Fixed-Parameter Algorithms for Finding Minimal Models*

Martin Lackner and **Andreas Pfandler**
Vienna University of Technology, Austria
$\{$*lackner*, *pfandler*$\}$@*dbai.tuwien.ac.at*

Abstract

Computing minimal models is an important task in Knowledge Representation and Reasoning that appears in formalisms such as circumscription, diagnosis and answer set programming. Even the most basic question of whether there exists a minimal model containing a given variable is known to be Σ_2^P-complete.

In this work we study the problem of computing minimal models from the viewpoint of parameterized complexity theory. We perform an extensive complexity analysis of this problem with respect to eleven parameters. Tractable fragments based on combinations of these parameters are identified by giving several fixed-parameter algorithms. For the remaining combinations we show parameterized hardness results and thus prove that under usual complexity theoretic assumptions no further fixed-parameter algorithms exist for these parameters.

Introduction

Computing (subset) minimal models of propositional formulas is an important task in reasoning formalisms such as closed-world reasoning and circumscription (McCarthy 1980; Lifschitz 1985), diagnosis (Reiter 1987) and answer set programming (Marek and Truszczynski 1999; Gelfond and Leone 2002; Baral, Provetti, and Son 2003). However, finding minimal models is computationally hard: The corresponding decision problem – asking whether there is a minimal model containing a given variable – is Σ_2^P-complete (Eiter and Gottlob 1993). This problem is harder than the classical satisfiability problem since one has not only to guess the model but also to verify its minimality. Since computing minimal models is an ubiquitous task in many reasoning formalisms, ways have to be found to deal with its high computational complexity.

Parameterized complexity theory has been developed to cope with computationally hard problems. The main idea is to study the complexity of a problem from a multivariate point of view. This is in contrast to a classical complexity analysis where only the input size n is considered. Additional variables – so called parameters – describe characteristics of the input instance. The main concept is fixed-parameter tractability. A problem is called *fixed-parameter tractable* (fpt) with respect to the parameters $k_1, k_2, \ldots, k_l$ if it can be solved in time $\mathcal{O}(f(k_1, k_2, \ldots, k_l) \cdot n^c)$. Here f is a computable (usually exponential) function and c a constant. Under the assumption that the parameter values are of moderate size this runtime bound can be very favorable even for large instances. Thus, the search for fpt algorithms is a major technique to deal with hard problems.

Parameterized complexity theory has often and successfully been applied to problems in AI, as can be observed in the survey (Gottlob and Szeider 2008). Answer-set programming, for example, has been extensively studied (Truszczynski 2002; Jakl, Pichler, and Woltran 2009; Pichler et al. 2010; Fichte and Szeider 2011). Its underlying problem however – the computation of minimal models – has received less attention. In (Ben-Eliyahu and Dechter 1996) the first fpt algorithms for computing an arbitrary minimal model have been presented. However, the paper does not make use of parameterized complexity theory and therefore contains no systematic study of the interplay of parameters. A first parameterized complexity analysis of computing minimal models was performed in (Gottlob, Scarcello, and Sideri 2002). The focus of that paper lies on intractability proofs and presents an algorithm only for a rather restricted case. In (Chen and Flum 2008) the problem of whether there exists a minimal model of a specific size is studied but again the focus lies on intractability classifications. In (Gottlob, Pichler, and Wei 2010) fixed-parameter tractability with respect to the parameter tree-width is shown but no further parameters are considered.

The aim of this paper is to design efficient fpt algorithms for finding minimal models of a CNF formula φ that additionally satisfy a property π. This paper contains the first comprehensive parameterized complexity analysis of this problem. We consider eleven natural parameters which are all efficiently computable. These include the maximum clause size, the number of non-Horn clauses, how often variables occur as positive literals, the maximal cardinality of the model we are looking for, etc. (see Table 1).

Our fpt algorithms are especially efficient in case the corresponding parameter values are small. A potential area of

*The research was supported by the Austrian Science Fund (FWF): P20704-N18.

Copyright © 2012, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

application for our results is to use the fpt algorithms as extensions of existing methods. For example, we present an fpt algorithm parameterized by the maximum cardinality of the model and the maximum number of positive literals per clause. This algorithm is especially well-suited for applications in diagnosis where the theory is mostly rule-based, i.e. close to Horn formulas. Furthermore, in diagnosis one is mainly interested in small models (explanations). The maximum number of positive literals per clause can be computed in linear time and therefore one can efficiently decide whether or not to use this fpt algorithm.

We make full use of the fpt machinery by proving parameterized hardness results. These results rule out the possibility of further fpt algorithms – under usual complexity theoretic assumptions. We are able to show that this paper contains all possible fixed-parameter tractable fragments with regard to the parameters studied. Therefore we provide a complete parameterized complexity classification for the 2^{11} combinations of parameters. Our main contributions are:

- We present several fpt algorithms, each of which makes use of a different combination of parameters. These complement each other since they perform especially well on distinct classes of formulas. The results are of additional interest since so far few fpt algorithms for Σ_2^P-complete problems are known.
- For all other combinations of parameters we show that under usual complexity theoretic assumptions no fixed-parameter algorithms exist. This is achieved by elaborate fixed-parameter intractability proofs.
- In particular, we prove W[2]-completeness when parameterizing by the maximum cardinality of the model. This answers a long-standing open question posed in (Gottlob, Scarcello, and Sideri 2002).

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 size 2. Given a graph and a vertex v, the neighborhood $N(v) \subseteq V$ is the set containing v and all vertices connected to v by an edge. For $m \in \mathbb{N}$, we use $[m]$ to denote the set $\{1, \ldots, m\}$. The power set of a set A is denoted by $\mathbb{P}(A)$.

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 $\mathit{var}(\varphi)$ the set of variables occurring in φ. An *interpretation* $\mathcal{I} \subseteq \mathit{var}(\varphi)$ is a subset of the variables. An interpretation $\mathcal{I}$ is called a *model* (of the formula φ) if φ is satisfied by setting the variables in $\mathcal{I}$ to true and the variables in $\mathit{var}(\varphi) \setminus \mathcal{I}$ to false. The *weight* of an interpretation or a 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 $\mathcal{V} \subseteq \mathit{var}(\varphi)$ 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 second 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 $\varphi[\mathcal{T}, \mathcal{F}] := \{\emptyset\}$. We say that an assignment $(\mathcal{T}', \mathcal{F}')$ is a subassignment of an assignment $(\mathcal{T}, \mathcal{F})$, denoted by $(\mathcal{T}', \mathcal{F}') \prec (\mathcal{T}, \mathcal{F})$, if $\mathcal{T}' \cup \mathcal{F}' = \mathcal{T} \cup \mathcal{F}$ and there is some non-empty $\Delta \subseteq \mathcal{T}$ such that $(\mathcal{T}', \mathcal{F}') = (\mathcal{T} \setminus \Delta, \mathcal{F} \cup \Delta)$. Finally, $\varphi[\mathcal{T}]$ is an abbreviation of $\varphi[\mathcal{T}, \emptyset]$.

Parameterized complexity theory. In this paper 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 is 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.

WSat$_\leq$ / WSat$_=$

Instance: A formula $\varphi \in$ CNF and $k \in \mathbb{N}$.
Question: Does φ have a model with weight at most / exactly k ?

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$_=$ as well as WSat$_\leq$ are W[2]-complete. The class para-NP (Flum and Grohe 2003) 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 broadly believed that problems that are hard for W[1] or higher classes are not fpt, i.e. FPT $\neq$ W[1]. We exemplarily mention that Vertex Cover is in FPT, Independent Set is W[1]-complete and Dominating Set is W[2]-complete (all parameterized by the solution size). Further details can be found e.g. in (Downey and Fellows 1999; Flum and Grohe 2006).

Weighted Minimal Model Satisfiability

The main goal of this paper is to study the (parameterized) complexity of finding minimal models that satisfy a given (Cnf) property. We do this by studying the following decision problem:

Weighted Minimal Model Sat (WMMSat)

Instance: A triple (φ, π, k), where φ and π are Cnf formulas such that $\mathit{var}(\pi) \subseteq \mathit{var}(\varphi)$ and $k \in \mathbb{N}$.

Question: Is there a minimal model $\mathcal{M}$ of φ with $|\mathcal{M}| \leq k$ that is also a model of π?

Note that WMMSat does *not* ask the question whether there is a minimal model of $\varphi \wedge \pi$. Consider for example $\varphi := (x \vee \neg x)$ and $\pi := x$. This instance has no minimal model of φ that also satisfies π since the empty set is the only minimal model of φ. However, $\{x\}$ is a minimal model of $\varphi \wedge \pi$.

Also, observe that although WMMSat is a decision problem, it is closely related to the problem of actually computing a minimal model. The reason for this is that we can use repeated calls of a procedure solving WMMSat to actually compute a model. For this let φ be the formula we want to find a minimal model of and $\mathit{var}(\varphi) = \{x_1, \ldots, x_m\}$. First, we solve the WMMSat instance (φ, x_1, m). If this is a Yes-instance, we continue with the instance $(\varphi, x_1 \wedge x_2, m)$. Otherwise we continue with the instance $(\varphi, \neg x_1 \wedge x_2, m)$. Each call determines the truth value of one variable. As long as there exists a minimal model, we can compute a minimal model by solving m WMMSat instances.

