we write concept assertions $C(a)$ in the form $a : C$

Lemma 16 *Let C be an $\mathcal{ALC}$ concept whose only role is r and whose only concept name is A. Let D be a concept not containing A and whose only role is r. Then, for every $m > 0$, the following conditions are equivalent:*

(i) $\mathsf{Circ}_{\mathsf{CP}}(\emptyset, \mathcal{A}) \models a : \forall^m r.C \sqcap \neg D$ *with* $\mathsf{CP} = (\emptyset, \{A\}, \{r\}, \emptyset)$ *and*

$$\mathcal{A} = \{a : (\neg\forall^m r.C \sqcup \forall^{m+\mathsf{rd}(C)} r.A)\};$$

(ii) there exists a pointed frame $(\mathfrak{F}, d)$ *such that* $(\mathfrak{F}, d) \models \forall^m r.C$ *and* $(\mathfrak{F}, d) \not\models D$.

Proof. (i) implies (ii). Let $\mathcal{I}$ be a model of $\mathsf{Circ}_{\mathsf{CP}}(\emptyset, \mathcal{A})$ such that $a^{\mathcal{I}} \in (\forall^m r.C \sqcap \neg D)^{\mathcal{I}}$. Suppose $\mathcal{I}$ is based on the frame $\mathfrak{F}$, and set $d := a^{\mathcal{I}}$. We show that $(\mathfrak{F}, d) \models \forall^m r.C$ and $(\mathfrak{F}, d) \not\models D$. The latter is easy as it is witnessed by the interpretation $\mathcal{I}$. To show the former, let $\mathcal{J}$ be an interpretation based on $\mathcal{F}$. We distinguish two cases:

- $A^{\mathcal{J}} \supseteq \{e \in \Delta^{\mathcal{J}} \mid d(r^{\mathcal{J}})^{\leq m+\mathsf{rd}(C)} e\}$.

  Since $a^{\mathcal{I}} \in (\forall^m r.C)^{\mathcal{I}}$ and $\mathcal{I}$ is a model of $\mathsf{Circ}_{\mathsf{CP}}(\emptyset, \mathcal{A})$, it is not hard to see that

  $$A^{\mathcal{I}} = \{e \in \Delta^{\mathcal{I}} \mid d(r^{\mathcal{I}})^{\leq m+\mathsf{rd}(C)} e\}. \qquad (*)$$

  Moreover, $d \in (\forall^m r.C)^{\mathcal{I}}$. Since $\mathcal{I}$ and $\mathcal{J}$ are based on the same frame and the truth of $\forall^m r.C$ at d depends on the truth value of A only at those objects $e \in \Delta^{\mathcal{I}}$ with $d(r^{\mathcal{I}})^{\leq m+\mathsf{rd}(C)} e$, we have $d \in (\forall^m r.C)^{\mathcal{J}}$ and are done.

- $A^{\mathcal{J}} \not\supseteq \{e \in \Delta^{\mathcal{J}} \mid d(r^{\mathcal{J}})^{\leq m+\mathsf{rd}(C)} e\}$.

  Let $\mathcal{J}'$ be the modification of $\mathcal{J}$ where $A^{\mathcal{J}'} = A^{\mathcal{J}} \cap \{e \in \Delta^{\mathcal{J}} \mid d(r^{\mathcal{J}})^{\leq m+\mathsf{rd}(C)} e\}$. By $(*)$, $\mathcal{J}' \prec_{\mathsf{CP}} \mathcal{I}$. If $d \in (\neg\forall^m r.C)^{\mathcal{J}'}$, then $\mathcal{J}'$ is a model of $\mathcal{A}$ and we have a contradiction to the fact that $\mathcal{I}$ is a model of $\mathsf{Circ}_{\mathsf{CP}}(\emptyset, \mathcal{A})$. Thus, $d \in (\forall^m r.C)^{\mathcal{J}'}$. Since the truth of $\forall^m r.C$ at d depends on the truth value of A only at those objects $e \in \Delta^{\mathcal{J}'}$ with $d(r^{\mathcal{J}'})^{\leq m+\mathsf{rd}(C)} e$, we have $d \in (\forall^m r.C)^{\mathcal{J}}$ and are done.

