In: Proceedings of the International Logic Programming Symposium, Vancouver, October 1993, ed. D. Miller, MIT Press, pages 266–278.

Complexity Results for Disjunctive Logic Programming and Application to Nonmonotonic Logics

Thomas Eiter
Christian Doppler Laboratory for Expert Systems
Information Systems Department
Technical University of Vienna
Paniglgasse 16, A-1040 Wien, Austria
eiter@vexpert.dbai.tuwien.ac.at

Georg Gottlob
Christian Doppler Laboratory for Expert Systems
Information Systems Department
Technical University of Vienna
Paniglgasse 16, A-1040 Wien, Austria
gottlob@vexpert.dbai.tuwien.ac.at

Abstract

Ben-Eliyahu and Dechter have shown that stable semantics of a large class of extended propositional disjunctive logic programs (EDLPs) can be efficiently expressed in the language of propositional logic. They left it as an open issue whether this possible for all such programs. We provide strong evidence that this is not so, which is a consequence of a precise complexity characterization of query answering problems for EDLPs. In particular, deciding the existence of an answer set and deciding occurrence of literals in any respectively every answer sets of a finite propositional EDLP are shown to be complete for classes within the polynomial hierarchy. The results have applications to nonmonotonic logics and provide new complexity results for autoepistemic logic and disjunctive default theories.

1 Introduction

As pointed out by Gelfond and Lifschitz [5], traditional logic programming does not allow to deal directly with incomplete information, which is a shortcoming for convenient knowledge representation by logic programs. In order to overcome this limitation, they extended logic programs by permitting classical negation besides negation as failure, and they defined a stable models semantics for such programs and extended disjunctive logic programs (EDLPs). Przymusinski developed a similar extension in a more general framework [14].

Gelfond and Lifschitz emphasized that the stable models semantics of extended programs can be equivalently described by a reduction of logic programs into a fixpoint nonmonotonic formalism, and described such a reduction to Reiter's default logic [16]. The view of extended logic programs as default theories was an impetus to Ben-Eliyahu and Dechter to apply techniques developed for answering queries on default theories to EDLPs [1]. They showed that the answer set semantics of a large class of finite grounded EDLPs can be efficiently expressed in propositional logic in polynomial time. Extending previous results by Marek and Truszczyński [9, 10], they obtained that for this class of logic programs, the problems of deciding whether an answer set exists and whether a set of literals occurs in any (resp. every) answer set is efficiently reducible to deciding satisfiability (resp. provability) of a propositional formula. A similar efficient reduction for the class of all finite grounded EDLPs was left as an open issue [1].

This paper presents results which give strong evidence that this open question has a negative answer. This is a consequence of a precise complexity characterization of the above problems for finite grounded EDLPs, which are shown to be complete for certain classes of the polynomial hierarchy. Our analysis also takes care of disjunctive programs without classical negation and shows that the complexity of the problems remains unchanged.

It is well-known that EDLPs are polynomial time expressible in autoepistemic logic [14] and disjunctive default theory [6]. We exploit these relationships to obtain new complexity results for nonmonotonic logics.

The rest of this paper is organized as follows. Section 2 reviews concepts from complexity theory, the answer set semantics of EDLPs, and previous results. Section 3 contains the main results. Section 4 mentions implications to Przymusinski's stable disjunctive semantics, and Section 5 describes applications to nonmonotonic logics. Section 6 gives some conclusions.

2 Preliminaries and previous results

We assume that the reader has some background on NP-completeness, rf. [4]. Recall that P (resp. NP) is the class of problems that are deterministically (resp. nondeterministically) decidable in polynomial time, and that coNP are the problems whose complements are in NP. The classes Δ_k^P, Σ_k^P, and Π_k^P of the polynomial hierarchy are defined as follows:

$$\Delta_0^P = \Sigma_0^P = \Pi_0^P = \mathrm{P},$$

and for all $k \geq 0$,

$$\Delta_{k+1}^P = \mathrm{P}^{\Sigma_k^P},\ \ \Sigma_{k+1}^P = \mathrm{NP}^{\Sigma_k^P},\ \ \Pi_{k+1}^P = \mathrm{co}\Sigma_{k+1}^P.$$

