In: Proc. Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01), 2001.

Complexity of Nested Circumscription and Abnormality Theories

Marco Cadoli
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, I-00198 Roma, Italy
cadoli@dis.uniroma1.it

Thomas Eiter **Georg Gottlob**
Institut für Informationssysteme
Technische Universität Wien
Favoritenstrasse 9, A-1040 Wien, Austria
eiter@kr.tuwien.ac.at
gottlob@dbai.tuwien.ac.at

Abstract

We propose $\mathcal{L}_{Circ}$, an extension of circumscription by allowing propositional combinations and nesting of circumscriptive theories. As shown, Lifschitz's nested abnormality theories (NATs, introduced in AIJ, 1995) are naturally embedded into this language. We analyze the complexity of $\mathcal{L}_{Circ}$ and NATs, and in particular the effect of nesting. The latter is found a source of complexity, as both formalisms are proved to be PSPACE-complete. We identify cases of lower complexity, including a tractable case. Our results give insight into the "cost" of using $\mathcal{L}_{Circ}$ resp. NATs as a host language for expressing other formalisms, such as narratives.

1 Introduction

Circumscription is a very powerful method for knowledge representation and common-sense reasoning, which has been used for a variety of tasks, including temporal reasoning, diagnosis, and reasoning in inheritance networks. The basic semantical notion underlying circumscription is *minimization* of the extension of selected predicates. This is especially useful when a predicate is meant to represent an *abnormality* condition, e.g., a bird which does not fly. Circumscription is applied to a formula T, either propositional or first-order, and it is used to eliminate some *unintended* models of T.

Since the seminal definition of circumscription in [McCarthy, 1980], several extensions have been proposed (see, e.g., Lifschitz's survey [1994]), all of them retaining the basic idea of minimization. In this paper, we propose $\mathcal{L}_{Circ}$, a language which extends propositional circumscription in two important and rather natural ways:

1. on one hand, we allow the *propositional combination* of circumscriptive theories;
2. on the other hand, we allow *nesting* of circumscriptions.

As for the former extension, we claim that it can be useful in several cases. As an example, when different sources of knowledge $Circ(T_1)$ and $Circ(T_2)$ coming from two equally trustable agents who perform circumscription are to be integrated, it seems plausible to take their disjunction, i.e., $Circ(T_1) \vee Circ(T_2)$[1]. In $\mathcal{L}_{Circ}$, all propositional connectives are allowed.

As for the latter extension, the original idea of *nested abnormality theories* (NATs) has been proposed by Lifschitz [1995], and its usefulness for the formalization of narratives has been shown by Baral *et al.* [1998]. Nesting circumscription is useful for the modularization of a knowledge base. As an example concerning diagnosis of an artifact designed in a modular way, e.g., a car, a piece of knowledge $Circ(T_1)$ may model the intended behavior of a subpart, e.g., the engine. Analogously, $Circ(T_2)$ may model the intended behavior of the electrical part. Further unintended models can be eliminated by taking the circumscription of a suitable propositional combination of $Circ(T_1)$, $Circ(T_2)$, and a plain propositional formula encoding observations.

In this paper, we are mainly concerned with the computational properties of $\mathcal{L}_{Circ}$ and NATs and with their relationship to plain circumscription. In particular, we tackle the following questions:

- Can NATs be embedded into $\mathcal{L}_{Circ}$?
- What is the precise complexity of reasoning under nested circumscription? By *reasoning*, we mean both model checking and formula inference from an $\mathcal{L}_{Circ}$ or NAT theory.
- Is there a simple restriction of NATs (and thus of $\mathcal{L}_{Circ}$) for which some relevant reasoning tasks are feasible in polynomial time?

We are able to give a satisfactory answer to all these questions. In particular, in Section 3, after providing a precise definition of $\mathcal{L}_{Circ}$, we prove the main results about its complexity: model checking and inference are shown to be PSPACE-complete (the latter even for literals); moreover, complexity is proven to increase w.r.t. the nesting. It appears that nesting, and not propositional combination, is responsible for the increase in complexity. Similar results are proven for NATs in Section 4, where we also prove that every NAT can be easily (and polynomially) translated into a formula of $\mathcal{L}_{Circ}$.

Given the high complexity of nested circumscription, we look for significant fragments of the languages in which complexity is lower, and focus on Horn NATs: it is proven in Sec-

[1] We remind that $Circ(T_1) \vee Circ(T_2) \not\equiv Circ(T_1 \vee T_2)$ in general (take, e.g., $T_1 = a \wedge b$, $T_2 = b$).

tion 4 that here nesting can be efficiently eliminated if fixed variables are not allowed, and both model checking and inference are polynomial. Section 5 compares $\mathcal{L}_{Circ}$ and NATs to other generalizations of circumscription such as *prioritized* circumscription [Lifschitz, 1985; 1994] and *theory curbing* [Eiter *et al.*, 1993; Eiter and Gottlob, 2000].

Our results prove that the expressive power that makes $\mathcal{L}_{Circ}$ and NATs useful tools for the modularization of knowledge has indeed a cost, because the complexity of reasoning in such languages is higher than reasoning in a "flat" circumscriptive knowledge base. Anyway the PSPACE upper bound of the complexity of reasoning, and the similarity of their semantics with that of *quantified Boolean formulas* (QBFs), makes fast implementations possible by translating them into a QBF and then using one of the several available solvers, e.g., [Rintanen, 1999]. This approach could be used also for implementing meaningful fragments of NATs, such as the one in [Baral *et al.*, 1998], although this might be inefficient, like using a first-order theorem prover for propositional logic.

Given that QBFs can be polynomially encoded into NATs, we can show (in the full paper) that nested circumscription is more *succinct* than plain (unnested) circumscription, i.e., by nesting *Circ* operators (or NATs), we can express some circumscriptive theories in polynomial space, while they could be written in exponential space only, if nesting were not allowed. In this sense, we add new results to the *comparative linguistics of knowledge representation* [Gogic *et al.*, 1995].

