TLP **11** (6): 881–904, 2011. © Cambridge University Press 2010
doi:10.1017/S1471068410000463 First published online 8 October 2010

Trichotomy and dichotomy results on the complexity of reasoning with disjunctive logic programs

MIROSŁAW TRUSZCZYŃSKI
Department of Computer Science, University of Kentucky, Lexington, KY 40506, USA
(*e-mail:* mirek@cs.uky.edu)

submitted 18 November 2009; revised 11 June 2010; accepted 25 June 2010

Abstract

We present *trichotomy* results characterizing the complexity of reasoning with disjunctive logic programs. To this end, we introduce a certain definition schema for classes of programs based on a set of allowed arities of rules. We show that each such class of programs has a finite representation, and for each of the classes definable in the schema, we characterize the complexity of the existence of an answer set problem. Next, we derive similar characterizations of the complexity of skeptical and credulous reasoning with disjunctive logic programs. Such results are of potential interest. On the one hand, they reveal some reasons responsible for the hardness of computing answer sets. On the other hand, they identify classes of problem instances, for which the problem is "easy" (in P) or "easier than in general" (in NP). We obtain similar results for the complexity of reasoning with disjunctive programs under the supported-model semantics.

KEYWORDS: answer sets, supported models, complexity of reasoning

1 Introduction

It is well known that the problem to decide whether a *propositional* disjunctive logic program has an answer set (the *Existence of an Answer Set* problem, or the *EAS* problem, for short) is Σ_2^P-complete (Eiter and Gottlob 1995). It is also well known that putting restrictions on the input instances may affect the complexity. For example, the EAS problem for normal logic programs is NP-complete (Marek and Truszczyński 1991).

In this paper, we study the complexity of the EAS problem for classes of propositional disjunctive logic programs that can be defined by sets of program rule *arities*. We show that for each such class the problem is either in P, is NP-complete, or is Σ_2^P-complete, and we fully characterize the classes of programs that fall into each category. We extend this result to establish similar characterizations for the problems of skeptical and credulous reasoning with disjunctive logic programs. Such results are of potential interest. On the one hand, they reveal some reasons

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

responsible for the hardness of computing answer sets; cf. Lemmas 4 and 5. On the other hand, they identify classes of problem instances, for which the problem is "easy" (in P) or "still easier than in general" (in NP); cf. Lemmas 1 and 2.

We also consider the corresponding reasoning problems for the semantics of supported models (Marek and Subrahmanian 1992; Brass and Dix 1997; Inoue and Sakama 1998) and obtain similar results. However, the classification is simpler, as the reasoning problems under the semantics of supported models are confined to the first level of the polynomial hierarchy. Thus, for each class of programs characterized in terms of arities, the decision problems related to reasoning tasks for programs from that class with respect to supported models turn out to be either in P or NP-complete (coNP-complete, depending on the task).

Our results can be regarded as *trichotomy* (respectively, *dichotomy*) results for the complexity of reasoning tasks in disjunctive logic programming. Similar results are known for the complexity of reasoning in other formalisms: propositional satisfiability (Schaefer 1978; Creignou *et al.* 2001; Bulatov *et al.* 2005), reasoning with minimal models (Cadoli 1992), default logic (Chapdelaine *et al.* 2007), and abductive reasoning (Nordh and Zanuttini 2008). There is, however, an important distinction between those earlier papers and our approach. The results contained there are concerned with the setting in which formulas are conjunctions of Boolean relations, and the set of models of a formula is the intersection of the sets of models of its constituent relations (in particular, it implies the monotonicity of inference from such formulas). The basic results concern the complexity of the satisfiability problem for classes of formulas determined by sets of Boolean relations allowed as formula conjuncts. It turns out that there is a simple characterization of all these classes, for which the problem is in P; moreover, for all other classes, the problem is NP-complete (Schaefer 1978; Creignou *et al.* 2001; Bulatov *et al.* 2005). This result can be exploited to characterize the complexity of reasoning with systems, in which basic reasoning tasks reduce to series of satisfiability tests (Cadoli 1992; Chapdelaine *et al.* 2007; Nordh and Zanuttini 2008). In the setting of disjunctive logic programs, these earlier results seem to be of little help. It is well known that the answer-set and the supported-model semantics are *nonmonotone* and so, logic programs under the answer-set semantics are *not* conjunctions of their rules. Thus, it is unclear whether defining classes of programs in terms of semantic properties of individual rules could yield any useful insights.

Finally, we stress that we are concerned with propositional programs only. For results on the complexity of reasoning with disjunctive logic programs in the general (nonground) case, we refer the reader to the paper by Dantsin *et al.* (2001).

2 Preliminaries

We fix an infinite countable set At of propositional variables. A *disjunctive program* (or simply, a program) over the set of atoms At is a collection of *disjunctive logic program rules*, that is, expressions of the form

$$r = a_1 | \ldots | a_k \leftarrow b_1, \ldots, b_m, not\ c_1, \ldots, not\ c_n, \tag{1}$$

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

where a_i, b_i, and c_i are atoms from At. The disjunction $a_1|\ldots|a_k$ is the *head* of r and the conjunction $b_1,\ldots,b_m,not\ c_1,\ldots,not\ c_n$ is the *body* of r. We call the triple $\alpha = [k,m,n]$ the *arity* of r. If $k \geqslant 1$, we call r *proper*. Otherwise, $k = 0$ and r is a *constraint*. We allow the possibility that the head and body of a rule are both empty. Such rule (it is unique) is contradictory.

We recall that given a program P and a set M of atoms (an interpretation), the *reduct* of P with respect to M, P^M, is the program obtained by removing for all $c \in M$ all rules with a literal *not* c in the body and, then, removing all negative literals (negated atoms) from the bodies of all remaining rules. A set of atoms M is an *answer set* of a disjunctive program P if M is a minimal model of P^M (Gelfond and Lifschitz 1991).

For a program P, we denote by $\overline{P}$ and $\overline{\overline{P}}$ the programs consisting of all proper rules and of all constraints in P, respectively. The following result is well known.

Theorem 1
A set $M \subseteq At$ is an answer set of a program P if-and-only-if M is an answer set of $\overline{P}$ and a model of $\overline{\overline{P}}$.

One can define classes of logic programs by specifying arities of rules. For instance, the set $\{[1,1,0],[0,1,0]\}$ defines the set of all Horn programs with constraints such that each rule has at most one atom in the body. Some classes of programs do not have such a finitary representation in terms of arities. For instance, the class of all Horn programs with constraints can be defined by the set $\{[k,m,0] \mid k \leqslant 1, 0 \leqslant m\}$, but there is no *finite* set of arities that could be used instead. To handle such cases, we introduce now a general representation schema for defining classes of programs in terms of sets of arities.

Let $U = \{0,1,\ldots\} \cup \{\infty\}$. We consider U to be ordered by the relation $\leqslant$ (the standard $\leqslant$ ordering relation on nonnegative integers, extended by $i \leqslant \infty$, for every $i = 0,1,\ldots$). Next, we define $\mathscr{A} = \{[k,m,n] \mid k,m,n \in U\}$. Thus, $\mathscr{A}$ contains all arities, as well as *additional* triples — those containing at least one occurrence of ∞. We refer to triples of that latter sort as *superarities*. We emphasize that superarities are *not* arities as we do not consider infinitary rules. If $\alpha \in \mathscr{A}$, we write α_1, α_2, and α_3 for the components of α.

Let $\alpha,\beta \in \mathscr{A}$. We define $\alpha \preceq \beta$ if

(1) $\alpha_i \leqslant \beta_i$, for $i = 1,2,3$, and
(2) if $\alpha_1 = 0$, then $\beta_1 = 0$.

We write $\alpha \prec \beta$ when $\alpha \preceq \beta$ and $\alpha \neq \beta$.

If $\Delta \subseteq \mathscr{A}$, then we define $\mathscr{F}(\Delta)$ to be the set of all *finite* programs P that satisfy the following condition: for every rule $r \in P$ there is $\alpha \in \Delta$ such that $\alpha_r \preceq \alpha$, where α_r denotes the arity of r. The condition (2) in the definition of $\preceq$ allows us to distinguish between classes of proper programs and classes of programs with constraints. Indeed, without the condition (2), every class of programs of the form $\mathscr{F}(\Delta)$ would contain constraints. With the condition (2), we can specify classes of constraint-free programs by means of sets Δ such that for every $\alpha \in \Delta$, $\alpha_1 \geqslant 1$. Including in Δ elements α with $\alpha_1 = 0$ yields classes of programs with constraints.

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

As there are classes of proper programs that are of interest (Horn programs and normal logic programs are typically defined as consisting of proper rules only), the distinction is needed and motivates the condition (2) in the definition of $\preceq$.

Using this schema, we can define several important classes of programs. For instance, the class of proper Horn programs can be described as $\mathscr{F}(\{[1,\infty,0]\})$ and the class of normal logic programs with constraints as $\mathscr{F}(\{[1,\infty,\infty],[0,\infty,\infty]\})$.

Our main goal in this paper is to determine the complexity of the EAS problem when input programs come from classes $\mathscr{F}(\Delta)$, for $\Delta \subseteq \mathscr{A}$. We also consider the corresponding problem for the case of supported models.

3 The case of finite Δ

In this section, we tackle the case when Δ is finite. We note that given a finite set $\Delta \subseteq \mathscr{A}$, the problem to decide the membership of a program in the class $\mathscr{F}(\Delta)$ is in P.

We start by establishing upper bounds on the complexity of the EAS problem for classes $\mathscr{F}(\Delta)$ given by some particular finite sets Δ of arities. Our first result is concerned with the following classes: $\mathscr{F}(\{[\infty,\infty,0],[0,0,0]\})$ — the class of proper positive disjunctive programs; $\mathscr{F}(\{[1,\infty,0],[0,\infty,\infty]\})$ — the class of programs whose every rule is either a proper Horn rule or a constraint; $\mathscr{F}(\{[\infty,1,0],[0,1,0]\})$— the class of *dual Horn programs*, that is, programs whose every rule, when viewed as a propositional clause, is a dual Horn clause; and $\mathscr{F}(\{[i,j,0] \in \mathscr{A} \mid i+j \leqslant 2\})$— the class of positive programs whose every rule consists of at most two literals. For each of these classes of programs, the EAS problem is easy (that is, in P).