Observe that in this case we do not make use of the additional requirement that we are looking for minimal models of bounded weight. However, in many contexts one is especially interested in small models (e.g. diagnosis). This is the reason why we have included the upper bound k on the weight in the WMMSat problem definition.

Furthermore, notice that if we set π to true, WMMSat is equivalent to WSat$_{\leq}$. This is because φ has a minimal model of weight $\leq k$ iff it has a model of weight $\leq k$.

One can assume that φ does not contain clauses of the form $\{v\}$ (facts) and $\{\neg v\}$ (anti-facts) since these already fix a truth value for each possible model. Additionally, we can assume that π does not contain anti-facts because this would also allow to fix the corresponding truth value to false. For facts in π this is not the case. Consider again the example $\varphi := (x \vee \neg x)$ and $\pi := x$. Fixing the truth value of x to true is not valid since φ has no *minimal* model containing x.

k	the maximum weight of the minimal model
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 (explained in the corresponding section)
p	the maximum number of positive occurrences of a variable in φ
v^+, v^-	the number of variables that occur as positive/negative literals in φ or in π
d^+_π	the maximum positive clause size in π
$\lVert\pi\rVert$	the length of π, i.e. the total number of variable occurrences in π

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

In this paper we perform an extensive parameterized complexity analysis of WMMSat. The parameters that are considered are listed in Table 1. All these parameters (except for b) can be computed in polynomial time. For every combination of these parameters we either show fixed-parameter tractability or present a hardness result.

A Parameterized Completeness Result for WMMSat Parameterized by k

The first theorem answers an open problem posed in (Gottlob, Scarcello, and Sideri 2002, at the end of Section 6.3): For which complexity class is WMMSat complete when parameterized by the weight? To answer this question we make use of a model-checking problem over $\Sigma_{t,u}$ formulas, a fragment of first-order formulas. The class $\Sigma_{t,u}$ contains all first-order formulas of the form $\exists \overline{x}_1 \forall \overline{x}_2 \exists \overline{x}_3 \ldots Q \overline{x}_t \psi(\overline{x}_1, \ldots, \overline{x}_t)$, where ψ is quantifier free and Q is an $\exists$ if t is odd and a $\forall$ if t is even. The length of the quantifier blocks – with the exception of the first $\exists$ block – are of length at most u.

MC[$\Sigma_{t,u}$]

Instance: A finite structure $\mathcal{A}$, a formula $\psi \in \Sigma_{t,u}$.

Question: Is $\mathcal{A}$ a model of ψ ?

When parameterized by the length of the formula ψ, the problem MC[$\Sigma_{2,u}$] is W[2]-complete for $u \geq 1$ (Downey, Fellows, and Regan 1998; Flum and Grohe 2005). In the following proof the encoding of Cnf formulas as logical structures builds upon ideas presented in (Flum and Grohe 2006, Chapter 7).

Theorem 1. WMMSat *parameterized by the maximal weight k is* W[2]*-complete.*

Proof. It is easy to see that WMMSat is W[2]-hard. Just observe that $(\varphi, \mathit{true}, k)$ is a Yes-instance iff φ has a model of weight $\leq k$. This problem is the W[2]-hard WSAT$_{\leq}$ problem as introduced in the preliminaries.

Membership is shown by a reduction of a WMMSAT instance (φ, π, k) to an MC[$\Sigma_{2,1}$] instance $(\mathcal{A}, \psi)$. Observe that if a clause c contains more than k negative literals, then c is satisfied by all weight k interpretations. Therefore, we will assume that such clauses were removed from φ and π in a preprocessing step. In addition, we check during the preprocessing whether the empty set is a model of φ. In this case, we reduce (φ, π, k) to a trivial Yes-instance if $\emptyset$ is also a model of π and to a trivial No-instance if $\emptyset$ is not a model of π. If the empty set is not a model of φ we continue as follows. The structure $\mathcal{A} = (A, E, ROOT_\varphi, ROOT_\pi, VAR, POS, N_0, \ldots, N_k)$ is obtained as follows. The domain A contains an element for each clause in φ and π, for each variable in φ and two root elements (denoted by $root_\varphi$ and $root_\pi$). Since $var(\pi) \subseteq var(\varphi)$, elements for all variables of π are contained in A. The relation E contains a pair $(root_\varphi, c)$ (resp. $(root_\pi, c)$) for each clause $c \in A$ that occurs in φ (resp. π). The relation POS contains a pair (c, p) for each variable p occurring positively in clause c. Furthermore, $ROOT_\varphi := \{root_\varphi\}$ and $ROOT_\pi := \{root_\pi\}$. The set VAR contains all variables of φ (and of π). For each clause $c = \{p_1, \ldots, p_j, \neg n_1, \ldots, \neg n_i\}$ of φ and π we create a tuple $(c, n_1, \ldots, n_i) \in N_i$. Notice that $0 \leq i \leq k$. It is also worth mentioning that although the formulas φ and π share the relations $E, VAR, POS, N_0, \ldots, N_k$, they can be still distinguished via the distinct root elements.

Let $X := \{x_1, \ldots, x_k\}$ be a set of first-order variable symbols and $\#(\cdot)$ an enumeration of all proper subsets of X, i.e. a bijection from $\mathbb{P}(X) \setminus \{X\}$ to $[2^k - 1]$. The formula ψ is defined as:

$$\psi := \exists x_1 \ldots \exists x_k \, \exists r_\varphi \, \exists r_\pi \, \exists c_1 \ldots \exists c_{2^k-1} \, \forall c \quad ROOT_\varphi(r_\varphi) \wedge ROOT_\pi(r_\pi) \wedge \bigwedge_{i \in [k]} VAR(x_i) \wedge \psi_{\mathrm{SAT}}(X, r_\varphi, c) \wedge \psi_{\mathrm{MIN}}(r_\varphi) \wedge \psi_{\mathrm{SAT}}(X, r_\pi, c)$$

$$\psi_{\mathrm{SAT}}(S, r, e) := \; E(r, e) \rightarrow \Big[\bigvee_{x \in S} POS(e, x) \vee \bigwedge_{0 \leq i \leq k} \bigwedge_{\overline{z} \in S^i} \neg N_i(e, \overline{z}) \Big]$$

$$\psi_{\mathrm{MIN}}(r) := \bigwedge_{X' \subset X} \neg \psi_{\mathrm{SAT}}(X', r, c_{\#(X')})$$

The variables $x_1, \ldots, x_k$ are used to select a model of weight $\leq k$, the variables $c, c_1, \ldots, c_{2^k-1}$ represent clauses and r_φ (resp. r_π) is the root node in the parse tree of φ (resp. π). The formula $\psi_{\mathrm{SAT}}(S, r, e)$ checks whether in the formula selected by root r, the clause e is satisfied by the interpretation S. Therefore $\forall c \; \psi_{\mathrm{SAT}}(X, r, c)$ checks whether X is a model of the formula selected by r. The formula $\psi_{\mathrm{MIN}}(r)$ then checks whether X is a minimal model (of the formula selected by root r) by demanding for each $X' \subset X$ a clause (namely $c_{\#(X')}$) that is not satisfied by S. This guarantees that each interpretation $X' \subset X$ violates at least one clause. Notice that while we require that both φ and π are satisfied by the interpretation $\{x_1, \ldots, x_k\}$, minimality has to be checked only for φ. Finally, it can easily be verified that the length of ψ can be bounded in terms of k. □

Algorithm 1: *branch*$(\varphi, k, \mathcal{S})$

```
$\mathcal{C} \leftarrow \emptyset$
if $\emptyset \in \varphi[\mathcal{S}]$ then
    return $\emptyset$    // $\varphi[\mathcal{S}]$ is not satisfiable
if every clause in $\varphi[\mathcal{S}]$ contains a negative literal then
    return $\{\mathcal{S}\}$    // $\emptyset$ is the only minimal model of $\varphi[\mathcal{S}]$
else
    if $|\mathcal{S}| \geq k$ then
        return $\emptyset$    // No model containing $\mathcal{S}$ with weight $\leq k$
    Pick a clause $c \in \varphi[\mathcal{S}]$ containing only positive literals.
    foreach variable $x \in c$ do
        $\mathcal{C} \leftarrow \mathcal{C} \cup branch(\varphi, k, \mathcal{S} \cup \{x\})$
    return $\mathcal{C}$
```

An Fpt Algorithm for Small Clause Size

In the previous section we have shown that WMMSAT is W[2]-complete when parameterized by the weight k. In this section we show that this hardness result does not hold if we bound the clause size of φ. More specifically we show that WMMSAT is fpt with respect to the weight k and the clause size d. Actually we can even show fixed-parameter tractability with respect to k and the positive clause size d^+.

The corresponding fpt algorithm relies heavily on the recursive procedure *branch*$(\varphi, k, \mathcal{S})$. This procedure computes a set of models that satisfy φ, have weight $\leq k$ and are supersets of $\mathcal{S}$. In particular, this set contains all minimal models of φ of weight $\leq k$ that are supersets of $\mathcal{S}$. However, the set computed by *branch* also contains models that are not minimal. See Algorithm 1 for details.

Example 1. In Figure 1 we illustrate *branch* with the arguments $\varphi = (a \vee b) \wedge (a \vee c) \wedge (\neg b \vee d) \wedge (\neg a \vee \neg c \vee \neg d)$, $k = 3$ and $\mathcal{S} = \emptyset$. The clause chosen to be branched on is always written in bold. We see that *branch*$(\varphi, 3, \emptyset)$ returns three models: $\{a\}$ and $\{b, c, d\}$ are minimal; $\{a, b, d\}$ is not minimal.

