Description Logics with Circumscription

Piero Bonatti
Department of Physics
University of Naples
Italy

Carsten Lutz
Institute for Theoretical CS
TU Dresden
Germany

Frank Wolter
Department of CS
University of Liverpool
UK

Abstract

We show that circumscription can be used to extend description logics (DLs) with non-monotonic features in a straightforward and transparent way. In particular, we consider extensions with circumscription of the expressive DLs $\mathcal{ALCIO}$ and $\mathcal{ALCQO}$ and prove that reasoning in these logics is decidable under a simple restriction: only concept names can be circumscribed, and role names vary freely during circumscription. We pinpoint the exact computational complexity of reasoning as complete for $\mathrm{NP}^{\mathrm{NExp}}$ and $\mathrm{NExp}^{\mathrm{NP}}$, depending on whether or not the number of minimized and fixed predicates is assumed to be bounded by a constant. We also show that we cannot allow role names to be fixed during minimization rather than having them vary: this modification renders reasoning undecidable already in the basic DL $\mathcal{ALC}$. Finally, we argue that non-monotonic DLs based on circumscription are an appropriate tool for modelling defeasible inheritance. In particular, we can avoid the restriction of non-monotonic reasoning to domain elements that are named by an individual constant, as adopted by other non-monotonic DLs.

Introduction

Description Logics (DLs) are descendents of frame based systems and semantic networks (Minsky 1975; Quillian 1968). Their main improvement upon these early ancestors is that DLs are equipped with a formal semantics and with well-defined reasoning services that are decidable. To achieve decidability, the expressive means included in a DL have to be chosen carefully: whereas frame based systems and semantic networks followed an all-embracing approach and tried to include all expressive means that appear useful for knowledge representation, DLs establish a careful balance between expressive power and computational complexity of reasoning. For this reason, there are several expressive means that have usually been included in frame based systems, but are not available in standard DLs. One of the most important such omissions is that standard DLs such as $\mathcal{SHIQ}$ (Horrocks, Sattler, & Tobies 2000) do not include any non-monotonic features and, in particular, do not allow to model defeasible inheritance. This has often been conceived as a serious shortcoming. For example, DLs are nowadays a popular tool for the formalization of biomedical ontologies such as GALEN (Rector & Horrocks 1997) and SNOMED (Cote *et al.* 1993), and it has recently been argued by Rector et al. in (Rector 2004; Stevens *et al.* 2005) that 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".

There have been many proposals to extend DLs with non-monotonic features, based e.g. on Reiter's default logic and on epistemic operators (Baader & Hollunder 1995a; 1995b; Straccia 1993; Donini, Nardi, & Rosati 1997; 2002; Donini *et al.* 1998; Padgham & Zhang 1993; Lambrix, Shahmehri, & Wahlloef 1998; Eiter *et al.* 2004; Rosati 2005). However, identifying a non-monotonic DL that is of sufficient expressivity and computationally well-behaved is a non-trivial task. In particular, the resulting formalisms often suffer from one or more of the following problems: (i) they have limitations in expressivity such as treating objects that are named by an individual constant different from unnamed ones; (ii) they are computationally very hard and easily become undecidable; and (iii) they are often conceived as being difficult to understand.

It is surprising that, although circumscription is one of the main approaches to non-monotonic reasoning, DLs based on circumscription have hardly received any attention. After the early (Brewka 1987; Cadoli, Donini, & Schaerf 1990) there appears to be no publication devoted to the subject. *In this paper, we propose non-monotonic description logics based on circumscription and investigate their computational complexity*. We show that circumscription allows to define non-monotonic DLs in a straightforward and transparent way, and that the restrictions that have to be imposed for attaining decidability are quite simple and do not appear to be prohibitive for modelling defeasible inheritance. We also pinpoint the exact computational complexity of our logics, thus innervating the sparse landscape of complexity results for non-monotonic DLs.

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 these predicates is required to be minimal w.r.t. set inclusion. Following McCarthy (McCarthy 1986), the minimized predicates will usually be "abnormality predicates" identifying instances that are not typical for their class. When minimizing predicates, circumscription patterns can require other predicates to be fixed or allow them to vary freely. Circumscription patterns also allow to express preferences between minimized predicates in terms of a partial ordering. As argued in (Baader & Hollunder 1995b), this is important to ensure a smooth interplay between defeasible inheritance and DL subsumption.

Concerning the computational properties of 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 are minimized and fixed, while all role names vary freely. 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 non-monotonic logics. First, the *arity* of predicates has an impact on decidability: allowing 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.