(ii) implies (i). Suppose there exists a pointed frame $(\mathfrak{F}, d)$ such that $(\mathfrak{F}, d) \models \forall^m r.C$ and $(\mathfrak{F}, d) \not\models D$. We may assume that d is a root of $\mathfrak{F}$. Let $\mathcal{I}$ be an interpretation based on $\mathfrak{F}$ such that $d \in (\neg D)^{\mathcal{I}}$. We may assume that $A^{\mathcal{I}} = \{e \in \Delta^{\mathcal{I}} \mid d(r^{\mathcal{I}})^{\leq m+\mathsf{rd}(C)} d\}$ (since A does not occur in D) and $a^{\mathcal{I}} = d$. Then $a^{\mathcal{I}} \in (\forall^m r.C \sqcap \neg D)^{\mathcal{I}}$. It remains to show that there does not exist an $\mathcal{I}' \prec_{\mathsf{CP}} \mathcal{I}$ such that $a^{\mathcal{I}'} \in (\neg\forall^m r.C \sqcup \forall^{m+\mathsf{rd}(C)} r.A))^{\mathcal{I}'}$. This is straightforward: from $(\mathfrak{F}, d) \models \forall^m r.C$, we obtain that there does not exist any $\mathcal{I}'$ such that $d \in (\neg\forall^m r.C)^{\mathcal{I}'}$ and clearly there does not exist any $A^{\mathcal{I}'} \subset A^{\mathcal{I}}$ such that $d \in (\forall^{m+\mathsf{rd}(C)} r.A)^{\mathcal{I}'}$. ❑

The following lemma relates the bounded version of the semantic consequence problem (on unrestricted frames) to the semantic consequence problem on transitive frames. It utilizes the concept $\forall r.A \rightarrow \forall r.\forall r.A$, the DL version of the modal formula $\Box p \rightarrow \Box\Box p$ that is well-known to be valid on a frame iff the frame is transitive.

Lemma 17 *Let* $C_1 = \neg\forall r.A \sqcup \forall r.\forall r.A$, C_2 *be an* $\mathcal{ALC}$ *concept containing only the role* r *and the concept name* A*, and let* D *be a concept containing only the role* r*. Then the following conditions are equivalent:*

(i) there exists a transitive frame $\mathfrak{F}$ *such that* $\mathfrak{F} \models C_2$ *and* $\mathfrak{F} \not\models D$*;*

(ii) There exists a pointed frame $(\mathfrak{F}, w)$ *such that* $(\mathfrak{F}, w) \models \forall^1 r.(C_1 \sqcap C_2)$ *and* $(\mathfrak{F}, w) \not\models D$.

Proof. (i) implies (ii). Let $\mathcal{F}$ be a transitive frame such that $\mathfrak{F} \models C_2$ and $\mathfrak{F} \not\models D$. Take a $w \in \Delta^{\mathfrak{F}}$ such that $(\mathfrak{F}, w) \not\models D$. We may assume that w is a root of $\mathfrak{F}$. Since $\mathfrak{F} \models C_2$ and $\mathfrak{F}' \models C_1$ for every transitive frame $\mathfrak{F}'$, $(\mathfrak{F}, w)$ is as required for (ii).

(ii) implies (i). Let $(\mathfrak{F}, w)$ be a pointed frame such that $(\mathfrak{F}, w) \models \forall^1 r.(C_1 \sqcap C_2)$ and $(\mathfrak{F}, w) \not\models D$. We may assume that w is a root of $\mathfrak{F}$. It is not difficult to show that $(\mathfrak{F}, w) \models \forall^1 r.C_1$ implies that $r^{\mathfrak{F}}$ is transitive. Therefore, from $(\mathfrak{F}, w) \models \forall^1 r.C_2$ we obtain $\mathfrak{F} \models C_2$. We conclude that $\mathfrak{F}$ is as required for (i). ❑

We are now in a position to prove the undecidability result.