$\mathrm{P}^{\Sigma_k^P}$ (resp. $\mathrm{NP}^{\Sigma_k^P}$) are the problems decidable in deterministic (resp. nondeterministic) polynomial time with an oracle for problems in Σ_k^P. Roughly speaking, an oracle for Σ_k^P corresponds to a subroutine that allows to solve problems in Σ_k^P in unit time. Π_k^P are the complementary problems from Σ_k^P. In particular, $\Delta_1^P = \mathrm{P}$, $\Sigma_1^P = \mathrm{NP}$, and $\Pi_1^P = \mathrm{coNP}$. Clearly $\Sigma_k^P \subseteq \Sigma_k^P \cup \Pi_k^P \subseteq \Delta_{k+1}^P \subseteq \Sigma_{k+1}^P$, but for $k \geq 1$ any equality is considered similar unlikely as $\mathrm{P} = \mathrm{NP}$. A problem is hard for a class C if every problem in C is efficiently reducible to it by a polynomial time transformation, and is in addition complete for C if it belongs to C. A well-known Σ_2^P-complete problem is deciding if a quantified Boolean formula $\exists x_1 \ldots \exists x_m \forall x_{m+1} \cdots \forall x_n E$, $n > m \geq 1$, is valid, where E is a propositional formula built from atoms $x_1, \ldots x_n$, cf. [4].

Gelfond and Lifschitz [5] define an *extended disjunctive database* (*extended disjunctive logic program* (EDLP) in [1]) as a set of rules of the form

$$L_1 | \cdots | L_k \leftarrow L_{k+1}, \ldots, L_m, not\, L_{m+1}, \ldots, not\, L_n,$$

where $n \geq m \geq k \geq 0$ and each L_i is a literal, i.e. an atom A or the classical negation $\neg A$ of an atom in a first-order language, and "*not*" is a negation-as-failure operator. The symbol "|" is used to distinguish disjunction in the head of a rule from disjunction "$\vee$" used in classical logic. We refer to EDLPs in which "$\neg$" does not occur as *disjunctive logic programs* (DLPs). EDLPs (resp. DLPs) generalize *extended logic programs* (ELPs) (resp. *general logic programs* (GLPs)), which allow only rules of type $k = 1$.

The semantics of variable-free EDLPs is defined in terms of answer sets, which correspond for GLPs one-to-one to stable models. Let as in [1] be a *context* any subset of Lit, the set of grounded literals in the language of P. First, consider variable-free EDLPs P in which "*not*" does not occur. An answer set for such a P is any minimal (in terms of set inclusion "$\subseteq$") context S such that

1. for each rule $L_1 | \cdots | L_k \leftarrow L_{k+1}, \ldots, L_m$ in P, if $L_{k+1}, \ldots, L_m \in S$, then for some $i = 1, \ldots, k$, $L_i \in S$
2. If S contains a pair of complementary literals, then $S = Lit$.

The definition is extended to variable-free EDLPs P that contain *not* as follows. Let for every context S be P^S the EDLP obtained from P by deleting

(i) each rule that has $not\, L$ in its body of with $L \in S$, and

(ii) all formulas of the form $not\, L$ of the bodies of the remaining rules.

Note that "*not*" does not occur in P^S. The answer sets of context S for P are the answer sets for P^S. If S is one of them, then S is an answer set of P. To apply the definition to an EDLP with variables, each rule has to be replaced by its grounded instances.

For example, consider the following propositional EDLP from [14], which states that everyone is pronounced not guilty unless proven otherwise:

$$\begin{array}{rcl} innocent \,|\, guilty & \leftarrow & charged \\ \neg guilty & \leftarrow & not\, proven \\ charged & \leftarrow & \end{array}$$

P has the single answer set $\{\neg guilty, innocent, charged\}$. Neither *proven* nor $\neg proven$ appears in the answer set, which informally means that nothing is known about *proven*, and both *not proven*, *not* ($\neg proven$) are assumed.

Ben-Eliyahu and Dechter studied in [1] the following three problems on EDLPs. Given an EDLP P and a context V,

Existence: does P have a consistent answer set ?

Set-Membership: does V occur in any consistent answer set of P ?

Set-Entailment: does V occur in every consistent answer set of P ?