Lemma 1
If Δ is one of:

(1) $\{[\infty,\infty,0],[0,0,0]\}$,
(2) $\{[1,\infty,0],[0,\infty,\infty]\}$,
(3) $\{[\infty,1,0],[0,1,0]\}$,
(4) $\{[i,j,0] \in \mathscr{A} \mid i+j \leqslant 2\}$,

then the EAS problem for $\mathscr{F}(\Delta)$ is in P.

Proof: If $P \in \mathscr{F}(\{[\infty,\infty,0],[0,0,0]\})$, then P either contains a contradictory rule and so does not have answer sets, or it is a proper positive program. In the latter case, P has models and so, minimal models, too. It follows that the EAS problem is in P, in this case.

Next, let $P \in \mathscr{F}(\{[1,\infty,0],[0,\infty,\infty]\})$. In this case, $\overline{P}$ is a proper Horn program. It has a least model, say M, which is the only answer set of $\overline{P}$. It is well known that M can be computed in polynomial time (Dowling and Gallier 1984). Moreover, it can be verified in polynomial time whether M satisfies the constraint part $\overline{\overline{P}}$ of P. Thus, the assertion follows in this case, as well.

Thus, let us assume that $P \in \mathscr{F}(\{[\infty,1,0],[0,1,0]\})$ or $P \in \mathscr{F}(\{[i,j,0] \in \mathscr{A} \mid i+j \leqslant 2\})$. Then P is a dual Horn program, or P is positive and every clause in P consists of two literals. In each case, one can decide in polynomial time whether P has a

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

model.[1] If the answer is "no," then P has no answer sets. Otherwise, P has a model, say M. Since M is a model of $\overline{P}$, there is a subset M' of M such that M' is a minimal model of $\overline{P}$. We have $\overline{P}^{M'} = \overline{P}$. Thus, M' is an answer set of $\overline{P}$. Since M satisfies $\overline{\overline{P}}$ and each rule in $\overline{\overline{P}}$ is of the form $\leftarrow a$ or $\leftarrow a, b$, M' satisfies $\overline{\overline{P}}$, too. Thus, M' is an answer set of P. Again, the assertion follows. □

The second result establishes sufficient conditions for the EAS problem to be in the class NP. It turns out to be the case for the following three classes of programs: $\mathscr{F}(\{[1, \infty, \infty], [0, \infty, \infty]\})$ — the class of normal logic programs with constraints; $\mathscr{F}(\{[\infty, 1, \infty], [0, \infty, \infty]\})$ — the class of programs whose reducts consist of proper dual Horn rules and constraints; and $\mathscr{F}(\{[\infty, \infty, 0], [0, \infty, 0]\})$ — the class of positive programs.

Lemma 2
If Δ is one of:

(1) $\{[1, \infty, \infty], [0, \infty, \infty]\}$,
(2) $\{[\infty, 1, \infty], [0, \infty, \infty]\}$,
(3) $\{[\infty, \infty, 0], [0, \infty, 0]\}$,

then the EAS problem for $\mathscr{F}(\Delta)$ is in NP.

Proof: If $\Delta = \{[1, \infty, \infty], [0, \infty, \infty]\}$, $\mathscr{F}(\Delta)$ consists of normal logic programs with constraints. In this case, the result is well known (Marek and Truszczyński 1991).

Next, let $\Delta = \{[\infty, 1, \infty], [0, \infty, \infty]\}$ and $P \in \mathscr{F}(\Delta)$. To prove the assertion, it is enough to show that there is a polynomial time algorithm for deciding whether a set of atoms $M \subseteq At(P)$ is an answer set of P. To this end, we note that $M \subseteq At(P)$ is a minimal model of $\overline{P}^M$ if-and-only-if for every $a \in M$, the program $\overline{P}^M \cup \{\leftarrow a\} \cup \{\leftarrow b \mid b \in At(P) \setminus M\}$ does not have a model. Since $\overline{P}^M \cup \{\leftarrow a\} \cup \{\leftarrow b \mid b \in At(P) \setminus M\}$ is dual Horn, verifying whether M is a minimal model of $\overline{P}^M$ can be accomplished in polynomial time. In addition, checking that M is a model of $\overline{\overline{P}}$ can be done in polynomial time, too. Thus, in this case, the assertion follows.

Finally, if $\Delta = \{[\infty, \infty, 0], [0, \infty, 0]\}$ and $P \in \mathscr{F}(\Delta)$, then deciding whether P has an answer set is equivalent to deciding whether P has a model. Indeed, if M is an answer set of P, then M is a model of P. Conversely, if M is a model of P, then let $M' \subseteq M$ be a minimal model of $\overline{P}$ (such a model exists, as M is a model of $\overline{P}$). Clearly, $\overline{P}^{M'} = \overline{P}$ and so, M' is an answer set of $\overline{P}$. Since $M' \subseteq M$ and M is a model of $\overline{\overline{P}}$, M' is a model of $\overline{\overline{P}}$ (it follows from the fact that every rule in $\overline{\overline{P}}$ is of the form $\leftarrow a_1, \ldots, a_m$). Thus, M' is an answer set of P. Since the problem to decide whether P has a model is in NP, the assertion follows. □

While the upperbounds provided by Lemmas 2(1) and (3) are not surprising (in fact, as we noted, Lemma 2(1) is well known), Lemma 2(2) warrants additional comments. The class of programs considered there has not been identified before

[1] For instance, for a dual Horn program P, the case that is perhaps less broadly known, one can compute in the bottom-up fashion the complement of its greatest model contained in $At(P)$, or determine that no models exist (by means of a "dual" Dowling-Gallier algorithm).

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

as one of the classes of disjunctive programs for which the complexity of reasoning drops down to the first level of the polynomial hierarchy. In the same time, it is an interesting class of programs. In particular, programs in this class are not, in general *head-cycle free* (Ben-Eliyahu and Dechter 1997).

Next, we will prove several lower bound results. We will first exhibit classes of programs of the form $\mathscr{F}(\Delta)$ for which the EAS problem is NP-hard. To this end, we need a lemma, establishing the NP-hardness of the *Satisfiability* (*SAT*) problem for some simple classes of theories in the *Conjunctive Normal Form* (*CNF* theories, for short). For the most part, the result is folklore. We sketch an argument for the sake of completeness.

Lemma 3
The SAT problem restricted to each of the following classes of CNF theories is NP-hard:

(1) the class of all CNF formulas ψ such that each clause of ψ is a disjunction of two negated atoms, or of at most three atoms;
(2) the class of all CNF formulas ψ such that each clause of ψ consists of at most two negated atoms, or is a disjunction of two atoms, or is a disjunction of two atoms and one negated atom;
(3) the class of all CNF formulas ψ such that each clause of ψ consists of at most two atoms, or of one negated atom, or is a disjunction of an atom and two negated atoms;
(4) the class of all CNF formulas ψ such that each clause of ψ is a disjunction of two atoms, or of at most three negated atoms.

Proof: We will only prove the case (3). The argument in all other cases is similar.

Let φ be a CNF formula whose every clause has three literals, and let X be a set of atoms occurring in φ. For each atom $z \in X$, we introduce a fresh atom z'. Next, in each clause c, we replace some of its positive literals a with $\neg a'$ and some of its negative literals $\neg b$ with b' so that the resulting clause, we will denote it by $\hat{c}$, is the disjunction of exactly one atom and two negated atoms. Such replacements can always be found.

Finally, we introduce one more fresh atom, say f, and define $F(\varphi)$ as follows:

$$F(\varphi) = \{z \vee z' \mid z \in X\} \cup \{f \vee \neg z \vee \neg z' \mid z \in X\} \cup \{\neg f\} \cup \{\hat{c} \mid c \text{ is a clause in } \varphi\}.$$

It is evident that $F(\varphi)$ is in the class of theories under consideration.

We will show that φ has a model if-and-only-if $F(\varphi)$ has a model. To this end, we note that models of $F(\varphi)$ (if exist) are of the form $M \cup \{z' \mid z \in X \setminus M\}$, where $M \subseteq X$. It is now easy to see that M is a model of φ if-and-only-if $M \cup \{z' \mid z \in X \setminus M\}$ is a model of $F(\varphi)$. Thus, the claim and, consequently, the assertion, follow.

As we noted, the argument for the remaining classes is similar. We only need to change the definition of $\hat{c}$ and use clauses $\neg z \vee \neg z'$ instead of $f \vee \neg z \vee \neg z'$ [there is no need to introduce f, as clauses being the disjunctions of two negated atoms are allowed in formulas in each of the classes considered in (1), (2), and (4)]. □

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

We will now use Lemma 3 to establish the NP-hardness of the EAS problem for $\mathscr{F}(\Delta)$ for several simple sets $\Delta \subseteq \mathscr{A}$.

Lemma 4
If Δ is any of:

(1) $\{[1,0,1]\}$,
(2) $\{[2,0,0],[0,0,1]\}$,
(3) $\{[3,0,0],[0,2,0]\}$,
(4) $\{[2,1,0],[0,2,0]\}$,
(5) $\{[2,0,0],[1,2,0],[0,1,0]\}$,
(6) $\{[2,0,0],[0,3,0]\}$,

then the EAS problem for $\mathscr{F}(\Delta)$ is NP-hard.

Proof: (1) The proof of the NP-completeness of the EAS problem for normal logic programs given by Marek and Truszczyński (1991) establishes the assertion (1).

(2) We will construct a reduction from the SAT problem, concerning the class considered in Lemma 3(4). Let φ be a CNF of the appropriate form. We denote by $pos(\varphi)$ the set of all clauses in φ that are of the form $a \vee b$, where $a, b \in At$. We denote by $neg(\varphi)$ the set of all remaining clauses in φ (all of them are disjunctions of at most three negative literals).

For every clause $c = \neg y_1 \vee \ldots \vee \neg y_k$ in $neg(\varphi)$ (where, as we pointed out, $k \leqslant 3$), we introduce a fresh atom x_c. Next, we define

$$\begin{aligned} P(\varphi) &= \{a|b \leftarrow \mid a \vee b \in pos(\varphi)\} \cup \\ &\qquad \{x_c|y_i \leftarrow \mid c \in neg(\varphi),\ c = \neg y_1 \vee \ldots \vee \neg y_k,\ 1 \leqslant i \leqslant k\}, \\ Q(\varphi) &= \{\leftarrow not\ x_c \mid c \in neg(\varphi)\},\ \text{and} \\ R(\varphi) &= P(\varphi) \cup Q(\varphi). \end{aligned}$$

To simplify notation, we will write P, Q, and R for $P(\varphi)$, $Q(\varphi)$, and $R(\varphi)$, respectively.