As mentioned before, we can prove that *branch* finds all minimal models:

Lemma 2. *The procedure branch$(\varphi, k, \emptyset)$ returns a set of models of φ having weight $\leq k$. In particular, all subset minimal models of φ having weight $\leq k$ are contained.*

Proof. Let $\mathcal{M}$ be a minimal model of φ. The main observation is that in each branching step of the *branch* procedure we can choose a variable from $\mathcal{M}$. Since we branch on positive clauses and $\mathcal{M}$ is a model, one variable in each of these clauses has to be an element of $\mathcal{M}$. Also every element of $\mathcal{M}$

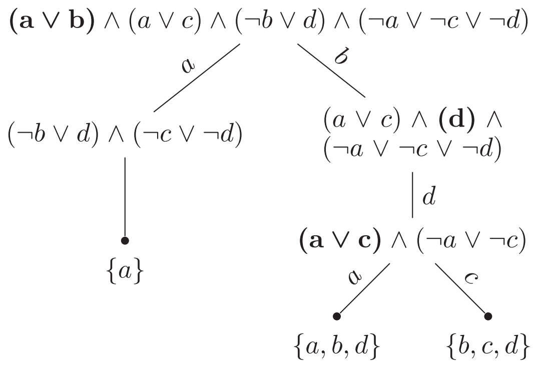
$(\mathbf{a} \lor \mathbf{b}) \land (a \lor c) \land (\neg b \lor d) \land (\neg a \lor \neg c \lor \neg d)$
a
b
$(\neg b \lor d) \land (\neg c \lor \neg d)$
$(a \lor c) \land (\mathbf{d}) \land (\neg a \lor \neg c \lor \neg d)$
d
$\{a\}$
$(\mathbf{a} \lor \mathbf{c}) \land (\neg a \lor \neg c)$
a
c
$\{a, b, d\}$
$\{b, c, d\}$

Figure 1: Example of the *branch* procedure.

has to appear in a clause on which we branch because otherwise $\mathcal{M}$ would not be minimal. Therefore $\mathcal{M}$ is returned by $branch(\varphi, k, \emptyset)$. □

We now describe the first fpt algorithm in this paper.

Theorem 3. WMMSAT *can be solved in time* $\mathcal{O}\big((d^+)^k \cdot 2^k \cdot n\big)$.

Proof. The algorithm works as follows: First, the procedure $branch(\varphi, k, \emptyset)$ computes a set of models $\mathcal{C}$ of weight $\leq k$. By Lemma 2, $\mathcal{C}$ contains all minimal models of φ and some models that are not minimal. The algorithm now considers every model $\mathcal{M} \in \mathcal{C}$ that is also a model for π. It remains to check whether it is a *subset minimal* model for φ. This can be done by testing whether any proper subset of $\mathcal{M}$ is a model of φ as well. If not, (φ, π, k) is a Yes-instance. If $\mathcal{M}$ is not minimal, we continue with the next $\mathcal{M} \in \mathcal{C}$.

In order to determine the runtime, observe that *branch* has a branching factor of at most d^+ and the recursion depth is at most k. In case a model for φ and π is found, we have to check for at most $2^k - 1$ subsets whether they are models of φ. In total we obtain a runtime of $\mathcal{O}\big((d^+)^k \cdot 2^k \cdot n\big)$. □

Remark. Since $d^+ \leq d$, the runtime bound holds for the parameters d and k as well.

Fpt Algorithms for (Almost) Monotone Formulas

This section contains an fpt algorithm that performs well on almost monotone formulas with a bounded number of non-Horn clauses. The algorithm itself is rather intricate. We therefore do not give a detailed "low level" run time analysis. This means we ignore polynomial factors by using the $\mathcal{O}^*(\cdot)$ notation. $\mathcal{O}^*(\cdot)$ is defined in the same way as $\mathcal{O}(\cdot)$ but ignores polynomial factors. We start by introducing the central notion of this section: *succinct sets of models*.

Succinct Sets of Models

Consider the monotone formula $\psi = (a \lor b) \land (b \lor c) \land (d \lor e)$. This formula has four minimal models: $\{a, c, d\}$, $\{a, c, e\}$, $\{b, d\}$ and $\{b, e\}$. Observe that the decision whether to include d or e in the minimal model is independent of the choice which of a, b, c to include. Actually we can group these models in two sets: those that include a, c and those that include b. These two sets of models can be denoted by $\big\{\{a\}, \{c\}, \{d, e\}\big\}$ and $\big\{\{b\}, \{d, e\}\big\}$. Variables that occur together in the same set are equivalent in the sense that they satisfy exactly the same clauses. These observations already hint at the merit of such *succinct sets of models*: the elimination of redundancies. If one is just interested in the satisfiability of ψ, the distinction between d and e does not make much sense since they appear in exactly the same clauses. This distinction is however crucial for solving WMMSAT since for example d and e might appear in different clauses in π. These observations give rise to the following definition.

In this subsection ψ is always a monotone formula. While a model of ψ is a subset of $var(\psi)$, a *succinct set of models* (SSM) is a subset of $\mathbb{P}(var(\psi))$, i.e. it contains sets of variables.

Definition. Let $\mathcal{S} \subseteq \mathbb{P}(var(\psi))$. The set $\mathcal{S}$ is an SSM of ψ if the conditions **S0**, **S1**, **S2** and **S3** hold.

S0 For all distinct sets $s_1, s_2 \in \mathcal{S}$, $s_1 \cap s_2 = \emptyset$.

S1 For all $x, y \in var(\psi)$ it has to hold that $\forall c \in \psi\, (x \in c \leftrightarrow y \in c)$ iff $\forall s \in \mathcal{S}\, (x \in s \leftrightarrow y \in s)$.

Whenever variables appear in exactly the same clauses, they have to be together in an element of an SSM. Therefore every element in an SSM is in this sense maximal.

S2 For each clause $c \in \psi$ there is a set $s \in \mathcal{S}$ such that $c \cap s \neq \emptyset$.

This property requires an SSM to cover all clauses in ψ.

S3 There is no $s \in \mathcal{S}$ such that $\mathcal{S} \setminus \{s\}$ satisfies **S2**.

This forces an SSM to be minimal. Note that \ is the usual set difference operator.

Example 2. We consider again the formula $\psi = (a \lor b) \land (b \lor c) \land (d \lor e)$. The set $\big\{\{b\}, \{d\}\big\}$ is not an SSM because of **S1** (d appears in each clause e appears in and therefore they have to be contained in the same set in an SSM). The set $\big\{\{b\}\big\}$ is also not an SSM since it violates **S2** (the clause $(d \lor e)$ is not covered). **S3** is e.g. not fulfilled by $\big\{\{a\}, \{b\}, \{d, e\}\big\}$ (the subset $\big\{\{b\}, \{d, e\}\big\}$ already is an SSM and hence satisfies **S2**). Also the set $\big\{\{a\}, \{c\}, \{d\}, \{e\}\big\}$ violates **S3** since the subset $\big\{\{a\}, \{c\}, \{d\}\big\}$ satisfies **S2**.

Intuitively, a succinct set of models is – in contrast to a model – a set of possible choices for variables instead of a set of variables. This is reflected by the following definition.

Definition. Let $\mathcal{M} \subseteq var(\psi)$ and $\mathcal{S} \subseteq \mathbb{P}(var(\psi))$ (e.g. $\mathcal{S}$ is an SSM of ψ). Then $\mathcal{M}$ *is in* $\mathcal{S}$ (denoted by $\mathcal{M} \tilde{\in} \mathcal{S}$) if there exists a bijection $\gamma : \mathcal{M} \to \mathcal{S}$ such that $x \in \gamma(x)$ for every $x \in \mathcal{M}$.

Let $\mathcal{S} = \{s_1, \ldots, s_l\}$ be an SSM. Observe that this definition implies that $\forall x_1 \in s_1\, \forall x_2 \in s_2 \ldots \forall x_l \in s_l$ the set $\{x_1, \ldots, x_l\} \tilde{\in} \mathcal{S}$. The following two lemmas point out the close connection between SSMs and minimal models.

Lemma 4. *Let $\mathcal{M}$ be a minimal model of ψ. Then there exists an SSM $\mathcal{S}$ such that $\mathcal{M} \tilde{\in} \mathcal{S}$.*

Proof. We can construct an SSM in the following way:

$\mathcal{S} := \big\{\{y \in \mathit{var}(\psi) \mid \forall c \in \psi (y \in c \leftrightarrow x \in c)\} \mid x \in \mathcal{M}\big\}$.

It can easily be checked that **S0**, **S1**, **S2** and **S3** hold. □

Lemma 5. *Let $\mathcal{S}$ be an SSM of ψ. If $\mathcal{M} \tilde{\in} \mathcal{S}$, then $\mathcal{M}$ is a minimal model of ψ.*

Proof. Let c be a clause of ψ. We are going to show that $c \cap \mathcal{M} \neq \emptyset$. From **S2** we know that there is a set $s \in \mathcal{S}$ such that $c \cap s \neq \emptyset$. We then conclude from **S0** that for all $x, y \in c \cap s$ it holds $\forall t \in \mathcal{S} (x \in t \leftrightarrow y \in t)$. Then **S1** implies that $s \subseteq c$. Finally $\mathcal{M} \tilde{\in} \mathcal{S}$ implies that there is an $x \in \mathcal{M}$ with $x \in s$ and therefore $\mathcal{M} \cap c \neq \emptyset$. This proves that $\mathcal{M}$ is a model of ψ.

