Fixed-parameter complexity of semantics for logic programs

ZBIGNIEW LONC
Technical University of Warsaw
and
MIROSŁAW TRUSZCZYŃSKI
University of Kentucky

A decision problem is called *parameterized* if its input is a pair of strings. One of these strings is referred to as a *parameter*. The problem: given a propositional logic program P and a non-negative integer k, decide whether P has a stable model of size no more than k, is an example of a parameterized decision problem with k serving as a parameter. Parameterized problems that are NP-complete often become solvable in polynomial time if the parameter is fixed. The problem to decide whether a program P has a stable model of size no more than k, where k is fixed and not a part of input, can be solved in time $O(mn^k)$, where m is the size of P and n is the number of atoms in P. Thus, this problem is in the class P. However, algorithms with the running time given by a polynomial of order k are not satisfactory even for relatively small values of k.

The key question then is whether significantly better algorithms (with the degree of the polynomial not dependent on k) exist. To tackle it, we use the framework of fixed-parameter complexity. 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 definite 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.

Categories and Subject Descriptors: D.1.6 [**Programming Techniques**]: Logic Programming; F.1.3 [**Complexity Measures and Classes**]: Complexity hierarchies

General Terms: Logic programming, Complexity

Additional Key Words and Phrases: Normal logic programs, Stable models, Supported models, Fixed-parameter complexity

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

Author's address: Z. Lonc, Institute of Mathematics, Technical University of Warsaw, 00-661 Warsaw, Poland. M. Truszczyński, Department of Computer Science, University of Kentucky, Lexington, KY 40506-0046.
This paper was written while the first author was a visitng professor at the University of Kentucky.

investigations, we use the framework of the *fixed-parameter complexity* introduced by Downey and Fellows [Downey and Fellows 1997]. This framework was previously used to study the problem of the existence of stable models of logic programs in [Truszczyński 2002]. 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 definite 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 called *parameterized* if its inputs are *pairs* of items. The second item in a pair is referred to as a *parameter*. The problem to decide, given a logic program P and an integer k, whether P has a stable model with *at most* k atoms is an example of a parameterized decision problem. This parameterized problem is NP-complete. However, fixing k (in other words, k is no longer regarded as a part of the input) makes the problems simpler. It becomes solvable in polynomial time. The following straightforward algorithm works: for every subset $M \subseteq At(P)$ of cardinality at most k, check whether M is a stable model of P. The check can be implemented to run in linear time in the size of the program. If n stands for the number of atoms in P, there are $O(n^k)$ sets to be tested. Thus, the overall running time of this algorithm is $O(mn^k)$, where m is the size of the input program P. This discussion also applies to analogous problems in logic programming concerned with the existence of models and supported models.

respectively) problem is as, in solvable of is

Unfortunately, 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" ($\frac{k}{|At(P)|}$ converges to 0 as $|At(P)|$ goes to infinity). 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 thought of as "large" ($\frac{|At(P)|-k}{|At(P)|}$ converges to 1 as $|At(P)|$ goes to infinity).

We address these questions using the framework of fixed-parameter complexity [Downey and Fellows 1997]. 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 1.1. 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 [Downey and Fellows 1997] defined a hierarchy of complexity classes called the *W hierarchy*:

$$\text{FPT} \subseteq \text{W}[1] \subseteq \text{W}[2] \subseteq \text{W}[3] \subseteq \dots . \tag{1}$$

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 of the form of 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 definite 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 definite 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 I - III.

	$\mathcal{H}$	$\mathcal{N}$	$\mathcal{A}$
$\mathcal{M}$	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

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

In Table I, we list the complexities of all problems in which Δ = "$\geq$". 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 I, 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 I 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 I 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. Indeed, for a fixed k and the number of atoms in a program going to infinity, the ratio of the number of true atoms to the number of all atoms converges to 0 (k is "small" with respect to $|At(P)|$). The fixed-parameter complexities of these problems are summarized in Table II.

	$\mathcal{H}_{\leq}$	$\mathcal{H}_{=}$	$\mathcal{N}_{\leq}$	$\mathcal{N}_{=}$	$\mathcal{A}_{\leq}$	$\mathcal{A}_{=}$
$\mathcal{M}$	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 II. 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 definite Horn logic program P has a stable model of size exactly k is in P. However, perhaps somewhat surprisingly, the remaining two problems involving definite 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. Indeed, for a fixed k and the number of atoms in a program going to infinity, the ratio of the number of true atoms to the number of all atoms converges to 1 (k is "large" with respect to $|At(P)|$). The fixed-parameter complexities of these problems are summarized in Table III.

	$\mathcal{H}_{\leq}$	$\mathcal{H}_{=}$	$\mathcal{N}_{\leq}$	$\mathcal{N}_{=}$	$\mathcal{A}_{\leq}$	$\mathcal{A}_{=}$
$\mathcal{M}'$	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 III. 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 definite 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 [Truszczyński 2002] and [Gottlob et al. 1999]. 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 [Downey and Fellows 1997]. 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 definite 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. Next, we present several useful fixed-parameter complexity results for problems of the existence of models for propositional theories of certain special types. We also state and prove there some auxiliary results on the hardness of problems concerning the existence of stable and supported models. We study the complexity of the problems $\mathcal{D}_\geq(\mathcal{C})$ and $\mathcal{D}'_\geq(\mathcal{C})$ in Section 3. We consider the complexity of problems concerning small and large stable models in Sections 4 and 5, respectively.

2. PRELIMINARIES

We start by introducing some basic logic programming terminology. We refer the reader to [Lloyd 1984; Apt 1990] for a detailed treatment of the subject.

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 $b(r)$. In addition, 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 *definite Horn* program. If every clause in P has an empty positive body, we call P a *purely negative* program. In this paper we will consider finite programs only.

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 definite 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 [Clark 1978], for every atom $p \in At(P)$ we define a propositional formula $comp(p)$ by

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

where

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

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. Thus, in this case, $comp(p)$ is logically equivalent to $\neg p$. On the other hand, if $b(r) = \emptyset$ then $c(r)$ is an empty conjunction, which we interpret as a tautology. In this case $comp(p)$ is logically equivalent to p.

We define the *program completion* (also referred to as the *Clark completion*) of P as the propositional theory

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

A set of atoms $M \subseteq At(P)$ is called 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* (also referred to as the *Gelfond-Lifschitz 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 definite 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 [Gelfond and Lifschitz 1988].

It follows directly from the definition that if M is a stable model of a program P then $M \subseteq At(P)$ and M is a model of P. In fact, an even stronger property holds. It is well known that every stable model of a program P is not only a model of P — it 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 [Fages 1994].

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 three theorems establish several complexity results that we will use later in the paper.

THEOREM 2.1. *The problems $\mathcal{M}_=(2N)$, $\mathcal{M}_=(2NM)$, $\mathcal{M}_\leq(2NM)$ and $\mathcal{M}'_=(2N)$ are all* W[2]*-complete.*

Proof: The first two statements, concerning the W[2]-completeness of $\mathcal{M}_=(2N)$ and $\mathcal{M}_=(2NM)$, are proved in [Downey and Fellows 1997].

To prove the next statement, we will show that the problem $\mathcal{M}_\leq(2NM)$ is equivalent to the problem $\mathcal{M}_=(2NM)$. To this end, we first describe a reduction of $\mathcal{M}_=(2NM)$ to $\mathcal{M}_\leq(2NM)$. Let us consider a monotone 2-normalized formula Φ and an integer k. We define $k' = k$. If $k \leq |At(\Phi)|$, we define $\Phi' = \Phi$. Otherwise, $\Phi' = a_1 \wedge \ldots \wedge a_{k+1}$, where $a_i, i = 1, \ldots, k+1$, are pairwise different atoms.

It is easy to see that Φ has a model with exactly k atoms if and only if Φ' has a model with at most k' atoms. Indeed, let M be a model of Φ with k atoms. Since $M \subseteq At(\Phi)$, $k \leq |At(\Phi)|$. Thus, $\Phi' = \Phi$. Consequently, M is a model of Φ' and $|M| \leq k'$.

Conversely, let us consider a model M of Φ' such that $|M| \leq k'$. If $k' = k > |At(\Phi)|$ then $\Phi' = a_1 \wedge \ldots \wedge a_{k+1}$. The only model of Φ' has $k+1 = k'+1$ atoms, a contradiction with $|M| \leq k'$. Thus, $k' = k \leq |At(\Phi)|$ and we have $\Phi' = \Phi$. It follows that there is a set $M' \subseteq At(\Phi)$ such that $M \subseteq M'$ and $|M'| = k$. Since Φ is a monotone 2-normalized formula, a superset of a model of Φ is also a model of Φ. In particular, M' is a model of Φ and it has exactly k elements.

