Arch. Math. Logic (2015) 54:685–710
DOI 10.1007/s00153-015-0435-x

Mathematical Logic

On the parameterized complexity of non-monotonic logics

Arne Meier[1] · Irina Schindler[1] · Johannes Schmidt[2] · Michael Thomas[1] · Heribert Vollmer[1]

Received: 7 August 2014 / Accepted: 10 June 2015 / Published online: 24 June 2015

Abstract We investigate the application of Courcelle's theorem and the logspace version of Elberfeld et al. in the context of non-monotonic reasoning. Here we formalize the implication problem for propositional sets of formulas, the extension existence problem for default logic, the expansion existence problem for autoepistemic logic, the circumscriptive inference problem, as well as the abduction problem in monadic second order logic and thereby obtain fixed-parameter time and space efficient algorithms for these problems. On the other hand, we exhibit, for each of the above problems, families of instances of a very simple structure that, for a wide

This work is supported in part by DFG Grant VO 630/6-2 and ME 4279/1-1. Part of this work has been published in a preliminary form in: A. Meier and J. Schmidt and M. Thomas and H. Vollmer, On the Parameterized Complexity of Default Logic and Autoepistemic Logic, Proc. LATA 2012, pp. 389–400, vol. 7183 LNCS.

✉ Arne Meier
meier@thi.uni-hannover.de

Irina Schindler
schindler@thi.uni-hannover.de

Johannes Schmidt
johannes.schmidt@liu.se

Michael Thomas
thomas@thi.uni-hannover.de

Heribert Vollmer
vollmer@thi.uni-hannover.de

1 Institut für Theoretische Informatik, Gottfried Wilhelm Leibniz Universität Hannover, Appelstrasse 4, 30167 Hannover, Germany

2 Department of Computer and Information Science, Linköpings Universitet, 581 83 Linköping, Sweden

range of different parameterizations, do not have efficient fixed-parameter algorithms (even in the sense of the large class XP_{nu}, resp., XL_{nu}) under standard complexity assumptions.

Keywords Abduction · Autoepistemic logic · Default logic · Circumscription · Parameterized complexity · Courcelle's theorem · Monadic second order logic

Mathematics Subject Classification 03D15 · 03B42

1 Introduction

Non-monotonic reasoning formalisms were introduced in the 1970s as a formal model for human reasoning and have developed into one of the most important topics in computational logic and artificial intelligence. However, as it turns out, most interesting reasoning tasks are computationally intractable already for propositional versions of non-monotonic logics [18], in fact presumably much harder than for classical propositional logic. Because of this, a lot of effort has been spent to identify fragments of the logical language for which at least some of the algorithmic problems allow efficient algorithms; a survey of this line of research can be found in [37]. Probably the most prominent non-monotonic concepts are abduction, autoepistemic logic, circumscription, and default logic.

The method of abduction has been deeply studied by Peirce [32] and Hartshorne et al. [33], and fundamentally influenced several areas in and around artificial intelligence as described by Eiter and Gottlob [12] as well as by Creignou et al. [10]. Informally the reasoning problem is defined through an observation creating a hypothesis and the search for an explanation.

Circumscription has been firstly introduced by McCarthy [27]. Initially this logic was invented as a first order variant but has been formalized for other logics later. The main idea of this non-monotonic concept is to model the idea of general assumptions, i.e., when making a statement everything that is not stated explicitly is assumed to be false.

Nowadays Circumscription has become one of the most developed and well studied formalisms for non-monotonic reasoning [6,23,31,36]. Lifschitz [24] has proven that reasoning in Circumscription is equivalent to reasoning under the extended closed world assumption, i.e., the assumption of $\neg p$ whenever $p \in P$ is not derivable for some

Table 1 Overview of non-monotonic logics with their different concepts

Logic	Concept	Introduced by
Abduction	Observations and explanations	Hartshorne et al. [33] and Peirce [32]
Autoepistemic logic	Modal operator L	Moore [28]
Circumscription	Minimality Semantics	McCarthy [27]
Default logic	Inference rules $\frac{\alpha:\beta}{\gamma}$	Reiter [35]

set P [17]. From a computational complexity aspect the circumscriptive inference problem is well understood. Generally the problem has been classified to be complete for the second level of the polynomial hierarchy, $\Pi_2^P = \text{coNP}^{\text{NP}}$, by Cadoli and Lenzerini [6] who showed containment, and by Eiter and Gottlob [11] who proved its hardness.

In this paper a different approach is chosen to deal with hard problems, namely the framework of parameterized complexity. Gottlob et al. [19] made it clear that the treewidth of a (suitable graph theoretic encoding of a) given knowledge base is a useful parameter in this context: making use of Courcelle's theorem it was shown that many reasoning tasks for logical formalisms such as circumscription, abduction, and logic programming become tractable in the case of bounded treewidth.

As we also focus on two of the logics classified by Gottlob et al. [19] let us briefly discuss the difference to our approach. Gottlob et al. aim for efficient algorithms which usually is not a general benefit of exploiting Courcelle's theorem. They also explain why this is the case in the sense of state size explosion of the resulting FTAs. However they use monadic datalog instead to reach their goal. By this approach all proven results in their paper (which obtain FPT results though Courcelle's theorem) deal with formulas in conjunctive normal form, or to be more precise, with the satisfiability equivalent formulas in conjunctive normal form. Therefore they have to use clause and literal predicates in their vocabulary to express the problems on this type of formulas. In particular, their conjunctive normal form approach appears in nearly every constructed formula wherefore it is simply not possible to plug in our construction from Lemma 1 and use their other formulas to obtain general results for formulas out of CNF—circumvention of this problem would yield the modification of several formulas.

In contrast to their approach, we will work with arbitrary formulas over any restricted set of allowed Boolean connectives. We do not construct any satisfiability equivalent representations in CNF wherefore we also have to embody the more complex formula structure in the construction of the MSO formulas (which alas have to be more complex as well).

Although the obtained results of Gottlob et al. in comparison to our results may convey the impression that they are similar at first sight, the difference is rather fundamental. For instance, if one wants to consider as a natural next step in this line of research the question about counting complexity in this setting, the results from Gottlob et al. cannot be used as satisfiability equivalent formulas do not maintain the counting complexity in general. However we admit that such general notions are more of theoretical interest and may not be used to obtain time or space efficient algorithms in the sense as Gottlob et al. did. Nevertheless we think that it is good to have a different and more general point of view on the results as we did than previously existed.

Additionally to our investigation of Circumscription and Abduction, we here examine a family of non-monotonic logics where the semantics of a given set of formulas (axioms, knowledge base) is defined in terms of a fixed-point equation. In particular we turn to default logic [35] and autoepistemic logic [28]. In the first, human reasoning is mimicked using "default rules" (in the absence of contrary information, assume this and that); in the second, a modal operator is introduced to model the beliefs of a perfect rational agent. For both logics the algorithmic tasks of satisfiability and reasoning have been shown to be complete in the second level of the polynomial hierarchy [18]. Also the so to speak Boolean fragments of the important decision problems in these two logics have been completely classified with respect to its computational complexity [2,9].

Much in the vein of [19] we here examine the parameterized complexity of all these non-monotonic problems and, making again use of Courcelle's theorem [8] and a recent variant by Elberfeld et al., we obtain time and space efficient algorithms if the treewidth of the given knowledge base is bounded. This underlines once again how important this parameter is. Let us shortly sketch the intuition behind Courcelle's theorem. Assume we are able to express a problem Q in MSO. If instances $x \in Q$ can be modeled via some relational structure $\mathscr{A}_x$ over some finite vocabulary τ and we see Q as a parameterized problem (Q, κ) where κ is the treewidth of $\mathscr{A}_x$ then by Courcelle's theorem we immediately obtain that (Q, κ) is in FPT.

Previously the parameterized complexity of non-monotonic reasoning has gained attention by three further publications we do like to mention explicitly. First, Gottlob et al. [20] have taken an approach via constrained satisfaction problems (CSPs) with respect to logic programs and model enumeration (here they focus on a version for circumscription). Second, Zhao and Ding [38] classified disjunction-free default reasoning to be fixed-parameter tractable. Last to mention, recently in 2012, Fellows et al. [15] took a deep look at the parameterized complexity landscape of Abuction. Here they distinguished between several different kinds of Horn and Krom variants of the problem and classified fragments also to be complete in the W-hierarchy. Also a good survey article is available by Gottlob and Szeider [21].

Another non-montonic formalism which received a lot of interest is answer set programming [26,30]. This concept is closely related to the declarative language Prolog. Basically one encodes solutions to a problem into the models of a program such that solutions are corresponding to terms of rules and constraints. The parameterized complexity has been again investigated by Gottlob et al. [19]. Also the parameterized view but from a more efficiency focused perspective has been considered by Jakl et al. who examined counting versions and enumeration tasks of answer set programming [22].

A second contribution of our paper concerns lower bounds: Under the assumption $\mathrm{P} \neq \mathrm{NP}$ we show that, even for certain families of very simple knowledge bases and for any parameterization taken from a broad variety, no efficient fixed-parameter algorithms exist, not even in the sense of the quite large parameterized class $\mathrm{XP_{nu}}$. These simple families of knowledge bases are defined in terms of severe syntactic restrictions, e.g., using default rules with literals or propositions only. Restricting the input structure even further we obtain that no fixed-parameter algorithm in the sense of the space-bounded class $\mathrm{XL_{nu}}$ (the logarithmic space analogue of $\mathrm{XP_{nu}}$) exists, unless $\mathrm{L} = \mathrm{NL}$.

Unfortunately, treewidth is not among the parameters for which our lower bound can be proven—otherwise we would have proven $\mathrm{P} \neq \mathrm{NP}$. In a third part of our paper, we show that those structurally very simple families of knowledge bases, for which we gave our lower bounds, already have unbounded treewidth. For this result, we introduce the notion of pseudo-cliques, stemming from the application of edge contractions, yielding some kind of topological minors, and show how to embed these into our graph-theoretic encodings of knowledge bases.

2 Preliminaries

Complexity Theory In this paper we will make use of several standard notions of complexity theory such as the complexity classes L, ⊕L, NL, P, NP, coNP, and Σ_2^{p} and their completeness notions under logspace-many-one $\leq_{\mathrm{m}}^{\log}$ reductions.

A *parameterized problem* is a set $P \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet. Given an instance $(x, k) \in \Sigma^* \times \mathbb{N}$, we refer to x as the *input* and to k as the *parameter*. Further we refer to the value of k by using a polynomial-time computable function $\kappa : \Sigma^* \to \mathbb{N}$ which maps x to k; we then say κ is a *parameterization* of P. Also we will use k interchangeably with $\kappa(x)$ for some input x if the context is clear.

Given a problem P and a parameterization κ, (P, κ) belongs to the class FPT iff there is a deterministic algorithm solving P in time $f(\kappa(x)) \cdot |x|^{O(1)}$ where x is an input of P and f is an arbitrary recursive function; (P, κ) is said to be *fixed parameter tractable* then. If (P, κ) is a parameterized problem, then $(P, \kappa)_\ell := \{x \in P \mid \kappa(x) = \ell\}$ is the ℓ-th slice of (P, κ). Define (P, κ) to be a member of $\mathrm{XP}_{\mathrm{nu}}$ (in words, XP nonuniform) iff $(P, \kappa)_\ell \in \mathrm{P}$ for all $\ell \in \mathbb{N}$; similarly the class $\mathrm{XL}_{\mathrm{nu}}$ is defined over logspace machines. For background on parameterized complexity we recommend [16].

