Annals of Mathematics and Artificial Intelligence (2019) 86:121–147
https://doi.org/10.1007/s10472-019-09633-x

A multiparametric view on answer set programming

Johannes K. Fichte[1] · **Martin Kronegger**[2,3] · **Stefan Woltran**[3]

Published online: 8 June 2019
© Springer Nature Switzerland AG 2019

Abstract

Disjunctive answer set programming (ASP) is an important framework for declarative modeling and problem solving, where the computational complexity of basic decision problems like consistency (deciding whether a program has an answer set) is located on the second level of the polynomial hierarchy. During the last decades different approaches have been applied to find tractable fragments of programs, in particular, also using parameterized complexity. However, the full potential of parameterized complexity has not been unlocked since only one or very few parameters have been considered at once. In this paper, we consider several natural parameters for the consistency problem of disjunctive ASP. In addition, we also take the sizes of the answer sets into account; a restriction that is particularly interesting for applications requiring small solutions as encoding subset minimization problems in ASP can be done directly due to inherent minimization in its semantics. Previous work on parameterizing the consistency problem by the size of answer sets yielded mostly negative results. In contrast, we start from recent findings for the problem WMMSAT and show several novel fixed-parameter tractability (fpt) results based on combinations of parameters. Moreover, we establish a variety of hardness results (paraNP, W[2], and W[1]-hardness) to assess tightness of our parameter combinations.

Keywords Answer set programming (ASP) · Parameterized complexity theory · Multiparametric complexity · Fixed-parameter tractability · Multi parameterizations

The work has been supported by the Austrian Science Fund (FWF), Grant Y698 and S11408-N23. Main work was carried out while the first two authors were PostDocs at TU Wien and the first author was also affiliated with the Institute of Computer Science and Computational Science at University of Potsdam, Germany.

✉ Johannes K. Fichte
johannes.fichte@tu-dresden.de

Martin Kronegger
martin.kronegger@jku.at; martin.kronegger@tuwien.ac.at.at

Stefan Woltran
woltran@dbai.tuwien.ac.at

[1] TU Dresden, Dresden, Germany

[2] Johannes Kepler University Linz, Linz, Austria

[3] TU Wien, Vienna, Austria

Mathematics Subject Classification (2010) Primary 03D15 · Secondary 68T30, 03B42, 03B70, 68T27, 8Q15

1 Introduction

Answer set programming (ASP) is an important framework for declarative modeling and problem solving [50]. In propositional ASP, a problem is described in terms of a logic program consisting of rules over propositional atoms. Answer sets, which are sometimes also referred to as stable models [35, 48, 54], are then the solutions to such a logic program.

We illustrate answer set programming by showing how to encode a generalized version of the well-known vertex cover problem on hypergraphs where no more than two vertices of a hyperedge are selected. The problem HITTING SET$_2$ asks given a hypergraph $H = (V, E)$ with $E \subseteq 2^V \setminus \emptyset$ and an integer k whether there exists a set $S \subseteq V$ of size at most k which intersects with each $e \in E$ and for every distinct vertices $u, v, w \in e$ of any hyperedge $e \in E$ we have $\{u, v, w\} \not\subseteq S$. To decide the problem HITTING SET$_2$ one can use the encoding $\{\, v_1 \vee \ldots \vee v_\ell \leftarrow \ : \{v_1, \ldots, v_\ell\} \in E \,\} \cup \{\, \leftarrow v_1, v_2, v_3 : \{v_1, v_2, v_3\} \subseteq e, v_1 \neq v_2, v_2 \neq v_3, v_1 \neq v_3, e \in E \,\}$ and then decide whether there is an answer set matching the bound k. Note that this program in fact already encodes the subset minimal version of the problem, however, still allows to find a hitting setting with the properties mentioned above if one exists. Namely, we simply ask for an answer set of size at most k of the program above.

Fundamental problems of ASP are the CONSISTENCY and BRAVE REASONING problem. CONSISTENCY asks to decide whether a given disjunctive, propositional program has an answer set. BRAVE REASONING asks given a disjunctive, propositional program and an atom that occurs in the program whether the atom occurs in some answer set of the program. Interestingly, both problems CONSISTENCY and BRAVE REASONING are complete for the second level of the polynomial hierarchy [15]. In consequence, one can encode problems into ASP that are harder than NP such as conformant planning [61], maximal satisfiable sets [42], minimal diagnosis [33], and 2-QBF [55].

Even though CONSISTENCY and BRAVE REASONING are of high classical worst-case complexity, this does not rule out that we can efficiently find solutions if the input instances are restricted. On that score, several restrictions on input programs have been identified in the literature that make the CONSISTENCY and BRAVE REASONING problem tractable such as forbidding disjunctions and negations (Horn programs) or NP-complete such as forbidding disjunctions (normal programs). Truszczyński [60] has established detailed trichotomy results (P-membership, NP-completeness, and Σ_2^P-completeness) for the classical complexity of ASP reasoning problems depending on syntactic properties of programs similar to Schaefer's base classes for propositional satisfiability [57]. Fichte, Truszczyński, and Woltran [28] provide classes of programs (dual-normal and body-cycle-free programs) where the complexity of deciding whether a program has an answer set is NP-complete.

Further, the high classical worst-case complexity of the ASP problems CONSISTENCY and BRAVE REASONING does not rule out that a certain (hidden) structure is present in an input instance and then we can use this structure for more efficient problem solving. A prominent approach to analyze and understand computational complexity incorporating the existence of certain hidden structure is to use the framework of parameterized complexity [13]. The main idea of parameterized complexity is to fix a certain structural property (the parameter) of a problem instance and to consider the computational complexity of the problem in dependency of the parameter. Many parameterized complexity analyzes have been carried out for ASP problems with various parameters. Gottlob, Scarcello, and Sideri [37] have provided fixed-parameter tractability results of several problems in artificial intelligence (AI) and non-monotonic reasoning, including answer set programming without disjunctions. Gottlob and Szeider [38] presented a survey on parameterized complexity of problems in AI, database theory and automated reasoning. Gottlob, Pichler, and Wei [36] have shown various tractability results for problems including CONSISTENCY and BRAVE REASONING for disjunctive answer set programming. Their results are based on a logical characterizations of the problems in terms of a formula in monadic second order logic and a well-known theorem by Courcelle [8], which states that a problem is fixed-parameter tractable when parameterized by treewidth if the problem is expressible in monadic second order logic. Jakl, Pichler, and Woltran [39] have established tractability results for bounded treewidth of a graph representation of a program, namely the incidence graph, by means of dynamic programming algorithms. Pichler et al. [56] have shown both hardness and tractability results for bounded treewidth and bounded weights for programs that contain in addition weight constraints. Bliem, Ordyniak, and Woltran [3] have established that for most of the natural directed width measures of disjunctive programs one cannot expect tractability. However, they have given a fixed-parameter tractable algorithm for the parameter signed clique-width. Fichte et al. [20, 21] have considered decision, counting, and optimization problems for ASP programs that allow for all kinds of ASP rules used in modern solvers [59] when parameterized by the treewidth of the primal graph as well as the incidence graph. Both works prove that the presented algorithms are fixed-parameter tractable and also give solvers to exploit treewidth in practice. There have been also initial investigations on non-ground (first-order) answer set programming and the effect of grounding on treewidth [2]. Fichte and Szeider have established several tractability results and reductions for the parameter size of a backdoor into various tractable [17, 25] and intractable classes of programs [26] for disjunctive programs, where a backdoor is a set of atoms that represents in a way “clever reasoning shortcuts” through the search space. Also fine-grained approaches to the evaluation of strong backdoors in terms of backdoor trees have been considered [27]. Finally, Lonc and Truszczyński [49] have considered the parameterized complexity of the consistency problem parameterized by a given integer k, when the input is restricted to normal (i.e., disjunction-free) programs and when then answer sets are allowed to be of size exactly k, or at most k, or at least k, and established various hardness results.

While all results stated above only consider one parameter at a time, one may also investigate on multiple parameter combinations. This has been done already in the 90s for problems in computational biology [4, 5]. Later in AI systematic analysis of various parameter combinations with the goal to cover all possible parameter combinations have been done for weighted minimal model satisfiability (WMMSAT) [46], sub-graph isomorphism [51], and planning [43–45]. So far, there has been no rigorous study of disjunctive ASP when considering various combinations of structural properties.

In this paper, we study the computational complexity of propositional disjunctive ASP for the problems CONSISTENCY and BRAVE REASONING using the framework of parameterized complexity theory [10, 13]. We consider several combinations of natural ASP parameters at once, which allows us to draw a detailed map for a multiparametric view on ASP complexity. In particular, we also take the sizes of the answer sets into account. Such a restriction is particularly interesting for applications that require small solutions. Small solutions are often interesting for minimization problems originating in graph theory and closest string problems, which have applications to computational biology [47].

Paper organization and detailed contributions After giving some preliminary explanations on propositional satisfiability, answer set programming, and parameterized complexity in Sections 2.1–2.3, we formally define the considered parameters in Section 2.4. We briefly state connections between the considered ASP problems when we are interested in answer sets of size at most k and of arbitrary size in Section 2.5. We then turn our attention to the hardness results (Section 3). It will turn out that most hardness results hold for the problems CONSISTENCY and BRAVE REASONING as well as for its versions where we restrict the size of the answer set. In particular, we present a paraNP-, three W[2]- and three W[1]-hardness results for each of the problems. Table 3 on page 12 provides an overview on our results. The proofs of four results are based on a novel, direct fixed-parameter tractable reduction from the problem WMMSAT. Our reduction preserves the semantics in the sense that models are in a one to one correspondence, which immediately allows WMMSAT solving by means of an ASP solver (proof of Theorem 9). Furthermore, we incorporate results by Lonc and Truszczyński [49] and Truszczyński [60]. In Section 4, we then present several novel fixed-parameter tractability results for k-CONSISTENCY and BRAVE REASONING. Since the problem WMMSAT and ASP are quite related in terms of their semantics when the answer set programs are restricted to programs where no negations except in rules with an empty head occur, we start from results by Lackner and Pfandler [46] for WMMSAT, transform several of these results to this restricted version of ASP (Lemma 12) and point out limitations where the methods used for WMMSAT are insufficient. Then, we show which additional structural properties we need to take into account to still obtain fixed-parameter tractability. Therefore, we construct a reduction that builds multiple programs, which can then be solved in fpt-time using results for the restricted version from above. Beyond, we relate our parameters to parameters such as backdoors and treewidth, which have previously been studied in the literature only for ASP problems when arbitrarily large answer sets are allowed (Section 5). Our results allow us to draw a detailed map for various ASP parameter combinations, however, also open up a few remaining interesting cases for which we can neither establish tractability nor hardness.

2 Preliminaries

2.1 Answer set programming

Let U be a universe of propositional *atoms*. A *literal* is an atom $a \in U$ or its negation $\neg a$. A *disjunctive logic program* (or simply a *program*) P is a set of *rules* of the form $a_1 \vee \ldots \vee a_l \leftarrow b_1, \ldots, b_n, \neg c_1, \ldots, \neg c_m$ where $a_1, \ldots, a_l, b_1, \ldots, b_n, c_1, \ldots, c_m$ are atoms and l, n, m are non-negative integers. Further, let H, B^+, and B^- map rules to sets of atoms such that for a rule r we have $H(r) = \{a_1, \ldots, a_l\}$ (the *head* of r), $B^+(r) = \{b_1, \ldots, b_n\}$ (the *positive body* of r), and $B^-(r) = \{c_1, \ldots, c_m\}$ (the *negative body* of r). In addition to the traditional representation of a rule above, we sometimes also write $H(r) \leftarrow B^+(r), \neg B^-(r)$, and $H(r) \leftarrow B^+(r)$ instead of $H(r) \leftarrow B^+(r), \neg\emptyset$. We denote the sets of atoms occurring in a rule r or in a program P by $\text{at}(r) := H(r) \cup B^+(r) \cup B^-(r)$ and $\text{at}(P) := \bigcup_{r \in P} \text{at}(r)$, respectively. We write $\text{occ}_P(a) := \{\, r \in P : a \in \text{at}(r)\,\}$. We denote the number of rules of P by $|P| := |\{\, r : r \in P\,\}|$. The *size* $\|P\|$ of a program P is defined as $\sum_{r \in P} |H(r)| + |B^+(r)| + |B^-(r)|$.