We will now show that φ is satisfiable if-and-only-if R has an answer set. Since $R \in \mathscr{F}(\Delta)$, by Lemma 3(4) the assertion will follow.

($\Leftarrow$) Let M be an answer set of R. It follows that M is an answer set of P and so, a minimal model of P. Let c be a clause in φ. There are two cases. First, $c = a \vee b$, where $a, b \in At$. In that case, $a|b \leftarrow$ is a rule in P. Since M is a model of P, M is a model of $a|b \leftarrow$ and, consequently, of c. Second, $c = \neg y_1 \vee \ldots \vee \neg y_k$. We observe that $x_c \in M$ (M being an answer set of R is a model of Q). Since M is a minimal model of P, there is j, $1 \leqslant j \leqslant k$, such that $y_j \notin M$ (otherwise $M \setminus \{x_c\}$ would be a model of P, contradicting the minimality of M). Thus, M is a model of c also in this case. Consequently, φ is satisfiable.

($\Rightarrow$) Let us assume that φ has a model, say M. Let $M' \subseteq M$ be a minimal model of φ. It follows that M' is also a minimal model of $pos(\varphi)$, as every subset of M' is a model of $neg(\varphi)$. Let us define $M'' = M' \cup \{x_c \mid c \in neg(\varphi)\}$. Let r be a clause in P. If $r = a|b \leftarrow$, then $a \vee b \in pos(\varphi)$ and so, M'' is a model of r. Otherwise, $r = x_c|y_i \leftarrow$, where $c = \neg y_1 \vee \ldots \vee \neg y_k$ is from $neg(\varphi)$ and $1 \leqslant i \leqslant k$. Since $x_c \in M''$, M'' is a model of r. Thus, M'' is a model of P.

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

Let $N \subseteq M''$ be a model of P. Then $N' = N \setminus \{x_c \mid c \in neg(\varphi)\}$ is a model of every rule of the form $a|b \leftarrow$ in P and, consequently, N' is a model of $pos(\varphi)$. Since $N' \subseteq M'$ and M' is a minimal model of $pos(\varphi)$, $N' = M'$. Let $c \in neg(\varphi)$, say $c = \neg y_1 \vee \ldots \vee \neg y_k$. Since M' is a model of c, there is i, $1 \leqslant i \leqslant k$, such that $y_i \notin M'$. Consequently, $y_i \notin N$. As N is a model of P, $x_c \in N$. Thus, $\{x_c \mid c \in neg(\varphi)\} \subseteq N$. It follows that $N = M''$, that is, M'' is a minimal model of P. Since $P = \overline{R}$ and $P^{M''} = P$, M'' is an answer set of $\overline{R}$. Moreover, M'' is a model of Q (by the definition of M'') and $Q = \overline{\overline{R}}$. Thus, M'' is an answer set of R.

(3)–(6) In all the remaining cases, we exploit the fact that $P \in \mathscr{F}(\Delta)$ has an answer set if-and-only-if P has a model (the same argument that we used in the proof of Lemma 2 applies). The latter problem for each of the cases (3)–(6) can be shown to be equivalent to the satisfiability problem for the classes considered in Lemma 3(1)–(4), respectively. In each of these cases the problem is NP-hard (Lemma 3), and so the assertion follows. □

The next lemma establishes conditions guaranteeing Σ_2^P-hardness of the EAS problem. Eiter and Gottlob (1995) proved that given $P \in \mathscr{F}(\{[2,0,0],[1,3,0],[1,0,1]\})$, it is Σ_2^P-hard to decide whether P has an answer set. The proof can be modified to the case when the class of input programs is restricted to $\mathscr{F}(\{[2,0,0],[1,2,0],[1,0,1]\})$, as clauses of the arity $[1,3,0]$ can be simulated by clauses of arity $[1,2,0]$. Moreover, in the construction provided by Eiter and Gottlob, the only rule of the arity $[1,0,1]$ used is of the form $x \leftarrow not\ x$, and it can be simulated by the rule (constraint) $\leftarrow not\ x$, which has the arity $[0,0,1]$. Thus, the Σ_2^P-hardness holds also for the class $\mathscr{F}(\{[2,0,0],[1,2,0],[0,0,1]\})$ of programs. We omit the details and state the result only.

Lemma 5
If Δ is any of:

(1) $\{[2,0,0],[1,2,0],[0,0,1]\}$,
(2) $\{[2,0,0],[1,2,0],[1,0,1]\}$,

then the EAS problem for $\mathscr{F}(\Delta)$ is Σ_2^P-hard.

We will now derive the main result of this section. It provides a complete characterization of the complexity of the EAS problem for the class $\mathscr{F}(\Delta)$. To state the result, we introduce one more piece of notation. Given $\Delta, \Theta \subseteq \mathscr{A}$, we write $\Delta \preceq \Theta$ if for every $\alpha \in \Delta$ there is $\beta \in \Theta$ such that $\alpha \preceq \beta$. For instance, we have $\{[1,\infty,1],\{[\infty,0,0]\} \preceq \{[\infty,\infty,1]\}$. Clearly, if $\Delta \preceq \Theta$ then $\mathscr{F}(\Delta) \subseteq \mathscr{F}(\Theta)$. We will use this property frequently in proofs throughout the paper.

Theorem 2
Let $\Delta \subseteq \mathscr{A}$ be finite.

(A) If

(1) $\Delta \preceq \{[\infty,\infty,0],[0,0,0]\}$, or
(2) $\Delta \preceq \{[1,\infty,0],[0,\infty,\infty]\}$, or

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

(3) $\Delta \preceq \{[\infty, 1, 0], [0, 1, 0]\}$, or
(4) $\Delta \preceq \{[i, j, 0] \in \mathscr{A} \mid i + j \leqslant 2\}$,

then the EAS problem for $\mathscr{F}(\Delta)$ is in P.

(B) Otherwise, if

(1) $\Delta \preceq \{[1, \infty, \infty], [0, \infty, \infty]\}$, or
(2) $\Delta \preceq \{[\infty, 1, \infty], [0, \infty, \infty]\}$, or
(3) $\Delta \preceq \{[\infty, \infty, 0], [0, \infty, 0]\}$,

then the EAS problem for $\mathscr{F}(\Delta)$ is NP-complete.

(C) Otherwise, the EAS problem for $\mathscr{F}(\Delta)$ is Σ_2^P-complete.

Proof: The claim (A) follows directly from Lemma 1. Thus, let us assume that Δ does not fall under the scope of (A) and satisfies the assumptions of (B). By Lemma 2, the latter implies that the EAS problem for $\mathscr{F}(\Delta)$ is in NP.

If $\{[1, 0, 1]\} \preceq \Delta$ or $\{[2, 0, 0], [0, 0, 1]\} \preceq \Delta$, the NP-hardness of the EAS problem for $\mathscr{F}(\Delta)$ follows from Lemma 4, parts (1) and (2), respectively. Thus, let us assume that $\{[1, 0, 1]\} \not\preceq \Delta$ and $\{[2, 0, 0], [0, 0, 1]\} \not\preceq \Delta$.

Since $\{[1, 0, 1]\} \not\preceq \Delta$, we have $\Delta \preceq \{[\infty, \infty, 0], [0, \infty, \infty]\}$. Since Δ does not satisfy the condition (A2), $\{[2, 0, 0]\} \preceq \Delta$. Since $\{[2, 0, 0], [0, 0, 1]\} \not\preceq \Delta$, $\{[0, 0, 1]\} \not\preceq \Delta$. Thus, $\Delta \preceq \{[\infty, \infty, 0], [0, \infty, 0]\}$.

Since Δ does not satisfy the condition (A1), $\{[0, 1, 0]\} \preceq \Delta$. Similarly, since Δ does not satisfy the condition (A3), $\{[1, 2, 0]\} \preceq \Delta$ or $\{[0, 2, 0]\} \preceq \Delta$. We also have that Δ does not satisfy the condition (A4). Thus, there is $\alpha \in \Delta$ such that $\alpha_1 + \alpha_2 \geqslant 3$. Since we already proved that $\{[2, 0, 0]\} \preceq \Delta$, it follows that at least one of the following conditions holds: $\{[3, 0, 0], [0, 2, 0]\} \preceq \Delta$, $\{[2, 1, 0], [0, 2, 0]\} \preceq \Delta$, $\{[2, 0, 0], [1, 2, 0], [0, 1, 0]\} \preceq \Delta$, or $\{[2, 0, 0], [0, 3, 0]\} \preceq \Delta$. Thus, the NP-hardness of the EAS problem for $\mathscr{F}(\Delta)$ follows again from Lemma 4 and completes the proof of (B).

To prove (C), let us assume that Δ does not fall under the scope of (B). Since Δ does not satisfy (B1), $\{[2, 0, 0]\} \preceq \Delta$. Similarly, since Δ does not satisfy (B2), $\{[1, 2, 0]\} \preceq \Delta$. Finally, since Δ does not satisfy (B3), $\{[0, 0, 1]\} \preceq \Delta$ or $\{[1, 0, 1]\} \preceq \Delta$. Thus, the Σ_2^P-hardness follows by Lemma 5. Since the EAS problem is in Σ_2^P even without any restrictions on the class of programs, both (C) and the assertion of the lemma follows. □

4 The case of infinite Δ

The question we study now is whether there are interesting classes of programs of the form $\mathscr{F}(\Delta)$, when Δ is infinite. The main result of this section is that by allowing Δ to be infinite, we do not obtain *any* new classes of programs. In other words, for every class of programs of the form $\mathscr{F}(\Delta)$, there is a finite set $\Delta' \subseteq \mathscr{A}$ such that $\mathscr{F}(\Delta) = \mathscr{F}(\Delta')$.

A sequence $\{\alpha^k\}_{k=1}^{\infty}$ is *monotone* (*strictly* monotone) if for every k, $\alpha^k \preceq \alpha^{k+1}$ ($\alpha^k \prec \alpha^{k+1}$, respectively). Let $\{\alpha^k\}_{k=1}^{\infty}$ be a *monotone* sequence of elements of $\mathscr{A}$.

