Fixed-parameter complexity of semantics for logic programs

Zbigniew Lonc* and Mirosław Truszczyński**

Department of Computer Science, University of Kentucky
Lexington KY 40506-0046, USA
{lonc, mirek}@cs.engr.uky.edu

Abstract. In the paper we establish the fixed-parameter complexity for several parameterized decision problems involving models, supported models and stable models of logic programs. We also establish the fixed-parameter complexity for variants of these problems resulting from restricting attention to Horn programs and to purely negative programs. Most of the problems considered in the paper have high fixed-parameter complexity. Thus, it is unlikely that fixing bounds on models (supported models, stable models) will lead to fast algorithms to decide the existence of such models.

1 Introduction

In this paper we study the complexity of parameterized decision problems concerning models, supported models and stable models of logic programs. In our investigations, we use the framework of the *fixed-parameter complexity* introduced by Downey and Fellows [DF97]. This framework was previously used to study the problem of the existence of stable models of logic programs in [Tru01]. Our present work extends results obtained there. First, in addition to the class of all finite propositional logic programs, we consider its two important subclasses: the class of Horn programs and the class of purely negative programs. Second, in addition to stable models of logic programs, we also study supported models and arbitrary models.

A decision problem is *parameterized* if its inputs are *pairs* of items. The second item in a pair is referred to as a *parameter*. The problems to decide, given a logic program P and an integer k, whether P has a model, supported model or a stable model, respectively, with *at most* k atoms are examples of parameterized decision problems. These problems are NP-complete. However, fixing k (that is, k is no longer regarded as a part of input) makes each of the problems simpler. They become solvable in polynomial time. The following straightforward algorithm works: for every subset $M \subseteq At(P)$ of cardinality at most k, check whether M

* On leave from Faculty of Mathematics and Information Science, Warsaw University of Technology.

** The second author was partially supported by the NSF grants CDA-9502645, IRI-9619233 and EPS-9874764.

is a model, supported model or stable model, respectively, of P. The check can be implemented to run in linear time in the size of the program. Since there are $O(n^k)$ sets to be tested, the overall running time of this algorithm is $O(mn^k)$, where m is the size of the input program P and n is the number of atoms in P.

The problem is that algorithms with running times given by $O(mn^k)$ are not practical even for quite small values of k. The question then arises whether better algorithms can be found, for instance, algorithms whose running-time estimate would be given by a polynomial of the order that *does not depend on* k. Such algorithms, if they existed, could be practical for a wide range of values of k and could find applications in computing stable models of logic programs.

This question is the subject of our work. We also consider similar questions concerning related problems of deciding the existence of models, supported models and stable models of cardinality *exactly* k and *at least* k. We refer to all these problems as *small-bound* problems since k, when fixed, can be regarded as "small". In addition, we study problems of existence of models, supported models and stable models of cardinality at most $|At(P)| - k$, exactly $|At(P)| - k$ and at least $|At(P)| - k$. We refer to these problems as *large-bound* problems, since $|At(P)| - k$, for a fixed k, can be informally thought of as "large".

We address these questions using the framework of fixed-parameter complexity [DF97]. Most of our results are negative. They provide strong evidence that for many parameterized problems considered in the paper there are no algorithms whose running time could be estimated by a polynomial of order independent of k.

Formally, a *parameterized* decision problem is a set $L \subseteq \Sigma^* \times \Sigma^*$, where Σ is a fixed alphabet. By selecting a concrete value $\alpha \in \Sigma^*$ of the parameter, a parameterized decision problem L gives rise to an associated *fixed-parameter* problem $L_\alpha = \{x : (x, \alpha) \in L\}$.

A parameterized problem $L \subseteq \Sigma^* \times \Sigma^*$ is *fixed-parameter tractable* if there exist a constant t, an integer function f and an algorithm A such that A determines whether $(x, y) \in L$ in time $f(|y|)|x|^t$ ($|z|$ stands for the length of a string $z \in \Sigma^*$). We denote the class of fixed-parameter tractable problems by FPT. Clearly, if a parameterized problem L is in FPT, then each of the associated fixed-parameter problems L_y is solvable in polynomial time by an algorithm whose exponent does not depend on the value of the parameter y. Parameterized problems that are not fixed-parameter tractable are called *fixed-parameter intractable.*

To study and compare the complexity of parameterized problems Downey and Fellows proposed the following notion of *fixed-parameter reducibility* (or, simply, *reducibility*).

Definition 1. *A parameterized problem L can be* reduced *to a parameterized problem L' if there exist a constant p, an integer function q, and an algorithm A such that:*

1. *A assigns to each instance (x, y) of L an instance (x', y') of L',*
2. *A runs in time $O(q(|y|)|x|^p)$,*

3. x' *depends upon* x *and* y*, and* y' *depends upon* y *only,*
4. $(x, y) \in L$ *if and only if* $(x', y') \in L'$*.*

We will use this notion of reducibility throughout the paper. If for two parameterized problems L_1 and L_2, L_1 can be reduced to L_2 and conversely, we say that L_1 and L_2 are *fixed-parameter equivalent* or, simply, *equivalent*.

Downey and Fellows [DF97] defined a hierarchy of complexity classes called the *W hierarchy*:

$$\mathrm{FPT} \subseteq \mathrm{W}[1] \subseteq \mathrm{W}[2] \subseteq \mathrm{W}[3] \subseteq \ldots \quad (1)$$

The classes W[t] can be described in terms of problems that are complete for them (a problem D is *complete* for a complexity class $\mathcal{E}$ if $D \in \mathcal{E}$ and every problem in this class can be reduced to D). Let us call a Boolean formula *t-normalized* if it is a conjunction-of-disjunctions-of-conjunctions ... of literals, with t being the number of conjunctions-of, disjunctions-of expressions in this definition. For example, 2-normalized formulas are conjunctions of disjunctions of literals. Thus, the class of 2-normalized formulas is precisely the class of CNF formulas. We define the *weighted t-normalized satisfiability problem* as:

WS(t)**:** Given a t-normalized formula Φ and a non-negative integer k, decide whether there is a model of Φ with exactly k atoms (or, alternatively, decide whether there is a satisfying valuation for Φ which assigns the logical value **true** to exactly k atoms).