A rule r is *negation-free* if $B^-(r) = \emptyset$, r is *normal* if $|H(r)| \leq 1$, r is a *constraint* (an integrity rule) if $|H(r)| = 0$, r is *Horn* if it is negation-free and normal or a constraint, r is *definite Horn* if it is Horn and not a constraint, r is *tautological* if $B^+(r) \cap (H(r) \cup B^-(r)) \neq$

$\emptyset$, and *non-tautological* if it is not tautological, and r is *positive-body-free* if $B^+(r) = \emptyset$. We say that a program has a certain property if all its rules have the property. **NF+Cons** refers to the class of all programs where negation-free rules and arbitrary constraint rules (may also contain negative atoms) are allowed. Let P and P' be programs. We say that P' is a *subprogram* of P (in symbols $P' \subseteq P$) if for each rule $r' \in P'$ there is some rule $r \in P$ with $H(r') \subseteq H(r)$, $B^+(r') \subseteq B^+(r)$, $B^-(r') \subseteq B^-(r)$. We identify the parts of a program P consisting of proper rules as $P_r := \{ r \in P : H(r) \neq \emptyset \}$ and constraints as $P_c := P \setminus P_r$. We occasionally write $\perp$ as head if $H(r) = \emptyset$. If $B^+(r) \cup B^-(r) = \emptyset$, we simply write $H(r)$ instead of $H(r) \leftarrow \emptyset, \emptyset$. We also write $H(P) := \bigcup_{r \in P} H(r)$, $B^-(P) := \bigcup_{r \in P} B^-(r)$.

A set M of atoms *satisfies* a rule r if $(H(r) \cup B^-(r)) \cap M \neq \emptyset$ or $B^+(r) \setminus M \neq \emptyset$. The set M is a *model* of P if it satisfies all rules of P. We say that M is a *minimal model* if M is a model of P and there is no $M' \subsetneq M$ such that M' is a model of P.

The *Gelfond-Lifschitz (GL) reduct* of a program P under a set M of atoms is the program $P^M := \{ H(r) \leftarrow B^+(r) : r \in P, B^-(r) \cap M = \emptyset \}$ [35]. M is an *answer set* (or *stable model*) of a program P if M is a minimal model of P^M. We denote by $\mathrm{AS}(P)$ the set of all answer sets of P and for some integer $k \geq 0$ by $\mathrm{AS}_k(P)$ the set of all answer sets of P of size at most k.

Example 1 Consider the following program P consisting of the rules:

$$P = \{ \quad a \vee c \leftarrow b; \qquad b \vee c \leftarrow e; \qquad b \leftarrow c, \neg g; \\ e \leftarrow; \qquad \leftarrow e, \neg a, \neg c; \qquad g \leftarrow a \}.$$

The set $M_1 = \{a, b, e, g\}$ is an answer set of P, since M_1 is a model of P and a minimal model of the reduct $P^{M_1} = \{ a \vee c \leftarrow b;\ b \vee c \leftarrow e;\ e \leftarrow;\ g \leftarrow a \}$. Further, $M_2 = \{b, c, e\}$ is an answer set, since M_2 is a model of P and a minimal model of the reduct $P^{M_2} = \{ a \vee c \leftarrow b;\ b \vee c \leftarrow e;\ b \leftarrow c;\ e \leftarrow;\ g \leftarrow a \}$.

It is easy to see that M_1 and M_2 are the only answer sets of P. Then, we have $\mathrm{AS}(P) = \{M_1, M_2\}$. However, the set of all answer sets of P of size at most 3 consists only of one answer set, namely $\mathrm{AS}_3(P) = \{M_2\}$.

It is well known that Horn programs have a unique answer set or no answer set and that this set can be found in linear time. Note that every definite Horn program P has a unique minimal model which equals the least model $LM(P)$ [34]. Dowling and Gallier [11] have established a linear-time algorithm for testing the satisfiability of propositional Horn formulas which easily extends to Horn programs.

Observation 2 (Folkore) Let P be a program and $M \subseteq \mathrm{at}(P)$. If M is a minimal model of P^M, then (i) M is a minimal model of P, (ii) $M \subseteq \bigcup_{r \in P} H(r)$, and (iii) $|M| \leq |P_r|$.

Proof We prove Statement (i). Assume that M is a minimal model of P^M. By definition of an answer set for each rule $r \in P$ we have (a) $B^-(r) \cap M \neq \emptyset$ or (b) there is a corresponding rule $r' \in P^M$ such that $H(r) = H(r')$, $B^+(r) = B^+(r')$, and $B^-(r') = \emptyset$. If Case (a) holds, M satisfies r. If Case (b) holds, M satisfies r' as M is a minimal model of P^M. Thus, M also satisfies r. Consequently, M satisfies every $r \in P$ and is hence a model of P.

In order to show that no proper subset of M is a model of P choose arbitrarily a proper subset $N \subsetneq M$. Since M is a minimal model of P^M, N cannot be a minimal model of P^M. Consequently, there must be a rule $r \in P$ such that $B^-(r) \cap M = \emptyset$ (i.e., r is not deleted by forming P^M), $B^+(r) \subseteq N$ and $H(r) \cap N = \emptyset$. Since $N \subsetneq M$ and $B^-(r) \cap M = \emptyset$, we obtain $B^-(r) \cap N = \emptyset$. Hence, $(H(r) \cap B^-(r)) \cap N = \emptyset$ and $B^+(r) \setminus N \neq \emptyset$. Thus, N does not satisfy r and is consequently not a model of P. We conclude that M is a minimal model of P.

For Statement (ii) we refer to standard texts [31, Chapter 2]. Statement (iii) is an immediate consequence of Statements (i) and (ii). □

In this paper, we consider the following decision problems of ASP.

Problem: k-CONSISTENCY
Input: A program P and an integer k.
Task: Decide whether P has an answer set of size at most k.

Problem: k-BRAVE REASONING
Input: A program P, an atom $a \in \text{at}(P)$, and an integer k.
Task: Decide whether P has an answer set M of size at most k such that $a \in M$.

We denote by k-*AspProblems* the family of the decision problems k-CONSISTENCY and k-BRAVE REASONING.

Further, we use the following problem.

Problem: k-ENUM
Input: A program P and an integer k.
Task: List all answer sets of size at most k of P.

We refer to the problems as CONSISTENCY, BRAVE REASONING, and ENUM, respectively, if the integer k can be arbitrarily large. We denote by *AspProblems* the family of the reasoning problems CONSISTENCY and BRAVE REASONING.

2.2 Propositional satisfiability

We also need some notions from *propositional satisfiability*. A *literal* is a variable or its negation. A *clause* is a finite set of literals. For a clause c of the form $c = \{x_1, \dots, x_\ell, \neg x_{\ell+1}, \dots \neg x_m\}$ we let $L^+(c) := \{x_1, \dots, x_\ell\}$ (the *positive literals*) and $L^-(c) := \{x_{\ell+1}, \dots, x_m\}$ (the *negative literals*). For a positive literal $\ell = x$ we sometimes write x^1 and for a negative literal $\ell = \neg x$ we sometimes write x^0. A CNF formula is a finite set of clauses. The set of variables of a CNF formula F is denoted by $var(F)$.

A set M of atoms *satisfies* a clause c if $L^+(c) \cap M \neq \emptyset$ or $L^-(c) \setminus M \neq \emptyset$. M is a *model* of F if it satisfies all clauses of F.

The problem WSAT$_\leq$ is defined as follows.

Problem: WEIGHTED SATISFIABILITY (WSAT$_\leq$)
Input: A CNF formula F and some integer k.
Task: Decide whether F has a model $M \subseteq var(F)$ of cardinality $|M| \leq k$.

We say that M is a *minimal model* of F if M is a model of P and there is no $M' \subsetneq M$ such that M' is a model of F.

The problem WEIGHTED MINIMAL MODEL SATISFIABILITY (WMMSAT) is defined as follows.

Problem: WEIGHTED MINIMAL MODEL SATISFIABILITY (WMMSAT)
Input: CNF formulas φ and π and some integer k.
Task: Decide whether φ has a minimal model $M \subseteq var(F)$ such that M is also a model of π and is of cardinality $|M| \leq k$.

Intuitively, the purpose of π is to select particular minimal models among the minimal models of φ. Note that if the formula φ is empty the problem WMMSAT is equivalent to WSAT$_\leq$. Clearly, WMMSAT asks for "more than" just deciding whether the formula $\varphi \wedge \pi$ has a minimal model. Therefore, consider formulas $\varphi := (x \vee \neg x)$ and $\pi := x$. While $\{x\}$

is a minimal model of $\varphi \wedge \pi$, the only minimal model of φ is $\emptyset$. However, $\emptyset$ is not a model of π. Consequently, the answer to the WMMSAT problem is no.

2.3 Parameterized complexity

We assume that the reader is familiar with the main concepts of computational complexity theory, in particular, algorithms, standard encodings, (decision) problems, and complexity classes [1, 53]. In the following, we briefly give some basic background on parameterized complexity. For more detailed information we refer to other sources [10, 13].

A *parameterized problem* L is a subset of $\Sigma^* \times \mathbb{N}$ for some finite alphabet Σ. For an instance $(I, k) \in \Sigma^* \times \mathbb{N}$ we call I the *main part* and k the *parameter*. L is *fixed-parameter tractable* if there exist a computable function f and a constant c such that we can decide by an algorithm whether $(I, k) \in L$ in time $\mathcal{O}(f(k) \cdot \|I\|^c)$ where $\|I\|$ denotes the size of I. We call such an algorithm an *fpt-algorithm*. In other words, the running time of an fpt-algorithm is bounded by a computable function f on the parameter k and a polynomial $\|I\|^c$ of the size of the main part of the input I. Then, FPT is the class of all fixed-parameter tractable decision problems. Let $L \subseteq \Sigma^* \times \mathbb{N}$ and $L' \subseteq \Sigma'^* \times \mathbb{N}$ be two parameterized problems for some finite alphabets Σ and Σ'. An *fpt-reduction* r from L to L' is a many-to-one reduction from $\Sigma^* \times \mathbb{N}$ to $\Sigma'^* \times \mathbb{N}$ such that for all $I \in \Sigma^*$ we have $(I, k) \in L$ if and only if $r(I, k) = (I', k') \in L'$ such that $k' \leq g(k)$ for a fixed computable function $g : \mathbb{N} \to \mathbb{N}$ and there is a computable function f and a constant c such that r is computable in time $\mathcal{O}(f(k)\|I\|^c)$ where $\|I\|$ denotes the size of I [30]. Thus, an fpt-reduction is, in particular, an fpt-algorithm. It is easy to see that the class FPT is closed under fpt-reductions.

Parameterized complexity also facilitates a hardness theory to rule out the existence of fpt-algorithms. Next, we will define several parameterized complexity classes capturing fixed-parameter intractability needed in this work. We would like to note that the theory of fixed-parameter intractability is based on fpt-reductions [12, 29]. The *Weft hierarchy* consists of parameterized complexity classes $\mathrm{W}[1] \subseteq \mathrm{W}[2] \subseteq \cdots$ which are defined as the closure of certain parameterized problems under parameterized reductions. Note that showing NP-hardness is not sufficient here, since we can not distinguish between problems that are solvable in time $n^{f(k)}$ from problems solvable in time $f(k) \cdot n^{\mathcal{O}(1)}$. More precisely, the class W[1] contains all problems that are fpt-reducible to the problem INDEPENDENT SET when parameterized by the size of the solution [12, 30], which asks given an undirected graph and an integer k to decide whether G has an independent set of size at most k. An *independent set* of a graph $G = (V, E)$ is a subset $I \subseteq V$ of vertices for which no two vertices $v, w \in I$ are neighbors, i.e., $vw \notin E$. A prominent $W[2]$-complete problem is HITTING SET [12, 13] which asks given a family of sets (S, k) where $\mathsf{S} = \{S_1, \ldots, S_m\}$ and an integer k whether there exists a set H of size at most k which intersects with all the S_i. There is strong theoretical evidence that parameterized problems that are hard for classes $\mathrm{W}[i]$ are not fixed-parameter tractable [13]. This often results in the fundamental assumption $\mathrm{FPT} \subsetneq \mathrm{W}[1]$ in parameterized complexity, which is a natural parameterized analogue of the conjecture that $\mathrm{P} \neq \mathrm{NP}$. It is well-known that different variations of $\mathrm{WSAT}_{\leq}$ can be used to define the W-hierarchy (see, e.g., the work of Flum and Grohe [30]).

The class XP of *non-uniform* polynomial-time tractable problems consists of all parameterized decision problems that can be solved in polynomial time if the parameter is considered constant. That is, $(I, k) \in L$ can be decided in time $\mathcal{O}(\|I\|^{f(k)})$ for some computable function f.