For a long time in parameterized complexity theory the most interesting results were concerned with the parameterized time classes. Here most of our results in this paper connected to parameterized space classes deal with the class XL which is defined as follows similarly to Chen et al. [7]. A given parametrized problem (P, κ) over Σ is in XL if there exists a deterministic algorithm deciding $x \in P$ within space $f(\kappa(x)) \cdot O(\log |x|)$ for an arbitrary function f. The class XL is different to FPT (under standard assumptions), e.g., there are complete problems in XL which are not in FPT as shown by Elberfeld et al. [14].

Clones Let B be a finite set of Boolean functions. B is called a *clone* if it is closed under superposition (i.e., arbitrary compositions and projections of functions). With $[B]$ we denote the smallest clone containing B. If B is a set of Boolean functions, then a B-formula φ only uses functions from B. The underlying framework of Post's lattice is a fundamental basis here [34].

Propositional logic Let φ, ψ be propositional formulas. We say ψ is *implied* by φ (or ψ is a *consequence* of φ), in symbols $\varphi \models \psi$, if for every assignment θ such that $\theta \models \varphi$ it holds $\theta \models \psi$. Further if A is a set of propositional formulas, then we define $\mathrm{Th}(A) := \{\psi \mid \varphi \models \psi, \varphi \in A\}$ as the set of all consequences of the set A. If B is a finite set of Boolean functions then we denote with $\mathscr{L}(B)$ the set of propositional formulas which only use connectives from B.

Monadic second order logic A *vocabulary* is a set of *relation symbols*, where each relation symbol has a finite arity $k \geq 1$; if $k = 1$ then we say the relation is *unary* (then we also say the relation is a set). A *structure* $\mathscr{A}$ over a vocabulary τ consists of a *universe* A which is a non-empty set, and a relation $A^{\mathscr{A}} \subseteq A^k$ for each relation symbol R, where k denotes the arity of R. Usually we write R^k to denote that relation R has arity $k \in \mathbb{N}$.

Monadic second order logic (or MSO) is the fragment of second order logic in which only quantification over unary relations is allowed. Further we will use the following notion: if M is a set then let $M(x)$ be $\top$ if $x \in M$ holds and be $\bot$ otherwise.

Treewidth A *tree decomposition* of a structure $\mathscr{A}$ (with universe A) is a pair (T, X), where $X = \{B_1, \ldots, B_r\}$ is a family of subsets of A (the set of *bags*), and T is a tree whose nodes are the bags B_i, satisfying the following conditions:

(i) $\bigcup X = A$, i.e., every element of the universe appears in at least one bag,
(ii) for each tuple $(a_1, \ldots, a_k)$ appearing in a relation of $\mathscr{A}$, there exists a $B \in X$: $\{a_1, \ldots, a_k\} \in B$, i.e., every tuple is 'contained' in a bag, and
(iii) $\forall a \in A : \{B \mid a \in B\}$ forms a connected subtree T, i.e., for every element a the set of bags containing a is connected in T.

The *width* of a decomposition (T, X), $width(T, X)$, is the number $\max\{|B| \mid B \in X\} - 1$, i.e., the size of the largest bag minus 1. The *treewidth* of a structure $\mathscr{A}$, $tw(\mathscr{A})$, is the minimum of the widths of the tree decompositions of $\mathscr{A}$.

3 MSO-Encodings

The aim of this paper is the application of Courcelle's theorem for obtaining fixed-parameter algorithms in the context of non-monotonic reasoning. For this, we will have to describe the relevant decision problems by monadic second-order formulas. In this section we will explain how to do this and obtain a preliminary result for the implication problem.

Now fix a finite set B of Boolean functions. Denote by τ_B the vocabulary $\{const^1_f \mid f \in B, \text{arity}(f) = 0\} \cup \{conn^2_{f,i} \mid f \in B, 1 \le i \le \text{arity}(f)\}$. With respect to a set Γ of propositional B-formulas we associate a $\tau_{B,prop}$-structure $\mathscr{A}_\Gamma$ where $\tau_{B,prop} := \tau_B \cup \{var^1, repr^1\}$ such that the universe of $\mathscr{A}_\Gamma$ is the set of subformulas of the formulas in Γ, and

1. $var(x)$ holds iff x represents a variable,
2. $repr(x)$ holds iff x represents a formula from Γ,
3. $const^1_f(x)$ holds iff x represents the constant f, and
4. $conn^2_{f,i}(x, y)$ holds iff x represents the ith argument of the function f at the root of the formula tree represented by y.

Lemma 1 *Let B be a finite set of Boolean functions. Then there exists an MSO-formula θ_{sat} over $\tau_{B,prop}$ such that for any $\Gamma \subseteq \mathscr{L}(B)$ it holds that Γ is satisfiableiff $\mathscr{A}_\Gamma \models \theta_{sat}$.*

Proof The formula θ_{struc} defined as follows states that if an element is not representing a formula $\varphi \in \Gamma$, then there must be at least one subformula in which it occurs. If an element is not a variable, then it represents either a constant or a Boolean function $f \in B$ and needs to have corresponding arity(f) elements.

$$\theta_{struc} := \forall x \left(\neg repr(x) \rightarrow \exists y \left(\neg var(y) \wedge \bigvee_{\substack{f \in B, \\ 1 \le i \le \text{arity}(f)}} conn_{f,i}(x, y) \right) \right) \wedge \forall x \Big(\neg var(x) \rightarrow$$

$$\bigvee_{\substack{f\in B,\\ \text{arity}(f)=0}} const_f(x) \oplus \bigvee_{f\in B} \bigwedge_{1\le i\le \text{arity}(f)} \exists y\big(conn_{f,i}(y,x) \land \forall z(conn_{f,i}(z,x) \to z = y)\big)\Big).$$

Let n denote the maximal arity of B, i.e., $n := \max\{\text{arity}(f) \mid f \in B\}$.

$$\theta_{assign}(M) := \forall x \forall y_1 \cdots \forall y_n \bigwedge_{f\in B} \Big(\bigwedge_{\text{arity}(f)=0} const_f(x) \to (M(x) \leftrightarrow f)$$
$$\land \bigwedge_{1\le i\le \text{arity}(f)} conn_{f,i}(y_i, x) \to \big(M(x) \leftrightarrow f(M(y_1), \ldots, M(y_{\text{arity}(f)}))\big)\Big).$$

Now define

$$\theta_{\exists assign} := \exists M\big(\theta_{assign}(M) \land \forall x\Big(repr(x) \to M(x)\Big).$$

It is easy to verify that $\theta_{sat} := \theta_{struc} \land \theta_{\exists assign}$ satisfies the lemma. □

Let B be a finite set of Boolean functions and F, G be sets of B-formulas. Answering the implication problem of sets of propositional formulas, i.e., the question whether $F \models G$, requires to extend our vocabulary $\tau_{B,prop}$ to $\tau_{B,imp} := \tau_{B,prop} \cup \{repr^1_{prem}, repr^1_{conc}\}$ as well as our structure which we will denote by $\mathscr{A}_{F,G}$: $repr_{prem}(x)$ is true iff x represents a formula from F, and $repr_{conc}(x)$ is true iff x represents a formula from G. Now it is straightforward to formalize implication.

Lemma 2 *Let B be a finite set of Boolean functions. Then there exists an MSO-formula θ_{imp} over $\tau_{B,imp}$ such that for any $\Gamma \subseteq \mathscr{L}(B)$ and any $F, G \subseteq \Gamma$ it holds that $F \models G$ iff $\mathscr{A}_{F,G} \models \theta_{imp}$.*

Proof Define the MSO-formulas $\theta_{premise}(M)$, $\theta_{conclusion}(M)$, and $\theta_{implies}$ as follows:

$$\theta_{premise}(M) := \forall x\big(repr_{prem}(x) \to M(x)\big),$$
$$\theta_{conclusion}(M) := \forall x\big(repr_{conc}(x) \to M(x))\big),$$
$$\theta_{implies} := \forall M\Big(\big(\theta_{assign}(M) \land \theta_{premise}(M)\big) \to \theta_{conclusion}(M)\Big).$$

Then, we can define the formula θ_{imp} as $\theta_{imp} := \theta_{struc} \land \theta_{implies}$, where θ_{struc} and θ_{assign} are defined as above in Lemma 1. □

The application of Courcelle's Theorem [8] and the logspace version of Elberfeld et al. [13] directly leads to the following theorem.

Theorem 1 *Let B be a finite set of Boolean functions, let $k \in \mathbb{N}$ be fixed, and let F, G be sets of B-formulas such that $\mathscr{A}_{F,G}$ has treewidth bounded by k. Then the implication problem for sets of B-formulas is solvable in time $O(f(k) \cdot (|F| + |G|))$ and space $O(\log(f(k)) \cdot \log(|F| + |G|))$, where f is a recursive function.*

In other words, the implication problem of sets of formulas parameterized by the treewidth of $\mathscr{A}_{F,G}$ is fixed-parameter tractable, and in XL. In the following sections we will extend this result to default logic and autoepistemic logic.

4 Default logic

For a deep introduction into the area of non-monotonic logic we refer to [4,25]. In the following we will capture only the important parts which are relevant to the investigated logics and formalisms. Following Reiter [35], a *default rule* is a triple $\frac{\alpha:\beta}{\gamma}$; α is called the *prerequisite*, β is called the *justification*, and γ is called the *conclusion*. If B is a set of Boolean functions, then $d = \frac{\alpha:\beta}{\gamma}$ is a B-default rule if α, β, γ are B-formulas, i.e., formulas that use only connectors for functions in B. A *B-default theory* (W, D) consists of a set of propositional B-formulas W and a set of B-default rules D.

Reiter also introduced a very intuitive notion of stable extensions (another around fix-point semantics can be found in the original paper of Reiter [35]). For a given default theory (W, D) and a set E of formulas, define $E_0 = W$ and $E_{i+1} = \mathrm{Th}(E_i) \cup \{\gamma \mid \frac{\alpha:\beta}{\gamma} \in D, \alpha \in E_i \text{ and } \neg\beta \notin E\}$. Then E is a *stable extension* of (W, D) if and only if $E = \bigcup_{i \in \mathbb{N}} E_i$, and the set $G = \{\frac{\alpha:\beta}{\gamma} \in D \mid \alpha \in E \wedge \neg\beta \notin E\}$ is called the set of *generating defaults*.

The so to speak satisfiability problem for default logic, here called *extension existence problem*, EXT, is the problem, given a theory (W, D), to decide if it has a stable extension. Gottlob [18] proved that this problem is Σ_2^{P}-complete.

Example 1 The default theory $(\emptyset, \{\frac{\top:x}{x}, \frac{\top:\neg x}{\neg x}\})$ has exactly two stable extensions, namely $E_1 = \mathrm{Th}(\{x\})$ and $E_2 = \mathrm{Th}(\{\neg x\})$, whereas the theory $(\emptyset, \{\frac{\top:x}{\neg x}\})$ has no stable extension. Further consider the default theory (W, D), where

$$W = \{\text{football}, \text{rain}, \text{cold} \wedge \text{rain} \rightarrow \text{snow}\}, \quad \text{and} \quad D = \left\{\frac{\text{football} : \neg\text{snow}}{\text{takesPlace}}\right\}.$$

Then (W, D) has exactly the stable extension $\mathrm{Th}(W \cup \{\text{takesPlace}\})$. However, if the considered default theory is $(W \cup \{\text{cold}\}, D)$ now, then the unique stable extension is $\mathrm{Th}(W)$. Therefore one says that the reasoning in this logic is non-monotonic, i.e., having more information (it is cold) lets you deduce less than before.