We also indicate how defeasible inheritance can be modelled using concept-circumscribed knowledge bases. Our approach avoids a restriction that is common to many non-monotonic DLs: we do not require that 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. Such a restriction is adopted by non-monotonic DLs based on default logic as introduced, e.g., in (Baader & Hollunder 1995a; 1995b; Straccia 1993; Lambrix, Shahmehri, & Wahlloef 1998). 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 quite significant.

Due to space limitations, we refer to the accompanying technical report (Bonatti, Lutz, & Wolter 2005) for full proofs.

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	$(\geqslant n\ r\ C)$	$\{d \mid \#\{e \in C^\mathcal{I} \mid (d,e) \in r^\mathcal{I}\} \geq n\}$
at-most restriction	$(\leqslant n\ r\ C)$	$\{d \mid \#\{e \in C^\mathcal{I} \mid (d,e) \in r^\mathcal{I}\} \leq n\}$

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

Description Logics

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 and roles 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. For simplicity, we assume that the inverse role constructor is only applied to role names. Thus, a role is either a role name or the inverse of a role name. 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, 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 $(\leqslant 0\ r\ \neg C)$. An example for an $\mathcal{ALCQIO}$ concept description is the following on, describing mammals that are grey and do not live on land:

$$\texttt{Mammal} \sqcap \texttt{Grey} \sqcap \neg\exists\texttt{habitat}.\texttt{Land}.$$

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 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. A *knowledge base (KB)* is a pair $(\mathcal{T}, \mathcal{A})$ consisting of a TBox and an ABox.

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 $\mathcal{T}$, a *model* of an ABox $\mathcal{A}$ if it satisfies all assertions in $\mathcal{A}$, and a model of the knowledge base $(\mathcal{T}, \mathcal{A})$ if it is a model of $\mathcal{T}$ and $\mathcal{A}$. One of the most important reasoning tasks in DLs is subsumption: a concept C is *subsumed by* a concept D w.r.t. a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all models $\mathcal{I}$ of $\mathcal{K}$.

Circumscription

The basic idea of circumscription is to introduce a special sort of predicate that is minimized during reasoning (McCarthy 1986; Lifschitz 1993): instead of admitting all models as in classical logic, we consider only those models where the extension of these predicates is minimal w.r.t. set inclusion. Since circumscription has originally been used in first-order logic and description logics are fragments of first-order logic, circumscription can be readily applied to DLs: let $M \subseteq \mathsf{N_C} \cup \mathsf{N_R}$ be a finite set of *minimized predicates*. We obtain a preference relation $<_M$ on interpretations by setting $\mathcal{I} <_M \mathcal{J}$ if

1. $\Delta^{\mathcal{I}} = \Delta^{\mathcal{J}}$ and, for all $a \in \mathsf{N_I}$, $a^{\mathcal{I}} = a^{\mathcal{J}}$;
2. for all $p \in M$, $p^{\mathcal{I}} \subseteq p^{\mathcal{J}}$;
3. there is a $p \in W$ with $p^{\mathcal{I}} \subset p^{\mathcal{J}}$;
4. for all $p \in (\mathsf{N_C} \cup \mathsf{N_R}) \setminus M$, $p^{\mathcal{I}} = p^{\mathcal{J}}$.

Then C is subsumed by a concept D w.r.t. a KB $(\mathcal{T}, \mathcal{A})$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all models $\mathcal{I}$ of $\mathcal{K}$ that are minimal w.r.t. $<_M$.

Circumscription is well-suited for modelling what *normally* or *typically* holds, and thus admits the representation of defeasible inheritance. The idea is to introduce 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, abnormality predicates are circumscribed. Intuitively, this means that reasoning is done only on models that are "as normal as possible". For example, we can use $\mathcal{ALC}$ syntax to assert that mammals normally inhabitate land, and that whales do not live on land:

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

The upper inclusion states that any mammal not inhabitating land is an abnormal mammal, thus satisfying the abnormality predicate $\mathtt{Ab}_{\mathtt{Mammal}}$. When minimizing this predicate, we obtain the subsumptions

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

However, this approach to modelling abnormality turned out to be too strong for many applications: it is often more natural to allow non-minimized predicates to vary during circumscription, instead of fixing their extension as in Point 4 of the definition of $<_M$ above. Not surprisingly, 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 KB. 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:

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

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` during minimization. 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

$$\mathtt{Whale}(\mathtt{mobydick}).$$

As expected, the result of this change is that $\mathtt{Whale} \doteq \mathtt{Ab}_{\mathtt{Mammal}} \doteq \{\mathtt{mobydick}\}$. 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 in the literature that there is an interplay between subsumption and abnormality predicates (for a discussion in the context of DLs, see (Baader & Hollunder 1995b)). Consider, for example, the TBox in Figure 2. 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 version 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

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

Figure 2: Multiple exceptions and specificity

default rules as proposed by Brewka (Brewka 1994), but different from standard prioritized circumscription which assumes a total ordering (McCarthy 1986; Lifschitz 1985). More information can be found in (Baader & Hollunder 1995b).

