I N F S Y S
R E S E A R C H
R E P O R T

INSTITUT FÜR INFORMATIONSSYSTEME

ABTEILUNG WISSENSBASIERTE SYSTEME

COMPLEXITY OF NESTED CIRCUMSCRIPTION AND NESTED ABNORMALITY THEORIES

Marco CADOLI Thomas EITER Georg GOTTLOB

INFSYS RESEARCH REPORT 1843-02-10

JULY 2002

Institut für Informationssysteme
Abtg. Wissensbasierte Systeme
Technische Universität Wien
Favoritenstraße 9-11
A-1040 Wien, Austria
Tel: +43-1-58801-18405
Fax: +43-1-58801-18493
sek@kr.tuwien.ac.at
www.kr.tuwien.ac.at

TECHNISCHE UNIVERSITÄT WIEN

INFSYS RESEARCH REPORT
INFSYS RESEARCH REPORT 1843-02-10, JULY 2002

COMPLEXITY OF NESTED CIRCUMSCRIPTION AND NESTED ABNORMALITY THEORIES

Marco Cadoli [1] Thomas Eiter [2] Georg Gottlob [3]

Abstract. Circumscription has been recognized as an important principle for knowledge representation and common-sense reasoning. The need for a circumscriptive formalism that allows for simple yet elegant modular problem representation has led Lifschitz (AIJ, 1995) to introduce nested abnormality theories (NATs) as a tool for modular knowledge representation, tailored for applying circumscription to minimize exceptional circumstances. Abstracting from this particular objective, we propose $\mathcal{L}_{\mathrm{CIRC}}$, which is an extension of generic propositional circumscription by allowing propositional combinations and nesting of circumscriptive theories. As shown, NATs are naturally embedded into this language, and are in fact of equal expressive capability. We then analyze the complexity of $\mathcal{L}_{\mathrm{CIRC}}$ and NATs, and in particular the effect of nesting. The latter is found to be a source of complexity, which climbs the Polynomial Hierarchy as the nesting depth increases and reaches PSPACE-completeness in the general case. We also identify meaningful syntactic fragments of NATs which have lower complexity. In particular, we show that the generalization of Horn circumscription in the NAT framework remains coNP-complete, and that Horn NATs without fixed letters can be efficiently transformed into an equivalent Horn CNF, which implies polynomial solvability of principal reasoning tasks. Finally, we also study extensions of NATs and briefly address the complexity in the first-order case. Our results give insight into the "cost" of using $\mathcal{L}_{\mathrm{CIRC}}$ (resp. NATs) as a host language for expressing other formalisms such as action theories, narratives, or spatial theories.

Keywords: Circumscription, nested abnormality theories, computational complexity, Horn theories, knowledge representation and reasoning, nonmonotonic reasoning

---

[1]Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Salaria 113, I-00198 Roma, Italy. E-mail: cadoli@dis.uniroma1.it

[2]Institut für Informationssysteme, Technische Universität Wien, Favoritenstraße 9-11, A-1040 Wien, Austria. E-mail: eiter@kr.tuwien.ac.at

[3]Institut für Informationssysteme, Technische Universität Wien, Favoritenstraße 9-11, A-1040 Wien, Austria. E-mail: gottlob@dbai.tuwien.ac.at

Acknowledgements: This work was supported by the Austrian Science Fund (FWF) project Z29-INF.

Some of the results in this paper appeared, in preliminary form, in *Proc. 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, pp. 169–174, Morgan Kaufmann, 2001.

Copyright © 2002 by the authors

Contents

1 Introduction **1**

2 Preliminaries **3**
2.1 Complexity classes 4

3 Language $\mathcal{L}_{\mathrm{CIRC}}$ **4**
3.1 Complexity results 5

4 Nested Abnormality Theories (NATs) **6**
4.1 Embedding NATs into $\mathcal{L}_{\mathrm{CIRC}}$ 8
4.2 Complexity of NATs 10

5 Horn NATs **16**
5.1 Horn NATs without fixed letters 16
5.2 Horn NATs with fixed letters 19

6 Further Issues **21**
6.1 $\mathcal{L}_{\mathrm{CIRC}}$ formulas and NATs without fixed letters 21
6.1.1 Eliminating fixed letters from a $\mathcal{L}_{\mathrm{CIRC}}$ formula 21
6.1.2 Eliminating fixed letters from a NAT 21
6.2 Maximizing and minimizing predicates 22
6.3 First-order case 26

7 Comparison to Other Generalizations of Circumscription **27**
7.1 Prioritized Circumscription 27
7.2 Theory Curbing 28

8 Conclusion **30**

1 Introduction

Circumscription [33, 36, 37] 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 φ, either propositional or first-order, and it is used to eliminate some *unintended* models of φ.

Since the seminal definition of circumscription in [36], several extensions have been proposed (see, e.g., Lifschitz's survey [34]), all of them retaining the basic idea of minimization. In this paper, we propose $\mathcal{L}_{\mathrm{CIRC}}$, a language which extends propositional circumscription in two important and rather natural ways:

- on one hand, we allow the *propositional combination* of circumscriptive theories;
- 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, we consider a scenario from knowledge integration. Suppose that two different sources of knowledge $\mathrm{CIRC}(\varphi_1)$ and $\mathrm{CIRC}(\varphi_2)$, coming from two equally trustable agents who perform circumscription, should be integrated. Then, it seems plausible to take as the result the disjunction of the two sources, i.e., $\mathrm{CIRC}(\varphi_1) \vee \mathrm{CIRC}(\varphi_2)$.[1] In $\mathcal{L}_{\mathrm{CIRC}}$, all propositional connectives are allowed.

As for the latter extension, the concept of *nested abnormality theories* (NATs) has been proposed by Lifschitz [35], in order to enable a hierarchical application of the circumscription principle, which supports modularization of a knowledge base and, as argued, leads sometimes to more economical and elegant formalization of knowledge representation problems. Since then, NATs have been used by a number of authors and are gaining popularity as a circumscriptive knowledge representation tool. For example, NATs have been used in reasoning about actions [25, 24, 31, 32, 48], for handling the qualification problem [39], formalizing narratives [3], expressing function value minimization [2], information filtering [1], describing action selection in planning [47], and in spatial reasoning [43].

As another simple example for combining circumscriptions, imagine the task to diagnose a malfunctioning artifact which is composed of modular components, e.g., a car. A piece of knowledge $\mathrm{CIRC}(\varphi_1)$ may model the behavior of a subpart, e.g., the engine, while another one $\mathrm{CIRC}(\varphi_2)$ may model the behavior of the electrical part, and a plain propositional formula ψ might encode some observations that are being made on the car. Then, by taking the circumscription of a suitable propositional combination of $\mathrm{CIRC}(\varphi_1)$, $\mathrm{CIRC}(\varphi_2)$, and ψ, unintended models for this scenario may be eliminated (see Section 6.2 for a more concrete realization of model-based diagnosis).

In this paper, we are mainly concerned with the computational properties of $\mathcal{L}_{\mathrm{CIRC}}$ and NATs, and with the relationships of these formulas to plain circumscription in this respect. In particular, we tackle the following questions:

- Can NATs be embedded into $\mathcal{L}_{\mathrm{CIRC}}$, i.e., is there an (efficiently) computable mapping from NATs to equivalent $\mathcal{L}_{\mathrm{CIRC}}$ formulas? Here, different interpretations of "equivalence" are possible; a strict one requires that $\mathcal{L}_{\mathrm{CIRC}}$ formulas and NATs are built on the same alphabets, and that their models must coincide. A more liberal one permits the usage of an extended alphabet for $\mathcal{L}_{\mathrm{CIRC}}$ formulas, such that the models of a NAT $\mathcal{T}$ correspond to the projection of the models of its transformation, mapping onto the original alphabet.

[1] We remind that $\mathrm{CIRC}(\varphi_1) \vee \mathrm{CIRC}(\varphi_2) \not\equiv \mathrm{CIRC}(\varphi_1 \vee \varphi_2)$ in general (take, e.g., $\varphi_1 = a \wedge b$ and $\varphi_2 = b$).

• What is the precise complexity of reasoning under nested circumscription? By *reasoning*, we mean both model checking and formula inference from an $\mathcal{L}_{\mathrm{CIRC}}$ formula or a NAT. Note that methods for computing certain NATs, by reduction of circumscription axioms to first-order logic, have been developed; Su [49] implemented a program called CS (Circumscription Simplifier), while Doherty *et al.* came up with their DLS algorithm [16], which has been refined by Gustafsson [28]. However, the precise complexity of NATs was not addressed in these works.

• Is there a simple syntactic restriction of NATs (analogously, of $\mathcal{L}_{\mathrm{CIRC}}$) for which some relevant reasoning tasks are not harder than reasoning in classical logic, or even feasible in polynomial time?

We are able to give a satisfactory answer to all these questions, and obtain the following main results.

(1) After providing a formal definition of $\mathcal{L}_{\mathrm{CIRC}}$, 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.

(2) Similar results are proven for NATs in Section 4. In this section, we also prove that every NAT can be easily (and with polynomial effort) translated into a formula of $\mathcal{L}_{\mathrm{CIRC}}$ using auxiliary letters, and thus NATs can be semantically regarded as a (projective) fragment of $\mathcal{L}_{\mathrm{CIRC}}$. By virtue of the complexity results for NATs, we also provide complexity results for the corresponding syntactic fragment of $\mathcal{L}_{\mathrm{CIRC}}$.

(3) Given the high complexity of nested circumscription, we look for meaningful fragments of the languages in which the complexity is lower. In this paper, we identify Horn NATs, which are a natural generalization of Horn circumscriptions, as such fragments. It is proven in Section 5 that here nesting can be efficiently eliminated if no fixed variables are allowed, and that both model checking and inference are polynomial. In particular, we provide the result that given a Horn NAT $\mathcal{T}$ without fixed letters, an unnested Horn NAT $\mathcal{T}'$ logically equivalent to $\mathcal{T}$ is constructible from $\mathcal{T}$ in time linear in the size of the input.

(4) Furthermore, we show that also for general Horn NATs (i.e., where fixed letters are allowed), model checking is polynomial. Consequently, inference from a Horn NAT is in coNP (and thus, by virtue of results on inference from a Horn circumscription in [9], coNP-complete). This shows that in general, nesting does not add to the complexity of Horn NATs. On the other hand, we show that the use of predicate maximization, proposed in [35] as a convenient declaration primitive, increases the complexity of Horn NATs, which climbs the polynomial hierarchy and reaches PSPACE if the nesting depth is unlimited.

(5) Finally, we compare $\mathcal{L}_{\mathrm{CIRC}}$ and NATs to other generalizations of circumscription, in particular to the well-known method of *prioritized* circumscription [33, 34] and to *theory curbing* [20, 19]. Prioritized circumscription can be modeled in a fragment of $\mathcal{L}_{\mathrm{CIRC}}$, which has the same complexity and expressivity as ordinary (unnested) circumscription. On the other hand, for theory curbing, both model checking and inference are like for $\mathcal{L}_{\mathrm{CIRC}}$ and NATs PSPACE-complete [19]. Our main result of the comparison concerns the *expressiveness* of $\mathcal{L}_{\mathrm{CIRC}}$ and NATs, which appears to be lower than in curbing: in particular, unless some unexpected collapse in complexity classes occurs, there is no fixed $\mathcal{L}_{\mathrm{CIRC}}$ expression that expresses any PSPACE-complete problem, while we present a curb expression of this kind.

As side results, we provide methods for efficiently eliminating fixed letters from $\mathcal{L}_{\mathrm{CIRC}}$ formulas and from NATs, respectively.

Our results prove that the expressive power that makes $\mathcal{L}_{\mathrm{CIRC}}$ 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 prototype implementations possible by translating them into a QBF and then using one of the several available solvers, e.g., [44]. This approach could be used also for implementing meaningful fragments of NATs, such as the one in [3], 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 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* [26].

The rest of this paper is structured as follows. The next section contains some necessary preliminaries and fixes notation. After this, we introduce in Section 3 the language $\mathcal{L}_{\mathrm{CIRC}}$, defining its syntax and semantics, and determine its complexity. In Section 4, we then turn to nested abnormality theories; we show how NATs can be embedded into $\mathcal{L}_{\mathrm{CIRC}}$, and by means of this relationship, we derive the complexity results for the case of general NATs. In the subsequent Section 5, we then focus our attention to the syntactic class of Horn NATs. Section 6 addresses further issues and presents, among others, some results for the first-order case and linguistic extensions to NATs, while Section 7 compares NATs and $\mathcal{L}_{\mathrm{CIRC}}$ to some other generalizations of circumscription, in particular to prioritized circumscription and to curbing. The final Section 8 draws some conclusions and presents open issues for further work.

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. An *interpretation* (or *model*) M is an assignment of truth values 0 (false) or 1 (true) to all atoms. As usually, we identify M also with the set of atoms which are true in M. The projection of a model M on a set of atoms A is denoted by $M[A]$. Furthermore, for any formula φ and model M, we denote by $\varphi[M]$ resp. $\varphi[M[A]]$ the result of substituting in φ for each atom resp. atom from the set A the constant for its truth value.

Satisfaction of a formula φ by an interpretation M, denoted $M \models \varphi$, is defined as usual; we denote by $mod(\varphi)$ the set of all models of φ. 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 $\bigwedge_{i=1}^{n}(x_i \rightarrow x'_i)$.

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, $M <_{P;Z} M'$ stands for $M \leq_{P;Z} M' \wedge M \neq M'$.

We denote by $\mathrm{CIRC}(\varphi; P; Z)$ the second-order circumscription [33] of the formula φ where the atoms in P are minimized, the atoms in Z float, and all other atoms are fixed, defined as the following formula:

$$\mathrm{CIRC}(\varphi; P; Z) = \varphi[P; Z] \wedge \forall P' Z'((\varphi[P'; Z'] \wedge P' \leq P) \rightarrow P \leq P'). \quad (1)$$

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 $M' <_{P;Z} M$.

2.1 Complexity classes

We assume that the reader is familiar with the basic concept and notions of complexity theory, such as P, NP, complete problems and polynomial-time transformations; for a background, see [30, 40]. We shall mainly encounter complexity classes from the Polynomial Hierarchy (PH), which is contained in PSPACE. We recall that $\mathrm{P} = \Sigma_0^P = \Pi_0^P$, $\mathrm{NP} = \Sigma_1^P$, $\mathrm{coNP} = \Pi_1^P$, $\Sigma_{k+1}^P = \mathrm{NP}^{\Sigma_k^P}$, and $\Pi_k^P = \text{co-}\Sigma_k^P$, $k \geq 1$, are major classes in PH. The class $\mathrm{D}_k^{\mathrm{P}} = \{L \times L' \mid L \in \Sigma_k^P, L' \in \Pi_k^P\}$, $k \geq 0$, is the "conjunction" of Σ_k^P and Π_k^P; in particular, $\mathrm{D}_1^{\mathrm{P}}$ is the familiar class D^{P}. All the classes with $k \geq 1$ have complete problems under polynomial-time transformations, and canonical ones in terms of evaluating formulas from certain classes of QBFs. The problems in the class $\Delta_{k+1}^P[O(\log n)]$ are those which can be solved in polynomial time with $O(\log n)$ many calls to an oracle for Σ_k^P, where n is the input size.

A complexity class C is called *closed under polynomial conjunctive reductions*, if the existence of any polynomial-time transformation of problem A into a logical conjunction of (polynomially many) instances of a fixed set of problems $A_1,\ldots,A_l$ in C in implies that A belongs to C. Note that many common complexity classes are closed under polynomial conjunctive reductions. In particular, it is easily seen that this holds for all complexity classes mentioned above.

3 Language $\mathcal{L}_{\mathrm{CIRC}}$

The language $\mathcal{L}_{\mathrm{CIRC}}$ extends the standard propositional language $\mathcal{L}$ (over a set of atoms At) by circumscriptive atoms.

Definition 3.1 *Formulas of $\mathcal{L}_{\mathrm{CIRC}}$ are inductively built as follows:*

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

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

1. $\tau(a) = a$, for any atom $a \in At$;
2. $\tau(\varphi \wedge \psi) = \tau(\varphi) \wedge \tau(\psi)$;
3. $\tau(\neg\varphi) = \neg\tau(\varphi)$; and
4. $\tau(\mathrm{CIRC}(\varphi; P; Z)) = \tau(\varphi[P; Z]) \wedge \forall P' Z'((\tau(\varphi[P'; Z']) \wedge P' \leq P) \rightarrow P \leq P')$.