Let B be a finite set of Boolean functions. Write $W \Cup D$ as a shorthand for the set of formulas $W \cup \{\alpha, \beta, \gamma \mid \frac{\alpha:\beta}{\gamma} \in D\}$. To any B-default theory (W, D), we associate a $\tau_{B,dl} := \tau_{B,prop} \cup \{kb^1, def^1, prem^2, just^2, concl^2\}$-structure $\mathscr{A}_{(W,D)}$ such that the universe of $\mathscr{A}_{(W,D)}$ is the union of the set of subformulas of $W \Cup D \cup \{\neg\beta \mid \frac{\alpha:\beta}{\gamma} \in D\}$ together with a set corresponding to the defaults in D, the relations from $\tau_{B,prop}$ are interpreted as in Sect. 3, and $kb(x)$ holds iff x represents a formula from the knowledge base W, $def(x)$ holds iff x represents a default $d \in D$, $prem(x, y)$ (resp. $just(x, y)$, $concl(x, y)$) holds iff x represents the premise α (resp. justification β, conclusion γ) and y represents the rule $\frac{\alpha:\beta}{\gamma}$.

Lemma 3 *Let B be a finite set of Boolean functions and let (W, D) be a B-default theory. There exists an MSO-formula $\theta_{extension}$ such that (W, D) possesses a stable extension iff $\mathscr{A}_{(W,D)} \models \theta_{extension}$.*

Proof First the formula θ_{isneg} expresses the fact that one formula is the negation of another formula: $\theta_{isneg}(\varphi, \overline{\varphi}) := \theta_{struc} \wedge \forall M\Big(\theta_{assign}(M) \rightarrow \big(M(\varphi) \leftrightarrow \neg M(\overline{\varphi})\big)\Big)$.

Observe that φ and $\overline{\varphi}$ are not formulas but placeholders for elements. The following two formulas define the applicability of defaults, i.e., whether a premise α is entailed or a justification β is violated which uses the shortcut $\chi(C, M, x) := (kb(x) \vee C(x)) \rightarrow M(x)$:

$$\theta_{W \cup C \models \alpha}(C, \alpha) := \forall M \Big(\theta_{assign}(M) \rightarrow \forall x \Big(\chi(C, M, x) \rightarrow M(\alpha)\Big)\Big),$$

$$\theta_{W \cup C \models \neg\beta}(C, \beta) := \exists \overline{\beta} \exists M \Big(\theta_{assign}(M) \rightarrow \forall x \Big(\chi(C, M, x) \wedge M(\overline{\beta}) \wedge \theta_{isneg}(\beta, \overline{\beta})\Big)\Big).$$

Now we can define the MSO-formulas θ_{app} (a default d is applicable), θ_{stable} (a set of defaults is stable), θ_{gd} (a set of defaults is generating) as follows.

$$\begin{aligned}
\theta_{app}(d, G) &:= \exists \alpha \exists \beta \exists C \Big(prem(\alpha, d) \wedge just(\beta, d) \wedge \\
&\quad \forall x \big(C(x) \leftrightarrow \exists y (G(y) \wedge concl(x, y))\big) \wedge \theta_{W \cup C \models \alpha}(C, \alpha) \wedge \neg \theta_{W \cup C \models \neg\beta}(C, \beta)\Big), \\
\theta_{stable}(G) &:= \forall d \big(def(d) \rightarrow (G(d) \leftrightarrow \theta_{app}(d, G))\big), \\
\theta_{gd}(G) &:= \theta_{stable}(G) \wedge \forall G' (G' \subsetneq G \rightarrow \neg \theta_{stable}(G')).
\end{aligned}$$

Then $\theta_{extension} := \theta_{struc} \wedge \exists G(\theta_{gd}(G))$ is true under $\mathscr{A}_{(W,D)}$ iff (W, D) has a stable extension. □

As a consequence of Lemma 3, we obtain from Courcelle's Theorem [8] and the logspace version of Elberfeld et al. [13] that, given the treewidth of $\mathscr{A}_{(W,D)}$ as a parameter, the extension existence problem for default logic is fixed-parameter tractable, and in XL.

Theorem 2 *Let B be a finite set of Boolean functions, let $k \in \mathbb{N}$ be fixed, and let (W, D) be a B-default theory such that $\mathscr{A}_{(W,D)}$ has treewidth bounded by k. Then the extension existence problem for B-default logic is solvable in time $O(f(k) \cdot |(W, D)|)$ and space $O(f(k) \cdot \log |(W, D)|)$, where f is a recursive function.*

So again and maybe with no big surprise, similar to the study by Gottlob et al. [19] for different non-monotonic formalisms, we see here that bounding the treewidth of a default theory yields time and space efficient algorithms for satisfiability. In the following we want to contrast this with a strong lower bound. We consider knowledge bases with very simple defaults rules, namely consisting only of literals (and in a second step even only propositions). Then we consider any parameterization of the extension existence problem that is bounded for all knowledge bases that obey this restriction. It follows that even for these very restricted knowledge bases, the parameterized extension existence problem is not even in the class $\mathrm{XP_{nu}}$, unless $\mathrm{P} \neq \mathrm{NP}$.

We want to point out that this theorem comprises for example the usual parameterizations for SAT (in terms of, e.g., size of backdoor sets or formula treewidth): For all these, we have FPT-algorithms for propositional satisfiability, but still the extension existence problem is not in $\mathrm{XP_{nu}}$.

Theorem 3 *Let B be a finite set of Boolean functions such that $\neg \in [B \cup \{\top\}]$ and let $\mathbf{D}$ be the set of sets D of default rules such that each default $d \in D$ is composed of literals only. Further let κ be an arbitrary parameterization function for which there exists always a $c \in \mathbb{N}$ such that $\kappa\big((\emptyset, D)\big) < c$ holds whenever $D \in \mathbf{D}$. If* $\mathrm{P} \neq \mathrm{NP}$*, then the extension existence problem for B-default logic, parameterized by κ, is not contained in* $\mathrm{XP_{nu}}$.

Proof The reduction from SAT to default logic restricted to default theories with $W = \emptyset$ and default rules composed of literals only, shown in Lemma 5.6 of [2], proves that the extension existence problem of default logic restricted to theories of this kind (which will be denoted by $\textsc{Ext}'$) is NP-hard. Now let κ be such a parameterization and suppose $\mathrm{P} \neq \mathrm{NP}$. For contradiction assume $(\textsc{Ext}', \kappa) \in \mathrm{XP_{nu}}$. Hence, by definition of $\mathrm{XP_{nu}}$, it holds $(\textsc{Ext}', \kappa)_\ell \in \mathrm{P}$ for every $\ell \in \mathbb{N}$. As also $\ell < c$ holds we can compose a deterministic polynomial time algorithm which solves $\textsc{Ext}'$. This contradicts $\mathrm{P} \neq \mathrm{NP}$ and concludes the proof. □

Theorem 4 *Let B be a finite set of Boolean functions such that $\bot \in [B]$ and let $\mathbf{D}$ be the set of sets D of default rules such that each default $d \in D$ is composed of propositions or the constant $\bot$ only. Further let κ be an arbitrary parameterization function for which there exists always an $c \in \mathbb{N}$ such that $\kappa\big((W, D)\big) < c$ holds whenever $D \in \mathbf{D}$ and W consists of at most one proposition. If* $\mathrm{L} \neq \mathrm{NL}$*, then the extension existence problem for B-default logic, parameterized by κ, is not contained in* $\mathrm{XL_{nu}}$.

Proof The reduction from the graph accessibility problem to default logic restricted to default theories with $|W| \leq 1$ and default rules composed of propositions or the constant $\bot$ only, shown in Lemma 5.8 of [2], proves that the extension existence problem of default logic restricted to theories of this kind (which will be denoted by $\textsc{Ext}'$) is NL-hard. Following the argumentation in the proof of Theorem 3, we conclude for $\mathrm{L} \neq \mathrm{NL}$ and $(\textsc{Ext}', \kappa) \in \mathrm{XL_{nu}}$ that $(\textsc{Ext}', \kappa)_\ell \in \mathrm{L}$ holds for every ℓ. This eventually leads to the desired contradiction proving the theorem. □

5 Autoepistemic logic

Moore [28] introduced a new modal operator L stating that its argument is "believed" as an extension of propositional logic. Further the expression $L\varphi$ is treated as an atomic formula with respect to the consequence relation $\models$. Given a set of Boolean functions B, we define with $\mathscr{L}_{\mathrm{ae}}(B)$ the set of all autoepistemic B-formulas through $\varphi ::= p \mid f(\varphi, \ldots, \varphi) \mid L\varphi$ for f being a Boolean functions in B and a proposition p. If $\Sigma \subseteq \mathscr{L}_{\mathrm{ae}}(B)$, then a set $\Delta \subseteq \mathscr{L}_{\mathrm{ae}}(B)$ is a *stable expansion* of Σ if it satisfies the condition $\Delta = \mathrm{Th}(\Sigma \cup L(\Delta) \cup \neg L(\overline{\Delta}))$, where $L(\Delta) := \{L\varphi \mid \varphi \in \Delta\}$ and $\neg L(\overline{\Delta}) := \{\neg L\varphi \mid \varphi \notin \Delta\}$, and $L(\Delta), \neg L(\overline{\Delta}) \subseteq \mathscr{L}_{\mathrm{ae}}(B)$.

Let $\mathrm{SF}(\varphi)$ denote the set of subformulas of φ, let $\mathrm{SF}^L(\varphi)$ denote the set of those subformulas of φ that have prefix L, and let us use the shorthand $\neg S = \{\neg\varphi \mid \varphi \in S\}$ for a set of (autoepistemic) formulas S. By abuse of notation given a set of formulas S we define $\mathrm{SF}(S) := \{\mathrm{SF}(\varphi) \mid \varphi \in S\}$ be the set of subformulas of the formulas in S

(and analogously for $\mathrm{SF}^L()$). Given a set of autoepistemic B-formulas $\Sigma \subseteq \mathscr{L}_{\mathrm{ae}}(B)$, we say a set $\Lambda \subseteq \mathrm{SF}^L(\Sigma) \cup \neg\mathrm{SF}^L(\Sigma)$ is Σ-*full* if for each $L\varphi \in \mathrm{SF}^L(\Sigma)$ it holds $\Sigma \cup \Lambda \models \varphi$ iff $L\varphi \in \Lambda$, and $\Sigma \cup \Lambda \not\models \varphi$ iff $\neg L\varphi \in \Lambda$. The connection of Σ-full sets and stable expansions of Σ has been observed by Niemelä [29]: if $\Sigma \subseteq \mathscr{L}_{\mathrm{ae}}(B)$ is a set of autoepistemic formulas and Λ is a Σ-full set, then for every $L\varphi \in \mathrm{SF}^L(\Sigma)$ either $L\varphi \in \Lambda$ or $\neg L\varphi \in \Lambda$. The stable expansions of Σ and Σ-full sets are in one-to-one correspondence.

The *expansion existence problem*, Exp, is the problem, given a set of autoepistemic formulas Σ, to decide if it has a stable expansion. Again, Gottlob [18] proved that this problem is complete for the class Σ_2^{p}.