Theorem 18 *In $\mathcal{ALC}$, the instance problem w.r.t. concept-minimizing cKBs is undecidable. This even holds in the case of empty TBoxes.*

Proof. Take the concept E from Theorem 15, the concept C_1 from Lemma 17, and set $C_2 := E$ and $C := C_1 \sqcap C_2$. Then, by Theorem 15 and Lemma 17, the following is undecidable: given a concept D, does there exists a pointed frame $(\mathfrak{F}, w)$ such that $(\mathfrak{F}, w) \models \forall^1 r.C$ and $(\mathfrak{F}, w) \not\models D$. Since we are concerned with validity on frames, we may w.l.o.g. assume that D does not contain the concept name A. Therefore, by Lemma 16, the following is undecidable: given a concept D not containing A, is a an instance of $\forall^1 r.C \sqcap \neg D$ w.r.t. $\mathsf{Circ}_{\mathsf{CP}}(\emptyset, \{a : (\neg\forall^1 r.C \sqcup \forall^{1+\mathsf{rd}(C)} r.A)\}$, where $\mathsf{CP} = (\emptyset, \{A\}, \{r\}, \emptyset)$. ❑

By the reductions given in Section 2, it follows that satsfiability and subsumption w.r.t. concept-minimizing cKBs are undecidable as well (also in the case of empty TBoxes).

Minimized vs. Fixed Role Names

Unlike fixed concept names, fixed role names cannot be simulated using minimized role names. This is due to the fact that Boolean operators on roles are not available in standard DLs. Thus, Theorem 18 does not imply undecidability of reasoning w.r.t. *concept-fixing* cKBs, in which role names are allowed to be minimized, but only concept names can be fixed. In general, we have to leave decidability of reasoning w.r.t. concept-fixing cKBs as an open problem. However, we show in the following that reasoning w.r.t. such cKBs is decidable when TBoxes are empty. Together with Theorem 18, which also applies to the case of empty TBoxes, we have thus identified a case where reasoning with minimized role names is decidable, but reasoning with fixed role names is not.

Theorem 19 *In $\mathcal{ALC}$, satisfiability w.r.t. concept-fixing cKBs $\mathsf{Circ}_{\mathsf{CP}}(\mathcal{T}, \mathcal{A})$ is decidable in* NExp$^{\text{NP}}$ *if $\mathcal{T}$ is empty.*

Proof. We establish a bounded model property using a "selective filtration"-style argument. To prove Theorem 19, we can then proceed as in Theorem 6. Details are omitted.

Let $\mathsf{Circ}_{\mathsf{CP}}(\emptyset, \mathcal{A})$ be a concept-fixing cKB with $\mathsf{CP} = (\prec, M, F, V)$, and let C_0 be a concept that is satisfiable w.r.t. $\mathsf{Circ}_{\mathsf{CP}}(\emptyset, \mathcal{A})$.[2] Set

$$n := \max(\{\mathsf{rd}(C_0)\} \cup \{\mathsf{rd}(C) \mid C(a) \in \mathcal{A}\}) \text{ and } m := (|\mathcal{A}| + |C_0|)^{n+1},$$

We show that there exists a model $\mathcal{J}$ of $\mathsf{Circ}_{\mathsf{CP}}(\emptyset, \mathcal{A})$ satisfying C_0 such that $|\Delta^{\mathcal{J}}| \leq m$. Let $\mathcal{I}$ be a model of $\mathsf{Circ}_{\mathsf{CP}}(\emptyset, \mathcal{A})$ such that there exists a $d_0 \in C_0^{\mathcal{I}}$. Let $\mathsf{ex}(C_0, \mathcal{A})$ be the set of concepts of the form $\exists r.C$ that occur as a (not necessarily proper) subconcept in C_0 or $\mathcal{A}$. For each $d \in \Delta^{\mathcal{I}}$, fix a minimal set $D(d) \subseteq \Delta^{\mathcal{I}}$ such that, for every concept $\exists r.C \in \mathsf{ex}(C_0, \mathcal{A})$, there exists $e \in D(d)$ such that $(d, e) \in r^{\mathcal{I}}$ and $e \in C^{\mathcal{I}}$. Clearly, $|D(e)| \leq |C_0| + |\mathcal{A}|$ for each $e \in \Delta^{\mathcal{I}}$. Next, define a set $D_0 \subseteq \Delta^{\mathcal{I}}$ by setting