2 Preliminaries

We assume a finite set At of propositional atoms, and let $\mathcal{L}(At)$ (for short, $\mathcal{L}$, if At does not matter or is clear from the context) be a standard propositional language over At. For any formula $\varphi \in \mathcal{L}$, we denote by $mod(\varphi)$ the set of its models. Capitals P, Q, Z etc stand for ordered sets of atoms, which we also view as lists. If $X = \{x_1, \ldots, x_n\}$ and $X' = \{x'_1, \ldots, x'_n\}$, then $X \leq X'$ denotes the formula $\bigwedge_{i=1}^{n}(x_i \rightarrow x'_i)$.

The extension of a model M on atoms A is denoted by $M[A]$. We denote by $\leq_{P;Z}$ the preference relation on models which minimizes P in parallel while Z is varying and all other atoms are fixed; i.e., $M \leq_{P;Z} M'$ (M is more or equally preferable to M') iff $M[P] \subseteq M'[P]$ and $M[Q] = M'[Q]$, where $Q = At \setminus P \cup Z$ and $\subseteq$ and $=$ are taken componentwise. As usual, $M <_{P;Z} M'$ means $M \leq_{P;Z} M' \wedge M \neq M'$.

We denote by $Circ(\varphi; P; Z)$ the second-order circumscription [Lifschitz, 1985] of the formula φ where the atoms in P are minimized, the atoms in Z float, and all other atoms are fixed, which is defined as the following formula:

$$Circ(\varphi; P; Z) = \\ \varphi[P; Z] \wedge \forall P'Z'((\varphi[P'; Z'] \wedge P' \leq P) \rightarrow P \leq P'). \quad (1)$$

Here P' and Z' are lists of fresh atoms (not occurring in φ) corresponding to P and Z, respectively. The second-order formula (1) is a quantified Boolean formula (QBF) with free variables, whose semantics is defined in the standard way. Its models, i.e., assignments to the free variables such that the resulting sentence is valid, are the models M of φ which are $P; Z$*-minimal*, where a model M of φ is $P; Z$-minimal, if no model M' of φ exists such that $M' <_{P;Z} M$.

3 Language $\mathcal{L}_{Circ}$

The language $\mathcal{L}_{Circ}$ extends the standard propositional language $\mathcal{L}$ (over a set of atoms At) by circumscriptive atoms. Formulas of $\mathcal{L}_{Circ}$ are inductively built as follows:

1. $a \in \mathcal{L}_{Circ}$, for every $a \in At$;
2. if φ, ψ are in $\mathcal{L}_{Circ}$, then $\varphi \wedge \psi$ and $\neg\varphi$ are in $\mathcal{L}_{Circ}$;
3. if $\varphi \in \mathcal{L}_{Circ}$ and P, Z are disjoint lists of atoms, then $Circ(\varphi; P; Z)$ is in $\mathcal{L}_{Circ}$ (called *circumscriptive atom*).

Further Boolean connectives ($\vee$, $\rightarrow$, etc) are defined as usual. The semantics of any formula φ from $\mathcal{L}_{Circ}$ is given in terms of models of a naturally associated QBF $\tau(\varphi)$, which is inductively defined as follows:

1. $\tau(a) = a$, if φ is an atom $a \in At$;
2. $\tau(\varphi \wedge \psi) = \tau(\varphi) \wedge \tau(\psi)$ and $\tau(\neg\varphi) = \neg\tau(\varphi)$;
3. $\tau(Circ(\varphi; P; Z)) =$
   $\tau(\varphi[P; Z]) \wedge \forall P'Z'((\tau(\varphi[P'; Z']) \wedge P' \leq P) \rightarrow P \leq P')$.

Note that in 3, the second-order definition of circumscription is used to map the circumscriptive atom to a QBF which generalizes the circumscription formula in (1). In particular, if φ is an "ordinary" propositional formula ($\varphi \in \mathcal{L}$), then $\tau(Circ(\varphi; P; Z))$ coincides with the formula in (1).

Example 1 *Consider the formula*

$$\varphi = Circ(Circ(a \vee b; a; b) \vee Circ(b \vee c; b; c); a; c).$$

Applying rule 3 to inner circumscriptions we get $\tau(\varphi) = Circ((b \wedge \neg a) \vee (c \wedge \neg b); a; c)$. *Applying rule 3 again we get* $\tau(\varphi) = (\neg a \wedge \neg b \wedge c) \vee (\neg a \wedge b) = \neg a \wedge (b \vee c)$.

As usual, we write $M \models \varphi$ if M is a model of φ, and $\varphi \models \psi$ if ψ is a logical consequence of φ.

Eliminating fixed letters

As shown in [de Kleer and Konolige, 1989], the fixed letters can be removed from an ordinary circumscription. The same technique can be applied for formulas from $\mathcal{L}_{Circ}$ as well. More precisely, let $\varphi = Circ(\psi; P; Z)$ be a circumscriptive atom. For each letter $q \notin P \cup Z$, introduce a fresh letter $\bar{q}$, add q,$\bar{q}$ to P, and add a conjunct $q \leftrightarrow \neg\bar{q}$ to ψ. Let $\varphi' = Circ(\psi'; P'; Z)$ be the resulting circumscriptive atom. Then, the following holds.

Proposition 1 φ *and* φ' *are equivalent modulo the set of all auxiliary letters* $\bar{q}$.

Using this equivalence, we can eliminate all fixed letters from a formula $\alpha \in \mathcal{L}_{Circ}$, by replacing each circumscriptive atom φ in α with φ', where the fresh atoms $\bar{q}$ are made floating inside φ' and outside φ. Note that the resulting formula α' has size polynomial in the size of α.