$\mathcal{M}$ is also a minimal model. Towards a contradiction assume that there is a model $\mathcal{M}' \subset \mathcal{M}$. Let $x \in \mathcal{M} \setminus \mathcal{M}'$. Then there is an $s \in \mathcal{S}$ with $x \in s$. Since ψ is monotone, $\mathcal{M}'$ is a model and $\mathcal{M}' \subseteq \mathcal{M} \setminus \{x\}$, we know that $\mathcal{M} \setminus \{x\}$ is a model. Therefore for each clause $c \in \psi$ it holds that $(\mathcal{M} \setminus \{x\}) \cap c \neq \emptyset$. From **S0** it follows that x is not contained in $t \in \mathcal{S}$, $t \neq s$, i.e. x is only contained in s. Since $\tilde{\in}$ is defined by the bijection γ, x is the only element in $\gamma(x) = s$ that is contained in $\mathcal{M}$. By that we know that $(\mathcal{M} \setminus \{x\}) \tilde{\in} (\mathcal{S} \setminus \{s\})$. Hence it also holds for every $c \in \psi$ that there exists a $t \in (\mathcal{S} \setminus \{s\})$ such that $t \cap c \neq \emptyset$ (since $t \cap c$ contains an element of $\mathcal{M} \setminus \{x\}$). This proves that $\mathcal{S} \setminus \{s\}$ satisfies **S2** – which contradicts **S3**. □

The following lemma proves an upper bound on the number of SSMs that plays a crucial role in the runtime estimates later on.

Lemma 6. *Let ψ be a monotone* CNF *formula and let $|\psi|$ denote the number of its clauses. Furthermore, let $\mathcal{C}$ denote the set of all SSMs of ψ with cardinality $\leq k$. Then $|\mathcal{C}| \leq 2^{k \cdot |\psi|}$.*

Proof. We show this bound with the help of an injective function μ from $\mathcal{C}$ to a set whose size can easier be bounded. Let μ be a function from $\mathcal{C}$ into

$$\big\{\{q_1, \dots, q_k\} \mid q_i \subseteq \psi \text{ for } i \in [k]\big\}.$$

Notice that $q_1, \dots, q_k$ are sets of clauses and need not to be distinct. The formula ψ has $2^{|\psi|}$ subsets (of clauses) and hence this set has a cardinality of less than $2^{k \cdot |\psi|}$.

In order to describe the injective function μ let $\mathcal{S}$ be an SSM of ψ. Recall that we can consider ψ as a set containing clauses and clauses can be considered as sets of variables. We define $\mu(\mathcal{S})$ as the set $\big\{\{c \in \psi \mid s \subseteq c\} \mid s \in \mathcal{S}\big\}$. Since $|\mathcal{S}| \leq k$ the cardinality of $\mu(\mathcal{S})$ is also $\leq k$.

It remains to prove that μ is injective. Assume that $\mu(\mathcal{S}) = \mu(\mathcal{T})$. We are going to show that this implies that $\mathcal{S} = \mathcal{T}$. Let $s \in \mathcal{S}$. Then there is a corresponding set $\{c \in \psi \mid s \subseteq c\}$ in $\mu(\mathcal{S})$. Since $\mu(\mathcal{S}) = \mu(\mathcal{T})$ there is a $t \in \mathcal{T}$ such that $\{c \in \psi \mid s \subseteq c\} = \{c \in \psi \mid t \subseteq c\} \in \mu(\mathcal{T})$. It therefore has to hold for all $c \in \psi$ that $s \subseteq c \leftrightarrow t \subseteq c$. Let $x \in s$ and $y \in t$. From **S0** and **S1** it follows that for all $c \in \psi$ that $x \in c \leftrightarrow s \subseteq c$ and also $y \in c \leftrightarrow t \subseteq c$. Hence for all $c \in \psi$, $x \in c \leftrightarrow y \in c$. Now **S1** implies that $y \in s$ and $x \in t$. Consequently $s \cap t \neq \emptyset$ and hence by **S0** $s = t$. Therefore $\mathcal{S} = \mathcal{T}$ and by that injectivity of μ is shown. □

Remark. Note that due to the simple nature of this injective function μ one can easily see that all SSMs of ψ with cardinality $\leq k$ can be generated in time $\mathcal{O}^*\big(2^{k \cdot |\psi|}\big)$.

We end this subsection by introducing the operator $\tilde{\cap}$.

Definition. Let $\mathcal{S}$ be an SSM of ψ and $\mathcal{I} \subseteq \mathit{var}(\psi)$. $\mathcal{S} \,\tilde{\cap}\, \mathcal{I}$ is defined the following way:

- If there is an $x \in \mathcal{I}$ such that $x \notin s$ for all $s \in \mathcal{S}$ then $\mathcal{S} \,\tilde{\cap}\, \mathcal{I} := \{\emptyset\}$.
- If there is an $s \in \mathcal{S}$ such that $|s \cap \mathcal{I}| \geq 2$ then $\mathcal{S} \tilde{\cap} \mathcal{I} := \{\emptyset\}$.
- Otherwise $\mathcal{S} \,\tilde{\cap}\, \mathcal{I} := \{s \cap \mathcal{I} \mid s \in \mathcal{S} \wedge s \cap \mathcal{I} \neq \emptyset\} \cup \{s \mid s \in \mathcal{S} \wedge s \cap \mathcal{I} = \emptyset\}$.

The intuition behind this definition is that $\mathcal{S} \,\tilde{\cap}\, \mathcal{I}$ contains (with respect to $\tilde{\in}$) all minimal models contained in $\mathcal{S}$ that are supersets of $\mathcal{I}$. Therefore the two exceptions are required in this definition: If the first property holds then no $\mathcal{M} \tilde{\in} \mathcal{S}$ contains x. If the second property holds then no $\mathcal{M} \tilde{\in} \mathcal{S}$ can contain all variables in $s \cap \mathcal{I}$. In these cases the result of $\mathcal{S} \tilde{\cap} \mathcal{I}$ is $\{\emptyset\}$ because there is no $\mathcal{M} \tilde{\in} \{\emptyset\}$.

Observe that $\mathcal{S} \,\tilde{\cap}\, \mathcal{I}$ is not necessarily an SSM since it might be that **S1** is no longer fulfilled. However, the following lemma holds.

Lemma 7. *Let $\mathcal{S}$ be an SSM of ψ and $\mathcal{I} \subseteq \mathit{var}(\psi)$. Then $\mathcal{M} \tilde{\in} (\mathcal{S} \,\tilde{\cap}\, \mathcal{I})$ iff $\mathcal{M} \tilde{\in} \mathcal{S}$ and $\mathcal{I} \subseteq \mathcal{M}$.*

This lemma is immediate consequence of the definitions of $\tilde{\cap}$ and $\tilde{\in}$.

The Algorithm for Monotone Formulas

Before we present the full algorithm in the next subsection, we first explain the special case where the formulas φ and π are monotone. Algorithm 2 provides an overview of the procedure.

Algorithm 2: Fpt-algorithm for monotone formulas – Theorem 8

```
1 Let $\mathcal{N}$ be the set of all variables that occur as facts in $\varphi$.
2 Let $l := \min\{k - |\mathcal{N}|, |\varphi[\mathcal{N}]|\}$.
3 $\mathcal{C}_\pi \leftarrow branch(\pi[\mathcal{N}], l, \emptyset)$
4 foreach $\mathcal{M}_\pi \in \mathcal{C}_\pi$ do
5     foreach SSM $\mathcal{S}$ of $\varphi[\mathcal{N}]$ with cardinality $\leq l$ do
6         if $\mathcal{S} \,\tilde{\cap}\, \mathcal{M}_\pi \neq \{\emptyset\}$ then return Yes
7 return No
```

If variables occur as facts, i.e. clauses of size 1, in φ they have to be a part of every model. Therefore we only consider $\varphi[\mathcal{N}]$ and $\pi[\mathcal{N}]$, where $\mathcal{N}$ denotes the set of all facts in φ. The constant l is a bound on the maximum possible size of a minimal model of $\varphi[\mathcal{N}]$. On the one hand $l \leq k - |\mathcal{N}|$ since we are only interested in models of weight $\leq k$. On the other hand a minimal model of the formula φ can have weight at most $|\varphi|$ and hence $l \leq |\varphi[\mathcal{N}]|$. In Line 3 we employ the *branch* procedure as described earlier to compute a set containing all minimal models of $\pi[\mathcal{N}]$ of weight $\leq l$. Recall that this set might also contain non-minimal models. Then we check for each $\mathcal{M}_\pi \in \mathcal{C}$ and for each SSM $\mathcal{S}$ of $\varphi[\mathcal{N}]$