Given a pair (Φ, k), the pair (Φ', k') can clearly be constructed in time bounded by a polynomial in the size of Φ. Thus, all the requirements of the Definition 1.1 are satisfied. Since Φ' is a monotone 2-normalized formula, the problem $\mathcal{M}_=(2NM)$ is reducible to the problem $\mathcal{M}_\leq(2NM)$.

The converse reduction can be constructed in a similar way. We define $k' = k$. If $k \leq |At(\Phi)|$, we define $\Phi' = \Phi$. Otherwise, $\Phi' = a_1 \wedge \ldots \wedge a_{k+1}$, where $a_i, i = 1, \ldots, k+1$, are pairwise different atoms. It is easy to see that Φ has a model with at most k atoms if and only if Φ' has a model with exactly k' atoms (a similar argument as before can be applied). Clearly, the pair (Φ', k') can be constructed in time polynomial in the size of Φ. Thus, the problem $\mathcal{M}_\leq(2NM)$ is reducible to the problem $\mathcal{M}_=(2NM)$.

It follows that the problem $\mathcal{M}_\leq(2NM)$ is equivalent to the problem $\mathcal{M}_=(2NM)$ which, as we already stated, is known to be W[2]-complete [Downey and Fellows

1997]. Consequently, the problem $\mathcal{M}_{\leq}(2NM)$ is W[2]-complete.

To prove the last statement of the theorem we reduce $\mathcal{M}_{=}(2N)$ to $\mathcal{M}'_{=}(2N)$ and conversely. Let us consider a 2-normalized formula $\Phi = \bigwedge_{i=1}^{m} \bigvee_{j=1}^{m_i} x[i,j]$, where $x[i,j]$ are literals. We observe that Φ has a model of cardinality k if and only if a related formula $\bar{\Phi} = \bigwedge_{i=1}^{m} \bigvee_{j=1}^{m_i} \bar{x}[i,j]$, 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 $|At(\bar{\Phi})| - k$. This construction defines a reduction of $\mathcal{M}_{=}(2N)$ to $\mathcal{M}'_{=}(2N)$. It is easy to see that this reduction satisfies all the requirements of the definition of fixed-parameter reducibility.

A reduction of $\mathcal{M}'_{=}(2N)$ to $\mathcal{M}_{=}(2N)$ can be constructed in a similar way. Since the problem $\mathcal{M}_{=}(2N)$ is W[2]-complete, so is the problem $\mathcal{M}'_{=}(2N)$. □

In the proof of Theorem 2.1, we presented several reductions and observed that they satisfy all the requirements specified in Definition 1.1 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.

THEOREM 2.2. *The problems $\mathcal{M}_{=}(2N_3)$, $\mathcal{M}_{=}(2NA)$, $\mathcal{M}'_{=}(2NM)$ and $\mathcal{M}'_{\geq}(2NM)$ are* W[1]*-complete.*

Proof: The assertions concerning the first two problems are proved in [Downey and Fellows 1997].

Using the reductions described in the proof of the last statement of Theorem 2.1, it is easy to show that the problems $\mathcal{M}'_{=}(2NM)$ and $\mathcal{M}_{=}(2NA)$ are equivalent. Thus, the problem $\mathcal{M}'_{=}(2NM)$ is W[1]-complete.

Let Φ be a monotone 2-normalized theory. Clearly, Φ has a model of size at most $|At(\Phi)| - k$ if and only if it has a model of size exactly $|At(\Phi)| - k$. Thus, the problem $\mathcal{M}'_{\geq}(2NM)$ is equivalent to the problem $\mathcal{M}'_{=}(2NM)$. We have just proved that this last problem is W[1]-complete. Thus, the problem $\mathcal{M}'_{\geq}(2NM)$ is also W[1]-complete. □

THEOREM 2.3. *The problems $\mathcal{M}'_{=}(3NM)$ and $\mathcal{M}'_{\leq}(3N)$ are W[3]-complete.*

Proof: The problems $\mathcal{M}_{=}(3NA)$ and $\mathcal{M}_{\leq}(3N)$ are W[3]-complete [Downey and Fellows 1997]. Let us now observe that the problems $\mathcal{M}'_{=}(3NM)$ and $\mathcal{M}_{=}(3NA)$ are equivalent. Similarly, the problems $\mathcal{M}'_{\leq}(3N)$ and $\mathcal{M}_{\leq}(3N)$ are equivalent. Both equivalences can be argued in a similar way to that we used in the proof of the last statement of Theorem 2.1. Thus, the theorem follows. □

We will now present some general results that imply that in many cases, problems with $\Delta =$ "$\leq$", concerning stable and supported models, are not harder than the corresponding problems with $\Delta =$ "$=$".

For every integer k, $1 \leq k$, we denote by Y_k the set of propositional variables $y_{i,j}$, where $i = 1, 2 \ldots, k+1$, and $j = 1, 2, \ldots, i$. Next, for each i and j, where $1 \leq i \leq k+1$ and $1 \leq j \leq i$, we define a logic program clause $q_{i,j}$ by:

$$y_{i,j} \leftarrow \mathbf{not}(y_{1,1}), \ldots, \mathbf{not}(y_{i-1,1}), \mathbf{not}(y_{i+1,1}), \ldots, \mathbf{not}(y_{k+1,1})$$

(let us note that for every i, $1 \leq i \leq k+1$, rules $q_{i,j}$, $1 \leq j \leq i$, have the same body). We then define a logic program Q_k by setting

$$Q_k = \{q_{i,j} : 1 \leq i \leq k+1 \text{ and } 1 \leq j \leq i\}.$$

LEMMA 2.4. *For every i, $1 \leq i \leq k+1$, the set $\{y_{i,1}, y_{i,2}, \ldots, y_{i,i}\}$ is a stable model (supported model) of Q_k. Moreover, Q_k has no other stable models (supported models).*

Proof: Let us consider any integer i such that $1 \leq i \leq k+1$. We define $M = \{y_{i,1}, y_{i,2}, \ldots, y_{i,i}\}$. Since $y_{i,1}$ appears negated in the body of every rule $q_{i',j}$ of Q_k, with $i' \neq i$ and $1 \leq j \leq i'$, none of these rules contributes to the Gelfond-Lifschitz reduct of Q_k with respect to M. On the other hand, no atom of M appears negated in the bodies of the rules $q_{i,j}$, $1 \leq j \leq i$. Thus, the Gelfond-Lifschitz reduct of Q_k with respect to M consists of the rules

$$y_{i,j} \leftarrow$$

for $j = 1, 2, \ldots, i$. Clearly, the least model of the reduct is M and, consequently, M is a stable model of Q_k.

Let us consider now an arbitrary stable model M of Q_k. Since Q_k has nonempty stable models and since stable models are incomparable with respect to inclusion [Marek and Truszczyński 1993], $M \neq \emptyset$. Let $y_{i,j} \in M$, for some i and j such that $1 \leq i \leq k+1$ and $1 \leq j \leq i$. Since $q_{i,j}$ is the only rule of Q_k with the head $y_{i,j}$, it follows that its body is satisfied by M. Since all rules $q_{i,j}$, $1 \leq j \leq i$, have the same body and since M is a model of Q_k, the heads of all these rules belong to M. Thus, $\{y_{i,1}, y_{i,2}, \ldots, y_{i,i}\} \subseteq M$. We proved earlier that $\{y_{i,1}, y_{i,2}, \ldots, y_{i,i}\}$ is a stable model of Q_k. Since stable models are incomparable with respect to inclusion, $M = \{y_{i,1}, y_{i,2}, \ldots, y_{i,i}\}$. This completes the proof of the assertion for the case of stable models.