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

We define the *limit* of this sequence as $\alpha^\infty = [(\alpha^\infty)_1, (\alpha^\infty)_2, (\alpha^\infty)_3]$, where $(\alpha^\infty)_i = \sup\{(\alpha^k)_i \mid k = 1, 2 \ldots, \}$, for $i = 1, 2, 3$. We stress that we do not consider transfinite sequences here. All sequences have the set of natural numbers as their domain. Let $\Delta \subseteq \mathscr{A}$. A monotone sequence $\{\alpha^k\}_{k=1}^\infty$ of elements of Δ is *maximal* if there is *no* $\alpha \in \Delta$ such that $\alpha^\infty \prec \alpha$. We define $A(\Delta)$ to be the set of the limits of maximal sequences in Δ.

We have the following two lemmas (we omit the proof of the first one as it is evident).

Lemma 6
Let $\{\beta^k\}_{k=1}^\infty$ be a strictly monotone sequence of elements from $\mathscr{A}$. For every $\alpha \in \mathscr{A}$, if $\alpha \prec \beta^\infty$, then there is k such that $\alpha \prec \beta^k$.

Lemma 7
Let $\Delta \subseteq \mathscr{A}$ and $\alpha \in \Delta$. Then, there is $\alpha' \in A(\Delta)$ such that $\alpha \preceq \alpha'$.

Proof: Let $X = \{\beta \in \Delta \mid \alpha \preceq \beta\}$. If X has a maximal element, say γ, a sequence with each term equal to γ is maximal. Its limit, also equal to γ, clearly satisfies $\gamma \in A(\Delta)$ and $\alpha \preceq \gamma$. Thus, the assertion follows.

Otherwise, X has no maximal elements. Let α^1 be any element in X (we note that $X \neq \emptyset$, as $\alpha \in X$). Let $k \geqslant 1$ and let $\langle \alpha^1, \ldots, \alpha^k \rangle$ be a strictly monotone sequence of k elements in X, for some $k \geqslant 1$. Since X has no maximal elements, X contains elements that are strictly greater than α^k. Let us select as α^{k+1} an element $\beta \in X$ such that $\alpha^k \prec \beta$ and $\alpha_i^k < \beta_i$ holds on as many positions $i = 1, 2, 3$ as possible. An infinite sequence we define in this way, we will denote it by α, is strictly monotone. Let us assume that there is $\beta \in \Delta$ such that $\alpha^\infty \prec \beta$. It follows that there is j, $1 \leqslant j \leqslant 3$, such that $(\alpha^\infty)_j < \beta_j$. Thus, $(\alpha^\infty)_j = m$, for some integer m, and there is n such that $(\alpha^n)_j = m$. Since $\alpha^{n+1} \prec \beta$ and $(\alpha^n)_j = (\alpha^{n+1})_j = m$, the number of positions i such that $(\alpha^n)_i < (\alpha^{n+1})_i$ is strictly smaller than the number of positions i such that $(\alpha^n)_i < \beta_i$. Since $\beta \in X$, that contradicts the way we constructed the sequence α.

It follows that the sequence $\{\alpha^k\}_{k=1}^\infty$ is maximal for Δ and so, the assertion follows in this case, too. □

We now have the following properties.

Proposition 1
For every $\Delta \subseteq \mathscr{A}$, $\mathscr{F}(\Delta) = \mathscr{F}(A(\Delta))$.

Proof: To prove the assertion, it is enough to show that for every *arity* α (no occurrence of ∞), $\{\alpha\} \preceq \Delta$ if-and-only-if $\{\alpha\} \preceq A(\Delta)$. Let us first assume that $\{\alpha\} \preceq \Delta$. It follows that there is an element $\alpha' \in \Delta$ such that $\alpha \preceq \alpha'$. By Lemma 7, there is $\alpha'' \in A(\Delta)$ such that $\alpha' \preceq \alpha''$. Thus, $\alpha \preceq \alpha''$ and so, $\{\alpha\} \preceq A(\Delta)$.

Conversely, let $\{\alpha\} \preceq A(\Delta)$. It follows that there is $\beta \in A(\Delta)$ such that $\alpha \preceq \beta$. Since $\beta \in A(\Delta)$, there is a monotone sequence $\{\beta^k\}_{k=1}^\infty$ of elements of Δ such that its limit is β. Without the loss of generality, we can assume that either starting with some k_0, the sequence $\{\beta^k\}_{k=1}^\infty$ is constant, or the sequence β is strictly monotone. In the first case, $\alpha \preceq \beta = \beta^{k_0}$. Since $\beta^{k_0} \in \Delta$, $\{\alpha\} \preceq \Delta$. In the second case, Lemma 6 implies that there is k such that $\alpha \prec \beta^k$ and again $\{\alpha\} \preceq \Delta$ follows. □

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

Proposition 2
For every Δ, $A(\Delta)$ is an antichain.

Proof: Let us assume that $A(\Delta)$ is not an antichain. Then there are $\alpha, \alpha' \in A(\Delta)$ such that $\alpha \prec \alpha'$. We have that $\alpha' \notin \Delta$ (otherwise, the sequence that α is a limit of would not be maximal). Consequently, α' is the limit of a strictly monotone sequence of elements from Δ. By Lemma 6, there is an element β in the sequence such that $\alpha \prec \beta$. That contradicts the fact that α is the limit of a maximal sequence and yields the assertion. □

Proposition 3
Every antichain in the partially ordered set $\langle \mathscr{A}, \preceq \rangle$ is finite.

Proof: We will first prove that every sequence $s = \{[b^k, c^k]\}_{k=1}^{\infty}$ such that (i) $b^k, c^k \in U$, for $k = 1, 2, \ldots$, and (ii) the set of distinct elements occurring in the sequence is infinite, which contains an infinite strictly monotone subsequence.

We start by observing that by (ii) one can select an infinite subsequence of s, in which all elements are distinct. Thus, without the loss of generality, we may assume that, in fact, all elements in s are distinct. If there is b such that $C(b) = \{k \mid b^k = b\}$ is infinite, then the assertion is evident. Indeed, the sequence $\{c^k\}_{k \in C(b)}$ contains no repetitions (as s contains no repetitions) and, consequently, contains a strictly increasing subsequence. Thus, let us assume that for every b, the set $C(b)$ is finite. It follows that the sequence $\{b^k\}_{k=1}^{\infty}$ contains a strictly increasing subsequence, say $\{b^{k_j}\}_{j=1}^{\infty}$.

If there is c that occurs in the corresponding sequence $\{c^{k_j}\}_{j=1}^{\infty}$ infinitely many times, then these occurrences yield a strictly monotone subsequence of $\{[b^{i_j}, c^{i_j}]\}_{j=1}^{\infty}$ and, consequently, a strictly monotone subsequence of $\{[b^k, c^k]\}_{k=1}^{\infty}$. Otherwise, $\{c^{k_j}\}_{j=1}^{\infty}$ contains infinitely many elements. Thus, it contains a strictly monotone subsequence, which, together with the corresponding b's, yields a strictly monotone subsequence of $\{[b^k, c^k]\}_{k=1}^{\infty}$.

To prove the assertion of the proposition, let us assume that there is an *infinite* antichain A in $\langle \mathscr{A}, \preceq \rangle$. Let $\{\alpha^k\}_{k=1}^{\infty}$ be any enumeration of the elements of A (without repetitions). If there is $a \in U$ such that $A_a = \{\alpha \in A \mid \alpha_1 = a\}$ is infinite, then the fact proved above yields a contradiction.

Thus, for every $a \in U$, the set A_a is finite. Since A is infinite, there is an infinite subsequence β of α such that for every $n = 1, 2, \ldots$, $(\beta^n)_1 < (\beta^{n+1})_1$. As A is an antichain, the sequence $\{[(\beta^n)_2, (\beta^n)_3]\}_{n=1}^{\infty}$ contains no repeating elements. By the property proved above, it contains an infinite strictly monotone subsequence, a contradiction. □

These properties imply the main result of this section. It asserts that every class of programs $\mathscr{F}(\Delta)$ can be defined by means of a finite set Δ' that is an antichain in $\langle \mathscr{A}, \preceq \rangle$.

Theorem 3
For every set $\Delta \subseteq \mathscr{A}$, there is a finite subset $\Delta' \subseteq \mathscr{A}$ such that Δ' is an antichain and $\mathscr{F}(\Delta) = \mathscr{F}(\Delta')$.

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

Proof: Let us define $\Delta' = A(\Delta)$. By Propositions 2 and 3, Δ' is a finite antichain in $\langle \mathscr{A}, \preceq \rangle$, and by Proposition 1, $\mathscr{F}(\Delta) = \mathscr{F}(\Delta')$. Thus, the theorem follows. □

5 The complexity of skeptical and credulous reasoning

The EAS problem is just one example of a reasoning task that arises in the context of disjunctive logic programs with the answer-set semantics. There are several other tasks that are of interest, too. They concern deciding whether a program nonmonotonically entails a literal, that is an atom, say a, or its negation $\neg a$.

We recall that if M is a set of atoms (an interpretation) and a is an atom, then $M \models a$ if $a \in M$, and $M \models \neg a$ if $a \notin M$. For a disjunctive logic program P and a literal l we say that

(1) P *skeptically entails* l, written $P \models_s l$, if $M \models l$, for every answer set M of P;
(2) P *credulously entails* l, written $P \models_c l$, if there is an answer set M of P such that $M \models l$.

We note that $P \models_s l$ if-and-only-if $P \not\models_c \bar{l}$, where $\bar{l}$ is l's *dual* literal. Thus, to establish fully the complexity of deciding nonmonotonic entailment, it is enough to focus on deciding whether $P \models_c \neg a$ and $P \models_s \neg a$, where a is an atom. These two decision tasks were studied by Eiter and Gottlob (1995) , who proved that, in general, the first one is Σ_2^P-complete and the second one is Π_2^P-complete.

Reasoning with answer sets is related to circumscription and closed-world reasoning with propositional theories. A detailed study of the complexity of those forms of reasoning was conducted by Cadoli and Lenzerini (1994). Using Theorem 2 and one of the results from that paper (which we state in the proof below), one can characterize in terms of our definition schema the complexity of deciding, given a program P and an atom a, whether $P \models_c \neg a$ and $P \models_s \neg a$. The two problems are addressed in the following two theorems.

Theorem 4
Let $\Delta \subseteq \mathscr{A}$ be finite.

(A) If

(1) $\Delta \preceq \{[1, \infty, 0], [0, \infty, \infty]\}$, or
(2) $\Delta \preceq \{[\infty, 1, 0], [0, 1, 0]\}$, or
(3) $\Delta \preceq \{[i, j, 0] \in \mathscr{A} \mid i + j \leqslant 2\}$,

then the problem to decide whether $P \models_c \neg a$, where $P \in \mathscr{F}(\Delta)$ and a is an atom, is in P.