3.1 Complexity results

Let the *Circ-nesting depth* (for short, nesting depth) of $\varphi \in \mathcal{L}_{Circ}$, denoted $nd(\varphi)$, be the maximum number of circumscriptive atoms along any path in the formula tree of φ.

Theorem 1 *Model checking for $\mathcal{L}_{Circ}$, i.e., deciding whether a given interpretation M is a model of a given formula $\varphi \in \mathcal{L}_{Circ}$, is PSPACE-complete. If $nd(\varphi) \leq k$ for a constant $k > 0$, then the problem is complete for $\Delta^P_{k+1}[O(\log n)]$, i.e., polynomial time with $O(\log n)$ many calls to an oracle for Σ^P_k, where n is the input size.*

Proof: (Sketch) Clearly, any formula $\varphi \in \mathcal{L}_{Circ}$ can be encoded, by definition, to a QBF $\tau(\varphi)$ (not necessarily in prenex form) in polynomial time. Thus deciding, $M \models \varphi$ is in PSPACE. By an inductive argument, we can see that for any circumscriptive atom $\varphi = Circ(\psi; P; Z)$ such that $nd(\varphi) \leq k$, deciding $M \models \varphi$ is in Π^P_k. Thus for any Boolean combination φ of atoms and circumscriptive atoms $\varphi_1, \ldots, \varphi_m$ such that $nd(\varphi)$ $(= \max_{i=1}^m nd(\varphi_i))$ $\leq k$, deciding $M \models \varphi$ is possible in $\Delta^P_{k+1}\|$, i.e., polynomial time with one round of parallel Σ^P_k oracle calls. Since $\Delta^P_{k+1}\| = \Delta^P_{k+1}[O(\log n)]$, this proves the membership part for $nd(\varphi) \leq k$.

PSPACE-hardness of $M \models \varphi$ for general φ follows directly from the PSPACE-hardness of model checking for NAT theories shown below in Theorem 4, and from the fact that NATs can be polynomial-time embedded into $\mathcal{L}_{Circ}$ (Corollary 2). Similarly, from the respective results on NATs we obtain that for circumscriptive atoms φ with $nd(\varphi) \leq k$ the problem is Π^P_k-hard (and thus Π^P_k-complete).

The $\Delta^P_{k+1}[O(\log n)]$-hardness part for the case where φ is a Boolean combination of formulas $\varphi_1, \ldots, \varphi_m \in \mathcal{L}_{Circ}$ such that $\max_{i=1}^m nd(\varphi_i) \leq k$ is then shown by a reduction from deciding, given m instances $(M_1, \varphi_1), \ldots, (M_n, \varphi_m)$ of the model checking problem for circumscriptive atoms, whether the number of yes-instances among them is even. The $\Delta^P_{k+1}[O(\log n)]$-completeness of this problem is an instance of Wagner's [1990] general result for all Π^P_k-complete problems. Moreover, we may use that all instances are on disjoint alphabets and the assertion [Wagner, 1990] that (M_i, φ_i) is a yes-instance only if (M_{i+1}, φ_{i+1}) is a yes-instance, for all $i \in \{1, \ldots, m-1\}$. Then, we can define

$$\varphi = e \leftrightarrow \bigvee_{0 \leq 2i \leq m} \left(\bigwedge_{j=0}^{2i} \varphi_j \wedge \bigwedge_{j=2i+1}^{n} \neg\varphi_j \right)$$

where e is a fresh letter. The interpretation $M = \bigcup_{i=1}^m M_i \cup \{e\}$ is a model of φ if and only if the number of yes-instances among $(M_1, \varphi_1), \ldots, (M_n, \varphi_m)$ is even. Clearly, φ and M can be constructed in polynomial time. □

Theorem 2 *Deciding, given formulas $\varphi, \psi \in \mathcal{L}_{Circ}$ whether $\varphi \models \psi$ is* PSPACE*-complete. Hardness holds even if $\psi \in \mathcal{L}$. If the nesting depth of φ and ψ is bounded by the constant $k \geq 0$, then the problem is Π^P_{k+1}-complete.*

Proof: The problem is in PSPACE (resp., Π^P_{k+1}): An interpretation M such that $M \models \varphi \wedge \neg\psi$ can be guessed and verified in polynomial space (resp., $\Delta^P_{k+1}[O(\log n)]$, thus in polynomial time with an oracle for Π^P_k). Hence the problem is in NPSPACE = PSPACE (resp., Π^P_{k+1}). Hardness follows from the polynomial time embedding of NATs into $\mathcal{L}_{Circ}$ (Corollary 2) and Theorem 3. □

Corollary 1 *Deciding satisfiability of a given formula $\varphi \in \mathcal{L}_{Circ}$ is* PSPACE*-complete. If the nesting depth is bounded by a constant $k \geq 0$, then the problem is Σ^P_{k+1}-complete.*

4 Nested Abnormality Theories (NATs)

Lifschitz [1995] proposed the hierarchical use of the circumscription principle, applied to building blocks of a more complex theory. This method was used, e.g., in [Baral *et al.*, 1998] for defining the formal semantics of narratives.

We assume that the atoms At include a set of distinguished atoms $Ab = \{ab_1, \ldots, ab_k\}$ (which intuitively represent abnormality properties). *Blocks* are defined as the smallest set such that if $c_1, \ldots, c_m$ are distinct atoms not in Ab, and each of $B_1, \ldots, B_n$ is either a formula in $\mathcal{L}$ or a block, then

$$B = \{c_1, \ldots, c_m : B_1, \ldots, B_n\},$$

is a block [Lifschitz, 1995], where $c_1, \ldots, c_m$ are called *described* by this block. The *nesting depth* of B, denoted $nd(B)$, is 0 if every B_i is from $\mathcal{L}$, and $1 + \max\{nd(B_i) \mid 1 \leq i \leq n\}$ otherwise.

A *nested abnormality theory* (NAT) [Lifschitz, 1995] is a collection $\mathcal{T} = B_1, \ldots, B_n$ of blocks; its nesting depth, $nd(\mathcal{T})$, is defined as $\max\{nd(B_i) \mid 1 \leq i \leq n\}$.