$$D_0 := \{d_0\} \cup \{a^{\mathcal{I}} \mid a \in \mathsf{N_I} \text{ occurs is } \mathcal{A}\}.$$

Define sets $D_i \subseteq \Delta^{\mathcal{I}}$, $1 \leq i \leq n$, inductively by

$$D_{i+1} := (\bigcup_{d \in D_i} D(d))$$

and set $\Delta_n := \bigcup_{0 \leq i \leq n} D_i$. Define an interpretation $\mathcal{I}'$ with domain $\Delta^{\mathcal{I}}$ as follows:

- $a^{\mathcal{I}'} = a^{\mathcal{I}}$, for all object names a;
- for $r \in M \cup V$, $(d, e) \in r^{\mathcal{I}'}$ if $d \in \Delta_n \setminus D_n$, $e \in D(d)$, and $(d, e) \in r^{\mathcal{I}}$;
- for $A \in M \cup V$, $A^{\mathcal{I}'} = A^{\mathcal{I}} \cap \Delta_n$;
- for $A \in F$, $A^{\mathcal{I}'} = A^{\mathcal{I}}$.

A straighforward inductive argument shows that $\mathcal{I}'$ is a model of $\mathcal{A}$ such that $d_0 \in C_0^{\mathcal{I}'}$. Note that we did not change the interpretation of the $A \in F$. Moreover, we have $p^{\mathcal{I}'} \subseteq p^{\mathcal{I}}$ for every $p \in M$. Together with the fact that $\mathcal{I}'$ is a model of $\mathcal{A}$ and $\mathcal{I}' \not< \mathcal{I}$, we even get $p^{\mathcal{I}'} = p^{\mathcal{I}}$ for every $p \in M$. It follows that $\mathcal{I}'$ is a model for $\mathsf{Circ}_{\mathsf{CP}}(\emptyset, \mathcal{A})$ because $\mathcal{J} <_{\mathsf{CP}} \mathcal{I}'$ would imply $\mathcal{J} <_{\mathsf{CP}} \mathcal{I}$.

Note that $r^{\mathcal{I}'} \subseteq \Delta_n \times \Delta_n$, for every role r. Now define an interpretation $\mathcal{J}$ with domain $\Delta^{\mathcal{J}} = \Delta_n$ by putting

- $A^{\mathcal{J}} = A^{\mathcal{I}'} \cap \Delta_n$, for every concept name A;
- $r^{\mathcal{J}} = r^{\mathcal{I}'}$, for every role name r;
- $a^{\mathcal{J}} = a^{\mathcal{I}}$, for every object name a from $\mathcal{A}$.

We still have that $\mathcal{J}$ is a model for $\mathcal{A}$ satisfying C_0. Moreover, any interpretation $\mathcal{J}' <_{\mathsf{CP}} \mathcal{J}$ satisfying $\mathcal{A}$ can be easily extended to an interpretation $\mathcal{I}'' <_{\mathsf{CP}} \mathcal{I}'$ satisfying $\mathcal{A}$. Hence, no such interpretation exists and $\mathcal{J}$ is a model for $\mathsf{Circ}_{\mathsf{CP}}(\emptyset, \mathcal{A})$. From $|\Delta_n| \leq m$ we derive $|\Delta^{\mathcal{J}}| \leq m$. ❏

[2] Note that we cannot eliminate fixed atomic concepts from the circumscription pattern because this would require the introduction of a TBox.