We now introduce circumscription patters, which describe 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 mutually disjoint 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 is a* $q \in M$*,* $q \prec p$*, s.t.* $q^{\mathcal{I}} \subset q^{\mathcal{J}}$*,*
4. *there is a* $p \in M$ *s.t.* $p^{\mathcal{I}} \subset p^{\mathcal{J}}$ *and for all* $q \in M$ *with* $q \prec p$*,* $q^{\mathcal{I}} = q^{\mathcal{J}}$*.*

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* $\mathsf{Circ_{CP}}(\mathcal{T}, \mathcal{A})$*, 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* $\mathsf{Circ_{CP}}(\mathcal{T}, \mathcal{A})$ *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 $\mathsf{Circ_{CP}}(\mathcal{T}, \mathcal{A})$ is called a *concept-circumscribed knowledge base (KB)* if CP is a concept circumscription pattern. The main reasoning tasks 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* $\mathsf{Circ_{CP}}(\mathcal{T}, \mathcal{A})$ *if some model* $\mathcal{I}$ *of* $\mathsf{Circ_{CP}}(\mathcal{T}, \mathcal{A})$ *satisfies* $C^{\mathcal{I}} \neq \emptyset$*.*
- *A concept* C *is* subsumed by *a concept* D *w.r.t. a cKB* $\mathsf{Circ_{CP}}(\mathcal{T}, \mathcal{A})$ *(written* $\mathsf{Circ_{CP}}(\mathcal{T}, \mathcal{A}) \models C \sqsubseteq D$*) if* $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ *for all models* $\mathcal{I}$ *of* $\mathsf{Circ_{CP}}(\mathcal{T}, \mathcal{A})$*.*
- *An individual name* a *is an* instance of *a concept* C *w.r.t. a cKB* $\mathsf{Circ_{CP}}(\mathcal{T}, \mathcal{A})$ *(written* $\mathsf{Circ_{CP}}(\mathcal{T}, \mathcal{A}) \models C(a)$*) if* $a^{\mathcal{I}} \in C^{\mathcal{I}}$ *for all models* $\mathcal{I}$ *of* $\mathsf{Circ_{CP}}(\mathcal{T}, \mathcal{A})$*.*

These reasoning problems can be polynomially reduced to one another: first, C is satisfiable w.r.t. $\mathsf{Circ_{CP}}(\mathcal{T}, \mathcal{A})$ iff $\mathsf{Circ_{CP}}(\mathcal{T}, \mathcal{A}) \not\models C \sqsubseteq \bot$, and $\mathsf{Circ_{CP}}(\mathcal{T}, \mathcal{A}) \models C \sqsubseteq D$ iff $C \sqcap \neg D$ is unsatisfiable w.r.t. $\mathsf{Circ_{CP}}(\mathcal{T}, \mathcal{A})$. And second, C is satisfiable w.r.t. $\mathsf{Circ_{CP}}(\mathcal{T}, \mathcal{A})$ iff $\mathsf{Circ_{CP}}(\mathcal{T}, \mathcal{A}) \not\models \neg C(a)$, where a is an individual name not appearing in $\mathcal{T}$ and $\mathcal{A}$; conversely, we have $\mathsf{Circ_{CP}}(\mathcal{T}, \mathcal{A}) \models C(a)$ iff $A \sqcap \neg C$ is unsatisfiable w.r.t. $\mathsf{Circ_{CP'}}(\mathcal{T}, \mathcal{A} \cup \{A(a)\})$, 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 (McCarthy 1986; Lifschitz 1985) and *nested circumscription* (Lifschitz 1995). 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 well-known that minimized concepts and fixed concepts are closely connected (de Kleer & Konolige 1989): using TBoxes, the latter can be simulated by the former. The proof of the following lemma is based on this simulation.

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 $\mathcal{ALCQIO}$.

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, and $|\mathcal{T}|$ and $|\mathcal{A}|$ to denote the size of the TBox $\mathcal{T}$ and ABox $\mathcal{A}$.