Table 1 List of considered parameters in the work of Lackner and Pfandler [46] for the problem WMMSAT

k	Maximum weight of the minimal model
d	Maximum clause size
d^+, d^-	Maximum positive / negative clause size
h	Number of non-Horn clauses
p	Maximum number of positive occurrences of a variable in φ
v^+, v^-	Number of variables that occur as positive / negative literals in φ or in π
d^+_π	Maximum positive clause size in π
$\|\|\pi\|\|$	Size of π, i.e., the total number of variable occurrences in π

Further, the parameterized complexity class paraNP contains all parameterized decision problems L such that $(I, k) \in L$ can be decided *non-deterministically* in time $\mathcal{O}(f(k) \cdot \|I\|^c)$, for some computable function f and constant c [30]. Intuitively, the class paraNP consists of all problems that are in NP after a pre-computation that only involves the parameter [29]. The complexity class paraNP can be seen as an analogue of NP in parameterized complexity. The complexity classes XP and paraNP are incomparable subject to standard assumptions in computational complexity. Further, we have the classes of the Weft hierarchy between FPT and paraNP and XP in the following relation: FPT $\subseteq W[1] \subseteq \ldots \subseteq$ W[i] $\subseteq$ XP $\cap$ paraNP.

For function problems like k-ENUM, it is sometimes important to check whether it is possible to output all solutions in time $\mathcal{O}(f(k) \cdot n^c)$ for some computable function f and constant c or more precisely $\mathcal{O}(f(k) \cdot (n + |\mathrm{Sol}(n)|)^c)$, which is also known as outputFPT. This intuition is enough for what we consider in this paper. For a more formal definition we refer to the work of Creignou et al. [9].

The parameterized complexity of the problems WSAT$_{\leq}$ and WMMSAT has been studied in the work of Lackner and Pfandler [46]. In their work, they have considered the parameters as listed in Table 1. Several hardness and tractability results for parameter combinations turn out to be useful to show hardness and tractability results for the considered ASP problems.

2.4 Considered parameters

In this section, we introduce a list of ASP parameters, which mainly originate from earlier work for WMMSAT, for our parameterized complexity analysis. In particular, we are interested in parameter combinations. First, we give a definition what we mean by an ASP parameter.

Definition 3 An *ASP parameter* is a function p that assigns to every program P some non-negative integer $p(P)$.

Table 2 lists the considered parameters, and their intuitive descriptions. All parameters except k can be computed in polynomial time. For a more formal description, let P be a program and $X \subseteq \{H, B^+, B^-\}$ where H, B^+, and B^- are mappings defined as in Section 2.1. We omit P if the program is clear from the context. Further, let $\mathrm{at}_{X,r} := \bigcup_{f \in X} f(r)$.

Table 2 List and informal description of the considered parameters

k	Maximum size of an answer set
$\text{maxsize}^r_{H,B+,B-}$	Maximum size of a non-constraint rule
$\text{maxsize}^r_{H,B-}$	Maximum size of the head and negative body of a rule
$\text{maxsize}^r_{H,B+}$	Maximum size of the head and positive body of a rule
maxsize_H	Maximum size of the head of a rule
maxsize^r_{B+}	Maximum size of the positive body of a non-constraint rule
maxsize^r_{B-}	Maximum size of the negative body of a rule
maxsize^c_{B-}	Maximum size of the negative body of a constraint
$\#\text{non-Horn}^r$	Number of non-Horn rules
$\text{maxocc}^r_{H,B-}$	Maximum number of occurrences of a variable in P_r when only the head and negative-body occurrences are counted
maxocc^r_{B+}	Maximum number of occurrences of a variable in P_r when only the positive-body occurrences are counted
$\#\text{at}_H$	Number of atoms that occur in the head
$\#\text{at}_{B+}$	Number of atoms that occur in the positive body
$\#\text{at}_{B-}$	Number of atoms that occur in the negative body
$\#\text{at}^r_{B-}$	Number of atoms that occur in the negative body of all non-constraint rules
$\|\|P_c\|\|$	The total number of variable occurrences in P_c

Then, we define the following parameters.

$$
\begin{aligned}
\#\text{at}_X &:= |\bigcup_{r \in P} \text{at}_{X,r}| \\
\#\text{at}^r_X &:= |\bigcup_{r \in P, |H(r)|>0} \text{at}_{X,r}| \\
\text{maxsize}^r_X &:= \max\Big\{ \sum_{f \in X} |f(r')| : |H(r')| > 0, r' \in P \Big\} \\
\text{maxsize}^c_X &:= \max\Big\{ \sum_{f \in X} |f(r')| : |H(r')| = 0, r' \in P \Big\} \\
\#\text{non-Horn}^r &:= |\{r' : r' \in P, r' \text{ non-Horn}\}| \\
\text{maxocc}^r_X &:= \max\Big\{ i : a \in at(P), i = \sum_{f \in X,\, r' \in P,\, |H(r')|>0} |\{a : a \in f(r')\}| \Big\}
\end{aligned}
$$

Definition 4 Let p and q be ASP parameters. We say that p *dominates* q if there is a computable function f such that $p(P) \leq f(q(P))$ holds for all programs P.

Figure 1 depicts the relationship in terms of domination of parameters that are useful for our results. Note that this list is not complete.

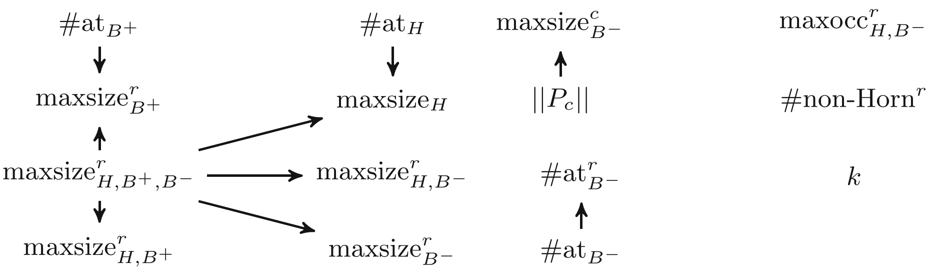

Fig. 1 Domination graph (relationship between parameters). Let x and y be parameters. There is an arc $x \rightarrow y$ whenever x dominates y

2.5 Relationship of *AspProblems* and *k*-*AspProblems*

Recent research in parameterized complexity in the setting of answer set programming has mainly focused on consistency or reasoning problems, which allow arbitrarily large answer sets. However, we focus on ASP problems that also take the sizes of the answer sets into account. In the following, we explain and summarize connections between both versions. We observe that if the parameters do not depend on the maximum size of an answer set, we can trivially extend known membership and hardness results for problems in *AspProblems* to the respective problem in k-*AspProblems*. In other words, the problem k-CONSISTENCY is at least as hard as CONSISTENCY. Finally, we state how to extend known results for CONSISTENCY using standard counters that do not effect the other considered parameters.

Observation 5 Let p be an ASP parameter, $\mathcal{C}$ be a parameterized complexity class, and $L \in$ {CONSISTENCY, BRAVE REASONING}, and k-L its corresponding decision problem k-CONSISTENCY or k-BRAVE REASONING, respectively. In other words, k-L decides the question of L when restricted to answer sets of size at most k. Then the following statements are true.

1. If the problem k-L is in $\mathcal{C}$ when parameterized by p and p does not depend on k, then the problem L is in $\mathcal{C}$ under fpt-reductions when parameterized by p.
2. Further, if the problem L is $\mathcal{C}$-hard when parameterized by p and p does not depend on k, then the problem k-L is $\mathcal{C}$-hard under fpt -reductions when parameterized by p.

Note that in Observation 5 the restriction "p does not depend on k" is quite weak as in that case both problems coincide.

Next, we will see that if a decision problem in *AspProblems* is fixed-parameter tractable when parameterized by some fixed parameter p and p is not affected by restricting the solution size to at most k, then the corresponding problem for answer sets of size at most k is fixed-parameter tractable when parameterized by the parameter combination $p+k$ where k is the size of the answer set.

Definition 6 Let p be an ASP parameter. Then we call p *counter-preserving* if $p(P) = f(p(P_k))$ for some computable function f, an integer k and $P_k := P \cup \{ c_{n+1,0} \leftarrow \top \} \cup \{ \bot \leftarrow c_{1,k+1} \} \cup \{ c_{i,j+1} \leftarrow c_{i+1,j}, a_i;\ c_{i,j} \leftarrow c_{i+1,j} : 1 \leq i \leq n, 0 \leq j \leq k+1 \}$ where $a_1, \ldots, a_n$ are the atoms of P.

Proposition 7 *Let p be a counter-preserving ASP parameter, $\mathcal{C}$ be a parameterized complexity class, $L \in$ {CONSISTENCY, BRAVE REASONING}, and k-L its corresponding decision problem k-CONSISTENCY or k-BRAVE REASONING, respectively. If the problem L belongs to class $\mathcal{C}$ when parameterized by p, then the problem k-L belongs to class $\mathcal{C}$ under fpt-reductions when parameterized by p.*

Proof Let P be a program and n an integer such that $n = |\text{at}(P)|$. We restrict the decision question to answer sets of size at most k by means of a simple counter. Therefore, we apply the construction from Definition 6, which implements an at most k constraint as described in standard literature [31, p.18ff.], to ensure that at most k atoms are set to true and hence belong to an answer set of P. In P_k we introduce auxiliary atoms $c_{i,j}$ for $1 \leq i \leq n+1$ and $0 \leq j \leq k+2$ resulting in $\mathcal{O}(n \cdot k)$ additional auxiliary atoms and $\mathcal{O}(n \cdot k)$ additional rules. We can then simply decide L on P_k instead of P and obtain the result for our initial problem k-L. Since L is fixed-parameter tractable, $p(P) = p(P_k)$, and $\|P_k\|$ is polynomial in $n \cdot k$, the overall construction gives an fpt-algorithm with respect to k. Hence, the proposition sustains. □

3 Hardness results

In this section, we present several hardness results for ASP reasoning problems. The results are also summarized in Table 3. Observe that hardness for a combination of parameters trivially implies hardness for any subset of these parameters.

In the next proposition, we summarize known hardness results for WMMSAT which turn out to be useful for showing hardness for several parameter combination for k-CONSISTENCY and k-BRAVE REASONING.

Table 3 Summary of multiparametric complexity results for k-CONSISTENCY. We refer to Section 6 for open parameter combinations

Parameter	Result	Reference
$\#\text{at}_H$	$\in$ FPT	Obs. 10 (i)
$k + \text{maxsize}^r_{H,B-}$	$\in$ FPT	Thm. 16 (i)
$\#\text{non-Horn}^r + \text{maxsize}^r_{H,B-}$	$\in$ FPT	Thm. 16 (ii)
$k + \#\text{at}_{B+} + \text{maxocc}^r_{H,B-} + \text{maxsize}^c_{B-} + \#\text{at}^r_{B-}$	$\in$ FPT	Thm. 19 (i)
$\#\text{at}_{B+} + \#\text{non-Horn}^r + \text{maxsize}^c_{B-} + \#\text{at}^r_{B-}$	$\in$ FPT	Thm. 19 (ii)
k	$\in$ XP	Obs. 11 (i)
$k + \text{maxsize}^r_{B+} + \#\text{non-Horn}^r + \text{maxocc}^r_{H,B-} + {}$ $+ \text{maxsize}^c_{B-} + \|\|P_c\|\|$	W[1]-h	Thm. 9 (vii)
$k + \text{maxsize}^r_{H,B+} + \text{maxocc}^r_{B+} + \text{maxsize}^c_{B+} + \text{maxsize}^c_{B-}$	W[2]-h	Thm. 9 (iii)
$k + \text{maxsize}^r_{B-} + \text{maxsize}^c_{B+} + \text{maxsize}^c_{B-}$	W[2]-h	Thm. 9 (vi)
$k + \text{maxsize}^r_{B+} + \#\text{at}_{B+} + \text{maxsize}^c_{B-} + \|\|P_c\|\|$	W[2]-h	Thm. 9 (v)
$k + \text{maxsize}^r_{B+} + \#\text{non-Horn}^r + \text{maxocc}^r_{H,B-} + \#\text{at}_{B+}$	W[2]-h	Thm. 9 (vi)
$\text{maxsize}^r_{H,B+,B-} + \text{maxsize}^r_{H,B-} + \text{maxsize}^r_{B+} + \#\text{at}_{B+} + {}$ $+ \text{maxsize}^c_{B-} + \|\|P_c\|\|$	paraNP-h	Thm. 9 (i)
$\text{maxsize}_{H,B+,B-} + \text{maxsize}^r_{H,B-} + \text{maxsize}^r_{B+} + \#\text{at}_{B-}$	paraNP-h	Thm. 9 (ii)

Proposition 8 ([46]) WMMSAT *is paraNP-hard when parameterized by the following parameter combination*

(i) $d + d^+ + d^- + p + v^- + d^+_\pi + ||\pi||$;

WMMSAT *is W[2]-hard when parameterized by each of the following parameter combination*

(ii) $k + d^- + v^- + d^+_\pi + ||\pi||$ *and*
(iii) $k + d^- + h + p + v^-$;

WMMSAT *is W[1]-hard when parameterized by the following parameter combination*

(iv) $k + d^- + h + p + d^+_\pi + ||\pi||$,

where k is the maximum weight of the minimal model, d is the maximum clause size, d^+ *and* d^- *are the maximum positive or negative clause size, respectively, h is the number of non-horn clauses, p is the maximum number of positive occurrences of a variable in* φ*,* v^+ *and* v^- *are the numbers of variables that occur as positive or negative literals in* φ *or in* π*, respectively,* $||\pi||$ *is the size of* π*, i.e., the total number of variable occurrences in* π *(see also Table 1).*

Theorem 9 *Let* $L \in \{k$-CONSISTENCY, k-BRAVE REASONING$\}$. *Then L is paraNP-hard when parameterized by each of the following parameter combinations*

(i) $\text{maxsize}^r_{H,B^+,B^-} + \text{maxsize}^r_{H,B^-} + \text{maxsize}^r_{B^+} + \#at_{B^+} + \text{maxsize}^c_{B^-} + ||P_c||$, *and*
(ii) $\text{maxsize}^r_{H,B^+,B^-} + \text{maxsize}^r_{H,B^-} + \text{maxsize}^r_{B^+} + \#at_{B^-}$.