Downey and Fellows show that for every $t \geq 2$, the problem *WS*(t) is complete for the class W[t]. They also show that a restricted version of the problem *WS*(2):

$WS_2(2)$**:** Given a 2-normalized formula Φ with each clause consisting of at most two literals, and an integer k, decide whether there is a model of Φ with exactly k atoms

is complete for the class W[1]. There is strong evidence suggesting that all the implications in (1) are proper. Thus, proving that a parameterized problem is complete for a class W[t], $t \geq 1$, is a strong indication that the problem is not fixed-parameter tractable.

As we stated earlier, in the paper we study the complexity of parameterized problems related to logic programming. All these problems ask whether an input program P has a model, supported model or a stable model satisfying some cardinality constraints involving another input parameter, an integer k. They can be categorized into two general families: *small-bound* problems and *large-bound* problems. In the formal definitions given below, $\mathcal{C}$ denotes a class of logic programs, $\mathcal{D}$ represents a class of models of interest and Δ stands for one of the three arithmetic relations: "$\leq$", "$=$" and "$\geq$".

$\mathcal{D}_\Delta(\mathcal{C})$**:** Given a logic program P from class $\mathcal{C}$ and an integer k, decide whether P has a model M from class $\mathcal{D}$ such that $|M| \; \Delta \; k$.

$\mathcal{D}'_\Delta(\mathcal{C})$**:** Given a logic program P from class $\mathcal{C}$ and an integer k, decide whether P has a model M from class $\mathcal{D}$ such that $(|At(P)| - k) \; \Delta \; |M|$.

In the paper, we consider three classes of programs: the class of Horn programs $\mathcal{H}$, the class of purely negative programs $\mathcal{N}$, and the class of all programs $\mathcal{A}$. We also consider three classes of models: the class of all models $\mathcal{M}$, the class of supported models $\mathcal{SP}$ and the class of stable models $\mathcal{ST}$.

Thus, for example, the problem $\mathcal{SP}_{\leq}(\mathcal{N})$ asks whether a purely negative logic program P has a supported model M with no more than k atoms ($|M| \leq k$). The problem $\mathcal{ST}'_{\leq}(\mathcal{A})$ asks whether a logic program P (with no syntactic restrictions) has a stable model M in which at most k atoms are false ($|At(P)| - k \leq |M|$). Similarly, the problem $\mathcal{M}'_{\geq}(\mathcal{H})$ asks whether a Horn program P has a model M in which at least k atoms are false ($|At(P)| - k \geq |M|$).

In the three examples given above and, in general, for all problems $\mathcal{D}_{\Delta}(\mathcal{C})$ and $\mathcal{D}'_{\Delta}(\mathcal{C})$, the input instance consists of a logic program P from the class $\mathcal{C}$ and of an integer k. We will regard these problems as parameterized with k. Fixing k (that is, k is no longer a part of input but an element of the problem description) leads to the fixed-parameter versions of these problems. We will denote them $\mathcal{D}_{\Delta}(\mathcal{C}, k)$ and $\mathcal{D}'_{\Delta}(\mathcal{C}, k)$, respectively.

In the paper, for all but three problems $\mathcal{D}_{\Delta}(\mathcal{C})$ and $\mathcal{D}'_{\Delta}(\mathcal{C})$, we establish their fixed-parameter complexities. Our results are summarized in Tables 1 - 3.

	$\mathcal{H}$	$\mathcal{N}$	$\mathcal{A}$
$\mathcal{M}$	P	P	P
$\mathcal{M}'$	P	W[1]-c	NP-c
$\mathcal{SP}$	P	NP-c	NP-c
$\mathcal{SP}'$	P	NP-c	NP-c
$\mathcal{ST}$	P	NP-c	NP-c
$\mathcal{ST}'$	P	NP-c	NP-c

Table1. The complexities of the problems $\mathcal{D}_{\geq}(\mathcal{C})$ and $\mathcal{D}'_{\geq}(\mathcal{C})$.

In Table 1, we list the complexities of all problems in which $\Delta =$ "$\geq$". Small-bound problems of this type ask about the existence of models of a program P that contain at least k atoms. Large-bound problems in this group are concerned with the existence of models that contain at most $|At(P)| - k$ atoms (the number of false atoms in these models is at least k). From the point of view of the fixed-parameter complexity, these problems are not very interesting. Several of them remain NP-complete even when k is fixed. In other words, fixing k does not simplify them enough to make them tractable. For this reason, all the entries in Table 1, listing the complexity as NP-complete (denoted by NP-c in the table), refer to fixed-parameter versions $\mathcal{D}_{\geq}(\mathcal{C}, k)$ and $\mathcal{D}'_{\geq}(\mathcal{C}, k)$ of problems $\mathcal{D}_{\geq}(\mathcal{C})$ and $\mathcal{D}'_{\geq}(\mathcal{C})$. The problem $\mathcal{M}'_{\geq}(\mathcal{A}, k)$ is NP-complete for every fixed $k \geq 1$. All other fixed-parameter problems in Table 1 that are marked NP-complete are NP-complete for every value $k \geq 0$.

On the other hand, many problems $\mathcal{D}_{\geq}(\mathcal{C})$ and $\mathcal{D}'_{\geq}(\mathcal{C})$ are "easy". They are fixed-parameter tractable in a strong sense. They can be solved in polynomial time even *without* fixing k. This is indicated by marking the corresponding entries in Table 1 with P (for the class P) rather than with FPT. There is only one exception, the problem $\mathcal{M}'_{\geq}(\mathcal{N})$, which is W[1]-complete.

Small-bound problems for the cases when Δ = "=" or "$\leq$" can be viewed as problems of deciding the existence of "small" models (that is, models containing exactly k or at most k atoms). The fixed-parameter complexities of these problems are summarized in Table 2.

	$\mathcal{H}_{\leq}$	$\mathcal{H}_{=}$	$\mathcal{N}_{\leq}$	$\mathcal{N}_{=}$	$\mathcal{A}_{\leq}$	$\mathcal{A}_{=}$
$\mathcal{M}$	P	W[1]-c	W[2]-c	W[2]-c	W[2]-c	W[2]-c
$\mathcal{SP}$	P	W[1]-h, in W[2]	W[2]-c	W[2]-c	W[2]-c	W[2]-c
$\mathcal{ST}$	P	P	W[2]-c	W[2]-c	W[2]-c	W[2]-c

Table 2. The complexities of the problem of computing small models (small-bound problems, the cases of Δ = "=" and "$\leq$").