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. We only prove Point (i) since Point (ii) is similar. Details can be found in (Bonatti, Lutz, & Wolter 2005). 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. Choose one member from each equivalence class $[d]$ and denote the resulting subset of $\Delta^{\mathcal{I}}$ by Δ'. We define a new interpretation $\mathcal{J}$ as follows:

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

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 $d' \in [d]$ 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:

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

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 $d' \in [d]$ 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}. ❏

It is interesting to note that the proof of Lemma 5 does not go through if role names are minimized 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: a non-deterministic decision procedure for satisfiability w.r.t. concept-circumscribed KBs may 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 $\mathrm{NExp}^{\mathrm{NP}}$, which contains all problems that can be solved by a non-deterministic, exponentially time-bounded Turing machine that has access to an NP oracle. The following inclusions between time-complexity classes are known: $\mathrm{NExp} \subseteq \mathrm{NExp}^{\mathrm{NP}} \subseteq$ 2-ExpTime.

Theorem 6 *In $\mathcal{ALCIO}$ and $\mathcal{ALCQO}$, it is in $\mathrm{NExp}^{\mathrm{NP}}$ 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 $\mathrm{NExp}^{\mathrm{NP}}$. ❏

Due to the mutual reductions between satisfiability, subsumption, and the instance problem, Theorem 6 yields co-$\mathrm{NExp}^{\mathrm{NP}}$ upper bounds for subsumption and the instance problem. We will show in the next section that these upper bounds are tight. However, since $\mathrm{NExp}^{\mathrm{NP}}$ is a relatively large complexity class, it is a natural question whether

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

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 $\mathrm{NP}^{\mathrm{NExp}}$. For readers uninitiated to oracle complexity classes, we remind that $\mathrm{NExp} \subseteq \mathrm{NP}^{\mathrm{NExp}} \subseteq \mathrm{NExp}^{\mathrm{NP}} \subseteq$ 2-ExpTime, and that $\mathrm{NP}^{\mathrm{NExp}}$ is believed to be much less powerful than $\mathrm{NExp}^{\mathrm{NP}}$, see e.g. (Eiter *et al.* 2004).

To prove the $\mathrm{NP}^{\mathrm{NExp}}$ 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

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

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 $\mathrm{NP}^{\mathrm{NExp}}$ 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 (Tobies 2000) 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 (Grädel, Otto, & Rosen 1997), and (ii) satisfiability in C2 is in NExp even if the numbers in counting quantifiers are coded in binary (Pratt-Hartmann 2005).

Theorem 8 *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 (where 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. Its soundness and completeness is shown in (Bonatti, Lutz, & Wolter 2005). ❑

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.

Lower Complexity Bounds

We show that the upper bounds given above 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 9 *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* $\mathsf{CP}_1, \dots, \mathsf{CP}_k$.

Now for the reductions. Concerning Theorem 8, we establish the following matching lower bound.

Theorem 10 *Let* n *be a constant. In* $\mathcal{ALC}$, *it is* $\mathrm{NP}^{\mathrm{NExp}}$-*hard 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* $|F| \leq n$.

The proof is by reduction of the word problem for nondeterministic Turing machines that run in polynomial time and have access to a NExp oracle. We omit details and refer to (Bonatti, Lutz, & Wolter 2005). It follows that satisfiability w.r.t. (bounded) concept-circumscribed KBs is $\mathrm{NP}^{\mathrm{NExp}}$-complete in $\mathcal{ALC}$, $\mathcal{ALCIO}$, and $\mathcal{ALCQO}$.

A matching lower bound for Theorem 6 is obtained by reduction of a succinct version of the problem co-CERT3COL (Eiter, Gottlob, & Mannila 1997). Let us first introduce the regular (non-succinct) version of co-CERT3COL:

Instance of size n: an undirected graph G on the vertices $\{0, 1, \dots, n-1\}$ 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 evaluate to true under t is not 3-colorable.

