ECAI 2012
Luc De Raedt et al. (Eds.)

doi:10.3233/978-1-61499-098-7-492

Fixed-Parameter Algorithms for Closed World Reasoning[1]

Martin Lackner and **Andreas Pfandler**[2]

Abstract. Closed world reasoning and circumscription are essential tasks in AI. However, their high computational complexity is a serious obstacle for their practical application. In this work we employ the framework of parameterized complexity theory in order to search for fixed-parameter algorithms. We consider eleven parameters describing different characteristics of the input. For several combinations of these parameters we are able to design efficient fixed-parameter tractable algorithms. All our algorithms have a runtime only single-exponential in the parameters and linear in the input size. Furthermore, by providing parameterized hardness results we show that we have actually found all tractable fragments involving these eleven parameters. We hereby offer a complete picture of the parameterized complexity of brave closed world reasoning and circumscription.

1 Introduction

Closed world reasoning [21] is a central technique used in AI and database theory. The core of this formalism is the closed world assumption. The closed world assumption asserts that any information not stored in a knowledge base is assumed to be false. In this paper a knowledge base is represented as a propositional formula φ. We consider the question of whether an information π, also represented as a propositional formula, is entailed by φ under the closed world assumption. A common way to formalize the closed world assumption for propositional formulas is with the help of (subset) minimal models: The formula φ entails π if π is true in at least one minimal model of φ (brave reasoning) or π is true in all minimal models (cautious reasoning).

An especially expressive form of the closed world assumption is ECWA, the extended closed world assumption [10]. Here three types of variables of φ are distinguished: P, Q and Z variables. A model $\mathcal{M}$ is $\langle P, Q, Z\rangle$-minimal if there is no model $\mathcal{M}'$ with $\mathcal{M}' \cap P \subset \mathcal{M} \cap P$ and $\mathcal{M}' \cap Q = \mathcal{M} \cap Q$. In [10] it was shown that ECWA coincides with circumscription on propositional formulas. Circumscription [18] is a powerful reasoning tool applicable to many formalisms such as belief revision [17] and answer-set programming [6].

The computational complexity of closed world reasoning and circumscription is well explored. In [5] it was shown that deciding whether there is a minimal model containing a given variable is Σ_2^P-complete. In [1] the complexity for several restricted classes of formulas is studied. However, only few tractable cases have been discovered in these papers. Since these tractability results are only applicable to rather restricted classes of formulas, the search for efficient algorithms has to continue.

One approach how to tackle computationally hard problems is parameterized complexity theory. The idea of a parameterized complexity analysis is to identify properties of the input – so-called parameters – and to analyze the complexity of the problem from a multivariate point of view. This is in contrast to a classical complexity analysis where only the input size is considered. The main concept is *fixed-parameter tractability*. An algorithm is called *fixed-parameter tractable* (fpt) with respect to the parameters $k_1, k_2, \dots, k_l$ if it solves the given problem in time $\mathcal{O}(f(k_1, k_2, \dots, k_l) \cdot n^c)$. Here f is a computable (usually exponential) function and c a constant. Observe that the runtime of such an fpt-algorithm is only polynomially influenced by the input size. Hence such an algorithm might perform well even on large instances given that the parameter values are comparably small. The framework of parameterized complexity theory also offers tools for proving hardness results and therefore allows to rule out the possibility of fpt-algorithms in some cases (under standard complexity theoretic assumptions).

Parameterized complexity theory has received increasing interest in the field of AI as can be seen in the survey [13]. The parameterized complexity of answer-set programming – which also involves minimal models – has been extensively studied in [7, 15, 20, 22]. In [16] the parameterized complexity of computing minimal models is studied. However, the minimization is performed with respect to all variables, i.e., there are only P-variables. Tree-width based fpt-algorithms for several closed world reasoning formalisms are presented in [11]. While this work is remarkable for the great number of considered formalisms, the algorithms require the expensive computation of a tree decomposition. A first parameterized complexity analysis in the area of AI that also involves hardness results has been performed in [12]. The authors introduce the problem SMALL MODEL CIRCUMSCRIPTION (SMC) which is also the problem we will focus on in our paper. SMC asks whether – given two propositional formulas φ and π – there exists a $\langle k; P, Q, Z\rangle$-minimal model of φ in which the property π holds. This means that for finding a model of φ as well as for the minimality check only "small models" are considered, i.e., at most k variables can be set to true. SMC can therefore be considered as brave reasoning under ECWA restricted to small models. As stated in [12], this restriction makes sense if one has large theories but is mainly interested in small models (such as in abductive diagnosis). In [12] only the parameter k is studied and several hardness results are shown for this parameter. Also, an fpt-algorithm is obtained but only for a drastically restricted version of SMC. Despite the wealth of parameterized complexity results in AI, a systematic parameterized complexity analysis of circumscription has not been performed until now.

In our work we conduct an extensive parameterized complexity analysis of SMC aiming at efficient fpt-algorithms. We take eleven parameters into consideration (listed in Table 1) all of them being efficiently computable. For each of the 2^{11} combinations of param-

---

[1] Supported by the Austrian Science Fund (FWF): P20704-N18.
[2] Institute of Information Systems, Vienna University of Technology, Austria, email:{lackner, pfandler}@dbai.tuwien.ac.at

eters we provide either an fpt-algorithm or prove a hardness result. The following results have been achieved:

- We present five fpt-algorithms. They are fpt with respect to combinations of parameters – single parameters do not yield fixed-parameter tractability for SMC. These algorithms are single exponential in the parameters and linear in the input size n and can therefore be expected to perform especially well on instances with moderate parameter values.
- Two of these algorithms are obtained by making use of backdoor sets [23]. Backdoor sets are distance measures to tractable formula classes – in our case they measure the distance to Horn formulas. We apply this concept to SMC.
- Finally we show that this paper contains all fpt-algorithms possible for the considered set of parameters. This is achieved by proving parameterized hardness results for the remaining combinations of parameters. As a consequence we can conclude that our analysis is indeed *complete* for the considered set of parameters.

2 Preliminaries

Graphs and sets. An (undirected) graph is defined as a pair $G = (V, E)$ where V is the set of vertices and E consists of subsets of V of cardinality 2. For $m \in \mathbb{N}$, we use $[m]$ to denote the set $\{1, \ldots, m\}$.