The problems involving the class of all purely negative programs and the class of all programs are W[2]-complete. This is a strong indication that they are fixed-parameter intractable. All problems of the form $\mathcal{D}_{\leq}(\mathcal{H})$ are fixed-parameter tractable. In fact, they are solvable in polynomial time even without fixing the parameter k. We indicate this by marking the corresponding entries with P. Similarly, the problem $\mathcal{ST}_{=}(\mathcal{H})$ of deciding whether a Horn logic program P has a stable model of size exactly k is in P. However, perhaps somewhat surprisingly, the remaining two problems involving Horn logic programs and Δ = "=" are harder. We proved that the problem $\mathcal{M}_{=}(\mathcal{H})$ is W[1]-complete and that the problem $\mathcal{SP}_{=}(\mathcal{H})$ is W[1]-hard. Thus, they most likely are not fixed-parameter tractable. We also showed that the problem $\mathcal{SP}_{=}(\mathcal{H})$ is in the class W[2]. The exact fixed-parameter complexity of $\mathcal{SP}_{=}(\mathcal{H})$ remains unresolved.

Large-bound problems for the cases when Δ = "=" or "$\leq$" can be viewed as problems of deciding the existence of "large" models, that is, models with a small number of false atoms — equal to k or less than or equal to k. The fixed-parameter complexities of these problems are summarized in Table 3.

	$\mathcal{H}_{\leq}$	$\mathcal{H}_{=}$	$\mathcal{N}_{\leq}$	$\mathcal{N}_{=}$	$\mathcal{A}_{\leq}$	$\mathcal{A}_{=}$
$\mathcal{M}'$	P	W[2]-c	P	W[1]-c	P	W[2]-c
$\mathcal{SP}'$	P	W[3]-c,	W[2]-c	W[2]-c	W[3]-c	W[3]-c
$\mathcal{ST}'$	P	P	W[2]-c	W[2]-c	W[3]-h	W[3]-h

Table 3. The complexities of the problems of computing large models (large-bound problems, the cases of Δ = "=" and "$\leq$").

The problems specified by Δ = "$\leq$" and concerning the existence of models are in P. Similarly, the problems specified by Δ = "$\leq$" and involving Horn programs are solvable in polynomial time. Lastly, the problem $\mathcal{ST}'_{=}(\mathcal{H})$ is in P, as well. These problems are in P even without fixing k and eliminating it from input. All other problems in this group have higher complexity and, in all likelihood, are fixed-parameter intractable. One of the problems, $\mathcal{M}'_{=}(\mathcal{N})$, is W[1]-complete. Most of the remaining problems are W[2]-complete. Surprisingly, some problems are even harder. Three problems concerning supported models are W[3]-complete. For two problems involving stable models, $\mathcal{ST}'_{=}(\mathcal{A})$ and $\mathcal{ST}'_{\leq}(\mathcal{A})$, we could only prove that they are W[3]-hard. For these two problems we did not succeed in establishing any upper bound on their fixed-parameter complexities.

The study of fixed-parameter tractability of problems occurring in the area of nonmonotonic reasoning is a relatively new research topic. The only two other papers we are aware of are [Tru01] and [GSS99]. The first of these two papers provided a direct motivation for our work here (we discussed it earlier). In the second one, the authors focused on parameters describing *structural* properties of programs. They showed that under some choices of the parameters decision problems for nonmonotonic reasoning become fixed-parameter tractable.

Our results concerning computing stable and supported models for logic programs are mostly negative. Parameterizing basic decision problems by constraining the size of models of interest does not lead (in most cases) to fixed-parameter tractability.

There are, however, several interesting aspects to our work. First, we identified some problems that are W[3]-complete or W[3]-hard. Relatively few problems from these classes were known up to now [DF97]. Second, in the context of the polynomial hierarchy, there is no distinction between the problem of existence of models of specified sizes of clausal propositional theories and similar problems concerning models, supported models and stable models of logic programs. All these problems are NP-complete. However, when we look at the complexity of these problems in a more detailed way, from the perspective of fixed-parameter complexity, the equivalence is lost. Some problems are W[3]-hard, while problems concerning existence of models of 2-normalized formulas are W[2]-complete or easier. Third, our results show that in the context of fixed-parameter tractability, several problems involving models and supported models are hard even for the class of Horn programs. Finally, our work leaves three problems unresolved. While we obtained some bounds for the problems $\mathcal{SP}_=(\mathcal{H})$, $\mathcal{ST}'_\leq(\mathcal{A})$ and $\mathcal{ST}'_=(\mathcal{A})$, we did not succeed in establishing their precise fixed-parameter complexities.

The rest of our paper is organized as follows. In the next section, we review relevant concepts in logic programming. After that, we present several useful fixed-parameter complexity results for problems of the existence of models for propositional theories of certain special types. In the last section we give proofs of some of our complexity results.

2 Preliminaries

In the paper, we consider only the propositional case. A logic program *clause* (or *rule*) is any expression r of the form

$$r = \quad p \leftarrow q_1, \ldots, q_m, \mathbf{not}(s_1), \ldots, \mathbf{not}(s_n), \tag{2}$$

where p, q_i and s_i are propositional atoms. We call the atom p the *head* of r and we denote it by $h(r)$. Further, we call the set of atoms $\{q_1, \ldots, q_m, s_1, \ldots, s_n\}$ the *body* of r and we denote it by $b(r)$. We distinguish the *positive body* of r,

$\{q_1, \ldots, q_m\}$ ($b^+(r)$, in symbols), and the *negative body* of r, $\{s_1, \ldots, s_n\}$ ($b^-(r)$, in symbols).

A *logic program* is a collection of clauses. For a logic program P, by $At(P)$ we denote the set of atoms that appear in P. If every clause in a logic program P has an empty negative body, we call P a *Horn* program. If every clause in P has an empty positive body, we call P a *purely negative* program.

A clause r, given by (2), has a *propositional interpretation* as an implication

$$pr(r) = \quad q_1 \wedge \ldots \wedge q_m \wedge \neg s_1 \wedge \ldots \wedge \neg s_n \Rightarrow p.$$

Given a logic program P, by a *propositional interpretation* of P we mean the propositional formula

$$pr(P) = \bigwedge \{pr(r) \colon r \in P\}.$$

We say that a set of atoms M is a *model* of a clause (2) if M is a (propositional) model of the clause $pr(r)$. As usual, atoms in M are interpreted as true, all other atoms are interpreted as false. A set of atoms $M \subseteq At(P)$ is a *model* of a program P if it is a model of the formula $pr(P)$. We emphasize the requirement $M \subseteq At(P)$. In this paper, given a program P, we are interested only in the truth values of atoms that actually occur in P.