Ben-Eliyahu and Dechter focused on grounded (i.e. propositional) EDLPs and showed that the problems are for finite head-cycle-free EDLPs (HEDLPs) efficiently reducible to testing satisfiability resp. provability of a propositional formula. In particular, they provided a polynomial time mapping of such HEDLPs into propositional formulas such that the consistent answer sets correspond to the models of a formula.

Notice that HEDLPs generalize ELPs. As Ben-Eliyahu and Dechter observe, Gelfond and Lifschitz's transformation of ELPs into default logic and complexity results for default logic in [8, 21] imply that for finite propositional ELPs, "Existence" and "Set-Membership" are NP-hard and "Set-Entailment" is coNP-hard. This establishes NP-completeness resp. coNP-completeness for finite propositional HEDLPs. Marek and Truszczyński previously showed that given a finite propositional GLP, deciding whether a stable model exists is NP-complete [9] and that deciding membership of an atom in all stable models is transformable into propositional provability [10]. (For comprehensive overviews of complexity results for logic programming and nonmonotonic reasoning see [2, 19].)

Ben-Eliyahu and Dechter leave the complexity of general EDLPs in the finite propositional case as an open issue. In particular, they pose the question whether answer set semantics semantics of all finite propositional EDLPs can be expressed in propositional logic in polynomial time [1].

3 Complexity results

We first derive upper bounds by locating the problems in appropriate classes of the polynomial hierarchy. We make use of the following lemma.

Lemma 3.1 *Given a finite propositional EDLP P and a context S, deciding whether S is a consistent answer set of P is in* coNP.[1)]

Proof. Clearly, P^S is constructible in polynomial time. S is not an answer set of P iff there exists $S' \subset S$ that satisfies 1. and 2. of the definition of answer set of P^S. A guess for S' is verifiable in polynomial time, thus deciding whether S is not an answer set is in NP, and the result follows. □

Theorem 3.2 *"Existence" for finite propositional EDLPs is in* Σ_2^P.

Proof. Membership in this class follows easily from Lemma 3.1: A guess S for a consistent answer set of P can be verified with a call to an NP-oracle, hence the problem is in $\mathrm{NP}^{\mathrm{NP}} = \Sigma_2^P$. $\Box$

Theorem 3.3 *"Set-Membership" for finite propositional EDLPs is in* Σ_2^P.

Proof. A guess for a consistent answer set S of P such that $V \subseteq S$ can be verified in polynomial time with an NP oracle. $\Box$

Theorem 3.4 *"Set-Entailment" for finite propositional EDLPs is in* Π_2^P.

Proof. A guess for a consistent answer set S of P such that $V \not\subseteq S$ can be verified in polynomial time with an NP-oracle. Thus the complement of "Set-Entailment" is in Σ_2^P, which means that "Set-Entailment" is in Π_2^P. $\Box$

The following results show that these upper bounds become lower bounds for fairly restricted classes of EDLPs.

Theorem 3.5 *"Existence" for finite propositional DLPs is* Σ_2^P*-hard, even under the following restrictions: "not" has a single occurrence in* P *and each rule head contains one or two atoms.*

Proof. We show this by the following reduction of deciding the validity of a quantified Boolean formula $F = \exists x_1 \cdots \exists x_n \forall y_1 \cdots \forall y_m E$, $n, m \geq 1$. We may assume that $E = D_1 \vee \cdots \vee D_r$ and each $D_i = L_{i,1} \wedge L_{i,2} \wedge L_{i,3}$ is a conjunction of literals $L_{i,j}$ over atoms $x_1, \ldots, x_n$, $y_1, \ldots, y_m$ as deciding if such a F is valid is still Σ_2^P-hard. Let $v_1, \ldots, v_n$ and $z_1, \ldots, z_m, w$ be new propositional atoms and define the following propositional DLP P:

$$
\begin{array}{ll}
x_i | v_i \leftarrow \; , & \text{for each } i = 1, \ldots, n \\[6pt]
y_j | z_j \leftarrow \; , \qquad y_j \leftarrow w, \qquad z_j \leftarrow w, & \\[6pt]
w \leftarrow y_j, z_j, & \text{for each } j = 1, \ldots, m \\[6pt]
w \leftarrow \sigma(L_{k,1}), \sigma(L_{k,2}), \sigma(L_{k,3}) & \text{for each } k = 1, \ldots, r \\[6pt]
w \leftarrow \mathit{not}\ w, &
\end{array}
$$