Example 2 (Example 3.2 in [9]) Consider the following set Σ_{car} of autoepistemic formulas formalizing knowledge about cars.

$$\begin{aligned}\Sigma_{\mathrm{car}} := \{&\mathrm{car}, \mathrm{threeWheeler} \rightarrow \mathrm{rickshaw}, \mathrm{car} \wedge \neg L\mathrm{threeWheeler} \rightarrow \mathrm{fourWheeler},\\ &L\mathrm{rickshaw} \rightarrow \mathrm{threeWheeler}\}\end{aligned}$$

The set Σ_{car} has two stable expansions: one being a superset of $\{\neg L\mathrm{threeWheeler}, \neg L\mathrm{rickshaw}\}$, and one being a superset of $\{L\mathrm{threeWheeler}, L\mathrm{rickshaw}\}$.

Indeed, if Lrickshaw is contained in a stable expansion Δ, then threeWheeler is derivable from the formulas in Σ_{car}, and by definition of stable expansions, $L\mathrm{threeWheeler} \in \Delta$. On the other hand, if Lrickshaw is not contained in Δ, then we cannot derive threeWheeler from $\mathrm{Th}(\Sigma \cup L(\Delta) \cup \neg L(\bar{\Delta}))$, which implies $\neg L$ threeWheeler $\in \neg L(\bar{\Delta}) \subseteq \Delta$. Thus, any stable expansion Δ has to satisfy L threeWheeler $\in \Delta$ iff $L\mathrm{rickshaw} \in \Delta$.

To see that the given sets characterize stable expansions, observe that

$$\{L\mathrm{rickshaw}, L\mathrm{threeWheeler}\} \subseteq \Delta$$

implies that rickshaw, threeWheeler $\in \mathrm{Th}(\Sigma \cup L(\Delta) \cup \neg L(\bar{\Delta})) = \Delta$, whereas

$$\{\neg L\mathrm{rickshaw}, \neg L\mathrm{threeWheeler}\} \subseteq \Delta$$

implies that neither rickshaw nor threeWheeler can be derived from $\Sigma \cup L(\Delta) \cup \neg L(\bar{\Delta})$. Thus both sets can be extended to yield a stable expansion.

Let B be a finite set of Boolean functions. To any set Σ of autoepistemic B-formulas, we associate a $\tau_{B,ae} := \tau_B \cup \{L^1, repr^1\}$-structure $\mathscr{A}_\Sigma$ such that the universe of $\mathscr{A}_\Sigma$ is the union of the set of subformulas of $\Sigma \cup \{\neg L\varphi \mid L\varphi \in \mathrm{SF}(\Sigma)\}$, the relations from τ_B are interpreted as in Sect. 3, and $L(x)$ holds iff the subformulas represented by x is prefixed by an L, and $repr(x)$ holds iff x represents a formula in Σ.

Lemma 4 *Let B be a finite set of Boolean functions and let Σ be a set of autoepistemic B-formulas. There exists an MSO-formula θ such that Σ possesses a stable expansion iff $\mathscr{A}_\Sigma \models \theta$.*

Proof For a set of formulas G and a formula φ, similar to $\theta_{W \cup C \models \alpha}(C, \alpha)$ in the proof of Lemma 3, define be the MSO-formula

$$\theta_{\Sigma \cup \Lambda \models \varphi}(\Lambda, \varphi) := \forall M \Big(\theta_{assign}(M) \rightarrow \forall x \Big(\big((repr(x) \vee \Lambda(x)) \rightarrow M(x)\big) \rightarrow M(\varphi)\Big)\Big)$$

to test for $\Sigma \cup \Lambda \models \varphi$. Now define the MSO-formula θ_{full} as

$$\theta_{full}(\Lambda) := \forall x \Big(L(x) \rightarrow \big(\Lambda(x) \oplus \exists y (conn_{\neg}(x, y) \wedge \Lambda(y))\big)\Big)$$
$$\wedge \forall x \Big(L(x) \rightarrow \big(\Lambda(x) \leftrightarrow \theta_{\Sigma \cup \Lambda \models \varphi}(\Lambda, x)\big)\Big).$$

Then $\theta := \theta_{struc} \wedge \exists \Lambda(\theta_{full}(\Lambda))$ is true under $\mathscr{A}_\Sigma$ iff Σ has a Σ-full set Λ, which is the case iff Σ has a stable expansion. □

As above we obtain from Lemma 4 that, given the treewidth of $\mathscr{A}_\Sigma$ as a parameter, the expansion existence problem for autoepistemic logic is fixed-parameter tractable, and in XL.

Theorem 5 *Let B be a finite set of Boolean functions, let $k \in \mathbb{N}$ be fixed, and let Σ be a set of autoepistemic B-formulas such that $\mathscr{A}_\Sigma$ has treewidth bounded by k. Then the expansion problem is solvable in time $O(f(k) \cdot |\Sigma|)$ and space $O(\log(f(k)) \cdot \log |\Sigma|)$.*

On the other hand, analogues of Theorems 3 and 4 are easily obtained:

Theorem 6 *Let B be a finite set of Boolean functions such that $\vee \in [B \cup \{\bot, \top\}]$ and let $\mathscr{S}$ be the set of sets Σ of autoepistemic B-formulas such that all $\varphi \in \Sigma$ are disjunctions of propositions or L-prefixed propositions. Further let κ be an arbitrary parameterization function for which there exists always an $c \in \mathbb{N}$ such that $\kappa(\Sigma) < c$ holds whenever $\Sigma \in \mathscr{S}$. If* P $\neq$ NP, *then the expansion existence problem for sets of autoepistemic B-formulas, parameterized by κ, is not contained in* $\mathrm{XP_{nu}}$.

Proof Observe that there exists a reduction f from 3-SAT to autoepistemic logic restricted to B-formulas shown in Lemma 4.5 of [9]. This implies our claim, as membership in $\mathrm{XP_{nu}}$ implies a polynomial-time algorithm for any fixed κ. □

Theorem 7 *Let B be a finite set of Boolean functions such that $\oplus, \top \in [B]$. Further let κ be an arbitrary parameterization function for which there exists always an $c \in \mathbb{N}$ such that $\kappa(\Sigma) < c$ holds for all Σ. If* L $\neq \oplus$L, *then the expansion existence problem for sets of autoepistemic B-formulas, parameterized by κ, is not contained in* $\mathrm{XL_{nu}}$.

Proof Observe that there exists a reduction f from the implication problem restricted to B-formulas shown in Lemma 4.8 of [1]. This implies our claim, as membership in $\mathrm{XL_{nu}}$ implies a logspace algorithm for any fixed κ. □

We remark that similar lower bounds as given for default logic in the previous section and for autoepistemic logic here hold for the implication problem as well as follows.

Theorem 8 *Let B be a finite set of Boolean functions such that $x \oplus y \oplus z \in [B]$ and let $\mathscr{G}$ be the set of sets Γ of formulas such that each formula $\varphi \in \Gamma$ is composed of functions $f(x, y, z) = x \oplus y \oplus z$ only. Further let κ be an arbitrary parameterization function for which there exists always an $c \in \mathbb{N}$ such that $\kappa\big(F, G\big) < c$ whenever both $F, G \in \mathscr{G}$. If* $\mathrm{L} \neq \oplus\mathrm{L}$*, then the implication problem for sets of B-formulas, parameterized by κ, is not contained in* $\mathrm{XL}_{\mathrm{nu}}$.

Proof From Lemma 4.4 in [1] it follows that the implication problem $\text{I}{\small MP}'$ for these sets of formulas is $\oplus$L-hard. Suppose $\mathrm{L} \neq \oplus\mathrm{L}$ and let κ be a parameterization such that $\kappa(F, G) < c$ for every $F, G \in \mathscr{G}$. Now following the argumentation of Theorem 3 yields a contradiction to $\mathrm{L} \neq \oplus\mathrm{L}$. □

Theorem 9 *Let B be a finite set of Boolean functions such that $\wedge, \vee \in [B]$. Let $\mathscr{G}_1$ be the set of sets Γ of formulas in monotone 2-CNF and let $\mathscr{G}_2$ be the set of sets Γ of formulas in DNF. Further let κ be an arbitrary parameterization function for which there exists always an $c \in \mathbb{N}$ such that $\kappa\big(F, G\big) < c$ whenever both $F \in \mathscr{G}_1, G \in \mathscr{G}_2$. If* $\mathrm{P} \neq \mathrm{NP}$*, then the implication problem for sets of B-formulas, parameterized by κ, is not contained in* $\mathrm{XP}_{\mathrm{nu}}$.

Proof From Lemma 4.2 in [1] it follows that the implication problem $\text{I}{\small MP}'$ for these sets of formulas is coNP-hard. Following an analogue argumentation as in the proof of Theorem 8 implies this theorem. □

6 Pseudo-Cliques

Looking at Theorems 3 and 4 one might hope that the syntactic restriction imposed there, namely allowing only defaults that involve literals or propositions, is so severe that it will bound the treewidth of every such input structure. Combining this with Theorem 2 would then yield $\mathrm{P} = \mathrm{NP}$ (or $\mathrm{L} = \mathrm{NL}$, resp.). Stated the other way round, if $\mathrm{P} \neq \mathrm{NP}$ then the treewidth of $\mathscr{A}_{(W,D)}$ is a non-trivial parameterization, i.e., a parameterization κ for which there exists no $c \in \mathbb{N}$ such that $\kappa((\emptyset, D)) < c$ holds for all D consisting of defaults rules involving only literals.

In the following we directly prove the non-triviality of the parameterization by treewidth (i.e., without any complexity hypothesizes). As a tool we utilize the subsequent definition of *pseudo-cliques*.

Definition 1 Let $G = (V, E)$ be an undirected graph. A *pseudo-clique* is a set of vertices $V' \subseteq V$ that can be partitioned into the set of *main-nodes* V_{main} and sets of *edge-nodes* $V_{u,v}$ for each $u \neq v \in V_{main}$ such that the following holds: for $v_1, \ldots, v_m \in V_{u,v}$ the nodes in $V_{u,v}$ form a simple path from u to v, i.e., it holds that $(u, v_1), (v_1, v_2), \ldots, (v_{m-1}, v_m), (v_m, v) \in E$ and no other edges are present.

The *size* of a pseudo-clique is $|V_{main}|$, i.e., the number of main-nodes. The *cardinality* of a pseudo-clique is $\max_{u \neq v \in V_{main}} |V_{u,v}|$, i.e., the length of the longest simple path between edge-nodes. A pseudo-clique is said to have *exact cardinality* k if $\forall u, v \in V_{main} : |V_{u,v}| = k$.

Observe that a pseudo-clique is a specific kind of topological minor which is a well-known concept in graph theory. The first four pseudo-cliques of exact cardinality

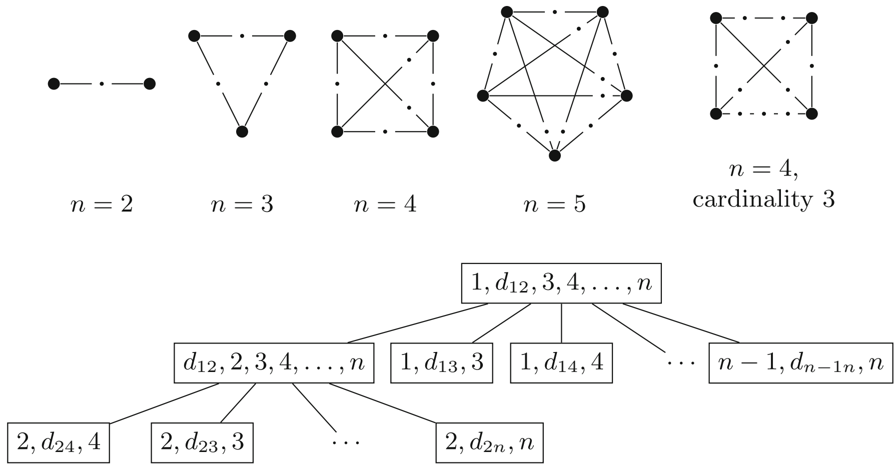

Fig. 1 Pseudo-cliques of exact card. 1 and size $\in \{2, \ldots, 5\}$, one of card. 3 and size 4, and a tree decomposition of a pseudo-clique of exact card. 1 and size n

1, and one of cardinality 3 are visualized in Fig. 1. The thick vertices correspond to the main-nodes whereas the small dots correspond to the edge-nodes.