Note that in 4, 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 $\tau(\mathrm{CIRC}(\varphi; P; Z))$ coincides with the formula in (1). Furthermore, observe that $\mathcal{L}_{\mathrm{CIRC}}$ permits replacement by equivalence, i.e., if ψ_1 and ψ_2 are logically equivalent formulas from $\mathcal{L}_{\mathrm{CIRC}}$ and ψ_1 occurs in formula φ, then any formula resulting from φ by replacing arbitrary occurrences of ψ_1 in φ by ψ_2 is logically equivalent to φ.

Example 3.1 Consider the formula

$$\varphi = \mathrm{CIRC}(\mathrm{CIRC}(a \vee b; a; b) \vee \mathrm{CIRC}(b \vee c; b; c); a; c).$$

Since $\mathrm{CIRC}(a \vee b; a; b) \equiv (b \wedge \neg a)$ and $\mathrm{CIRC}(b \vee c; b; c) \equiv (c \wedge \neg b)$, we get

$$\tau(\varphi) \equiv \mathrm{CIRC}((b \wedge \neg a) \vee (c \wedge \neg b); a; c).$$

From rule 4, we get by applying ordinary circumscription that

$$\tau(\varphi) \equiv (\neg a \wedge \neg b \wedge c) \vee (\neg a \wedge b) \equiv \neg a \wedge (b \vee c).$$

□

As usual, we write $M \models \varphi$ if M is a model of φ (i.e., M satisfies φ), and $\varphi \models \psi$ if ψ is a logical consequence of φ, for any formulas φ and ψ from $\mathcal{L}_{\mathrm{CIRC}}$.

3.1 Complexity results

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

Theorem 3.1 *Model checking for $\mathcal{L}_{\mathrm{CIRC}}$, i.e., deciding whether a given interpretation M is a model of a given formula $\varphi \in \mathcal{L}_{\mathrm{CIRC}}$, is* PSPACE*-complete. If $nd(\varphi) \leq k$ for a constant $k > 0$, then the problem is (i) Π_k^P-complete, if φ is a circumscriptive atom $\mathrm{CIRC}(\psi; P; Z)$, and (ii) $\Delta_{k+1}^P[O(\log n)]$-complete in general.*

Proof: By an inductive argument, we can see that for any circumscriptive atom $\varphi = \mathrm{CIRC}(\psi; P; Z)$ such that $nd(\varphi) \leq k$ for constant k, deciding $M \models \varphi$ is in Π_k^P. Indeed, if $k = 1$, then φ is an ordinary circumscription, for which deciding $M \models \psi$ is well-known to be in $\mathrm{coNP} = \Pi_1^P$, cf. [18]. Assume the statement holds for $k \geq 1$, and consider $k' = k + 1$. Note that $M \not\models \varphi$ iff either (a) $M \not\models \psi$ or (b) some model N exists such that $N <_{P;Z} M$ and $N \models \psi$. By the induction hypothesis, we can guess N and check whether either (a) or (b) holds for this N in polynomial time using a Π_k^P oracle. It follows that deciding $M \models \varphi$ is in Π_{k+1}^P, as claimed. This establishes the membership part for (i). If $nd(\varphi) = k$ but k is not fixed, we obtain similarly that deciding $M \models \varphi$ is possible by a recursive algorithm, whose nesting depth is bounded by $nd(\varphi)$ and which cycles through all possible candidates N for refuting M, in quadratic space. Since any $\mathcal{L}_{\mathrm{CIRC}}$ formula φ is equivalent to the circumscriptive atom $\mathrm{CIRC}(\varphi; \emptyset; \emptyset)$, deciding $M \models \varphi$ is thus in PSPACE in general.

For the membership part of (ii), observe that φ is a Boolean combination of ordinary and circumscriptive atoms $\varphi_1, \ldots, \varphi_m$ such that $nd(\varphi_i)) \leq k$ holds for $i \in \{1, \ldots, m\}$. Deciding $M \models \varphi$ is easy if the values of all φ_i in M are known; by (i), they can be determined in parallel with calls to Π_k^P oracles. Thus, deciding $M \models \varphi$ is possible in $\Delta_{k+1}^P\|$, i.e., in polynomial time with one round of parallel Σ_k^P oracle calls. Since, as well-known, $\Delta_{k+1}^P\| = \Delta_{k+1}^P[O(\log n)]$ (see [50] for $k = 1$, which easily generalizes), this proves the membership part for (ii).

PSPACE-hardness of deciding $M \models \varphi$ for general φ and Π_k^P-hardness for (i) can be shown by a reduction from evaluating suitable prenex QBFs. We exploit that nested abnormality theories (NATs) can be easily embedded into $\mathcal{L}_{\mathrm{CIRC}}$ in polynomial time (cf. Proposition 4.3), and thus a slight adaptation of the reduction of QBFs to model checking for NATs in the proof of Theorem 4.11 proves those hardness results. In particular, we perform the reduction there for empty X_{n+1} (the formulas φ_g and φ_c, which become tautologies, can be removed), and observe that in this case, each auxiliary letter $p \in A^*(\mathcal{T})$ is uniquely defined by some formula $u \leftrightarrow p$ or $u \leftrightarrow \neg p$, respectively, in some $\mathcal{T}_j'$. Thus, the problem $M \models \mathcal{T}_n'$ in the proof of Theorem 4.11 can be reduced, for empty X_{n+1}, in polynomial time to an equivalent model checking problem $M^* \models \sigma^\star(\mathcal{T}_n')$ for $\mathcal{L}_{\mathrm{CIRC}}$. It follows that model checking for $\mathcal{L}_{\mathrm{CIRC}}$ is PSPACE-hard in general and Π_k^P-hard in case (i).

The $\Delta_{k+1}^P[O(\log n)]$-hardness part for the case where φ is a Boolean combination of formulas $\varphi_1,\ldots,\varphi_m$ in $\mathcal{L}_{\mathrm{CIRC}}$ such that $\max\{nd(\varphi_i) \mid i \in \{1,\ldots,m\}\} \leq k$ is then shown by a reduction from the problem of deciding, given m instances $(M_1,\varphi_1),\ldots,(M_n,\varphi_m)$ of the model checking problem for circumscriptive atoms on disjoint alphabets $At_1,\ldots,At_m$, respectively, whether the number of yes-instances among them is even. The $\Delta_{k+1}^P[O(\log n)]$-completeness of this problem is an instance of Wagner's [50] general result for all Π_k^P-complete problems. Moreover, we may assume that m is even and use the assertion (cf. [50]) 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

$$\varphi = e \leftrightarrow \Big(\varphi_1 \vee \bigvee_{0<2i<m} (\neg\varphi_{2i} \wedge \varphi_{2i+1}) \vee \neg\varphi_m\Big),$$

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 $(M_1,\varphi_1),\ldots,(M_m,\varphi_m)$ is even. Clearly, φ and M can be constructed in polynomial time. □

Theorem 3.2 *Deciding, given formulas $\varphi,\psi \in \mathcal{L}_{\mathrm{CIRC}}$ 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_{k+1}^P-complete.*

Proof: The problem is in PSPACE (resp., Π_{k+1}^P): By Theorem 3.1, an interpretation M such that $M \models \varphi \wedge \neg\psi$ can be guessed and verified in polynomial space (resp., in $\Delta_{k+1}^P[O(\log n)]$, thus in polynomial time with an oracle for Π_k^P). Hence the problem is in NPSPACE = PSPACE (resp., in Π_{k+1}^P). Hardness follows from the polynomial time embedding of NATs into $\mathcal{L}_{\mathrm{CIRC}}$ (Corollary 4.4) and Theorem 4.9 below. □

As an immediate corollary, we obtain the following results for the satisfiability in $\mathcal{L}_{\mathrm{CIRC}}$.

Corollary 3.3 *Deciding satisfiability of a given formula $\varphi \in \mathcal{L}_{\mathrm{CIRC}}$ is* PSPACE*-complete. If the nesting depth is bounded by a constant $k \geq 0$, then the problem is Σ_{k+1}^P-complete.*

Observe that some of the hardness proofs in this section make use of results from Section 4. In turn, the membership results for reasoning problems in $\mathcal{L}_{\mathrm{CIRC}}$ will be convenient to establish membership results for some of the problems considered there.

4 Nested Abnormality Theories (NATs)

In this section, we turn to Lifschitz's [35] formalization of nested circumscription, which we introduce here in the propositional setting (see Section 6.3 for the predicate logic context).

We assume that the atoms At include a set of distinguished atoms $Ab = \{ab_1, \ldots, ab_k\}$ (which intuitively represent abnormality properties).

Definition 4.1 Blocks *are defined as the smallest set such that if $c_1, \ldots, c_n$ are distinct atoms not in Ab, and each of $B_1, \ldots, B_m$ is either a formula in $\mathcal{L}$ or a block, then*

$$B = \{c_1, \ldots, c_n : B_1, \ldots, B_m\},$$

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

Definition 4.2 *A* nested abnormality theory *(NAT) is a collection $\mathcal{T} = B_1, \ldots, B_n$ of blocks;*[2] *its nesting depth, denoted $nd(\mathcal{T})$, is defined by $nd(\mathcal{T}) = \max\{nd(B_i) \mid 1 \leq i \leq n\}$.*

Example 4.1 This is a propositional version of the example in section 3.1 of [35]. $\mathcal{T}$ is the following NAT with two blocks:

$$\{f : \quad f \rightarrow ab, \quad B\},$$

where block B is defined as:

$$\{f : \quad b \wedge \neg ab \rightarrow f, \quad c \rightarrow b, \quad c\}.$$

Letters f, b, and c stand for "flies", "bird", and "canary", respectively. The outer block describes the ability of objects to fly; the inner block B gives more specific information about the ability of birds to fly. □

The semantics of a NAT $\mathcal{T}$ is defined by a mapping $\sigma(\mathcal{T})$ to a QBF as follows:

$$\sigma(\mathcal{T}) = \bigwedge_{B \in \mathcal{T}} \sigma(B), \tag{2}$$

where for any block $B = \{C : B_1, \ldots, B_m\}$,

$$\sigma(B) = \exists Ab.\mathrm{CIRC}\Big(\textstyle\bigwedge_{i=1}^{m} \sigma(B_i); Ab; C \Big) \tag{3}$$

given that $\sigma(\varphi) = \varphi$ for any formula $\varphi \in \mathcal{L}$. Satisfaction of a block B (resp., NAT $\mathcal{T}$) in a model M is denoted by $M \models B$ (resp., $M \models \mathcal{T}$).

A standard circumscription $\mathrm{CIRC}(\varphi; P; Z)$, 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 $nd(\mathcal{T}) = 0$. However, in this expression, the letters P are projected from the models of $\mathcal{T}$. Furthermore, any ordinary formula $\varphi \in \mathcal{L}(At \setminus Ab)$ is logically equivalent to the NAT $\{ : \varphi\}$.

Remark 4.1 By our definitions, a model M of a block B comprises all letters, At, including Ab, which is not the case according to [35]. More rigorously, we would need to use abnormality letters as 0-ary predicate (i.e., propositional) variables and distinguish them from the other letters, which are 0-ary predicate constants. For the purpose of this paper, it simplifies the discussion to have models of blocks and NATs on an alphabet which has Ab also constants; our results are not affected by this in essence. Note that M can take any value on Ab for B, since by $\sigma(B)$ as in (3), the valuation of Ab as a variable in $\exists Ab$ is locally defined and projected away via the quantifier.

[2] In [35], the collection may be infinite. For our concerns, only finite collections are of interest.

For later use, we note the following simple characterization of the models of a block.

Proposition 4.1 *Let M be an interpretation of all letters in At and $B = \{C : B_1, \ldots, B_m\}$ a block. Then $M \models B$ if and only if there exists a model M^* which extends $M[At \setminus Ab]$ (i.e., $M[At \setminus Ab] = M^*[At \setminus Ab]$) and is a $(Ab; C)$-minimal model of $B_1, \ldots, B_m$.*

We call any model M^* as in the previous proposition a *witness extension of* M (w.r.t. B); if M is a witness extension of itself (i.e., $M = M^*$), then we call M a *witness model of* B. Thus, M is a witness model of B precisely if $M \models \mathrm{CIRC}\Big(\bigwedge_{i=1}^{n} \sigma(B_i); Ab; C \Big)$ holds.

Example 4.1 (cont.) The semantics $\sigma(\mathcal{T})$ of $\mathcal{T}$ can be easily obtained using the above definition:

$$\begin{aligned}
\sigma(\mathcal{T}) &= \sigma(\{f : f \rightarrow ab\}) \wedge \sigma(\{f : B\}) \\
&= f \rightarrow ab \wedge \exists Ab.\mathrm{CIRC}(b \wedge \neg ab \rightarrow f,\ c \rightarrow b,\ c; ab; f) \\
&= f \rightarrow ab \wedge \exists Ab.\mathrm{CIRC}(c \wedge b \wedge (f \vee ab); ab; f) \\
&= f \rightarrow ab \wedge \exists Ab.(c \wedge b \wedge f \wedge \neg ab) \\
&= f \rightarrow ab \wedge (c \wedge b \wedge f) \\
&= f \wedge ab \wedge c \wedge b.
\end{aligned}$$

Note that $\{c, b, f, \neg ab\}$ is a witness model of B. □

The following useful proposition states that we can easily group multiple blocks into a single one.

Proposition 4.2 *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.

4.1 Embedding NATs into $\mathcal{L}_{\mathrm{CIRC}}$

In the translation $\sigma(\mathcal{T})$, the minimized letters Ab are under an existential quantifier, and thus semantically "projected" from the models of the formula $\mathrm{CIRC}(\cdots)$ (recall that Ab, which is by our convention respected by models of $\sigma(\mathcal{T})$, has arbitrary value in them.) We can, modulo abnormality and auxiliary letters, eliminate the existential quantifiers from the NAT formula $\sigma(\mathcal{T})$ as follows.

Definition 4.3 *Let, for any NAT $\mathcal{T}$, be $\sigma^\star(\mathcal{T})$ the formula obtained from $\sigma(\mathcal{T})$ as follows:*

1. *Rename every quantifier $\exists Ab$ in $\sigma(\mathcal{T})$ such that every quantified variable is different from every other variable.*
2. *In every circumscriptive subformula $\mathrm{CIRC}(\varphi; P; Z)$ of the renamed formula, add to the floating atoms all variables which are quantified in φ (including in its subformulas).*
3. *Drop all quantifiers. Let $A^*(\mathcal{T})$ denote the set of all variables whose quantifier was dropped.*

Note that the size of $\sigma^\star(\mathcal{T})$ is polynomial (more precisely, quadratic) in the size of $\sigma(\mathcal{T})$, and also quadratic in the size of $\mathcal{T}$.

Example 4.2 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,

$$\begin{aligned} \sigma(\mathcal{T}) &= \exists ab_1, ab_2.\mathrm{CIRC}(\sigma(\mathcal{T}_1) \wedge (ab_1 \leftrightarrow z); ab_1, ab_2; z), \text{ where} \\ \sigma(\mathcal{T}_1) &= \exists ab_1, ab_2.\mathrm{CIRC}((ab_1 \leftrightarrow \neg ab_2) \wedge (ab_1 \leftrightarrow z); ab_1, ab_2; z). \end{aligned}$$

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:

$$\begin{aligned} \sigma^\star(\mathcal{T}) &= \mathrm{CIRC}(\sigma^\star(\mathcal{T}_1) \wedge (ab_1 \leftrightarrow z);\ ab_1, ab_2; z, ab_3, ab_4), \\ \sigma^\star(\mathcal{T}_1) &= \mathrm{CIRC}((ab_3 \leftrightarrow \neg ab_4) \wedge (ab_3 \leftrightarrow z);\ ab_3, ab_4; z). \end{aligned}$$

Furthermore, $A^*(\mathcal{T}) = \{ab_1, ab_2, ab_3, ab_4\}$. □

The following result states the correctness of $\sigma^\star$.