where σ maps literals from atoms $x_1, \ldots, x_n$, $y_1, \ldots, y_m$ to literals as follows:

$$
\sigma(L) = \begin{cases} v_i & \text{if } L = \neg x_i \text{ for some } i = 1, \ldots, n \\ z_j & \text{if } L = \neg y_j \text{ for some } j = 1, \ldots, m \\ L & \text{otherwise} \end{cases}
$$

Informally, v_i corresponds to $\neg x_i$ and z_j corresponds to $\neg y_j$.

If S is a consistent answer set of P, then $w \in S$ must hold (this follows from the rule $w \leftarrow \textit{not}\, w$). Thus P^S consists of all rules of P except $w \leftarrow \textit{not}\, w$. Consequently, each y_j and z_j must be in S (follows from $y_j \leftarrow w$, $z_j \leftarrow w$). Further, at least one of x_i, v_i must be in S (follows from $x_i|v_i \leftarrow$), for every $i = 1, \ldots, n$; by the minimality condition of answer set, however, at most one of x_i, v_i can be in S, for otherwise $S - \{v_i\}$ satisfies both conditions 1. and 2. for an answer set of P^S. Consequently, exactly one of x_i, v_i must be in S.

We show that P has a consistent answer set iff the formula F is valid.

"$\Rightarrow$": Assume that S is a consistent answer set of P. Let the truth assignment φ to the atoms $x_1, \ldots, x_n$ be defined by

$$\varphi(x_i) = \begin{cases} \textit{true} & \text{if } x_i \in S \\ \textit{false} & \text{if } v_i \in S \end{cases}, \quad \text{for } i = 1, \ldots, n.$$

Notice that φ is well-defined. Since S is an answer set, it follows that for each $S' \subset S$ that contains exactly one of y_j, z_j for each $j = 1, \ldots, m$ and coincides with S on $x_1, v_1, \ldots, x_n, v_n$, there must be some $i = 1, \ldots, k$ such that $\sigma(L_{i,1}), \sigma(L_{i,2}), \sigma(L_{i,3}) \in S'$. For if not, then $S' - \{w\}$ would satisfy 1. and 2. in the definition of answer set of P^S, which contradicts that S is an answer set of P^S. It follows that for every extension of φ to the atoms $y_1, \ldots, y_m$, D_i is *true* for some $i = 1, \ldots, k$, thus E evaluates to *true*. Consequently, $\exists x_1 \cdots \exists x_n \forall y_1 \cdots \forall y_m E$, i.e. F, is valid.

"$\Leftarrow$": assume that F is valid. That is, there exists a truth assignment φ to the atoms $x_1, \ldots, x_n$ such that every extension of φ to $y_1, \ldots, y_m$ satisfies E. Let S be the following context:

$$\begin{aligned} S \;=\; & \{x_i : \varphi(x_i) = \textit{true}, i = 1, \ldots, n\} \cup \{v_i : \varphi(x_i) = \textit{false}, i = 1, \ldots, n\} \cup \\ & \{y_1, z_1, \ldots, y_m, z_m, w\}. \end{aligned}$$

Notice that P^S are the rules of P except $w \leftarrow \textit{not}\, w$, and that S is consistent and satisfies 1. and 2. of the definition of an answer set of P^S. We show that there is no proper subset $S' \subset S$ of that property. Assume to the contrary that such a S' exists. We observe that S' must coincide with S on $x_1, v_1, \ldots, x_n, v_n$, that $w \notin S'$, and that exactly one of y_j, z_j is in S' for every $j = 1, \ldots, m$. By the extension property of φ, however, it follows that for some $i = 1, \ldots, k$, $\sigma(L_{i,1}), \sigma(L_{i,2}), \sigma(L_{i,3}) \in S'$. But this implies $w \in S'$, contradiction. Thus S is an answer set of P^S, and consequently S is a (consistent) answer set of P.