The program Q_k is purely negative. Thus, its stable and supported models coincide [Fages 1994]. Consequently, the assertion follows for the case of supported models, as well. □

THEOREM 2.5. *Let P be a logic program and let k be a non-negative integer. Let $Y_k = \{y_{i,j} : i = 1, 2 \ldots, k+1,\ j = 1, 2, \ldots, i\}$ be a set of atoms disjoint with $At(P)$ and let Q_k be the program constructed above. Then:*

(1) *P has a supported model (stable model) of cardinality at most k if and only if $P \cup Q_k$ has a supported model (stable model) of cardinality equal to $k+1$.*
(2) *P has a supported model (stable model) of cardinality at least $|At(P)| - k$ if and only if $P \cup Q_k$ has a supported model (stable model) of cardinality equal to $|At(P \cup Q_k)| - k(k+3)/2$.*

Proof: First, we observe that since $Y_k \cap At(P) = \emptyset$, supported models (stable models) of $P \cup Q_k$ are precisely the sets $M' \cup M''$, where M' is a supported model (stable model) of P and M'' is a supported model (stable model) of Q_k.

The proofs for parts (1) and (2) of the assertion are very similar. We provide here only the proof for part (2).

Let us assume that M is a supported model of P of cardinality at least $|At(P)| - k$. Then, $|M| = |At(P)| - k + a$, for some a, $0 \leq a \leq k$. Clearly, $i = (k+1) - a$ satisfies $1 \leq i \leq k+1$ and $\{y_{i,1}, y_{i,2}, \ldots, y_{i,i}\}$ is a supported model of Q_k. It follows that

$M' = M \cup \{y_{i,1}, y_{i,2}, \ldots, y_{i,i}\}$ is a supported model of $P \cup Q_k$ and its cardinality is $|At(P)| - k + a + i$. It is now easy to see that

$$|At(Q_k)| = (k+1)(k+2)/2.$$

Thus, we have that

$$\begin{aligned} |M'| &= |At(P)| - k + a + i = |At(P)| + 1 \\ &= |At(P \cup Q_k)| - (k+1)(k+2)/2 + 1 = |At(P \cup Q_k)| - k(k+3)/2. \end{aligned}$$

Conversely, let us assume that M' is a supported model of $P \cup Q_k$ of cardinality exactly $|At(P \cup Q_k)| - k(k+3)/2$. It follows that $M' = M \cup \{y_{i,1}, y_{i,2}, \ldots, y_{i,i}\}$, where M is a supported model of P and $1 \leq i \leq k+1$. Clearly,

$$\begin{aligned} |M| &= |M'| - i = |At(P \cup Q_k)| - k(k+3)/2 - i \\ &= |At(P)| + (k+1)(k+2)/2 - k(k+3)/2 - i \geq |At(P)| - k. \end{aligned}$$

This completes the argument for part (2) of the assertion for the case of supported models. The same reasoning works also for the case of stable models because all auxiliary facts used in this reasoning hold for stable models, too. □

The program Q_k can be constructed in time bounded by a polynomial in the size of P and k. Thus, Theorem 2.5 has the following corollary on the reducibility of some problems $\mathcal{D}_{\leq}(\mathcal{C})$ to the respective problems $\mathcal{D}_{=}(\mathcal{C})$.

COROLLARY 2.6. *For every class of logic programs $\mathcal{C}$ such that $\mathcal{C}$ is closed under unions and $\mathcal{N} \subseteq \mathcal{C}$, problems $\mathcal{SP}_{\leq}(\mathcal{C})$, $\mathcal{ST}_{\leq}(\mathcal{C})$, $\mathcal{SP}'_{\leq}(\mathcal{C})$ and $\mathcal{ST}'_{\leq}(\mathcal{C})$ can be reduced to (are not harder than) problems $\mathcal{SP}_{=}(\mathcal{C})$, $\mathcal{ST}_{=}(\mathcal{C})$, $\mathcal{SP}'_{=}(\mathcal{C})$ and $\mathcal{ST}'_{=}(\mathcal{C})$, respectively.* □

3. THE PROBLEMS $\mathcal{D}_{\geq}(\mathcal{C})$ AND $\mathcal{D}'_{\geq}(\mathcal{C})$

These problems ask about the existence of models with at least k true atoms (in the case of small bound problems) or with at least k false atoms (for the large-bound problems). From the point of view of the fixed-parameter complexity, these problems (with one exception) are not very interesting. Several of them remain NP-complete even if k is fixed (in other words, fixing k does not render them tractable). Others are "easy" — they can be solved in polynomial time even *without* fixing k. The one exception, the problem $\mathcal{M}'_{\geq}(\mathcal{N})$, turns out to be W[1]-complete.

THEOREM 3.1. *The following parameterized problems are in* P*: $\mathcal{M}_{\geq}(\mathcal{H})$, $\mathcal{M}_{\geq}(\mathcal{N})$, $\mathcal{M}_{\geq}(\mathcal{A})$, $\mathcal{SP}_{\geq}(\mathcal{H})$, $\mathcal{ST}_{\geq}(\mathcal{H})$, $\mathcal{M}'_{\geq}(\mathcal{H})$, $\mathcal{SP}'_{\geq}(\mathcal{H})$ and $\mathcal{ST}'_{\geq}(\mathcal{H})$.*

Proof: (1) The problems $\mathcal{M}_{\geq}(\mathcal{H})$, $\mathcal{M}_{\geq}(\mathcal{N})$ and $\mathcal{M}_{\geq}(\mathcal{A})$ are all in P. Indeed, if Q is a logic program, the set of all atoms of Q is a model of Q. Thus, if $|At(Q)| \geq k$, the answer (in each case) is YES. Otherwise, the answer is NO. Clearly, the question whether $|At(Q)| \geq k$ can be decided in polynomial time (in the size of Q and k).
(2) $\mathcal{SP}_{\geq}(\mathcal{H})$ is in P. To see this, we observe that there is a polynomial-time algorithm to compute the greatest supported model of a definite Horn program [Apt and van Emden 1982]. A definite Horn program Q has a supported model of size at least k if and only if the greatest supported model of Q has size at least k. Thus, the assertion follows.

(3) The problem $\mathcal{ST}_{\geq}(\mathcal{H})$ is in P. Indeed, the least model of a definite Horn program Q is the only stable model of Q. The least model of a definite Horn program Q can be computed in linear time [Dowling and Gallier 1984]. So, the assertion follows.
(4) The problems $\mathcal{M}'_{\geq}(\mathcal{H})$, $\mathcal{SP}'_{\geq}(\mathcal{H})$ and $\mathcal{ST}'_{\geq}(\mathcal{H})$ are all in P. Indeed, a definite Horn logic program has the least model which is also the least supported and the only stable model of Q. Thus, in the case of each of these three problems, the answer is YES if and only if the least model of Q has size at most $|At(Q)| - k$. Since the least model of Q can be computed in linear time, the three assertions of (4) follow. □

In contrast to the problems covered by Theorem 3.1, which are solvable in polynomial time even if k is *not* a part of the input, problems in the next group remain hard even if k is fixed.

THEOREM 3.2. *Let k be a fixed non-negative integer. The following* fixed-parameter *problems are* NP*-complete: $\mathcal{SP}_{\geq}(\mathcal{N},k)$, $\mathcal{SP}_{\geq}(\mathcal{A},k)$, $\mathcal{ST}_{\geq}(\mathcal{N},k)$, $\mathcal{ST}_{\geq}(\mathcal{A},k)$, $\mathcal{SP}'_{\geq}(\mathcal{N},k)$, $\mathcal{SP}'_{\geq}(\mathcal{A},k)$, $\mathcal{ST}'_{\geq}(\mathcal{N},k)$ and $\mathcal{ST}'_{\geq}(\mathcal{A},k)$.*

Proof: (1) The problems $\mathcal{SP}_{\geq}(\mathcal{N},k)$, $\mathcal{SP}_{\geq}(\mathcal{A},k)$, $\mathcal{ST}_{\geq}(\mathcal{N},k)$ and $\mathcal{ST}_{\geq}(\mathcal{A},k)$ are all NP-complete. Clearly, all these problems are in NP. To prove their NP-hardness, we recall that the problems to decide whether a logic program has a supported (stable) model are NP-complete, even under the restriction to purely negative programs [Marek and Truszczyński 1991]. Let P be a logic program. Let y_i, $i = 1, 2, \ldots, k$, be atoms not appearing in P. We define

$$P' = P \cup \{y_i \leftarrow \,:\, i = 1, 2, \ldots, k\}.$$

Since $At(P) \cap \{y_1, y_2, \ldots, y_k\} = \emptyset$, P has a stable (supported) model if and only if P' has a stable (supported) model of size at least k. Moreover, if $P \in \mathcal{N}$, then $P' \in \mathcal{N}$, as well. Thus, NP-hardness of the problems $\mathcal{SP}_{\geq}(\mathcal{N},k)$, $\mathcal{SP}_{\geq}(\mathcal{A},k)$, $\mathcal{ST}_{\geq}(\mathcal{N},k)$ and $\mathcal{ST}_{\geq}(\mathcal{A},k)$ follows.
(2) The problems $\mathcal{SP}'_{\geq}(\mathcal{N},k)$, $\mathcal{SP}'_{\geq}(\mathcal{A},k)$, $\mathcal{ST}'_{\geq}(\mathcal{N},k)$ and $\mathcal{ST}'_{\geq}(\mathcal{A},k)$ are all NP-complete. Clearly, all these problems are in NP. To prove their NP-hardness, we use (as in (1)) the fact that the problems to decide whether a logic program has a supported (stable) model are NP-complete (even under the restriction to purely negative programs). Let P be a logic program and let y_i, z_i, $i = 1, 2, \ldots, k$ be atoms not appearing in P. We define

$$P' = P \cup \{y_i \leftarrow \mathbf{not}(z_i); z_i \leftarrow \mathbf{not}(y_i) \colon i = 1, 2, \ldots, k\}.$$