Proposition 4.3 *For any NAT $\mathcal{T}$, $\sigma(\mathcal{T})$ and $\sigma^\star(\mathcal{T})$ are logically equivalent modulo $A^*(\mathcal{T})$. Moreover, if $\mathcal{T}$ is a single block B and renaming takes place inside the nesting, then an interpretation M of At is a witness model of $\mathcal{T}$ if and only if $M = N[At]$ for some model N of $\sigma^\star(\mathcal{T})$.*

Proof: We prove the result for any $\mathcal{T}$ which is a single block B by induction on $k \geq 0$ given $nd(\mathcal{T}) \leq k$. The equivalence result for arbitrary $\mathcal{T}$ follows then from Proposition 4.2. In what follows, we use the obvious fact that the models and the witness models of B coincide modulo Ab.

(Basis) If $k = 0$, then $\sigma(\mathcal{T})$ is an ordinary circumscription $\exists Ab.\mathrm{CIRC}(\varphi; Ab; Z)$ where $\varphi \in \mathcal{L}$. Clearly, every witness model M of $\sigma(\mathcal{T})$ (in the alphabet At) is, modulo possible renamings of letters from Ab in $\sigma^\star(\mathcal{T})$, a model of $\sigma^\star(\mathcal{T})$ (in the alphabet $(At \setminus Ab) \cup A^*(\mathcal{T})$), and vice versa. Thus the statements hold in this case.

(Induction) Assume the statements hold for $k \geq 0$. Let $\mathcal{T}$ be a single block $B = \{Z : B_1, \ldots, B_n\}$ of nesting depth $nd(B) = k + 1$. Then, $\sigma(\mathcal{T}) = \exists Ab.\mathrm{CIRC}(\varphi; Ab; Z)$ where $\varphi = \bigwedge_i \sigma(B_i)$. Suppose $\sigma^\star(\mathcal{T}) = \mathrm{CIRC}(\varphi'; Ab'; Z')$ where $\varphi' = \bigwedge_i \sigma^\star(B_i)$, such that, without loss of generality, $Ab' = Ab$ (i.e., renaming in Step 1 of $\sigma^\star(\mathcal{T})$ takes place inside the nesting) and $B_1, \ldots, B_l$ ($l \leq n$) are all the blocks B_i in B such that $B_i \in \mathcal{L}$. Note that $Z' = Z \cup \bigcup_{i=l+1}^n Ab_i$, where Ab_i are the abnormality letters in $\sigma^\star(B_i)$; note that the sets $Ab_{l+1}, \ldots, Ab_n$ and Ab are pairwise disjoint.

Let M be any witness model of B, i.e., $M \models \mathrm{CIRC}(\varphi; Ab; Z)$. We show that $\sigma^\star(\mathcal{T})$ has a model N such that $M = N[At]$. Since $M \models \varphi$, we have $M \models \sigma(B_i)$, for $i \in \{1, \ldots, n\}$. Thus, $M \models \sigma^\star(B_i)$, if $i \leq l$, since $\sigma^\star(B_i) = \sigma(B_i)$ (=B_i). For $i > l$, $\sigma(B_i)$ is of the form $\exists Ab.\mathrm{CIRC}(\varphi_i; Ab; Z_i)$. By the induction hypothesis, there is a truth assignment ν_i to $Ab_i^\star$ such that the extension of M to $Ab_i^\star$ by ν_i is a model of $\sigma^\star(B_i) = \mathrm{CIRC}(\varphi_i'; Ab_i'; Z_i')$. Since the sets $Ab_{l+1}^\star, \ldots, Ab_n^\star$ and Ab are pairwise disjoint, the extension of M to $\bigcup_{i=l+1}^n Ab_i^\star$ by $\nu_{l+1}, \ldots, \nu_n$, denoted N, is therefore a model of φ'. Furthermore, it holds that $N \models \mathrm{CIRC}(\varphi'; Ab'; Z')$. Indeed, assume towards a contradiction that some model N' of φ' exists such that $N' <_{Ab';Z'} N$. Then, projected to the letters of $\sigma^\star(B_i)$, N' is a model of $\sigma^\star(B_i)$, for each $i \in \{1, \ldots, n\}$. The induction hypothesis implies that $M' := N'[At]$ is a model of each $\sigma(B_i)$, and thus $M' \models \varphi$. Since $Ab = Ab'$, we have $M' <_{Ab;Z} M$, and thus M is not an $(Ab; Z)$-minimal model of φ. This contradicts that M is a witness model of B. Consequently, $N \models \mathrm{CIRC}(\varphi'; Ab'; Z')$. Thus, N is a model of $\sigma^\star(\mathcal{T})$ such that $M = N[At]$.

Conversely, let N be a model of $\sigma^\star(\mathcal{T})$. Then, for each $i \in \{1, \ldots, n\}$, the projection of N to the letters for $\sigma^\star(B_i)$, denoted N_i, is a model of $\sigma^\star(B_i)$. Thus, $N_i[At]$ if $i \leq l$, and, as follows from the induction

hypothesis, $N_i[At \setminus Ab_i^\star]$ if $i > l$ is a model of $\sigma(B_i)$. Hence, $M := N[At]$ is a model of φ. Moreover, M is an $(Ab; Z)$-minimal model of φ. Indeed, suppose that $M' <_{Ab;Z} M$ is a smaller model of φ. Since $M' \models \sigma(B_i)$, for $i \in \{1, \ldots, n\}$, we have $M' \models \sigma^\star(B_i)$ if $i \leq l$ and, by the induction hypothesis, there exists an extension N_i' of $M_i'[At \setminus Ab]$ to $Ab_i^\star$ such that $N_i' \models \sigma^\star(B_i)$, for each $i \in \{l+1, \ldots, n\}$. Since the sets $A_{l+1}^\star, \ldots, A_n^\star$ and Ab are pairwise disjoint, $N' = M' \cup \bigcup_{i=l+1}^n N_i$ extends M' to $\bigcup_{i=l+1}^n Ab_i^\star$ such that $N' \models \varphi'$ and $N' <_{Ab;Z'} N$. This implies that N is not a model of $\sigma^\star(\mathcal{T})$, which is a contradiction. This shows that M is an $(Ab; Z)$-minimal model of φ. Consequently, M is a witness model of $\mathcal{T}$.

Thus, the statements hold for $k+1$, which concludes the induction. □

Corollary 4.4 *Modulo the letters $A^*(\mathcal{T})$, NATs are (semantically) a fragment of $\mathcal{L}_{\mathrm{CIRC}}$, and polynomial-time embedded into $\mathcal{L}_{\mathrm{CIRC}}$ via $\sigma^\star$.*

We remark that auxiliary letters seem indispensable for an efficient embedding of NAT into $\mathcal{L}_{\mathrm{CIRC}}$; intuitively, they are needed in compensation for repetitive local use of projected abnormality letters. Notice that it is not possible to add in Step 2 of the embedding $\sigma^\star(\mathcal{T})$ the quantified variables in φ to the fixed atoms. This is shown by the following example.

Example 4.3 Reconsider the NAT $\mathcal{T}$ in Ex. 4.2. 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 4.3 would fail. □

We finally remark that $\mathcal{L}_{\mathrm{CIRC}}$ formulas can be embedded, modulo auxiliary letters, into equivalent NATs in polynomial time. This can be seen from the fact that $\mathcal{L}_{\mathrm{CIRC}}$ formulas can be embedded into QBFs (having free variables) in polynomial time, and that such QBFs can be embedded, using auxiliary letters, into NATs in polynomial time (cf. also the next section). However, by the limited set of constructors in NATs, and in particular the lack of negation applied to blocks, a simple and appealing polynomial-time embedding of $\mathcal{L}_{\mathrm{CIRC}}$ into NATs seems not straightforward.

4.2 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 atom.

Proposition 4.5 (cf. [18]) *Φ is true if and only if $\mathrm{CIRC}(\varphi; u; Y) \models \neg u$, 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 Proposition 4.5 implies that $\mathcal{T}_1 \models \neg u$ iff Φ is true. Recall that $M[S]$ denotes the assignment to the atoms in S as given by M. Then, every model M of $\mathcal{T}_1$ must be, if we fix the atoms in X to their values in M, a model of φ such that $M \models u$ if and only if $\psi[M[X]]$ is unsatisfiable.

Starting from this result, we prove PSPACE-hardness of inference $\mathcal{T} \models \varphi$ from a NAT $\mathcal{T}$. The basic technique is to introduce further variables as *parameters* V into the formula Φ from Proposition 4.5, 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, viewed as "parameters". We nest $\mathcal{T}_1$ into the following theory $\mathcal{T}_2$:

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

This amounts to the following circumscription:

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

The outer circumscription minimizes ab and thus maximizes u. The formula $\sigma(\mathcal{T}_2)$ is, by Proposition 4.3, modulo the atoms a_1 and a_2 equivalent to the formula

$$\begin{aligned} \sigma^\star(\mathcal{T}_2) &= \mathrm{CIRC}(\sigma^\star(\mathcal{T}_1) \wedge (u \leftrightarrow \neg a_2); a_2; X, Y, u, a_1), \\ \sigma^\star(\mathcal{T}_1) &= \mathrm{CIRC}(\varphi \wedge (u \leftrightarrow a_1); a_1; Y, u). \end{aligned}$$

The following holds:

Proposition 4.6 *$\mathcal{T}_2 \models u$ if and only if for every truth assignment ν to V, the QBF $\exists X \forall Y \neg\psi[\nu(V)]$ is true (i.e., $\Phi[\nu(V)]$ is false).*

Proof: ($\Leftarrow$) Suppose $\mathcal{T}_2 \not\models u$. Then, there exists a model M of $\mathcal{T}_2$ such that $M \models \neg u$. Since $M' \models a_2$ holds for any model M' of $\sigma^\star(\mathcal{T}_2)$ which extends M to a_1, a_2, we conclude that every model N of $\sigma^\star(\mathcal{T}_1) \wedge (u \leftrightarrow \neg a_2)$ such that $N[V] = M[V]$ satisfies $N \models a_2 \wedge \neg u$ (otherwise, $N <_{a_2; X \cup Y \cup \{u, a_1\}} M$ would hold, which contradicts that M is a model of $\sigma^\star(\mathcal{T}_2)$). Since $V \cup X$ is fixed in $\mathcal{T}_1$, it is clear that every assignment ν to $(V \cup X)$ which extends $M[V]$ can be completed to a model M_ν of $\sigma^\star(\mathcal{T}_1)$. By minimality of M, we have $M_\nu \models \neg u$, and thus $M \models \psi[\nu(V \cup X)]$. In other words, $\forall X \exists Y \psi[\nu[V]]$ is true, which means that $\forall X \exists Y \neg\psi[\nu(V)]$ is false for $\nu(V) = M[V]$.

($\Rightarrow$) Assume the assignment $\nu(V)$ is such that $\forall X \exists Y \psi[\nu(V)]$ is true. Let M be any model such that $M[V] = \nu(V)$, $M \models \psi$, and $M \models a_2 \wedge \neg u \wedge \neg a_1$. Then M is a model of $\sigma^\star(\mathcal{T}_2)$. Indeed, clearly M is a model of $\sigma^\star(\mathcal{T}_1)$, since $M \models \varphi \wedge (u \leftrightarrow a_1)$ and the minimized letter a_1 is false in M. Furthermore, $M \models u \leftrightarrow \neg a_2$. It remains to show that there is no model N of $\sigma^\star(\mathcal{T}_1) \wedge (u \leftrightarrow \neg a_2)$ such that $N <_{a_2; X \cup Y \{u, a_1\}} M$. Suppose such an N would exist. Then, $N \models u$, and we obtain that $\psi[N[V \cup X]]$ is unsatisfiable. Since $N[V] = M[V] = \nu(V)$, this means that $\forall X \exists Y \psi[\nu(V)]$ is false. This is a contradiction, and thus N can not exist. It follows that M is a model of $\sigma^\star(\mathcal{T}_2)$. Since $M \models \neg u$, the proposition is proved. □

A consequence of the preceding proposition is 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

$$\Phi = Q_n X_n Q_{n-1} X_{n-1} \cdots \forall X_2 \exists X_1 \psi, \quad n \geq 1, \tag{4}$$

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

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

$$\begin{aligned} \mathcal{T}_1 &= \{X_1, u : \varphi, u \leftrightarrow ab\}, \\ \mathcal{T}_{2k} &= \{X_1, \ldots, X_{2k}, u : \mathcal{T}_{2k-1}, u \leftrightarrow \neg ab\}, \quad \text{for all } 2k \in \{2, \ldots, n\}, \\ \mathcal{T}_{2k+1} &= \{X_1, \ldots, X_{2k+1}, u : \mathcal{T}_{2k}, u \leftrightarrow ab\}, \quad \text{for all } 2k+1 \in \{3, \ldots, n\}, \end{aligned}$$

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 4.7 *For every $n \geq 1$ and possible truth assignment $\nu(X_n)$ to X_n, $\mathcal{T}_{n-1}$ has some model extending $\nu(X_n)$, and*

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

Proof: The proof of this statement is by induction on $n \geq 1$. For $n = 1$, clearly $\mathcal{T}_0$ has for each truth assignment $\nu(X_1)$ some model (just assign u value true), and $\mathcal{T}_0 \models u$ if and only if $\Psi = \exists X_1 \psi$ is false. Suppose that the statement holds for $n \geq 1$ and consider $n+1$. Consider any truth assignment $\nu = \nu(X_{n+1})$ to X_{n+1}, and let $\mathcal{T}_j^\nu$ be the NAT $\mathcal{T}_j$ for $\Phi^\nu = \Phi[\nu(X_{n+1})]$, $j \in \{0, \ldots, n\}$. Then, the induction hypothesis implies that $\mathcal{T}_{n-1}^\nu$ has some model, which can be extended to some model of $u \leftrightarrow ab$ (resp., $u \leftrightarrow \neg ab$), and thus to all blocks in $\mathcal{T}_n^\nu$. Since the variables X_{n+1} are fixed in $\mathcal{T}_n$, also $\mathcal{T}_n$ must have a model which extends $\nu(X_{n+1})$. Thus, the first part of the statement holds.

For the second part, assume first that $n + 1$ is odd. Then, n is even, and by the induction hypothesis $\mathcal{T}_{n-1}^\nu \models \neg u$ iff Φ^ν is true. Since $u \leftrightarrow \neg ab$ is a block of $\mathcal{T}_n$ and X_n floats in $\mathcal{T}_n$ (while it is fixed in $\mathcal{T}_{n-1}$), it follows from minimization of ab that every model M of $\mathcal{T}_n$ such that $M[X_{n+1}] = \nu(X_{n+1})$ satisfies u iff $\mathcal{T}_{n-1}^\nu \not\models \neg u$, i.e., Φ^ν is false. Since the letters X_{n+1} are fixed in $\mathcal{T}_n$, it follows that $\mathcal{T}_n \models u$ iff $\neg\Phi^\nu$ is true for all truth assignments $\nu(X_{n+1})$, which is equivalent to Φ being false. Thus the statement holds in this case.

The case where $n + 1$ is even is similar. By the induction hypothesis, $\mathcal{T}_{n-1}^\nu \models u$ iff Φ^ν is false. Since $u \leftrightarrow ab$ is a block of $\mathcal{T}_n$ and X_n floats in $\mathcal{T}_n$ for minimizing ab, every model M of $\mathcal{T}_n$ such that $M[X_{n+1}] = \nu(X_{n+1})$ satisfies $\neg u$ iff $\mathcal{T}_{n-1}^\nu \not\models u$, i.e., Φ^ν is true. Since X_{n+1} is fixed in $\mathcal{T}_n$, it follows that $\mathcal{T}_n \models \neg u$ iff Φ^ν is true for all truth assignments $\nu(X_{n+1})$, i.e., Φ is true. Thus, the statement holds also in this case, which completes the induction step. □

We now turn to the problem of model checking. By our embedding of NATs into nested circumscription, we obtain the following upper bound for this problem.

Lemma 4.8 *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 Σ_{k+2}^P.*

Proof: By Proposition 4.3, $M \models \mathcal{T}$ (thus equivalently, $M \models \sigma(\mathcal{T})$) if and only if there exists some interpretation M^* which extends $M[At \setminus Ab]$ to $A^*(\mathcal{T})$ such that $M^* \models \sigma^\star(\mathcal{T})$. By definition, $\sigma^\star(\mathcal{T}) = \bigwedge_{i=1}^k A_i \wedge \bigwedge_{j=1}^m \varphi_j$ is a conjunction of circumscriptive atoms A_i and ordinary formulas $\varphi_j \in \mathcal{L}((At \setminus Ab) \cup A^*(\mathcal{T}))$. Thus, we can decide $M \models \sigma(\mathcal{T})$ by guessing a proper M^* and check that $M^* \models A_i$ and $M^* \models \varphi_j$, for all A_i and φ_j. We observe that $nd(A_i) \leq k + 1$ holds, since $nd(\sigma(\mathcal{T})) = nd(\sigma^\star(\mathcal{T})) \leq nd(\mathcal{T}) + 1$. Thus, by Theorem 3.1, each $M^* \models A_i$ can be decided by a call to a Π_{k+1}^P oracle; deciding $M^* \models \varphi_j$ is polynomial, for every φ_j.