The semantics of a NAT $\mathcal{T}$ is defined by a mapping $\sigma(\mathcal{T})$ to a QBF as follows: $\sigma(\mathcal{T}) = \bigwedge_{B \in \mathcal{T}} \sigma(B)$, where for any block $B = \{C : B_1, \ldots, B_n\}$,

$$\sigma(B) = \exists Ab.Circ(\textstyle\bigwedge_{i=1}^n \sigma(B_i); Ab; C)$$

given that $\sigma(B_i) = B_i$ if B_i is a formula in $\mathcal{L}$.

Thus, a standard circumscription $Circ(\varphi; P; Z)$, where $\varphi \in \mathcal{L}$, is equivalent to a NAT $\mathcal{T} = \{Z : \varphi\}$ where P is viewed as the set of abnormality letters Ab; notice that $nd(\mathcal{T}) = 0$.

We note the following useful proposition.

Proposition 2 *Let $\mathcal{T} = B_1, \ldots, B_n$ be any NAT. Let $\mathcal{T}' = \{Z : B_1, \ldots, B_n\}$ where Z is any subset of the atoms (disjoint with Ab). Then, $\mathcal{T}$ and $\mathcal{T}'$ have the same models.*

Indeed, $\mathcal{T}'$ has void minimization of Ab (making each ab_j in Ab false), and fixed and floating letters can have any values.

Embedding NATs into $\mathcal{L}_{Circ}$

In the translation $\sigma(\mathcal{T})$, the minimized letters Ab are under an existential quantifier, and thus "projected" from the models of the formula $Circ(\cdots)$. We can, modulo auxiliary letters, eliminate existential quantifiers as follows:

1. Rename every quantifier $\exists Ab$ in $\sigma(\mathcal{T})$ such that every quantified variable is different from any other variable.
2. In every circumscriptive subformula $Circ(\varphi; P; Z)$ of the renamed formula, add to the floating atoms all variables which are quantified in φ (including subformulas).
3. Drop all quantifiers.

Let $\sigma^\star(\mathcal{T})$ be the resulting formula. Note that its size is polynomial (quadratic) in the size of $\sigma(\mathcal{T})$.

Example 2 *Let $AB = \{ab_1, ab_2\}$ and $\mathcal{T} = \{z : \mathcal{T}_1, ab_1 \leftrightarrow z\}$, where $\mathcal{T}_1 = \{z : ab_1 \leftrightarrow \neg ab_2, ab_1 \leftrightarrow z\}$. Then,*

$$\sigma(\mathcal{T}) = \exists ab_1, ab_2.Circ(\sigma(\mathcal{T}_1) \wedge (ab_1 \leftrightarrow z); ab_1, ab_2; z),$$

where

$$\sigma(\mathcal{T}_1) = \exists ab_1, ab_2.Circ((ab_1 \leftrightarrow \neg ab_2) \wedge (ab_1 \leftrightarrow z); ab_1, ab_2; z).$$

In Step 1, we rename ab_1 and ab_2 in $\sigma(\mathcal{T}_1)$ to ab_3 and ab_4, respectively, and add in Step 2 ab_3, ab_4 to the floating letter z of $\mathcal{T}$. After dropping quantifiers in Step 3, we obtain:

$$\sigma^\star(\mathcal{T}) = Circ(\sigma^\star(\mathcal{T}_1) \wedge (ab_1 \leftrightarrow z);\, ab_1, ab_2; z, ab_3, ab_4),$$
$$\sigma^\star(\mathcal{T}_1) = Circ((ab_3 \leftrightarrow \neg ab_4) \wedge (ab_3 \leftrightarrow z);\, ab_3, ab_4; z).$$

Let $A^*(\mathcal{T})$ be the set of atoms from Step 3 of the embedding $\sigma^\star(\mathcal{T})$. The following result, which can be proven by induction on the nesting depth of $\mathcal{T}$, states the correctness of $\sigma^\star$.

Proposition 3 *For any NAT $\mathcal{T}$, the formulas $\sigma(\mathcal{T})$ and $\sigma^\star(\mathcal{T})$ are equivalent modulo $A^*(\mathcal{T})$.*

Corollary 2 *Modulo auxiliary letters, NAT is (semantically) a fragment of $\mathcal{L}_{Circ}$, and polynomial-time embedded via $\sigma^\star$.*

Notice that is not possible to add in Step 2 of the embedding $\sigma^\star(\mathcal{T})$ the quantified variables in φ to the fixed atoms.

Example 3 *Reconsider the NAT $\mathcal{T}$ in Ex. 2. Note that $\emptyset$ is the unique model of $\sigma(\mathcal{T})$. The formula $\sigma^\star(\mathcal{T}_1)$ has, if we disregard ab_1, ab_2 (which are fixed in it), the models $M_1 = \{ab_3, z\}$ and $M_2 = \{ab_4\}$. They give rise to the two models $N_1 = \{ab_1, z, ab_3\}$ and $N_2 = \{ab_4\}$ of $\sigma^\star(\mathcal{T}_1) \wedge (ab_1 \leftrightarrow z)$, of which N_2 is $ab_1, ab_2; ab_3, ab_4, z$-minimal.*

However, if ab_3, ab_4 were fixed in $\sigma^\star(\mathcal{T})$, then both N_1 and N_2 would be models of $\sigma^\star(\mathcal{T})$, as they are $ab_1, ab_2; z$-minimal. Therefore, Proposition 3 would fail.