The important fact for us is the observation that pseudo-cliques of size n have the same treewidth as the clique of size n. This observation immediately follows from the well-known technique of edge contractions, see e.g., [3, Lemma 16].

Whenever one wants to show that a parametrization by treewidth is non-trivial, the most obvious method is to show that the family of graphs has (sub-) cliques of arbitrary size. Hence talking about pseudo-cliques suffices. Corollary 1 (1.) shows that, for families used for the lower bounds in the previous sections, it is not possible to use cliques in order to prove unbounded treewidth and therefore additionally motivates the definition and purpose of pseudo-cliques.

Corollary 1 *Let B be a finite set of Boolean functions such that $\bot \in [B]$ and let $(\emptyset, D)$ be a B-default theory in the sense of Theorem 3, i.e., each default in D is composed of literals only. Then there exists an MSO formula θ fulfilling the property $(W, D) \in \text{EXT}(B)$ iff $\mathscr{A}_{(W,D)} \models \theta$, and*

1. *$\mathscr{A}_{(W,D)}$ is neither ℓ-connected nor contains a clique of size ℓ for any $\ell \geq 3$.*
2. *There exists a family of default theories $(\emptyset, D)_k$ such that the treewidth of $\mathscr{A}_{(\emptyset,D)_k}$ is not constant.*

Proof For (1.) we construct the MSO formula θ according to Lemma 3. At first observe that the universe of $\mathscr{A}_{(W,D)}$ comprises only literals and defaults. Further, there are no edges between literals, and no edges between defaults. Every default can be connected to at most three different literals. Obviously the graph does not contain a clique of size $\ell \geq 3$. Furthermore, the graph is not ℓ-connected for any $\ell \geq 3$ by the following observation. Let x_d be some element representing the default $d = \frac{\alpha:\beta}{\gamma}$. Then there are elements $x_\alpha, x_\beta, x_\gamma$ to represent the respective parts of d which are all connected to x_d. If now x_α, x_β and x_γ are removed from the graph, then there is no other element to which x_d is connected yielding a contradiction to the connectivity.

Turning to (2.) observe that (1.) prohibits using ℓ-cliques or ℓ-connectivity for any $\ell \geq 3$ to measure the treewidth of $\mathscr{A}_{(W,D)_k}$. Now define a default theory $(\emptyset, D)$ complying with Theorem 6, where $D := \left\{ d_{ij} = \frac{x_i : y_j}{\bot} \;\middle|\; 1 \leq i \leq j \leq n \right\}$, and x_i, y_j are variables for $1 \leq i \leq j \leq n$. Consisting only of this kind of default rules implies that the structure forms a pseudo-clique. □

An analogous result holds for autoepistemic logic.

Corollary 2 *Let B be a finite set of Boolean functions. There exists a family of autoepistemic B-formulas Σ_k and all $\varphi \in \Sigma_k$ are disjunctions of propositions or L-prefixed propositions such that there exists an MSO formula θ fulfilling the property $\Sigma_k \in \mathrm{EXP}(B)$ iff $\mathscr{A}_{\Sigma_k} \models \theta$ and the treewidth of $\mathscr{A}_{\Sigma_k}$ is not constant.*

Proof Define Σ_k as $\Sigma_k := \{x_i \vee x_j \mid 1 \leq i \leq j \leq k\}$. Then the structure $\mathscr{A}_{\Sigma_k}$ consist of cliques of size k, in fact. □

Corollary 3 *Let B be a finite set of Boolean functions such that $\wedge, \vee \in [B]$. Let $\mathscr{G}_1$ be the set of sets Γ of formulas in monotone 2-CNF and let $\mathscr{G}_2$ be the set of sets Γ of formulas in DNF. There exists a family of sets of B-formulas $(F, G)_k$ with $F \in \mathscr{G}_1, G \in \mathscr{G}_2$ such that there exists an MSO formula θ fulfilling the property $(F, G)_k \in \mathrm{IMP}(B)$ iff $\mathscr{A}_{(F,G)_k} \models \theta$ and the treewidth of $\mathscr{A}_{(F,G)_k}$ is not constant.*

7 Circumscription

We will follow the notion of McCarthy [27] and work with propositional Circumscription as defined by Lifschitz [24]. In the following the variables of a propositional formula φ are partitioned in three disjoined subsets (P, Q, Z). Those partitioned subsets are defined as follows:

- P is the set of variables to minimize,
- Q is the set of variables with a fixed value, in order to let the minimal models be comparable, and
- Z is the set of variables allowed to vary.

In the case of $Q = Z = \emptyset$ we work with basic propositional Circumscription. Otherwise we will talk about general propositional Circumscription. If Z contains all variables of the instance then we have usual propositional inference in the sense of $\models$. The guideline in propositional Circumscription is the compliance of the minimal models to obtain as few exceptions as possible. Informally, a minimal model is a set $M_{\min}$ which contains variables that are necessarily assigned to true. The central idea is that the variables which *can* be falsified *must* be assigned to false. The assignments of propositional formulas are partially ordered according to the coordinate wise partial order $\leq$ on Boolean vectors which obeys the ordering $0 \leq 1$ on $\{0, 1\}$.

With respect to a given partition of the variables we define the preorder $\leq_{(P,Q,Z)}$ on assignments as follows.

Definition 2 ($\leq_{(P,Q,Z)}$ *Preorder*) Let σ and σ' be two assignments of a given formula φ, then $\sigma \leq_{(P,Q,Z)} \sigma'$ if σ and σ' assign the same value to the variables in Q and for every $p \in P$, $\sigma(p) \leq \sigma'(p)$ (if there exists a variable $p \in P$ such that $\sigma(p) \neq \sigma'(p)$, we write $\sigma <_{(P,Q,Z)} \sigma'$).

Let us emphasize again that an assignment is not necessarily a model wherefore we denote assignments usually with greek letters θ, σ and models with the latin letter M. When talking about models in Circumscription it will be significant whether the model is minimal or not. The following definition captures this fact.

Definition 3 (*Minimal models*) Let φ be some formula, (P, Q, Z) be a partition of its variables, and M be a model of φ, i.e., $M \models \varphi$. We say M is a *minimal model of* φ if there exists no other model M' of φ such that $M' \subset M$ holds (and simultaneously $M' \models \varphi$). Further we say M is a (P, Q, Z)*-minimal model of* φ if there exists no other model M' of φ such that $M' <_{(P,Q,Z)} M$ (and also $M' \models \varphi$ holds).

Now we are able to define the inference relation from the circumscriptive perspective. Observe that it is possible to set Z as the set of all variables only, in order to mimic usual propositional inference.

Definition 4 (*Circumscriptive inference*) Let φ, ψ be two formulas and (P, Q, Z) be a partition of the variables of φ and ψ. An assignment σ is a *(circumscriptive) model* of φ, in symbols $\sigma \models^{\mathrm{circ}}_{(P,Q,Z)} \varphi$, if σ is a (P, Q, Z)-minimal model of φ. A formula ψ can be *circumscriptively inferred* from φ, in symbols $\varphi \models^{\mathrm{circ}}_{(P,Q,Z)} \psi$, if ψ holds in all (P, Q, Z)-minimal models of φ. Whenever we talk about basic Circumscription, i.e., $Q = Z = \emptyset$ then we just write $\models^{\mathrm{circ}}$.

Analogously we will define models of sets of formulas Γ as usual by using the conjunction over its elements, i.e., a model of $\bigwedge_{\varphi\in\Gamma} \varphi$ is a model of Γ. The *circumscriptive inference problem* (or *Circumsciption*), $\textsc{CircInf}(B)_{(P,Q,Z)}$, is the problem, given a B-formula φ, a set $\Gamma \subseteq \mathscr{L}(B)$, and a partition (P, Q, Z) of the variables, to decide if $\Gamma \models^{\mathrm{circ}}_{(P,Q,Z)} \varphi$.

Whenever we simply write $\textsc{CircInf}(B)$ we talk about *basic Circumscription* where the partition is not relevant.

Now let us consider an example which visualizes the difference between basic and general Circumscription, and also compares them to usual propositional inference. Here we show how the minimal models of a given formula can be different with respect to the partition (P, Q, Z).

Example 3 Given is the set $\Gamma = \{\psi_1, \psi_2, \psi_3\}$, consisting of the three formulas

$$\psi_1 = x_1 \vee x_3, \quad \psi_2 = x_1 \rightarrow x_2, \text{ and } \quad \psi_3 = \neg(x_3 \rightarrow x_1) \oplus (x_2 \wedge x_3),$$

the formula $\varphi = (x_1 \wedge x_2) \oplus x_3$, and the partition is $P = \{x_1\}$, $Q = \{x_2, x_3\}$, $Z = \emptyset$.

The assignments are shown in Table 2. The only two satisfying assignments (marked in light gray) of Γ are $M_1 = \{x_3\}$ and $M_2 = \{x_1, x_2, x_3\}$. However only $M_1 \models \varphi$ holds and $M_2 \not\models \varphi$. Wherefore for usual propositional inference we can say that $\Gamma \not\models \varphi$. Nevertheless for basic Circumscription we get $\Gamma \models^{\mathrm{circ}} \varphi$ as $M_1 \subset M_2$ and therefore M_2 is not minimal. Hence we consider M_1 only.

Table 2 Assignment table for Example 3

x_1	x_2	x_3	ψ_1	ψ_2	ψ_3	φ
0	0	0	0	1	0	0
0	0	1	1	1	1	1
0	1	0	0	1	0	0
0	1	1	1	1	0	1
1	0	0	1	0	0	0
1	0	1	1	0	0	1
1	1	0	1	1	0	1
1	1	1	1	1	1	0

Yet, for general Circumscription together with a given partition we have indeed two minimal models, namely both M_1 is (P, Q, Z)-minimal and M_2 is (P, Q, Z)-minimal as well because we are only allowed to remove x_1 from the current model, whereas x_2 and x_3 are fixed (as they are in Q) and not allowed to change. Hence $\Gamma \not\models^{\text{circ}}_{(P,Q,Z)} \varphi$ because $M_2 \models^{\text{circ}}_{(P,Q,Z)} \Gamma$ and $M_2 \not\models \varphi$.

Intuitively speaking by defining the partition it is possible to make (previously) non-minimal assignments minimal, as happened for M_2 in the example.

Thomas completely classified the computational complexity with respect to every set of Boolean functions [36]. There he established five different cases ranging from Π_2^p-completeness down to containment in AC^0. Hence from a computational complexity perspective the problem admits a wide spectrum of complexity degrees.

At first we will start to work with the basic Circumscription inference problem CIRCINF, i.e., having $Q = Z = \emptyset$ for the partition. Afterwards we turn towards the general case $\text{CIRCINF}(B)_{(P,Q,Z)}$ and show how to extend our result to the unrestricted inference problem allowing arbitrary partitions. This allows us to stick to the most introductory vocabulary τ_B.