(B) Otherwise, if

(1) $\Delta \preceq \{[1, \infty, \infty], [0, \infty, \infty]\}$, or
(2) $\Delta \preceq \{[\infty, 1, \infty], [0, \infty, \infty]\}$, or
(3) $\Delta \preceq \{[\infty, \infty, 0], [0, \infty, 0]\}$,

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

then the problem to decide whether $P \models_c \neg a$, where $P \in \mathscr{F}(\Delta)$ and a is an atom, is NP-complete.

(C) Otherwise, the problem to decide whether $P \models_c \neg a$, where $P \in \mathscr{F}(\Delta)$ and a is an atom, is Σ_P^2-complete.

Proof: It is well known that P has an answer set M such that $M \models \neg a$ (that is, $a \notin M$) if-and-only-if $P \cup \{ \leftarrow a\}$ has an answer set. Let $\Delta \subseteq \mathscr{A}$ be finite and let us define $\Delta' = \Delta \cup \{[0,1,0]\}$. Clearly, if $P \in \mathscr{F}(\Delta)$, then $P \cup \{ \leftarrow a\} \in \mathscr{F}(\Delta')$. Moreover, if Δ falls under the scope of (A) [(A) or (B), respectively] of this theorem then Δ' falls under the scope of (A) [(A) or (B), respectively] of Theorem 2. Consequently, the upper bound follows by Theorem 2.

To prove the lower bounds, we first consider the case when Δ does not satisfy any of the conditions listed in (A) and $\Delta \preceq \{[\infty,\infty,0],[0,0,0]\}$ [thus, in particular, Δ falls under the scope of (B)]. Since Δ does not fall under the condition (A1), $\{[2,0,0]\} \preceq \Delta$. Similarly, since Δ does not fall under the condition (A2), $\{[1,2,0]\} \preceq \Delta$.

Let $P \in \mathscr{F}(\Delta')$, where $\Delta' = \{[2,0,0],[1,2,0],[0,1,0]\}$. We select a fresh atom a and define $P' = \overline{P} \cup \{a \leftarrow bd(r) \mid r \in \overline{\overline{P}}\}$. Clearly, $P' \in \mathscr{F}(\Delta)$. Moreover, P has an answer set if-and-only-if $P' \models_c \neg a$. Thus, the claim follows by Lemma 4(5).

From now on, we assume that $\Delta \not\preceq \{[\infty,\infty,0],[0,0,0]\}$ and Δ does not satisfy any of the conditions listed in (A). Let us also assume that Δ falls under the scope of (B) or (C) of this theorem. It follows that Δ falls under the scope of the corresponding case of Theorem 2.

If Δ satisfies (B1), then $\{[1,0,1]\} \preceq \Delta$ [otherwise, (A1) would hold]. Let $P \in \mathscr{F}(\Delta)$. Let a, b be fresh atoms and let us set $P' = P \cup \{a \leftarrow not\ b; b \leftarrow not\ a\}$. If Δ does not satisfy (B1), $\{[2,0,0]\} \preceq \Delta$. Let $P \in \mathscr{F}(\Delta)$, and let us set $P' = P \cup \{a|b\}$, where a, b are fresh atoms. Clearly, in either case, $P' \in \mathscr{F}(\Delta)$, and P has an answer set if-and-only-if $P' \models_c \neg a$. Thus, all the remaining lower bounds follow from Theorem 2. □

Theorem 5
Let $\Delta \subseteq \mathscr{A}$ be finite.

(A) If $\Delta \preceq \{[1,\infty,0],[0,\infty,\infty]\}$, then the problem to decide whether $P \models_s \neg a$, where $P \in \mathscr{F}(\Delta)$ and a is an atom, is in P.

(B) Otherwise, if

(1) $\Delta \preceq \{[1,\infty,\infty],[0,\infty,\infty]\}$, or
(2) $\Delta \preceq \{[\infty,1,\infty],[0,\infty,\infty]\}$,

then the problem to decide whether $P \models_s \neg a$, where $P \in \mathscr{F}(\Delta)$ and a is an atom, is coNP-complete.

(C) Otherwise, the problem to decide whether $P \models_s \neg a$, where $P \in \mathscr{F}(\Delta)$ and a is an atom, is Π_2^P-complete.

Proof: It is well known that P has an answer set such that $M \not\models \neg a$ (that is, $a \in M$) if-and-only-if $P \cup \{ \leftarrow not\ a\}$ has an answer set. That observation implies all upper bound results (by a similar argument as that used in the proof of the previous theorem).

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

We will now prove the lower bounds for the cases (B) and (C). Let us assume that Δ does not satisfy (A) but falls under the scope of (B). If Δ satisfies (B1), then $\{[1,0,1]\} \preceq \Delta$. Let $P \in \mathscr{F}(\Delta)$ and let a and a' be the fresh atoms. We note that P has an answer set if and only if $P \cup \{a \leftarrow not\ a'\}$ has an answer set M such that $a \in M$ or, equivalently, if and only if $P \cup \{a \leftarrow not\ a'\} \not\models_s \neg a$. Thus, the hardness follows [cf. Lemma 4(1)].

Let us assume then that Δ does not satisfy (B1). Then, we have that $\{[2,0,0]\} \preceq \Delta$. It follows from the results of Cadoli and Lenzerini (1994) that it is NP-complete to decide whether a given 2CNF theory whose every clause is a disjunction of two atoms has a minimal model that contains a given atom a. As minimal models of such theories are precisely answer sets of the corresponding disjunctive program, it follows that given a program $P \in \mathscr{F}(\Delta)$ and an atom a, it is coNP-hard to decide whether $P \models_s \neg a$.

Finally, let us assume that neither (A) nor (B) apply to Δ. Then $[1,2,0] \in \Delta$ and $[2,0,0] \in \Delta$. Eiter and Gottlob (1995) proved that if $P \in \mathscr{F}(\{[2,0,0],[1,3,0]\})$ and a is an atom, then it is Π_2^P-hard to decide whether $P \models_s \neg a$. That result can be strengthened to the case when $P \in \mathscr{F}(\{[2,0,0],[1,2,0]\})$, as clauses of the arity $[1,3,0]$ can be simulated by clauses of arity $[1,2,0]$ (cf. the comments preceding Lemma 5). □

We note that credulous reasoning is simple (in P or in NP) for several classes of programs. In contrast, there are fewer classes of programs, for which skeptical reasoning is simple (in P or coNP). The main reason behind this asymmetry is that in the cases (A2), (A3), and (B3) of Theorem 4 (positive programs) answer sets and minimal models coincide. Thus, in these cases, credulous reasoning asks for the existence of a *minimal* model that does not contain an atom a, which is equivalent to the existence of a model (not necessarily minimal) that does not contain a. In other words, the requirement of minimality becomes immaterial (one source of complexity disappears). This is not so with skeptical reasoning, where not having a in any minimal model is not the same as not having a in any model. A similar comparison of skeptical and credulous reasoning for positive programs was offered by Eiter and Gottlob (1995) for the coarser setting of classes of programs they considered.

6 Another representation schema — explicit arities

Next, we consider briefly an alternative way, in which classes of programs could be described by means of arities of rules. When defining the class $\mathscr{F}(\Delta)$, we view each element $\alpha \in \Delta$ as a shorthand for the set of all arities β such that $\beta \preceq \alpha$. In other words, Δ is an *implicit* representation of the set of all allowed arities: not only those arities that are explicitly listed in Δ are legal but also those that are "dominated" by them.

There is another, more direct (more explicit), way to use arities to define classes of programs. Let $\Delta \subseteq \mathscr{A}$ be a set of *arities*, that is, we now do not allow superarities in Δ. We define $\mathscr{G}(\Delta)$ to consist of all finite programs P such that for every rule $r \in P$, the arity α of r belongs to Δ. Thus, when defining the class $\mathscr{G}(\Delta)$, Δ serves as an *explicit* specification of the set of allowed arities.

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

One can show that the results of Section 3 can be adapted to the setting of classes of the form $\mathscr{G}(\Delta)$, where Δ is a set of arities. In particular, we have the following result.

Theorem 6
Let $\Delta \subseteq \mathscr{A}$ be a set of arities. If there are no $k \geqslant 1$ and $m \geqslant 0$ such that $\{[k,0,m]\} \in \Delta$, then the EAS problem for $\mathscr{G}(\Delta)$ is in P. Otherwise:

(A) If

(1) $\Delta \preceq \{[\infty,\infty,0],[0,0,0]\}$, or
(2) $\Delta \preceq \{[1,\infty,0],[0,\infty,\infty]\}$, or
(3) $\Delta \preceq \{[\infty,1,0],[0,1,0]\}$, or
(4) $\Delta \preceq \{[i,j,0] \in \mathscr{A} \mid i+j \leqslant 2\}$,

then the EAS problem for $\mathscr{G}(\Delta)$ is in P.

(B) Otherwise, if

(1) $\Delta \preceq \{[1,\infty,\infty],[0,\infty,\infty]\}$, or
(2) $\Delta \preceq \{[\infty,1,\infty],[0,\infty,\infty]\}$, or
(3) $\Delta \preceq \{[\infty,\infty,0],[0,\infty,0]\}$,

then the EAS problem for $\mathscr{G}(\Delta)$ is NP-complete.

(C) Otherwise, the EAS problem for $\mathscr{G}(\Delta)$ is Σ^2_P-complete.

Proof (sketch): Let us first assume that there are no $k \geqslant 1$ and $m \geqslant 0$ such that $\{[k,0,m]\} \in \Delta$, and let $P \in \mathscr{G}(\Delta)$. Then every rule in $\overline{P}$ has at least one positive atom in the body and so, $M = \emptyset$ is the unique answer set of $\overline{P}$. It can be verified in polynomial time whether $M = \emptyset$ is a model of $\overline{\overline{P}}$. Thus, the EAS problem for programs in $\mathscr{G}(\Delta)$ can be decided in polynomial time.