Boolean logic. A *literal* is a variable (*positive literal*) or a negated variable (*negative literal*). A *clause* is a disjunction of literals. A formula is in conjunctive normal form if it is a conjunction of disjunctions of literals. The class of such formulas is denoted by CNF. It is convenient to also view a CNF formula as a set of clauses and a clause as a set of literals. A formula is *monotone* if it does not contain negations. *Horn* formulas are CNF formulas with at most one positive literal per clause.

Given some formula φ we denote by $var(\varphi)$ the set of variables occurring in φ. An *interpretation* $\mathcal{I} \subseteq var(\varphi)$ is a subset of the variables. An interpretation $\mathcal{I}$ is called a *model* (of formula φ) if φ is satisfied by setting the variables in $\mathcal{I}$ to true and the variables in $var(\varphi) \setminus \mathcal{I}$ to false. In this case we write $\mathcal{I} \models \varphi$. The *weight* of an interpretation (model) is its cardinality. We call a model $\mathcal{M}$ *(subset) minimal* if there exists no model $\mathcal{M}' \subset \mathcal{M}$, i.e., $\mathcal{M}'$ is a proper subset of $\mathcal{M}$.

Assignments and reduced formulas. Given a formula φ, an *assignment* of a set $\mathcal{V} \subseteq var(\varphi)$ is a pair $(\mathcal{T}, \mathcal{F})$ such that $\mathcal{T} \cup \mathcal{F} = \mathcal{V}$ and $\mathcal{T} \cap \mathcal{F} = \emptyset$. The set $\mathcal{T}$ denotes the variables that are set to true; the set $\mathcal{F}$ those that are set to false. Given an assignment $(\mathcal{T}, \mathcal{F})$ and a CNF formula φ, the *reduced formula* $\varphi[\mathcal{T}, \mathcal{F}]$ is φ where all variables in $\mathcal{T}$ are set to true and all variables in $\mathcal{F}$ are set to false. More specifically, $\varphi[\mathcal{T}, \mathcal{F}]$ is obtained from φ by first removing all clauses that contain variables in $\mathcal{T}$ as positive literals or variables in $\mathcal{F}$ as negative literals and secondly, removing all remaining literals of variables in $\mathcal{T} \cup \mathcal{F}$. In case the empty clause is produced by this procedure, $\varphi[\mathcal{T}, \mathcal{F}]$ is not satisfiable and hence we define $\varphi[\mathcal{T}, \mathcal{F}] := \{\emptyset\}$.

Parameterized complexity theory. We denote the input size, i.e., the size of the encoding of the instance, by n. In contrast to classical complexity theory, a parameterized complexity analysis studies the runtime of an algorithm with respect to one or more parameters $k_1, \ldots, k_l \in \mathbb{N}$ together with the input size n. A problem parameterized by $k_1, \ldots, k_l$ is *fixed-parameter tractable* (fpt) if there exists a computable function f and a constant c such that there is an algorithm solving it in time $\mathcal{O}(f(k_1, \ldots, k_l) \cdot n^c)$. Such an algorithm is called *fixed-parameter tractable* as well. We define parameterized problems as subsets of $\Sigma^* \times \mathbb{N}$, where Σ is the input alphabet. If a problem is parameterized by two or more parameters, the second component of an instance (x, k) corresponds to the sum of all parameter values. The class FPT consists of all parameterized problems that are fixed-parameter tractable. In order to show parameterized intractability results, we make use of fpt-reductions.

Definition. Let L_1 and L_2 be parameterized problems, i.e., $L_1 \subseteq \Sigma_1^* \times \mathbb{N}$ and $L_2 \subseteq \Sigma_2^* \times \mathbb{N}$. An *fpt-reduction* from L_1 to L_2 is a mapping $R : \Sigma_1^* \times \mathbb{N} \to \Sigma_2^* \times \mathbb{N}$ such that

1. $(I, k) \in L_1$ iff $R(I, k) \in L_2$.
2. R is computable by an fpt-algorithm with parameter k.
3. There is a computable function g such that for $R(I, k) = (I', k')$, $k' \leq g(k)$ holds.

We now define the parameterized complexity classes that will be needed in this work. A central problem, which can be used to define the so-called W-hierarchy, is WSAT$_=$. Given a CNF formula φ and an integer k, the question is whether φ can be satisfied by setting exactly k variables to true. W[1] can be defined as the class of problems fpt-reducible to WSAT$_=$ restricted to CNF formulas with clause size at most 2 (parameterized by the weight k). The general WSAT$_=$ is W[2]-complete. The class para-NP [8] is defined as the class of all problems which can be solved in fpt-time on a nondeterministic Turing machine. In particular all unparameterized problems that are in NP are in para-NP for *any* parameterization. If a problem remains NP-hard even when the parameter is set to a constant value, it is para-NP-hard. The following relations between these complexity classes are known: FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq$ para-NP. It is widely believed that problems that are hard for W[1] or higher classes are not fpt, i.e., FPT $\neq$ W[1]. Further details can be found, for example, in [4, 9].

3 Small Model Circumscription

Before we introduce the decision problem SMALL MODEL CIRCUMSCRIPTION, we give some fundamental definitions.

Definition. Let ψ be a formula and P, Q and Z pairwise disjoint sets with $P \cup Q \cup Z = var(\psi)$. Given two assignments $(\mathcal{T}, \mathcal{F})$ and $(\mathcal{T}', \mathcal{F}')$ with $\mathcal{T}' \cup \mathcal{F}' = \mathcal{T} \cup \mathcal{F}$ it holds that $(\mathcal{T}', \mathcal{F}') \prec_{\langle P,Q,Z \rangle} (\mathcal{T}, \mathcal{F})$ if $\mathcal{T}' \cap P \subset \mathcal{T} \cap P$ and $\mathcal{T}' \cap Q = \mathcal{T} \cap Q$. Given two interpretations $\mathcal{I}, \mathcal{I}' \subseteq var(\psi)$ it holds that $\mathcal{I}' \prec_{\langle P,Q,Z \rangle} \mathcal{I}$ if $(\mathcal{I}', var(\psi) \setminus \mathcal{I}') \prec_{\langle P,Q,Z \rangle} (\mathcal{I}, var(\psi) \setminus \mathcal{I})$.