7.1 Basic propositional circumscription

Instances of the basic circumscriptive inference problem contain a set Γ of formulas, as well as a formula φ. Hence we need a slight extension of the vocabulary to be able to distinguish between φ- and Γ-representatives. We define $\tau_{B,\Gamma,\varphi} := \tau_B \cup \{repr^1, repr^1_\Gamma, repr^1_\varphi\}$, where

- $repr(x)$ holds iff x represents a formula from $\Gamma \cup \{\varphi\}$,
- $repr_\Gamma(x)$ holds iff x represents a formula from Γ, and
- $repr_\varphi(x)$ holds iff x represents the formula φ.

The universe of the associated $\tau_{B,\Gamma,\varphi}$-structure $\mathscr{A}_{\Gamma,\varphi}$ consists of the formulas as well as subformulas of $\Gamma \cup \{\varphi\}$.

Lemma 5 *Let B be set of Boolean functions. Then there exists an MSO-formula θ_{circ} over $\tau_{B,\Gamma,\varphi}$ such that for any set of formulas $\Gamma \subseteq \mathscr{L}(B)$ over connectives in B the following holds $\Gamma \models^{\text{circ}} \varphi$ iff $\mathscr{A}_{\Gamma,\varphi} \models \theta_{circ}$.*

Proof Informally speaking the formula is satisfiable iff for all elements x holds: (i) if x is not representative of a formula in $\Gamma \cup \{\varphi\}$ then it is a part of a connective, (ii) if

x is not a variable, then it is either representing a constant or a connective for which the successors are well-defined.

On our way to express circumscriptive inference we need two formulas stating the satisfiability of formulas in Γ, resp., the satisfiability of φ:

$$\theta_{sat,\Gamma}(M) := \theta_{struc} \wedge \Big(\theta_{assign}(M) \wedge \forall x\big(repr_{\Gamma}(x) \rightarrow M(x)\big)\Big),$$
$$\theta_{sat,\varphi}(M) := \theta_{struc} \wedge \Big(\theta_{assign}(M) \wedge \forall x\big(repr_{\varphi}(x) \rightarrow M(x)\big)\Big).$$

Now we are able to describe minimality of a given model M of Γ by stating that there is no smaller model $M' \subset M$ which still satisfies Γ.

$$\theta_{min}(M) := \neg\exists M'\Big(\forall y(M'(y) \rightarrow M(y)) \wedge \exists x(M'(x) \wedge \neg M(x)) \wedge \theta_{sat,\Gamma}(M')\Big).$$

All of this points to the fact that $\Gamma \models^{\mathrm{circ}} \varphi$ iff $\mathscr{A}_{\Gamma,\varphi} \models \theta_{circ}$, where

$$\theta_{circ} := \forall M\Big(\big(\theta_{sat,\Gamma}(M) \wedge \theta_{min}(M)\big) \rightarrow \theta_{sat,\varphi}(M)\Big)$$

and satisfies the lemma. □

Applying the results from Courcelle and Elberfeld et al. will immediately lead to the following theorem which settles an FPT runtime and a XL space requirement if one parameterizes the problem with treewidth of the underlying structures.

Theorem 10 *Let B be a finite set of Boolean functions, $k \in \mathbb{N}$ be fixed, Γ be a set of B-formulas and φ be a B-formula such that $\mathscr{A}_{\Gamma,\varphi}$ has a treewidth bounded by k. Then the basic propositional circumscriptive inference problem for B-formulas is solvable in time $O\big(f(k)\cdot(|\Gamma|+|\varphi|)\big)$ and space $O(\log(f(k))\cdot\log(|\Gamma|+|\varphi|))$, where f is a recursive function.*

7.2 General propositional circumscription

In contrast to basic Circumscription additionally to the input Γ, φ we get a partition of their variables into three disjoined subsets (P, Q, Z) where

- P is the set of variables to minimize,
- Q is the set of variables with a fixed value, in order to let the minimal models be comparable, and
- Z is the set of variables allowed to vary.

Therefore we need to extend the vocabulary $\tau_{B,\Gamma,\varphi}$ with two new unary predicates: $repr_P(x)$, $repr_Q(x)$ which hold iff an element x represents a variable from P, or Q respectively. We will denote this extension with $\tau_{B,circ}$.

The next lemma proves the existence of an MSO-formula θ_{circ} with respect to a given $\textsc{CircInf}(B)_{(P,Q,Z)}$ instance $(\Gamma, \varphi, (P, Q, Z))$ which is satisfiable iff $\Gamma \models^{\mathrm{circ}}_{(P,Q,Z)} \varphi$. In contrast to basic propositional Circumscription we use an associated extended

$\tau_{B,circ}$-structure $\mathscr{A}_{\Gamma,\varphi,(P,Q,Z)}$ now. The universe of $\mathscr{A}_{\Gamma,\varphi,(P,Q,Z)}$ additionally consists of the predicates for P and Q.

Lemma 6 *Let B be set of Boolean functions. Then there exists an MSO-formula $\theta_{(P,Q,Z)\text{-}circ}$ over $\tau_{B,circ}$ such that for any set of formulas $\Gamma \subseteq \mathscr{L}(B)$ over connectives in B, every B-formula φ, and every partition (P, Q, Z) of variables from $\Gamma \cup \{\varphi\}$ the following holds: $\Gamma \models^{\text{circ}}_{(P,Q,Z)} \varphi$ iff $\mathscr{A}_{\Gamma,\varphi,(P,Q,Z)} \models \theta_{(P,Q,Z)\text{-}circ}$.*

Proof In the following we will build on the formulas from the proof of Lemma 5. We will start with a formula that describes the ordering on the models with respect to the partition (P, Q, Z). Let us recapitulate the notion: $M \leq_{(P,Q,Z)} M'$ holds iff for all $x \in Q$ we have $M(x) \leftrightarrow M'(x)$ and for all $x \in P$ we have $M(x) \leq M'(x)$. The following formula expresses this behavior and is valid if and only if $M \leq_{(P,Q,Z)} M'$ holds:

$$\theta_{ord}(M, M') := \forall x\Big(repr_Q(x) \to \big(M(x) \leftrightarrow M'(x)\big)\Big) \\ \wedge \forall x\Big(repr_P(x) \to \big(M(x) \to M'(x)\big)\Big).$$

The set Z needs no further consideration as the variables are allowed to vary here. Compared to basic Circumscription we need to extend the formula θ_{min} to obey the partition ordering as follows:

$$\theta'_{min}(M) := \neg\exists M'\Big(\theta_{ord}(M', M) \wedge \exists x(M'(x) \wedge \neg M(x)) \wedge \theta_{sat,\Gamma}(M')\Big).$$

Finally $\theta_{(P,Q,Z)\text{-}circ}$ is defined by transforming θ_{min} in θ_{circ} to use the previously adjusted θ'_{min}:

$$\theta_{(P,Q,Z)\text{-}circ} := \forall M\Big(\big(\theta_{sat,\Gamma}(M) \wedge \theta'_{min}(M)\big) \to \theta_{sat,\varphi}(M)\Big).$$

By construction we get that $\Gamma \models^{\text{circ}}_{(P,Q,Z)} \varphi$ if and only if $\mathscr{A}_{\Gamma,\varphi,(P,Q,Z)} \models \theta_{(P,Q,Z)\text{-circ}}$. □

Using Courcelle's theorem and the version from Elberfeld et al. we can show the following result now.

Theorem 11 *Let B be a finite set of Boolean functions, $k \in \mathbb{N}$ be fixed, Γ be a set of B-formulas, φ be a B-formula, and (P, Q, Z) be a partition of the variables in $\Gamma \cup \{\varphi\}$ such that $\mathscr{A}_{\Gamma,\varphi,(P,Q,Z)}$ has a treewidth bounded by k. Then the general propositional circumscriptive inference problem for B-formulas is solvable in time $O\big(f(k) \cdot (|\Gamma| + |\varphi|)\big)$ and space $O(\log(f(k)) \cdot \log(|\Gamma| + |\varphi|))$, where f is a recursive function.*

Note that Theorem 11 and Lemma 6 especially hold for the case where Q and Z are empty in $\textsc{CircInf}(B)$.

Now we will investigate some restricted versions (in the sense of allowed Boolean connectives) of the circumscriptive inference problem and show the validity of lower bounds under reasonable complexity assumptions for a wide range of parameterization functions.

Theorem 12 *Let B be a finite set of Boolean functions such that either*

1. $\wedge, \vee \in [B]$ *and κ is a parameterization function for which there exists a $c \in \mathbb{N}$ such that $\kappa\big((\Gamma, \varphi, (P, \emptyset, \emptyset))\big) < c$, or*
2. $\vee \in [B]$, *κ is a parameterization function for which there exists a $c \in \mathbb{N}$ such that $\kappa\big((\Gamma, z, (P, Q, \{z\}))\big) < c$, whenever Γ consists only of formulas which are disjunctions of two variables and z is a single variable.*

If NP $\neq$ P *then the circumscriptive inference problem for B-formulas parameterized by κ is not in* $\mathrm{XP_{nu}}$.

Proof 1. Assume that NP $\neq$ P holds and denote with CIRCINF(B) the circumscriptive inference problem for B-formulas. In [36, Lemma 5.3.5] a reduction from 3TAUT to CIRCINF(B) is shown proving its coNP-hardness. Hence, if (CIRCINF(B), κ) $\in$ $\mathrm{XP_{nu}}$ holds then (CIRCINF, $\kappa)_\ell \in$ P for every $\ell \in \mathbb{N}$ by definition of $\mathrm{XP_{nu}}$. Through $\ell < c$ we get CIRCINF $\in$ P contradicting the assumption.
2. In [36, Lemma 5.3.6] a similar reduction was shown using only $\{\vee\}$-formulas for instances of CIRCINF(B) and a form as required in the theorem. We proceed analogously as in (1.). □

Observe that a possible non-trivial parameterization function as in the sense of Theorem 12(*2.*) is the length of formulas in Γ (which is fixed as only disjunctions of two variables are allowed). Also note that (*1.*) holds for basic Circumscription as well whereas (*2.*) does not.

Theorem 13 *Let B be a finite set of Boolean functions such that $\leftrightarrow \in [B]$. Further let κ be a parameterization function such that $\kappa\big((\Gamma, \varphi, (X, \emptyset, \emptyset))\big) < c$. If* $\oplus$L $\neq$ L *then the circumscriptive inference problem for B-formulas parameterized by κ is not in* $\mathrm{XL_{nu}}$.

Proof Let B a set of Boolean functions such that $\leftrightarrow \in [B]$. A reduction from the $\oplus$L-complete propositional inference problem of 1-reproducing affine formulas to circumscriptive inference of B-formulas has been stated in [36, Lemma 5.3.8]. Again, the existence of an algorithm settling circumscriptive inference of B-formulas in $\mathrm{XL_{nu}}$ would imply solving propositional inference problem of 1-reproducing affine formulas in logarithmic space. This contradicts the assumption $\oplus$L $\neq$ L. □

This theorem also holds for basic Circumscription as we have $Q = Z = \emptyset$ in the condition of the result.