As shown in (Stewart 1991), co-CERT3COL is complete for $\mathrm{NP}^{\mathrm{NP}}$. To obtain a problem complete for $\mathrm{NExp}^{\mathrm{NP}}$, 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 $\mathrm{NExp}^{\mathrm{NP}}$ (Eiter, Gottlob, & Mannila 1997). More precisely, the succinct version co-CERT3COL$_S$ of co-CERT3COL is obtained by representing the input graph G with nodes $\{0, \dots, 2^n - 1\}$ 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 two variables in the disjunction labelling this edge by generating the numbers $k_1, \dots, k_4$: 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 9, 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 a representation of the graph $t(G)$ obtained from G by including only those edges whose label evaluate 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 four 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 3, 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 $X_0, \dots, X_{n-1}$

$$
\begin{array}{rcll}
\top & \sqsubseteq & \exists \mathsf{aux}.\mathsf{Val} & (1)\\
\mathsf{Val} & \sqsubseteq & \exists \mathsf{nextx}.\top \sqcap \forall \mathsf{nextx}.\mathsf{Val} & (2)\\
\mathsf{Val} & \sqsubseteq & \forall \mathsf{nextx}.(X{+}{+}) \sqcap \forall \mathsf{nextx}.(Y{=}Y) & (3)\\
\mathsf{Val} & \sqsubseteq & \exists \mathsf{nexty}.\top \sqcap \forall \mathsf{nexty}.\mathsf{Val} & (4)\\
\mathsf{Val} & \sqsubseteq & \forall \mathsf{nexty}.(Y{+}{+}) \sqcap \forall \mathsf{nexty}.(X{=}X) & (5)\\
\\
\mathsf{Edge} & \doteq & \mathsf{Val} & (6)\\
\mathsf{Edge} & \sqsubseteq & \exists \mathsf{var1}.\top \sqcap \forall \mathsf{var1}.\mathsf{Val} & (7)\\
\mathsf{Edge} & \sqsubseteq & \forall \mathsf{var1}.(K^{(1)}{=}X) \sqcap \forall \mathsf{var1}.(K^{(2)}{=}Y) & (8)\\
\mathsf{Edge} & \sqsubseteq & \exists \mathsf{var2}.\top \sqcap \forall \mathsf{var2}.\mathsf{Val} & (9)\\
\mathsf{Edge} & \sqsubseteq & \forall \mathsf{var2}.(K^{(3)}{=}X) \sqcap \forall \mathsf{var2}.(K^{(4)}{=}Y) & (10)\\
\\
\mathsf{Edge} & \sqsubseteq & S_1 \rightarrow (\mathsf{Tr}_1 \leftrightarrow \forall \mathsf{var1}.\mathsf{Tr}) & (11)\\
\mathsf{Edge} & \sqsubseteq & \neg S_1 \rightarrow (\neg\mathsf{Tr}_1 \leftrightarrow \forall \mathsf{var1}.\mathsf{Tr}) & (12)\\
\mathsf{Edge} & \sqsubseteq & S_2 \rightarrow (\mathsf{Tr}_2 \leftrightarrow \forall \mathsf{var2}.\mathsf{Tr}) & (13)\\
\mathsf{Edge} & \sqsubseteq & \neg S_2 \rightarrow (\neg\mathsf{Tr}_2 \leftrightarrow \forall \mathsf{var1}.\mathsf{Tr}) & (14)\\
\mathsf{Edge} & \sqsubseteq & \mathsf{Elim} \leftrightarrow (\neg E \sqcup \neg(\mathsf{Tr}_1 \sqcup \mathsf{Tr}_2)) & (15)
\end{array}
$$

Figure 3: The TBox $\mathcal{T}_G$ (partly).

$$
\begin{array}{rcll}
\mathsf{Node} & \doteq & \mathsf{Val} \sqcap (Y=0) & (16)\\
\\
\mathsf{Node} & \sqsubseteq & R \sqcup B \sqcup G & (17)\\
\mathsf{Node} & \sqsubseteq & \neg(R \sqcap B) \sqcap \neg(R \sqcap G) \sqcap \neg(B \sqcap G) & (18)\\
\\
\mathsf{Edge} & \sqsubseteq & \exists \mathsf{col1}.\top \sqcap \forall \mathsf{col1}.\mathsf{Node} \sqcap \forall \mathsf{col1}.(X{=}X) & (19)\\
\mathsf{Edge} & \sqsubseteq & \exists \mathsf{col2}.\top \sqcap \forall \mathsf{col2}.\mathsf{Node} \sqcap \forall \mathsf{col2}.(Y{=}X) & (20)\\
\\
P & \sqsupseteq & \mathsf{Edge} \sqcap \neg\mathsf{Elim} \sqcap \exists \mathsf{col1}.R \sqcap \exists \mathsf{col2}.R & (21)\\
P & \sqsupseteq & \mathsf{Edge} \sqcap \neg\mathsf{Elim} \sqcap \exists \mathsf{col1}.G \sqcap \exists \mathsf{col2}.G & (22)\\
P & \sqsupseteq & \mathsf{Edge} \sqcap \neg\mathsf{Elim} \sqcap \exists \mathsf{col1}.B \sqcap \exists \mathsf{col2}.B & (23)
\end{array}
$$