Eliminating fixed letters

Every fixed letter q can be removed from a NAT $\mathcal{T}$ similarly as from a formula $\varphi \in \mathcal{L}_{Circ}$. However, we must take into account that a fixed letter q may not be simply declared as a minimized letter in the rewriting, since there is a special set Ab of minimized letter which has restricted uses. We surpass this by introducing two special abnormality letters ab_q and $\overline{ab_q}$ in Ab, and by adding the formula $(q \leftrightarrow ab_q) \wedge (ab_q \leftrightarrow \neg\overline{ab_q})$ as a new block to $\mathcal{T}$, a well as $ab_q \leftrightarrow \neg\overline{ab_q}$ in every other block of $\mathcal{T}$. The letter q is declared floating in $\mathcal{T}$.

4.1 Complexity of NATs

Ordinary circumscription can express a QBF sentence $\Phi = \forall X \exists Y \psi$ (where $\psi \in \mathcal{L}$) as follows. Let u be a fresh variable.

Lemma 1 (cf. [Eiter and Gottlob, 1993]) *Φ is true if and only if $Circ(\varphi; u; Y) \models \neg u$, where $\varphi = \psi \vee u$.*

This circumscription can be easily stated as a NAT. Set

$$\mathcal{T}_1 = \{Y, u : \varphi, u \leftrightarrow ab\}.$$

Then Lemma 1 implies that $\mathcal{T}_1 \models \neg u$ iff Φ is true. In fact, denote for any interpretation M and set of atoms S by $t_M(S)$ the assignment to S as given by M. Then, every model M of $\mathcal{T}_1$ must be, if we fix the atoms in X to $t_M(X)$, a model of φ such that $M \models u$ if and only if $\psi[X/t_M(X)]$ is unsatisfiable.

Starting from this result, we prove PSPACE-hardness of inference $\mathcal{T} \models \varphi$ from a NAT theory $\mathcal{T}$. The basic technique is to introduce further variables as *parameters* V into the formula Φ from Lemma 1, which are kept fixed at the inner levels. At a new outermost level to be added, the letter u is used for evaluating the formula at a certain level. We must in alternation minimize and maximize the value of u.

Consider the case of a QBF $\Phi = \forall X \exists Y \psi[V]$, where V are free variables in it, considered as "parameters". We nest $\mathcal{T}_1$ into the following theory $\mathcal{T}_2$:

$$\mathcal{T}_2 = \{X \cup Y, u : \mathcal{T}_1, u \leftrightarrow \neg ab\}$$

This amounts to the following circumscription:

$$\sigma(\mathcal{T}_2) = \exists ab Circ(\exists ab Circ(\varphi \wedge (u \leftrightarrow ab); ab; Y, u) \wedge (u \leftrightarrow \neg ab);\, ab;\, X \cup Y, u).$$

The outer circumscription minimizes ab and thus maximizes u. $\sigma(\mathcal{T}_2)$ is, by Proposition 3, equivalent to the formula

$$\sigma^\star(\mathcal{T}_2) = Circ(\sigma^\star(\mathcal{T}_1) \wedge (u \leftrightarrow \neg a_2); a_2; X \cup Y, u, a_1),$$
$$\sigma^\star(\mathcal{T}_1) = Circ(\varphi \wedge (u \leftrightarrow a_1); a_1; Y, u)$$

modulo the atoms a_1 and a_2. The following can be shown:

Lemma 2 *$\mathcal{T}_2 \models u$ if and only if for every assignment $s(V)$, the QBF $\exists X \forall Y \neg\psi[V/s(V)]$ is true (i.e., $\Phi[V/s(V)]$ is false).*

It follows from the lemma that deciding, given a NAT $\mathcal{T}_2$ of nesting depth 1 and $\psi \in \mathcal{L}$, whether $\mathcal{T}_2 \models \psi$ is Π_3^P-hard.

We generalize this pattern to encode the evaluation of a QBF

$$\Phi = Q_n X_n Q_{n-1} X_{n-1} \cdots \forall X_2 \exists X_1 \psi, \quad n \geq 1, \quad (2)$$

where the quantifiers Q_i alternate, into inference $\mathcal{T} \models \psi$ from a NAT theory $\mathcal{T}$ as follows.

Let $\varphi = \psi \vee u$, where u is a fresh atom. Define inductively

$$\mathcal{T}_1 = \{X_1, u : \varphi, u \leftrightarrow ab\},$$
$$\mathcal{T}_{2k} = \{X_1, \ldots, X_{2k}, u : \mathcal{T}_{2k-1}, u \leftrightarrow \neg ab\},$$
$$\mathcal{T}_{2k+1} = \{X_1, \ldots, X_{2k+1}, u : \mathcal{T}_{2k}, u \leftrightarrow ab\},$$

for all $2k, 2k+1 \in \{2, \ldots, n\}$, and let $\mathcal{T}_0 = \{\ : \varphi\}$. Note that $\mathcal{T}_0$ is equivalent to φ, and that $nd(\mathcal{T}_i) = i - 1$, for all $i \in \{1, \ldots, n\}$ while $nd(\mathcal{T}_0) = 0$. We obtain the following.

Lemma 3 *For every $n \geq 1$, $\mathcal{T}_{n-1}$ has some model, and*

- *if n is odd, then $\mathcal{T}_{n-1} \models u$ iff Φ is false, i.e., $\neg\Phi$ is true;*
- *if n is even, then $\mathcal{T}_{n-1} \models \neg u$ iff Φ is true.*

This statement can be proved by induction on $n \geq 1$.

Lemma 4 *Model checking for NATs, i.e., deciding whether a given interpretation M is a model of a given NAT $\mathcal{T}$, is in PSPACE. If $nd(\mathcal{T}) \leq k$ for constant $k \geq 0$, then it is in Π_{k+1}^P.*

Proof: For $k = 0$, the problem amounts to model checking for ordinary circumscription, which is in coNP. For $k > 0$, we can test recursively for each block B of $\mathcal{T}$ whether M is a model of each block $B_1, \ldots, B_n$ in $B = \{C : B_1, \ldots, B_n\}$, and that no model N of $B_1, \ldots, B_n$ exists such that $N <_{Ab;C} M$. The recursion depth is linear, and the procedure runs in quadratic space. Moreover, if $nd(\mathcal{T})$ is bounded by a constant k, then the check for M and the existence of N witnessing that M is not model of B, can be done in nondeterministic polynomial time with an oracle for Π_k^P. Thus, the problem is in PSPACE (resp., in Π_{k+1}^P). $\Box$

The construction in Lemma 2 shows how it is possible to polynomially embed QBF evaluation into inference wrt a NAT. In turn, Proposition 3 shows that a NAT can be polynomially embedded into an $\mathcal{L}_{Circ}$ formula. The following theorem highlights the consequences of such relations on complexity of inference wrt a NAT.