Algorithm 3: Fpt-algorithm for arbitrary CNF formulas – Theorem 9 and 10

```
1  Let $V^-$ denote all variables that occur as negative literals in $\varphi$ or in $\pi$.
2  foreach assignment $(\mathcal{T}, \mathcal{F})$ of $V^-$ with weight $|\mathcal{T}| \leq k$ do
3      Let $\mathcal{N}$ be the set of all variables that occur as facts in $\varphi[\mathcal{T}, \mathcal{F}]$.
4      Let $l := \min(k - |\mathcal{T}| - |\mathcal{N}|, |\varphi[\mathcal{T} \cup \mathcal{N}, \mathcal{F}]|)$.        // The "remaining" weight.
5      $\mathcal{C}_\pi \leftarrow branch(\pi[\mathcal{T} \cup \mathcal{N}, \mathcal{F}], l, \emptyset)$
6      foreach $\mathcal{M}_\pi \in \mathcal{C}_\pi$ do
7          foreach SSM $\mathcal{S}$ of $\varphi[\mathcal{T} \cup \mathcal{N}, \mathcal{F}]$ with cardinality $\leq l$ do
               // For each $\mathcal{M} \tilde{\in} (\mathcal{S} \tilde{\cap} \mathcal{M}_\pi)$, $\mathcal{M} \cup \mathcal{T} \cup \mathcal{N}$ is a model of $\varphi \wedge \pi$. We have to check
                  whether there is a $\mathcal{M} \tilde{\in} (\mathcal{S} \tilde{\cap} \mathcal{M}_\pi)$ such that $\mathcal{M} \cup \mathcal{T} \cup \mathcal{N}$ is minimal.
8              Let $\mathcal{L}$ be the set of all $(\mathcal{T}', \mathcal{F}') \prec (\mathcal{T}, \mathcal{F})$ where $\varphi[\mathcal{T}' \cup \mathcal{N}, \mathcal{F}']$ contains at most $(k - |\mathcal{T}| - |\mathcal{N}|) \cdot p$ clauses.
9              foreach function $\beta$ with domain $\mathcal{L}$ and $\beta(\mathcal{T}', \mathcal{F}') \in \varphi[\mathcal{T}' \cup \mathcal{N}, \mathcal{F}']$ do    // Choose one clause per
               subassignment. Is there an $\mathcal{M} \tilde{\in} (\mathcal{S} \tilde{\cap} \mathcal{M}_\pi)$ that is not a model for these
               clauses?
10                 Let $var(\beta)$ be the set of all variables that appear in the range of $\beta$.
11                 if for all $s \in (\mathcal{S} \tilde{\cap} \mathcal{M}_\pi)$, $s \setminus var(\beta) \neq \emptyset$ then return Yes
               // There is no $\mathcal{M} \tilde{\in} (\mathcal{S} \tilde{\cap} \mathcal{M}_\pi)$ such that $\mathcal{M} \cup \mathcal{T} \cup \mathcal{N}$ is a minimal model of $\varphi$.
           // There is no minimal model of $\varphi$ extending $(\mathcal{T} \cup \mathcal{M}_\pi, \mathcal{F})$.
       // There is no minimal model of $\varphi$ extending $(\mathcal{T}, \mathcal{F})$ that also satisfies $\pi$.
12 return No
```

with cardinality $\leq l$ whether there is a $\mathcal{M} \tilde{\in} \mathcal{S}$ with $\mathcal{M}_\pi \subseteq \mathcal{M}$. Lemma 7 implies that there is a $\mathcal{M} \tilde{\in} \mathcal{S}$ with $\mathcal{M}_\pi \subseteq \mathcal{M}$ if $\mathcal{M} \tilde{\in} (\mathcal{S} \tilde{\cap} \mathcal{M}_\pi)$. This is exactly the case if $\mathcal{S} \tilde{\cap} \mathcal{M}_\pi \neq \{\emptyset\}$. Furthermore, by Lemma 4 we know that every minimal model is contained in an SSM. Hence checking this property decides the WMMSAT instance.

Example 3. Let $\varphi := (a \vee b \vee c) \wedge (d \vee e \vee h \vee i) \wedge (f \vee h \vee i)$, $\pi := (a \wedge f)$ and $k := 3$. We now demonstrate how Algorithm 2 works. First $\mathcal{N} = \emptyset$ since φ does not contain facts. The formula π has only one model: $\{a, f\}$. There are two SSMs of φ: $\mathcal{S}_1 := \big\{\{a, b, c\}, \{d, e\}, \{f\}\big\}$ and $\mathcal{S}_2 := \big\{\{a, b, c\}, \{h, i\}\big\}$. We obtain $\mathcal{S}_1 \tilde{\cap} \mathcal{M}_\pi = \big\{\{a\}, \{d, e\}, \{f\}\big\}$ and $\mathcal{S}_2 \tilde{\cap} \mathcal{M}_\pi = \{\emptyset\}$ (since there is no $s \in \mathcal{S}$ with $f \in s$). Therefore only $\mathcal{S}_1$ contains models that are supersets of $\{a, f\}$, namely $\{a, d, f\}$ and $\{a, e, f\}$. Consequently this is a Yes-instance.

Theorem 8. WMMSAT *for monotone φ and π can be solved in time*

$$\mathcal{O}^*((d_\pi^+)^k \cdot 2^{k^2 \cdot p}) \quad \textit{or} \tag{1}$$

$$\mathcal{O}^*((d_\pi^+)^h \cdot 2^{h^2}). \tag{2}$$

Proof. Recall that the procedure *branch* considers at most $(d_\pi^+)^k$ models. From Lemma 6 we know that there are at most $2^{k \cdot |\varphi[\mathcal{N}]|}$ SSMs of $\varphi[\mathcal{N}]$. Observe that computing all these SSMs can be done in time $\mathcal{O}^*(2^{k \cdot |\varphi[\mathcal{N}]|})$. This yields a runtime of $\mathcal{O}^*\left((d_\pi^+)^k \cdot 2^{k \cdot |\varphi[\mathcal{N}]|}\right)$ for Algorithm 2.

Regarding Equation (1) observe that if the number of clauses is larger than $k \cdot p$, φ is not satisfiable by a weight k interpretation. Hence we can assume that $|\varphi[\mathcal{N}]| \leq k \cdot p$ which yields Equation (1).

When parameterizing by h we have no immediate bound on the maximum weight k. However, we know that the monotone formula $\varphi[\mathcal{N}]$ has at most h clauses since it contains no facts. Hence, a minimal model of $\varphi[\mathcal{N}]$ has weight at most h. This yields Equation (2). □

The Algorithm for Arbitrary Formulas

Algorithm 3 extends the algorithm for monotone formulas (Algorithm 2) to arbitrary formulas. This is possible due to the parameter v^-. This parameter allows us to perform a brute-force search over all assignments to variables that occur negatively. Clearly, all remaining variables occur only positively. Finding a model is straight-forward with this parameter whereas checking for minimality is more involved.

In a first step (Line 2) we pick an assignment $(\mathcal{T}, \mathcal{F})$ of V^-, the set of all variables that appear as negative literals either in φ or π. The reduced formulas $\varphi[\mathcal{T}, \mathcal{F}]$ and $\pi[\mathcal{T}, \mathcal{F}]$ are monotone and therefore we can apply the strategy of Algorithm 2. Let $\mathcal{N}$ denote the set of all facts in $\varphi[\mathcal{T}, \mathcal{F}]$. We iterate over a set containing all minimal models $\mathcal{M}_\pi$ of $\pi[\mathcal{T} \cup \mathcal{N}, \mathcal{F}]$ (generated by the *branch* procedure) and over all SSMs $\mathcal{S}$ of $\varphi[\mathcal{T} \cup \mathcal{N}, \mathcal{F}]$. From Lemma 4 it follows that for every minimal model of $\varphi[\mathcal{T} \cup \mathcal{N}, \mathcal{F}]$, $\mathcal{M} \tilde{\in} \mathcal{S}$. Therefore this algorithm indeed considers all candidates of appropriate minimal models. By Lemma 5 and Lemma 7 it holds for each $\mathcal{M} \tilde{\in} (\mathcal{S} \tilde{\cap} \mathcal{M}_\pi)$ that $\mathcal{M} \cup \mathcal{N}$ is a minimal model of $\varphi[\mathcal{T}, \mathcal{F}]$ and a model of $\pi[\mathcal{T}, \mathcal{F}]$.

It remains to check whether there is an $\mathcal{M} \tilde{\in} (\mathcal{S} \tilde{\cap} \mathcal{M}_\pi)$ such that $\mathcal{M} \cup \mathcal{N} \cup \mathcal{T}$ is minimal with respect to φ. For this we consider all subassignments $(\mathcal{T}', \mathcal{F}') \prec (\mathcal{T}, \mathcal{F})$ (Line 8). For each $(\mathcal{T}', \mathcal{F}')$ we consider $\varphi[\mathcal{T}' \cup \mathcal{N}, \mathcal{F}']$. Observe that since $\varphi[\mathcal{T}' \cup \mathcal{N}, \mathcal{F}']$ is monotone, $\varphi[\mathcal{T}' \cup \mathcal{N}, \mathcal{F}']$

is satisfiable iff $\varphi[\mathcal{T}', \mathcal{F}']$ is satisfiable. Furthermore, since $\mathcal{T}' \subset \mathcal{T}$, if there exists a model $\mathcal{M}'$ of $\varphi[\mathcal{T}', \mathcal{F}']$ with $\mathcal{M}' \cup \mathcal{T}' \subset \mathcal{M} \cup \mathcal{T} \cup \mathcal{N}$ then there also exists a model $\mathcal{M}''$ of $\varphi[\mathcal{T}' \cup \mathcal{N}, \mathcal{F}']$ with $\mathcal{M}'' \cup \mathcal{T}' \cup \mathcal{N} \subset \mathcal{M} \cup \mathcal{T} \cup \mathcal{N}$. As observed in Algorithm 2, if $\varphi[\mathcal{T}' \cup \mathcal{N}, \mathcal{F}']$ has more than $k{\cdot}p$ clauses it cannot be satisfied by a weight k interpretation. Therefore a subassignment $(\mathcal{T}', \mathcal{F}')$ where $\varphi[\mathcal{T}' \cup \mathcal{N}, \mathcal{F}']$ is "too large" cannot have a model that is a counterexample to the minimality property.