Figure 4: The TBox $\mathcal{T}'_G$.

at all r-successors agree with the values of $K_0^{(i)}, \ldots, K_{n-1}^{(i)}$ at x. And second, $\forall r.(X{+}{+})$ is an abbreviation for the well-known concept stating that the value of the counter $X_0, \ldots, X_{n-1}$ is incremented when going to r-successors:

$$
\begin{array}{l}
\displaystyle\bigsqcap_{k=0..n-1} \Big(\bigsqcap_{j=0..k-1} X_j \Big) \rightarrow \big((X_k \rightarrow \forall r.\neg X_k) \\
\qquad\qquad\qquad\qquad\qquad\qquad \sqcap\, (\neg X_k \rightarrow \forall r.X_k) \big) \\
\displaystyle\bigsqcap_{k=0..n-1} \Big(\bigsqcup_{j=0..k-1} \neg X_j \Big) \rightarrow \big((X_k \rightarrow \forall r.X_k) \\
\qquad\qquad\qquad\qquad\qquad\qquad \sqcap\, (\neg X_k \rightarrow \forall r.\neg X_k) \big)
\end{array}
$$

The intuition behind $\mathcal{T}_G$ is as follows: Lines (1) to (5) 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 information about the variables V_{ij}. Concerning the variables, each element of $\mathsf{Val}^\mathcal{I}$ where counter X has value i and counter Y has value 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 (6) introduces Edge as another name for Val. We do this since we use the instances of $\mathsf{Val}^\mathcal{I}$ also to store information about the edges of $t(G)$, and we refer to them as instances of Edge for this purpose. Intuitively, an element of $d \in \mathsf{Edge}^\mathcal{I}$ where counter X has value i and counter Y has value 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 3.

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 $b_0, \ldots, b_{2n-1}$ 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 $X_0, \ldots, X_{n-1}, Y_0, \ldots, Y_{n-1}$ at x is described by $b_0, \ldots, b_{n-1}$, the truth value of some output concept name 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 3 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 Lines (7) and (8) ensure the following: each element $d \in \mathsf{Edge}^\mathcal{I}$ representing an edge (i, j) is connected via the role $\mathsf{var1}$ to the element of $\mathsf{Val}^\mathcal{I}$ that represents the variable in the first disjunct of the label of (i, j). Lines (9) and (10) are analogous for the role $\mathsf{var2}$ and the variable in the second disjunct of the edge label. Then, Lines (11) to (15) 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 4. Here, $(X = i)$ stands for the concept expressing that $X_0, \ldots, X_{n-1}$ are the binary encoding of the number i. We use the following strategy for ensuring that the graph $t(G)$ induced by the truth assignment t described by Val does not have a 3-coloring: we use the 2^n elements of $(\mathsf{Val} \sqcap (Y = 0))^\mathcal{I}$ to store the colors of the nodes. By Line (16), these elements are identified by the concept name Node, and there is a unique coloring due to Lines (17) and (18). Then, Line (19) ensures that each element $d \in \mathsf{Edge}^{\mathcal{I}}$ is connected via the role $\mathsf{col1}$ to the element of $\mathsf{Node}^{\mathcal{I}}$ storing the color of the first node of the edge corresponding to d. Line (20) is analogous for the role $\mathsf{col2}$ and the second node of the edge. Lines (21) to (23) 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 $t(G)$. Observe that fixing all concept names except R, G, B also means that, implicitly, 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 $\mathcal{T}'_G$.*

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* NExp^{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 above.

Undecidability

We now slightly generalize concept-circumscribed KBs by allowing role names to be fixed: 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. Interestingly, this seemingly harmless modification leads to undecidability of reasoning.

Theorem 13 *In $\mathcal{ALC}$, satisfiability w.r.t. concept-minimizing cKBs is undecidable. This even holds in the case of empty TBoxes.*

The proof is by a reduction of the semantic consequence problem of modal logic on transitive frames, which has been proved undecidable in (Chagrov 1994). A *frame* is a structure $\mathfrak{F} = (\Delta^{\mathfrak{F}}, r^{\mathfrak{F}})$, 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 $d_0, \ldots, d_m \in \Delta^{\mathfrak{F}}$ with $m \leq n$, $d = d_0$, $e = d_n$, and $d_i \; r^{\mathfrak{F}} d_{i+1}$ for $i < m$. Moreover, $d \in \Delta^{\mathfrak{F}}$ is called a *root* of $\mathfrak{F}$ if for every $e \in \Delta^{\mathfrak{F}}$, there exists n such that $d(r^{\mathfrak{F}})^{\leq n}e$.