To prove the remaining part of the assertion, we note that the upper bound is implied directly by Theorem 2 (as $\mathscr{G}(\Delta) \subseteq \mathscr{F}(\Delta)$). To prove the lower bounds, we observe that if there are $k \geqslant 1$ and $m \geqslant 0$ such that $\{[k,0,m]\} \in \Delta$, then the EAS problem for $\mathscr{F}(\Delta)$ can be reduced to the EAS problem for $\mathscr{G}(\Delta)$. Indeed, let $P \in \mathscr{F}(\Delta)$ and let $r \in P$. Then there is $\alpha \in \Delta$ such that $\alpha_r \preceq \alpha$, where α_r is the arity of r. Having $\{[k,0,0]\} \in \Delta$, where $k \geqslant 1$, or $\{[k,0,m]\} \in \Delta$, where $k,m \geqslant 1$, allows us to "simulate" the effect of r with a rule r' of arity α obtained by repeating atoms in the head of r, and by inserting an atom a and a negated atom *not* b, where a and b are fresh, as many times as necessary in the body of r to "reach" the arity α. We also add the rule $a|\ldots|a \leftarrow$ or $a|\ldots|a \leftarrow \mathit{not}\ a', \ldots, \mathit{not}\ a'$, where a' is another fresh atom and a and *not* a' are repeated k, or k and m times, respectively. □

7 The case of supported models

Lastly, we will now consider the problem of the *Existence of the Supported Models* problem, or the *ESPM* problem, for short. Let M be a set of atoms. A logic program rule r is *M-applicable* if M satisfies the body of r (that is, satisfies every literal in the body). For a logic program P and a set of atoms M, we define $H(P,M)$ to consist

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

of the heads of all rules in $\overline{P}$ that are M-applicable. We say that a set of atoms M is a *supported* model of P if M is a minimal model of $H(P, M)$ and a model of $\overline{\overline{P}}$ (Brass and Dix 1997; Inoue and Sakama 1998). The following properties of supported models are well known.

(S1) Every supported model of a program P is a model of P.
(S2) If M is a supported model of P then every atom in M occurs in the head of some M-applicable rule in P and is the *only* atom from M in the head of that rule.
(S3) Every answer set of a program P is a supported model of P.
(S4) If all rules in $\overline{P}$ are *purely negative* (their bodies contain no nonnegated occurrences of atoms), then every supported model of P is an answer set of P.
(S5) If P is a finite proper Horn program, then P has the greatest supported model, and it can be computed in polynomial time.
(S6) The ESPM problem is NP-complete.

The main result of this section is a *dichotomy* result for the ESPM problem (by the property (S6), even for the general class of all disjunctive programs, the ESPM problem is in the class NP). Specifically, we will show that for every set $\Delta \subseteq \mathscr{A}$ the ESPM problem for programs in $\mathscr{F}(\Delta)$ is either in P or is NP-complete, and we will characterize sets Δ, for which the former holds. The proof relies on a series of lemmas, providing upper and lower bounds for the complexity of the ESPM problem for programs in $\mathscr{F}(\Delta)$ for some particular sets $\Delta \subseteq \mathscr{A}$. We note that due to our results from Section 4, it is enough to restrict attention to the case of finite sets Δ only. Thus, we adopt this assumption here.

We start with a lemma that, for some programs, connects the problem of the existence of a supported model to that of the existence of a model (and so, to the SAT problem).

Lemma 8
Let $P \in \mathscr{F}(\{[\infty, \infty, 0], [0, \infty, 0]\})$. Then P has a supported model if-and-only-if P has a model.

Proof: Because of the property (S1) above, only the "if" part needs to be proved. Thus, let us assume that M is a model of P. It follows that M is a model of $\overline{P}$. Let M' be a minimal model of $\overline{P}$ such that $M' \subseteq M$ [its existence follows by the finiteness of P, cf. the definition of $\mathscr{F}(\Delta)$]. Then M' is an answer set of $\overline{P}$ and, by the property (S3), a supported model of $\overline{P}$. Moreover, since $\overline{\overline{P}} \in \mathscr{F}(\{[0, \infty, 0]\})$ and M is a model of $\overline{\overline{P}}$, M' is a model of $\overline{\overline{P}}$. Thus, M' is a supported model of P. □

Next, we present and prove a lemma that exhibits classes of programs, for which the ESPM problem is in the class P.

Lemma 9
Let Δ satisfies at least one of the following conditions:

(1) $\Delta \preceq \{[1, 0, 0], [0, \infty, \infty]\}$,
(2) $\Delta \preceq \{[\infty, \infty, 0], [0, 0, 0]\}$,

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

(3) $\Delta \preceq \{[\infty, 1, 0], [0, 1, 0]\}$,
(4) $\Delta \preceq \{[1, \infty, 0], [0, \infty, 0]\}$,
(5) $\Delta \preceq \{[2, 0, 0], [1, 1, 0], [0, 2, 0]\}$,
(6) $\Delta \preceq \{[1, \infty, 0], [0, 0, \infty]\}$,
(7) $\Delta \preceq \{[1, 1, 0], [0, 1, \infty]\}$,

then the ESPM problem for programs in $\mathscr{F}(\Delta)$ is in P.

Proof: (1) If $P \in \mathscr{F}([1, 0, 0], [0, \infty, \infty])$, then $\overline{P}$ is just a set of facts. This set of facts, say M, is the only supported model of $\overline{P}$. Thus, if M satisfies the constraints in $\overline{\overline{P}}$, M is a supported model of P. Otherwise, P has no supported models. Since checking whether a set of atoms satisfies constraints in $\overline{\overline{P}}$ is a polynomial-time task, checking whether $P \in \mathscr{F}([1, 0, 0], [0, \infty, \infty])$ has a supported model is a polynomial-time task, too.

(2)–(5) In each of these cases, there is a polynomial-time algorithm for testing whether $P \in \mathscr{F}(\Delta)$ has a model. Thus, the claim follows from Lemma 8.

(6) To decide whether $P \in \mathscr{F}(\{[1, \infty, 0], [0, 0, \infty]\})$ has a supported model, we proceed as follows. First, we compute the greatest supported model of $\overline{P}$ [cf. the property (S5)]. Let us denote this supported model by M. Since $\overline{\overline{P}} \in \mathscr{F}(\{[0, 0, \infty]\})$, $\overline{P}$ has a supported model that satisfies $\overline{\overline{P}}$ if-and-only-if M satisfies $\overline{\overline{P}}$. Thus, once we compute M, we check whether M satisfies $\overline{\overline{P}}$. If so, we decide the ESPM problem for P in positive; otherwise, we decide the ESPM problem for P in negative. Since M can be computed in polynomial time [again, the property (S5)], the claim follows.

(7) The key to a proof is a certain transformation of programs in $\mathscr{F}(\{[1, 1, 0], [0, 1, \infty]\})$ that does not affect the status of the ESPM problem and runs in polynomial time. Let $P \in \mathscr{F}(\{[1, 1, 0], [0, 1, \infty]\})$.

(1) If P contains a fact, say a, remove all proper rules with a in the head, and all constraints with a negated occurrence of a, then remove a from all remaining rules.
(2) If there is an atom, say a, that occurs in P but never in the head of a rule, remove from P all rules with a nonnegated occurrence of a in the body and remove all negated occurrences of a from the remaining rules.
(3) If there is an atom, say a, such that $\leftarrow a$ is a constraint and a appears in the head of a proper rule in P, say $a \leftarrow b$, remove the rule from P and add the constraint $\leftarrow b$.

For each rule above, one can check that P has a supported model if and only if the program resulting from applying the rule has a supported model. It is also clear that the process of applying the steps (1)–(3) will terminate after polynomially many iterations. Indeed, in each case, when we apply a rule, we decrease the size of the program. It follows that the process runs in polynomial time. Let us denote by P' the program that results when no step is applicable.

Clearly, every atom that appears in the body of a rule in $\overline{P'}$ appears also in the head of a rule in $\overline{P'}$. Thus, the set of atoms, say H, that appear in the heads of rules in $\overline{P'}$ is a supported model of $\overline{P'}$. Moreover, for every constraint $\leftarrow a$, $a \notin H$,

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

and for every atom a occurring negated in a constraint, $a \in H$. If P' contains a contradictory constraint (empty head and empty body), then it has no models and so no supported models. Thus, P has no supported models either. Otherwise, H satisfies the constraints of P' and so is a supported model of P'. Thus, P has a supported model, too. □

The next lemma exhibits several simple classes of programs $\mathscr{F}(\Delta)$ for which the ESPM problem is NP-complete.

Lemma 10
Let Δ be one of the following sets:

(1) $\Delta = \{[1,0,1]\}$,
(2) $\Delta = \{[2,0,0],[0,0,1]\}$,
(3) $\Delta = \{[3,0,0],[0,2,0]\}$,
(4) $\Delta = \{[2,1,0],[0,2,0]\}$,
(5) $\Delta = \{[2,0,0],[1,2,0],[0,1,0]\}$,
(6) $\Delta = \{[2,0,0],[0,3,0]\}$,
(7) $\Delta = \{[1,1,0],[0,2,0],[0,0,1]\}$,
(8) $\Delta = \{[1,2,0],[0,1,0],[0,0,1]\}$.

Then the ESPM problem for programs in $\mathscr{F}(\Delta)$ is NP-complete.

Proof: (1)–(2) In each case, if $P \in \mathscr{F}(\Delta)$, then supported models of P and answer sets of P coincide [by the properties (S3) and (S4)]. Thus, the result follows from Lemma 4(1–2), respectively.

(3)–(6) In each of these cases, the ESPM problem is equivalent to the problem of the existence of a model of the program (by Lemma 8). Since CNF theories considered in Lemma 3(1–4) can be written as programs in the classes considered in (3–6), respectively, and the rewriting does not affect models, the assertions follow by Lemma 3.

(7) Let φ be a CNF formula whose every clause has three literals, and let X be the set of atoms occurring in φ. For each atom $z \in X$, we introduce a fresh atom z'. Moreover, for each clause c, we introduce a fresh atom $\hat{c}$. We define a program P_φ as follows. We include in P_φ clauses of the form $x \leftarrow x$, $x' \leftarrow x'$, and $\leftarrow x, x'$, for every $x \in X$. Next, for each clause c and each atom x that occurs nonnegated in c, we include in P_φ the clause $\hat{c} \leftarrow x$, and for each atom y that occurs negated in c, we include in P_φ the clause $\hat{c} \leftarrow y'$. Finally, we include in P_φ all clauses $\leftarrow not\ \hat{c}$, where c is a clause. Clearly, P_φ belongs to $\mathscr{F}(\{[1,1,0],[0,2,0],[0,0,1]\})$. Let M be a model of φ. We define $U = M \cup \{x' \mid x \in X \setminus M\} \cup \{\hat{c} \mid c \text{ is a clause of } \varphi\}$. One can check that U is a supported model of P_φ. Conversely, let U be a supported model of P_φ. Let $M = U \cap X$. Let c be a clause in φ. It follows that $\hat{c} \in U$. Thus, either c has a nonnegated atom x, such that $x \in U$, or a negated atom y, such that $y' \in U$. In the first case, $x \in M$, and so, M is a model of c. In the second case, $y \notin M$ (due to the constraint $\leftarrow y, y'$) and so, M is a model of c, too. Thus, M is a model of φ. It follows that the satisfiability problem for 3CNF formulas can be reduced to the ESPM problem for $\mathscr{F}(\{[1,1,0],[0,2,0],[0,0,1]\})$, which implies the assertion.