Since P is polynomial time constructible from F, the theorem follows. □

Theorem 3.6 *Given a finite propositional DLP P and a context V, deciding whether V occurs in any consistent answer set of P is Σ_2^P-hard, even under the following restrictions: "not" does not occur in P, each rule head contains one or two atoms, and V is a single atom.*

Proof. Reconsider the DLP P in the proof of Theorem 3.5, and recall that every consistent answer set S of P must contain w. Let P' be P except the rule $w \leftarrow not\, w$. Notice that "*not*" does not occur in P', hence $P' = P^S$ for every answer set S of P'. Thus the following statements (i) and (ii) are equivalent. (i) P has a consistent answer set. (ii) w occurs in any consistent answer set of P'. Since deciding (i) is Σ_2^P-hard, the result follows. □

Theorem 3.7 *Given a finite propositional DLP P and a context V, deciding whether V occurs in every consistent answer set of P is Π_2^P-hard, even under the following restrictions: "not" has a single occurrence in P, each rule head contains one or two atoms, and V is a single atom.*

Proof. Let a be a new propositional atom not occurring in the program P of the proof of Theorem 3.5, and define the program P' as P augmented by the rule $a \leftarrow a$. Then the following statements (i) and (ii) are equivalent. (i) P has a consistent answer set. (ii) a does not occur in every consistent answer set of P'. Since deciding (i) is Σ_2^P-hard by Theorem 3.5, it follows that deciding the complement of (ii) is Π_2^P-hard; hence the result. □

We thus obtain the following main results.

Corollary 3.8 *"Existence" for finite propositional EDLPs is Σ_2^P-complete, and Σ_2^P-hard even restricted to DLPs.*

Corollary 3.9 *"Set-Membership" for finite propositional EDLPs is Σ_2^P-complete, and Σ_2^P-hard even restricted to DLPs in which "not" does not occur.*

Corollary 3.10 *"Set-Entailment" for finite propositional EDLPs is Π_2^P-complete, and Σ_2^P-hard even restricted to DLPs.*

In order to preclude any speculations about the effect of not permitting inconsistent answer sets, we remark that the same results hold if inconsistent answer sets are included in the problem statements.

The restrictions of Theorems 3.5,3.6 are close to the boundary between Σ_2^P-hard subclasses of "Existence" (resp. "Set-Entailment") and subclasses which are in NP. This is expressed by the following theorem.

Theorem 3.11 *"Existence" (resp. "Set-Entailment") for finite propositional EDLPs is in* NP *(resp.* coNP*) if "not" does not occur in P.*

Proof. Notice that under this restriction, $P^S = P$. Thus if there is any consistent context S that satisfies 1. and 2. of an answer set for P^S, there must be a minimal one, and P has a consistent answer set. Since a guess for S can be verified in polynomial time, the result on "Existence" follows. For "Set-Entailment" it is sufficient to impose on S in addition that $V \not\subseteq S$ for a proof that V does not occur in every consistent answer set of P. Thus "Set-Entailment" is in coNP. □

We remark that the theorem can be strengthened to completeness results.

The above results give strong evidence that expressing answer set semantics semantics of all finite grounded EDLPs in propositional logic, i.e. a mapping from such an EDLPs to propositional formulas such that the answer sets of an EDLP correspond to the models of a propositional formula, is most likely not possible in polynomial time. (However, a mapping exists computable in single exponential time, i.e. in time $2^{p(\cdot)}$ where $p(\cdot)$ is a polynomial, which follows from Lemma 3.1 and the fact that every problem in NP can be solved in single exponential time.) Notice that if such a polynomial mapping were available, "Existence" and "Set-Membership" could be efficiently reduced to propositional satisfiability and set-entailment to propositional provability as done in [1]. This would imply $\Sigma_2^P = \mathrm{NP}$ (resp. $\Pi_2^P = \mathrm{coNP}$), which is considered very unlikely.

Corollary 3.12 *Unless $\Sigma_2^P = \mathrm{NP}$ (resp. $\Pi_2^P = \mathrm{coNP}$), answer set semantics of finite propositional EDLPs (or DLPs as well) cannot be expressed in propositional logic in polynomial time.*

Since disjunction-free answer set semantics *can* be expressed in propositional logic in polynomial time, we also obtain that expressing answer set semantics of disjunctive programs by general logic programs or extended logic programs in terms of corresponding stable models or answer sets, respectively, is most likely not possible in polynomial time.