The question is now: Is there an $\mathcal{M} \tilde{\in} (\mathcal{S} \tilde{\cap} \mathcal{M}_\pi)$ such that for each subassignment $(\mathcal{T}', \mathcal{F}') \prec (\mathcal{T}, \mathcal{F})$, $\varphi[\mathcal{T}' \cup \mathcal{N}, \mathcal{F}']$ contains a clause that is not satisfied by $\mathcal{M}$. We check that in Line 9 by choosing one clause per subassignment. If $\mathcal{S} \tilde{\cap} \mathcal{M}_\pi$ still contains a model after removing the variables in these clauses, we have a Yes-instance. This is because for every remaining model $\mathcal{M}$ in $\mathcal{S} \tilde{\cap} \mathcal{M}_\pi$, $\mathcal{M} \cup \mathcal{T} \cup \mathcal{N}$ is a minimal model of φ and a model of π. Also $|\mathcal{M} \cup \mathcal{T} \cup \mathcal{N}| \leq k$.

Example 4. Let $\varphi := (\neg x \vee a \vee b \vee c) \wedge (\neg x \vee d \vee e \vee h \vee i) \wedge (\neg x \vee f \vee h \vee i) \wedge (x \vee a) \wedge (x \vee d \vee e \vee h)$, $\pi := (a \wedge f)$ and $k := 3$. The set $V^- = \{x\}$. Assume that we are currently considering the assignment $(\{x\}, \emptyset)$ (Line 2). There are no facts in $\varphi[\mathcal{T}, \mathcal{F}]$ hence $\mathcal{N} = \emptyset$. The only (minimal) model of $\pi[\mathcal{T} \cup \mathcal{N}, \mathcal{F}]$ is $\mathcal{M}_\pi = \{a, f\}$. Observe that $\varphi[\mathcal{T} \cup \mathcal{N}, \mathcal{F}] = (a \vee b \vee c) \wedge (d \vee e \vee h \vee i) \wedge (f \vee h \vee i)$ is equal to formula φ in Example 3. Hence $\varphi[\mathcal{T} \cup \mathcal{N}, \mathcal{F}]$ has two SSMs: $\mathcal{S}_1 := \big\{\{a, b, c\}, \{d, e\}, \{f\}\big\}$ and $\mathcal{S}_2 := \big\{\{a, b, c\}, \{h, i\}\big\}$. As before we obtain $\mathcal{S}_1 \tilde{\cap} \mathcal{M}_\pi = \big\{\{a\}, \{d, e\}, \{f\}\big\}$ and $\mathcal{S}_2 \tilde{\cap} \mathcal{M}_\pi = \big\{\emptyset\big\}$. There is only one subassignment: $(\emptyset, \{x\})$. Since $\varphi[\emptyset, \{x\}] = a \wedge (d \vee e \vee h)$, there are two possible functions β: $\beta_1(\emptyset, \{x\}) = \{a\}$ and $\beta_2(\emptyset, \{x\}) = \{d, e, h\}$. Observe that for no pairing of $\mathcal{S}_1, \mathcal{S}_2$ with β_1, β_2 the condition in Line 11 is fulfilled. We therefore continue with the assignment $(\emptyset, \{x\})$ in Line 2. Here $\mathcal{N} = \{a\}$ and $\varphi[\mathcal{T} \cup \mathcal{N}, \mathcal{F}] = (d \vee e \vee h)$. This yields one SSM $\mathcal{S}_3 = \big\{\{d, e, h\}\big\}$. We obtain $\mathcal{S}_3 \tilde{\cap} \mathcal{M}_\pi = \big\{\emptyset\big\}$. Hence the condition in Line 11 cannot be fulfilled and $(\varphi, \pi, 3)$ is a No-instance.

Theorem 9. WMMSAT *can be solved in*

$$\mathcal{O}^* \left(2^{v^-} \cdot 2^{k^2 \cdot p} \cdot (d_\pi^+)^k \cdot (1.19 \cdot p)^{2^k}\right).$$

Proof. There are $\binom{|V^-|}{k} = \binom{v^-}{k} \leq 2^{v^-}$ assignments of V^- with weight $\leq k$. From Lemma 6 we know that there at most $2^{k \cdot |\varphi[\mathcal{T} \cup \mathcal{N}, \mathcal{F}]|}$ SSMs of $\varphi[\mathcal{T} \cup \mathcal{N}, \mathcal{F}]$ we have to consider. As in the proof of Theorem 8 this yields an upper bound of $2^{k^2 \cdot p}$ in our case. The *branch* procedure applied to $\pi[\mathcal{T} \cup \mathcal{N}, \mathcal{F}]$ (Line 5) generates at most $(d_\pi^+)^k$ models.

We now bound the number of functions described in Line 9. The domain has size at most $2^{|\mathcal{T}|} - 1$. For each considered $(\mathcal{T}', \mathcal{F}') \prec (\mathcal{T}, \mathcal{F})$ (Line 8), the range of $\beta(\mathcal{T}', \mathcal{F}')$ is at most $(k - |\mathcal{T}| - |\mathcal{N}|) \cdot p$. An assignment $(\mathcal{T}', \mathcal{F}') \prec (\mathcal{T}, \mathcal{F})$ where $\varphi[\mathcal{T}' \cup \mathcal{N}, \mathcal{F}']$ has more than $(k - |\mathcal{T}| - |\mathcal{N}|) \cdot p$ clauses can never yield a model of φ with weight $\leq k$. In total this yields an upper bound of $((k - |\mathcal{T}|) \cdot p)^{2^{|\mathcal{T}|}}$. One can show that $(k - |\mathcal{T}|)^{2^{|\mathcal{T}|}} \leq 1.19^{2^k}$. Since $|\mathcal{T}| \leq k$ we obtain the upper bound $(1.19 \cdot p)^{2^k}$ on the number of functions. Observe that these functions can also be generated in time $\mathcal{O}^*\big((1.19 {\cdot} p)^{2^k}\big)$. These bounds taken together yield the theorem. □

We can also achieve a different upper bound for Algorithm 3:

Theorem 10. WMMSAT *can be solved in*

$$\mathcal{O}^* \left(2^{v^-} \cdot 2^{h^2} \cdot (d_\pi^+)^h \cdot h^{2^{v^-}}\right).$$

Proof. The proof is similar to the one of Theorem 9. There are $(2^{v^-}$ assignments of V^-. Since $\varphi[\mathcal{T} \cup \mathcal{N}, \mathcal{F}]$ is monotone and contains no facts, it contains at most h clauses. Hence $l \leq h$. In Line 5 at most $(d_\pi^+)^h$ models are generated by the *branch* procedure. From Lemma 6 follows that there are at most $\mathcal{O}(2^{h^2})$ SSMs of $\varphi[\mathcal{T} \cup \mathcal{N}, \mathcal{F}]$ to be considered.

We now bound the number of functions described in Line 9. The domain has size at most $2^{v^-} - 1$. For each $(\mathcal{T}', \mathcal{F}') \prec (\mathcal{T}, \mathcal{F})$, the range of $\beta(\mathcal{T}', \mathcal{F}')$ has cardinality at most h. This is because every clause in $\varphi[\mathcal{T}' \cup \mathcal{N}, \mathcal{F}']$ either is a non-Horn clause or is a fact that originates from a non-Horn clause in φ. If $\varphi[\mathcal{T}' \cup \mathcal{N}, \mathcal{F}']$ contained a fact that originates from a Horn clause in φ, this would be a fact in $\varphi[\mathcal{T} \cup \mathcal{N}, \mathcal{F}]$ as well. This would be a contradiction since $\varphi[\mathcal{T} \cup \mathcal{N}, \mathcal{F}]$ does not contain facts. In total this yields an upper bound of $h^{2^{v^-}}$ on the number of functions. □

Fixed-parameter Tractability with Backdoor Sets

A promising type of parameters are distance measures to trivial instances. For instance, the number of non-Horn clauses h can be considered as a distance measure to Horn formulas. Recall that Horn formulas have an unique minimal model, which can be computed in linear time (Dowling and Gallier 1984). Another way of looking at the distance to trivial instances is the concept of (strong) backdoor sets. This concept was introduced in the context of FPT in (Nishimura, Ragde, and Szeider 2004). With the help of backdoor sets promising results for SAT, Quantified Boolean Formulas (Samer and Szeider 2009) and ASP (Fichte and Szeider 2011) have been obtained. We make use of strong Horn backdoor set, a special type of backdoor set.

Definition. A *strong Horn backdoor set* of a formula φ is a set $\mathcal{B} \subseteq \mathit{var}(\varphi)$ such that for any assignment $(\mathcal{T}, \mathcal{F})$ of $\mathcal{B}$, $\varphi[\mathcal{T}, \mathcal{F}]$ is a Horn formula.