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

(8) We can use a similar reduction to the one we employed in (7). We need one more fresh atom, say f and we modify P_φ by replacing each constraint $\leftarrow x, x'$ with $f \leftarrow x, x'$, and by adding the constraint $\leftarrow f$. By this change, we "simulate" the constraints $\leftarrow x, x'$ within the syntactic confines of the class $\mathscr{F}(\{[1,2,0],[0,1,0],[0,0,1]\})$ and so, do not affect the existence of supported models. □

We are now ready to establish the main result of this section.

Theorem 7
Let $\Delta \subseteq \mathscr{A}$. The ESPM problem for $\mathscr{F}(\Delta)$ is in P if Δ satisfies at least one of the following conditions:

(1) $\Delta \preceq \{[1,0,0],[0,\infty,\infty]\}$,
(2) $\Delta \preceq \{[\infty,\infty,0],[0,0,0]\}$,
(3) $\Delta \preceq \{[\infty,1,0],[0,1,0]\}$,
(4) $\Delta \preceq \{[1,\infty,0],[0,\infty,0]\}$,
(5) $\Delta \preceq \{[2,0,0],[1,1,0],[0,2,0]\}$,
(6) $\Delta \preceq \{[1,\infty,0],[0,0,\infty]\}$,
(7) $\Delta \preceq \{[1,1,0],[0,1,\infty]\}$.

Otherwise, the ESPM problem for $\mathscr{F}(\Delta)$ is NP-complete.

Proof: If Δ satisfies any of the conditions listed above, the assertion follows by Lemma 10. Thus, let us assume otherwise. We will show that the ESPM problem for $\mathscr{F}(\Delta)$ is NP-complete. By the property (S6), it suffices to show the NP-hardness.

If $[1,0,1] \preceq \Delta$, then the NP-hardness of the ESPM problem for $\mathscr{F}(\Delta)$ follows by Lemma 10. Thus, let $[1,0,1] \not\preceq \Delta$. Under this assumption, we have $\Delta \preceq \{[\infty,\infty,0],[0,\infty,\infty]\}$.

We will first assume that $[0,0,1] \preceq \Delta$. If also $[2,0,0] \preceq \Delta$, Lemma 10(2) applies and implies the assertion. Thus, we can assume that $[2,0,0] \not\preceq \Delta$. Consequently, $\Delta \preceq \{[1,\infty,0],[0,\infty,\infty]\}$. Since $\Delta \not\preceq \{[1,\infty,0],[0,0,\infty]\}$, $[0,1,0] \preceq \Delta$. If also $[1,2,0] \preceq \Delta$, then Lemma 10(8) applies. Otherwise, $\Delta \preceq \{[1,1,0],[0,\infty,\infty]\}$. Since $\Delta \not\preceq \{[1,1,0],[0,1,\infty]\}$, $[0,2,0] \preceq \Delta$. Similarly, since $\Delta \not\preceq \{[1,0,0],[0,\infty,\infty]\}$, $[1,1,0] \preceq \Delta$. Thus, Lemma 10(7) applies.

We can therefore assume that $[0,0,1] \not\preceq \Delta$. It follows that $\Delta \preceq \{[\infty,\infty,0],[0,\infty,0]\}$. Since $\Delta \not\preceq \{[\infty,\infty,0],[0,0,0]\}$, $[0,1,0] \preceq \Delta$.

Let us assume that $[3,0,0] \preceq \Delta$. If also $[0,2,0] \preceq \Delta$, then Lemma 10(3) applies. Otherwise, we have $\Delta \preceq \{[\infty,\infty,0],[0,1,0]\}$. Since $\Delta \not\preceq \{[\infty,1,0],[0,1,0]\}$, $[1,2,0] \preceq \Delta$ and so, Lemma 10(5) applies. That completes the reasoning for the case when $[3,0,0] \preceq \Delta$. Therefore, from now on we will assume that $\Delta \preceq \{[2,\infty,0],[0,\infty,0]\}$.

Since $\Delta \not\preceq \{[1,\infty,0],[0,\infty,0]\}$, $[2,0,0] \preceq \Delta$. If $[0,3,0] \preceq \Delta$, Lemma 10(6) applies. Otherwise, $\Delta \preceq \{[2,\infty,0],[0,2,0]\}$. Since $\Delta \not\preceq \{[2,0,0],[1,1,0],[0,2,0]\}$, we have $[1,2,0] \preceq \Delta$ or $[2,1,0] \preceq \Delta$. In the former case, Lemma 10(5) applies. Thus, let us assume that $[2,1,0] \preceq \Delta$. If $[0,2,0] \preceq \Delta$, then Lemma 10(4) applies. Otherwise, we have $\Delta \preceq \{[2,\infty,0],[0,1,0]\}$. Since $\Delta \not\preceq \{[\infty,1,0],[0,1,0]\}$, it follows that $[1,2,0] \preceq \Delta$, the case that we handled above. □

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

Next, we establish the complexity of the problem to decide whether a program P credulously entails under the supported-model semantics the literal $\neg a$, where a is an atom.

Theorem 8
Let $\Delta \subseteq \mathscr{A}$. If

(1) $\Delta \preceq \{[1,0,0],[0,\infty,\infty]\}$, or
(2) $\Delta \preceq \{[\infty,1,0],[0,1,0]\}$, or
(3) $\Delta \preceq \{[1,\infty,0],[0,\infty,0]\}$, or
(4) $\Delta \preceq \{[2,0,0],[1,1,0],[0,2,0]\}$, or
(5) $\Delta \preceq \{[1,1,0],[0,1,\infty]\}$,

then the problem to decide, given an atom a and a program $P \in \mathscr{F}(\Delta)$, whether P has a supported model that does not contain a is in P. Otherwise, the problem is NP-complete.

Proof: The problem is clearly in NP. Let Δ satisfies one of the conditions listed above. Let $P \in \mathscr{F}(\Delta)$ and let a be an atom. We define $P' = P \cup \{\leftarrow a\}$. Clearly, P has a supported model that does not contain a if-and-only-if P' has a supported model. Since $P' \preceq \mathscr{F}(\Delta)$, the assertion follows by Lemma 9.

Thus, from now on, we assume that Δ does not satisfy any of the conditions (1)–(5). Let us start by considering the case $\Delta \preceq \{[\infty,\infty,0],[0,0,0]\}$. Since $\Delta \not\preceq \{[1,\infty,0],[0,\infty,0]\}$, it follows that $[2,0,0] \preceq \Delta$. Moreover, since $\Delta \not\preceq \{[\infty,1,0],[0,1,0]\}$, $[1,2,0] \preceq \Delta$. Let $P \in \mathscr{F}(\{[2,0,0],[1,2,0],[0,1,0]\})$. We select a fresh atom a and define $P' = \overline{P} \cup \{a \leftarrow bd(r) \mid r \in \overline{\overline{P}}\}$. Clearly, $P' \in \mathscr{F}(\Delta)$. Moreover, P has a supported model if-and-only- if P' has a supported model that does not contain a. Thus, the claim follows by Lemma 10(5).

Next, let us assume that $\Delta \preceq \{[1,\infty,0],[0,0,\infty]\}$. Since $\Delta \not\preceq \{[1,\infty,0],[0,\infty,0]\}$, $[0,0,1] \preceq \Delta$. Similarly, since $\Delta \not\preceq \{[1,1,0],[0,1,\infty]\}$, it follows that $[1,2,0] \preceq \Delta$. Let $P \in \mathscr{F}(\{[1,2,0],[0,1,0],[0,0,1]\})$. We select a fresh atom a and define $P' = \overline{P} \cup \{a \leftarrow b \mid \leftarrow b \in \overline{\overline{P}}\} \cup \{\leftarrow not\ b \mid \leftarrow not\ b \in \overline{\overline{P}}\}$. Clearly, $P' \in \mathscr{F}(\Delta)$. Moreover, P has a supported model if-and-only-if P' has a supported model that does not contain a. Thus, the claim follows by Lemma 10(8).

If $\Delta \not\preceq \{[\infty,\infty,0],[0,0,0]\}$ and $\Delta \not\preceq \{[1,\infty,0],[0,0,\infty]\}$ then, by Theorem 7, the ESPM problem for the class $\mathscr{F}(\Delta)$ is NP-complete. Let $P \in \mathscr{F}(\Delta)$ and let a be a fresh atom. Clearly, P has a supported model if-and-only-if P has a supported model that does not contain a. Thus the assertion follows. □

Finally, we settle the complexity of the problem to decide whether a program P skeptically entails under the supported model semantics the literal $\neg a$, where a is an atom.

Theorem 9
Let $\Delta \subseteq \mathscr{A}$. If

(1) $\Delta \preceq \{[1,0,0],[0,\infty,\infty]\}$, or
(2) $\Delta \preceq \{[1,\infty,0],[0,0,\infty]\}$, or

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

(3) $\Delta \preceq \{[1,1,0],[0,1,\infty]\}$, or
(4) $\Delta \preceq \{[1,1,0],[0,\infty,0]\}$,

then the problem to decide, given an atom a and a program $P \in \mathscr{F}(\Delta)$, whether every supported model of P satisfies $\neg a$ is in P. Otherwise, the problem is coNP-complete.

Proof: We prove the assertion by establishing the complexity of the complementary problem to decide, given an atom a and a program $P \in \mathscr{F}(\Delta)$, whether P has a supported model that contains a. We will denote this problem by $\mathscr{C}$.

Let Δ satisfies any of the conditions (1)–(3), $P \in \mathscr{F}(\Delta)$ and a be an atom. We define $P' = P \cup \{\leftarrow \mathit{not}\ a\}$. Clearly, P has a supported model containing a if-and-only-if P' has a supported model. Since $P' \in \mathscr{F}(\Delta \cup \{[0,0,1]\})$ and $\Delta \cup \{[0,0,1]\}$ falls under the same condition as Δ does, the latter property can be verified in polynomial time (cf. Lemma 9). Thus, the problem $\mathscr{C}$ can be decided in polynomial time.