It is well known that every Horn program P has a least model (with respect to set inclusion). We will denote this model by $lm(P)$.

Let P be a logic program. Following [Cla78], for every atom $p \in At(P)$ we define a propositional formula $comp(p)$ by

$$comp(p) = \quad p \Leftrightarrow \bigvee \{c(r) \colon r \in P,\ h(r) = p\},$$

where

$$c(r) = \bigwedge \{q \colon q \in b^+(r)\} \wedge \bigwedge \{\neg s \colon s \in b^-(r)\}.$$

If for an atom $p \in At(P)$ there are no rules with p in the head, we get an empty disjunction in the definition of $comp(p)$, which we interpret as a contradiction.

We define the *program completion* [Cla78] of P as the propositional theory

$$comp(P) = \{comp(p) \colon p \in At(P)\}.$$

A set of atoms $M \subseteq At(P)$ is a *supported model* of P if it is a model of the completion of P. It is easy to see that if p does not appear as the head of a rule in P, p is false in every supported model of P. It is also easy to see that each supported model of a program P is a model of P (the converse is not true in general).

Given a logic program P and a set of atoms M, we define the *reduct* of P with respect to M (P^M, in symbols) to be the logic program obtained from P by

1. removing from P each clause r such that $M \cap b^-(r) \neq \emptyset$ (we call such clauses *blocked by* M),

2. removing all negated atoms from the bodies of all the rules that remain (that is, those rules that are not blocked by M).

The reduct P^M is a Horn program. Thus, it has a least model. We say that M is a *stable model* of P if $M = lm(P^M)$. Both the notion of the reduct and that of a stable model were introduced in [GL88].

It is known that every stable model of a program P is a supported model of P. The converse does not hold in general. However, if a program P is purely negative, then stable and supported models of P coincide [Fag94].

In our arguments we use fixed-parameter complexity results on problems to decide the existence of models of prescribed sizes for propositional formulas from some special classes. To describe these problems we introduce additional terminology. First, given a propositional theory Φ, by $At(\Phi)$ we denote the set of atoms occurring in Φ. As in the case of logic programming, we consider as models of a propositional theory Φ only those sets of atoms that are subsets of $At(\Phi)$. Next, we define the following classes of formulas:

tN: the class of t-normalized formulas (if $t = 2$, these are simply CNF formulas)
2N$_3$: the class of all 2-normalized formulas whose every clause is a disjunction of at most three literals (clearly, *2N*$_3$ is a subclass of the class *2N*)
tNM: the class of *monotone* t-normalized formulas, that is, t-normalized formulas in which there are no occurrences of the negation operator
tNA: the class of *antimonotone* t-normalized formulas, that is, t-normalized formulas in which every atom is directly preceded by the negation operator.

Finally, we extend the notation $\mathcal{M}_\Delta(\mathcal{C})$ and $\mathcal{M}'_\Delta(\mathcal{C})$, to the case when $\mathcal{C}$ stands for a class of propositional formulas. In this terminology, $\mathcal{M}'_=(3NM)$ denotes the problem to decide whether a monotone 3-normalized formula Φ has a model in which exactly k atoms are false. Similarly, $\mathcal{M}_=(tN)$ is simply another notation for the problem *WS*[t] that we discussed above. The following theorem establishes several complexity results that we will use later in the paper.

Theorem 1. *(i) The problems $\mathcal{M}_=(2N)$ and $\mathcal{M}_=(2NM)$ are* W[2]*-complete.*
(ii) The problems $\mathcal{M}_=(2N_3)$ and $\mathcal{M}_=(2NA)$ are W[1]*-complete.*
(iii) The problem $\mathcal{M}'_\leq(3N)$ is $W[3]$-complete.

Proof: The statements (i) and (ii) are proved in [DF97]. To prove the statement (iii), we use the fact that the problem $\mathcal{M}_\leq(3N)$ is $W[3]$-complete [DF97]. We reduce $\mathcal{M}_\leq(3N)$ to $\mathcal{M}'_\leq(3N)$ and conversely. Let us consider a 3-normalized formula $\Phi = \bigwedge_{i=1}^{m} \bigvee_{j=1}^{m_i} \bigwedge_{\ell=1}^{m_{ij}} x[i,j,\ell]$, where $x[i,j,\ell]$ are literals. We observe that Φ has a model of cardinality at most k if and only if a related formula $\bar{\Phi} = \bigwedge_{i=1}^{m} \bigvee_{j=1}^{m_i} \bigwedge_{\ell=1}^{m_{ij}} \bar{x}[i,j,\ell]$, obtained from Φ by replacing every negative literal $\neg x$ by a new atom $\bar{x}$ and every positive literal x by a negated atom $\neg\bar{x}$, has a model of cardinality at least $|At(\bar{\Phi})| - k$. This construction defines a reduction of $\mathcal{M}_\leq(3N)$ to $\mathcal{M}'_\leq(3N)$. It is easy to see that this reduction satisfies all the requirements of the definition of fixed-parameter reducibility.

A reduction of $\mathcal{M}'_\leq(3N)$ to $\mathcal{M}_\leq(3N)$ can be constructed in a similar way. Since the problem $\mathcal{M}_\leq(3N)$ is W[3]-complete, so is the problem $\mathcal{M}'_\leq(3N)$. □

In the proof of part (iii) of Theorem 1, we observed that the reduction we described there satisfies all the requirements specified in Definition 1 of fixed-parameter reducibility. Throughout the paper we prove our complexity results by constructing reductions from one problem to another. In most cases, we only verify the condition (4) of the definition which, usually, is the only non-trivial part of the proof. Checking that the remaining conditions hold is straightforward and we leave these details out.

3 Some proofs

In this section we present some typical proofs of fixed-parameter complexity results for problems involving existence of models, supported models and stable models of logic programs. Our goal is to introduce key proof techniques that we used when proving the results discussed in the introduction.

Theorem 2. *The problems* $\mathcal{M}'_=(\mathcal{H})$ *and* $\mathcal{M}'_=(\mathcal{A})$ *are* W[2]*-complete.*

Proof: Both problems are clearly in W[2] (models of a logic program P are models of the corresponding 2-normalized formula $pr(P)$). Since $\mathcal{H} \subseteq \mathcal{A}$, to complete the proof it is enough to show that the problem $\mathcal{M}'_=(\mathcal{H})$ is W[2]-hard. To this end, we will reduce the problem $\mathcal{M}_=(\mathit{2NM})$ to $\mathcal{M}'_=(\mathcal{H})$.