The logic program $\{y_i \leftarrow \mathbf{not}(z_i); z_i \leftarrow \mathbf{not}(y_i) \colon i = 1, 2, \ldots, k\}$ has 2^k stable models. Each of these models has exactly k elements (for each $i = 1, 2, \ldots, k$, it contains either y_i or z_i but not both). Since $At(P) \cap \{y_i, z_i \colon i = 1, \ldots, k\} = \emptyset$, P has a stable (supported) model if and only if P' has a stable (supported) model of size at most $|At(P')| - k$. Moreover, if $P \in \mathcal{N}$ then $P' \in \mathcal{N}$, as well. Thus, NP-hardness of the $\mathcal{SP}'_{\geq}(\mathcal{N},k)$, $\mathcal{SP}'_{\geq}(\mathcal{A},k)$, $\mathcal{ST}'_{\geq}(\mathcal{N},k)$ and $\mathcal{ST}'_{\geq}(\mathcal{A},k)$ follows. □

We will next study the problem $\mathcal{M}'_{\geq}(\mathcal{A},k)$. It turns out that it is NP-complete for all $k \geq 1$ and is trivially solvable in polynomial time if $k = 0$.

THEOREM 3.3. *The problem $\mathcal{M}'_{\geq}(\mathcal{A},0)$ is in P. For every $k \geq 1$, the problem $\mathcal{M}'_{\geq}(\mathcal{A},k)$ is NP-complete.*

Proof. The first part of the assertion is evident. The answer to the problem $\mathcal{M}'_{\geq}(\mathcal{A},0)$ is always YES. Indeed, for every logic program P, the set $M = At(P)$ is a model of P and it satisfies the inequality $|At(P)| \geq |M|$.

Let us now assume that $k \geq 1$. We will first consider the problem $\mathcal{P}(k)$ to decide whether a 2-normalized (that is, CNF) formula Φ has a model of size at most $|At(\Phi)| - k$ (k is fixed and not a part of the input). This problem is NP-complete. It is clearly in NP. To show its NP-hardness, we will reduce to it the general CNF satisfiability problem. Let Ψ be a CNF theory and let y_i, $1 \leq i \leq k$, be atoms not occurring in Ψ. Then Ψ has a model if and only if $\Psi' = \Psi \cup \{\neg y_i : i = 1, 2, \ldots, k\}$ has a model of size at most $|At(\Psi')| - k$. Hence, NP-completeness of the problem $\mathcal{P}(k)$, where $k \geq 1$, follows.

Problem $\mathcal{M}'_{\geq}(\mathcal{A},k)$ is clearly in NP. To prove NP-hardness of $\mathcal{M}'_{\geq}(\mathcal{A},k)$ we will reduce the problem $\mathcal{P}(k)$ to it. Let Φ be a CNF theory. Let us assume that $At(\Phi) = \{x_1, x_2, \ldots, x_n\}$. For each clause $C = a_1 \vee \ldots \vee a_p \vee \neg b_1 \vee \ldots \vee \neg b_r$ of Φ we define program clauses $r_{C,i}$, $1 \leq i \leq n$:

$$r_{C,i} = \quad x_i \leftarrow b_1, \ldots, b_r, \mathbf{not}(a_1), \ldots, \mathbf{not}(a_p).$$

Let $P_\Phi = \{r_{C,i} : C \in \Phi,\ i = 1, \ldots, n\}$. Clearly, $At(P_\Phi) = \{x_1, x_2, \ldots, x_n\}$ (that is, the formula Φ and the program P_Φ have the same atoms).

Let M be a model of Φ and let C be a clause of Φ. Since M satisfies C, M does not satisfy the body of the rules $r_{C,i}$, $1 \leq i \leq n$. In other words, M satisfies all the rules $r_{C,i}$, $1 \leq i \leq n$. Thus, if M is a model of Φ then M is a model of P_Φ. Since $At(\Phi) = At(P_\Phi)$, it follows that if Φ has a model of size at most $|At(\Phi)| - k$ then the program P_Φ has a model of size at most $|At(P_\Phi)| - k$.

Conversely, let us consider a model M of P_Φ such that $|M| \leq n - k$. Since $k \geq 1$, we have $|M| < n$. Let us assume that there is a clause C of Φ that is not satisfied by M. Then, the bodies of all program clauses $r_{C,i}$, $1 \leq i \leq n$, are satisfied. Hence, $\{x_1, \ldots, x_n\} \subseteq M$ and $|M| \geq |At(P_\Phi)| = n$, a contradiction. It follows that M is a model of Φ.

Thus, indeed, the problem $\mathcal{P}(k)$ can be reduced to the problem $\mathcal{M}'_{\geq}(\mathcal{A},k)$ and NP-hardness of $\mathcal{M}'_{\geq}(\mathcal{A},k)$ follows. □

The only problem with Δ = "$\geq$" whose complexity is affected by fixing k is $\mathcal{M}'_{\geq}(\mathcal{N})$. fixed-parameter polynomial time. Namely, we have the following result.

THEOREM 3.4. *The problem* $\mathcal{M}'_{\geq}(\mathcal{N})$ *is* W[1]*-complete.*

Proof: Let us consider a monotone 2-normalized formula Φ. In each clause $C = x_1 \vee \ldots \vee x_k$ of Φ we pick an arbitrary atom, say x_1. We then define a logic program clause $r_C = x_1 \leftarrow \mathbf{not}(x_2), \ldots, \mathbf{not}(x_k)$. Finally, we define a logic program $P_\Phi = \{r_C : C \in \Phi\}$. Clearly, P_Φ is a purely negative program, it is built over the same set of atoms as Φ and it has the same models as Φ. Similarly, for every purely negative program P, the 2-normalized formula $pr(P)$ is monotone. Moreover, the set of atoms of $pr(P)$ is the same as that of P, and $pr(P)$ and P have the same models.

It follows that the problem $\mathcal{M}'_{\geq}(\mathcal{N})$ is equivalent to the problem $\mathcal{M}'_{\geq}(\mathit{2NM})$). Thus, the assertion follows by Theorem 2.2. □

4. THE CASE OF SMALL MODELS

In this section we deal with the problems $\mathcal{M}_\Delta(\mathcal{C})$, $\mathcal{SP}_\Delta(\mathcal{C})$ and $\mathcal{ST}_\Delta(\mathcal{C})$ for $\Delta =$ "$=$" and $\Delta =$ "$\leq$". Speaking informally, we are interested in the existence of models that are *small*, that is, contain no more than some specified number of atoms. The problem $\mathcal{ST}_\leq(\mathcal{A})$ was first studied in [Truszczyński 2002]. In that work, it was proved that the problem $\mathcal{ST}_\leq(\mathcal{A})$ is W[2]-hard and belongs to the class W[3]. In this section we establish the exact location of the problem $\mathcal{ST}_\leq(\mathcal{A})$ in the W hierarchy and obtain similar results for problems concerning the existence of models and supported models.

THEOREM 4.1. *The problems $\mathcal{M}_\leq(\mathcal{N})$, $\mathcal{M}_=(\mathcal{N})$, $\mathcal{M}_\leq(\mathcal{A})$ and $\mathcal{M}_=(\mathcal{A})$ are all* W[2]*-complete.*

Proof: Since $\mathcal{N} \subseteq \mathcal{A}$, it is enough to prove that the problems $\mathcal{M}_\Delta(\mathcal{N})$, $\Delta =$ "$\leq$" and "$=$", are W[2]-hard, and that the problems $\mathcal{M}_\Delta(\mathcal{A})$, $\Delta =$ "$\leq$" and "$=$", are in W[2].