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 14 (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 $\forall^{m-1} r.C \sqcap \forall r.\forall^{m-1} r.C$ 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 15 *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) $\mathsf{Circ}_{\mathsf{CP}}(\emptyset, \mathcal{A}) \models a : \forall^m r.C \sqcap \neg D$, *where we define* $\mathsf{CP} = (\emptyset, \{A\}, \{r\}, \emptyset)$ *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 $(*)$, $\mathcal{J}' <_{\mathsf{CP}} \mathcal{I}$. 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 $\mathcal{I}' <_{\mathsf{CP}} \mathcal{I}$ 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 16 *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 (but arbitrary concept names). 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 $(\mathfrak{F}, w) \not\models D$.

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

Proof. (of Theorem 13) Take the concept E from Theorem 14, the concept C_1 from Lemma 16, and set $C := C_1 \sqcap E$. Then, by Theorem 14 and Lemma 16, 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 15, 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 $\mathsf{CP} = (\emptyset, \{A\}, \{r\}, \emptyset)$. ❏

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 Recall that, unlike fixed concept names, fixed role names cannot be simulated using minimized role names. Thus, Theorem 13 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 13, which also applies to the case of empty TBoxes, this result provides evidence that fixing role names can be computationally harder than minimizing role names.

Theorem 17 *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.*

The proof is given in (Bonatti, Lutz, & Wolter 2005) and uses a technique known as *selective filtration* in modal logic.

Conclusions and Perspectives

We have shown that circumscription can be used to define elegant non-monotonic DLs and that reasoning in these DLs is decidable if role names are neither fixed nor minimized. However, we view this paper only as a (promising) first step towards usable non-monotonic DLs. In particular, the following issues deserve further research:

– The upper bounds presented in this paper are based on massive non-deterministic guessing, and thus do not allow for an efficient implementation. Ideally, one would like to have algorithms that are well-behaved extensions of the tableau algorithms that underly state-of-the-art DL reasoners (Baader & Sattler 2000). It seems that existing sequent calculi for (propositional) circumscription and minimal entailment (Bonatti & Olivetti 2002; Olivetti 1992) could provide a starting point.

– It seems necessary to develop a design methodology for modelling defeasible inheritance. The examples given in this paper indicate that the main challenge is to find rules that determine the predicates that should be fixed, varied, and minimized.

– There are many different non-monotonic DLs that have been proposed in the literature, and their exact relationship to the DLs investigated in this paper deserves being studied. For example, it is known that circumscription is slightly less expressive than default logic and autoepistemic logic in the propositional case (Bonatti & Eiter 1996). Still, for certain applications our logics are more powerful than the non-monotonic DLs based on default logic in (Baader & Hollunder 1995a) since the latter impose drastic restrictions on expressivity in order to attain decidability.

Also from a theoretical perspective, our initial investigation leaves open a number of interesting questions of which we mention only two. First, we currently do not know 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 overcome this limitation to obtain non-monotonic extensions of DLs such as $\mathcal{SHIQ}$.

Acknowledgements

The first and the third author were partially supported by the European Network of Excellence REWERSE, IST-2004-506779 and by UK EPSRC grant no. GR/S63182/01, respectively.

References

Baader, F., and Hollunder, B. 1995a. Embedding defaults into terminological knowledge representation formalisms. *J. Autom. Reasoning* 14(1):149–180.

Baader, F., and Hollunder, B. 1995b. Priorities on defaults with prerequisites, and their application in treating specificity in terminological default logic. *J. Autom. Reasoning* 15(1):41–68.

Baader, F., and Sattler, U. 2000. Tableau algorithms for description logics. In *Proc. of Tableaux 2000*, volume 1847 of *LNAI*, 1–18. Springer-Verlag.

Bonatti, P. A., and Eiter, T. 1996. Querying disjunctive databases through nonmonotonic logics. *Theor. Comput. Sci.* 160(1&2):321–363.

Bonatti, P. A., and Olivetti, N. 2002. Sequent calculi for propositional nonmonotonic logics. *ACM Trans. Comput. Log.* 3(2):226–278.

Bonatti, P.; Lutz, C.; and Wolter, F. 2005. Description Logics with Circumscription. LTCS-Report 05-06, TU Dresden. see http://lat.inf.tu-dresden.de/research/reports.html.