L is W[2]-hard when parameterized by each of the following parameter combinations

(iii) $k + \text{maxsize}^r_{H,B^+} + \text{maxocc}^r_{B^+} + \text{maxsize}^c_{B^+} + \text{maxsize}^c_{B^-}$,
(iv) $k + \text{maxsize}^r_{B^-} + \text{maxsize}^c_{B^+} + \text{maxsize}^c_{B^-}$,
(v) $k + \text{maxsize}^r_{B^+} + \#at_{B^+} + \text{maxsize}^c_{B^-} + ||P_c||$, *and*
(vi) $k + \text{maxsize}^r_{B^+} + \#\text{non-Horn}^r + \text{maxocc}^r_{H,B^-} + \#at_{B^+}$.

L is W[1]-hard when parameterized by the following parameter combination

(vii) $k + \text{maxsize}^r_{B^+} + \#\text{non-Horn}^r + \text{maxocc}^r_{H,B^-} + \text{maxsize}^c_{B^-} + ||P_c||$.

Proof We proceed by a reduction from the problem WMMSAT for Statements (i) and (v)–(vii) and WSAT$_\leq$ for Statement (iv). Statement (ii) has already been established by Truszczyński [60]. Statement (iii) is an immediate consequence from a reduction established by Lonc and Truszczyński [49, Theorem 4.4].

Let (φ, π, k) be an instance of WMMSAT. We assume w.l.o.g.that φ contains no clauses without positive literals, since otherwise we can shift such clauses into π without affecting the size of the models and hence the minimality.[1] We now construct an instance (P, k) of k-CONSISTENCY as follows. For a clause C and $i \in \{0, 1\}$ we define

$$C^i := \{ a^i : x^i \in C, x \in var(C) \}$$

[1]Note that this has also no effect to the results we use for WMMSAT, since the parameters used in the proofs for WMMSAT remain unaffected (it only effects d and d^-, however, there d^- is already bounded by v^-; see the proofs of Theorems 16 and 17 in the work by Lackner and Pfandler [46]).

where a is a fresh atom and $a^0 = \neg a$ and $a^1 = a$. Now, let

$$P^{\min} := \{ C^1 \leftarrow C^0 : C \in \varphi \}$$

and

$$P^{\text{cons}} := \{ \leftarrow \neg C^1, C^0 : C \in \pi \}$$

and we define a program

$$P := P^{\min} \cup P^{\text{cons}}.$$

Next, we show that φ has a minimal model X of size at most k such that X is also a model of π if and only if P has an answer set of size at most k.
($\Rightarrow$): Let X be a minimal model of φ of size at most k such that X is also a model of π. Further, let $M := X^1$. For every rule $r \in P^{\min}$, there is a corresponding clause $C \in \varphi$. Since for each clause $C \in \varphi$ it is true that (i) $C^1 \cap M \neq \emptyset$, or (ii) $C^0 \setminus M \neq \emptyset$, we obtain by construction of $P^{\min}$ that (i) $H(r) \cap M \neq \emptyset$, or (ii) $B^+(r) \setminus M \neq \emptyset$ holds. Hence, M is a model of $P^{\min}$. Since $(P^{\min})^M = P^{\min}$, the set M is also a model of $(P^{\min})^M$. For every rule $r \in P^{\text{cons}}$, there is a corresponding clause $C \in \pi$. Since for each clause $C \in \varphi$ it holds that (i) $C^1 \cap M \neq \emptyset$, or (ii) $C^0 \setminus M \neq \emptyset$, we have by construction of P^{cons} that (i) $B^-(r) \cap M \neq \emptyset$, or (ii) $B^+(r) \setminus M \neq \emptyset$. Hence, M is a model of P^{cons}. Then, for every rule $r \in P^{\text{cons}}$ there is either (i) a corresponding rule $r' \in (P^{\text{cons}})^M$ with $B^+(r) = B^+(r')$ and $B^+(r') \setminus M \neq \emptyset$, since $B^+(r) \setminus M \neq \emptyset$, or (ii) $B^-(r) \setminus M \neq \emptyset$ and the rule r has been removed from P^{cons} when constructing $(P^{\text{cons}})^M$. Consequently, M is also a model of $(P^{\text{cons}})^M$. It remains to observe that M is also a minimal model of P^M. For proof by contradiction assume that P^M has a model N such that $N \subsetneq M$. Let now $r \in P^M$. Then, by the construction of P there is a corresponding clause C_r such that either (i) $C_r \in \varphi$, $C_r^1 = H(r)$ and $C_r^0 = B^+(r)$, or (ii) $C_r \in \pi$ and $C_r^0 = B^+(r)$. Since N is a model of P for every rule $r \in P^M$, it holds that $H(r) \cap N \neq \emptyset$ or $B^+(r) \setminus N \neq \emptyset$. Thus we can conclude in Case (i) $C_r^1 \cap N \neq \emptyset$ or $C_r^0 \setminus N \neq \emptyset$ and thus N is also a model of φ, which however contradicts the assumption that X is a minimal model of φ. Further, we can conclude in Case (ii) that $C_r^0 \setminus N \neq \emptyset$, which however contradicts the assumption that X is a model of π. Consequently, M is an answer set of P of size at most k.
($\Leftarrow$): Conversely, assume that M is an answer set of P of size at most k. Let X consist of the corresponding variables of M in $var(\pi) \cup var(\varphi)$. For each rule $r \in P$ there is a corresponding clause (i) $C_r \in \pi$ such that $C_r^0 = B^+(r)$ and $C_r^1 = B^-(r)$ if $H(r) = \emptyset$, or (ii) $C_r \in \varphi$ such that $C_r^1 = H(r)$ and $C_r^0 = B^+(r)$ if $B^-(r) = \emptyset$. We proceed with Case (i): By definition of an answer set, M is a model of P. Hence, for rules where $H(r) = \emptyset$, we have $B^+(r) \setminus M \neq \emptyset$ or $B^-(r) \cap M \neq \emptyset$. Thus, we obtain $C_r^1 \cap M \neq \emptyset$ or $C_r^0 \setminus M \neq \emptyset$, which yields that X is a model of π. We proceed with Case (ii): By definition of an answer set, the set M is a model of P. Hence, for rules where $H(r) \neq \emptyset$, we have $H(r) \cap M \neq \emptyset$ or $B^+(r) \setminus M \neq \emptyset$. Since $H(r) = C_r^1$ and $B^+(r) = C_r^0$, we have $C_r^1 \cap M \neq \emptyset$ or $C_r^0 \setminus M \neq \emptyset$. Hence, X is a model of φ. For proof by contradiction assume that there is some model Y of φ such that $Y \subsetneq X$ and Y is also a model of π. By construction of P for a clause $C \in \varphi$ there is a corresponding rule $r_c \in P$ such that $H(r_c) = C^1$, $B^+(r_c) = C^0$, and $B^-(r_c) = \emptyset$. Since $B^-(r') = \emptyset$ for every rule $r \in P^{\min}$, we have that N is also a model of $(P^{\min})^M$, which contradicts the assumption that M is an answer set of P. Further, by construction of P for a clause $C \in \pi$ there is a corresponding rule $r_c \in P$ such that $H(r_c) = \emptyset$, $B^+(r_c) = C^0$, and $B^-(r_c) = C^1$. Since Y is a model of π we conclude that (i) $B^-(r_c) \cap N \neq \emptyset$, or (ii) $B^+(r_c) \setminus N \neq \emptyset$. Hence, N is also a model of $(P^{\text{cons}})^M$. Thus, by Statement (ii) of Observation 2 the set N is also an answer set of P, which contradicts our assumption. Consequently, X is a minimal model of φ, has size at most k, and is also a model of π.

We have established the claim that φ has a minimal model X of size at most k such that X is also a model of π if and only if P has an answer set of size at most k.

Next, we can employ the construction and proofs from above to establish a reduction from an instance (φ, k) of WSAT$_{\leq}$ for Statement (iv). Note that WSAT$_{\leq}$ is a well known to be W[2]-hard, e.g., [13]. Therefore, observe that φ has a model M of size at most k if and only if $P^{\min}$ has an answer set of size at most k.

Finally, it remains to observe that our reduction preserves the parameters:

- k: directly corresponds to the maximum weight of a minimal model (k)
- $\text{maxsize}^r_{H,B^+,B^-}$: directly corresponds to the maximum clause size in φ (d)
- $\text{maxsize}^r_{H,B^-}$: directly corresponds to the maximum positive clause size in φ (d^+)
- $\text{maxsize}^r_{B^+}$: directly corresponds to the maximum negative clause size in φ (d^-)
- $\#\text{non-Horn}^r$: directly corresponds to the number of non-Horn clauses (h)
- maxocc^r_{H,B^-}: directly corresponds to the maximum number of positive occurrences of a variable in φ (p)
- $\#\text{at}_{B^+}$: directly corresponds to the number of variables that occur as negative literals in φ or in π (v^-)
- $\text{maxsize}^c_{B^-}$: directly corresponds to the maximum positive clause size in π (d^+_π)
- $||P_c||$: directly corresponds to the size of π, i.e., the total number of variable occurrences in π ($\|\pi\|$)

Fixed-parameter tractability follows from the results by Lackner and Pfandler [46] for WMMSAT as stated in Proposition 8. We obtain hardness for k-BRAVE REASONING by the same arguments. This concludes the proof. □

4 Membership results

In this section, we present several novel fixed-parameter tractability results for ASP reasoning problems, which are summarized in Table 3. Observe that fpt results for a combination of parameters trivially imply fpt results for any superset of these parameters.

We first observe that parameterizing in the number of head atoms already yields fixed-parameter tractability.

Observation 10 Each problem $L \in \{k$-CONSISTENCY, k-BRAVE REASONING$\}$ is fixed-parameter tractable when parameterized by each of the following parameters (i) $\#\text{at}_H$ and (ii) $\text{maxsize}_H + |P_r|$ where $|P_r|$ is the number of rules in P_r.

Proof First we show Statement (i) for k-CONSISTENCY. Therefore, let $h := \#\text{at}_H$. By Statement (ii) of Observation 2 for every answer set $M \in \text{AS}_k(P)$ holds that $M \subseteq \bigcup_{r \in P} H(r)$. Hence, we use a simple bounded search tree approach.

Therefore, we use slightly extended concepts from earlier work [26, 27] and define the concept of a reduct under sets M and N of atoms. Intuitively, M contains atoms that are interpreted as set to true and N contains atoms that are interpreted as set to false. Let P be a program, $M \subseteq \text{at}(P)$, and $N \subseteq \text{at}(P) \setminus M$. The *reduct of program P under (M, N)* is the logic program $P_{M,N}$ obtained from program P by (i) removing all rules r with $H(r) \cap M \neq \emptyset$; (ii) removing all rules r with $B^+(r) \cap N \neq \emptyset$; (iii) removing all rules r with $B^-(r) \cap M \neq \emptyset$; (iv) removing from the heads and negative bodies of the remaining rules all atoms a with $a \in N$; and (v) removing from the positive bodies of the remaining rules all atoms a with $a \in M$.

We construct a complete binary search tree T of depth h.

(1) Therefore, we label the root of the tree with the triple $(P, \emptyset, \emptyset)$.
(2) Then, we label the remaining nodes of the tree recursively as follows: Let (R, M, N) be the label of a node t of T whose two children are not labeled yet. Choose an atom $a \in \text{at}(P)_H \setminus (M \cup N)$.

   (i) Label the left child of t with $(R_{M,N}, M \cup \{a\}, N)$.
   (ii) Label the right child of t with $(R_{M,N}, M, N \cup \{a\})$.