In fact, Σ_2^P-hardness of "Existence" suggests the somewhat stronger statement that a consistent answer set of a finite propositional EDLP can not be found in polynomial time even with an NP oracle. The results in [1] imply that this is possible, however, if this program is an HEDLP.

4 Przymusinski's stable semantics

Przymusinski introduced in parallel with Gelfond and Lifschitz a stable model semantics for disjunctive programs which also allow classical negation [14]. However, Przymusinski's approach, which uses concepts of 3-valued logic, is more general since it includes also partial (3-valued) disjunctive models and not just total models. His construction applied to stable model semantics is roughly equivalent to the concept of answer set [5]. In particular, the concept of total stable disjunctive model for classical 2-valued interpretations is equivalent to the concept of consistent answer set. The above main results for "Existence", "Set-Membership", and "Set-Entailment" carry over to the corresponding problems of existence of a (disjunctive total) stable model, membership of literals in any stable model, and entailment of literals from all stable models. The same results can be shown if disjunctive partial stable models are used instead of disjunctive total stable models.

5 Application to Nonmonotonic Logics

We show in this section two applications of the above results to nonmonotonic logics; further applications can be found. We assume that the reader is familiar with Reiter's default logic [16] and Moore's autoepistemic logic [12].

5.1 disjunctive default logic

While an efficient embedding of ELPs into default logic is straightforward, this is not immediate if disjunctions are present [5]. (Such a translation has been most recently provided in [17].) EDLPs can be easily embedded into disjunctive default theory, however, which has been proposed in [6] as a generalization to default logic to overcome difficulties of default logic in handling disjunctive information.

A disjunctive default is a rule of the form

$$\frac{\alpha : \beta_1, \ldots, \beta_m}{\gamma_1 | \cdots | \gamma_n},$$

which generalizes the familiar default rule ($n = 1$) and allows to conclude one of the *consequents* γ_i if the *prerequisite* α is derivable and for each *justification* β_j its negation $\neg\beta_j$ is not derivable. A disjunctive default theory is a set of such defaults.

From recent complexity results for default logic [7, 22], it follows that the main reasoning tasks in disjunctive default logic (existence of extensions, brave reasoning, and cautious reasoning) are Σ_2^P-hard resp. Π_2^P-hard if arbitrary propositional formulas are permitted to appear in the defaults. Our results on EDLPs allow to strengthen these results considerably, namely to disjunctive default theories where all α, β_i and γ_j are conjunctions of literals. Notice that for classical default logic, the results in [8] imply that the problems under this restriction are NP-complete resp. coNP-complete.

A propositional EDLP P can be translated into a disjunctive default theory $emb_D(P)$ by replacing every rule

$$L_1 | \cdots | L_k \leftarrow L_{k+1}, \ldots, L_m, not\, L_{m+1}, \ldots, not\, L_n,$$

with the disjunctive default

$$\frac{L_{k+1} \wedge \cdots \wedge L_m : \overline{L}_{m+1}, \ldots, \overline{L}_n}{L_1 | \cdots | L_k}, \tag{1}$$

where $\overline{L}$ is the opposite literal of L.

Proposition 5.1 [6] *Let P be a propositional EDLP. Then S is an answer set of P iff S is the set of literals from an extension of $emb_D(P)$.*

From the results in Section 3, we thus immediately obtain the following result.

Theorem 5.2 *Let D be a finite propositional disjunctive default theory such that each prerequisite α and all justifications β_i and consequents γ_j occurring in D are conjunctions of literals. Then, deciding (i) whether D has an extension is Σ_2^P-hard; (ii) whether a given literal L belongs to some extension of D is Σ_2^P-hard; and (iii) whether a given literal L occurs in every extension of D is Π_2^P-hard.*

These results hold if "consistent extension" is replaced with "extension".