An example of a strong Horn backdoor set can be found later on in Example 5. Computing a strong Horn backdoor set of size b is equivalent to computing a vertex cover of size b. The currently best algorithm for computing a vertex cover of size b (Chen, Kanj, and Xia 2010) has a runtime of $\mathcal{O}(1.2738^b + b \cdot n)$ and consequently a strong Horn backdoor set can be computed in this time. An fpt algorithm for computing an arbitrary minimal model that makes use of a concept similar to strong Horn backdoor sets was already presented in (Ben-Eliyahu and Dechter 1996). We are, however, interested in finding minimal models which fulfill a certain property π. The following algorithm solves WMMSAT with the help of a strong Horn backdoor set.

Theorem 11. *Let b be the size of a strong Horn backdoor set for φ, which is given as additional input. Then* WMMSAT *can be solved in time $\mathcal{O}(3^b \cdot n)$.*

Proof. Let $\mathcal{B} \subseteq \mathit{var}(\varphi)$ with $|\mathcal{B}| = b$ be the backdoor set given in the input. In the algorithm all possible assignments of $\mathcal{B}$ are considered. For each such assignment $(\mathcal{T}, \mathcal{F})$ we compute the reduced formula $\varphi[\mathcal{T}, \mathcal{F}]$. Since $\varphi[\mathcal{T}, \mathcal{F}]$ is a Horn formula we can compute its unique minimal model in linear time. In case $\varphi[\mathcal{T}, \mathcal{F}]$ is unsatisfiable, we continue with the next assignment of $\mathcal{B}$. If $\varphi[\mathcal{T}, \mathcal{F}]$ is satisfiable we have obtained a model $\mathcal{M}$ of φ and check whether it also satisfies π and is of size $\leq k$. If one of these conditions is violated we continue the search. In case both conditions are fulfilled we check for all $\mathcal{S} \subset \mathcal{M} \cap \mathcal{B}$ – using the previously described method – whether $\varphi[\mathcal{S}, \mathcal{B} \setminus \mathcal{S}]$ has a model $\mathcal{M}' \subseteq \mathcal{M}$. If no such $\mathcal{M}'$ can be found, the algorithm terminates with Yes. Should there be such an $\mathcal{M}'$ then $\mathcal{S} \cup \mathcal{M}'$ is a model of φ and a proper subset of $\mathcal{S} \cup \mathcal{M}$. Hence the search is continued until each assignment of $\mathcal{B}$ has been considered.

We now establish an upper bound on the runtime. Given an assignment $(\mathcal{T}, \mathcal{F})$, computing $\varphi[\mathcal{T}, \mathcal{F}]$ and finding the unique minimal model of $\varphi[\mathcal{T}, \mathcal{F}]$ can be done in linear time. In order to ensure minimality we have to check $2^p - 1$ subsets for every assignment of $\mathcal{B}$ setting p variables to true. Thus, the minimality check for such an assignment of $\mathcal{B}$ takes $(2^p - 1) \cdot \mathcal{O}(n)$ time and the total runtime for an assignment is $\mathcal{O}(n) + (2^p - 1) \cdot \mathcal{O}(n)$. There are $\binom{b}{p}$ possible assignments that set p variables to true. Taken together the runtime of the algorithm can be bounded by $\sum_{i=0}^{b} \binom{b}{i} 2^i \cdot \mathcal{O}(n) = \mathcal{O}(3^b \cdot n)$. □

Example 5. We now illustrate the concept of strong Horn backdoor sets and the presented algorithm. Let a WMMSAT instance be given by $\pi := (\neg b \vee d)$, $k := 2$ and $\varphi := (a \vee b \vee c) \wedge (\neg c \vee f) \wedge (\neg a \vee d) \wedge (\neg c \vee \neg d) \wedge (\neg a \vee \neg b \vee f) \wedge (a \vee b \vee \neg c)$. It can easily be verified that $\{a, b\}$ is a strong Horn backdoor set. We depict the results of the algorithm in Table 2. Observe that the first row is dropped because it is not subset minimal. Furthermore, notice that $\{b\}$ cannot be a solution since it does not satisfy π. Thus, the only solution for the given WMMSAT instance is $\{a, d\}$.

a	b	$\varphi[\mathcal{T}, \mathcal{F}]$	Model?
~~1~~	~~1~~	~~$(\neg c \vee f) \wedge d \wedge (\neg c \vee \neg d) \wedge f$~~	~~$\{a, b, d, f\}$~~
1	0	$(\neg c \vee f) \wedge d \wedge (\neg c \vee \neg d)$	$\{a, d\}$
0	1	$(\neg c \vee f) \wedge (\neg c \vee \neg d)$	$\{b\}$
0	0	$c \wedge (\neg c \vee f) \wedge (\neg c \vee \neg d) \wedge \neg c$	UNSAT

Table 2: Example of the algorithm presented in Theorem 11

The next two results follow from Theorem 11.

Corollary 12. WMMSAT *can be solved in time $\mathcal{O}(3^{v^+} \cdot n)$.*

Proof. Let V^+ be the set of all variables that appear as positive literals in $\varphi \wedge \pi$. Observe that V^+ is a strong Horn backdoor set and can be computed even in linear time. □

Corollary 13. WMMSAT *can be solved in time $\mathcal{O}(3^{h \cdot d^+} \cdot n)$.*

Proof. Notice that all variables that appear positively in non-Horn clauses (at most $d^+ \cdot h$ many) are a strong Horn backdoor set. □

Hardness Results

In this section we show that this paper contains all possible fpt results with respect to the parameters in Table 1. This is achieved by four proofs showing hardness with respect to combinations of parameters. Note that hardness for a set of parameters implies hardness for any subset of these parameters as well. In contrast to this, an fpt result with respect to a set of parameters implies fixed-parameter tractability for any superset of these parameters.

Theorem 14. WMMSAT *parameterized by k, d^-, h, p, d^+_π and $\|\pi\|$ is* W[1]*-hard.*

Proof. We give an fpt-reduction from INDEPENDENT SET, which is W[1]-complete when parameterized by the solution size k – see e.g. (Downey and Fellows 1999). An instance of INDEPENDENT SET is given by a graph $G = (V, E)$ and an integer $k > 0$. The question is whether there is some $V' \subseteq V$ with $|V'| = k$ such that there is no $\{x, y\} \in E$ with $x \in V'$ and $y \in V'$.

We construct the WMMSAT instance as follows. Let $V = \{v_1, \ldots, v_m\}$. The variables used in φ are going to be $\{v_1, \ldots, v_m\} \cup \{v^1_1, \ldots, v^1_m, \ldots, v^k_1, \ldots, v^k_m\}$. The WMMSAT instance is then given by $(\varphi, \mathit{true}, 2k)$, where φ is defined in the following way:

$$\varphi_{\text{IS}} := \bigwedge_{\{x,y\} \in E} (\neg x \vee \neg y)$$

$$\varphi_1 := \bigwedge_{l \in [k]} (v^l_1 \vee \ldots \vee v^l_m)$$

$$\varphi_2 := \bigwedge_{i \in [m]} \; \bigwedge_{1 \leq l < l' \leq k} (\neg v^l_i \vee \neg v^{l'}_i)$$

$$\varphi_3 := \bigwedge_{i \in [m]} \; \bigwedge_{l \in [k]} (v^l_i \rightarrow v_i)$$

$$\varphi := \varphi_{\text{IS}} \wedge \varphi_1 \wedge \varphi_2 \wedge \varphi_3$$

We continue by explaining the functionality of the subformulas. The independent set property is enforced by subformula φ_{IS}. The subformula φ_1 introduces k copies of each vertex and ensures that at least k of these $m \cdot k$ copies have to be in the model. Subformula φ_2 ensures that it is not possible to set more than one copy of each vertex to true. Finally, subformula φ_3 ensures that a vertex has to be in the model if one of its copies is in the model. The correctness proof of this reduction is omitted due to space constraints.

It remains to be verified that all parameters can be bounded in terms of k. The parameters d^+_π and $\|\pi\|$ are 0.The only non-Horn clauses occur in φ_1 and there are k of them. Each variable occurs positively at most k times and therefore p is bounded by k. Finally, the maximal negative clause size in φ is 2, which gives a bound on d^-. □

Before we continue with the next hardness proof concerning WMMSAT, we show that the following variation of DOMINATING SET is W[2]-hard as well.

MULTICOLOR DOMINATING SET
Instance: A graph $G = (V, E)$, an integer $k > 0$ and a k-coloring $c : V \to [k]$
Question: Is there a size k dominating set containing exactly one vertex of each color?