Since $\sigma^\star(\mathcal{T})$ and $A^*(\mathcal{T})$ are constructible from $\mathcal{T}$ in polynomial time, it follows that deciding $M \models \sigma^\star(\mathcal{T})$, and thus $M \models \mathcal{T}$, is in Σ_{k+2}^P. □

The construction in Lemma 4.7 shows a polynomial-time encoding of QBF evaluation into inference from a NAT. In turn, Proposition 4.3 shows that a NAT can be polynomially embedded into an $\mathcal{L}_{\mathrm{CIRC}}$ formula. The following theorem highlights the consequences of such relations on complexity of inference with respect to a NAT.

Theorem 4.9 *Deciding, given a NAT $\mathcal{T}$ 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_{k+2}^P-complete.*

Proof: The hardness part follows from Lemma 4.7 above. As for the membership part, a model M^* of $\sigma^\star(\mathcal{T})$ such that $M^* \not\models \varphi$ (i.e., $M \not\models \varphi$) can be guessed and verified in PSPACE (resp., with the help of a Π^P_{k+1}-oracle in polynomial time). Thus the problem is in co-NPSPACE = PSPACE (resp., Π^P_{k+2}). □

The complexity of NAT- satisfiability is now an easy corollary to the previous results.

Corollary 4.10 *Deciding whether a given NAT $\mathcal{T}$ is satisfiable is* PSPACE*-complete. If $nd(\mathcal{T}) \leq k$, for constant $k \geq 0$, then the problem is Σ^P_{k+2}-complete.*

The next theorem shows that the upper bounds on model checking for NATs have matching lower bounds. For the general case, this is expected from Theorem 4.9: if model checking would be in PH, then also inference would be in PH. For the case of bounded nestings, it turns out that compared to $\mathcal{L}_{\mathrm{CIRC}}$, the minimization process of NATs has subtle effects on the complexity. In particular, local abnormality letters are a source of complexity and lift the problem, compared to similar $\mathcal{L}_{\mathrm{CIRC}}$ instances, higher up in PH. For example, in case $\mathcal{T}$ is a collection of blocks B_i with nesting depth zero, model checking for $\mathcal{T}$ is Σ^P_2-complete, while for a corresponding conjunction of circumscriptive atoms $\mathrm{CIRC}(\varphi_i; P_i; Z_i)$ where each φ_i is an ordinary formula (having circumscriptive nesting depth 1), model checking is coNP-complete.

Theorem 4.11 *Given a NAT $\mathcal{T}$ 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 Σ^P_{k+2}-complete.*[3]

Proof: By Lemma 4.8, it remains to show the hardness part. To this end, we use an extension of the encoding of a QBF in Lemma 4.7, and construct in polynomial time NATs $\mathcal{T}'_1, \ldots, \mathcal{T}'_n$ and a model M such that $M \models \mathcal{T}'_n$ iff the formula Φ in (4) for $n+1$ is true if n is odd (resp., false if n is even).

Let the NATs $\mathcal{T}_1, \ldots, \mathcal{T}_n$ be similar as there, but with the following differences. Let $X_{n+1} = \{x_{n+1,1}, \ldots, x_{n+1,l}\}$.

- $\varphi = \psi \vee u$ is replaced by φ', where

$$\varphi' = \begin{cases} (\psi \vee u \vee (X_n \wedge v)) \wedge ((X_n \wedge v) \rightarrow u), & \text{if } n \text{ is odd,} \\ (\psi \vee u \vee (X_n \wedge v)), & \text{if } n \text{ is even.} \end{cases}$$

  Here v is a new letter, which is described (i.e., floating) in $\mathcal{T}_n$ and fixed elsewhere.

- We add in $\mathcal{T}_n$ the formulas

$$\varphi_g \;:=\; (X_n \wedge v) \rightarrow \bigwedge_{j=1}^{l} \Big(ab_{n+1,j} \leftrightarrow \neg ab'_{n+1,j} \Big) \quad \text{and}$$

$$\varphi_c \;:=\; \neg(X_n \wedge v) \rightarrow \bigwedge_{j=1}^{l} \Big(x_{n+1,j} \leftrightarrow ab_{n+1,j} \Big),$$

  where $Ab_{n+1} = \{ab_{n+1,1}, \ldots, ab_{n+1,l}\}$ and $Ab'_{n+1} = \{ab'_{n+1,1}, \ldots, ab'_{n+1,l}\}$ are fresh disjoint sets of abnormality letters.

[3] In the preliminary IJCAI '01 conference abstract of this paper, incorrectly Π^P_{k+1}-completeness of the problem was reported. This result applies to a large natural subclass of theories (which we had in mind), in particular, to theories which allow polynomial model completion (see this section).

- We describe (i.e., let float) the letters of X_{n+1} in $\mathcal{T}_n$.

The resulting NATs, denoted $\mathcal{T}'_1, \ldots, \mathcal{T}'_n$, are thus as follows. If $n = 1$, then $\mathcal{T}'_1 = \{X_1, \ldots, X_{n+1}, u, v : \varphi', u \leftrightarrow ab, \varphi_g, \varphi_c\}$; otherwise,

$$\begin{array}{rcll} \mathcal{T}'_1 & = & \{X_1, u : \varphi', \varphi_{u,1}\}, & \\ \mathcal{T}'_j & = & \{X_1, \ldots, X_j, u : \mathcal{T}'_{j-1}, \varphi_{u,j}\}, & \text{for all } j \in \{2, \ldots, n-1\}, \\ \mathcal{T}'_n & = & \{X_1, \ldots, X_n, X_{n+1}, u, v : \mathcal{T}'_{n-1}, \varphi_{u,n}, \varphi_g, \varphi_c\}, & \end{array}$$

where $\varphi_{u,j} = u \leftrightarrow ab$, if j is odd, and $\varphi_{u,j} = u \leftrightarrow \neg ab$, if j is even. Note that $nd(\mathcal{T}'_n) = n - 1$.

The intuition behind these modifications is as follows. Informally, $X_n \wedge v$ will be true in a designated candidate model M, which enforces that the value of u is true if n is odd (resp., false, if n is even by minimization of ab in $\mathcal{T}'_1$). The candidate model M can only be eliminated by some other model which does not satisfy $X_n \wedge v$, and thus must satisfy ψ.

Informally, φ_g serves for guessing a truth assignment ν to the letters in Ab_{n+1} for extending the designated model M to a witness M^* for $M \models \mathcal{T}'_n$. The assignment ν is transfered by φ_c to X_{n+1} when the minimality of M^* is checked; for that, it is assured that any possible smaller model $M' <_{Ab;At \setminus Ab} M^*$ of the blocks in $\mathcal{T}'_n$ must falsify the conjunction $X_n \wedge v$.

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 $M \models \varphi'$.

We claim that $M \models \mathcal{T}'_n$ iff the QBF Φ in (4) for $n+1$ is false if n is odd (resp., true, if n is even). Since M and $\mathcal{T}'_n$ are constructible in polynomial time, this will prove the result.

We use the following lemmas:

Lemma A. Let M^* be any extension of M such that $M^* \models \varphi_{u,n}$ and $M^* \models ab_{n+1,j} \leftrightarrow \neg ab'_{n+1,j}$, for all $j \in \{1, \ldots, l\}$. Then $M^* \models \mathcal{T}'_{n-1}$ and $M^* \models ab$.

Proof: Note that v and the letters in $X_n \cup X_{n+1}$ are fixed in $\mathcal{T}'_1, \ldots, \mathcal{T}'_{n-1}$. Thus, if any model M' such that $M' \models \mathcal{T}'_1$ coincides with M on $X_n \cup \{v\}$, then it follows $M' \models u$ if n is odd (resp., $M' \models \neg u$, if n is even). Next, all models M' of $\mathcal{T}'_2$ which coincide with M on $X_n \cup \{v\}$ satisfy $M' \models u$ (resp., $M' \models \neg u$). Continuing this argument, it follows that $M'[X_n \cup \{v\}] = M[X_n \cup \{v\}]$ and $M' \models \mathcal{T}'_{n-1}$ implies that $M' \models u$ (resp., $M' \models \neg u$). Hence, $M^* \models \mathcal{T}'_{n-1}$. Clearly $M^* \models ab$ holds.

Lemma B. Let M' be any model such that $M' \models \varphi_{u,n} \wedge \neg ab$ and $M' \not\models X_n \wedge v$. Then, $M' \models \mathcal{T}'_{n-1}$ iff $\exists X_n \forall X_{n-1} \cdots \exists X_1 \psi[M'[X_{n+1}]]$ is true if n is odd, and $\forall X_n \exists X_{n-1} \cdots \exists X_1 \psi[M'[X_{n+1}]]$ is false if n is even.

Proof: For any such M', the problem $M' \models \mathcal{T}'_{n-1}$ is equivalent to $M' \models \mathcal{T}'_{n-1}$ (with the letters X_{n+1} fixed to their values in M'), since $X_n \wedge v$ in $\mathcal{T}'_1$ is false; the new abnormality letters introduced above are irrelevant for $M' \models \mathcal{T}_{n-1}$. Note that $M' \models \neg u$ if n is odd (resp., $M' \models u$ if n is even). Lemma 4.7 implies that $M' \models \mathcal{T}_{n-1}$ iff $Q_n X_n Q_{n-1} X_{n-1} \cdots \exists X_1 \psi[M'[X_{n+1}]]$ is true if n is odd (resp., false if n is even). This proves the lemma.

We now prove the claim.

($\Leftarrow$) Suppose $M \not\models \mathcal{T}'_n$. Then, for each extension M^* of M as in Lemma A, there exists some model $M' <_{Ab;At \setminus Ab} M^*$ of the blocks in $\mathcal{T}'_n$ such that $M' \models \neg ab$, which implies that $M' \not\models X_n \wedge v$. Hence, by Lemma B, it follows that $Q_n X_n \forall X_{n-1} \cdots \exists X_1 \psi[M'[X_{n+1}]]$ is true if n is odd (resp., false if n is even). Since the different M^* induce all possible truth assignments to X_{n+1}, and M' was arbitrary, it follows that the formula $Q_{n+1} X_{n+1} Q_n X_n \cdots \exists X_1 \psi$ is true if n is odd (resp., false if n is even).

($\Rightarrow$) Suppose that $M \models \mathcal{T}_n$. Hence, by Proposition 4.1 there exists a witness extension M^* of M w.r.t. $\mathcal{T}_n$ which $(Ab; At \setminus Ab)$-minimally satisfies the blocks in $\mathcal{T}_n$. Thus, for each model $M' <_{Ab;At \setminus Ab} M^*$ which coincides with M on $Ab_{n+1} \cup Ab'_{n+1}$ and such that $M' \models \neg ab$ and $M' \not\models X_n \wedge v$, it follows that

$M' \not\models \mathcal{T}'_{n-1}$. By Lemma B, it follows that $Q_n X_n \cdots \exists X_1 \psi[M'[X_{n+1}]]$ is false if n is odd (resp., true if n is even), and thus $\exists X_{n+1} \neg(Q_n X_n \cdots \exists X_1 \psi[M'[X_{n+1}]])$ is true if n is odd (resp., false if n is even). Rewritten to prenex form, this is means that the QBF in (4) for $n+1$ is false if n is odd (resp., true if n is even). □

We note that in the proof of the previous result, the fact that abnormality letters are local to a NAT block plays an important role for the complexity of model checking. The precise extension of these letters is a priori unknown; an exponential search space may need to be explored to find a suitable extension which satisfies the propositional formulas in a block. By eliminating this source of complexity, model checking becomes easier. This motivates the following concept.

Definition 4.4 *We say that a block $B = \{C : B_1, \ldots, B_m\}$ allows* polynomial model completion *if, given any model M of B, a model M^* is computable in polynomial time (as a function $f(M, B)$ of M and B), such that M^* is a witness extension of M w.r.t. B if $M \models B$. A NAT $\mathcal{T} = B_1, \ldots, B_n$ allows* polynomial model completion*, if each block B_i allows polynomial model completion.*

Note that in general, assessing whether a block allows polynomial model completion is a hard (intractable) problem. There are some important cases, though, where this can be ensured. Namely, if the abnormality letters ab are used to minimize or maximize other letters p, to which they are connected e.g. by equivalences $ab \leftrightarrow p$ or inequivalences $ab \leftrightarrow \neg p$, respectively (this will be further explored in Section 6.2). We obtain the following result.

Theorem 4.12 *Let $B = \{C : B_1, \ldots, B_m\}$ be any block that allows polynomial model completion such that $nd(B) \leq k$, for constant $k \geq 0$. Then, model checking $M \models B$ is in $\mathrm{D}^{\mathrm{P}}_{k+1}$. Moreover, if each $B_i \notin \mathcal{L}$ allows polynomial model completion, then deciding $M \models B$ is in Π^P_{k+1}.*

Proof: Given M, by hypothesis and Proposition 4.1, we can complete it in polynomial time to a model M^* such that $M \models B$ iff M^* is a $(Ab; C)$-minimal model of $B_1, \ldots, B_m$. By Lemma 4.8, each test $M \models B_i$ is in Σ^P_{k+1}. Furthermore, deciding whether some model $M' <_{Ab;C} M^*$ exists such that $M' \models B_i$, for $i = 1, \ldots, m$ is in Σ^P_{k+1}; we can guess such an M' and for every $i \in \{1, \ldots, m\}$ a polynomial-size "proof" for $M' \models B_i$ which can be checked with the help of a Π^P_k oracle in polynomial time. Thus, deciding $M \models B$ is reducible in polynomial time to a conjunction of problems in Σ^P_{k+1} and Π^P_{k+1}. Since these problems are in $\mathrm{D}^{\mathrm{P}}_{k+1}$ and this class is closed under polynomial conjunctive reductions, it follows that deciding $M \models B$ is in $\mathrm{D}^{\mathrm{P}}_{k+1}$. If each $B_i \notin \mathcal{L}$ allows polynomial model completion, then by what we already showed deciding $M \models B_i$ is in $\mathrm{D}^{\mathrm{P}}_k$ for every $i = 1, \ldots, m$. Thus, deciding whether no model $M' <_{Ab;C} M^*$ exists such that $M' \models B_i$ for all $i = 1, \ldots, m$ is in Π^P_{k+1}, which means that $M \models B$ is reducible in polynomial time to a conjunction of problems in Π^P_{k+1}. Since Π^P_{k+1} is closed under polynomial conjunctive reductions, it follows that deciding $M \models B$ is in Π^P_{k+1}. □

This membership result clearly generalizes from a single block to NATs $\mathcal{T} = B_1, \ldots, B_n$ comprising multiple blocks, where each block B_i is as B in the statement of Theorem 4.12. We remark that these upper bounds are actually sharp, i.e., have matching lower bounds, but omit a proof of this; for the case of nested polynomial model completion, a proof of Π^P_{k+1}-hardness is subsumed by the reduction in Theorem 4.11, if we take X_{n+1} to be empty (and thus can eliminate the formulas φ_g and φ_c there).

We note that Theorem 4.12 also shows that the construction in the proof of Theorem 4.11 uses abnormality letters which are hard to complete in the right place (thus revealing the source of complexity), namely

in the outermost block (and nowhere else). Indeed, moving them elsewhere would lead to a decrease in complexity and the reduction would fail.

A final observation is that in the proof of the hardness part of Theorem 4.9, the NATs $\mathcal{T}_j$ constructed allow polynomial model completion. Thus, different from the case of model checking, this property does not lower the complexity of inference from NATs.

5 Horn NATs

In this section, we consider a restricted class of NATs, which generalizes Horn theories. Notice that Horn theories are an important class of theories in knowledge representation, and the application of the circumscription principle to Horn theories is underlying the semantics of several logic programming languages, as well as expressive database languages such as DATALOG^{Circ} [10].