Reasoning as in the proof of Theorem 3.4, we argue that the problems $\mathcal{M}_\Delta(\mathit{2NM})$ can be reduced to the problems $\mathcal{M}_\Delta(\mathcal{N})$, for $\Delta =$ "$\leq$" and "$=$". Indeed, M is a model of a monotone 2-normalized formula Φ if and only if M is a model of the logic program P_Φ, as defined in the proof of Theorem 3.4. Since Φ is a monotone 2-normalized formula, P_Φ is a purely negative logic program. This establishes the reducibility. By Theorem 2.1, it follows that the problems $\mathcal{M}_\leq(\mathcal{N})$ and $\mathcal{M}_=(\mathcal{N})$ are W[2]-hard.

Since M is a model of a logic program P if and only if M is a model of $pr(P)$, it follows that the problems $\mathcal{M}_\leq(\mathcal{A})$ and $\mathcal{M}_=(\mathcal{A})$ can be reduced to the problems $\mathcal{M}_\leq(\mathit{2N})$ and $\mathcal{M}_=(\mathit{2N})$, respectively. Hence, by Theorem 2.1, the problems $\mathcal{M}_\leq(\mathcal{A})$ and $\mathcal{M}_=(\mathcal{A})$ are in W[2]. □

THEOREM 4.2. *The problems $\mathcal{M}_\leq(\mathcal{H})$, $\mathcal{SP}_\leq(\mathcal{H})$, $\mathcal{ST}_\leq(\mathcal{H})$ and $\mathcal{ST}_=(\mathcal{H})$ are in* P.

Proof: A definite Horn logic program has a model (supported model, stable model) of size at most k if and only if its least model (which is also the least supported model and the only stable model) has size at most k. The least model of a definite Horn program can be computed in linear time. Thus, the problems $\mathcal{M}_\leq(\mathcal{H})$, $\mathcal{SP}_\leq(\mathcal{H})$ and $\mathcal{ST}_\leq(\mathcal{H})$ are in P. Since the least model of a definite Horn program is the *unique* stable model of the program, it follows that also the problem $\mathcal{ST}_=(\mathcal{H})$ is in P. □

We emphasize that k is a part of the input for problems dealt with in Theorem 4.2. Thus, all these problems are solvable in polynomial time even without fixing k.

THEOREM 4.3. *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 2.2.

To prove that the problem $\mathcal{M}_=(\mathcal{H})$ is in the class W[1], we will reduce it to the problem $\mathcal{M}_=(2N_3)$. To this end, for every definite 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 definite 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. Let P be such a program and let k be a non-negative integer. If $k = 0$, we define P' to consist of all the facts in P. Otherwise, we define P' to be the program obtained from P by removing all clauses with bodies consisting of more than k atoms and adding clauses of the form $a \leftarrow a$, for $a \in At(P)$. It is evident that P has a model of size k if and only if P' has a model of size k. It is also clear that the body of every rule of P' consists of no more than k atoms. Finally, the program P' can be constructed in time linear in the size of P.

Thus, we will describe the construction of the formula Φ_P only for definite 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 $x[i]$, $i = 1,\ldots,2^k$. $i = 1,2,\ldots,2^k$. Finally, we introduce yet another set of new atoms $z[1],\ldots,z[2^k]$.

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

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

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

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

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

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

For each rule r in P we introduce a formula

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

Finally, for each $t = 1, \dots, 2^k$ let

$$H(t) = \quad z[t] \Leftrightarrow z[t] \quad (\text{or } (\neg z[t] \vee z[t])).$$

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 denote by p the number of sets $B \in \mathcal{B}$ such that $B \subseteq M$. We define

$$M' = M \cup \{x[i]: x \in M, i = 1, \dots, 2^k\} \cup \{u[B]: B \subseteq M \text{ and } B \in \mathcal{B}\} \cup \{z[t]: t = 1, \dots, 2^k - p\}.$$

The set M' satisfies all formulas $D(x,i)$, $x \in At(P)$, $i = 1, \dots, 2^k$ and $H(t)$, $t = 1, \dots 2^k$. 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 definite 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 $|M'| = k + k2^k + p + 2^k - p = (k+1)2^k + k$.

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

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

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

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

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

where $i = 1, \dots, p$. 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 $u[\{b_1, \dots, b_p\}]$ is satisfied by M'.

The formula $G(r) = u[\{b_1, \dots, b_p\}] \Rightarrow h$ 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 $x[i]$, $1 \leq i \leq 2^k$. Hence, if $|M| > k$ then $|M'| \geq |M| + |M| \times 2^k \geq (k+1)(1+2^k) > (k+1)2^k + k$, 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, $|M'| < |M| + |M| \times 2^k + 2^k \leq (k-1)(1+2^k) + 2^k < (k+1)2^k + k$, 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 2.2, the problem $\mathcal{M}_=(\mathcal{H})$ is in the class W[1]. This completes our argument. □

THEOREM 4.4. *The problems $\mathcal{ST}_{\leq}(\mathcal{N})$ and $\mathcal{SP}_{\leq}(\mathcal{N})$ are $W[2]$-hard.*

Proof: Since stable and supported models of purely negative programs coincide [Fages 1994], we will show $W[2]$-hardness for stable models only. To this end, we will find a reduction of $\mathcal{M}_{\leq}(2NM)$ (which is $W[2]$-hard, see Theorem 2.1) to $\mathcal{ST}_{\leq}(\mathcal{N})$.

Let Φ be a monotone 2-normalized formula and let $\{x_1, \ldots, x_n\}$ be the set of atoms that occur in Φ. We define a program $P_\Phi \in \mathcal{N}$ as follows. For every atom x_j, $j = 1, \ldots, n$, occurring in Φ we introduce k new atoms $x_j[1], x_j[2], \ldots, x_j[k]$. For each of these atoms we include in P_Φ the following rule:

$$r_{j,\ell} = \quad x_j[\ell] \leftarrow \mathbf{not}(x_1[\ell]), \ldots, \mathbf{not}(x_{j-1}[\ell]), \mathbf{not}(x_{j+1}[\ell]), \ldots, \mathbf{not}(x_n[\ell]),$$

$j = 1, \ldots, n$, $\ell = 1, \ldots, k$. Next, for each clause $C = x_{i_1} \vee \ldots \vee x_{i_s}$ in Φ, we introduce a new atom f_C and include in P_Φ the rule:

$$\begin{aligned} r_C = \quad f_C \leftarrow \quad & \mathbf{not}(x_{i_1}[1]), \ldots, \mathbf{not}(x_{i_1}[k]), \\ & \mathbf{not}(x_{i_2}[1]), \ldots, \mathbf{not}(x_{i_2}[k]), \\ & \ldots, \\ & \mathbf{not}(x_{i_s}[1]), \ldots, \mathbf{not}(x_{i_s}[k]), \mathbf{not}(f_C). \end{aligned}$$

We will show that Φ has a model of cardinality at most k if and only if P_Φ has a stable model of size at most k.

Let $M = \{x_{t_1}, x_{t_2}, \ldots, x_{t_m}\}$, $m \leq k$, be a model of Φ. We claim that

$$M' = \{x_{t_1}[1],\ x_{t_2}[2], \ldots, x_{t_m}[m], x_{t_m}[m+1], \ldots, x_{t_m}[k]\}$$

is a stable model of P_Φ. Let C be a clause from Φ. Since M is a model of Φ, C contains an atom, say x_{t_j}, from M. Then, however, $j \leq m$ and $x_{t_j}[j] \in M'$. The atom $x_{t_j}[j]$ occurs negated in the body of the rule r_C. Thus, the rule r_C does not contribute to the reduct $P_\Phi^{M'}$. In the same time, the rules $r_{j,\ell}$ contribute the following rules to the reduct:

$$x_{t_j}[j] \leftarrow,$$

for $j = 1, \ldots, m$, and

$$x_{t_m}[j] \leftarrow,$$

for $j = m+1, \ldots, k$. Thus, $lm(P_\Phi^{M'}) = M'$ and, consequently, M' is a stable model of P_Φ of size k.