Definition. A model $\mathcal{M}$ of some formula ψ is a $\langle k; P, Q, Z \rangle$-minimal model of ψ if there exists no $\mathcal{M}' \models \psi$ with $|\mathcal{M}'| \leq k$ and $\mathcal{M}' \prec_{\langle P,Q,Z \rangle} \mathcal{M}$.

In this work we will study the complexity of the following decision problem.

SMALL MODEL CIRCUMSCRIPTION (SMC)	
Instance:	A tuple $(\varphi, \pi, k, P, Q, Z)$, where φ and π are CNF formulas such that $var(\pi) \subseteq var(\varphi)$, $k \in \mathbb{N}$ and three pairwise disjoint sets P, Q and Z such that $P \cup Q \cup Z = var(\varphi)$.
Question:	Is there a $\langle k; P, Q, Z \rangle$-minimal model $\mathcal{M}$ of φ with $|\mathcal{M}| \leq k$ that is also a model of π?

SMC was first defined in [12] with the restriction that $Q = \emptyset$. We perform a complete parameterized complexity analysis of SMC with

Table 1. List of considered parameters. Unless otherwise mentioned all these parameters refer to φ.

k	the maximum weight of the minimal model searched for
d	the maximum clause size
d^+, d^-	the maximum positive/negative clause size, i.e., only positive/negative literals are counted
h	the number of non-Horn clauses
b	the size of a strong Horn backdoor set (strong Horn backdoor sets will be explained in Section 5)
d^+_π	the maximum positive clause size in π
$\|\pi\|$	the length of π, i.e., the total number of variable occurrences in π
$\|P\|, \|Q\|, \|Z\|$	the cardinality of the set P/Q/Z

respect to the parameters listed in Table 1. Thereby we show for each combination of parameters either a fixed-parameter tractability or a parameterized hardness result.

The first result studies SMC parameterized by the number of variables $var(\varphi) = P \cup Q \cup Z$. The $\mathcal{O}(4^{var(\varphi)} \cdot n)$ runtime of the trivial algorithm (guessing a model, then checking minimality) can be improved if we distinguish between the three types of variables.

Proposition 1. SMC *can be solved in time* $\mathcal{O}\big(2^{|Q|} \cdot 3^{|P|} \cdot 4^{|Z|} \cdot n\big)$ *and is therefore fpt with respect to* $|P|$, $|Q|$ *and* $|Z|$.

Proof. We consider all assignments of $var(\varphi)$. There are $2^{|var(\varphi)|} = 2^{|P|} \cdot 2^{|Q|} \cdot 2^{|Z|}$ of them. For the minimality check observe that the assignment of the Q-variables is fixed. Thus, there are at most $2^{|P|} \cdot 2^{|Z|}$ sets to be considered. We can, however, establish a tighter bound on the variables in P. For each assignment of P setting i variables to true, we have to consider $2^i - 1$ subsets. Moreover, for each i there are $\binom{|P|}{i}$ possible assignments. Hence considering all assignments of $var(\varphi)$ and verifying minimality requires $2^{|Q|} \cdot 4^{|Z|} \cdot \sum_{i=0}^{|P|} \big(\binom{|P|}{i} \cdot (2^i - 1)\big) \cdot \mathcal{O}(n) = \mathcal{O}\big(2^{|Q|} \cdot 3^{|P|} \cdot 4^{|Z|} \cdot n\big)$ time. □

4 Fpt-Algorithms Based on Bounded Search Trees

The fpt-algorithms in this section are based on bounded search trees, i.e., the size of the search tree can be bounded with respect to the parameters. This approach has already been successfully applied in [16] for finding minimal models. We extend this approach to $\langle k; P, Q, Z\rangle$-minimal models. In this paper we will frequently make use of a recursive procedure called $branch(\psi, k, \mathcal{S})$, which uses a bounded search tree. This procedure returns a set of some (but possibly not all) models of ψ that (i) have weight $\leq k$ and (ii) are supersets of $\mathcal{S} \subseteq var(\varphi)$. If a model satisfying (i) and (ii) is minimal, it is guaranteed to be included in the set.

The main idea of this procedure is as follows: If every clause in $\psi[\mathcal{S}, \emptyset]$ contains a negative literal then $\mathcal{S}$ is a model for ψ and is returned as a candidate for being a minimal model. Since we are looking for minimal models it certainly makes no sense to continue on this branch. If there exists a purely positive clause in $\psi[\mathcal{S}, \emptyset]$, let $(x_1 \vee \ldots \vee x_c)$ be this clause. Then the procedure branches on this clause by calling $branch(\psi, k, \mathcal{S} \cup \{x_i\})$ for each $i \in [c]$. In each model of $\psi[\mathcal{S}, \emptyset]$ at least one of these variables has to be true.

Since $branch(\psi, k, \mathcal{S})$ considers only models of weight $\leq k$, the maximum depth of the recursion is k. Observe that the runtime of $branch(\psi, k)$ is bounded by $\mathcal{O}((d^+)^k \cdot n)$. A more detailed description of $branch$ can be found in [16]. To simplify the notation we write $branch(\psi, k)$ for $branch(\psi, k, \emptyset)$. The following fact will be essential for the correctness of some of the algorithms for SMC.

Lemma 2 (cf. [16])**.** *The set of models returned by* $branch(\psi, k)$ *contains all subset minimal models of* $\psi \in$ CNF *with weight* $\leq k$.

We are now prepared to present the first algorithm.