Proposition 15. MULTICOLOR DOMINATING SET *parameterized by k is* W[2]*-hard.*

Proof. We give an fpt-reduction from DOMINATING SET, which is W[2]-complete when parameterized by k – see e.g. (Downey and Fellows 1999). Given a graph (V', E') and $k \in \mathbb{N}$, the problem asks whether there is a dominating set $S \subseteq V'$ of cardinality $\leq k$, i.e. $V' = \bigcup_{v \in S} N(v)$. We construct a MULTICOLOR DOMINATING SET instance (G, k) with $G = (V, E)$. Let $V = \{v_1, \dots, v_m\}$. The new vertex set consists of k copies of V', to be specific $V' := \{v_1^1, \dots, v_m^1, v_1^2, \dots, v_m^2, \dots, v_1^k, \dots, v_m^k\}$. We define $E := \big\{\{v_{i_1}^{j_1}, v_{i_2}^{j_2}\} \mid \{v_{i_1}, v_{i_2}\} \in E' \wedge j_1, j_2 \in [k]\big\} \cup \big\{\{v_i^{j_1}, v_i^{j_2}\} \mid 1 \leq j_1 < j_2 \leq k\big\}$. The coloring is defined by $c(v_i^j) := j$. Observe that G' has a dominating set of size $\leq k$ iff G has a multicolor dominating set of size k. □

Theorem 16. WMMSAT *parameterized by k, d^-, h, p and v^- is* W[2]*-hard.*

Proof. Let (G, k) be a given MULTICOLOR DOMINATING SET instance with $G = (V, E)$ and a coloring $c : V \to [k]$. We now define for each $j \in [k]$ the set of j-colored vertices $V_j := \{v \in V \mid c(v) = j\}$. The formula φ is defined as $\varphi := \big\{V_j \mid j \in [k]\big\}$, i.e. every set V_j corresponds to a (positive) clause. The formula π is defined as $\pi := \{N(v) \mid v \in V\}$, i.e. each neighborhood is a clause. This yields the WMMSAT instance (φ, π, k) with parameters $h = k$, $p = 1$ (recall that h and p are not concerned with π), $d^- = 0$ and $v^- = 0$. We are going to show that G has a multicolor dominating set of size k iff (φ, π, k) is a Yes-instance.

Let $D = \{v_{i_1}, \dots, v_{i_k}\}$ be a multicolor dominating set of G. Without loss of generality let $c(v_{i_1}) = 1$, $c(v_{i_2}) = 2$, etc. Observe that D is a minimal model of φ. D is also model of π since being a multicolor dominating set it has to contain a vertex of each neighborhood in G.

For the other direction let $\mathcal{M} = \{v_{i_1}, \dots, v_{i_k}\} \subseteq \mathit{var}(\varphi) = V$ be a minimal model of φ and a model of π. Since every variable occurs in only one clause in φ, the vertices $v_{i_1}, \dots, v_{i_k}$ have to be of different color. Furthermore, since $\mathcal{M}$ is model of π, $\mathcal{M}$ has a non-empty intersection with each neighborhood and is hence a multicolor dominating set of G. □

Theorem 17. WMMSAT *parameterized by k, d^-, v^-, d_π^+ and $\|\pi\|$ is* W[2]*-hard.*

Proof. We show this hardness result by an fpt-reduction from DOMINATING SET parameterized by k. Given the input graph $G = (V, E)$ and the solution size k, we create the formula $\varphi := \{N(v) \mid v \in V\}$. The WMMSAT instance is then defined as $(\varphi, \mathit{true}, k)$. Since π is trivially fulfilled, the problem "degenerates" to finding a model of φ with weight at most k. Such a model directly corresponds to a dominating set of G. Concerning the other parameters: d^- and v^- are 0 since φ contains only positive literals. The parameters d_π^+ and $\|\pi\|$ are also 0 since π does not contain any variables. □

Proposition 18. WMMSAT *parameterized by d, d^+, d^-, p, v^-, d_π^+ and $\|\pi\|$ is* para-NP*-hard.*

Proof. One can encode (similar to Theorem 17) the unparameterized VERTEX COVER problem, which is NP-hard on graphs with degree at most 3 (Garey and Johnson 1977). All considered parameters are then bounded by a constant. □

Conclusion

WMMSAT is a parameterized decision problem tailored to capture the complexity of computing minimal models. We have performed a complete parameterized complexity analysis with respect to the eleven parameters in Table 1. We have identified several fixed-parameter tractable fragments and designed corresponding fpt algorithms. The fpt results are presented as dashed boxes in Figure 2. Relations between parameters are indicated by arrows. Recall that an fpt result with respect to a set of parameters also holds for any *superset* of the parameters. For all remaining combinations of parameters we have shown hardness results and thereby ruled out the possibility of further fpt algorithms – unless FPT = W[1]. Note that hardness results with respect to a set of parameters also hold for any *subset* of the parameters. Therefore the six fpt results displayed in Figure 2 together with the four hardness theorems in this paper provide a complete parameterized complexity classification for all 2^{11} combinations of parameters.

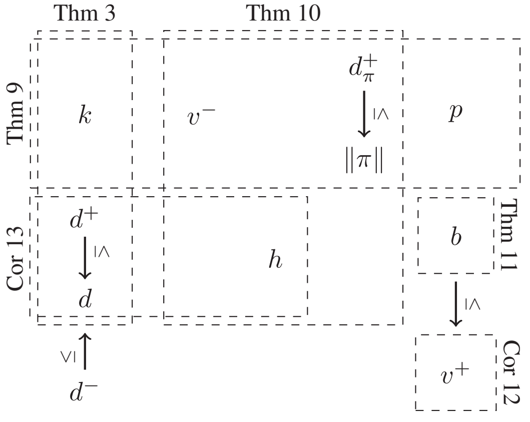

Figure 2: A map of the presented fpt algorithms.

The application of our results to minimal model-related problems such as circumscription, answer-set programming and diagnosis remains as future work. Also to experimentally evaluate how well the fpt algorithms behave in practice remains to be done. We expect that our fpt algorithms perform well and can significantly improve algorithms having to compute minimal models.

References

Baral, C.; Provetti, A.; and Son, T. C. 2003. Special issue on programming with answer sets. *Theory and Practice of Logic Programming* 3(4+5).

Ben-Eliyahu, R., and Dechter, R. 1996. On computing minimal models. *Ann. Math. Artif. Intell.* 18(1):3–27.

Chen, Y., and Flum, J. 2008. The parameterized complexity of maximality and minimality problems. *Ann. Pure Appl. Logic* 151(1):22–61.

Chen, J.; Kanj, I. A.; and Xia, G. 2010. Improved upper bounds for vertex cover. *Theor. Comput. Sci.* 411(40-42):3736–3756.

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

Downey, R. G., and Fellows, M. R. 1999. *Parameterized Complexity*. Springer.

Downey, R. G.; Fellows, M. R.; and Regan, K. W. 1998. Descriptive complexity and the W hierarchy. In *Proof Complexity and Feasible Arithmetic*, volume 39 of *AMS-DIMACS Volume Series*, 119–134. AMS.

Eiter, T., and Gottlob, G. 1993. Propositional circumscription and extended closed-world reasoning are Π_2^P-complete. *Theor. Comput. Sci.* 114(2):231–245.

Fichte, J. K., and Szeider, S. 2011. Backdoors to tractable answer-set programming. In *Proc. of IJCAI 2011*, 863–868. IJCAI/AAAI.

Flum, J., and Grohe, M. 2003. Describing parameterized complexity classes. *Inf. Comput.* 187(2):291–319.

Flum, J., and Grohe, M. 2005. Model-checking problems as a basis for parameterized intractability. *Logical Methods in Computer Science* 1(1):1–36.

Flum, J., and Grohe, M. 2006. *Parameterized Complexity Theory*. Springer.

Garey, M. R., and Johnson, D. S. 1977. The rectilinear steiner tree problem is NP-complete. *SIAM Journal on Applied Mathematics* 32(4):pp. 826–834.

Gelfond, M., and Leone, N. 2002. Logic programming and knowledge representation - the A-Prolog perspective. *Artif. Intell.* 138(1-2):3–38.

Gottlob, G., and Szeider, S. 2008. Fixed-parameter algorithms for artificial intelligence, constraint satisfaction and database problems. *The Computer Journal* 51(3):303–325.

Gottlob, G.; Pichler, R.; and Wei, F. 2010. Bounded treewidth as a key to tractability of knowledge representation and reasoning. *Artif. Intell.* 174(1):105–132.

Gottlob, G.; Scarcello, F.; and Sideri, M. 2002. Fixed-parameter complexity in AI and nonmonotonic reasoning. *Artif. Intell.* 138(1-2):55–86.

Jakl, M.; Pichler, R.; and Woltran, S. 2009. Answer-set programming with bounded treewidth. In *Proc. of IJCAI 2009*, 816–822.

Lifschitz, V. 1985. Closed-world databases and circumscription. *Artif. Intell.* 27(2):229–235.

Marek, V. W., and Truszczynski, M. 1999. Stable models and an alternative logic programming paradigm. In *The Logic Programming Paradigm: A 25-Year Perspective*, 375–398. Springer.

McCarthy, J. 1980. Circumscription - a form of non-monotonic reasoning. *Artif. Intell.* 13(1-2):27–39.

Nishimura, N.; Ragde, P.; and Szeider, S. 2004. Detecting backdoor sets with respect to Horn and binary clauses. In *Proc. of SAT 2004*.

Pichler, R.; Rümmele, S.; Szeider, S.; and Woltran, S. 2010. Tractable answer-set programming with weight constraints: Bounded treewidth is not enough. In *Proc. of KR 2010*, 508–517. AAAI Press.

Reiter, R. 1987. A theory of diagnosis from first principles. *Artif. Intell.* 32(1):57–95.

Samer, M., and Szeider, S. 2009. Backdoor sets of quantified Boolean formulas. *J. Autom. Reasoning* 42(1):77–97.

Truszczynski, M. 2002. Computing large and small stable models. *Theory and Practice of Logic Programming* 2(1):1–23.