Conversely, let us assume that P_Φ has a stable model M' of size at most k. The atoms f_C cannot be in M' and, if $x_j[\ell] \in M'$, then $x_i[\ell] \notin M'$, for $i \neq j$. Moreover, if for every j, $1 \leq j \leq n$, $x_j[\ell] \notin M'$, then the rule $r_{1,\ell}$ implies that $x_1[\ell] \in M'$, a contradiction. Hence, for every $\ell = 1, \ldots, k$, exactly one of the atoms $x_1[\ell], \ldots, x_n[\ell]$ is in M'. Thus, all stable models of P_Φ are of the form $M' = \{x_{t_1}[1], x_{t_2}[2], \ldots, x_{t_k}[k]\}$, where the indices $t_1, t_2, \ldots, t_k$ are not necessarily pairwise distinct. Let $M = \{x_{t_1}, \ldots, x_{t_k}\}$. Clearly, $|M| \leq k$. Suppose M is not a model of some clause $C = x_{i_1} \vee \ldots \vee x_{i_s}$. Then, none of the atoms $x_{i_1}, \ldots, x_{i_s}$ is in M. Consequently none of the atoms $x_{i_j}[\ell]$, $j = 1, \ldots, s$, $\ell = 1, \ldots, k$, is in M'. It follows that the rule $f_C \leftarrow$ is in the reduct $P_\Phi^{M'}$ and, so, $f_C \in M'$, a contradiction. Thus, M is indeed a model of Φ of cardinality at most k.

This completes the argument that $\mathcal{M}_{\leq}(2NM)$ can be reduced to $\mathcal{ST}_{\leq}(\mathcal{N})$ and the assertion of the theorem follows by Theorem 2.1. □

Later in the paper we will need a stronger version of Theorem 4.4. To state it, we need more terminology. We define $\mathcal{N}_1$ to be the class of purely negative programs such that each atom occurs exactly once in the head of a rule. It is clear that the program P_Φ constructed in the proof of the Theorem 4.4 belongs to the class $\mathcal{N}_1$. Thus, we obtain the following result.

THEOREM 4.5. *The problems $\mathcal{ST}_{\leq}(\mathcal{N}_1)$ and $\mathcal{SP}_{\leq}(\mathcal{N}_1)$ are $W[2]$-hard.* □

THEOREM 4.6. *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 2.1. 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

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

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

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

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

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

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 $M' \cap \{u[i,j] \colon i = 1, \ldots, n \ j = 1, \ldots, m_i\}$ 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' (in other words, that $x_i \in M'$). 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. □

THEOREM 4.7. *The problem* $\mathcal{ST}_=(\mathcal{A})$ *is in* $W[2]$.

Proof: In [Truszczyński 2002], it is shown that the problem $\mathcal{ST}_=(\mathcal{A})$ can be reduced to the problem of existence of a model of size k of a certain formula Φ. This formula Φ is a conjunction of formulas of the form

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

for $i = 1, \ldots, n$, where $\{x_1, \ldots, x_n\}$ is the set of atoms of Φ and $x[i,j,\ell]$ denote some literals over this set of atoms. This theory is the Clark completion of a certain logic program P. Thus, we get a reduction of $\mathcal{ST}_=(\mathcal{A})$ to $\mathcal{SP}_=(\mathcal{A})$. By Theorem 4.6, it follows that $\mathcal{ST}_=(\mathcal{A})$ is in $W[2]$. □

Theorems 4.4, 4.6 and 4.7, and Corollary 2.6 yield the following result.

Corollary 4.8. *The problems $\mathcal{ST}_\leq(\mathcal{N})$, $\mathcal{SP}_\leq(\mathcal{N})$, $\mathcal{ST}_\leq(\mathcal{A})$, $\mathcal{SP}_\leq(\mathcal{A})$, $\mathcal{ST}_=(\mathcal{N})$, $\mathcal{SP}_=(\mathcal{N})$, $\mathcal{ST}_=(\mathcal{A})$ and $\mathcal{SP}_=(\mathcal{A})$ are $W[2]$-complete.* □

Finally, in our last result of this section, we establish bounds on the fixed-parameter complexity of the problem $\mathcal{SP}_=(\mathcal{H})$.

Theorem 4.9. *The problem $\mathcal{SP}_=(\mathcal{H})$ is $W[1]$-hard and belongs to $W[2]$.*

Proof: Since $\mathcal{H}$ is a subclass of $\mathcal{A}$, it follows immediately from Theorem 4.6 that $\mathcal{SP}_=(\mathcal{H})$ is in $W[2]$. The $W[1]$-hardness can be proved in exactly the same way as for the problem $\mathcal{M}_=(\mathcal{H})$ (Theorem 4.3), except that for every atom x of Φ, we have to include the rule $x \leftarrow x$ in P_Φ. □

5. THE CASE OF LARGE MODELS

In this section we deal with the problems $\mathcal{M}'_\Delta(\mathcal{C})$, $\mathcal{SP}'_\Delta(\mathcal{C})$ and $\mathcal{ST}'_\Delta(\mathcal{C})$ for $\Delta =$ "=" and $\Delta =$ "≤". Speaking informally, we are now interested in the existence of models that are *large*, that is, models in which the number of false atoms is bounded from above by some integer.

Theorem 5.1. *The problems $\mathcal{M}'_\leq(\mathcal{H})$, $\mathcal{SP}'_\leq(\mathcal{H})$, $\mathcal{ST}'_\leq(\mathcal{H})$, $\mathcal{M}'_\leq(\mathcal{N})$ and $\mathcal{M}'_\leq(\mathcal{A})$ are in* P.

Proof: The problems $\mathcal{M}'_\leq(\mathcal{C})$, where $\mathcal{C} = \mathcal{H}, \mathcal{N}$ or $\mathcal{A}$, have always the answer YES (the set of all atoms is a model of any logic program). Hence, all these three problems are trivially in P.

Next, we observe that there is a polynomial-time algorithm to compute the greatest supported model of a definite Horn program [Apt and van Emden 1982]. Consequently, the problem $\mathcal{SP}'_\leq(\mathcal{H})$ is in P (there is a supported model in which no more than k atoms are false if and only if no more than k atoms are false in the greatest supported model). Finally, a definite Horn program has a unique stable model (its least model) that can be computed in polynomial time. Hence, the problem $\mathcal{ST}'_\leq(\mathcal{H})$ is also in P. □

Theorem 5.2. *The problem $\mathcal{M}'_=(\mathcal{N})$ is* W[1]*-complete.*

Proof: It is easy to see that this problem is equivalent to the problem $\mathcal{M}'_=(\mathit{2NM})$ (the same reductions as those used in Theorem 3.4 work). This latter problem is W[1]-complete (Theorem 2.2). Hence, the assertion follows. □

THEOREM 5.3. *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 $\mathcal{M}_=(2NM)$ 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 definite 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 2.1. □

THEOREM 5.4. *The problem $\mathcal{SP}'_=(\mathcal{H})$ is $W[3]$-hard.*

Proof: We will reduce the problem $\mathcal{M}'_=(3NM)$ (which is W[3]-complete by Theorem 2.3) to the problem $\mathcal{SP}'_=(\mathcal{H})$. Let

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

be a monotone 3-normalized formula, where $x[i,j,\ell]$ are atoms. Let us assume that $|At(\Phi)| = n$.

We define a definite Horn program P_Φ as follows. Let $u[1], \ldots, u[m], v[1], \ldots, v[k+1]$ be new atoms not occurring in Φ. First, for every $x \in At(\Phi)$, we include in P_Φ the rule

$$x \leftarrow x.$$

Next, for every $i = 1, \ldots, m$, we include in P_Φ m_i rules

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

where $j = 1, \ldots, m_i$. Finally, we include in P_Φ $k+1$ rules

$$v[q] \leftarrow u[1], \ldots, u[m],$$

where $q = 1, \ldots, k+1$.

We will show that Φ has a model of cardinality $n-k$ if and only if the definite Horn program P_Φ has a supported model of cardinality $|At(P_\Phi)|-k = n+m+k+1-k = n+m+1$.

Let M be a model of Φ, $|M| = n-k$. It is easy to see that $M' = M \cup \{u[1],\ldots,u[m],v[1],\ldots,v[k+1]\}$ is a supported model of P_Φ of cardinality $n+m+1$.