(3) If there exists a node labeled with (R, M, N) such that R has no rules, then M is a model of P, c.f., [25, 27]. It remains to check whether there is some $M' \subsetneq M$ such that M' is a model of P^M and M is of size at most k. Therefore,

   (a) we check for each $M' \subsetneq M$ whether M' is still a model of P^M; if so, we discard M; and
   (b) we check whether $|M| \leq k$; if the answer is no, we can discard M otherwise, M is an answer set of P of size at most k.

Since the depth of T is bounded by h, the size of M is at most h. We conclude that the above algorithm solves the problem k-Consistency in time $\mathcal{O}(2^{2^h} \cdot h \cdot n^c)$ for $n = |\text{at}(P)|$ and some constant c. Statement (ii) follows directly from the previous one as the value $\#\text{at}(P)$ is bounded by $\text{maxsize}_H \cdot |P_r|$.

Note that this results trivially extends to k-Brave Reasoning by adding a rule that consists of an empty head, an empty positive body and a negative body that contains only the atom we are interested in. Further, we would like to mention that partial evaluation techniques are frequently used in the literature of ASP, however, there the minimality check is often done in terms of the completion or loop formulas (cf. [32, 58]). □

Observation 11 For each problem $L \in$ {k-Consistency, k-Brave Reasoning} we have $L \in$ XP when parameterized by each of the following parameters (i) k and (ii) $|P_r|$ where $|P_r|$ is the number of rules in P_r.

Proof Let P be a program and n be an integer such that $n = |\text{at}(P)|$. For Statement (i) let further $k > 0$ be some integer. For Statement (ii) let k be an integer with $k = |P_r|$. By Statement (iii) of Observation 2 the size of an answer set is at most k. In both cases, we have at most $\sum_{i=1}^{k} \binom{n}{k}$ answer sets of size at most k. For each of these answer set candidates, the minimality check can be done in time $\mathcal{O}(2^k)$ by first checking whether the candidate is a model and then trying all smaller models. Since $\binom{n}{k} \leq \frac{n^k}{k!}$ is true, the algorithm runs in time $\mathcal{O}(n^k)$. Consequently, the observation is established. □

4.1 Negation-free programs

Lackner and Pfandler [46] presented several fixed-parameter tractability results that turn out to be useful for showing fixed-parameter tractability for k-Consistency and k-Brave Reasoning. In order to use their results, we first establish an fpt-reduction from the ASP problems to WMMSat for a restricted version of programs, namely, when the main part of the input is restricted to programs with negation-free non-constraint rules plus arbitrary constraints (**NF+Cons**). In fact, the problem Consistency is already Σ_2^P-complete when the main part of the input is restricted to programs from **NF+Cons** [15].

Lemma 12 *Let* $L \in \{k\text{-Consistency}, k\text{-Brave Reasoning}\}$ *and* $(x, y) \in \mathbb{N} \times \mathbb{N}$ *where x is a parameter for L and y a parameter for* WMMSAT. *If the main part of the input of L is restricted to programs from* **NF+Cons**, *then there is an fpt-reduction from L to* WMMSAT *such that x preserves y, i.e., the value of y only depends on x, for all of the following pairs* (x, y)

(i) (k, k),
(ii) $(\#at_{B^+}, v^-)$,
(iii) $(\#\text{non-Horn}^r, h)$,
(iv) $(\text{maxsize}^r_{H,B^-}, d^+)$,
(v) $(\text{maxsize}^r_{B^+}, d^-)$,
(vi) $(\text{maxocc}^r_{H,B^-}, p)$, *and*
(vii) $(\text{maxsize}^c_{B^-}, d^+_\pi)$.

Proof We give a reduction to WMMSAT, which preserves all parameters considered in the statement. Therefore, we use ideas from the construction in the proof of Theorem 9 for the opposite direction.

Let (P, k) be an instance of k-CONSISTENCY where $P \in$ **NF+Cons**. We now construct an instance (φ, π, k) of WMMSAT as follows. The variables of the CNF formulas φ and π will consist of a variable for each atom of P. Then for a rule $r \in P$ we let

$$C(r) := \{\, x_a : a \in H(r) \cup B^-(r) \,\} \cup \{\, \neg x_a : a \in B^+(r) \,\}.$$

Further, we define

$$\varphi := \{\, C(r) : r \in P, H(r) \neq \emptyset \,\}$$

and

$$\pi := \{\, C(r) : r \in P, H(r) = \emptyset \,\}.$$

Then, we show that formula φ has a minimal model M of size at most k such that M is also a model of π if and only if P has an answer set of size at most k. We can use the exact same construction as in the proof of Theorem 9 to establish the statement, since the program P^{cons} consists only of constraint rules and program P^{min} consists only of non-constraint rules.

Next, we observe that our reduction preserves the parameters:

(i) k: directly corresponds to the maximum weight of a minimal model (k),
(ii) $\#\text{at}_{B^+}$: directly corresponds to the number of variables that occur as negative literals in π or φ (v^-),
(iii) $\#\text{non-Horn}^r$: directly corresponds to the number of non-horn clauses (h),
(iv) $\text{maxsize}^r_{H,B^-}$: directly corresponds to the maximum positive clause size in φ (d^+)
(v) $\text{maxsize}^r_{B^+}$: directly corresponds to the maximum negative clause size in φ (d^-)
(vi) maxocc^r_{H,B^-}: directly corresponds to the maximum number of positive occurrences of a variable in π (p),
(vii) $\text{maxsize}^c_{B^-}$: directly corresponds to the maximum positive clause size in π (d^+_π),

We obtain fpt-reductions for k-BRAVE REASONING by the same arguments. This concludes the proof. □

Remark 13 We would like to mention that, using the reductions above, instances from WMMSAT and k-CONSISTENCY restricted to **NF+Cons** coincide. More precisely, the proofs give a linear time reduction that transforms an instance from WMMSAT into an instance of **NF+Cons** from k-CONSISTENCY and vice versa. In particular, our reductions make certain concepts of parameters in the setting of answer set programming such as fpt-results for acyclicity-based backdoors [25] or bounded treewidth [20, 21] directly accessible to WMMSAT.

The following proposition states a result by Lackner and Pfandler for WMMSAT, which we will later use in our proofs.

Proposition 14 ([46]) WMMSAT *is fixed-parameter tractable when parameterized by each of the following parameter combinations*

(i) $k + d^+$,
(ii) $d^+ + h$,
(iii) $k + v^- + p + d^+_\pi$, *and*
(iv) $v^- + h + d^+_\pi$,

where k is the maximum weight of the minimal model, d^+ is the maximum positive clause size, h is the number of non-Horn clauses, p is the maximum number of positive occurrences of a variable in π, v^- is the number of variables that occur as negative literals in φ or in π, and d^+_π maximum positive clause size in π.

From Proposition 14 and the reduction in Lemma 12 that preserves all parameter combinations, we immediately obtain the following result.

Proposition 15 *Let $L \in$ {k-CONSISTENCY, k-BRAVE REASONING}. If the main part of the input of L is restricted to programs from* **NF+Cons**, *then L is fixed-parameter tractable when parameterized by each of the following parameter combinations*

(i) $k + \text{maxsize}^r_{H,B^-}$,
(ii) $\#\text{non-Horn}^r + \text{maxsize}^r_{H,B^-}$,
(iii) $k + \#at_{B^+} + \text{maxocc}^r_{H,B^-} + \text{maxsize}^c_{B^-}$, *and*
(iv) $\#at_{B^+} + \#\text{non-Horn}^r + \text{maxsize}^c_{B^-}$.

Now, we consider how and under which conditions we can extend the above results to arbitrary programs. To this end, we construct fpt-reductions from arbitrary programs to programs that contain only negation-free non-constraint rules and arbitrary constraints (**NF+Cons**) while preserving certain parameters.

We obtain the following two stronger tractability results.

Theorem 16 *Let $L \in$ {k-CONSISTENCY, k-BRAVE REASONING}. Then, L is fixed-parameter tractable when parameterized by each of the following parameter combinations*

(i) $k + \text{maxsize}^r_{H,B^-}$ *and*
(ii) $\#\text{non-Horn}^r + \text{maxsize}^r_{H,B^-}$.

To show the statement, we first present the main idea of the proof by providing intuition and defining the reduction. Based on this reduction we then show two lemmas which in a next step will help to show the correctness.

Proof-idea and reduction The main idea of the proof is to shift negations in non-constraint rules and use the reduction in Lemma 12 to WMMSAT. WMMSAT is fixed-parameter tractable when parameterized by the parameter combination stated in Proposition 14. Our reduction runs in linear time and preserves all necessary parameters.

In more detail, for a program P this reduction consists of two steps:

- First, from P we construct in linear time a program $P^{\text{nshift}} \in$ **NF+Cons** in the following way. Let $P^{\text{nshift}} := P^{\text{pos}} \cup P^{\text{supset}}$ where

$$\begin{aligned} P^{\text{pos}} &:= P^{\text{min}} \cup P^{\text{subset}} \\ P^{\text{min}} &:= \left\{ H(r) \cup \left(B^-(r)\right)' \leftarrow B^+(r) : r \in P \right\} \\ P^{\text{subset}} &:= \left\{ a' \leftarrow a : a \in \text{at}(P) \right\} \\ P^{\text{supset}} &:= \left\{ \leftarrow a', \neg a : a \in \text{at}(P) \right\}. \end{aligned}$$

- Second, from P^{nshift} we construct in linear time an instance of WMMSAT using the reduction presented in the proof of Lemma 12.

To establish the correctness of this two-step reduction, we first show the following two lemmas.

For this, let P be a program and for a set X we denote by the macro $(X)'$ the set $\{a' \mid a \in X\}$.

Lemma 17 *Given a program P, we can construct in linear time a program $P^{nshift} = P^{pos} \cup P^{supset}$ over atoms $at(P) \cup (at(P))'$, as defined previously, such that every model O of P^{nshift} is of the form $O = M \cup M'$ where $M \subseteq at(P)$, $M' \subseteq (at(P))'$, and $M' = (M)'$.*

Proof It is straightforward to see that this reduction runs in linear time in the size of P. We show that a model O of P^{shift} is of the form $O = M \cup M'$ where $M \subseteq \text{at}(P)$, $M' \subseteq (\text{at}(P))'$, and $M = (M)'$.

Let O be a model of P^{nshift}. Let M and M' be sets such that $M \mathbin{\dot\cup} M' = O$, $M \subseteq \text{at}(P)$, and $M' \subseteq (\text{at}(P))'$. By definition O is also a model of P^{subset} and P^{supset}. Then, for every $a \in \text{at}(P)$ there is rule $r \in P^{\text{subset}}$ of the form $a' \leftarrow a$. By definition r is satisfied if (i) $H(r) \cap M' \neq \emptyset$ or (ii) $B^+(r) \setminus M \neq \emptyset$. Since $M \cup M'$ satisfies r, we conclude the following. If $a \in M$, we have that $a' \in M'$. If $a' \notin M'$, we have that $a \notin M$. Consequently, it remains to exclude the case $a' \in M$ and $a \notin M$. For every $a \in \text{at}(P)$ there is a rule $r' \in P^{\text{supset}}$ of the form $\leftarrow a', \neg a$. By definition r' is satisfied if (i) $B^+(r') \setminus M' \neq \emptyset$ or (ii) $B^-(r') \cap M \neq \emptyset$. Since $M \cup M'$ satisfies r, we conclude that if $a' \in M'$, we have that $a \in M$. Hence, we established that $(M)' = M'$. □

Lemma 18 *A set $M \subseteq at(P)$ is an answer set of a program P if and only if $M \cup (M)'$ is an answer set of P^{nshift} where P^{nshift} is defined in the reduction given before Lemma 17. Moreover, P^{pos} is negation-free and P^{supset} contains only constraints.*

Proof We show that M is an answer set of P if and only if $M \cup (M)'$ is an answer set of P^{nshift}.

($\Rightarrow$): Let M be an answer set of P and $M' := (M)' = \{a' : a \in M\}$.

Since $a \in M$ if and only if $a' \in M'$, $M \cup M'$ is also a model of P^{subset} and P^{supset}. By definition of an answer set M is a model of P. Since in addition $a' \in M'$ if and only if $a \in M$, we have that $M \cup M'$ is a model of P^{min}.

It remains to show that $M \cup M'$ is an answer set of P^{nshift}. For proof by contradiction assume that there is some $O \subsetneq (M \cup M')$ such that O is a model of $(P^{\text{nshift}})^{M \cup M'}$. Let $N := (O \cap M)$ and $N' := (O \cap M')$, and hence, we have that $N \cup N' = O$. Since O is a model of P^{subset} and P^{supset}, for every $a \in \text{at}(P)$ rule $a' \leftarrow a$ ensures that if $a \in N$, then $a' \in N'$. In other words, $(N)' \subseteq N'$.