Algorithm 1: Fpt-algorithm for k, d^+, $|Q|$ and $|Z|$ – Theorem 3

1 **foreach** *assignment* $(\mathcal{T}, \mathcal{F})$ *of* $Q \cup Z$ *with* $|\mathcal{T}| \leq k$ **do**
2     $\mathcal{C} \leftarrow branch(\varphi[\mathcal{T}, \mathcal{F}], k - |\mathcal{T}|)$.
3   **foreach** *model* $\mathcal{M}' \in \mathcal{C}$ **do**
4   Let $\mathcal{M} := \mathcal{M}' \cup \mathcal{T}$.
5   **if** $\mathcal{M} \models \pi$ **then**
6   Let $\hat{Q} := Q \cap \mathcal{M}$.
7   **foreach** $P' \subset (\mathcal{M} \cap P)$ **do**
8   Let $\varphi' := \varphi\Big[P' \cup \hat{Q}, (P \setminus P') \cup (Q \setminus \hat{Q})\Big]$.
9                 $\mathcal{C}' \leftarrow branch(\varphi', k - |P' \cup \hat{Q}|)$.
10   **if** $\mathcal{C}' \neq \emptyset$ **then** consider next model in Line 3
11   **return** *Yes*
12 **return** *No*

Theorem 3. *Algorithm 1 solves* SMC *in time*

$$\mathcal{O}\big((\max(|Q| + |Z|, d^+))^k \cdot (d^+ + 1)^k \cdot k \cdot n\big)$$

and is therefore fpt with respect to k, d^+*,* $|Q|$ *and* $|Z|$*.*

Proof. Algorithm 1 works as follows. Line 1 considers all assignments $(\mathcal{T}, \mathcal{F})$ of $Q \cup Z$. While Z-variables – contrary to Q-variables – do not have an impact on minimality, they may influence the satisfiability of π. Therefore all assignments of the Z-variables (in π) have to be considered. Notice that the formula $\varphi[\mathcal{T}, \mathcal{F}]$ contains only P-variables. In Line 2 we employ $branch$ to compute a set $\mathcal{C}$ of models containing all minimal models of $\varphi[\mathcal{T}, \mathcal{F}]$ (cf. Lemma 2). We are only interested in a model $\mathcal{M}' \in \mathcal{C}$ if $\mathcal{M}' \cup \mathcal{T} \models \pi$. Let $\mathcal{M} := \mathcal{M}' \cup \mathcal{T}$. Note that $\mathcal{M}$ has weight $\leq k$. It remains to be verified that the model $\mathcal{M}$ is $\langle k; P, Q, Z\rangle$-minimal. This is done in the loop in Line 7. There we construct for each subset $P' \subset \mathcal{M} \cap \mathcal{P}$ a reduced formula φ'. The formula φ' is obtained from φ by setting the variables in Q as in the assignment $(\mathcal{T}, \mathcal{F})$ and the variables in P according to the subset P'. Observe, that $var(\varphi') \subseteq Z$. If φ' is satisfiable then $\mathcal{M}$ is not $\langle k; P, Q, Z\rangle$-minimal and the next model $\mathcal{M}' \in \mathcal{C}$ is considered in Line 3. If φ' is not satisfiable for all $P' \subset \mathcal{M} \cap \mathcal{P}$ then $\mathcal{M}$ is a $\langle k; P, Q, Z\rangle$-minimal model of φ that additionally satisfies π.

Runtime. Observe that there are $(|Q| + |Z|)^s$ many assignments $(\mathcal{T}, \mathcal{F})$ of $Q \cup Z$ with weight $\leq s$, $s \leq k$. Given a weight s assignment the set $\mathcal{C}$ – computed in Line 2 – is of cardinality at most $(d^+)^{k-s}$. This yields that the number of models $\mathcal{M}$ considered in Line 4 is at most $\sum_{s=0}^{k} (|Q| + |Z|)^s \cdot (d^+)^{k-s} \leq \sum_{s=0}^{k} (\max(|Q| + |Z|, d^+))^k = (k+1) \cdot (\max(|Q| + |Z|, d^+))^k$. Minimality is ensured in the loop in Line 7. There for each subset of $\mathcal{M} \cap P$ the set $\mathcal{C}'$ is computed. For a given model $\mathcal{M}$ the runtime of the minimality check is $\sum_{i=0}^{k} \binom{|\mathcal{M} \cap P|}{i} \cdot (d^+)^{k-i} \cdot \mathcal{O}(n) \leq \sum_{i=0}^{k} \binom{k}{i} \cdot (d^+)^{k-i} \cdot \mathcal{O}(n) = (d^+ + 1)^k \cdot \mathcal{O}(n)$. □

Lemma 4. *Let* $(\varphi, \pi, k, P, Q, Z)$ *be an* SMC *instance. Furthermore, let* $\mathcal{M}$ *be a* $\langle k; P, Q, Z\rangle$*-minimal model of* φ *and a model of* π*. Then* $\mathcal{M}$ *is also a* $\langle k; P, Q, Z\rangle$*-minimal model of* $\varphi \wedge \pi$*.*

Proof. Assume towards a contradiction that $\mathcal{M}$ is not a $\langle k; P, Q, Z\rangle$-minimal model of $\varphi \wedge \pi$. Clearly $\mathcal{M} \models \varphi \wedge \pi$. Since $\mathcal{M}$ is not an $\langle k; P, Q, Z\rangle$-minimal model of $\varphi \wedge \pi$, there must be a $\mathcal{M}' \prec_{\langle P,Q,Z\rangle} \mathcal{M}$ with $|\mathcal{M}'| \leq k$. However, then $\mathcal{M}'$ is also a model of φ which contradicts the assumption that $\mathcal{M}$ is a $\langle k; P, Q, Z\rangle$-minimal model of φ. □

The parameterization in the next theorem is almost the same as in the previous one except that $|Z|$ has been replaced with d^+_π.

Algorithm 2: Fpt-algorithm for k, d^+, $|Q|$ and d^+_π – Theorem 5

1 **foreach** *assignment* $(\mathcal{T}, \mathcal{F})$ *of* Q *with* $|\mathcal{T}| \leq k$ **do**
2 $\quad \mathcal{C} \leftarrow branch(\varphi[\mathcal{T}, \mathcal{F}] \wedge \pi[\mathcal{T}, \mathcal{F}], k - |\mathcal{T}|)$.
3 $\quad$ **foreach** *model* $\mathcal{M}' \in \mathcal{C}$ **do**
4 $\quad\quad \mathcal{M} \leftarrow \mathcal{M}' \cup \mathcal{T}$
5 $\quad\quad$ Let $\hat{Q} := Q \cap \mathcal{M}$.
6 $\quad\quad$ **foreach** $\mathcal{P}' \subset (\mathcal{M} \cap P)$ **do**
7 $\quad\quad\quad$ Let $\varphi' := \varphi\left[P' \cup \hat{Q}, (P \setminus P') \cup (Q \setminus \hat{Q})\right]$.
8 $\quad\quad\quad \mathcal{C}' \leftarrow branch(\varphi', k - |P' \cup \hat{Q}|)$.
9 $\quad\quad\quad$ **if** $\mathcal{C}' \neq \emptyset$ **then** consider next model in Line 3
10 $\quad\quad$ **return** *Yes*
11 **return** *No*

Theorem 5. *Algorithm 2 solves* SMC *in time*

$$\mathcal{O}\big((\max(|Q|, d^+, d^+_\pi))^k \cdot (d^+ + 1)^k \cdot k \cdot n\big)$$

and is therefore fpt with respect to k, d^+, $|Q|$ and d^+_π.