Conversely, let M' be a supported model of P_Φ of cardinality $n+m+1$. Clearly M' is a model of the Clark completion $comp(P_\Phi)$ of P_Φ. If $u[i] \notin M'$, for some $i = 1,\ldots,m$, then $v[q] \notin M'$, for every $q = 1,\ldots,k+1$, because $v[q] \Leftrightarrow \bigwedge_{i=1}^{m} u[i]$ belongs to $comp(P_\Phi)$. Hence, $|M'| \leq n+m-1$, a contradiction. Therefore $u[i] \in M'$, for every $i = 1,\ldots,m$. Consequently, for every $q = 1,\ldots,k+1$, we have $v[q] \in M'$. Let $M = M' \cap At(\Phi)$. Clearly, $|M| = n+m+1-m-(k+1) = n-k$. Moreover, M is a model of each formula $\bigvee_{j=1}^{m_i} \bigwedge_{\ell=1}^{m_{ij}} x[i,j,\ell]$, $i = 1,\ldots,m$. Indeed, M' is a model of the formula $u[i] \Leftrightarrow \bigvee_{j=1}^{m_i} \bigwedge_{\ell=1}^{m_{ij}} x[i,j,\ell]$ belonging to $comp(P_\Phi)$ and $u[i] \in M'$, for $i = 1,\ldots,m$. Hence, M is a model of Φ of cardinality $n-k$. □

THEOREM 5.5. *The problem $\mathcal{SP}'_{=}(\mathcal{A})$ is in $W[3]$.*

Proof: Let P be a logic program with atoms $x_1,\ldots,x_n$. Its supported models coincide with models of the Clark completion $comp(P)$ of P. The formulas of the Clark completion are of the form

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

where $i = 1,\ldots,n$ and $x[i,j,\ell]$ are literals. It is a routine task to check that the completion $comp(P)$ can be converted into a 3-normalized formula in a number of steps being a polynomial with respect to the size of the program P. Hence, $\mathcal{SP}'_{=}(\mathcal{A})$ is in $W[3]$. □

COROLLARY 5.6. *$\mathcal{SP}'_{=}(\mathcal{H})$ and $\mathcal{SP}'_{=}(\mathcal{A})$ are* W[3]*-complete.* □

THEOREM 5.7. *The problems $\mathcal{ST}'_{\leq}(\mathcal{A})$ and $\mathcal{ST}'_{=}(\mathcal{A})$ are* W[3]*-hard.*

Proof: By Corollary 2.6, it suffices to show that $\mathcal{ST}'_{\leq}(\mathcal{A})$ is W[3]-hard. 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

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

and

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

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 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

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

(where $x_1, x_2, \ldots, x_k$ 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 2.3. □

COROLLARY 5.8. *The problem* $\mathcal{SP}'_{\leq}(\mathcal{A})$ *is* W[3]*-hard.*

Proof: A *positive cycle* in a logic program P is a sequence of rules $r_0, r_1, \ldots, r_n$ in P such that for every $i = 0,1,\ldots,n-1$, $h(r_i) \in b^+(r_{i+1})$ and $h(r_n) \in b^+(r_0)$. It is easy to see that the program P constructed in the proof of Theorem 5.7 does not contain positive cycles. Therefore, by the Fages lemma [Fages 1994], stable and supported models of P coincide. Thus, the proof of Theorem 5.7 applies in the case of supported models too. □

By Theorem 5.5, Corollary 5.8 and Corollary 2.6 we get the following result.

COROLLARY 5.9. *The problem* $\mathcal{SP}'_{\leq}(\mathcal{A})$ *is* W[3]*-complete.* □

THEOREM 5.10. *The problem* $\mathcal{SP}'_{=}(\mathcal{N})$ *is in* W[2].

Proof: We will reduce the problem $\mathcal{SP}'_{=}(\mathcal{N})$ to $\mathcal{M}'_{=}(2N)$ (which belongs to W[2] by Theorem 2.1).

Let us consider a purely negative program P with $At(P) = \{x_1,\ldots,x_n\}$. Its completion consists of formulas

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

where $x[i,j,\ell] \in At(P)$.

For each $x_i \in At(P)$, we introduce new atoms $x_i[1], x_i[2], \ldots, x_i[2^k]$. Next, for each set $U(i,j) = \{x[i,j,\ell] :\ \ell = 1,\ldots,m_{ij}\}$ we define a new atom $u[i,j]$. Finally, we introduce yet another set of new atoms: $z[1],\ldots,z[2^k]$.

Let us consider the following formulas:

$$A(i,t) = \quad x_i \Leftrightarrow x_i[t], \quad i = 1,\ldots,n,\ t = 1,\ldots,2^k,$$

$$B(x,i,j) = \quad x \Rightarrow u[i,j], \quad x \in U(i,j),\ i = 1,\ldots,n,\ j = 1,\ldots,m_i,$$

$$C(i) = \quad x_i \Rightarrow \bigvee_{j=1}^{m_i} \neg u[i,j], \quad i = 1,\ldots,n,$$

$$D(i) = \quad x_i \Leftarrow \bigvee_{j=1}^{m_i} \bigwedge_{\ell=1}^{m_{ij}} \neg x[i,j,\ell], \quad i = 1,\ldots,n,$$

$$E(t) = \quad z[t] \Leftrightarrow z[t], \quad t = 1,\ldots 2^k.$$

We define Φ_P to be the conjunction of the formulas listed above. Since each of these formulas can be rewritten as a conjunction of disjunctions, it is clear that without loss of generality we may assume that Φ_P is a 2-normalized formula. Let us also note that the number of atoms of Φ_P is given by the formula $|At(\Phi_P)| = n(2^k+1) + \sum_{i=1}^{n} m_i + 2^k$.

We claim that P has a supported model of size $n-k$ if and only if Φ_P has a model of size $|At(\Phi_P)| - (k+1)2^k - k$. To prove it, we proceed as follows.

Let $At(P)\setminus M$, where $M = \{x_{i_1}, \ldots, x_{i_k}\}$, be a supported model of P ($x_{i_1}, \ldots, x_{i_k}$ are some k distinct atoms of P). We denote by q the number of subsets of M different from all sets $U(i,j)$. We will show that $At(\Phi_P) \setminus M'$, where

$$M' = M \cup \{x_i[t]:\ x_i \in M,\ t = 1, \ldots, 2^k\} \cup \{u[i,j]:\ U(i,j) \subseteq M\} \cup \{z[1], \ldots, z[q]\}$$

is a model of Φ_P. First, let us observe that $|M'| = k + k2^k + (2^k - q) + q = (k+1)2^k + k$.

Clearly, by the definition of M', $At(\Phi_P) \setminus M'$ is a model of each formula $A(i,t)$. Let us consider a formula $B(x,i,j)$, for some i,j such that $1 \leq i \leq n$ and $1 \leq j \leq m_i$, and for some $x \in U(i,j)$. If $x \in At(\Phi_P) \setminus M'$, then $x \notin M$. It follows that $U(i,j) \not\subseteq M$. Consequently, $u[i,j] \in At(\Phi_P) \setminus M'$ and, so, $At(\Phi_P) \setminus M'$ is a model of $B(x,i,j)$.

Next, let us consider a formula $C(i)$, for some i, $1 \leq i \leq n$. Further, let us assume that $x_i \in At(\Phi_P) \setminus M'$. It follows that $x_i \in At(P) \setminus M$. Since $At(P) \setminus M$ is a supported model of P, $At(P) \setminus M$ satisfies the formula Φ_i. Thus, there is j, $1 \leq j \leq m_i$, such that for all $\ell = 1, \ldots, m_{ij}$, $x[i,j,\ell] \in M$. Hence, $U(i,j) \subseteq M$ and, consequently, $u[i,j] \notin At(\Phi_P) \setminus M'$. Thus, $At(\Phi_P) \setminus M'$ is a model of $C(i)$.

Since $At(P) \setminus M$ satisfies each formula Φ_i, $1 \leq i \leq n$, it is clear that $At(\Phi_P) \setminus M'$ satisfies the formula $D(i)$. Since all formulas $E(t)$, $1 \leq t \leq 2^k$, are tautologies, $At(\Phi_P) \setminus M'$ is a model of each of them, too. Thus, $At(\Phi_P) \setminus M'$ is a model of Φ_P.