Theorem 3 *Deciding, given a NAT theory $\mathcal{T}$ and a propositional formula φ, whether $\mathcal{T} \models \varphi$ is* PSPACE-*complete. If $nd(\mathcal{T}) \leq k$ for constant $k \geq 0$, then it is Π^P_{k+2}-complete.*

Proof: The hardness part follows from Lemma 3 above. As for the membership part, a model M of $\mathcal{T}$ such that $M \not\models \varphi$ can be guessed and verified in PSPACE (resp., with the help of a Π^P_{k+1}-oracle). Thus the problem is in co-NPSPACE = PSPACE (resp., Π^P_{k+2}). □

The next theorem shows that the upper bounds on model checking for NATs have matching lower bounds. (This is expected from Theorem 3: if model checking were in the Polynomial Hierarchy (PH), then also inference would be in PH.)

Theorem 4 *Given a NAT theory $\mathcal{T}$ and an interpretation M, deciding whether $M \models \mathcal{T}$ is* PSPACE-*complete. If $nd(\mathcal{T}) \leq k$ for a constant $k \geq 0$, then the problem is Π^P_{k+1}-complete.*

Proof: (Sketch) By Lemma 4, it remains to show the hardness parts. Note that for $k = 0$, this is immediate from results in [Eiter and Gottlob, 1993]. We can slightly adapt the above encoding of QBFs into NATs. Given a QBF Φ as in Eq. (2), let the NATs $\mathcal{T}_0, \ldots, \mathcal{T}_n$ be as there, with the only difference that $\varphi = \psi \vee u$ is replaced by

$$\varphi' = (\psi \vee u \vee (X_n \wedge v)) \wedge ((X_n \wedge v) \rightarrow X_1 \cup \cdots \cup X_{n-1} \cup \epsilon_n),$$

where $\epsilon_n = \{u\}$ if n is odd and $\epsilon_n = \{\neg u\}$ otherwise. Here v is a new letter, defined (i.e., floating) in $\mathcal{T}_n$ and fixed elsewhere; X_n resp. $X_1 \cup \cdots \cup X_{n-1} \cup \epsilon_n$ is viewed as conjunction of its formulas. Let $\mathcal{T}'_0, \ldots, \mathcal{T}'_n$ be the resulting NATs.

Define $M = \bigcup_{i=1}^{n} X_i \cup \{v, u\}$ if n is odd and $M = \bigcup_{i=1}^{n} X_i \cup \{v\}$ if n is even. Note that $M \models \varphi'$ and M is, modulo u, the only candidate for a model of $\mathcal{T}'_n$ in which X_n and v are true: any such model must satisfy φ', and if $X_n \wedge v$ are true, then all other atoms have a unique truth value.

It holds that M is a model of $\mathcal{T}'_n$ iff the QBF Φ is false (resp., true). It follows that model checking is PSPACE-hard and Π^P_{k+1}-hard if $nd(\mathcal{T}) \leq k$ for constant k; note that $nd(\mathcal{T}'_n) = n - 1$. □

4.2 A polynomial-time fragment

We call a block $\{C : B_1, \ldots, B_n\}$ *Horn*, if each B_i is a Horn CNF if $B_i \in \mathcal{L}$, and recursively B_i is Horn otherwise. A NAT $\mathcal{T}$ is *Horn*, if each of its blocks is Horn.

In [Cadoli and Lenzerini, 1994], it was shown that deciding $Circ(\varphi; P; \emptyset) \models \neg u$, where φ is a propositional Horn CNF and u is an atom, is coNP-complete. That is, already for Horn NATs $\mathcal{T}$ without nesting (i.e., $nd(\mathcal{T}) = 0$) inference is intractable, and nesting may further add on complexity.

Horn NATs without fixed letters

The technique of removing fixed letters from a Horn NAT does not work, since it uses non-Horn clauses. Thus, Horn NATs without fixed letters do not immediately inherit complexity, in particular intractability, from general Horn NATs.

In fact, for this class we obtain positive results. Note that here still minimization of atoms q is possible via auxiliary atoms $ab_q \in Ab$ and Horn axioms $q \rightarrow ab_q$, $ab_q \rightarrow q$ in $\mathcal{T}$.

Call any $P; Z$-minimal model of a NAT $\mathcal{T}$ such that $P = At \setminus Ab$ and $Z = \emptyset$ a *minimal model* of $\mathcal{T}$.

Theorem 5 *Let $\mathcal{T}$ be a Horn NAT without fixed letters. Then, (1) $\mathcal{T}$ has a least (i.e., a unique minimal) model, and (2) $\mathcal{T}$ is equivalent to a Horn CNF $\varphi(\mathcal{T})$. Furthermore, both $\varphi(\mathcal{T})$ and $M(\mathcal{T})$ are computable in polynomial time.*

Proof: (Sketch) Let, for any Horn CNF ψ and model $M \subseteq At$, be ψ^M the Horn CNF resulting from ψ after substituting each $ab_j \in Ab$ by $\top$ (true) if $M \models ab_j$ and by $\perp$ (false) if $M \models \neg ab_j$ and subsequent standard simplifications.