Proof. The minimality check in Algorithm 2 is identical to the one in Algorithm 1. Observe that given an assignment of Q all relevant models are produced by *branch* in Line 2. This follows from Lemma 4 which implies that every $\langle k; P, Q, Z\rangle$-minimal model of $\varphi[\mathcal{T}, \mathcal{F}]$ that is also a model of $\pi[\mathcal{T}, \mathcal{F}]$ is a $\langle k; P, Q, Z\rangle$-minimal model of $\varphi[\mathcal{T}, \mathcal{F}] \wedge \pi[\mathcal{T}, \mathcal{F}]$. The time bound can be seen by arguments similar to the ones in the proof of Theorem 3. □

5 Fpt-Algorithms Based on Backdoor Sets

In this section we make use of strong Horn backdoor sets. A *strong Horn backdoor set* of a formula φ is a set $B \subseteq var(\varphi)$ such that for each assignment $(\mathcal{T}, \mathcal{F})$ of B, $\varphi[\mathcal{T}, \mathcal{F}]$ is a Horn formula. Consider for example the formula $\psi := (a \vee b) \wedge (\neg d \vee a \vee c) \wedge (b \vee c)$. In this case $\{a, b\}$ is a strong Horn backdoor set of φ. A Horn formula has the advantageous property that checking satisfiability by computing its unique minimal model can be done in linear time [3].

Backdoor sets were introduced in the context of FPT in [19]. Finding a strong Horn backdoor set of size b is equivalent to finding a vertex cover of size b. The best known algorithm solves this problem in $\mathcal{O}(1.2738^b + b \cdot n)$, see [2]. We do not include the vertex cover computation in our algorithms and hence consider the backdoor set as additional input.

Theorem 6. *Given a strong Horn backdoor set B of cardinality b, Algorithm 3 solves* SMC *in time*

$$\mathcal{O}\big((\max(1, d^+_\pi, b + |Q|))^k \cdot (b+1)^k \cdot k \cdot n\big)$$

and is therefore fpt with respect to k, b, $|Q|$ and d^+_π.

Algorithm 3: Fpt-algorithm for k, b, $|Q|$ and d^+_π – Theorem 6

1 **foreach** *assignment* $(\mathcal{T}, \mathcal{F})$ *of* $B \cup Q$ *with* $|\mathcal{T}| \leq k$ **do**
2 $\quad \mathcal{C} \leftarrow branch(\varphi[\mathcal{T}, \mathcal{F}] \wedge \pi[\mathcal{T}, \mathcal{F}], k - |\mathcal{T}|)$.
3 $\quad$ **foreach** *model* $\mathcal{M} \in \mathcal{C}$ **do**
4 $\quad\quad$ **foreach** *assignment* $(\mathcal{T}', \mathcal{F}')$ *of* $B \cup Q$ *with* $\mathcal{T}' \cap Q = \mathcal{T} \cap Q$, $\mathcal{T}' \cap P \subseteq \mathcal{T} \cap P$ *and* $|\mathcal{T}'| \leq k$ **do**
$\quad\quad\quad$ `// ` $\varphi[\mathcal{T}', \mathcal{F}']$ `is a Horn formula`
5 $\quad\quad\quad$ **if** $\varphi[\mathcal{T}', \mathcal{F}']$ *is satisfiable* **then**
6 $\quad\quad\quad\quad$ Let $\mathcal{M}'$ be the unique minimal model of $\varphi[\mathcal{T}', \mathcal{F}']$.
7 $\quad\quad\quad\quad$ **if** $\mathcal{M}'$ *is of weight* $\leq k - |\mathcal{T}'|$ *and* $(\mathcal{M}' \cup \mathcal{T}') \cap P \subset (\mathcal{M} \cup \mathcal{T}) \cap P$ **then**
8 $\quad\quad\quad\quad\quad$ Consider the next model in Line 3.
9 $\quad\quad$ **return** *Yes*
10 **return** *No*

Proof. The algorithm starts by generating all assignments $(\mathcal{T}, \mathcal{F})$ of $B \cup Q$ of weight $\leq k$ (Line 1). Then, it executes the procedure *branch* for the formula $\varphi[\mathcal{T}, \mathcal{F}] \wedge \pi[\mathcal{T}, \mathcal{F}]$ in order to find all minimal models of weight $\leq k - |\mathcal{T}|$ (Line 2). It follows from Lemma 4 that indeed all relevant models are considered this way. For each model $\mathcal{M}$ returned by the procedure *branch* it holds that $\mathcal{M} \cup \mathcal{T} \models \varphi \wedge \pi$. The loop in Line 4 checks whether $\mathcal{M} \cup \mathcal{T}$ is minimal. There each assignment $(\mathcal{T}', \mathcal{F}')$ of $B \cup Q$ is considered with $\mathcal{T}' \cap Q = \mathcal{T} \cap Q$, $\mathcal{T}' \cap P \subseteq \mathcal{T} \cap P$ and $|\mathcal{T}'| \leq k$. Notice that we allow here that $\mathcal{T}' \cap P = \mathcal{T} \cap P$. For any such $(\mathcal{T}', \mathcal{F}')$, $\varphi[\mathcal{T}', \mathcal{F}']$ is a Horn formula and hence we can compute the unique minimal model $\mathcal{M}'$ in linear time. In case $\mathcal{M}'$ is of weight $\leq k - |\mathcal{T}'|$ and fulfills $(\mathcal{M}' \cup \mathcal{T}') \cap P \subset (\mathcal{M} \cup \mathcal{T}) \cap P$, we know that $\mathcal{M} \cup \mathcal{T}$ is not $\langle k; P, Q, Z\rangle$-minimal and the next model is considered in Line 3. Otherwise, if no such $\mathcal{M}'$ can be found in the loop, the algorithm returns Yes.