Recall that a clause is Horn, if it contain at most one positive literal.

Definition 5.1 *We call a block* $\{C : B_1, \ldots, B_n\}$ Horn*, if each* B_i *is a Horn CNF (i.e., a conjunction of Horn clauses) 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.*

Example 5.1 NAT $\mathcal{T}$ in Example 4.1 is not Horn, because block B contains non-Horn formula $b \wedge \neg ab \rightarrow f$. However, if we define block B' as:

$$\{f : \quad b \rightarrow f, \quad c \rightarrow b, \quad c\},$$

then NAT $\mathcal{T}'$, defined as:

$$\{f : \quad f \rightarrow ab, \quad B'\},$$

is indeed Horn. We can regard B' as a "simplified" theory in which a bird always flies. □

As for the complexity, it was shown in [9] that deciding $\mathrm{CIRC}(\varphi; P; \emptyset) \models \neg u$, where φ is a propositional Horn CNF and u is an atom, is coNP-complete. As a consequence, already for Horn NATs $\mathcal{T}$ without nesting (i.e., $nd(\mathcal{T}) = 0$), inference is intractable.

We thus address the following two questions: Firstly, are there cases under which (arbitrarily nested) Horn NATs are tractable, and secondly, does nesting increase the complexity of Horn NATs? In the following subsection, we show that Horn NATs without fixed letters are tractable, and that, fortunately, nesting does not increase the complexity of Horn NATs. The latter result is not immediate and has some implications for rewriting NATs, as will be discussed in Section 6.1.

5.1 Horn NATs without fixed letters

In this subsection, we consider the fragment of Horn NATs in which no fixed letters are allowed. That is, each letter p except the special abnormality letters must be described in any block. Note that in this fragment minimization of letters p is still possible, via an auxiliary atom $ab_p \in Ab$ and Horn axioms $p \rightarrow ab_p$, $ab_p \rightarrow p$ which are included in the NAT.

We can view Horn NATs without fixed letters as a generalization of (propositional) logic programs, which consist of Horn clauses $a \leftarrow b_1, \ldots, b_n$, and whose semantics is given in terms of the least (Herbrand) model, which amounts to parallel minimization of all letters. By the above method, any such logic program

Π can be easily transformed into a logically equivalent NAT; if P is the set of letters, simply construct $\mathcal{T}_\Pi = \{P : \Pi,\ p \to ab_p,\ ab_p \to p,\ p \in P\}$, where $At = P \cup Ab$ and $Ab = \{ab_p \mid p \in P\}$. However, NATs offer in addition nesting, and furthermore some of the letters may float to minimize the extension of other letters.

It is well-known that model checking and inference of literals from a logic program is possible in polynomial time (cf. [12]). It turns out that this generalizes to Horn NATs without fixed letters, which can be regarded as a positive result. In fact, as we shall show, any such NAT can be rewritten efficiently to a logically equivalent Horn CNF.

In what follows, let us 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 5.1 *Let $\mathcal{T}$ be a Horn NAT without fixed letters. Then, (i) $\mathcal{T}$ has the least (i.e., a unique minimal) model $M(\mathcal{T})$, and (ii) $\mathcal{T}$ is equivalent to some Horn CNF $\varphi(\mathcal{T})$. Furthermore, both $\varphi(\mathcal{T})$ and $M(\mathcal{T})$ are computable in polynomial time.*

Proof: Let, for any Horn CNF ψ and interpretation M, be ψ^M the Horn CNF which results from ψ after removing from it any clause which contains some literal $(\neg)ab_j \in Ab$ such that $M \models (\neg)ab_j$ and removing all literals $(\neg)ab_j$ such that $M \not\models (\neg)ab_j$ from the remaining clauses.

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

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

Furthermore, define

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

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

(Basis) If $nd(B) = 0$, then every B_i is a Horn CNF, and both $\varphi(B) = \bigwedge_i B_i^{M_0}$ and $\psi(B) = \bigwedge_i B_i$ are Horn CNFs. The block B is equivalent to $\exists Ab.\mathrm{CIRC}(\psi(B); Ab; Z)$, i.e., modulo Ab to $\mathrm{CIRC}(\psi(B); Ab; Z)$. Since $\psi(B)$ is a Horn CNF, it has the $At; \emptyset$-least (i.e., a unique $(At; \emptyset)$-minimal) model M_0. Notice that for every disjoint sets of atoms P and P' such that $P \cup P' = At$ and any model M of $\psi(B)$, it holds that $M_0 \leq_{P;P'} M$. Consequently, the projection $M(B) := M_0[Z]$ is the unique minimal model of B. Thus item (i) holds for B. Furthermore, if M^* is a witness extension of any model M of B, then M^* must coincide on Ab with M_0, i.e., $M^*[Ab] = M_0[Ab]$. Thus, after fixing the value of each atom $ab_j \in Ab$ as in M_0, the formula $\psi(B)$ describes all models of B. That is, $\varphi(B) = \bigwedge_i B_i^{M_0}$ is equivalent to B. Thus item (ii) holds for B.

(Induction) Assume the statement holds for all B with $nd(B) \leq m$, and consider $m+1$. By the induction hypothesis, every B_i in B is equivalent to $\varphi(B_i)$. Thus, B is equivalent to the block $B' = \{Z : B'_1, \ldots, B'_n\}$, where $B'_i = B_i$ if $B_i \in \mathcal{L}$ and $B'_i = \varphi(B_i)$ if $B_i \notin \mathcal{L}$. Since $nd(B') = 0$, by the induction hypothesis B' has the least model $M(B')$ and is equivalent to $\varphi(B')$. Since $\psi(B) = \psi(B')$, we have $M_0(B) = M_0(B')$ and $\varphi(B) = \varphi(B')$. Thus, the statement holds for B, which concludes the induction step.

Let us now estimate the time needed for computing $M(\mathcal{T})$ and $\varphi(\mathcal{T})$, respectively. For this purpose, 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 $\varphi(B)$ bottom up. 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 projection $M_0[V]$ to the set of atoms V which occur 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, cf. [38]. Furthermore, we can compute $\bigwedge_{B_i \in \mathcal{L}} B_i^{M_0}$ 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 (recursively occurring) blocks in $\mathcal{T}$, thus in polynomial time.

By Proposition 4.2, we can replace a multiple block NAT $\mathcal{T} = B_1, \ldots, B_n$ by the single block NAT $\mathcal{T}' = \{At \setminus Ab : B_1, \ldots, B_n\}$, which is Horn and without fixed letters, and obtain analogous results. □

Using sophisticated data structures, the (relevant parts of the) models $M_0(B)$ in the proof of Theorem 5.1 can be computed incrementally, where each clause in $\varphi(B)$ is fired at most once. The data structures refine those used for computing the least model of Horn CNF (see e.g. [38]). Overall, $\varphi(\mathcal{T})$ and $M(\mathcal{T})$ are computable in $O(\|\mathcal{T}\|)$ time. We thus have the following result:

Theorem 5.2 (Flat Normal Form) *Every Horn NAT $\mathcal{T}$ without fixed letters can be rewritten to an equivalent Horn NAT $\{Z : \psi\}$ without fixed letters, where $\psi \in \mathcal{L}$ 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 remark that our normal form result has a pendant in query languages based on fixpoint logic (FPL), which is first-order predicate logic enriched with a generalized quantifier for computing the least fixpoint of an operator, defined in terms of satisfaction of a formula (see [27, 29] for details). It has been shown [27, 29] that over finite structures, nested use of the fixpoint operator can be replaced by a single use of the fixpoint operator. Our result, however, differs in several respects. FPL is an extension to first-order logic, while strictly speaking, NATs are second-order propositional theories. Furthermore, FPL has higher expressiveness than the underlying logic, which is not the case for Horn NATs without fixed letters. Finally, the complexity of rewriting is not a concern in [27, 29] which focus on the existence of equivalent formulas without nestings, rather than on efficient computation.

We note some easy corollaries of Theorem 5.2.

Corollary 5.3 *Deciding the satisfiability of a given Horn NAT $\mathcal{T}$ without fixed letters is polynomial.*

Corollary 5.4 *Model checking for a given Horn NAT $\mathcal{T}$ without fixed letters and model M is polynomial.*

The latter result will be sharpened in the next subsection. For the inference problem, we obtain the following result.

Theorem 5.5 *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: By Corollary 5.4, the problem is clearly in coNP. The coNP-hardness part follows from coNP-completeness of checking the validity of a given formula $\varphi \in \mathcal{L}$ (ask whether $\{Z : \psi\} \models \varphi$, where ψ is any tautology and Z contains all letters occurring in φ).

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 needs $O(\|\varphi(\mathcal{T})\| + \|\alpha_i\|)$ time, for all $i \in \{1, \ldots .m\}$). □

5.2 Horn NATs with fixed letters

The fragment of Horn NATs where fixed letters are allowed generalizes, in a sense, the query language DATALOG^{Circ} considered by Cadoli and Palopoli [10].[4] In this language, circumscription is applied to a conjunction of non-negative Horn clauses, which describes an intensional database, viewing fixed predicates as "free" predicates for which any possible extension is considered, while the other predicates are minimized or floating, respectively. Thus, DATALOG^{Circ} programs can be viewed as unnested Horn NATs.

As we have shown in the previous section, inference from a Horn NAT without fixed letters is coNP-complete, while model checking is polynomial. As we now show, the presence of fixed letters in Horn nestings does not add complexity, i.e., reasoning stays coNP-complete and model checking remains polynomial.

These results build upon the fact that model checking for a Horn circumscription $\mathrm{CIRC}(\varphi; P; Z)$, which may have fixed letters, can be polynomially reduced to model checking for a Horn circumscription without fixed letters. Given an interpretation M, just check whether M is a model of $\mathrm{CIRC}(\varphi \wedge \varphi_{M,Q}; P; Z \cup Q)$, where $\varphi_{M,Q}$ is a conjunction of literals that fixes the values of the letters in Q to the value as given in M. Clearly, the formula $\varphi \wedge \varphi_{M,Q}$ is Horn.

Now the same method work recursively in a nested circumscription as well; we end up with a Horn NAT that has no fixed letters. For such a NAT, model checking is polynomial as we have shown in the previous section. Overall, this means then that we have a polynomial time procedure for model checking in the case of Horn NATs with fixed letters.

More formally, we define the transformation $\alpha(M, B)$, where M is any model and B is either a formula from $\mathcal{L}$ or a block, as follows:

$$\alpha(M,B) = \begin{cases} \varphi, & \text{if } B = \varphi \in \mathcal{L}; \\ \{Z \cup Q : \varphi_{Q,M}, \alpha(M,B_1), \ldots, \alpha(M,B_m)\}, & \text{if } B = \{Z : B_1, \ldots, B_m\} \\ & \text{and } Q = At \setminus (Z \cup Ab) \text{ is} \\ & \text{the set of fixed letters in } B, \end{cases}$$

where $\varphi_{Q,M} = \bigwedge_{q \in Q \cap M} q \wedge \bigwedge_{q \in Q \setminus M} \neg q$. Furthermore, we define

$$\alpha(M, \mathcal{T}) = \bigwedge_{B \in \mathcal{T}} \alpha(M, B)$$

for any interpretation M and NAT $\mathcal{T}$. Observe that $\alpha(M, B)$ and $\alpha(M, \mathcal{T})$ have no fixed letters. The following lemma states that by the transformation $\alpha(M, \mathcal{T})$, fixed letters can be eliminated gracefully for the purpose of model checking.

Lemma 5.6 *For any NAT $\mathcal{T}$ and interpretation M, we have that $M \models \mathcal{T}$ if and only if $M \models \alpha(M, \mathcal{T})$.*

[4] Strictly, this applies to the propositional fragment of DATALOG^{Circ}. The datalog setting of [10] is covered by the generalization of NATs to the first-order case discussed in Section 6.3.

Proof: By definition of $M \models \mathcal{T}$, it remains to show that the statement holds for any $\mathcal{T}$ which consists of a single block $B = \{Z : B_1, \ldots, B_m\}$. This is accomplished by induction on the nesting depth $n \geq 0$.

(Basis) For $n = 0$, we have $B_i = \varphi_i \in \mathcal{L}$, for all $i \in \{1 \ldots, m\}$. Suppose first that $M \models \alpha(M, B)$. By Proposition 4.1, there is some witness extension M^* of M w.r.t. $\alpha(M, B)$ which is an $(Ab; Z \cup Q)$ minimal model of $B_1, \ldots, B_m$ and $\varphi_{Q,M}$. We claim that M^* is a $(Ab; Z)$-minimal model of $B_1, \ldots, B_m$. Indeed, suppose that some $M' <_{Ab;Z} M^*$ exists such that $M' \models B_1, \ldots, M' \models B_m$. Since $M' \models \varphi_{Q,M}$ must hold, it follows that M^* is not a $(Ab; Z \cup Q)$-minimal model of $B_1, \ldots, B_m$ and $\varphi_{Q,M}$. This is a contradiction. Thus, M^* is a (Ab, Z)-minimal model of $B_1, \ldots, B_m$. Hence, $M \models B$.

Conversely, assume that $M \models B$. Then, some witness extension M^* of M w.r.t. B is a $(Ab; Z)$-minimal model of $B_1, \ldots, B_m$. By the definition, $M^* \models \varphi_{M,Q}$. Thus, $M^* \models \psi$ where $\psi = \varphi_{M,Q} \wedge B_1 \wedge \cdots \wedge B_m$. We claim that M^* is a $(Ab; Z \cup Q)$-minimal model of ψ. Towards a contradiction, assume that some $M' <_{Ab;Z \cup Q} M^*$ exists such that $M' \models \psi$. Then, we must have $M^*[Q] = M'[Q]$. Thus, M' is a model of $B_1, \ldots, B_m$ such that $M' <_{Ab;Z} M^*$. This means that $M \not\models B$, which raises a contradiction. Therefore M^* is an $(Ab; Z \cup Q)$-minimal model of $\varphi_{M,Q}, B_1, \ldots, B_m$. Consequently, $M \models \alpha(M, B)$. This proves the claim and concludes the case $n = 0$.

(Induction) Suppose the statement holds for $n \geq 0$, and consider the case $n + 1$. Let $B = \{Z : B_1, \ldots, B_m\}$. Then, $\alpha(M, B) = \{Z \cup Q : \varphi_{M,Q}, \alpha(M, B_1), \ldots, \alpha(M, B_m)\}$. By the induction hypothesis, we have that $M \models B_i$ iff $M \models \alpha(M, B_i)$, for all $i \in \{1, \ldots, m\}$. Using similar arguments as in the case $n = 0$, we can see that $M \models B$ holds precisely if $M \models \alpha(M, B)$ holds. □

By combining Lemma 5.6 and Corollary 5.4, we thus obtain that model checking for Horn NATs is polynomial. A careful analysis of the required computation effort reveals the following result.

Theorem 5.7 *Model checking for Horn NATs, i.e., deciding whether $M \models \mathcal{T}$ for a given interpretation M and a Horn NAT $\mathcal{T}$, is possible in $O(\|\mathcal{T}\|)$ time, i.e., in linear time in the input size.*

Proof: A simple, yet not immediately linear time method is to check that $M \models B$ for each block $B = \{Z : B_1, \ldots, B_m\}$ from $\mathcal{T}$ by exploiting Lemma 5.6 as follows:

1. recursively check that $M \models B_i$, for each $B_i \notin \mathcal{L}$;
2. compute the least model M'_0 of the Horn CNF $\psi'(B) = \varphi_{Q,M} \wedge \bigwedge_{B_i \in \mathcal{L}} B_i$;
3. check whether M is a model of $\psi'(B)[M'_0[Ab]]$.

Note that this method is related to constructing the Horn CNFs $\psi(B)$ and $\varphi(B)$ for a Horn block B without fixed letters in the proof of Theorem 5.1. Step 2 can be done in time $O(\max(|At|, \|\psi'(B)\|))$ and Step 3 in time $O(\|\psi'(B)\|)$. These upper bounds, however, may be reached and exceed $O(\|\{Z : B_{i_1}, \ldots, B_{i_l}\}\|)$, where the B_{i_j} are those blocks in B which are not from $\mathcal{L}$. If this happens recursively, the total time of the method fails to be $O(\|B\|))$ as desired.