Let Φ be a monotone 2-normalized formula and let $k \geq 0$. Let $\{x_1, \ldots, x_n\}$ be the set of atoms of Φ. We define a Horn program P_Φ corresponding to Φ as follows. We choose an atom a not occurring in Φ and include in P_Φ all rules of the form $x_i \leftarrow a$, $i = 1, 2, \ldots, n$. Next, for each clause $C = x_{i_1} \vee \ldots \vee x_{i_p}$ of Φ we include in P_Φ the rule

$$r_C = \quad a \leftarrow x_{i_1}, \ldots, x_{i_p}.$$

We will show that Φ has a model of size k if and only if P_Φ has a model of size $|At(P_\Phi)| - (k+1) = (n+1) - (k+1) = n - k$.

Let M be a model of Φ of size k. We define $M' = \{x_1, \ldots, x_n\} \setminus M$. The set M' has $n - k$ elements. Let us consider any clause $r_C \in P_\Phi$ of the form given above. Since M satisfies C, there is j, $1 \leq j \leq p$, such that $x_{i_j} \notin M'$. Thus, M' is a model of r_C. Since $a \notin M'$, M' satisfies all clauses $x_i \leftarrow a$. Hence, M' is a model of P_Φ.

Conversely, let M' be a model of P_Φ of size exactly $n - k$. If $a \in M'$ then $x_i \in M'$, for every i, $1 \leq i \leq n$. Thus, $|M'| = n + 1 > n - k$, a contradiction. Consequently, we obtain that $a \notin M'$. Let $M = \{x_1, \ldots, x_n\} \setminus M'$. Since $a \notin M'$, $|M| = k$. Moreover, M satisfies all clauses in Φ. Indeed, let us assume that there is a clause C such that no atom of C is in M. Then, all atoms of C are in M'. Since M' satisfies r_C, $a \in M'$, a contradiction. Now, the assertion follows by Theorem 1. □

Theorem 3. *The problem* $\mathcal{M}_=(\mathcal{H})$ *is* W[1]*-complete.*

Proof: We will first prove the hardness part. To this end, we will reduce the problem $\mathcal{M}_=(\mathit{2NA})$ to the problem $\mathcal{M}_=(\mathcal{H})$. Let Φ be an antimonotone 2-normalized formula and let k be a non-negative integer. Let $a_0, \ldots, a_k$ be $k+1$ different atoms not occurring in Φ. For each clause $C = \neg x_1 \vee \ldots \vee \neg x_p$ of Φ we define a logic program rule r_C by

$$r_C = \quad a_0 \leftarrow x_1, \ldots, x_p.$$

We then define P_Φ by

$$P_\Phi = \{r_C : C \in \Phi\} \cup \{a_i \leftarrow a_j : i, j = 0, 1, \ldots, k,\ i \neq j\}.$$

Let us assume that M is a model of size k of the program P_Φ. If for some i, $0 \leq i \leq k$, $a_i \in M$ then $\{a_0, \ldots, a_k\} \subseteq M$ and, consequently, $|M| > k$, a contradiction. Thus, M does not contain any of the atoms a_i. Since M satisfies all rules r_C and since it consists of atoms of Φ only, M is a model of Φ (indeed, the body of each rule r_C must be false so, consequently, each clause C must be true). Similarly, one can show that if M is a model of Φ then it is a model of P_Φ. Thus, W[1]-hardness follows by Theorem 1.

To prove that the problem $\mathcal{M}_=(\mathcal{H})$ is in the class W[1], we will reduce it to the problem $\mathcal{M}_=(\mathit{2N}_3)$. To this end, for every Horn program P we will describe a 2-normalized formula Φ_P, with each clause consisting of no more than three literals, and such that P has a model of size k if and only if Φ_P has a model of size $(k+1)2^k + k$. Moreover, we will show that Φ_P can be constructed in time bounded by a polynomial in the size of P (with the degree not depending on k).

First, let us observe that without loss of generality we may restrict our attention to Horn programs whose rules do not contain multiple occurrences of the same atom in the body. Such occurrences can be eliminated in time linear in the size of the program. Next, let us note that under this restriction, a Horn program P has a model of size k if and only if the program P', obtained from P by removing all clauses with bodies consisting of more than k atoms, has a model of size k. The program P' can be constructed in time linear in the size of P and k.

Thus, we will describe the construction of the formula Φ_P only for Horn programs P in which the body of every rule consists of no more than k atoms. Let P be such a program. We define

$$\mathcal{B} = \{B\colon\ B \subseteq b(r),\ \text{for some } r \in P\}.$$

For every set $B \in \mathcal{B}$ we introduce a new variable $u[B]$. Further, for every atom x in P we introduce 2^k new atoms $x[i]$, $i = 1, \ldots, 2^k$.

We will now define several families of formulas. First, for every $x \in \mathit{At}(P)$ and $i = 1, \ldots, 2^k$ we define

$$D(x,i) = \quad x \Leftrightarrow x[i] \quad (\text{or } (\neg x \vee x[i]) \wedge (x \vee \neg x[i])),$$

and, for each set $B \in \mathcal{B}$ and for each $x \in B$, we define

$$E(B,x) = \quad x \wedge u[B \setminus \{x\}] \Rightarrow u[B] \quad (\text{or } \neg x \vee \neg u[B \setminus \{x\}] \vee u[B]).$$

Next, for each set $B \in \mathcal{B}$ and for each $x \in B$ we define

$$F(B, x) = \quad u[B] \Rightarrow x \quad \text{(or } \neg u[B] \vee x\text{)}.$$

Finally, for each rule r in P we introduce a formula

$$G(r) = \quad u[b(r)] \Rightarrow h(r) \quad \text{(or } \neg u[b(r)] \vee h(r)\text{)}.$$

We define Φ_P to be the conjunction of all these formulas (more precisely, of their 2-normalized representations given in the parentheses) and of the formula $u[\emptyset]$. Clearly, Φ_P is a formula from the class $2N_3$. Further, since the body of each rule in P has at most k elements, the set $\mathcal{B}$ has no more than $|P|2^k$ elements, each of them of size at most k ($|P|$ denotes the cardinality of P, that is, the number of rules in P). Thus, Φ_P can be constructed in time bounded by a polynomial in the size of P, whose degree does not depend on k.