There are $(b + |Q|)^s$ assignments of $B \cup Q$ of weight $s \leq k$. Next we consider the runtime of the *branch* procedure. Note that $\varphi[\mathcal{T}, \mathcal{F}]$ contains at most one positive literal per clause and $\pi[\mathcal{T}, \mathcal{F}]$ contains at most d^+_π positive literals per clause. Therefore, $\varphi[\mathcal{T}, \mathcal{F}] \wedge \pi[\mathcal{T}, \mathcal{F}]$ contains $\leq \max(1, d^+_\pi)$ positive literals per clause. With a similar calculation as in the proof of Theorem 3 we obtain that the first two loops iterate over at most $(k+1) \cdot (\max(1, d^+_\pi, b + |Q|))^k$ models.

We now bound the time required to verify minimality. This mainly depends on the number of considered subassignments $(\mathcal{T}', \mathcal{F}')$. All these subassignments have weight $\leq k$ and it has to hold that $\mathcal{T}' \cap P \subseteq \mathcal{T} \cap P$ and $\mathcal{T}' \cap Q = \mathcal{T} \cap Q$. The variables in $Z \cap B$ may vary between $(\mathcal{T}', \mathcal{F}')$ and $(\mathcal{T}, \mathcal{F})$. Hence the number of considered subassignments is $\sum_{s=0}^{k} \binom{k}{s} \cdot b^{k-s} = (b+1)^k$. The first factor in this sum is the number of possibilities for $\mathcal{T}' \cap P \subseteq \mathcal{T} \cap P$ where $|\mathcal{T}' \cap P| = s$. The second factor is the number of possibilities to choose Z-variables in B to be set to true (at most $k - s$ many). For each subassignment the unique minimal model of a Horn formula has to be computed – in linear time. □

Corollary 7. *Algorithm 3 solves* SMC *in time*

$$\mathcal{O}\big((\max(1, d^+_\pi, h \cdot d^+ + |Q|))^k \cdot (h \cdot d^+ + 1)^k \cdot k \cdot n\big)$$

and is therefore fpt with respect to k, d^+, h, $|Q|$ and d^+_π.

Proof. We use the fact that the set of all positive literals appearing in non-Horn clauses forms a strong Horn backdoor set. The cardinality of this backdoor set is at most $h \cdot d^+$. Thus, the fpt result follows from Theorem 6. The runtime of Algorithm 3 applies as well. $\square$

Algorithm 4: Fpt-algorithm for b, $|Q|$ and at least one of $\|\pi\|$ and $|Z|$ – Theorem 8

1 Let $Z_\pi := \mathit{var}(\pi) \cap Z$.
2 **foreach** *assignment* $(\mathcal{T}, \mathcal{F})$ *of* $B \cup Q \cup Z_\pi$ *with* $|\mathcal{T}| \leq k$ **do**
3   **if** $\varphi[\mathcal{T}, \mathcal{F}]$ *is satisfiable* **then** // $\varphi[\mathcal{T}, \mathcal{F}]$ is Horn
4   Let $\mathcal{M}$ be the unique minimal model of $\varphi[\mathcal{T}, \mathcal{F}]$.
5   **if** $|\mathcal{M}| \leq k - |\mathcal{T}|$ *and* $\mathcal{M} \cup \mathcal{T} \models \pi$ **then**
6   **foreach** *assignment* $(\mathcal{T}', \mathcal{F}')$ *of* $B \cup Q$ *with* $\mathcal{T}' \cap Q = \mathcal{T} \cap Q$, $\mathcal{T}' \cap P \subseteq \mathcal{T} \cap P$ *and* $|\mathcal{T}'| \leq k$ **do**
7   **if** $\varphi[\mathcal{T}', \mathcal{F}']$ *is satisfiable* **then**
8   Let $\mathcal{M}'$ be the unique minimal model of $\varphi[\mathcal{T}', \mathcal{F}']$.
9   **if** $\mathcal{M}'$ *is of weight* $\leq k - |\mathcal{T}'|$ *and* $(\mathcal{M}' \cup \mathcal{T}') \cap P \subset (\mathcal{M} \cup \mathcal{T}) \cap P$ **then**
10   Consider next assignment in Line 2.
11   **return** *Yes*
12 **return** *No*

Theorem 8. *Given a strong Horn backdoor set B of cardinality b, Algorithm 4 solves* SMC *in time* $\mathcal{O}\big(4^b \cdot 2^{|Q|} \cdot 2^{\min(\|\pi\|,|Z|)} \cdot n\big)$ *and is therefore fpt with respect to b, $|Q|$ and at least one of $\|\pi\|$ and $|Z|$.*

Proof. The algorithm considers all assignments $(\mathcal{T}, \mathcal{F})$ of $B \cup Q \cup Z_\pi$. This means that we only consider Z-variables in this assignment that appear either in B or π. If a Z-variable is not contained in π, it can vary without influencing the satisfiability of π. Furthermore, the truth value of such a Z-variable is determined by the assignment of B. In Line 4 the formula $\varphi[\mathcal{T}, \mathcal{F}]$ is a Horn formula whose unique minimal model can be computed in linear time. Checking whether the model $\mathcal{M} \cup \mathcal{T}$ is of weight $\leq k$ and whether π is satisfied can also be done in linear time. If such a model is found we have to check whether the model is minimal. The minimality check is identical to the one in Algorithm 3.

In Line 2 the number of considered assignments is at most $2^{|B|} \cdot 2^{|Q|} \cdot 2^{|Z \cap \mathit{var}(\pi)|}$. The minimality check is done only with respect to variables in B since the variables in Q are fixed and the values of variables in $P \setminus B$ are implied by the variables in B. Therefore, at most $2^b - 1$ subsets have to be checked. $\square$

Corollary 9. *Algorithm 4 solves* SMC *in time*

$$\mathcal{O}\big(4^{h \cdot d^+} \cdot 2^{|Q|} \cdot 2^{\min(\|\pi\|,|Z|)} \cdot n\big)$$

and is therefore fpt with respect to d^+, h, $|Q|$ and at least one of $\|\pi\|$ and $|Z|$.