In Step 2, we can replace $\psi'(B)$ by $\psi''(B) = \bigwedge_{B_i \in \mathcal{L}} B_i[M[Q]]$ and compute the least model M''_0 of $\psi''(B)$ on the letters occurring in it; this is feasible in $O(\sum_{B_i \in \mathcal{L}} \|B_i\|)$ time. In Step 3 then, we can replace $\psi'(B)[M'_0[Ab]]$ by $\psi''(B)[M''_0[Ab]]$; checking whether $M \models \psi''(B)[M''_0[Ab]]$ is feasible in $O(\sum_{B_i \in \mathcal{L}} \|B_i\|)$ time. Thus, the revised Steps 2 and 3 can be done in $O(\|\{Z : B_{i_1}, \ldots, B_{i_l}\}\|)$ time. This implies that checking $M \models B$ is feasible in $O(\|B\|)$ time, from which the result follows. □

Furthermore, we obtain from Lemma 5.6, Corollary 5.4 and the intractability result for Horn circumscription in [9] the following result:

Theorem 5.8 *Deciding, given a Horn NAT $\mathcal{T}$ and a propositional formula φ, whether $\mathcal{T} \models \varphi$ is* coNP*-complete. Hardness holds even if $\mathcal{T}$ has nesting depth 0, and φ is a negative literal $\neg u$.*

This means that nesting is not a source of complexity for model checking and inference from Horn NATs, which can be viewed as positive result.

6 Further Issues

In this section, we consider possible extensions of the results in the previous sections to other representation scenarios. We first address the class of $\mathcal{L}_{\mathrm{CIRC}}$ formulas and of NATs which do not have fixed letters; as we have seen in the previous section, the presence of fixed letters did not matter for the complexity of Horn NATs. We then turn to a linguistic extension of NATs which has explicit maximization and minimization of letters as primitives. While this extension does not increase the expressiveness of NATs in general, it has some effects on restricted NAT classes, and in particular on Horn NATs. Finally, we briefly address the generalization of $\mathcal{L}_{\mathrm{CIRC}}$ and NATs to the predicate logic setting.

6.1 $\mathcal{L}_{\mathrm{CIRC}}$ formulas and NATs without fixed letters

In Sections 3.1 and 4.2, we have considered $\mathcal{L}_{\mathrm{CIRC}}$ formulas and NATs in a general setting which allows for fixed letters in circumscriptions, and we have seen in the previous section that the presence of fixed letters does not matter for the complexity of Horn NATs.

As shown below, fixed letters can be removed from $\mathcal{L}_{\mathrm{CIRC}}$ and NAT theories, respectively, by simple techniques. By exploiting them, the hardness results of Sections 3.1 and 4.2 can be sharpened to theories without fixed letters.

6.1.1 Eliminating fixed letters from a $\mathcal{L}_{\mathrm{CIRC}}$ formula

De Kleer and Konolige have shown [15] a simple technique for removing the fixed letters from an ordinary circumscription. The same technique can be applied for formulas from $\mathcal{L}_{\mathrm{CIRC}}$ as well. More precisely, let $\varphi = \mathrm{CIRC}(\psi; P; Z)$ be a circumscriptive atom. Then,

1. For each letter $q \notin P \cup Z$, introduce a fresh letter q', and add both q, q' to P;
2. add a conjunct $q \leftrightarrow \neg q'$ to ψ.

Let $\varphi' = \mathrm{CIRC}(\psi'; P'; Z)$ be the resulting circumscriptive atom. Then, the following holds.

Proposition 6.1 *Modulo the set of all auxiliary letters q', the formulas φ and φ' are logically equivalent.*

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

6.1.2 Eliminating fixed letters from a NAT

Every fixed letter q can be removed from a NAT $\mathcal{T}$ similarly as from a formula $\varphi \in \mathcal{L}_{\mathrm{CIRC}}$. 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 the special set Ab of minimized letter which has restricted uses. We surpass this as follows:

1. Introduce, for each fixed letter q, two special abnormality letters ab_q and ab'_q in Ab;
2. add the formula $(q \leftrightarrow ab_q) \wedge (ab_q \leftrightarrow \neg ab'_q)$ as a new block in each block B occurring in $\mathcal{T}$ where q is fixed;
3. declare q as described (i.e., floating) in each block occurring in $\mathcal{T}$.

Let $\mathcal{T}'$ be the resulting NAT (which has an extended set of abnormality letters, $Ab \cup Ab'$). Then, we have:

Proposition 6.2 *Modulo the set Ab' of auxiliary letters, the NATs $\mathcal{T}$ and $\mathcal{T}'$ are logically equivalent, i.e., have the same sets of models.*

Note that the rewriting adds $O(|At|)$ symbols in each block, and is feasible in $O(|At| \cdot \#b(\mathcal{T}))$ time, where $\#b(\mathcal{T})$ is the number of (recursively occurring) blocks in $\mathcal{T}$. Furthermore, observe that the method uses non-Horn clauses. This is not accidently; from the tractability result for inference of a CNF from a Horn NAT without fixed letters (Theorem 5.5) and the intractability of inference of a literal from a Horn circumscription [9], we can infer that there is no simple polynomial-time rewriting method which uses only Horn clauses, unless $\mathrm{P} = \mathrm{NP}$. This is also possible if we allow $\mathcal{T}'$ to be any Horn NAT without fixed letters (not necessarily equivalent) and the query to be replaced by any CNF φ', such that $\mathcal{T} \models \varphi$ is equivalent to $\mathcal{T}' \models \varphi'$ for the query φ at hand.

6.2 Maximizing and minimizing predicates

In his seminal paper [35], Lifschitz discussed two explicit constructs $\min p$ and $\max p$ for defining a minimal and a maximal extension of a letter p in a NAT, respectively. These constructs are easily implemented by using designated abnormality letters.

Definition 6.1 *An* extended block *is any expression*

$$\{C;\ \min C^-;\ \max C^+ : B_1, \ldots, B_m\}, \tag{5}$$

where C, C^-, and C^+ are disjoint sets of atoms from $At \setminus Ab$; if empty, the respective component is omitted. Intuitively, the letters in C are defined as usual while for those in C^- (resp., C^+), a minimal (resp., maximal) extension is preferred. An extended NAT *is a collection $\mathcal{T} = B_1, \ldots, B_n$ of extended blocks.*

Example 6.1 Let us consider model-based diagnosis at a superficial level. In Reiter's approach [42], a diagnosis problem consists of a system description SD, a set of observations OBS (which are facts), and a set of components $COMP = \{c_1, \ldots, c_m\}$ in the system. SD is a set of axioms which describe the structure and the functioning of the system, using designated atoms ok_i which informally expresses that component c_i works properly. A *diagnosis* is a minimal set $\Delta \subseteq C$ such that $SD \cup \{OBS\} \cup \{\neg ok_i \mid c_i \in \Delta\} \cup \{ok_j \mid c_j \in COMP \setminus \Delta\}$ is satisfiable. That is, Δ assumes as little malfunctionings as needed to explain the observations (equivalently, as many components as possible are assumed to work properly).

Assuming a modular system design, each component c_i may be represented as a block $B_i = \{ok_i, V_i : \ldots\}$, where inputs are passed to B_i via variables that are fixed, and outputs from c_i are modeled by variables V_i which are described, together with a variable ok_i which indicates whether B_i works properly. The components may be linked by some axioms $\varphi_1, \ldots, \varphi_n$, such that $B = \{V; \max ok_1, \ldots, ok_m : \varphi_1, \ldots, \varphi_n, B_1, \ldots, B_m\}$ represents the system. Then, the models of $\mathcal{T} = B, \{ : OBS \}$ correspond to the

diagnoses of the system. If a block B_i is hierarchically composed, further nesting of blocks may be used in the modeling.

As an example, consider the following very simplified model of a Web server for electronic commerce, composed of two modules:

1. an application server, with features for the client interface and the interaction with the database system,
2. a database system storing data on customers, orders, etc. with a query that must be executed on it for each interaction with the client.

The modules can be, respectively, modeled by means of the following blocks:

1. $B_1 = \{ok_1, V_1 : \ ci \wedge db \wedge ok_1 \rightarrow V_1\}$, where ci and db mean, respectively, that the interaction with the client and the database system have been performed;
2. $B_2 = \{ok_2, V_2 : \ V_1 \wedge q \wedge ok_2 \rightarrow V_2, \ V_1 \wedge \neg q \rightarrow \neg ok_2\}$, where q means that the query has been executed.

Description of the entire system can be made by means of the following block:

$$B = \{V, V_1, V_2; \max ok_1, ok_2 : \ V_1 \wedge V_2 \rightarrow V, \ B_1, \ B_2\},$$

where V is a new symbol. The above description can be used, for example, during the test phase of the Web server, in which interaction with one client is simulated. During such a phase, an administrator checks whether the query and the interactions between modules have been performed. Now, assume that the administrator determines that interactions with the client and the database system have been performed, but the query has not been executed, i.e., the set of his observations is $OBS = \{ci, db, \neg q\}$. It is easy to determine that the diagnoses of the system correspond to either $\{\neg ok_1\}$, or $\{\neg ok_2\}$, i.e., one subsystem is malfunctioning, but not both. □

Other examples for the use of maximization can be found in [35].

Formally, the semantics of an extended block B as in (5) can be defined by a transformation $(\cdot)^\circ$ to the ordinary block

$$B^\circ = \{C \cup C^- \cup C^+ : \varphi_{C^-}, \varphi_{C^+}, B_1, \ldots, B_m\}, \tag{6}$$

where $\varphi_{C^-} = \bigwedge_{p \in C^-}(p \rightarrow ab_p)$ and $\varphi_{C^+} = \bigwedge_{p \in C^+}(\neg ab_p \rightarrow p)$, and each ab_p is an abnormality letter not used in any B_i which is a formula from $\mathcal{L}$. For any extended NAT $\mathcal{T} = B_1, \ldots, B_n$, we then define $\mathcal{T}^\circ = B_1^\circ, \ldots, B_n^\circ$.

Thus, the constructs $\min$ and $\max$ do not increase the expressiveness of NATs in general. However, we have a different picture in restricted cases. In particular, maximization of letters increases the expressiveness of Horn NATs. As follows from the next theorem, extended Horn NATs climb the levels of PH closely behind general NATs as the nesting depth increases (at one level distance for inference and satisfiability, and at two levels for model checking), and they are PSPACE-complete for unbounded nesting depth. Note that maximization in Horn NATs is useful. In the diagnosis application of Example 6.1, the axioms describing the system might be Horn; for example, the clauses $in_1 \wedge in_2 \wedge ok_g \rightarrow out$, $out \wedge ok_g \rightarrow in_1$, and $out \wedge ok_g \rightarrow in_2$ may describe a logical and-gate g whose output is true, if working properly, exactly if both inputs are true. Note that in the particular diagnostic Web-Server scenario, all formulas in $\mathcal{T}$ are Horn clauses except $V_1 \wedge \neg q \rightarrow \neg ok_2$, which can be easily rewritten to a Horn clause.

We need some auxiliary results, which are of interest in their own right. In what follows, we denote for any block B by $SBl(B)$ the set containing B and all blocks B' that recursively occur in B, and for any NAT $\mathcal{T} = B_1, \ldots, B_n$, we define $SBl(\mathcal{T}) = \bigcup_{i=1}^{n} SBl(B_i)$.

Proposition 6.3 *Let B be any block such that for every $B' \in SBl(B)$, (i) B' allows polynomial model completion, and (ii) model checking $M \models B'$ is polynomial if $nd(B') = 0$. Then model checking $M \models B$ is in Π_k^P.*

Proof: The proof is similar to the proof of the Π_{k+1}^P-membership part in Theorem 4.12 for the case of polynomial model completion, but exploits that in the base case ($nd(B) = 0$), model checking is polynomial rather than in $\mathrm{coNP} = \Pi_1^P$. □

Proposition 6.4 *Let $\mathcal{T}$ be any extended Horn NAT. Then, every block $B \in SBl(\mathcal{T})$ allows polynomial model completion.*

Proof: Let $B = \{C;\, \min C^-;\, \max C^+ : B_1, \ldots, B_m\}$. Without loss of generality, we assume that C^- is empty, i.e., the min-part is missing: since the formula φ_{C^-} is Horn, we may add it in B while keeping the Horn property and move C^- to the ordinary defined letters C. Suppose that $B_1, \ldots, B_l$ ($l \leq m$) are all blocks B_j that are formulas (i.e., $B_j \in \mathcal{L}$), and let ψ be their conjunction. Let M be the model to be completed. Define

$$\psi' = \psi \wedge \bigwedge_{p \in (At \setminus Ab) \cap M} p \wedge \bigwedge_{p \in C^+ \cap M} ab_p \wedge \bigwedge_{p \in At \setminus (Ab \cup M)} \neg p \wedge \bigwedge_{p \in C^+ \setminus M} \neg ab_p \,.$$

Note that ψ' is Horn, and thus, if satisfiable, it has the unique least model M', which obviously coincides with M on the atoms in $(At \setminus Ab) \cup \{ab_p \mid p \in C^+\}$ and is computable in polynomial time.

Consider the transformed block $B^\circ = \{C \cup C^+ : \varphi_{C^+}, B_1, \ldots, B_m\}$. We claim that $M \models B$ if and only if M' is a $(Ab; C \cup C^+)$-minimal model of φ_{C^+}, B_1,..., B_m. By Proposition 4.1, the if-direction is immediate. For the only-if direction, suppose that $M \models B$. Then, by Proposition 4.1, there exists a witness extension M^* of M which is a $(Ab; C \cup C^+)$-minimal model of φ_{C^+}, B_1,..., B_m. Since M and M' coincide on $At \setminus Ab$ and each atom ab_p, $p \in C^+$ occurs only in φ_{C^+}, the minimality of M^* implies that M^* and M' coincide on $(At \setminus Ab) \cup \{ab_p \mid p \in C^+\}$ and thus $M^* \models \psi'$. Since M' is the least model of ψ', it follows $M' \leq_{Ab; C \cup C^+} M^*$. By construction, $M' \models \varphi_{C^+}$, and $M^* \models B_i$ implies that $M' \models B_i$, for each block B_j where $j \in \{l+1, \ldots, m\}$. Thus, from the minimality of M^*, we conclude that $M' = M^*$. This proves the claim. □

Theorem 6.5 *For extended Horn theories $\mathcal{T}$, (i) model checking $M \models \mathcal{T}$, (ii) inference $\mathcal{T} \models \varphi$, and (iii) deciding satisfiability are* PSPACE*-complete. Furthermore, (i) is polynomial if $k = 0$ and Π_k^P-complete if $k \geq 1$, (ii) is Π_{k+1}^P-complete, and (iii) is Σ_{k+1}^P-complete, if $nd(\mathcal{T}) \leq k$ for a constant $k \geq 0$.*

Proof: For the membership parts, by Theorems 4.9, 4.11 and Corollary 4.10, it remains to show the statement for bounded nesting depth. From Propositions 6.4 and 6.3, this is easily seen to hold, provided that model checking $M \models B$ for any extended Horn block $B = \{C;\, \max C^+ : B_1, \ldots, B_m\}$ such that $nd(B) = 0$ is polynomial.

To prove the latter, let M' be the least model of the Horn CNF ψ' constructed from B in the proof of Proposition 6.4. As shown there, $M \models B$ iff M' is a $(Ab; C \cup C^+)$-minimal model of φ_{C^+}, B_1,..., B_m. We can check $M' \models \varphi_{C^+}$ and $M' \models B_i$ for all B_i easily in polynomial time. Furthermore, we can check

$(Ab; C \cup C^+)$-minimality of M' by testing whether each of the following Horn CNFs ψ_ℓ is unsatisfiable. Let $F^+ = M \cap (At \setminus (Ab \cup C))$ and $F^- = At \setminus (M \cup C \cup C^+ \cup \{ab_p \mid p \in C^+\})$. For each literal $\ell \in (C^+ \setminus M) \cup \{\neg ab \mid ab \in M \cap (Ab \setminus \{ab_p \mid p \in C^+\})\}$, define

$$\psi_\ell = \ell \wedge \bigwedge_{p \in F^+} p \wedge \bigwedge_{p \in F^-} \neg p \wedge \bigwedge_{i=1}^{m} B_i;$$

that is, we fix the "interesting" letters which are not defined in B to their values in M', fix each letter from C^+ which is true in M', and fix each "regular" abnormality letter (not introduced for a letter in C^+) which is false in M'; furthermore, ℓ serves to increase one letter in C^+ (resp. decrease one regular abnormality letter) compared to M'. Thus, no model M'' exists such that $M'' <_{Ab;C \cup C^+} M'$ iff each ψ_ℓ is unsatisfiable, which can be checked in polynomial time. In summary, testing whether M' is an $(Ab; C \cup C^+)$-minimal model of φ_{C^+}, B_1,..., B_m, and thus whether $M \models B$, is possible in polynomial time. This concludes the proof of the membership parts.