8 Logic-based abduction

Generally the problem has several similarities with Circumscription but also some differences. At first the problem is defined over usual implication and does not consider an ordering or even partition of the variables. One works with a knowledge base Γ, a set of variables A and a formula φ which contains only variables from $\Gamma \backslash A$. Now the question is whether there exists a set of literals E over variables from A such that E agrees with Γ, i.e., is satisfiable simultaneously with Γ, and the conjunction of both implies φ. Hence one goes from an observation Γ to a hypothesis A which needs an explanation E. The problem of interest is the *abduction problem* ABD(B), given a set $\Gamma \subseteq \mathscr{L}(B)$, a set $A \subseteq \text{Vars}(\Gamma)$ of variables, and a B-formula φ with $\text{Vars}(\varphi) \subseteq \text{Vars}(\Gamma) \setminus A$, to decide if there exist a set $E \subseteq \text{Lits}(A)$ such that $\Gamma \wedge E$ is satisfiable and $\Gamma \wedge E \models \varphi$ holds.

One also says that E is an *explanation* or *solution* of the Abduction problem, where A is the set of *hypotheses* and φ is called the *manifestation* or *query*. Every explanation can be extended to the full one, i.e., $\text{Vars}(E) = A$. Let us start with a motivating example of the problem.

Example 4 This time let Γ be defined as follows $\Gamma = \{\gamma_1, \gamma_2\}$, where

$$\gamma_1 = (x_1 \wedge x_2 \wedge x_4) \oplus \neg(x_3 \rightarrow x_1), \text{ and } \quad \gamma_2 = \neg\big((x_1 \vee x_3) \rightarrow (x_1 \oplus x_2)\big).$$

Table 3 shows an overview of the assignments. So we get four satisfiable assignments for Γ, $M_1 = \{x_3\}$, $M_2 = \{x_3, x_4\}$, $M_3 = \{x_1, x_2, x_4\}$ and $M_4 = \{x_1, x_2, x_3, x_4\}$. The set of hypotheses A is given as $A = \{x_3, x_4\}$. The input formula φ with $\text{Vars}(\varphi) \subseteq \text{Vars}(\Gamma) \setminus A$ is $\varphi = x_1 \vee x_2$. So the valid sets $E \subseteq \text{Lits}(A)$ are $E_1 = \{\neg x_3\}$ and $E_2 = \{\neg x_3, x_4\}$, which can be seen in Table 3(b). Observe that although $\Gamma \wedge (\neg x_3 \wedge \neg x_4) \models \varphi$ is true, E is not a valid solution as $\Gamma \wedge (\neg x_3 \wedge \neg x_4)$ is unsatisfiable.

Further Creignou et al. [10] consider restrictions on the manifestations of this problem. They write

- ABD(B, Q) if φ is a single literal,
- ABD(B, C) if φ is a clause (disjunction of literals),
- ABD(B, T) if φ is a term (conjunction of literals), and

Table 3 (a) Assignment table for Example 4, and (b) Overview of all relevant sets E

Model	x_1	x_2	x_3	x_4	γ_1	γ_2	φ
	0	0	0	0	0	0	0
	0	0	0	1	0	0	0
M_1	0	0	1	0	1	1	0
M_2	0	0	1	1	1	1	0
	0	1	0	0	0	0	1
	0	1	0	1	0	0	1
	0	1	1	0	1	0	1
	0	1	1	1	1	0	1
	1	0	0	0	0	0	1
	1	0	0	1	0	0	1
	1	0	1	0	0	0	1
	1	0	1	1	0	0	1
	1	1	0	0	0	1	1
M_3	1	1	0	1	1	1	1
	1	1	1	0	0	1	1
M_4	1	1	1	1	1	1	1

(a)

E	$\Gamma \wedge E \models \varphi$?
$\{x_3\}$	M_1 ↯
$\{\overline{x_3}\}$	✓
$\{x_4\}$	M_2 ↯
$\{\overline{x_4}\}$	M_1 ↯
$\{x_3, x_4\}$	M_2 ↯
$\{x_3, \overline{x_4}\}$	M_1 ↯
$\{\overline{x_3}, x_4\}$	✓
$\{\overline{x_3}, \overline{x_4}\}$	$\Gamma \wedge E$ unsatisfiable

(b)

– $\textsc{Abd}(B, \mathscr{L}(B))$ if φ is a B-formula.

From a computational complexity point of view the unrestricted version has been classified by Eiter and Gottlob [12] as Σ_2^{P}-complete whereas a complete classification of its fragments by the aforementioned restrictions on the manifestation as well as the allowed Boolean connectives has been achieved by Creignou et al. [10]. They obtained a rather fine grained tractability frontier ranging through the classes L, $\oplus$L, P, NP, and Σ_2^{P}.

For our approach the different manifestation versions are only relevant at obtaining lower bounds. The MSO-formulas expressing the problem do not differ between, e.g., a single literal as manifestation or a term.

Again, we will consider an extension of the vocabulary τ_B which contains the necessary predicates to talk about Abduction: $\tau_{B,abd} := \tau_{B,\Gamma,\varphi} \cup \{\mathrm{lit}_A^1, \text{pos-lit}_A^1\}$, where $\mathrm{lit}_A(x)$ holds iff x represents a literal from $\mathrm{Lits}(A)$, and $\text{pos-lit}_A(x)$ holds iff X represents a positive literal from $\mathrm{Lits}(A)$. The structure $\mathscr{A}_{\Gamma,A,\varphi}$ of the vocabulary $\tau_{B,abd}$ has in its universe formulas and subformulas of $\Gamma \cup \{\varphi\}$ and a set A of variables.

Lemma 7 *Let B be set of Boolean functions. Then there exists an MSO-formula θ_{abd} over $\tau_{B,abd}$ such that for any set of formulas $\Gamma \subseteq \mathscr{L}(B)$ over connectives in B, any set $A \subseteq \mathrm{Vars}(\Gamma)$ of variables, and every B-formula φ with $\mathrm{Vars}(\varphi) \subseteq \mathrm{Vars}(\Gamma) \setminus A$ the following holds: there exists $E \subseteq \mathrm{Lits}(A)$ s.t. $\Gamma \wedge E$ is satisfiable and $\Gamma \wedge E \models \varphi$ iff $\mathscr{A}_{\Gamma,A,\varphi} \models \theta_{abd}$.*

Proof In the following we will use formulas from the proof of Lemma 5. In particular, the formulas $\theta_{sat,\Gamma}(M)$ and $\theta_{sat,\varphi}(M)$ will be used to construct the formula θ_{abd}.

At first we define a formula $\theta_{sat,\Gamma\wedge E}(E, M)$ which holds iff $M \models \Gamma \wedge E$:

$$\theta_{sat,\Gamma\wedge E}(E, M) := \theta_{sat,\Gamma}(M) \wedge \forall y\big(E(y) \rightarrow (\text{pos-lit}_A(y) \leftrightarrow M(y))\big).$$

Observe that solely the formula does not ensure that E is a set of representatives of literals from $\mathrm{Lits}(A)$. However in the following formula we ensure that only sets E with this property are used.

Finally we will utilize this formula to express Abduction as follows

$$\theta_{abd} := \exists E \forall x \Big(E(x) \rightarrow \mathrm{lit}_A(x) \wedge \big(\exists M\ \theta_{sat,\Gamma\wedge E}(E, M)\big) \\ \wedge \forall M (\theta_{sat,\Gamma\wedge E}(E, M)) \rightarrow \theta_{sat,\varphi}\Big).$$

The formula existentially quantifies the explanation E and forces valid ones (which are only over literals from $\mathrm{Lits}(A)$) to require a satisfying model for $E \wedge \Gamma$ as well as that $E \wedge \Gamma \rightarrow \varphi$ holds for all assignments. □

Again the application of Courcelle's theorem allows us to conclude with the following result.

Theorem 14 *Let B be a finite set of Boolean functions, $k \in \mathbb{N}$ be fixed, Γ be a set of B-formulas, $A \subseteq \mathrm{Vars}(\Gamma)$ be a set of Variables, and φ be a B-formula with*

$\mathrm{Vars}(\varphi) \subseteq \mathrm{Vars}(\Gamma)\backslash A$ *such that* $\mathscr{A}_{\Gamma,A,\varphi}$ *has a treewidth bounded by k. Then the abduction problem for B-formulas is solvable in time* $O\big(f(k)\cdot\log(|\Gamma|+|\varphi|)\big)$ *and space* $O(\log(f(k))\cdot\log(|\Gamma|+|\varphi|))$*, where f is a recursive function.*

Again we take the focus on lower bounds now. Therefore we will consider fragments of the four mentioned versions of the Abduction problem and show how some of them deny fixed parameter tractability under reasonable complexity assumptions.

Theorem 15 *Let B be a finite set of Boolean functions such that* $\oplus(x, y, z) \in [B]$. *Further let* κ *be a parameterization function such that* $\kappa\big((\Gamma, \emptyset, q)\big) < c$ *whenever q is single a variable. If* $\oplus\mathrm{L} \neq \mathrm{L}$ *then the abduction problem for B-formulas parameterized by* κ *is not in* $\mathrm{XL_{nu}}$.

Proof Creingou et al. [10] show a reduction from the problem to determine whether a system of linear equations over GF(2) has a solution (which has been proven to be $\oplus$L-complete [5]) to ABD(B, Q) where $[B] = \{\oplus(x, y, z)\}$. Hence a membership of the parameterized version of this problem in the sense of the theorem would imply $\mathrm{L} = \oplus\mathrm{L}$ if the problem is situated in $\mathrm{XL_{nu}}$. □

Here any parameterization function which is defined in some respect to the set of hypotheses obeys the claim of the theorem. Further parameterizations which interfere with the manifestation are also possible candidates for the required functions.

Theorem 16 *Let B be a finite set of Boolean functions such that either*

1. $\wedge, \vee \in [B]$ *and* κ *is a parameterization function for which there exists a* $c \in \mathbb{N}$ *such that* $\kappa\big((\Gamma, A, q)\big) < c$ *whenever q is a single variable, or*
2. $\vee \in [B]$ *and* κ *is a parameterization function for which there exists a* $c \in \mathbb{N}$ *such that* $\kappa\big((\Gamma, A, t)\big) < c$ *whenever t is a term.*

If $\mathrm{NP} \neq \mathrm{P}$ *then the abductive inference problem for B-formulas parameterized by* κ *is not in* $\mathrm{XP_{nu}}$.

Proof 1. From [10, Prop. 7] we know that ABD(B, Q) is NP-complete for clones $[B] \ni \wedge, \vee$.
2. In [10, Prop. 10] NP-completeness of ABD(B, T) is shown for clones $[B] \ni \vee$.

Whence in both cases a membership in $\mathrm{XP_{nu}}$ for such parameterized problems would imply P = NP. □

Unfortunately, for the previous theorem the constructed formulas in the investigated proofs will not immediately yield to parameterization functions as described after Theorem 15 or Theorem 12 as the inputs are not restricted in any way.