Proof. In Corollary 7 we have already used the fact that $b \leq h \cdot d^+$. This time we use it to obtain the above results from Theorem 8. $\square$

6 Hardness Results for SMC

In the previous sections several fpt-algorithms have been presented. As an immediate consequence SMC is also fpt with respect to every superset of these combinations of parameters. We are now going to show that these are all fpt-algorithms possible for the parameters listed in Table 1. This is achieved by showing parameterized hardness results for all remaining combinations of parameters. This completes our parameterized complexity analysis of SMC.

The next theorem shows that $|Q|$ has to be included in any combination of parameters yielding an fpt result. This is shown by proving hardness of SMC with respect to all parameters except $|Q|$. The proof builds upon Theorem 14 in [16] but extends the construction in such way that even the parameter d can be bounded.

Theorem 10. SMC *parameterized by k, d, d^+, d^-, h, b, d^+_π, $\|\pi\|$, $|P|$ and $|Z|$ is* W[1]*-hard.*

Proof. We give an fpt-reduction from the Independent Set problem. Given a graph (V, E) and an integer s, the Independent Set problem asks for a subset V' of the vertices of cardinality s such that the graph does not contain edges between vertices in V'. This problem is W[1]-complete when parameterized by s, see e.g. [4].

We construct an SMC instance as follows. Let the vertices be $V = \{v_1, \ldots, v_m\}$. The variables used in φ are going to be $\{v_1, \ldots, v_m\} \cup \{v_1^1, \ldots, v_m^1, \ldots, v_1^s, \ldots, v_m^s\} \cup \{c_1, \ldots, c_s\}$. We construct the formulas

$$\varphi_{\mathrm{IS}} := \bigwedge_{\{x,y\} \in E} (\neg x \vee \neg y) \qquad \varphi_1 := \bigwedge_{i \in [m]} \bigwedge_{1 \leq l < l' \leq s} (\neg v_i^l \vee \neg v_i^{l'})$$

$$\varphi_2 := \bigwedge_{i \in [m]} \bigwedge_{l \in [s]} (v_i^l \rightarrow v_i) \qquad \varphi_3 := \bigwedge_{l \in [s]} \bigwedge_{i \in [m]} (v_i^l \rightarrow c_l).$$

We now define $\varphi := \varphi_{\mathrm{IS}} \wedge \varphi_1 \wedge \varphi_2 \wedge \varphi_3$, $\pi := c_1 \wedge \ldots \wedge c_s$ and $k := 3s$. The SMC instance is then given by $(\varphi, \pi, k, P, Q, Z)$, where $P := \{c_1, \ldots, c_s\}$, $Q := \mathit{var}(\varphi) \setminus P$ and $Z := \emptyset$. Recall that the variables in Q are not subject to minimization. We continue by explaining the functionality of the subformulas. Subformula φ_{IS} encodes the independent set property. It enforces that it is not possible to choose two vertices connected by an edge. Subformula φ_1 ensures that it is not possible to set more than one copy of each vertex to true. Subformula φ_2 ensures that a vertex has to be in the model if one of its copies is in the model. Subformula φ_3 together with π enforces that at least s vertices are chosen to be in the independent set. The correctness proof is omitted due to space constraints.

We are now going show that all parameters can be bounded in terms of s. Observe that $d^+_\pi = 1$ and $\|\pi\| = s$ since all clauses in π are of size one and there are exactly s of them. All clauses in φ are of size two and hence d, d^+ and d^- are ≤ 2. There are no non-Horn clauses in φ. Thus, b and h are equal to 0. Finally, observe that $k = 3s$, $|P| = s$ and $|Z| = 0$. $\square$

Theorem 11. SMC *parameterized by k, d, d^+, d^-, h, b, $|P|$ and $|Q|$ is* W[2]*-hard.*

Proof. We reduce from the W[2]-complete Monotone Weighted Satisfiability problem [4, 9]. Given a monotone formula ψ in CNF and an integer k' this problem asks whether there is a model of ψ setting exactly k' variables to true. The SMC instance

$(\varphi, \pi, k, P, Q, Z)$ is defined as $\varphi := \bigwedge_{x \in \mathit{var}(\psi)}(x \vee \neg x)$, $\pi := \psi$, $k := k'$, $P := \emptyset$, $Q := \emptyset$ and $Z := \mathit{var}(\varphi)$. The correctness follows from the monotonicity of the problem, i.e., each superset of a model is a model as well. The parameter bounds are $d \leq 2$, $d^+ \leq 1$, $d^- \leq 1$ and $h = b = |P| = |Q| = 0$. □

Theorem 12. SMC *parameterized by d, d^+, d^-, h, b, d^+_π, $|P|$ and $|Q|$ is* para-NP*-hard.*

Proof. We reduce from the NP-complete 3-SAT problem, i.e., the satisfiability problem over propositional formulas in 3-CNF (which are CNF formulas with clause size at most 3). Given a formula ψ in 3-CNF we create an SMC instance by setting $\varphi := \bigwedge_{x \in \mathit{var}(\psi)}(x \vee \neg x)$, $\pi := \psi$, $k := |\mathit{var}(\psi)|$ (this is possible since k is not a parameter in this reduction), $P := \emptyset$, $Q := \emptyset$ and $Z := \mathit{var}(\psi)$. Concerning the parameters, we can bound d^+_π by 3, while $d \leq 2$, $d^+ \leq 1$, $d^- \leq 1$ and $h = b = |P| = |Q| = 0$. □

The last two theorems make use of results for the Weighted Minimal Model SAT problem (WMMSAT) discussed in [16]. This problem is a special case of SMC where $\mathit{var}(\varphi) = P$. In both proofs the minimization is irrelevant and hence it does not matter whether the variables of φ are contained in P, Q or Z.