Let us consider a model M of P such that $|M| = k$. We define

$$M' = M \cup \{x[i] \colon x \in M, i = 1, \ldots, 2^k\} \cup \{u[B] \colon B \subseteq M\}.$$

The set M' satisfies all formulas $D(x, i)$, $x \in At(P)$, $i = 1, \ldots, 2^k$. In addition, the formula $u[\emptyset]$ is also satisfied by M' ($\emptyset \subseteq M$ and so, $u[\emptyset] \in M'$).

Let us consider a formula $E(B, x)$, for some $B \in \mathcal{B}$ and $x \in B$. Let us assume that $x \wedge u[B \setminus \{x\}]$ is true in M'. Then, $x \in M'$ and, since $x \in At(P)$, $x \in M$. Moreover, since $u[B \setminus \{x\}] \in M'$, $B \setminus \{x\} \subseteq M$. It follows that $B \subseteq M$ and, consequently, that $u[B] \in M'$. Thus, M' satisfies all "E-formulas" in Φ_P.

Next, let us consider a formula $F(B, x)$, where $B \in \mathcal{B}$ and $x \in B$, and let us assume that M' satisfies $u[B]$. It follows that $B \subseteq M$. Consequently, $x \in M$. Since $M \subseteq M'$, M' satisfies x and so, M' satisfies $F(B, x)$.

Lastly, let us look at a formula $G(r)$, where $r \in P$. Let us assume that $u[b(r)] \in M'$. Then, $b(r) \subseteq M$. Since r is a Horn clause and since M is a model of P, it follows that $h(r) \in M$. Consequently, $h(r) \in M'$. Thus, M' is a model of $G(r)$.

We proved that M' is a model of Φ_P. Moreover, it is easy to see that $|M'| = k + k2^k + 2^k = (k+1)2^k + k$.

Conversely, let us assume that M' is a model of Φ_P and that $|M'| = (k + 1)2^k + k$. We set $M = M' \cap At(P)$. First, we will show that M is a model of P.

Let us consider an arbitrary clause $r \in P$, say

$$r = h \leftarrow b_1, \ldots, b_p,$$

where h and b_i, $1 \leq i \leq p$, are atoms. Let us assume that $\{b_1, \ldots, b_p\} \subseteq M$. We need to show that $h \in M$.

Since $\{b_1, \ldots, b_p\} = b(r)$, the set $\{b_1, \ldots, b_p\}$ and all its subsets belong to $\mathcal{B}$. Thus, Φ_P contains formulas

$$E(\{b_1, \ldots, b_{i-1}\}, b_i) = b_i \wedge u[\{b_1, \ldots, b_{i-1}\}] \Rightarrow u[\{b_1, \ldots, b_{i-1}, b_i\}],$$

where $i = 1, \ldots, p$. All these formulas are satisfied by M'. We also have $u[\emptyset] \in \Phi_P$. Consequently, $u[\emptyset]$ is satisfied by M', as well. Since all atoms b_i, $1 \leq i \leq p$, are also satisfied by M' (since $M \subseteq M'$), it follows that $u[\{b_1,\ldots,b_p\}]$ is satisfied by M'.

The formula $G(r) = u[\{b_1,\ldots,b_p\}] \Rightarrow h$ belongs to Φ_P. Thus, it is satisfied by M'. It follows that $h \in M'$. Since $h \in At(P)$, $h \in M$. Thus, M is a model of r and, consequently, of the program P.

To complete the proof we have to show that $|M| = k$. Since M' is a model of Φ_P, for every $x \in M$, M' contains all atoms $x[i]$, $1 \le i \le 2^k$. Hence, if $|M| > k$ then $|M'| \ge |M| + |M| \times 2^k \ge (k+1)(1+2^k) > (k+1)2^k + k$, a contradiction.

So, we will assume that $|M| < k$. Let us consider an atom $u[B]$, where $B \in \mathcal{B}$, such that $u[B] \in M'$. For every $x \in B$, Φ_P contains the rule $F(B,x)$. The set M' is a model of $F(B,x)$. Thus, $x \in M'$ and, since $x \in At(P)$, we have that $x \in M$. It follows that $B \subseteq M$. It is now easy to see that the number of atoms of the form $u[B]$ that are true in M' is smaller than 2^k. Thus, $|M'| < |M| + |M| \times 2^k + 2^k \le (k-1)(1+2^k) + 2^k < (k+1)2^k + k$, again a contradiction. Consequently, $|M| = k$.

It follows that the problem $\mathcal{M}_=(\mathcal{H})$ can be reduced to the problem $\mathcal{M}_=(2N_3)$. Thus, by Theorem 1, the problem $\mathcal{M}_=(\mathcal{H})$ is in the class W[1]. This completes our argument. □

Theorem 4. *The problem $\mathcal{SP}_=(\mathcal{A})$ is in $W[2]$.*

Proof: We will show a reduction of $\mathcal{SP}_=(\mathcal{A})$ to $\mathcal{M}_=(2N)$, which is in $W[2]$ by Theorem 1. Let P be a logic program with atoms $x_1,\ldots,x_n$. We can identify supported models of P with models of its completion $comp(P)$. The completion is of the form $comp(P) = \Phi_1 \wedge \ldots \wedge \Phi_n$, where

$$\Phi_i = x_i \Leftrightarrow \bigvee_{j=1}^{m_i} \bigwedge_{\ell=1}^{m_{ij}} x[i,j,\ell],$$

$i = 1,\ldots,n$, and $x[i,j,\ell]$ are literals. It can be constructed in linear time in the size of the program P.