Conversely, let $At(\Phi_P) \setminus M'$ be a model of Φ_P, for some set $M' \subseteq At(\Phi_P)$ such that $|M'| = (k+1)2^k + k$. Let $M = At(P) \cap M'$. If $|M| > k$ then, since all formulas $A(i,t)$ hold in $At(\Phi_P) \setminus M'$, $|M'| \geq |M|(2^k+1) \geq (k+1)(2^k+1) > (k+1)2^k + k$, a contradiction. Next, let us consider the case $|M| < k$ and let us assume that $u[i,j] \in M'$, for some i and j such that $1 \leq i \leq n$ and $1 \leq j \leq m_i$. Since $At(\Phi_P) \setminus M'$ is a model of all formulas $B(x,i,j)$, where $x \in U(i,j)$, it follows that for every $x \in U(i,j)$, $x \in M$. Thus, $U(i,j) \subseteq M$ and, consequently,

$$|\{u[i,j]:\ u[i,j] \in M'\}| = |\{U(i,j):\ U(i,j) \subseteq M\}| \leq 2^{|M|}.$$

Therefore,

$$|M'| \leq |M|(2^k+1) + 2^{|M|} + 2^k < (k+1)2^k + k - 1 < (k+1)2^k + k,$$

a contradiction again. Thus, $|M| = k$.

We will show that $At(P) \setminus M$ is a supported model of P. To this end, we will prove that $At(P) \setminus M$ is a model of all formulas Φ_i, $1 \leq i \leq n$. Since $At(\Phi_P) \setminus M'$ satisfies all formulas $D(i)$, $1 \leq i \leq n$, and since all atoms appearing in these formulas belong to $At(P)$, it follows that $At(P) \setminus M$ satisfies all formulas $D(i)$, $1 \leq i \leq n$.

To show that $At(P) \setminus M$ is a model of a formula Φ_i, $1 \leq i \leq n$, it is then sufficient to prove that $At(P) \setminus M$ is a model of the implication

$$x_i \Rightarrow \bigvee_{j=1}^{m_i} \bigwedge_{\ell=1}^{m_{ij}} \neg x[i,j,\ell]. \qquad (3)$$

Let $x_i \in At(P) \setminus M$. Then, by the implication $C(i)$, which holds in $At(\Phi_P) \setminus M'$, there exists j, $1 \leq j \leq m_i$, such that $u[i,j] \in M'$. Using the implications $B(x,i,j)$ and reasoning as before, it is easy to show that $U(i,j) \subseteq M$. Thus, $At(P) \setminus M$ is a model of $\bigwedge_{\ell=1}^{m_{ij}} \neg x[i,j,\ell]$ and, consequently, of the implication (3). □

A *kernel* of a digraph is an independent set S of vertices (that is, a set of vertices with no edge with both the initial and terminal vertices in S) such that every vertex not in S is a terminal vertex of some edge whose initial vertex is in S.

Let us recall that $\mathcal{N}_1$ denotes the class of purely negative programs such that each atom occurs exactly once in the head of a rule. We define $\mathcal{N}_2$ to be the class of purely negative programs such that there is exactly one negated atom in the body of each rule.

Let $P \in \mathcal{N}_i$, $i = 1, 2$. We define $G(P)$ to be a digraph with the vertex set $At(P)$ and the edge set consisting of pairs (y, x) such that there is a rule in P with the head x and $\mathbf{not}(y)$ in the body.

LEMMA 5.11. *(1) Let $P \in \mathcal{N}_1$. A set $S \subseteq At(P)$ is a stable model of P if and only if S is a kernel in $G(P)$.*

(2) Let $P \in \mathcal{N}_2$. A set $S \subseteq At(P)$ is a stable model of P if and only if $At(P) \setminus S$ is a kernel in $G(P)$.

Proof: (1) Let us assume that $S \subseteq At(P)$ is a stable model of a program $P \in \mathcal{N}_1$. For every $x \in At(P)$, let us denote by r_x the only rule of P with x as the head.

Let us consider a vertex $x \in S$. Then, r_x is not blocked by S. Hence, for every y in the body of r_x, $y \notin S$. In other words, for every y such that (y, x) is an edge of $G(P)$, $y \notin S$. Thus, S is an independent set.

Next, let us consider a vertex $x \notin S$. Then, r_x is blocked by S. Consequently, there is y in the body of r_x such that $y \in S$. In other words, there is an edge (y, x) in $G(P)$ such that $y \in S$.

It follows that S is a kernel of $G(P)$. The proof of the converse implication is similar.

(2) Let $S \subseteq At(P)$ be a stable model of a program $P \in \mathcal{N}_2$. Let us denote $S' = At(P) \setminus S$. We will show that S' is a kernel of $G(P)$. Let $x \in S'$. Then $x \notin S$. Since S is a stable model of P and since $P \in \mathcal{N}_2$, it follows that every rule $x \leftarrow \mathbf{not}(y)$ in P is blocked by S or, equivalently, that $y \in S$. Consequently, for every edge (y, x) in $G(P)$, if $x \in S'$, then $y \notin S'$. Thus, S' is an independent set.

Next, let us consider $x \notin S'$. Then, $x \in S$. Since S is a stable model of P, there is a rule $x \leftarrow \mathbf{not}(y)$ in P such that $y \notin S$. It follows that $y \in S'$. Thus, for every $x \notin S'$, there is an edge (y, x) in $G(P)$ such that $y \in S'$. Consequently, S' is a kernel of $G(P)$. The proof of the converse statement is similar. □

THEOREM 5.12. *The problems $\mathcal{ST}'_{\leq}(\mathcal{N})$, $\mathcal{ST}'_{=}(\mathcal{N})$, $\mathcal{SP}'_{\leq}(\mathcal{N})$ and $\mathcal{SP}'_{=}(\mathcal{N})$ are* W[2]*-complete.*

Proof: We will first reduce $\mathcal{ST}_{\leq}(\mathcal{N}_1)$ to $\mathcal{ST}'_{\leq}(\mathcal{N})$. Let $P \in \mathcal{N}_1$. We define Q to be a program in $\mathcal{N}_2 \subseteq \mathcal{N}$ such that $G(Q) = \overline{G}(P)$. The program Q is determined uniquely by the digraph $G(P)$. We will show that P has a stable model of size at most k if and only if Q has a stable model of size at least $|At| - k$, where At is the set of atoms of both Q and P. By Lemma 5.11 P has a stable model S of size at most k if and only if S is a kernel of the digraph $G(P)$ of cardinality at most k. Lemma 5.11 implies now that $G(Q) = G(P)$ has a kernel S of cardinality at most k if and only if $At \setminus S$ is a stable model of Q of cardinality at least $|At| - k$.

It follows that the problem $\mathcal{ST}_{\leq}(\mathcal{N}_1)$ can be reduced to the problem $\mathcal{ST}'_{\leq}(\mathcal{N})$. By Theorem 4.5 it follows that the problem $\mathcal{ST}'_{\leq}(\mathcal{N})$ is W[2]-hard. Since stable and supported models of purely negative programs coincide, $\mathcal{SP}'_{\leq}(\mathcal{N})$ is W[2]-hard. Theorems 5.10 and Corollary 2.6 imply now that both $\mathcal{SP}'_{\leq}(\mathcal{N})$ and $\mathcal{SP}'_{=}(\mathcal{N})$ are W[2]-complete. The W[2]-completeness of the problems $\mathcal{ST}'_{\leq}(\mathcal{N})$ and $\mathcal{ST}'_{=}(\mathcal{N})$ follows again from the fact that stable and supported models coincide for purely negative programs. □

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grants No. 9619233, 9874764 and 0097278. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

Apt, K. 1990. Logic programming. In *Handbook of theoretical computer science*, J. van Leeuven, Ed. Elsevier, Amsterdam, 493–574.

Apt, K. and van Emden, M. 1982. Contributions to the theory of logic programming. *Journal of the ACM 29,* 3, 841–862.

Clark, K. 1978. Negation as failure. In *Logic and data bases*, H. Gallaire and J. Minker, Eds. Plenum Press, New York-London, 293–322.

Dowling, W. and Gallier, J. 1984. Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *Journal of Logic Programming 1,* 3, 267–284.

Downey, R. G. and Fellows, M. R. 1997. *Parametrized Complexity.* Springer-Verlag.

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

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

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

Lloyd, J. W. 1984. *Foundations of logic programming.* Symbolic Computation. Artificial Intelligence. Springer-Verlag, Berlin-New York.

Marek, W. and Truszczyński, M. 1991. Autoepistemic logic. *Journal of the ACM 38,* 3, 588–619.

Marek, W. and Truszczyński, M. 1993. *Nonmonotonic logics; context-dependent reasoning.* Springer-Verlag, Berlin.

Truszczyński, M. 2002. Computing large and small stable models. *Theory and Practice of Logic Programming 2.*

Received August 2000; revised December 2001; accepted December 2001