9 Conclusion

In this paper we applied Courcelle's Theorem [8] and the logspace version of Elberfeld et al. [13] to the most prominent decision problems in the non-montonic abduction, autoepistemic logic, circumscription, and default logic. Thereby we showed that the

Table 4 Overview of established lower bounds

Fragment	Instance $x^{\dagger}$	Not in*	References
$\textsc{Ext}([\{\neg, \top\}])$	$(\emptyset, D)$	$\mathrm{XP}_{\mathrm{nu}}$	Theorem 3
$\textsc{Ext}([\{\bot\}])$	(W, D)	$\mathrm{XL}_{\mathrm{nu}}$	Theorem 4
$\textsc{Exp}([\{\vee, \top, \bot\}])$	(Σ)	$\mathrm{XP}_{\mathrm{nu}}$	Theorem 6
$\textsc{Exp}([\{\oplus, \top\}])$	(Σ)	$\mathrm{XL}_{\mathrm{nu}}$	Theorem 7
$\textsc{Imp}([\{x \oplus y \oplus z\}])$	(F, G)	$\mathrm{XL}_{\mathrm{nu}}$	Theorem 8
$\textsc{Imp}([\{\vee, \wedge\}])$	(F, G)	$\mathrm{XP}_{\mathrm{nu}}$	Theorem 9
$\textsc{CircInf}([\{\vee, \wedge\}])$	$(\Gamma, \varphi, (P, \emptyset, \emptyset))$	$\mathrm{XP}_{\mathrm{nu}}$	Theorem 12(1)
$\textsc{CircInf}([\{\vee\}])$	$(\Gamma, z, (P, Q, \{z\}))$	$\mathrm{XP}_{\mathrm{nu}}$	Theorem 12(2)
$\textsc{CircInf}([\{\leftrightarrow\}])$	$(\Gamma, \varphi, (X, \emptyset, \emptyset))$	$\mathrm{XL}_{\mathrm{nu}}$	Theorem 13
$\textsc{Abd}([\{\oplus(x, y, z)\}], \mathrm{Q})$	$(\Gamma, \emptyset, q)$	$\mathrm{XL}_{\mathrm{nu}}$	Theorem 15
$\textsc{Abd}([\{\vee, \wedge\}], \mathrm{Q})$	(Γ, A, q)	$\mathrm{XP}_{\mathrm{nu}}$	Theorem 16(1)
$\textsc{Abd}([\{\vee\}], \mathrm{T})$	(Γ, A, t)	$\mathrm{XP}_{\mathrm{nu}}$	Theorem 16(2)

W is a set of formulas, D is a set of default rules, Σ is a set of autoepistemic formulas, F is a set of monotone 2-CNF formulas, G a set of DNF formulas, z, q are single variables, and t is a term
† parameterization function $\kappa(x) < c$ for some $c \in \mathbb{N}$,
* unless $\mathrm{P} = \mathrm{NP}$ (case not in $\mathrm{XP}_{\mathrm{nu}}$), resp., unless $\oplus\mathrm{L} = \mathrm{L}$ (case not in $\mathrm{XL}_{\mathrm{nu}}$).

extension existence problem for a given default theory (W, D) is solvable in time $O(f(k) \cdot |(W, D)|)$ and space $O(f(k) \cdot \log |(W, D)|)$, i.e., FPT time and XL space, if the treewidth of the corresponding MSO structure is bounded by k; similarly for the expansion existence problem for a set of autoepistemic formulas, and as well for the implication problem for sets of formulas F, G. Analogue results have been obtained for reasoning in circumscription and abduction.

Further we mention that one can achieve similar results for the credulous (resp. brave) and skeptical (resp. cautious) reasoning problems of the non-montonic logics from above by slight extensions of the constructed MSO-formulas. We want to mention that obvious modifications of our formulas lead to general results for answer set programming as well.

Furthermore we consider with *pseudo-cliques* a weaker notion of cliques in the sense of edge contractions and topological minors.

If we investigate default theories (W, D) which contain an empty knowledge base W and only defaults which are composed of propositions or the constant $\bot$ only, then for constant parameterizations we show collapses of P and NP (resp. L and NL) if the corresponding parameterized problem is in $\mathrm{XP}_{\mathrm{nu}}$ (resp. $\mathrm{XL}_{\mathrm{nu}}$). Thus through the concept of pseudo-cliques we construct a family of default theories whose treewidth of its MSO-structures is unbounded. Therefore this parameter cannot be used to prove such complexity class collapses. Analogue claims can be made for the expansion existence problem in autoepistemic logic and the implication problem for sets of formulas. Next we have seen that for a wide range of parameterization functions whose value is bounded by some constant $c \in \mathbb{N}$ there are quite restrictive fragments (e.g., for basic Circumscription it suffices that $\wedge$ and $\vee$ are present) which are not contained in the large parameterized class $\mathrm{XP}_{\mathrm{nu}}$ unless $\mathrm{coNP} = \mathrm{NP}$ holds. Also we proved that if

$\oplus$L $\neq$ L holds basic Circumscription for propositional formulas allowing the existence of the biimplication $\leftrightarrow$ prohibits XL_{nu} algorithms for similar parameterizations as before.

Most interestingly for the Abduction fragment around *ternary exclusive-or* $\oplus(x, y, z)$ with any parameterization function defined around the hypotheses prohibits the existence of an XL_{nu} algorithm unless L = $\oplus$L.

For subsequent research it would be very interesting to find a parameterization that is non-trivial in the sense of Theorem 3 but uses many different values. For Theorem 12(2) we only observed a rather restrictive function asking for the size of formulas in Γ, having in mind that only disjunctions of variables are allowed in the investigated fragment. Also insights on new types of parameterizations, in particular in the context of the new space parameterized complexity classes, would be very engaging.

Acknowledgments For helpful hints and discussions we are grateful to Nadia Creignou (Marseille) and Thomas Schneider (Bremen). We also thank the anonymous referees for helpful comments.

References

1. Beyersdorff, O., Meier, A., Thomas, M., Vollmer, H.: The complexity of propositional implication. Inf. Process. Lett. **109**(18), 1071–1077 (2009). doi:10.1016/j.ipl.2009.06.015
2. Beyersdorff, O., Meier, A., Thomas, M., Vollmer, H.: The complexity of reasoning for fragments of default logic. J. Logic Comput. **22**(3), 587–604 (2012). doi:10.1093/logcom/exq061
3. Bodlaender, H.: A partial k-arboretum of graphs with bounded treewidth. Theor. Comput. Sci. **209**, 1–45 (1998)
4. Brewka, G., Niemelä, I., Truszczynski, M.: Nonmonotonic Reasoning. Elsevier, Amsterdam (2008)
5. Buntrock, G., Damm, C., Hertrampf, U., Meinel, C.: Structure and importance of logspace modclasses. Math. Syst. Theory **25**, 223–237 (1992)
6. Cadoli, M., Lenzerini, M.: The complexity of closed world reasoning and circumscription. In: Proceedings of 8th National Conference on Artificial Intelligence, pp. 550–555. AAAI Press (1990)
7. Chen, Y., Flum, J., Grohe, M.: Bounded nondeterminism and alternation in parameterized complexity theory. In: Proceedings of the 18th IEEE Annual Conference on Computational Complexity (CCC 2003), pp. 13–29 (2003). doi:10.1109/CCC.2003.1214407
8. Courcelle, B., Engelfriet, J.: Graph Structure and Monadic Second-Order Logic, A Language Theoretic Approach. Cambridge University Press, Cambridge (2012)
9. Creignou, N., Meier, A., Thomas, M., Vollmer, H.: The complexity of reasoning for fragments of autoepistemic logic. ACM Trans. Comput. Logic **13**(2) (2010)
10. Creignou, N., Schmidt, J., Thomas, M.: Complexity of propositional abduction for restricted sets of boolean functions. In: Proceedings of 12th International Conference on the Principles of Knowledge Representation and Reasoning. AAAI Press (2010)
11. Eiter, T., Gottlob, G.: Propositional circumscription and extended closed-world reasoning are Π_2^P-complete. Theor. Comput. Sci. **114**(2), 231–245 (1993)
12. Eiter, T., Gottlob, G.: The complexity of logic-based abduction. JACM **42**(1), 3–42 (1995)
13. Elberfeld, M., Jakoby, A., Tantau, T.: Logspace versions of the theorems of Bodlaender and Courcelle. In: Proceedings of 51th Annual IEEE Symposium on Foundations of Computer Science. IEEE Computer Society (2010)
14. Elberfeld, M., Stockhusen, C., Tantau, T.: On the space complexity of parameterized problems. In: Parameterized and Exact Computation, pp. 206–217. Springer, Berlin (2012)
15. Fellows, M. R., Pfandler, A., Rosamond, F. A., Rümmele, S.: The parameterized complexity of abduction. In: Proceedings of 26th AAAI Conference on Artificial Intelligence (2012)
16. Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer, Berlin (2006)

17. Gelfond, M., Przymusinska, H., Przymusinski, T.C.: On the relationship between circumscription and negation as failure. Artif. Intell. **38**(1), 75–94 (1989)
18. Gottlob, G.: Complexity results for nonmonotonic logics. J. Logic Comput. **2**(3), 397–425 (1992)
19. Gottlob, G., Pichler, R., Wei, F.: Bounded treewidth as a key to tractability of knowledge representation and reasoning. Artif. Intell. **174**(1), 105–132 (2010)
20. Gottlob, G., Scarcello, F., Sideri, M.: Fixed-parameter complexity in ai and nonmonotonic reasoning. Artif. Intell. **138**(1), 55–86 (2002)
21. Gottlob, G., Szeider, S.: Fixed-parameter algorithms for artificial intelligence, constraint satisfaction and database problems. Comput. J. **51**(3), 303–325 (2008)
22. Jakl, M., Pichler, R., Woltran, S.: Answer-Set Programming with Bounded Treewidth. In: Proceedings of 21st IJCAI, pp. 816–822 (2009)
23. Kirousis, L.M., Kolaitis, P.G.: The complexity of minimal satisfiability problems. Inf. Comput. **187**(1), 20–39 (2003)
24. Lifschitz, V.: Computing circumscription. In: Proceedings of 9th International Joint Conference on Artificial Intelligence, pp. 121–127. Morgan Kaufman (1985)
25. Marek, V., Truszczynski, M.: Nonmonotonic Logic: Context-Dependent Reasoning. Springer, Berlin (1993)
26. Marek, V., Truszczynski, M.: Stable models and an alternative logic programming paradigm. In: The Logic Programming Paradigm—A 25-Year Perspective, pp. 375–398 (1999)
27. McCarthy, J.: Circumscription—a form of non-monotonic reasoning. Artif. Intell. **13**(1–2), 27–39 (1980)
28. Moore, R.C.: Semantical considerations on modal logic. Artif. Intell. **25**, 75–94 (1985)
29. Niemelä, I.: Towards automatic autoepistemic reasoning. In: Proceedings of 2nd European Workshop on Logics in Artificial Intelligence, LNCS, vol. 478, pp. 428–443. Springer, Berlin (1990)
30. Niemelä, I.: Logic programming with stable model semantics as a constraint programming paradigm. Ann. Math. Artif. Intell. **25**(3–4), 241–273 (1999)
31. Nordh, G.: A trichotomy in the complexity of propositional circumscription. In: Proceedings of 11th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning, pp. 257–269. Springer, Berlin (2004)
32. Peirce, C.S.: Philosophical Writings of Peirce. Courier Dover Publications, New York (1955)
33. Peirce, C.S., Hartshorne, C., Weiss, P.: The Collected Papers of Charles Sanders Peirce. Cambridge Press, Cambridge (1932)
34. Post, E.: The two-valued iterative systems of mathematical logic. Ann. Math. Stud. **5**, 1–122 (1941)
35. Reiter, R.: A logic for default reasoning. Artif. Intell. **13**, 81–132 (1980)
36. Thomas, M.: On the complexity of fragments of nonmonotonic logics. Ph.D. thesis, Leibniz Universität Hannover (2010)
37. Thomas, M., Vollmer, H.: Complexity of non-monotonic logic. Bull. EATCS **102**, 53–82 (2010)
38. Zaho, X., Ding, D.: Fixed-parameter tractability of disjunction-free default reasoning. JCST **18**(1), 118–124 (2003)