We will use $comp(P)$ to define a formula Φ_P. The atoms of Φ_P are $x_1,\ldots,x_n$ and $u[i,j]$, $i = 1,\ldots,n$, $j = 1,\ldots,m_i$. For $i = 1,\ldots,n$, let

$$G_i = x_i \Rightarrow \bigvee_{j=1}^{m_i} u[i,j], \qquad (\text{or } \neg x_i \vee \bigvee_{j=1}^{m_i} u[i,j]),$$

$$G'_i = \bigvee_{j=1}^{m_i} u[i,j] \Rightarrow x_i, \qquad (\text{or } \bigwedge_{j=1}^{m_i} (x_i \vee \neg u[i,j])),$$

$$H_i = \bigwedge_{j=1}^{m_i-1} \bigwedge_{j'=j+1}^{m_i} (\neg u[i,j] \vee \neg u[i,j']), \quad \text{for every } i \text{ such that } m_i \ge 2,$$

$$I_i = \bigwedge_{j=1}^{m_i} (u[i,j] \Rightarrow \bigwedge_{\ell=1}^{m_{ij}} x[i,j,\ell]) \qquad (\text{or } \bigwedge_{j=1}^{m_i} \bigwedge_{\ell=1}^{m_{ij}} (\neg u[i,j] \vee x[i,j,\ell])),$$

$$J_i = \bigvee_{j=1}^{m_i} \bigwedge_{\ell=1}^{m_{ij}} x[i,j,\ell] \Rightarrow x_i, \qquad (\text{or } \bigwedge_{j=1}^{m_i} (x_i \vee \bigvee_{\ell=1}^{m_{ij}} \neg x[i,j,\ell])).$$

The formula Φ_P is a conjunction of the formulas written above (of the formulas given in the parentheses, to be precise). Clearly, Φ_P is a 2-normalized formula. We will show that $comp(P)$ has a model of size k (or equivalently, that P has a supported model of size k) if and only if Φ_P has a model of size $2k$.

Let $M = \{x_{p_1}, \ldots, x_{p_k}\}$ be a model of $comp(P)$. Then, for each $i = p_1, \ldots, p_k$, there is j, $1 \leq j \leq m_i$, such that M is a model of $\bigwedge_{\ell=1}^{m_{ij}} x[i,j,\ell]$ (this is because M is a model of every formula Φ_i). We denote one such j (an arbitrary one) by j_i. We claim that

$$M' = M \cup \{u[i, j_i] : \ i = p_1, \ldots, p_k\}$$

is a model of Φ_P. Clearly, G_i is true in M' for every i, $1 \leq i \leq n$. If $x_i \notin M$ then $u[i,j] \notin M'$ for all $j = 1, \ldots, m_i$. Thus, G'_i is satisfied by M'. Since for each i, $1 \leq i \leq n$, there is at most one j such that $u[i,j] \in M'$, it follows that every formula H_i is true in M'. By the definition of j_i, if $u[i,j] \in M'$ then $j = j_i$ and M' is a model of $\bigwedge_{\ell=1}^{m_{ij}} x[i,j,\ell]$. Hence, I_i is satisfied by M'. Finally, all formulas J_i, $1 \leq i \leq n$, are clearly true in M'. Thus, M' is a model of Φ_P of size $2k$.

Conversely, let M' be a model of Φ_P such that $|M'| = 2k$. Let us assume that M' contains exactly s atoms $u[i,j]$. The clauses H_i ensure that for each i, M' contains at most one atom $u[i,j]$. Therefore, the set $M' \cap \{u[i,j] : i = 1, \ldots, n \ j = 1, \ldots, m_i\}$ is of the form $\{u[p_1, j_{p_1}], \ldots, u[p_s, j_{p_s}]\}$ where $p_1 < \ldots < p_s$.

Since the conjunction of G_i and G'_i is equivalent to $x_i \Leftrightarrow \bigvee_{j=1}^{m_i} u[i,j]$, it follows that exactly s atoms x_i belong to M'. Thus, $|M'| = 2s = 2k$ and $s = k$. It is now easy to see that M' is of the form $\{x_{p_1}, \ldots, x_{p_k}, u[p_1, j_{p_1}], \ldots, u[p_k, j_{p_k}]\}$.

We will now prove that for every i, $1 \leq i \leq n$, the implication

$$x_i \Rightarrow \bigvee_{j=1}^{m_i} \bigwedge_{\ell=1}^{m_{ij}} x[i,j,\ell]$$

is true in M'. To this end, let us assume that x_i is true in M'. Then, there is j, $1 \leq j \leq m_i$, such that $u[i,j] \in M'$ (in fact, $i = p_t$ and $j = j_{p_t}$, for some t, $1 \leq t \leq k$). Since the formula I_i is true in M', the formula $\bigwedge_{\ell=1}^{m_{ij}} x[i,j,\ell]$ is true in M'. Thus, the formula $\bigvee_{j=1}^{m_i} \bigwedge_{\ell=1}^{m_{ij}} x[i,j,\ell]$ is true in M', too.

Since for every i, $1 \leq i \leq n$, the formula J_i is true in M', it follows that all formulas Φ_i are true in M'. Since the only atoms of M' that appear in the formulas Φ_i are the atoms $x_{p_1} \ldots, x_{p_k}$, it follows that $M = \{x_{p_1} \ldots, x_{p_k}\}$ is a model of $comp(P) = \Phi_1 \wedge \ldots \wedge \Phi_n$.

Thus, the problem $\mathcal{SP}_=(\mathcal{A})$ can be reduced to the problem $\mathcal{M}_=(2N)$, which completes the proof. □

For the problem $\mathcal{SP}_=(\mathcal{A})$ we also established the hardness result — we proved that it is W[2]-hard (we omit the proof due to space restrictions). Thus, we found the exact location of this problem in the W-hierarchy. For the problem $\mathcal{ST}'_\leq(\mathcal{A})$, that we are about to consider now, we only succeeded in establishing the lower bound on its complexity. We proved it to be W[3]-hard. We did not succeed in obtaining any non-trivial upper estimate on its complexity.

Theorem 5. *The problem* $\mathcal{ST}'_{\leq}(\mathcal{A})$ *is* W[3]*-hard.*

Proof: We will reduce the problem $\mathcal{M}'_{\leq}(3N)$ to the problem $\mathcal{ST}'_{\leq}(\mathcal{A})$. Let

$$\Phi = \bigwedge_{i=1}^{m} \bigvee_{j=1}^{m_i} \bigwedge_{\ell=1}^{m_{ij}} x[i,j,\ell]$$

be a 3-normalized formula, where $x[i,j,\ell]$ are literals. Let $u[1],\ldots,u[m],v[1],\ldots,$ $v[2k+1]$ be new atoms not occurring in Φ. For each atom $x \in At(\Phi)$, we introduce new atoms $x[s]$, $s = 1,\ldots,k$.