5.2 autoepistemic logic

Propositional DLPs without empty heads can be embedded into autoepistemic logic using a construction in [14] as follows. Every rule

$$L_1|\cdots|L_k \leftarrow L_{k+1},\ldots,L_m, not\, L_{m+1},\ldots, not\, L_n,$$

of a propositional DLP P (note that all L_i are atoms) is replaced with the formula

$$L_{k+1} \wedge \cdots \wedge L_m \wedge \neg \mathbf{L} L_{m+1} \wedge \cdots \wedge \neg \mathbf{L} L_n \Rightarrow L_1 \vee \cdots \vee L_k, \tag{2}$$

($\mathbf{L}$ is the belief operator) and for every propositional atom p the formula

$$p \Rightarrow \mathbf{L}p$$

is added to obtain the outcome $emb_A(P)$ of the transformation. It is easy to see that p or $\neg p$ must belong to every stable set containing $emb_A(P)$. Hence the consistent stable expansions, the consistent moderately grounded expansions, and the consistent parsimonious stable expansions of $emb_A(P)$ as defined in [3] coincide. The results in [14] on the correspondence between disjunctive stable models of P and stable expansions of $emb_A(P)$ entail the following.

Proposition 5.3 *There is a one-to-one correspondence between consistent answer sets of a propositional DLP P without empty heads and consistent stable expansions of $emb_A(P)$, such that an atom occurs in an answer set iff it occurs in the corresponding consistent stable expansion.*

This proposition and the results in Section 3 imply the following results for the class of autoepistemic theories consisting of disjunctions $D_1 \vee \cdots \vee D_n$, where each D_i is a literal or a modal literal $\mathbf{L}L$ or $\neg\mathbf{L}L$ for L a literal. Call such theories *disjunctive autoepistemic literal theories* (DALT).

Theorem 5.4 *Given a finite propositional DALT T, deciding whether (i) T has a consistent stable expansion is Σ_2^P-hard; (ii) whether a given atom p occurs in some consistent stable expansion of T is Σ_2^P-hard; (iii) whether a given atom p occurs in every consistent stable expansion of T is Π_2^P-hard.*

Note that Theorem 5.4 considerably strengthens the results in [7] which show the same complexity bounds for autoepistemic theories but require that arbitrary propositional formulas can occur in the theory. Theorem 5.4 holds if "consistent" is cancelled and extends by the results in [13] to completeness results. Furthermore, the results of Theorem 5.4 analogously apply to moderately grounded and parsimonious stable expansions. Notice that based on the results of [3], Schaerf was able to derive similar complexity results for these variants of autoepistemic logic [18].

Furthermore, complexity results for restricted fragments of reflexive autoepistemic logic [20] and 3-valued autoepistemic logic [15] can be obtained by embeddings of EDLPs into reflexive autoepistemic logic [11] and of DLPs into 3-valued autoepistemic logic [14].

6 Conclusions

This paper provides a precise complexity characterization of computational problems on extended disjunctive logic programs (EDLPs), i.e. disjunctive programs which permit negation-by-failure and classical negation [5], in terms of completeness results for classes of the polynomial hierarchy. The results suggest a negative answer to the question in [1] whether stable semantics of such EDLPs can be expressed in propositional logic in polynomial time. The complexity results for EDLPs apply also for disjunctive logic programs without classical negation, which demonstrates (as already observed by Gelfond and Lifschitz) that allowing classical negation does not increase the complexity of stable model semantics. Similar results carry over to Przymusinksi's total stable disjunctive model semantics [14].

Intuitively, stable model semantics feeds two interacting sources of complexity in disjunctive logic programs: disjunction in the head of the rules and the minimality condition of answer set, i.e. stable model. The problem with disjunctive heads in finding an answer set seems to be deciding how many (and which) literals must be included. Ben-Eliyahu and Dechter's work on HEDLPs provides insights by showing that this source of complexity takes no effect if, loosely speaking, pairs of literals from the head of the same rule are not involved in cyclic proofs.

We feel that the results of this paper support a better understanding of the computational properties of finite propositional disjunctive logic programs and provide a reference for issues of expressibility and intertranslatability. For example, the results imply that disjunctive stable semantics is most likely not expressible by stable semantics of disjunction-free logic programs in polynomial time. On the other hand, the results of this paper and recent complexity results for nonmonotonic logics [7, 2] imply that queries to EDLPs can be efficiently translated into reasoning tasks in many forms of nonmonotonic reasoning; for disjunctive default theory and autoepistemic logic we obtained by such transformations new complexity results.