The hardness proofs for (i) and (ii) are obtained by slight modifications of the reductions in the proofs of Theorems 4.11 and 4.9 (i.e., Lemma 4.7). The hardness proof for (iii) follows from the hardness proof of (i), since the formula φ in the reduction is a single literal and $\mathcal{T} \models \varphi$ iff the NAT $\mathcal{T}, \{ : \neg\varphi\}$ is unsatisfiable.

For (ii), the modifications to the NATs $\mathcal{T}_1, \ldots, \mathcal{T}_n$ in the proof of Theorem 4.9 are as follows:

1. Drop in each $\mathcal{T}_{2k+1}$ (resp., $\mathcal{T}_{2k}$) the formula $u \leftrightarrow ab$, (resp., $u \leftrightarrow \neg ab$), and declare u minimized (resp., maximized).

2. Introduce for each letter $p \in At \setminus (Ab \cup \{u\})$ (=: A) a fresh letter p'; intuitively, p' serves for emulating the negation of p. This is accomplished by adding in $\mathcal{T}_1$ an extended Horn block $B_{\not\equiv} = \left\{ \max A, A' : \bigwedge_{p \in A}(\neg p \vee \neg p') \right\}$. Informally, the parallel maximization of p and p' generates two models; one has p true and p' false, and the other has vice versa p' true and p false. In this way, p' is defined as the complement of the p.

3. We replace in $\mathcal{T}_1$ the formula $\varphi = \psi \vee u$ by the Horn CNF $\hat{\varphi}' = \bigwedge_{j=1}^{l}(\gamma'_j \vee u)$, where w.l.o.g. $\psi = \bigwedge_{j=1}^{l} \gamma_j$ is conjunction of clauses and γ'_j results from γ_j by replacing each positive literal x by the negative primed literal $\neg x'$.

4. We let p' be described in the same NATs $\mathcal{T}'_j$ where p is described, for each $p \in A$.

The resulting NATs, denoted $\hat{T}_1, \ldots, \hat{T}_n$, are thus as follows:

$$\begin{aligned}
\hat{T}_1 &= \{X_1, X'_1; \min u : \hat{\varphi}, B_{\not\equiv}\}, \\
\hat{T}_{2k} &= \{X_1, X'_1, \ldots, X_{2k}, X'_{2k}, \max u : \hat{T}_{2k-1}\}, && \text{for all } 2k \in \{2, \ldots, n\}, \\
\hat{T}_{2k+1} &= \{X_1, X'_1, \ldots, X_{2k+1}, X'_{2k+1}; \min u : \hat{T}_{2k}\}, && \text{for all } 2k+1 \in \{3, \ldots, n\}.
\end{aligned}$$

Observe that $nd(\hat{T}_n) = n$. It is easily seen that modulo the new letters, $\hat{T}_j$ and $\mathcal{T}_j$ have the same models, for $j = \{1, \ldots, n\}$. Thus, the hardness result for (ii) follows.

For (ii), the modifications to the NATs $\mathcal{T}'_1, \ldots, \mathcal{T}'_n$ in the proof of Theorem 4.11 are similar to those in (i), but with the following differences:

- We perform the reduction with empty X_{n+1}, i.e., we suppress the leading quantifier $Q_n X_{n+1}$; the formulas φ_g and φ_c are removed from $\mathcal{T}'_n$ (they are tautologies).

- In step 3, instead of $\varphi = \psi \vee u$ we replace in $\mathcal{T}_1'$ the formula $\psi \vee u \vee (X_n \wedge v)$ by the Horn CNF $\hat{\varphi}' = \bigwedge_{j=1}^{l} \bigwedge_{x \in X_n \cup \{v\}} (\gamma_j' \vee \neg x' \vee u)$.

The resulting NATs, denoted $\hat{T}_j$, are for odd $n > 1$ thus as follows (for even n, they are analogous):

$$\begin{aligned}
\hat{T}_1' &= \{X_1, X_1'; \min u : \hat{\varphi}', (X_n \wedge v) \rightarrow u, B_{\neq}\}, \\
\hat{T}_{2k}' &= \{X_1, X_1', \ldots, X_{2k}, X_{2k}'; \max u : \hat{T}_{2k-1}'\}, \qquad \text{for all } 2k \in \{2, \ldots, n-1\}, \\
\hat{T}_{2k+1}' &= \{X_1, X_1', \ldots, X_{2k+1}, X_{2k+1}'; \min u : \hat{T}_{2k}'\}, \qquad \text{for all } 2k+1 \in \{3, \ldots, n-1\}, \\
\hat{T}_n' &= \{X_1, X_1', \ldots, X_n, X_n', v, v'; \min u : \hat{T}_{n-1}'\}.
\end{aligned}$$

Notice that $nd(\hat{T}_n') = n$. Modulo the new letters, $\hat{T}_j'$ and $\mathcal{T}_j'$ have the same models, for $j \in \{1 \ldots, n\}$. Thus, the hardness result for (i) follows. □

Note that model checking for extended Horn NATs resides in PH two levels below arbitrary NATs of the same nesting depth. The proof reveals that this can be ascribed to the benign properties that both model completion and polynomial-time model checking for an extended Horn circumscription (where maximization of letters besides minimization is allowed) are polynomial. Each of these tasks is a source of complexity, i.e., intractable for arbitrary NATs. In particular, for a collection of unnested extended Horn blocks, model checking is polynomial and inference is coNP-complete, which means that the latter can be polynomially transformed to a SAT solver. Likewise, for nesting depth 1, inference is Π_2^P-complete, and thus polynomially reducible to inference from an ordinary (non-Horn) circumscription, as well as to engines for knowledge representation and reasoning which are capable of solving Π_2^P-complete problems, such as DLV [17, 22].

We finally remark that using maximization, fixed letters can be easily eliminated from extended Horn NATs similarly as from general NATs. (Namely, introduce in each block B for every fixed letter q a fresh letter q_B, and add the clause $\neg q \vee \neg q_B$ in B and declare q_B and q maximized; in all other blocks, let both q and q_B float.) Thus, the complexity results for extended Horn NATs from above can be strengthened to theories without fixed letters.

6.3 First-order case

In this paper, we have considered so far nested circumscription and NATs in a propositional language. There is no difficulty in extending the language $\mathcal{L}_{\mathrm{CIRC}}$ to the case of first-order predicate logic, along the definition of second-order parallel circumscription of predicates [33, 34]; the formulation of NATs in [35] is actually for predicate logic.

As shown by Schlipf [45], and further elaborated on in [8], circumscription is capable of expressing problems at the Σ_2^1 and Π_2^1 level of the prenex hierarchy of second-order logic, and thus highly expressive far beyond the computable. Thus, also nested circumscription and NATs are highly undecidable in the general first-order setting. However, decidable fragments can be obtained by imposing suitable restrictions.

An important such fragment is given if the theories include a *domain closure axiom*

$$\text{(DCA)} \quad \forall x.(x = c_1 \vee x = c_2 \vee \cdots \vee x = c_n), \tag{7}$$

where $c_1, \ldots, c_n$ are the (finitely many) constant symbols available, and the *unique names axioms*

$$\text{(UNA)} \quad c_i \neq c_j, \qquad \text{for all } i \in \{1, \ldots, n\} \text{ and } j \in \{i+1, \ldots, n\}. \tag{8}$$

Such a setting is quite popular in KR and, in the absence of function symbols, in deductive databases, where it is also known as the "datalog" setting. It is essentially propositional, where in the datalog setting

models correspond to Herbrand models over the given alphabet. The setting allows for a more compact representation, which on the other hand may lead to an exponential complexity increase. This is reflected in the complexity of $\mathcal{L}_{\mathrm{CIRC}}$ and NATs in this setting.

Theorem 6.6 *Inference and satisfiability of a first-order $\mathcal{L}_{\mathrm{CIRC}}$ formula (resp., NAT $\mathcal{T}$) under DCA and UNA is* EXPSPACE*-complete.*

The upper bounds are straightforward by reducing a $\mathcal{L}_{\mathrm{CIRC}}$ formula (resp., NAT) to its equivalent ground instance, which is propositional and constructible in exponential time; functions $f(x_1, \ldots, x_n)$ can be eliminated, as well-known, with polynomial overhead by introducing fresh predicates $F(x_1, \ldots, x_n, y)$ and axioms $\forall x_1 \cdots x_n ! \exists y . F(x_1, \ldots, x_n, y)$ such that $\lambda y F(x_1, \ldots, x_n, y)$ amounts to $\lambda y (y = f(x_1, \ldots, x_n))$. The lower bounds for these results are obtained by a straightforward generalization of the QBF encoding in Lemma 4.7 to encodings of sentences $Q_n P_n Q_{n-1} P_{n-1} \cdots \forall P_2 \exists P_1 \psi$ of second-order logic, where each P_i is a list of predicate variables of given arities and ψ is function-free first-order. For bounded nesting depth, the complexities parallel the respective levels of PH at its exponential analogue, the Weak EXP Hierarchy (EXP, NEXP, $\mathrm{NEXP}^{\mathrm{NP}}$, $\mathrm{NEXP}^{\Sigma_2^P}$,...). For example, inference $\varphi \models \psi$ of $\mathcal{L}_{\mathrm{CIRC}}$ sentences φ and ψ is co-$\mathrm{NEXP}^{\Sigma_k^P}$-complete, if the nesting depth of φ and ψ is bounded by a constant $k \geq 0$.

For model checking, things are slightly different. Under a common bitmap representation, in which $M \models a$ for any ground atom a is represented by a designated bit, the complexity of model checking in $\mathcal{L}_{\mathrm{CIRC}}$ does not increase, since the (exponential) size of the explicitly given model M compensates the succinctness of implicit representation.

Theorem 6.7 *Model checking for first-order $\mathcal{L}_{\mathrm{CIRC}}$ under DCA and UNA is* PSPACE*-complete.*

Notice, however, that the problem is PSPACE-hard already for sentences of CIRC-nesting depth 0, i.e., for ordinary first-order sentences, since model checking for a given first-order sentence is PSPACE-hard. As easily seen, model checking for first-order NATs is also PSPACE-complete, if the arities of abnormality predicates used do not exceed the arities of the other predicates and the functions by a constant factor, which is expected to be the case in practice. Similar as for $\mathcal{L}_{\mathrm{CIRC}}$, the problem is PSPACE-hard already for NATs of nesting depth 0. However, in the general case, the complexity can be seen to increase beyond NEXP; we leave a detailed investigation of this for further work.

7 Comparison to Other Generalizations of Circumscription

In this section, we briefly compare nested circumscription to some other generalizations of circumscription from the literature, namely prioritized circumscription [33, 34] and theory curbing [20]. Although there are several other generalizations, cf. [34], the ones considered here are of particular interest since the former has close semantic relationships to nested circumscription, while the latter is similar in terms of the complexity.

7.1 Prioritized Circumscription

Prioritized circumscription [33, 34] generalizes circumscription $\mathrm{CIRC}(\varphi; P; Z)$ by partitioning the letters P into priority levels $P_1 > P_2 > \cdots > P_n$; informally it prunes 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. [34]). This can be readily

expressed as the nested circumscription ψ_n, where

$$\begin{aligned}\psi_1 &= \mathrm{CIRC}(\varphi; P_1; Z \cup P_2 \cup \cdots \cup P_n),\\ \psi_i &= \mathrm{CIRC}(\psi_{i-1}; P_i; Z \cup P_{i+1} \cup \cdots \cup P_n), \quad i = 2, \ldots, n.\end{aligned}$$

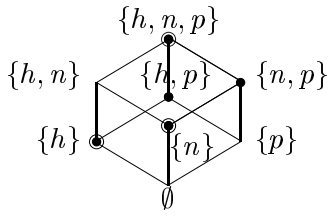
Thus, prioritized circumscription is semantically subsumed by $\mathcal{L}_{\mathrm{CIRC}}$. Compared to ordinary circumscription, the complexity does not increase, as inference and model checking remain Π_2^P-complete and coNP-complete, respectively.

Intuitively, the reason is that prioritization allows only for a restricted change of the role of the same letter in iterations (from floating to minimized and from minimized to fixed), which forbids to reconsider the value of minimized letters at a later stage of minimization. This enables a characterization of the models of a prioritized circumscription as the minimal models of a preference relation $\leq_{P_1,\ldots,P_n;Z}$ on the models, where $M \leq_{P_1,\ldots,P_n;Z} M'$ holds if and only if M and M' coincide on the fixed letters and either M and M' coincide on all P_i, or M is smaller than M' on the first P_i on which M and M' are different. This preference relation is polynomial-time computable. On the other hand, $\mathcal{L}_{\mathrm{CIRC}}$ formulas (and similarly NATs) permit that minimized letters are reconsidered at a later stage, by making them floating. This prevents a simple, hierarchical preference relation as the one for prioritized circumscription.

7.2 Theory Curbing

Theory curbing is yet another extension of circumscription [20, 19]. 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, $\mathrm{CURB}(\varphi; P; Z)$ for a formula $\varphi \in \mathcal{L}$ is the smallest set $\mathcal{M} \subseteq mod(\varphi)$ which contains all models of $\mathrm{CIRC}(\varphi; P; Z)$ 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 $N <_{P;Z} M'$.

Example 7.1 Suppose Alice is in a room with a painting, which she hangs on the wall p if he has a hammer (h) and a nail (n). It is known that Alice has a hammer or a nail or both. This scenario is represented by the formula φ in Figure 1. The models of φ are marked with bullets; the desired models are $\{h\}$, $\{n\}$, and $\{h, n, p\}$, which are encircled. Circumscribing φ by minimizing all letters, i.e., $\mathrm{CIRC}(\varphi; \{h, n, p\}; \emptyset)$ yields the two minimal models $\{h\}$ and $\{n\}$ (see Figure 1). Since p is false in the minimal models, circumscription



$$\varphi \;=\; (h \vee n) \wedge ((h \wedge n) \rightarrow p)$$

Figure 1: The hammer-nail-painting example

tells us that Alice does not hang the painting up. One might argue that p should not be minimized but fixed under circumscription. However, starting with the model of φ where h, n and p are all true and then circumscribing with respect to h and n while keeping p true, we obtain the smaller models $\{h, p\}$ and $\{n, p\}$,

which are not very intuitive. The remaining possibility is to let p float. However, this does not work either, since the circumscription $\mathrm{CIRC}(\varphi; h, n; p) \equiv ((h \leftrightarrow \neg n) \wedge \neg p)$ is equivalent to $\mathrm{CIRC}(\varphi; h, n, p; \emptyset)$ On the other hand, the model $\{h, n, p\}$, which corresponds to the inclusive interpretation of the disjunction $h \vee n$, seems plausible. Under curbing, we obtain the desired models from $\mathrm{CURB}(\varphi; h, n; p)$. □

Like for $\mathcal{L}_{\mathrm{CIRC}}$ and NATs, inference and model checking for $\mathrm{CURB}(\varphi; P; Z)$ are PSPACE-complete [19] in the propositional context, and can be shown to have likewise exponentially higher complexity in the datalog setting (i.e., in a function-free language under DCA and UNA, cf. Section 6.3).

However, while the complexity is the same, curbing and NATs have different expressiveness, if we consider these formalisms as query languages for uniformly expressing properties over collections of ground facts, such as 3-colorability of graphs which are described by their edge relations. It turns out that curbing can express some properties which $\mathcal{L}_{\mathrm{CIRC}}$ and NATs (most likely) can not express. For example, we can 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 of 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 any fixed $\mathcal{L}_{\mathrm{CIRC}}$ formula or NAT (unless PH=PSPACE).

We elaborate on this interpreter for propositional curbing in more detail. The constants represent the propositional atoms, and the clauses of φ are stored using 3-ary predicates R_0, R_1, R_2, and R_3, where $R_i(x_1, x_2, x_3)$ intuitively represents the clause $\bigvee_{j=1}^{i} x_j \vee \bigvee_{j=i+1}^{3} \neg x_j$. E.g., $R_2(a, c, b)$ represents the clause $a \vee c \vee \neg b$. Unary predicates *pvar* and *zvar* are used for designating the atoms in P and Z, respectively.