Next, we show that, however, then N is also a model of P^M, which yields a contradiction to the assumption that M is an answer set of P. Therefore, consider an arbitrarily chosen rule $r'' \in P^M$. Hence, by construction of the GL reduct, there is a rule $r' \in P$ such that $H(r'') = H(r')$, $B^+(r'') = B^+(r')$, and $B^-(r') \cap M = \emptyset$. By construction there is a rule $r \in P^{\min}$ such that $H(r) = H(r') \cup (B^-(r'))'$, $B^+(r) = B^+(r')$, and $B^-(r) = \emptyset$.

Since O is a model of $(P^{\min})^{M \cup M'}$, one of the following cases holds (i) $H(r) \cap O \neq \emptyset$ or (ii) $B^+(r) \setminus O \neq \emptyset$. If Case (i) holds, we know from $N = O \cap M$, $N' = O \cap M'$, and $H(r) = H(r') \cup (B^-(r'))'$, and $H(r) \cap N' = \emptyset$ that (ia) $B^-(r') \cap N' \neq \emptyset$ or (ib) $H(r') \cap N \neq \emptyset$ is true. Consider that (ia) $B^-(r') \cap N' \neq \emptyset$ holds. Since $N' \subsetneq (M)'$, we also have that $B^-(r') \cap M \neq \emptyset$. Consequently, the case is irrelevant. Consider that $B^-(r') \cap N' = \emptyset$ and (ib) $H(r') \cap N \neq \emptyset$ holds. Since $H(r') = H(r'')$ and $N \subsetneq M$, we have that $H(r'') \cap N \neq \emptyset$. Consequently, N also satisfies r''. If Case (ii) holds, by $B^+(r) \cap N' = \emptyset$, we have $B^+(r) \setminus N \neq \emptyset$. Since $B^+(r) = B^+(r') = B^+(r'')$, $B^+(r'') \cap N \neq \emptyset$. Consequently, N also satisfies r''. We conclude that N is also a model of P^M, which contradicts that M is minimal model of P^M and thus contradicts that M is an answer set of P. Hence, the only-if direction holds.

($\Leftarrow$): Let O be an answer set of $(P^{\text{nshift}})^O$ and $O := M \cup M'$, for $M \subseteq \text{at}(P)$ and $M' \subseteq (\text{at}(P))'$. Since O is a model of P^{nshift}, we know that $(M)' = M'$ is true according to the established Claim 1 above.

We first show that M is a model of P. Therefore, consider a rule $r' \in P$. By construction there is a rule $r \in P^{\min}$ such that $H(r) = H(r') \cup (B^-(r'))'$ and $B^+(r) = B^+(r')$. Since $M \cup M'$ is a model of r and $B^-(r) = \emptyset$, we have that one of the following is true (i) $H(r) \cap (M \cup M') \neq \emptyset$ or (ii) $B^+(r) \setminus (M \cup M') \neq \emptyset$. Assume that Case (i) holds. Since $(B^-(r'))' \cap M = \emptyset$ is true by construction of $P^{\min}$, we have that (ia) $H(r') \cap M \neq \emptyset$ or (ib) $(B^-(r'))' \cap M' \neq \emptyset$. If Case (ia) holds, M satisfies r'. If Case (ib) holds, we know from $(M)' = M'$ that $B^-(r') \cap M \neq \emptyset$ is true. Consequently, we conclude that M satisfies r'. Assume that Case (ii) holds. Since $(B^-(r'))' \cap M' = \emptyset$ is true by construction of $P^{\min}$ and $B^+(r) = B^-(r')$, we immediately obtain that $B^+(r') \setminus M \neq \emptyset$. We obtain that M satisfies r'. Both cases then yield that M is also a model of P.

It remains to show that M is also an answer set of P^M. For proof by contradiction assume that there is some $N \subsetneq M$ such that N is a model of P^M. Let $N' := (N)' = \{a' : a \in N\}$. We show that, however, then $N \cup N'$ is also a model of $(P^{\text{nshift}})^O$, which yields a contradiction to the assumption that M is an answer set of $(P^{\text{nshift}})^O$. Therefore, consider an arbitrarily chosen rule $r \in (P^{\text{nshift}})^O$. Since for every $a \in N$, $a' \in N'$ and vice versa. $N \cup N'$ satisfies every rule $r \in P^{\text{subset}} \cup P^{\text{supset}}$. Thus, assume that $r \in (P^{\min})^O$. Since $B^-(r) = \emptyset$, there is a rule $r' \in P$ such that $H(r) = H(r') \cup (B^-(r'))'$ and $B^+(r) = B^+(r')$ where $B^-(r) = \emptyset$. Further, by definition of the GL reduct, there is a rule $r'' \in P^M$ such that $H(r'') = H(r')$, $B^+(r'') = B^+(r')$, and $B^-(r') \cap M = \emptyset$ or there is no rule $r'' \in P^M$ because $B^-(r') \cap M \neq \emptyset$.

Since N is a model of P^M, one of the following cases holds (i) $H(r'') \cap N \neq \emptyset$, (ii) $B^+(r'') \setminus N \neq \emptyset$, or (iii) $B^-(r') \cap M \neq \emptyset$. Assume that Case (i) holds. Since $H(r) = H(r') \cup B^-(r')$ and $H(r') = H(r'')$, we obtain that $H(r) \cap N \neq \emptyset$. Thus, N also satisfies r. Assume that Case (ii) holds. Since $B^+(r'') \setminus N \neq \emptyset$ and $B^+(r) = B^+(r') = B^+(r'')$, we have that $B^+(r) \setminus N \neq \emptyset$. Hence, N also satisfies r. Assume that Case (iii) holds. Since $B^-(r) = \emptyset$, the rule r is not removed from $(P^{\text{nshift}})^O$ when constructing the GL reduct from P^{nshift}. As $B^-(r') \cap M \neq \emptyset$ and $H(r) = H(r') \cup B^-(r')$, we conclude that $H(r) \cap M \neq \emptyset$. Finally, since $N \subsetneq M$, we obtain that $H(r) \cap N \neq \emptyset$. Consequently, N also satisfies r. We conclude that N is also a model of $(P^{\text{nshift}})^O$, which contradicts the assumption that O is minimal model of $(P^{\text{nshift}})^O$ and thus contradicts that O is an answer set of P^{nshift}. Hence, we also established the if direction of the claim. □

After showing the previous two lemmas, we are ready to show the correctness of Theorem 16.

Proof (Correctness of Theorem 16) From Lemma 17, Lemma 18, and the reduction in Lemma 12 it immediately follows that we have a formula F_{pos}, which encodes P^{pos}, and a formula F_{supset}, which encodes P^{supset}. Then, P is a yes-instance, i.e., M is an answer set of P, if and only if the constructed WMMSAT instance with $\varphi = F_{\text{pos}}$ and $\pi = F_{\text{supset}}$ is a yes-instance, i.e., $V_{M \cup M'}$ is a minimal model of φ and $V_{M \cup M'}$ is a model of π where $V_{M \cup M'}$ is the set of variables that correspond to M and M'.

Using the reduction of the proof of Lemma 12, the WMMSAT instance is then given by $\varphi := F_{\text{pos}}$ and $\pi := F_{\text{supset}}$ for a formula that encodes P^{pos} and a formula that encodes P^{supset}. It remains to observe that the reduction preserves all parameters.

– $\text{maxsize}^r_{H,B^-}$: Let $d \geq 2$ be some integer. Moreover, assume that $\text{maxsize}^r_{H,B^-} \leq d$, by construction of F_{pos} each clause in F_{subset} contains at most 1 positive literal and the maximum number of positive literals in a clause of F_{min} is at most d. Moreover, each clause in F_{supset} contains at most 1 positive literal. Hence, maximum number of positive literals in each clause of the resulting formulas is at most d.
– k: Let $k \geq 0$ be some integer. Moreover, assume that $|M| \leq k$. By construction of F_{pos}, $M \subseteq \text{at}(P)$ is an answer set of P if and only if $V_{M \cup M'}$ is a minimal model of F_{pos} and $V_{M \cup M'}$ is a model of F_{supset}. Hence, we have $|V_{M \cup M'}| \leq 2k$ by construction. Consequently, the maximum weight of the minimal model of F_{pos} is bounded by $2k$.
– $\#\text{non-Horn}^r$: Let $h \geq 0$ be some integer and assume that $\#\text{non-Horn}^r \leq h$. By construction of F_{subset} and F_{supset} contain only Horn clauses. Moreover, a rule is not Horn if and only if the corresponding clause in F_{min} is not Horn. Hence, h provides an upper bound for the number of non-Horn clauses of F_{min} and thus of F_{pos} and F_{supset}.

Finally, we obtain according to Proposition 14 fixed-parameter tractability of the problem k-CONSISTENCY when parameterized by $k + \text{maxsize}^r_{H,B^-}$ or when parameterized by $\#\text{non-Horn} + \text{maxsize}^r_{H,B^-}$. We obtain membership for k-BRAVE REASONING by the same arguments. Hence, the statement of the theorem holds. □

Next, we lift Statements (iii) and (iv) of Proposition 15 to arbitrary programs. Recall that by definition programs in **NF+Cons** have an empty negative body for non-constraint rules and hence the parameter $\#\text{at}^r_{B^-}$ is of value 0. Then, we can see the parameter $\#\text{at}^r_{B^-}$ as an immediate measure for the number of GL reducts of non-constraint rules.

Theorem 19 *Let* $L \in \{k\text{-CONSISTENCY}, k\text{-BRAVE REASONING}\}$. *Then* L *is fixed-parameter tractable when parameterized by each of the following parameter combinations*

(i) $k + \#at_{B^+} + \text{maxocc}^r_{H,B^-} + \text{maxsize}^c_{B^-} + \#at^r_{B^-}$ *and*
(ii) $\#at_{B^+} + \#\text{non-Horn}^r + \text{maxsize}^c_{B^-} + \#at^r_{B^-}$.

Proof In order to decide k-CONSISTENCY, we give an fpt-reduction that constructs $2^{\#\text{at}^r_{B^-}}$ many programs that can be solved in fpt-time using results established in Proposition 15.

To this end, let (P, ℓ) be an instance of k-CONSISTENCY, $N := \cup_{r \in P_r} \left(\text{at}^r_{B^-}\right)$, $M_1 \subseteq N$, and $M_0 := N \setminus M_1$. Further, we define the *partial GL reduct* of a program P under the tuple (M_0, M_1) of disjoint sets of atoms as the program $P^{(M_0,M_1)} := \{ H(r) \leftarrow B^+(r), \neg(B^-(r) \setminus M_0) : B^-(r) \cap M_1 = \emptyset \}$. Then, we set $P^c_{M_1,M_0} := \{ \bot \leftarrow \neg a : a \in M_1 \} \cup \{ \bot \leftarrow a : a \in M_0 \}$ and let $P[M_0, M_1] := P^{(M_0,M_1)} \cup P^c_{M_1,M_0}$.

The program P has an answer set of size at most k if and only if at least one program $P[M_0, M_1]$ has an answer set of size at most $k - |M_1|$. Therefore, observe that $\mathrm{AS}(P) = \{ \mathrm{AS}(P[M_0, M_1]) : M_1 \subseteq N, M_0 = N \setminus M_1 \}$ is true from the definitions of an answer set and the construction of $P[M_0, M_1]$ (see also Proposition 4 in earlier work [28]). In this way, we give a reduction to $2^{\#\mathrm{at}^r_{B^-}}$ many instances of k-CONSISTENCY that consists of $2^{\#\mathrm{at}^r_{B^-}}$ many subprograms by constructing partial GL reducts under a set M_1 and a set M_0 together with constraints that enforce that any minimal model M of the GL reduct satisfies that atoms in M_1 belong to M and atoms in M_0 do not belong to M. Recall that the set M_1 consists of atoms that we have set to true and M_0 consists of atoms that we have set to false.

It remains to observe that our reduction preserves the parameters:

- k remains unaffected,
- $\#\mathrm{at}_{B^+}(P[M_0, M_1]) = \#\mathrm{at}_{B^+}(P) + \#\mathrm{at}^r_{B^-}(P)$,
- $\#\text{non-Horn}^r(P[M_0, M_1]) = \#\text{non-Horn}^r(P)$,
- $\mathrm{maxocc}^r_{H,B^-}(P[M_0, M_1]) \leq \mathrm{maxocc}^r_{H,B^-}(P)$, and
- $\mathrm{maxsize}^c_{B^-}(P[M_0, M_1]) = \max\{ \mathrm{maxsize}^c_{B^-}(P), 1 \}$.