Let P_Φ be a logic program with the following rules:

$$A(x,y,s) = x[s] \leftarrow \mathbf{not}(y[s]), \quad x,y \in At(\Phi),\ x \neq y,\ s = 1,\ldots,k,$$

$$B(x) = x \leftarrow x[1],x[2],\ldots,x[k], \quad x \in At(\Phi),$$

$$C(i,j) = u[i] \leftarrow x'[i,j,1],x'[i,j,2],\ldots,x'[i,j,m_{ij}], \quad i = 1,\ldots,m,\ j = 1,\ldots,m_i,$$

where

$$x'[i,j,\ell] = \begin{cases} x & \text{if } x[i,j,\ell] = x \\ \mathbf{not}(x) & \text{if } x[i,j,\ell] = \neg x, \end{cases}$$

$$D(q) = v[q] \leftarrow u[1],u[2],\ldots,u[m], \quad q = 1,\ldots,2k+1.$$

Clearly, $|At(P_\Phi)| = nk + n + m + 2k + 1$, where $n = |At(\Phi)|$. We will show that Φ has a model of cardinality at least $n-k$ if and only if P_Φ has a stable model of cardinality at least $|At(P_\Phi)| - 2k = n(k+1) + m + 1$.

Let $M = At(\Phi) \setminus \{x_1,\ldots,x_k\}$ be a model of Φ, where $x_1,\ldots,x_k$ are some atoms from $At(\Phi)$ that are not necessarily distinct. We claim that $M' = At(P_\Phi) \setminus$ $\{x_1,\ldots,x_k,x_1[1],\ldots,x_k[k]\}$ is a stable model of P_Φ.

Let us notice that a rule $A(x,y,s)$ is not blocked by M' if and only if $y = x_s$. Hence, the program $P_\Phi^{M'}$ consists of the rules:

$$x[1] \leftarrow \quad ,\ \text{for } x \neq x_1,$$

$$x[2] \leftarrow \quad ,\ \text{for } x \neq x_2$$

$$\ldots$$

$$x[k] \leftarrow \quad ,\ \text{for } x \neq x_k$$

$$x \leftarrow x[1],x[2],\ldots,x[k], \quad x \in At(\Phi)$$

$$v[q] \leftarrow u[1],u[2],\ldots,u[m], \quad q = 1,\ldots,2k+1,$$

and of some of the rules with heads $u[i]$. Let us suppose that every rule of P_Φ with head $u[i]$ contains either a negated atom $x \in M$ or a non-negated atom $x \notin M$. Then, for every $j = 1,\ldots,m_i$ there exists ℓ, $1 \leq \ell \leq m_{ij}$ such that either $x[i,j,\ell] = \neg x$ and $x \in M$, or $x[i,j,\ell] = x$ and $x \notin M$. Thus, M is not a model of the formula $\bigvee_{j=1}^{m_i} \bigwedge_{\ell=1}^{m_{ij}} x[i,j,\ell]$ and, consequently, M is not a model of Φ, a contradiction. Hence, for every $i = 1, \ldots, m$, there is a rule with head $u[i]$ containing neither a negated atom $x \in M$ nor a non-negated atom $x \notin M$. These rules also contribute to the reduct $P_\Phi^{M'}$.

All atoms $x[s] \neq x_1[1], x_2[2], \ldots, x_k[k]$ are facts in $P_\Phi^{M'}$. Thus, they belong to $lm(P_\Phi^{M'})$. Conversely, if $x[s] \in lm(P_\Phi^{M'})$ then $x[s] \neq x_1[1], x_2[2], \ldots, x_k[k]$. Moreover, it is evident by rules $B(x)$ that $x \in lm(P_\Phi^{M'})$ if and only if $x \neq x_1, x_2, \ldots, x_k$. Hence, by the observations in the previous paragraph, $u[i] \in lm(P_\Phi^{M'})$, for each $i = 1, \ldots, m$. Finally, $v[q] \in lm(P_\Phi^{M'})$, $q = 1, \ldots 2k+1$, because the rules $D(q)$ belong to the reduct $P_\Phi^{M'}$. Hence, $M' = lm(P_\Phi^{M'})$ so M' is a stable model of P_Φ and its cardinality is at least $n(k+1)+m+1$.

Conversely, let M' be a stable model of P_Φ of size at least $|At(P_\Phi)| - 2k$. Clearly all atoms $v[q]$, $q = 1, \ldots, 2k+1$, must be members of M' and, consequently, $u[i] \in M'$, for $i = 1, \ldots, m$. Hence, for each $i = 1, \ldots, m$, there is a rule in P_Φ

$$u[i] \leftarrow x'[i,j,1], x'[i,j,2], \ldots, x'[i,j,m_{ij}]$$

such that $x'[i,j,\ell] \in M'$ if $x'[i,j,\ell] = x$, and $x'[i,j,\ell] \notin M'$ if $x'[i,j,\ell] = \neg x$. Thus, M' is a model of the formula $\bigvee_{j=1}^{m_i} \bigwedge_{\ell=1}^{m_{ij}} x[i,j,\ell]$, for each $i = 1, \ldots, m$. Therefore $M = M' \cap At(\Phi)$ is a model of Φ.

It is a routine task to check that rules $A(x,y,s)$ and $B(x)$ imply that all stable models of P_Φ are of the form

$$At(P_\Phi) \setminus \{x_1, x_2, \ldots, x_k, x_1[1], x_2[2], \ldots, x_k[k]\}$$

($x_1, x_2, \ldots, x_k$ are not necessarily distinct). Hence, $|M| = |M' \cap At(\Phi)| \geq n - k$. We have reduced the problem $\mathcal{M}'_{\leq}(3N)$ to the problem $\mathcal{ST}'_{\leq}(\mathcal{A})$. Thus, the assertion follows by Theorem 1. □

References

[Cla78] K.L. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and data bases*, pages 293–322. Plenum Press, New York-London, 1978.

[DF97] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer-Verlag, 1997.

[Fag94] F. Fages. Consistency of Clark's completion and existence of stable models. *Journal of Methods of Logic in Computer Science*, 1:51–60, 1994.

[GL88] M. Gelfond and V. Lifschitz. The stable semantics for logic programs. In R. Kowalski and K. Bowen, editors, *Proceedings of the 5th International Conference on Logic Programming*, pages 1070–1080. MIT Press, 1988.

[GSS99] G. Gottlob, F. Scarcello, and M. Sideri. Fixed parameter complexity in AI and nonmonotonic reasoning. In M. Gelfond, N. Leone, and G. Pfeifer, editors, *Logic Programming and Nonmonotonic Reasoning, Proceedings of the 5th International Conference, LPNMR99*, volume 1730 of *Lecture Notes in Computer Science.* Springer-Verlag, 1999.

[Tru01] M. Truszczyński. Computing large and small stable models. *Theory and Practice of Logic Programming*, 2001. To appear.