The theory T_I is as follows:

$$
\begin{aligned}
&\forall x, y, z.\, R_0(x, y, z) \rightarrow \neg t(x) \vee \neg t(y) \vee \neg t(z),\\
&\forall x, y, z.\, R_1(x, y, z) \rightarrow \ \ t(x) \vee \neg t(y) \vee \neg t(z),\\
&\forall x, y, z.\, R_2(x, y, z) \rightarrow \ \ t(x) \vee \ t(y) \ \vee \neg t(z),\\
&\forall x, y, z.\, R_3(x, y, z) \rightarrow \ \ t(x) \vee \ t(y) \ \vee \ t(z),\\
&\qquad \forall x.\ p(x) \leftrightarrow (pvar(x) \wedge t(x)),\\
&\qquad \forall x.\ q(x) \leftrightarrow (\neg pvar(x) \wedge \neg zvar(x) \wedge t(x)).
\end{aligned}
$$

Intuitively, $t(x)$ means that x has value true. Here, the predicate p is minimized, while q is fixed and t is floating.

The set of facts $F(\varphi)$ contains

1. for each clause $(\neg)a \vee (\neg)b \vee (\neg)c$ from φ the respective atom $R_i(a, b, c)$;
2. for each $p \in P$ (resp., $z \in Z$) the atom $pvar(p)$ (resp., $zvar(z)$);
3. the negations of all other ground atoms (i.e., $F(\varphi)$ is the CWA given the atoms in 1 and 2).

Example 7.2 Reconsider $\mathrm{CURB}(\varphi; h, n; p)$ for the formula $\varphi = (h \vee n) \wedge (\neg h \vee \neg n \vee p)$ (rewritten as a CNF) from Example 7.1. Then, the constants are h, n, p. The positive facts in $F(\varphi)$ are $R_3(h, n, n)$ and $R_1(p, h, n)$ encoding the first and the second clause of φ, respectively (where we add a redundant disjunct n in the first clause), and $pvar(h)$, $pvar(n)$, and $zvar(p)$.

Note that $T_I \cup F(\varphi)$ logically implies $t(h) \vee t(n)$, $t(p) \vee \neg t(h) \vee \neg t(n)$, $\neg p(p)$, $p(h) \leftrightarrow t(h)$, $p(n) \leftrightarrow t(n)$, $\neg q(h)$, $\neg q(n)$, and $\neg q(p)$. Thus, Herbrand models of $T_I \cup F(\varphi)$ may differ only on the atoms $t(h)$, $t(n)$, $t(p)$, $p(h)$, and $p(n)$. The feasible assignments of these atoms correspond to the models of φ. If M is a

model of φ, then by assigning true to the atoms $t(a)$ where $a \in M$ and $p(a)$ where $a \in M \cap \{h, n\}$, we obtain a feasible such truth assignment. On the other hand, if M is a Herbrand model of $\mathrm{CURB}T_I \cup F(\varphi); p; t)$, then $\{a \mid t(a) \in M\}$ is a model of φ. Overall, the Herbrand models of $T_I \cup F(\varphi)$ correspond 1-1 to the models of φ. □

The following proposition, whose proof is omitted, states that the interpreter works similarly in the general case.

Proposition 7.1 *Under DCA and UNA, the models of* $\mathrm{CURB}(T_I \cup F(\varphi); p; t)$ *and* $\mathrm{CURB}(\varphi; P; Z)$ *are in 1-1 correspondence.*

From results in [19], we easily obtain that evaluating any given QBF Φ (which is PSPACE-complete) is polynomially reducible to deciding $\mathrm{CURB}(\varphi, P; Z) \models \neg a$, where $\varphi \in \mathcal{L}$ is in 3CNF and a is an atom. Thus, $\mathrm{CURB}(T_I \cup F(\varphi)); p; t) \models \neg t(a)$ expresses evaluating the QBF Φ given by $F(\varphi)$.

On the other hand, unless PH = PSPACE, a "datalog" $\mathcal{L}_{\mathrm{CIRC}}$ formula resp. NAT similar to T_I does not exist: due to fixed nesting depth, it can only express a problem in PH.

Further relationships between $\mathcal{L}_{\mathrm{CIRC}}$ resp. NATs and curbing, as well as other expressive knowledge representation formalisms (e.g., [4, 21, 41]), remain to be explored.

8 Conclusion

In this paper, we have studied the computational complexity of the logical language $\mathcal{L}_{\mathrm{CIRC}}$, which is a propositional language that allows the nested use of circumscription, and of the propositional fragment of nested abnormality theories (NATs) that were proposed by Lifschitz [35] as an elegant circumscriptive framework for modularized knowledge representation. As we have shown, NATs can be regarded as a semantic fragment of $\mathcal{L}_{\mathrm{CIRC}}$. As it turned out, NATs and thus $\mathcal{L}_{\mathrm{CIRC}}$ are capable of expressing more difficult (in terms of complexity) problems than ordinary unnested circumscription, and can represent PSPACE-complete problems. Furthermore, we have identified fragments of NATs which have lower complexity, where we focused on generalizations of Horn CNFs, such as Horn logic programs and the DATALOG^{Circ} query language [10]. In particular, we have provided an efficiently computable normal form for nested logic programs. Finally, we have compared nested circumscription to other generalizations of circumscription.

Our results give a clear picture of the complexity situation, and reveal nesting and the use of local variables in NATs as sources of complexity. This gives useful insight into the complexity of $\mathcal{L}_{\mathrm{CIRC}}$ formulas and NATs, which is useful for understanding their computational nature and requirements. For example, it can be fruitfully exploited in considerations on eliminating nestings, or on changes to the set of defined letters in a NAT. To give a concrete example, suppose we have an extended Horn NAT $\mathcal{T}$ which has nesting depth one. Then, by Theorem 6.5, inference of a formula φ from $\mathcal{T}$ is Π_2^P-complete in general, and thus can be polynomially transformed to a standard circumscriptive theorem prover. If, moreover, the blocks inside $\mathcal{T}$ have no fixed letters and do not use $\max$, then by Theorem 5.1 we can efficiently eliminate nesting from $\mathcal{T}$, and transform inference $\mathcal{T} \models \varphi$ via a standard Horn circumscription to a SAT solver in polynomial time.

While we have addressed and resolved the main issues concerning the complexity of nested circumscription in a propositional setting in this paper, several issues remain for future work:

- On the complexity side, our study may be extended to cover further fragments of NATs and $\mathcal{L}_{\mathrm{CIRC}}$ besides the ones considered in this paper. Besides Horn theories, other syntactic fragments were e.g. considered in [9], which provides a good starting point for such a programme. Furthermore, a detailed study of

the complexity of nested circumscription in the first-order case and restricted fragments (monadic theories, etc) would be interesting.

• Complementing the results on reasoning complexity, Cadoli *et al.* [7, 6], Gogic *et al.* [26], Selman and Kautz [46], Darwiche and Marquis [14, 13] and others have studied representability issues among KR formalisms, considering problems like representing theories in one KR formalism with polynomial resources in another target formalism, such that the set of models or certain inference relations are preserved. In particular, "knowledge compilation," whose idea is that off-line preprocessing with high computational resources might help to speed up on-line reasoning, and make sometimes intractable problems tractable, has been attracting attention during the last years (see [5] for an initial survey). A study of representation and compilability aspects of $\mathcal{L}_{\mathrm{CIRC}}$ and NATs, and a comparison to other KR formalisms remains as an interesting issue. In particular, it would be interesting to determine under which circumstances NATs can be compiled in other NATs with lower nesting.

• An important instance of the issue in the previous paragraph is when a NAT can be efficiently replaced by an equivalent standard or prioritized circumscription, or even by an ordinary propositional formula. Notice that this issue is highly significant for algorithms that implement NATs on top of circumscriptive theorem provers or classical SAT solvers. Our results give a very preliminary answer to this question, by showing that this is, e.g., possible for Horn NATs without fixed letters. However, other and more expressive fragments might be identified which have this property.

• Finally, it remains to develop efficient algorithms and methods for computing NATs, either by reduction to an engine for some related KR formalism or logic, or by designing genuine algorithms. Su's CS program [49] and Doherty *et al.*'s DLS algorithm [16, 28], which handle the case of predicate logic, are incomplete in general and presumably not highly efficient in the propositional context. The use of QBF solvers (e.g., [11, 44, 23]) is here a suggestive starting point for obtaining more suitable systems.

As we believe, addressing these issues is worthwhile since nesting circumscriptions is a natural generalization of circumscription, and yields, as shown by our results, a simple yet expressive knowledge representation formalism for encoding reasoning tasks with complexity in PSPACE.

Acknowledgments

This work was supported by the Austrian Science Fund (FWF) Project Z29-INF. We are grateful to the referees of the preliminary conference version of this paper, which had a number of useful suggestions for improvements.

References

[1] C. Baral. Abductive reasoning through filtering. *Artificial Intelligence*, 120(1):1–28, 2000.

[2] C. Baral, A. Gabaldon, and A. Provetti. Value minimization in circumscription. In L. C. Aiello, J. Doyle, and S. Shapiro, editors, *KR'96: Principles of Knowledge Representation and Reasoning*, pages 474–481. Morgan Kaufmann, San Francisco, California, 1996.

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

[4] 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.

[5] M. Cadoli and F. Donini. A survey on knowledge compilation. *AI Communications*, 10(3-4):137–150, 1997.

[6] M. Cadoli, F. Donini, and M. Schaerf. Space efficiency of propositional knowledge representation formalisms. *Journal of Artificial Intelligence Research*, 13:1–31, 2000.

[7] M. Cadoli, F. M. Donini, P. Liberatore, and M. Schaerf. Feasibility and unfeasibility of off-line processing. In *Proceedings of the Fourth Israeli Symposium on Theory of Computing and Systems (ISTCS'96)*, pages 100–109. IEEE Computer Society, 1996.

[8] M. Cadoli, T. Eiter, and G. Gottlob. An efficient method for eliminating varying predicates from a circumscription. *Artificial Intelligence*, 54:397–410, 1992.

[9] M. Cadoli and M. Lenzerini. The complexity of propositional closed world reasoning and circumscription. *Journal of Computer and System Sciences*, 43:165–211, 1994.

[10] M. Cadoli and L. Palopoli. Circumscribing DATALOG: Expressive power and complexity. *Theoretical Computer Science*, 193:215–244, 1998.

[11] M. Cadoli, M. Schaerf, A. Giovanardi, and M. Giovanardi. An algorithm to evaluate quantified Boolean formulae and its experimental evaluation. *Journal of Automated Reasoning*, 28:101–142, 2002.

[12] E. Dantsin, T. Eiter, G. Gottlob, and A. Voronkov. Complexity and expressive power of logic programming. *ACM Computing Surveys*, 33(3):374–425, 2001.

[13] A. Darwiche. Decomposable negation normal form. *Journal of the ACM*, 48(4):608–647, 2001.

[14] A. Darwiche and P. Marquis. A perspective on knowledge compilation. In B. Nebel, editor, *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, pages 175–182. Morgan Kaufmann, 2001.

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

[16] P. Doherty, W. Lukaszewicz, and A. Szalas. Computing circumscription revisited: A reduction algorithm. *Journal of Automated Reasoning*, 18(3):297–336, 1997.

[17] T. Eiter, W. Faber, N. Leone, and G. Pfeifer. Declarative problem-solving using the DLV system. In J. Minker, editor, *Logic-Based Artificial Intelligence*, pages 79–103. Kluwer Academic Publishers, 2000.

[18] 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. Addendum 118:315.

[19] T. Eiter and G. Gottlob. On the Complexity of Theory Curbing. In M. Parigot and A. Voronkov, editors, *Proceedings 7th International Conference on Logic for Programming and Automated Reasoning (LPAR 2000), Reunion Island, France*, number 1955 in LNCS, pages 1–19. Springer, 2000.

[20] T. Eiter, G. Gottlob, and Y. Gurevich. Curb your theory ! A circumscriptive approach for inclusive interpretation of disjunctive information. In R. Bajcsy, editor, *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 634–639. Morgan Kaufman, 1993.

[21] T. Eiter, G. Gottlob, and H. Veith. Modular logic programming and generalized quantifiers. In J. Dix, U. Furbach, and A. Nerode, editors, *Proceedings of the 4th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-97)*, number 1265 in LNCS, pages 290–309. Springer, 1997.

[22] W. Faber and G. Pfeifer. `dlv` homepage, since 1996. `http://www.dbai.tuwien.ac.at/proj/dlv/`.

[23] R. Feldmann, B. Monien, and S. Schamberger. A distributed algorithm to evaluate quantified Boolean formulae. In *Proceedings National Conference on AI (AAAI'00)*, Austin, Texas, July 30-August 3 2000. AAAI Press.

[24] E. Giunchiglia, G. N. Kartha, and V. Lifschitz. Representing action: Indeterminacy and ramifications. *Artificial Intelligence*, 95(2):409–438, 1997.

[25] E. Giunchiglia and N. K. V. Lifschitz. Actions with indirect effects (extended). Technical Report MRG/DIST # 95-0028, Università di Genova, Dipartimento Informatica Sistemistica Telematica, 1995. Also in: Proc. AAAI Spring Symposium 95 on Extending Theories of Actions.

[26] G. Gogic, H. Kautz, C. H. Papadimitriou, and B. Selman. The comparative linguistics of knowledge representation. In C. Mellish, editor, *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI '95)*, pages 862–869. Morgan Kaufmann, 1995.

[27] Y. Gurevich and S. Shelah. Fixpoint extensions of first-order logic. *Annals of Pure and Applied Logic*, 32:265–280, 1986.

[28] J. Gustafsson. An implementation and optimization of an algorithm for reducing formulae in second-order logic. Technical Report LiTH-MAT-R-96-04, Dept. of Mathematics, Linköping University, 1996.

[29] N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.

[30] D. S. Johnson. A catalog of complexity classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A, chapter 2. Elsevier Science, 1990.

[31] G. N. Kartha and V. Lifschitz. Actions with indirect effects (preliminary report). In J. Doyle, E. Sandewall, and P. Torasso, editors, *Proc. KR-94*, pages 341–350. Morgan Kaufmann, San Francisco, California, 1994.

[32] G. N. Kartha and V. Lifschitz. A simple formalization of actions using circumscription. In C. Mellish, editor, *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI '95)*, pages 1970–1975, San Francisco. Morgan Kaufmann, 1995.

[33] V. Lifschitz. Computing circumscription. In *Proceedings International Joint Conference on Artificial Intelligence (IJCAI-85)*, pages 121–127, 1985.

[34] V. Lifschitz. Circumscription. In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume III, pages 297–352. Clarendon Press, Oxford, 1994.

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

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

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

[38] M. Minoux. LTUR: a simplified linear time unit resolution for Horn formulae and computer implementation. *Information Processing Letters*, 29:1–12, 1988.

[39] G. Neelakantan Kartha. A circumscriptive formalization of the qualification problem. In B. Nebel, editor, *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, pages 139–144. Morgan Kaufmann, 2001.

[40] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[41] C. Pollett and J. Remmel. Nonmonotonic reasoning with quantified Boolean constraints. In *Proceedings of the 4th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-97)*, number 1265 in LNCS, pages 18–39. Springer, 1997.

[42] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.

[43] E. Remolina and B. Kuipers. A logical account of causal and topological maps. In B. Nebel, editor, *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, pages 5–11. Morgan Kaufmann, 2001.

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

[45] J. Schlipf. Decidability and definability with circumscription. *Annals of Pure and Applied Logic*, 35:173–191, 1987.

[46] B. Selman and H. Kautz. Knowledge compilation and theory approximation. *Journal of the ACM*, 43(2):193–224, 1996.

[47] J. Sierra-Santibáñez. Declarative formalization of strategies for action selection. In *Proceedings 7th Workshop on Nonmonotonic Reasoning (NMR-2000), Action and Causality track, Trento, Italy*, 1998.

[48] T. C. Son and C. Baral. Formalizing sensing actions: A transition function based approach. *Artificial Intelligence*, 125(1-2):19–91, 2001.

[49] Z. Su. Automating the computation of nested abnormality theories. Undergraduate Honors Thesis, Department of Computer Sciences, University of Texas at Austin, 1995.

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