We leave it as an open problem whether satisfiability w.r.t. concept-fixing cKBs is decidable in the case of non-empty TBoxes.

6 Conclusions and Perspectives

We have shown that non-monotonic extensions of DLs based on circumscription can result in formalisms that are much less restricted than existing non-monotonic DLs, but for which reasoning is still decidable. In particular, the resulting family of DLs allows to model defeasible inheritance without the usual and severe restriction to named individuals. However, we view this paper only as a first step towards usable non-monotonic DLs. In particular, our upper bounds are based on massive non-deterministic guessing, and are thus far from being implementable in efficient systems. Ideally, one would like to have well-behaved extensions of the tableau algorithms that underly state-of-the-art DL reasoners [4]. It seems that existing sequent calculi for (propositional) circumscription and minimal entailment [6, 23] could provide a good starting point. Additionally to having a usable implementation, it is desirable to develop a design methodology for modelling defeasible inheritance. In particular, such a methodology should address the problem of finding appropriate circumscription patterns. Also from a theoretical perspective, our initial investigation leaves open a number of exciting questions. First, it is open whether or not minimizing roles leads to undecidability in the presence of non-empty TBoxes. Second, our current techniques are limited to non-monotonic extensions of DLs that have the finite model property, and it would be desirable to aleviate this limitation. And third, it is interesting whether the observed impact of predicate number and arity on computational complexity can be observed in other formalisms such as the extension of the two-variable fragment of first-order logic with circumscription.

References

[1] Franz Baader and Bernhard Hollunder. Embedding defaults into terminological knowledge representation formalisms. *J. Autom. Reasoning*, 14(1):149–180, 1995.

[2] Franz Baader and Bernhard Hollunder. Priorities on defaults with prerequisites, and their application in treating specificity in terminological default logic. *J. Autom. Reasoning*, 15(1):41–68, 1995.

[3] Franz Baader, Ian Horrocks, and Ulrike Sattler. Description logics as ontology languages for the semantic web. In Dieter Hutter and Werner Stephan, editors, *Festschrift in honor of Jörg Siekmann*, Lecture Notes in Artificial Intelligence. Springer-Verlag, 2003.

[4] Franz Baader and Ulrike Sattler. Tableau algorithms for description logics. In R. Dyckhoff, editor, *Proceedings of the International Conference on Automated Reasoning with Tableaux and Related Methods (Tableaux 2000)*, volume 1847 of *Lecture Notes in Artificial Intelligence*, pages 1–18. Springer-Verlag, 2000.

[5] Piero A. Bonatti and Thomas Eiter. Querying disjunctive databases through nonmonotonic logics. *Theor. Comput. Sci.*, 160(1&2):321–363, 1996.

[6] Piero A. Bonatti and Nicola Olivetti. Sequent calculi for propositional nonmonotonic logics. *ACM Trans. Comput. Log.*, 3(2):226–278, 2002.

[7] Gerhard Brewka. Adding priorities and specificity to default logic. In *Logics in Artificial Intelligence, European Workshop, JELIA '94, York, UK, September 5-8, 1994, Proceedings*, volume 838 of *Lecture Notes in Computer Science*, pages 247–260. Springer, 1994.

[8] A. Chagrov. Undecidable properties of superintuitionistic logics. In S.V. Jablonskij, editor, *Mathematical Problems of Cybernetics*, volume 5, pages 67 – 108. Physmatlit, 1994. in Russian.

[9] R.A. Cote, D.J. Rothwell, J.L. Palotay, R.S. Beckett, and L. Brochu. The systematized nomenclature of human and veterinary medicine. Technical report, SNOMED International, Northfield, IL: College of American Pathologists, 1993.

[10] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Werner Nutt, and Andrea Schaerf. An epistemic operator for description logics. *Artif. Intell.*, 100(1-2):225–274, 1998.