Let $\mathcal{T}$ be a single block $B = \{Z : B_1, \ldots, B_n\}$, where $Z = At \setminus Ab$. Define the Horn CNF $\varphi(B)$ recursively by

$$\varphi(B) := \bigwedge_{B_i \in \mathcal{L}} B_i^{M_0} \wedge \bigwedge_{B_i \notin \mathcal{L}} \varphi(B_i),$$

where $M_0 = M_0(B)$ is the least model of the Horn CNF

$$\psi(B) := \bigwedge_{B_i \in \mathcal{L}} B_i \wedge \bigwedge_{B_i \notin \mathcal{L}} \varphi(B_i),$$

and

$$M(B) := M_0[At \setminus Ab] \ \ (= M_0[Z]).$$

Then, by induction on $nd(B) \geq 0$, we can show that (1) $M(B)$ is the least model of B, and (2) $\varphi(B)$ is equivalent to B.

Let, for any formula α, block B, NAT $\mathcal{T}$ etc denote $\|\alpha\|$, $\|B\|$, $\|\mathcal{T}\|$ etc the representation size of the respective object.

Obviously, we can compute $\varphi(B)$ inductively. For the Horn CNFs $\psi(B)$ and $\varphi(B)$, we have $\|\psi(B)\| \leq \|B\|$ and $\|\varphi(B)\| \leq \|B\|$. Of the model M_0, we only need its restriction $M_0[A]$ to the atoms A occurring in B (all other atoms are irrelevant for computing $\varphi(B)$). We can compute $M_0[A]$ from $\psi(B)$ in $O(\|B\|)$ time (recall that the least model of a Horn CNF α is computable in $O(\|\alpha\|)$ time), and $\bigwedge_{B_i \in \mathcal{L}} B_i^{M_0}$ from $M_0[A]$ in $O(\|B\|)$ time. Overall, it follows that for $\mathcal{T} = B$, we can compute both $\varphi(\mathcal{T})$ and its least model $M(\mathcal{T})$ in time $O(\#b(\mathcal{T})\|\mathcal{T}\|)$, where $\#b(\mathcal{T})$ is the number of (recursive) blocks in $\mathcal{T}$, thus in polynomial time.

By Prop. 2, we can replace a multiple block $\mathcal{T} = B_1, \ldots, B_n$ by $\mathcal{T}' = \{At \setminus Ab : B_1, \ldots, B_n\}$, which is Horn and without fixed letters, and obtain analogous results. □

Using sophisticated data structures, the (relevant parts of) models $M_0(B)$ above can be computed incrementally, where each clause in $\varphi(B)$ is fired at most once. Overall, $\varphi(\mathcal{T})$ and $M(\mathcal{T})$ are computable in $O(\|\mathcal{T}\|)$ time. We thus have:

Theorem 6 (Flat Normal Form) *Every Horn NAT $\mathcal{T}$ without fixed letters can be rewritten to an equivalent Horn NAT $\{Z : \varphi\}$ without fixed letters, where $\varphi \in \mathcal{L}$ is a Horn CNF, in $O(\|\mathcal{T}\|)$ time (i.e., in linear time).*

Thus, nesting in Horn NATs without fixed letters does not increase the expressiveness, and can be efficiently eliminated.

We note some easy corollaries of the previous result.

Corollary 3 *Deciding satisfiability of a given Horn NAT $\mathcal{T}$ without fixed letters is polynomial.*

Corollary 4 *Model checking for a given Horn NAT $\mathcal{T}$ without fixed letters and model M is polynomial.*

For the inference problem, we obtain the following result.

Theorem 7 *Given a Horn NAT $\mathcal{T}$ without fixed letters and $\varphi \in \mathcal{L}$, deciding $\mathcal{T} \models \varphi$ is* coNP-*complete. If φ is a CNF, then the problem is polynomial.*

Proof: Since model checking is polynomial, the problem is clearly in coNP. The coNP-hardness part follows from coNP-completeness of validity checking for a given $\varphi \in \mathcal{L}$ (ask whether $\{Z : \top\} \models \varphi$, where Z contains all letters).

We can reduce $\mathcal{T} \models \psi$ to $\varphi(\mathcal{T}) \models \psi$ in $O(\|\mathcal{T}\|)$ time, where $\varphi(\mathcal{T})$ is a Horn CNF. If $\psi = \bigwedge_{i=1}^{m} \alpha_i$ is a CNF of clauses α_i, the latter can be checked in $O(m\|\varphi(\mathcal{T})\| + \|\psi\|)$ time, thus in $O(m\|\mathcal{T}\|+\|\psi\|)$ time (check $\varphi(\mathcal{T}) \models \alpha_i$, which is $O(\|\varphi(\mathcal{T})\| + \|\alpha_i\|)$ time, for all $i \in \{1, \ldots, m\}$). $\Box$

5 Other Generalizations of Circumscription

As we have discussed above, fixed letters can be removed from $\mathcal{L}_{Circ}$ and NAT theories, respectively. Using these techniques, the hardness results of Sections 3.1 and 4.1 can be sharpened to theories without fixed letters.

Prioritized circumscription generalizes circumscription $Circ(\varphi; P; Z)$ by partitioning the letters P into priority levels $P_1 > P_2 > \cdots > P_n$, and informally pruning all models of φ which are not minimal on P_i, while $Z \cup P_{i+1} \cup \cdots \cup P_n$ floats and $P_1 \cup \cdots \cup P_{i-1}$ is fixed, for $i = 1, \ldots, n$ (cf. [Lifschitz, 1994]). This can be readily expressed as the nested circumscription $Circ(\cdots Circ(Circ(\varphi; P_1; Z \cup P_2 \cup \cdots \cup P_n); P_2; Z \cup P_3 \cup \cdots \cup P_n) \cdots)$. This generalization differs from $\mathcal{L}_{Circ}$ and NATs: The complexity does not increase, as inference and model checking remain Π_2^P-complete and coNP-complete, respectively. Intuitively, the reason is that prioritization allows only for a restricted change of the role of the same letter in iterations (floating to minimized and minimized to fixed) and forbids reconsidering minimized letters.