Since our algorithm constructs $2^{\#\mathrm{at}^r_{B^-}}$ many programs that can be solved in fpt-time according to Proposition 15, our algorithm runs in fpt-time. We obtain membership for k-BRAVE REASONING by the same arguments. Hence, the theorem follows. ☐

5 Discussion

The reduction in the proof of Theorem 19 states that ASP and WMMSAT are very related with respect to the considered reasoning problems. However, answer sets additionally require minimality with respect to the GL reduct of the given program. In consequence, we need to parameterize additionally in the number of negative atoms that occur in non-constraint rules of the given program. Particularly, we do not have a direct counterpart of the concept of a compact representation for atoms in the head (see the concept of SSMs in [46]) if the positive body is empty and the negative body is not empty. An alternative way to parameterize in the number of negative atoms that occur in non-constraint rules would be a transformation that compiles negative atoms away while the blowup is bounded by a function of the parameter. There are already results in the literature that provide transformations from programs that contain negations into programs with few negations, however with a slightly different focus. Eiter and Polleres [16] have established a reduction from head-cycle free programs, which are disjunctive programs where certain cyclic dependencies are forbidden and where the complexity of the problem CONSISTENCY is still in NP. Janhunen [40] considered programs and transformations between classes of programs if default negation is also allowed in the heads of disjunctive rules.

The next result states that a fixed-parameter tractability result for the ENUM problem directly extends to a fixed-parameter tractability result with the same parameter for our considered ASP reasoning problems, where we are interested only in answer sets of size at most k.

Proposition 20 *Let p be an ASP parameter. If the problem* ENUM *is fixed-parameter tractable when parameterized by p, then for every problem $L \in \{k$-*CONSISTENCY, k-BRAVE REASONING$\}$, *L is fixed-parameter tractable when parameterized by p.*

Proof If the problem ENUM is fixed-parameter tractable when parameterized by p, then the problem k-ENUM is fixed-parameter tractable with respect to p. Thus, we can simply enumerate all answer sets of size at most k in fpt-time and decide any of the listed problems in fpt-time. Hence, the claim holds. □

5.1 Backdoors

Known results for backdoors [26] immediately apply to our problems in k-$\mathcal{A}sp\mathcal{P}roblems$. Backdoors have been defined as follows. Let $\mathcal{C}$ be a class of programs. A set X of atoms is a *strong $\mathcal{C}$-backdoor* of a program P if $P_{M,N} \in \mathcal{C}$ for all sets $M \subseteq X$ and $N = X \setminus M$. A class $\mathcal{C}$ of programs is *enumerable* if for each $P \in \mathcal{C}$ we can compute $\mathrm{AS}(P)$ in polynomial time. Intuitively, if the class of programs is enumerable it has a small number of answer sets, which can then be used to easily answer various reasoning questions.

Proposition 21 ([26]) *Let $\mathcal{C}$ be an enumerable class of normal programs. The problem* ENUM *is fixed-parameter tractable when parameterized by the size of a strong $\mathcal{C}$-backdoor.*

Corollary 22 *Let $\mathcal{C}$ be an enumerable class of normal programs. Every problem $L \in$* $\{k$-CONSISTENCY, k-BRAVE REASONING$\}$ *is fixed-parameter tractable when parameterized by the size of a strong $\mathcal{C}$-backdoor.*

5.2 Treewidth

Earlier work on the computational complexity of ASP problems when parameterized by treewidth [20, 21, 39] has considered these problems only when arbitrarily large answer sets are allowed. In this section, we comment on these results under the restriction that we are interested in answer sets of size at most k.

Graph representations Treewidth is a parameter that originates in graph theory. To employ treewidth in our setting we require standard definitions [20, 39] of a graph representation of a program as follows. The *primal graph G_P* of program P has the atoms of P as vertices and an edge $\{a, b\}$ if there exists a rule $r \in P$ and $a, b \in \mathrm{at}(r)$. The *incidence graph I_P* of P is the bipartite graph that has the atoms and rules of P as vertices and an edge $\{a, r\}$ if $a \in \mathrm{at}(r)$ for some rule $r \in P$.

Treewidth Intuitively, treewidth measures the closeness of a graph to a tree. The idea of exploiting treewidth is mainly based on the observation that problems on trees are often computationally easier to solve on trees than on arbitrary graphs. The definition of treewidth is based on so-called tree decompositions of graphs where sets of vertices of the input graph are arranged as labels (bags) at the nodes of a tree such that certain conditions are satisfied. The width of a tree decomposition is the size of a largest bag minus 1. Then, the treewidth of a graph is the width of an tree decomposition of smallest width. Next, we provide a formal definition for the parameter treewidth. Let $G = (V, E)$ be a graph, $T = (N, F, n)$ a rooted tree, and $\chi : N \rightarrow 2^V$ a function that maps each node $t \in N$ to a set of vertices. We call the sets $\chi(\cdot)$ *bags* and N the set of nodes. Then, the pair $\mathcal{T} = (T, \chi)$ is a *tree decomposition* of G if the following conditions hold:

1. for every vertex $v \in V$ there is a node $t \in N$ with $v \in \chi(t)$;
2. for every edge $e \in E$ there is a node $t \in N$ with $e \subseteq \chi(t)$; and
3. for any three nodes $t_1, t_2, t_3 \in N$, if t_2 lies on the unique path from t_1 to t_3, then $\chi(t_1) \cap \chi(t_3) \subseteq \chi(t_2)$.

We call $\max\{\, |\chi(t)| - 1 : t \in N \,\}$ the *width* of the tree decomposition. The *treewidth* $\text{tw}(G)$ of a graph G is the minimum width over all possible tree decompositions of G. Then, $\text{primtw}(P) := \text{tw}(G_P)$, which defines the parameter *primal treewidth*, or primtw for short. Further, $\text{inctw}(P) := \text{tw}(I_P)$ yielding the parameter *incidence treewidth*, or inctw for short.

The next result states that previous results for treewidth on answer set programming can be extended when we consider answer sets of size at most k.

Corollary 23 *Let $L \in \{k$-*CONSISTENCY*, k-*BRAVE REASONING*$\}$. Then, the problem L is fixed-parameter tractable when parameterized by each of the following parameters (i)* $\text{primtw} + k$ *and (ii)* $\text{inctw} + k$.

Proof (Sketch) The result follows from earlier work [20, 21]. In the presented dynamic programming algorithm for the dependency graph one stores tuples in the table for each node of the tree decomposition. Such a tuple consist of a set of atoms relevant for the SAT part of the problem and a family of sets of atoms relevant for the UNSAT part. Similar when solving the counting problem (see [20, Algorithm 4]), we can extend each tuple by an integer ℓ which expresses how many atoms have been set to true during the (bottom-up) traversal of the tree decomposition. Then, the algorithm discards results where $\ell > k$. This algorithm is an fpt-algorithm for the parameter $\text{primtw} + k$. For the parameter inctw, previous dynamic programming algorithms require larger tuple, which store additional information. However, we can proceed in a similar way as before by extending each tuple with an integer ℓ. Again, the algorithm discards results where $\ell > k$ which in turn gives an fpt-algorithm. This concludes the proof of the corollary. □

6 Conclusion

We have identified several natural structural parameters of ASP instances, which are summarized in Table 2. We have carried out a fine-grained complexity analysis of the main reasoning problems in answer set programming when parameterized by various combinations of these parameters. Our study also considers the parameterized complexity of the main ASP reasoning problems while taking the size of answer sets into account. Such a restriction is particularly interesting for applications that require small solutions. We have presented various hardness and membership results, which are outlined in Table 4 and detailed references to the results can be found in Table 2. Every hardness result of the reasoning problems when parameterized by a parameter combination also holds for any parameter that consists of a subset of the combination. Further, every fixed-parameter tractability result of the considered problems when parameterized by a parameter combination also holds for any extension of the parameter by additional structural properties (superset of the parameter combination). In that way, we have improved on the theoretical understanding by providing a novel multiparametric view on the parameterized complexity of ASP, which allows us to draw a detailed map for various ASP parameter combinations.

Table 4 Summary of multiparametric complexity results for k-CONSISTENCY. For each line the marked columns indicate according to the header a ASP parameter combination. Membership or hardness results are stated in the last column

		#at					maxsize					maxocc			
k	$\|\|P_c\|\|$	$\cdot^r_H$	$\cdot_{B+}$	$\cdot^r_{B-}$	$\cdot^c_{B-}$	nH	$\cdot^r_H$	$\cdot^r_{B+}$	$\cdot^r_{B-}$	$\cdot^c_{B+}$	$\cdot^c_{B-}$	$\cdot^r_H$	$\cdot^r_{B+}$	$\cdot^r_{B-}$	Result
		×													$\in$ FPT
×							×		×						$\in$ FPT
						×	×		×						$\in$ FPT
×			×	×							×	×		×	$\in$ FPT
			×	×		×					×				$\in$ FPT
×															$\in$ XP
×	×					×		×			×	×		×	W[1]-h
×							×	×		×	×		×		W[2]-h
×									×	×	×				W[2]-h
×	×		×					×			×				W[2]-h
×			×			×		×				×		×	W[2]-h
	×		×				×	×	×		×				paraNP-h
				×	×		×	×	×						paraNP-h
k	$\|\|P_c\|\|$	$\cdot^r_H$	$\cdot_{B+}$	$\cdot^r_{B-}$	$\cdot^c_{B-}$	nH	$\cdot^r_H$	$\cdot^r_{B+}$	$\cdot^r_{B-}$	$\cdot^c_{B+}$	$\cdot^c_{B-}$	$\cdot^r_H$	$\cdot^r_{B+}$	$\cdot^r_{B-}$	Result

Future work The results and concepts of this paper give rise to several research questions. For instance, it would be interesting to close the gap for the remaining parameter combinations. Therefore, we need to identify important corner cases. Another interesting question is whether we can drop the number of atoms in the negative body of non-constraint rules from Statements (i) and (ii) in Theorem 19 (k + $\#\text{at}_{B+}$ + $\text{maxocc}^r_{H,B-}$ + maxsize^c_{B-} or $\#\text{at}_{B+}$ + $\#\text{non-Horn}^r$ + maxsize^c_{B-}), which does not complete the picture but would give insights on how to reduce ASP to WMMSAT without $\#\text{at}^r_{B-}$ as parameter. Moreover, we think that it would also be interesting to consider ASP with extended rules such as choice rules for a systematic analysis. Currently, the size of the program for the counters as used in Definition 6 are fairly big. One can reduce the size by using BDD-style [14, 41] or cardinality network [6, 7] based counters. We think that a comprehensive analysis of whether these translations preserve the various parameters, as presented in our work and additional parameters as present in the literature, are interesting for future work. Another interesting further research direction is to study how the parameters empirically distribute among ASP instances from the last ASP competitions, in particular, in random versus structured instances. An interesting problem for multiparametric view on the computational complexity is counting in answer set programming and in particular projected counting [18, 22] as well as related frameworks such as argumentation [19]. Additionally, it would be interesting to conduct a parameterized analysis as well as considering multiple parameters in the non-ground setting. Finally, it might be interesting to consider multiple parameters for generalizations of ASP such as default logic [23, 24, 52].