[11] Francesco M. Donini, Daniele Nardi, and Riccardo Rosati. Autoepistemic description logics. In *IJCAI (1)*, pages 136–141, 1997.

[12] Thomas Eiter, Georg Gottlob, and Heikki Mannila. Disjunctive Datalog. *ACM Transactions on Database Systems*, 22(3):364–418, September 1997.

[13] Thomas Eiter, Thomas Lukasiewicz, Roman Schindlauer, and Hans Tompits. Combining answer set programming with description logics for the semantic web. In *Proceedings of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2004)*, pages 141–151, 2004.

[14] E. Grädel, M. Otto, and E. Rosen. Two-Variable Logic with Counting is Decidable. In *Proceedings of Twelfth IEEE Symposium on Logic in Computer Science (LICS'97)*, 1997.

[15] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL*, 8(3):239–264, 2000.

[16] Patrick Lambrix, Nahid Shahmehri, and Niclas Wahlloef. A default extension to description logics for use in an intelligent search engine. In *HICSS '98: Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences-Volume 5*, page 28, Washington, DC, USA, 1998. IEEE Computer Society.

[17] V. Lifschitz. Computing circumscription. In *Proceedings of IJCAI'85*, pages 121–127, 1985.

[18] Vladimir Lifschitz. Circumscription. In D.M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *The Handbook of Logic in AI and Logic Programming 3*, pages 298–352. Oxford University Press, 1993.

[19] Vladimir Lifschitz. Nested abnormality theories. *Artif. Intell.*, 74(2):351–365, 1995.

[20] J. McCarthy. Circumscription: a form of nonmonotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.

[21] J. McCarthy. Applications of circumscription in formalizing common sense knowledge. *Artificial Intelligence*, 28:89–116, 1986.

[22] Marvin Minsky. A framework for representating knowledge. In Patrick Henry Winston, editor, *The Psychology of Computer Vision*, pages 211–277. McGraw-Hill, New York, USA, 1975.

[23] N. Olivetti. Tableaux and sequent calculus for minimal entailment. *Journal of Automated Reasoning*, 9:99–139, 1992.

[24] Leszek Pacholski, Wiesław Szwast, and Lidia Tendera. Complexity results for first-order two-variable logic with counting. *SIAM Journal on Computing*, 29(4):1083–1117, August 2000.

[25] Lin Padgham and Tingting Zhang. A terminological logic with defaults: A definition and an application. In Ruzena Bajcsy, editor, *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 662–668, San Mateo, California, 1993. Morgan Kaufmann.

[26] Ian Pratt-Hartmann. Complexity of the two-variable fragment with counting quantifiers. *Journal of Logic, Language, and Information*, 14(3):369–395, 2005.

[27] M. R. Quillian. Semantic memory. In M. Minsky, editor, *Semantic Information Processing*, pages 227–270. MIT Press, Cambridge, MA, USA, 1968.

[28] Alan Rector. Defaults, context, and knowledge: Alternatives for owl-indexed knowledge bases. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 226–237, 2004.

[29] Alan Rector and Ian Horrocks. Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In *Proceedings of the Workshop on Ontological Engineering, AAAI Spring Symposium (AAAI'97)*, Stanford, CA, 1997. AAAI Press.

[30] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[31] Riccardo Rosati. On the decidability and complexity of integrating ontologies and rules. *Journal of Web Semantics*, 2005. To appear.

[32] Robert Stevens, Mikel Egana Aranguren, Katy Wolstencroft, Ulrike Sattler, Nick Drummond, Matthew Horridge, and Alan Rector. Managing OWL's limitations in modelling biomedical knowledge. Submitted, 2005.

[33] Iain A. Stewart. Complete problems involving Boolean labelled structures and projection transactions. *Journal of Logic and Computation*, 1(6):861–882, December 1991.

[34] Umberto Straccia. Default inheritance reasoning in hybrid kl-one-style logics. In *IJCAI*, pages 676–681, 1993.

[35] Stephan Tobies. The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *Journal of Artificial Intelligence Research*, 12:199–217, 2000.