Curbing is yet another extension of circumscription [Eiter *et al.*, 1993; Eiter and Gottlob, 2000]. Rather than the (hierarchical) use of circumscription applied to blocks, curbing aims at softening minimization, and allows for inclusive interpretation of disjunction where ordinary circumscription returns exclusive disjunction. Semantically, $Curb(\varphi; P; Z)$ for a formula $\varphi \in \mathcal{L}$ is the smallest set $\mathcal{M} \subseteq mod(\varphi)$ which contains all models of $Circ(\varphi; P; Z)$ and is closed under minimal upper bounds in $mod(\varphi)$. A *minimal upper bound* (mub) of a set $\mathcal{M}'$ of models in $mod(\varphi)$ is a model $M \in mod(\varphi)$ such that (1) $M' \leq_{P;Z} M$, for every $M' \in \mathcal{M}'$, and (2) there exists no $N \in mod(\varphi)$ satisfying item 1 such that $N <_{P;Z} M'$.

For example, $\varphi = a \vee b$ has two minimal models, if all letters are minimized ($P = At = \{a, b\}$): $\{a\}$ and $\{b\}$. The model $\{a, b\}$ is not a minimal model of φ, but is a curb model of φ. Thus, $Curb(a \vee b) \equiv a \vee b$. On the other hand, if $At = \{a, b, c\}$, then $Curb(a \vee b) \equiv (a \vee b) \wedge \neg c$ is different from $a \vee b$; the unsupported letter c is set to false. Note that $Circ(a \vee b) \equiv (a \leftrightarrow \neg b) \wedge c$ is different from $Curb(a \vee b)$.

Like for $\mathcal{L}_{Circ}$ and NATs, inference and model checking for $Curb(\varphi; P; Z)$ are PSPACE-complete [Eiter and Gottlob, 2000]. However, in the first-order case, the formalisms have different expressiveness. Let us consider theories in a function-free language under *domain closure* (DC) and the *unique names assumption* (UNA), i.e., the (finite) "datalog" case. In this case, curbing can express some problems which $\mathcal{L}_{Circ}$ and NATs can not. We can, e.g., write a (fixed) interpreter T_I in this language for curbing *varying* propositional 3CNF formulas φ, input as ground facts $F(\varphi)$, such that the curb models of $T_I \cup F(\varphi)$ and φ are in 1-1 correspondence. Notice that curbing such 3CNFs φ is PSPACE-complete, and thus, by well-known results in complexity, this is not expressible by a fixed $\mathcal{L}_{Circ}$ or NAT theory (unless PH=PSPACE).

Further relationships between $\mathcal{L}_{Circ}$ resp. NATs and curbing, as well as other expressive KR formalisms [Baral *et al.*, 2000; Pollett and Remmel, 1997], remain to be explored.

Acknowledgments

This work was supported by the Austrian Science Fund (FWF) Project N. Z29-INF.

References

[Baral *et al.*, 1998] C. Baral, A. Gabaldon, and A. Provetti. Formalizing narratives using nested circumscription. *Artificial Intelligence*, 104(1-2):107–164, 1998.

[Baral *et al.*, 2000] C. Baral, V. Kreinovich, and R. Trejo. Computational complexity of planning and approximate planning in the presence of incompleteness. *Artificial Intelligence*, 122(1/2):241–267, 2000.

[Cadoli and Lenzerini, 1994] M. Cadoli and M. Lenzerini. The complexity of propositional closed world reasoning and circumscription. *JCSS*, 43:165–211, 1994.

[de Kleer and Konolige, 1989] J. de Kleer and K. Konolige. Eliminating the fixed predicates from a circumscription. *Artificial Intelligence*, 39:391–398, 1989.

[Eiter and Gottlob, 1993] T. Eiter and G. Gottlob. Propositional circumscription and extended closed world reasoning are Π_2^p-complete. *Theoretical Computer Science*, 114(2):231–245, 1993.

[Eiter and Gottlob, 2000] T. Eiter and G. Gottlob. On the complexity of theory curbing. In *Proc. LPAR-2000*, LNCS 1955, 1–19. 2000.

[Eiter *et al.*, 1993] T. Eiter, G. Gottlob, and Y. Gurevich. Curb your theory! A circumscriptive approach for inclusive interpretation of disjunctive information. In *Proc. IJCAI-93*, 634–639. Morgan Kaufman, 1993.

[Gogic *et al.*, 1995] G. Gogic, H.A. Kautz, Ch.H. Papadimitriou, and B. Selman. The Comparative Linguistics of Knowledge Representation. In *Proc. IJCAI-95*, 862–869.

[Lifschitz, 1985] V. Lifschitz. Computing circumscription. In *Proc. IJCAI-85*, 121–127, 1985.

[Lifschitz, 1994] V. Lifschitz. Circumscription. In D.M. Gabbay, C.J. Hogger, and J.A. Robinson, eds, *Handbook of Logic in AI and Logic Programming*, III, 297-352. 1994.

[Lifschitz, 1995] V. Lifschitz. Nested abnormality theories. *Artificial Intelligence*, 74(2):351–365, 1995.

[McCarthy, 1980] J. McCarthy. Circumscription – A form of non-monotonic reasoning. *Artif. Intell.*, 13:27–39, 1980.

[Pollett and Remmel, 1997] C. Pollett and J. Remmel. Non-monotonic reasoning with quantified boolean constraints. In *Proc. LPNMR'97*, 18–39, LNCS 1265. Springer, 1997.

[Rintanen, 1999] J. Rintanen. Improvements to the evaluation of quantified boolean formulae. In *Proc. IJCAI '99*, 1192–1197. AAAI Press, 1999.

[Wagner, 1990] K. Wagner. Bounded query classes. *SIAM Journal of Computing*, 19(5):833–846, 1990.