Brewka, G. 1987. The logic of inheritance in frame systems. In *Proc. of IJCAI'87*, 483–488.

Brewka, G. 1994. Adding priorities and specificity to default logic. In *Proc. of JELIA'94*, volume 838 of *LNCS*, 247–260. Springer.

Cadoli, M.; Donini, F. M.; and Schaerf, M. 1990. Closed world reasoning in hybrid systems. In *Proc. of ISMIS'90*, 474–481. Elsevier.

Chagrov, A. 1994. Undecidable properties of superintuitionistic logics. In *Mathematical Problems of Cybernetics*, volume 5. Physmatlit. 67 – 108. in Russian.

Cote, R.; Rothwell, D.; Palotay, J.; Beckett, R.; and Brochu, L. 1993. The systematized nomenclature of human and veterinary medicine. SNOMED International, Northfield, IL: College of American Pathologists.

de Kleer, J., and Konolige, K. 1989. Eliminating the fixed predicates from a circumscription. *Artif. Intell.* 39(3):391–398.

Donini, F. M.; Lenzerini, M.; Nardi, D.; Nutt, W.; and Schaerf, A. 1998. An epistemic operator for description logics. *Artif. Intell.* 100(1-2):225–274.

Donini, F. M.; Nardi, D.; and Rosati, R. 1997. Autoepistemic description logics. In *Proc. of IJCAI'97*, 136–141.

Donini, F. M.; Nardi, D.; and Rosati, R. 2002. Description logics of minimal knowledge and negation as failure. *ACM Trans. Comput. Log.* 3(2):177–225.

Eiter, T.; Lukasiewicz, T.; Schindlauer, R.; and Tompits, H. 2004. Combining answer set programming with description logics for the semantic web. In *Proc. of KR'04*, 141–151.

Eiter, T.; Gottlob, G.; and Mannila, H. 1997. Disjunctive Datalog. *ACM Transactions on Database Systems* 22(3):364–418.

Grädel, E.; Otto, M.; and Rosen, E. 1997. Two-Variable Logic with Counting is Decidable. In *Proc. of LICS'97*.

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

Lambrix, P.; Shahmehri, N.; and Wahlloef, N. 1998. A default extension to description logics for use in an intelligent search engine. In *Proc. of HICSS '98*, 28.

Lifschitz, V. 1985. Computing circumscription. In *Proc. of IJCAI'85*, 121–127.

Lifschitz, V. 1993. Circumscription. In Gabbay, D.; Hogger, C.; and Robinson, J., eds., *The Handbook of Logic in AI and Logic Programming 3*. 298–352.

Lifschitz, V. 1995. Nested abnormality theories. *Artif. Intell.* 74(2):351–365.

McCarthy, J. 1986. Applications of circumscription in formalizing common sense knowledge. *Artif. Intell.* 28:89–116.

Minsky, M. 1975. A framework for representating knowledge. In *The Psychology of Computer Vision*. McGraw-Hill. 211–277.

Olivetti, N. 1992. Tableaux and sequent calculus for minimal entailment. *J. of Autom. Reasoning* 9:99–139.

Padgham, L., and Zhang, T. 1993. A terminological logic with defaults: A definition and an application. In *Proc. of IJCAI'93*, 662–668. Morgan Kaufmann.

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

Quillian, M. R. 1968. Semantic memory. In Minsky, M., ed., *Semantic Information Processing*. MIT Press. 227–270.

Rector, A., and Horrocks, I. 1997. Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In *Proc. of Workshop on Ontological Engineering, AAAI'97 Spring Symposium*. AAAI Press.

Rector, A. 2004. Defaults, context, and knowledge: Alternatives for owl-indexed knowledge bases. In *Proc. of the Pacific Symposium on Biocomputing*, 226–237.

Rosati, R. 2005. On the decidability and complexity of integrating ontologies and rules. *J. of Web Semantics*. To appear.

Stevens, R.; Aranguren, M. E.; Wolstencroft, K.; Sattler, U.; Drummond, N.; Horridge, M.; and Rector, A. 2005. Managing OWL's limitations in modelling biomedical knowledge. Submitted.

Stewart, I. A. 1991. Complete problems involving Boolean labelled structures and projection transactions. *J. of Logic and Computation* 1(6):861–882.

Straccia, U. 1993. Default inheritance reasoning in hybrid kl-one-style logics. In *Proc. of IJCAI'93*, 676–681.

Tobies, S. 2000. The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *J. of Artif. Intell. Research* 12:199–217.