References

1. Arora, S., Barak, B.: Computational Complexity: a Modern Approach. Cambridge University Press, Cambridge (2009)
2. Bliem, B., Moldovan, M., Morak, M., Woltran, S.: The impact of treewidth on ASP grounding and solving. In: Sierra, C. (ed.) Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17). pp. 852–858. The AAAI Press (2017)
3. Bliem, B., Ordyniak, S., Woltran, S.: Clique-width and directed width measures for answer-set programming. In: Fox, M., Kaminka, G. (eds.) Proceedings of the 22st Eureopean Conference on Artificial Intelligence (ECAI'16). Frontiers in Artificial Intelligence and Applications, vol. 285, pp. 1105–1113. IOS Press, The Hague (2016). extended version arXiv:1606.09449
4. Bodlaender, H.L., Downey, R.G., Fellows, M.R., Hallett, M.T., Wareham, H.T.: Parameterized complexity analysis in computational biology. Bioinformatics **11**(1), 49–57 (1995)
5. Bodlaender, H.L., Downey, R.G., Fellows, M.R., Wareham, H.T.: The parameterized complexity of sequence alignment and consensus. Theor. Comput. Sci. **147**(1&2), 31–54 (1995)
6. Bomanson, J., Gebser, M., Janhunen, T.: Improving the normalization of weight rules in answer set programs. In: Fermé, E., Leite, J. (eds.) Proceedings of the 14th European Conference on Logics in Artificial Intelligence (JELIA'14), Lecture Notes in Computer Science, vol. 8761, pp. 166–180. Springer, Funchal (2014)
7. Bomanson, J., Janhunen, T.: Normalizing cardinality rules using merging and sorting constructions. In: Cabalar, P., Son, T. (eds.) Proceedings of the 12th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'13). Lecture Notes in Computer Science, vol. 8148, pp. 187–199. Springer, Corunna (2013)
8. Courcelle, B.: Graph rewriting: an algebraic and logic approach. In: van Leeuwen, J. (ed.) Handbook of Theoretical Computer Science, Vol.B, Vol. Formal Models and Semantics, pp. 193–242. Elsevier Science Publishers, North-Holland (1990)
9. Creignou, N., Meier, A., Müller, J.S., Schmidt, J., Vollmer, H.: Paradigms for parameterized enumeration. Theor. Comput. Sci. **60**(4), 737–758 (2017)
10. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Dániel Marx, M.P., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer, Berlin (2015)
11. Dowling, W.F., Gallier, J.H.: Linear-time algorithms for testing the satisfiability of propositional Horn formulae. J. Log. Program. **1**(3), 267–284 (1984)
12. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Monographs in Computer Science. Springer, New York (1999)
13. Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity. Texts in Computer Science. Springer, London (2013)
14. Eén, N., Sörensson, N.: Translating pseudo-Boolean constraints into SAT. J. on Satisfiability Boolean Modeling and Computation **2**(1-4), 1–26 (2006)
15. Eiter, T., Gottlob, G.: On the computational cost of disjunctive logic programming: Propositional case. Ann. Math. Artif. Intell. **15**(3–4), 289–323 (1995)
16. Eiter, T., Polleres, A.: Towards automated integration of guess and check programs in answer set programming: a meta-interpreter and applications. Theory Pract. Logic Program. **6**(1-2), 23–60 (2006)
17. Fichte, J.K.: The good, the bad, and the odd: Cycles in answer-set programs. In: Lassiter, D., Slavkovik, M. (eds.) Proceedings of the 23rd European Summer School in Logic, Language and Information (ESSLLI'11) and in New Directions in Logic, Language and Computation (ESSLLI'10 and ESSLLI'11 Student Sessions, Selected Papers Series), Lecture Notes in Computer Science, vol. 7415, pp. 78–90. Springer, Berlin (2012)
18. Fichte, J.K., Hecher, M.: Exploiting treewidth for counting projected answer sets. In: Lierler, Y., Woltran, S. (eds.) Proceedings of the 15th International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR'19). Philadelphia, (2019). to appear
19. Fichte, J.K., Hecher, M., Meier, A.: Counting complexity for reasoning in abstract argumentation. In: Hentenryck, P.V., Zhou, Z. (eds.) Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI'19). The AAAI Press, Honolulu (2019)
20. Fichte, J.K., Hecher, M., Morak, M., Woltran, S.: Answer set solving with bounded treewidth revisited. In: Balduccini, M., Janhunen, T. (eds.) Proceedings of the 14th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'17). Lecture Notes in Computer Science, vol. 10377, pp. 132–145. Springer, Espoo (2017)
21. Fichte, J.K., Hecher, M., Morak, M., Woltran, S.: dynASP2.5: Dynamic programming on tree decompositions in action. In: Lokshtanov, D., Nishimura, N. (eds.) Proceedings of the 12th International Symposium on Parameterized and Exact Computation (IPEC'17). pp. 17:1–17:12. Leibniz International Proceedings in Informatics (LIPIcs), Dagstuhl Publishing (2017)

22. Fichte, J.K., Hecher, M., Morak, M., Woltran, S.: Exploiting treewidth for projected model counting and its limits. In: Beyersdorff, O., Wintersteiger, C. (eds.) Proceedings on the 21th International Conference on Theory and Applications of Satisfiability Testing (SAT'18). Lecture Notes in Computer Science, vol. 10929, pp. 165–184. Springer, Oxford (2018)
23. Fichte, J.K., Hecher, M., Schindler, I.: Default logic and bounded Treewidth. In: Klein, S.T., Martín-Vide, C., Shapira, D. (eds.) Proceedings of the 12th International Conference on Language and Automata Theory and Applications (LATA'18). Lecture Notes in Computer Science, vol. 10792, pp. 130–142. Springer, Ramat Gan (2018)
24. Fichte, J.K., Meier, A., Schindler, I.: Strong backdoors for default logic. In: Creignou, N., Le Berre, D. (eds.) Proceedings of the 19th International Conference Theory and Applications of Satisfiability Testing (SAT'16), pp. 45–59. Springer, Bordeaux (Jul 2016)
25. Fichte, J.K., Szeider, S.: Backdoors to normality for disjunctive logic programs. ACM Trans. Comput. Log. **17**(1), 7 (2015)
26. Fichte, J.K., Szeider, S.: Backdoors to tractable answer-set programming. Artif. Intell. **220**(0), 64–103 (2015)
27. Fichte, J.K., Szeider, S.: Backdoor trees for answer set programming. In: Bogaerts, B., Harrison, A. (eds.) Proceedings of the 10th Workshop on Answer Set Programming and Other Computing Paradigms co-located with the 14th International Conference on Logic Programming and Nonmonotonic Reasoning (ASPOCP@LPNMR'17), vol. 1868, pp. 9:1–14. CEUR Workshop Proceedings (CEUR-WS.org), Espoo (Jul 2017)
28. Fichte, J.K., Truszczyński, M., Woltran, S.: Dual-normal programs – the forgotten class. Theory and Practice of Logic Programming. In: proceedings of the 31st International Conference on Logic Programming (ICLP'15), pp. 495–510 (2015)
29. Flum, J., Grohe, M.: Describing parameterized complexity classes. Inf. Comput. **187**(2), 291–319 (2003)
30. Flum, J., Grohe, M.: Parameterized Complexity Theory Theoretical Computer Science, vol. XIV. Springer, Berlin (2006)
31. Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T.: Answer set solving in practice. Morgan & Claypool (2012)
32. Gebser, M., Schaub, T.: Tableau calculi for logic programs under answer set semantics. ACM Trans. Comput. Log. **14**(2), 15 (2013)
33. Gebser, M., Schaub, T., Thiele, S., Veber, P.: Detecting inconsistencies in large biological networks with answer set programming. Theory Pract. Logic Program. **11**(2-3), 323—360 (2011)
34. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Kowalski, R.A., Bowen, K. (eds.) Proceedings of the 5th International Conference and Symposium on Logic Programming (ICLP/SLP'88), vol. 2, pp. 1070–1080. MIT Press, Seattle (1988)
35. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. N. Gener. Comput. **9**(3/4), 365–386 (1991)
36. Gottlob, G., Pichler, R., Wei, F.: Bounded treewidth as a key to tractability of knowledge representation and reasoning. Artif. Intell. **174**(1), 105–132 (2010)
37. Gottlob, G., Scarcello, F., Sideri, M.: Fixed-parameter complexity in AI and nonmonotonic reasoning. Artif. Intell. **138**(1-2), 55–86 (2002)
38. Gottlob, G., Szeider, S.: Fixed-parameter algorithms for artificial intelligence, constraint satisfaction and database problems. Comput. J. **51**(3), 303–325 (2008)
39. Jakl, M., Pichler, R., Woltran, S.: Answer-set programming with bounded treewidth. In: Boutilier, C. (ed.) Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09), vol. 2, pp. 816–822. The AAAI Press, Pasadena (2009)
40. Janhunen, T.: On the effect of default negation on the expressiveness of disjunctive rules. In: Eiter, T., Faber, W., Truszczyński, M. (eds.) Proceedings of the 6th International Conference on Logic Programming and Nonmotonic Reasoning (LPNMR'01). Lecture Notes in Computer Science, vol. 2173, pp. 93–106. Springer, Vienna (2001)
41. Janhunen, T., Niemelä, I.: Compact translations of non-disjunctive answer set programs to propositional clauses. In: Balduccini, M., Son, T. (eds.) Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning – Essays Dedicated to Michael Gelfond on the Occasion of His 65Th Birthday, Lecture Notes in Artificial Intelligence, vol. 6565, pp. 111–130. Springer, Berlin (2011)
42. Janota, M., Marques-Silva, J.: On deciding mus membership with QBF. In: Lee, J. (ed.) Proceedings of the 17th International Conference on Principles and Practice of Constraint Programming (CP'11). Lecture Notes in Computer Science, vol. 6876, pp. 414–428. Springer, Perugia (2011)
43. Kronegger, M., Ordyniak, S., Pfandler, A.: Backdoors to planning. In: Brodley, C.E., Stone, P. (eds.) Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI'14), pp. 2300–2307. The AAAI Press, Québec City (2014)

44. Kronegger, M., Ordyniak, S., Pfandler, A.: Variable-deletion backdoors to planning. In: Bonet, B., Koenig, S. (eds.) Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15), pp. 3305–3312. The AAAI Press, Austin (2015)
45. Kronegger, M., Pfandler, A., Pichler, R.: Parameterized complexity of optimal planning: A detailed map. In: Rossi, F., Thrun, S. (eds.) Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13), pp. 954–961. The AAAI Press, Beijing (2013)
46. Lackner, M., Pfandler, A.: Fixed-parameter algorithms for finding minimal models. In: Eiter, T., McIlraith, S. (eds.) Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR'12), pp. 85–95. The AAAI Press, Rome (2012)
47. Lanctot, J.K., Li, M., Ma, B., Wang, S., Zhang, L.: Distinguishing string selection problems. Inf. Comput. **185**(1), 41–55 (2003)
48. Lifschitz, V.: Thirteen definitions of a stable model. In: Blass, A., Dershowitz, N., Reisig, W. (eds.) Fields of Logic and Computation, Lecture Notes in Computer Science, vol. 6300, pp. 488–503. Springer, Berlin (2010)
49. Lonc, Z., Truszczyński, M.: Fixed-parameter complexity of semantics for logic programs. ACM Trans. Comput. Log. **4**(1), 91–119 (2003)
50. Marek, V.W., Truszczyński, M.: Stable Models and an Alternative Logic Programming Paradigm. In: Apt, K.R., Marek, V.W., Truszczyński, M., Warren, D. (eds.) The Logic Programming Paradigm: a 25-Year Perspective, pp. 375–398. Artificial Intelligence, Springer, Berlin (1999)
51. Marx, D., Pilipczuk, M.: Everything you always wanted to know about the parameterized complexity of Subgraph Isomorphism (but were afraid to ask). In: Mayr, E.W., Portier, N. (eds.) Proceedings of the 31st International Symposium on Theoretical Aspects of Computer Science (STACS'14). Leibniz International Proceedings in Informatics (LIPIcs), vol. 25, pp. 542–553. Dagstuhl Publishing, Lyon (2014)
52. Meier, A., Schindler, I., Schmidt, J., Thomas, M., Vollmer, H.: On the parameterized complexity of non-monotonic logics. Arch. Math. Log. **54**(5-6), 685–710 (2015)
53. Papadimitriou, C.H.: Computational complexity. Addison-Wesley (1994)
54. Pearce, D.: A new logical characterisation of stable models and answer sets. In: Dix, J., Pereira, L.M., Przymusinski, T.C. (eds.) Non-Monotonic Extensions of Logic Programming, pp. 57–70. Springer, Berlin (1997)
55. Peschiera, C., Pulina, L., Tacchella, A., Bubeck, U., Kullmann, O., Lynce, I.: The seventh QBF solvers evaluation (QBFEVAL'10). In: Strichman, O., Szeider, S. (eds.) Proceedings of the 13th International Conference Theory and Applications of Satisfiability Testing (SAT'10). Lecture Notes in Computer Science, vol. 6175, pp. 237–250. Springer, Edinburgh (2010)
56. Pichler, R., Rümmele, S., Szeider, S., Woltran, S.: Tractable answer-set programming with weight constraints: bounded treewidth is not enough. Theory Pract. Logic Program. **14**(2), 141–164 (2014)
57. Schaefer, T.J.: The complexity of satisfiability problems. In: Lipton, R.J., Burkhard, W.A., Savitch, W.J., Friedman, E.P., Aho, A. (eds.) Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC'78), pp. 216–226. New York, Assoc. Comput. Mach. (1978)
58. Simons, P., Niemelä, I., Soininen, T.: Extending and implementing the stable model semantics. Artif. Intell. **138**(1-2), 181–234 (2002)
59. Syrjänen, T.: Lparse 1.0 user's manual. http://www.tcs.hut.fi/Software/smodels/lparse.ps/ (2002)
60. Truszczyński, M.: Trichotomy and dichotomy results on the complexity of reasoning with disjunctive logic programs. Theory Pract. Logic Program. **11**, 881–904 (2011)
61. Tu, P.H., Son, T.C., Gelfond, M., Morales, A.R.: Approximation of action theories and its application to conformant planning. Artif. Intell. **175**(1), 79–119 (2011). John McCarthy's Legacy

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.