Thus, let us suppose that $\Delta \preceq \{[1,1,0],[0,\infty,0]\}$. Let $P \in \mathscr{F}(\{[1,1,0]\})$. Since P is a Horn program, then the least model of P, $LM(P)$, is a supported model of P. One can show that for every atom b such that there is a sequence of rules $b_0 \leftarrow b_1; b_1 \leftarrow b_2; \ldots; b_{k-1} \leftarrow b_k$ in P, where $b_0 = b_k = b$ and $k \geqslant 1$, the set $LM(P \cup \{b\})$ is a supported model of P. Moreover, one can show that every supported model of P is the union of some of the sets $LM(P)$ and $LM(P \cup \{b\})$, where b is self-supported (in the sense that it participates in a cycle of rules, as described above). It follows that $P \in \mathscr{F}(\{[1,1,0],[0,\infty,0]\})$ has a supported model containing a if-and-only-if one of the sets $LM(\overline{P} \cup \{b\})$, where b is an atom, is a supported model of P that contains a. The latter property can be verified in polynomial time. Thus the problem $\mathscr{C}$ is in P for the class of programs $\mathscr{F}(\Delta)$, where $\Delta \preceq \{[1,1,0],[0,\infty,0]\}$).

To complete the proof, let us assume that Δ does not satisfy any of the conditions (1)–(4). If $[2,0,0] \preceq \Delta$ then the NP-completeness of the problem $\mathscr{C}$ follows by the result of Cadoli and Lenzerini (1994), in the same way as in the proof of Theorem 4 (we note that, by the properties (S3) and (S4), for programs in the class $\mathscr{F}(\{[2,0,0]\})$ supported models and answer sets coincide).

Thus, let us assume that $[2,0,0] \not\preceq \Delta$, that is, $\Delta \preceq \{[1,\infty,\infty],[0,\infty,\infty]\}$. For programs in $\mathscr{F}(\{[1,0,1]\})$, the problem of the existence of an answer set that contains a given atom a is NP-complete (it follows, in particular, from Theorem 5). Since for programs in that class answer sets and supported models coincide [by the properties (S3) and (S4)], the problem of the existence of a supported model of $P \in \mathscr{F}(\{[1,0,1]\})$ that contains a given atom a is NP-complete, too. Thus, the assertion holds for every class Δ such that $[1,0,1] \preceq \Delta$. It follows that it is enough to prove the theorem under the assumption that $\Delta \preceq \{[1,\infty,0],[0,\infty,\infty]\}$.

Since $\Delta \not\preceq \{[1,\infty,0],[0,0,\infty]\}$, $[0,1,0] \preceq \Delta$. Let us assume that $[1,2,0] \preceq \Delta$. Let $P \in \mathscr{F}(\{[1,2,0],[0,1,0],[0,0,1]\})$. Let $a_1,\ldots,a_k$ be all the atoms such that the constraint $\leftarrow \mathit{not}\ a_i$ belongs to P. We introduce fresh atoms $b_1,\ldots,b_k$ and define P' to be obtained by replacing all constraints $\leftarrow \mathit{not}\ a_i$ with rules $b_1 \leftarrow a_1$ and $b_i \leftarrow b_{i-1}, a_i$. Clearly, $P' \in \mathscr{F}(\{[1,2,0],[0,1,0]\})$. One can check that P has a supported model if-and-only-if P' has a supported model containing b_k. Thus, in this case, the problem is NP-complete. It follows that $\Delta \preceq \{[1,1,0],[0,\infty,\infty]\}$. Since

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

$\Delta \not\preceq \{[1,0,0],[0,\infty,\infty]\}$, $[1,1,0] \preceq \Delta$. Next, since $\Delta \not\preceq \{[1,1,0],[0,1,\infty]\}$, $[0,2,0] \preceq \Delta$. Finally, since $\Delta \not\preceq \{[1,1,0],[0,\infty,0]\}$, $[0,0,1] \preceq \Delta$. Thus, the ESPM problem for $\mathscr{F}(\Delta)$ is NP-complete [Lemma 10(7)]. Let $P \in \mathscr{F}(\Delta)$ and let a be a fresh atom. We define $P' = P \cup \{a \leftarrow a\}$. Clearly, P' has a supported model containing a if-and-only-if P has a supported model. Thus, the NP-completeness of the problem $\mathscr{C}$ follows. □

8 Discussion

In the paper, we studied classes of programs defined in terms of "legal" arities of rules. Specifically, we focused on classes of programs of the form $\mathscr{F}(\Delta)$, where $\Delta \subseteq \mathscr{A}$. We proved that each such class has a finite representation and, for each finite set Δ, we determined the complexity of reasoning tasks for programs from $\mathscr{F}(\Delta)$ under the answer-set semantics. We also considered briefly a related family of classes of programs, namely, those of the form $\mathscr{G}(\Delta)$, where $\Delta \subseteq \mathscr{A}$ consists of arities only, and obtained similar results for them. Our results can be regarded as *trichotomy* results as, in each case, the complexity is given by one of three complexity classes (P, NP-complete, and Σ_2^P-complete; or P, coNP-complete, and Π_2^P-complete, depending on the type of the reasoning task). We also presented similar results for the reasoning tasks under the supported-model semantics. However, in that case, we obtain *dichotomy* results — the complexity is given either by the class P or NP-complete (or, either by the class P or coNP-complete, depending on the task).

As we noted, our trichotomy and dichotomy results have some similarity to the dichotomy result by Schaefer, and its corollaries for other logic formalisms: the abductive reasoning (Nordh and Zanuttini 2008), reasoning with minimal models (Cadoli 1992) and, reasoning in default logic (Chapdelaine *et al.* 2007). The classes of theories and formulas considered in these papers are defined in terms of Boolean relations that are allowed in the language (Schaefer 1978; Creignou *et al.* 2001; Bulatov *et al.* 2005). That definition schema satisfies the dichotomy property: for every class of formulas definable in that schema, the satisfiability problem is in P, or is NP-complete. The monotonicity of the propositional logic (the set of models of the conjunction of two formulas is the intersection of the sets of models of the conjuncts) is a fundamental property required by that result. Since logic programs with the answer-set semantics do not satisfy the monotonicity property, it is unclear how to extend that formalism the approach originated by Schaefer. Thus, we based our approach on a different definition schema developed specifically for programs, and related to the "complexity" of rules as measured by the numbers of atoms in the head, and positive and negative literals in the body.

It turns out though, that some classes of programs/theories appear prominently in both settings (for instance: Horn programs and Horn theories; positive programs with no more than two literals per rule and 2CNF theories). It is then an interesting problem whether a result based on the classification in terms of types of Boolean relations can be obtained for disjunctive logic programs. One possibility might be to consider a more general setting of answer-set programs in the language of

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

propositional logic under the semantics of equilibrium models (Ferraris and Lifschitz 2005).

Acknowledgments

This paper is an extended version of the paper presented at the 10th International Conference on Logic Programming and Nonmonotonic Reasoning (Truszczyński 2009). The work was partially supported by the NSF Grant IIS-0913459. The author gratefully acknowledges several helpful comments from the anonymous referees.

References

Ben-Eliyahu, R. and Dechter, R. 1997. Propositional semantics for disjunctive logic program. *Annals of Mathematics and Artificial Intelligence 12*, 53–87.

Brass, S. and Dix, J. 1997. Characterizations of the disjunctive stable semantics by partial evaluation. *Journal of Logic Programming 32*, 3, 207–228.

Bulatov, A. A., Jeavons, P. and Krokhin, A. A. 2005. Classifying the complexity of constraints using finite algebras. *SIAM J. Comput. 34*, 3, 720–742.

Cadoli, M. 1992. The complexity of model checking for circumscriptive formulae. *Information Processing Letters 44*, 3, 113–118.

Cadoli, M. and Lenzerini, M. 1994. The complexity of propositional closed world reasoning and circumscription. *Journal of Computer and System Sciences 48*, 255–310.

Chapdelaine, P., Hermann, M. and Schnoor, I. 2007. Complexity of default logic on generalized conjunctive queries. In *Proc. of 9th International Conference on Logic Programming and Nonmonotonic Reasoning, LPNMR '07*, C. Baral, G. Brewka and J. Schlipf, Eds. Lecture Notes in Computer Science, vol. 4483. Springer, Heidelberg, 58–70.

Creignou, N., Khanna, S. and Sudan, M. 2001. *Complexity Classifications of Boolean Constraint Satisfaction Problems*. SIAM, London, UK.

Dantsin, E., Eiter, T., Gottlob, G. and Voronkov, A. 2001. Complexity and expressive power of logic programming. *ACM Computing Surveys 33*, 3, 374–425.

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

Eiter, T. and Gottlob, G. 1995. On the computational cost of disjunctive logic programming: propositional case. *Annals of Mathematics and Artificial Intelligence 15*, 3–4, 289–323.

Ferraris, P. and Lifschitz, V. 2005. Mathematical foundations of answer set programming. In *We Will Show Them! Essays in Honour of Dov Gabbay*, S. Artëmov, H. Barringer, A. d'Avila Garcez, L. C. Lamb and J. Woods, Eds. College Publications, Philadelphia, USA, 615–664.

Gelfond, M. and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Computing 9*, 365–385.

Inoue, K. and Sakama, C. 1998. Negation as failure in the head. *Journal of Logic Programming 35*, 39–78.

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

Marek, W. and Subrahmanian, V. S. 1992. The relationship between stable, supported, default and autoepistemic semantics for general logic programs. *Theoretical Computer Science 103*, 2, 365–386.

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press

Nordh, G. and Zanuttini, B. 2008. What makes propositional abduction tractable. *Artificial Intelligence 172*, 10, 1245–1284.

Schaefer, T. 1978. The complexity of satisfiability problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing, STOC '78*. 216–226.

Truszczyński, M. 2009. Trichotomy results on the complexity of reasoning with disjunctive logic programs. In *Proceedings of the 10th International Conference on Logic Programming and Nonmonotonic Reasoning, LPNMR '09*, E. Erdem, F. Lin and T. Schaub, Eds. Lecture Notes in Computer Science, vol. 5753. Springer, Heidelberg, 303–315.

https://doi.org/10.1017/S1471068410000463 Published online by Cambridge University Press