Theorem 13 (follows from [16], Thm 14)**.** SMC *parameterized by k, d^-, h, d^+_π, $\|\pi\|$ and two elements of $\{P, Q, Z\}$ is* W[1]*-hard.*

Theorem 14 (follows from [16], Prop 18)**.** SMC *parameterized by d, d^+, d^-, d^+_π, $\|\pi\|$ and two elements of $\{P, Q, Z\}$ is* para-NP*-hard.*

This concludes our parameterized analysis. We remark that a parameterized hardness result with respect to a set of parameters implies hardness for any subset of these parameters. Using this fact, one can check that the fpt-algorithms together with the hardness results indeed cover all 2^{11} combinations of parameters.

7 Conclusion

In this paper we have performed a complete parameterized complexity analysis of SMC with respect to the parameters in Table 1. An interesting observation is that Algorithm 4 does not make use of the parameter k. Hence the runtime upper bound is not affected if we set $k = \mathit{var}(\varphi)$ (i.e., we ask for models containing at most *all* variables). This effectively eliminates the restriction to small models for Algorithm 4. However, all other algorithms require this parameter.

SMC restricts the number of P-, Q- and Z-variables set to true in a model. In some settings it makes more sense to limit only the number of P-variables set to true. The number of Q- and Z-variables in the model would be irrelevant, e.g., because these are auxiliary variables. It remains future work to explore the parameterized complexity of the parameter "number of P-variables set to true".

SMC asks whether there *exists* a $\langle k; P, Q, Z\rangle$-minimal model and is therefore a brave reasoning problem. Additional insight can be gained from analyzing the dual cautious reasoning problem, i.e., the question of whether all $\langle k; P, Q, Z\rangle$-minimal models satisfy a certain property. Observe that, for example, Algorithm 1 requires only small modifications to be able to handle cautious reasoning. In contrast, Algorithm 2 only generates models that satisfy π and is therefore not applicable to cautious reasoning.

Finally, improving the runtime of the presented algorithms, the study of further parameters and the search for kernelization results [14] is left as future work as well.

REFERENCES

[1] Marco Cadoli and Maurizio Lenzerini, 'The complexity of propositional closed world reasoning and circumscription', *J. Comput. Syst. Sci.*, **48**(2), 255–310, (1994).

[2] Jianer Chen, Iyad A. Kanj, and Ge Xia, 'Improved upper bounds for vertex cover', *Theor. Comput. Sci.*, **411**(40-42), 3736–3756, (2010).

[3] William F. Dowling and Jean H. Gallier, 'Linear-time algorithms for testing the satisfiability of propositional Horn formulae', *J. Log. Program.*, **1**(3), 267–284, (1984).

[4] Rodney G. Downey and Michael R. Fellows, *Parameterized Complexity*, Springer, 1999.

[5] Thomas Eiter and Georg Gottlob, 'Propositional circumscription and extended closed-world reasoning are Π^P_2-complete', *Theor. Comput. Sci.*, **114**(2), 231–245, (1993).

[6] Paolo Ferraris, Joohyung Lee, and Vladimir Lifschitz, 'Stable models and circumscription', *Artif. Intell.*, **175**(1), 236–263, (2011).

[7] Johannes Klaus Fichte and Stefan Szeider, 'Backdoors to tractable answer-set programming', in *Proc. of IJCAI 2011*, pp. 863–868. IJCAI/AAAI Press, (2011).

[8] Jörg Flum and Martin Grohe, 'Describing parameterized complexity classes', *Inf. Comput.*, **187**(2), 291–319, (2003).

[9] Jörg Flum and Martin Grohe, *Parameterized Complexity Theory*, Springer, 2006.

[10] Michael Gelfond, Halina Przymusinska, and Teodor C. Przymusinski, 'On the relationship between circumscription and negation as failure', *Artif. Intell.*, **38**(1), 75–94, (1989).

[11] Georg Gottlob, Reinhard Pichler, and Fang Wei, 'Bounded treewidth as a key to tractability of knowledge representation and reasoning', *Artif. Intell.*, **174**(1), 105–132, (2010).

[12] Georg Gottlob, Francesco Scarcello, and Martha Sideri, 'Fixed-parameter complexity in AI and nonmonotonic reasoning', *Artif. Intell.*, **138**(1-2), 55–86, (2002).

[13] Georg Gottlob and Stefan Szeider, 'Fixed-parameter algorithms for artificial intelligence, constraint satisfaction and database problems', *The Computer Journal*, **51**(3), 303–325, (2008).

[14] Jiong Guo and Rolf Niedermeier, 'Invitation to data reduction and problem kernelization', *SIGACT News*, **38**, 31–45, (March 2007).

[15] Michael Jakl, Reinhard Pichler, and Stefan Woltran, 'Answer-set programming with bounded treewidth', in *Proc. of IJCAI 2009*, pp. 816–822. IJCAI/AAAI Press, (2009).

[16] Martin Lackner and Andreas Pfandler, 'Fixed-parameter algorithms for finding minimal models', in *Proc. of KR 2012*. AAAI Press, (2012). To appear.

[17] Paolo Liberatore and Marco Schaerf, 'Reducing belief revision to circumscription (and vice versa)', *Artif. Intell.*, **93**, 261–296, (1997).

[18] John McCarthy, 'Circumscription - a form of non-monotonic reasoning', *Artif. Intell.*, **13**(1-2), 27–39, (1980).

[19] Naomi Nishimura, Prabhakar Ragde, and Stefan Szeider, 'Detecting backdoor sets with respect to Horn and binary clauses', in *Proc. of SAT 2004*, (2004).

[20] Reinhard Pichler, Stefan Rümmele, Stefan Szeider, and Stefan Woltran, 'Tractable answer-set programming with weight constraints: Bounded treewidth is not enough', in *Proc. of KR 2010*, pp. 508–517. AAAI Press, (2010).

[21] Raymond Reiter, 'On closed world data bases', in *Logic and Data Bases*, pp. 55–76. Plemum Press, (1977).

[22] Miroslaw Truszczynski, 'Computing large and small stable models', *Theory and Practice of Logic Programming*, **2**(1), 1–23, (2002).

[23] Ryan Williams, Carla P. Gomes, and Bart Selman, 'Backdoors to typical case complexity', in *Proc. of IJCAI 2003*, pp. 1173–1178. Morgan Kaufmann, (2003).