Vice versa, the results imply that reasoning tasks in many nonmonotonic formalisms can be efficiently reduced to disjunctive logic programming. The computational relationship underlines the close connection between disjunctive logic programming and nonmonotonic logics, and supports that logic programming is a competitive tool for knowledge representation.

Acknowledgment

The authors would like to thank Chiaki Sakama for his comments on a previous version of this paper.

[1] This problem is also coNP-hard as follows implicitly from the proof of Theorem 3.5, but this is not needed in the sequel.

References

[1] R. Ben-Eliyahu and R. Dechter. Propositional Semantics for Disjunctive Logic Programs. In *Proceedings JICSLP-92*, Washington DC, November 1992. Full paper to appear in *Annals of Mathematics and Artificial Intelligence*.

[2] M. Cadoli and M. Schaerf. A Survey on Complexity Results for Nonmonotonic Logics. Technical report, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", 1992. Journal of Logic Programming, to appear.

[3] T. Eiter and G. Gottlob. Reasoning with Parsimonious and Moderately Grounded Expansions. *Fundamenta Informaticae*, 17(1,2):31–53, 1992.

[4] M. Garey and D. S. Johnson. *Computers and Intractability – A Guide to the Theory of NP-Completeness.* W. H. Freeman, New York, 1979.

[5] M. Gelfond and V. Lifschitz. Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Computing*, 9:365–385, 1991.

[6] M. Gelfond, H. Przymusinska, V. Lifschitz, and M. Truszczyński. Disjunctive Defaults. In *Proceedings KR-91*, pages 230–237, 1991.

[7] G. Gottlob. Complexity Results for Nonmonotonic Logics. *Journal of Logic and Computation*, 2(3):397–425, June 1992.

[8] H. Kautz and B. Selman. Hard Problems for Simple Default Logics. *Artificial Intelligence*, 49:243–279, 1991.

[9] W. Marek and M. Truszczyński. Autoepistemic Logic. *Journal of the ACM*, 38(3):588–619, 1991.

[10] W. Marek and M. Truszczyński. Computing Intersection of Autoepistemic Expansions. In *Proceedings of the 1st Intl. Workshop on Logic Programming and Nonmonotonic Reasoning*, pages 37–50, Washington DC, July 1991.

[11] W. Marek and M. Truszczyński. Reflexive Autoepistemic Logic and Logic Programming. Technical Report 222-92, Computer Science Department, University of Lexington, Kentucky, November 1992.

[12] R. Moore. Semantical Considerations on Nonmonotonic Logics. *Artificial Intelligence*, 25:75–94, 1985.

[13] I. Niemelä. Towards Automatic Reasoning. In *Proc. Europ. Workshop on Logics in AI, Amsterdam, September 1990*, number 478 in Lecture Notes in Computer Science. Springer, 1991.

[14] T. Przymusinski. Stable Semantics for Disjunctive Programs. *New Generation Computing*, 9:401–424, 1991.

[15] T. Przymusinski. Three-valued Nonmonotonic Formalisms and Semantics of Logic Programming. *Artificial Intelligence*, 49:309–344, 1991.

[16] R. Reiter. A Logic for Default Reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[17] C. Sakama and K. Inoue. Relating Disjunctive Logic Programs to Default Theories. In L.-M. Pereira and A. Nerode, editors, *Proceedings of the Second International Workshop on Logic Programming and Nonmonotonic Reasoning*, pages 266–282, Lisbon, Portugal, June 1993. MIT Press.

[18] M. Schaerf. Logic Programming and Autoepistemic Logics: New Relations and Complexity Results. Submitted, December 1992.

[19] J. Schlipf. A Survey of Complexity and Undecidability Results in Logic Programming. In H. Blair, W. Marek, A. Nerode, and J. Remmel, editors, *Proceedings of the Workshop on Structural Complexity and Recursion-Theoretic Methods in Logic Programming*, pages 93–102, Washington DC, November 13 1992. Cornell University, Mathematical Sciences Institute.

[20] G. Schwarz. Autoepistemic Logic of Knowledge. In *Proceedings of the 1st Intl. Workshop on Logic Programming and Nonmonotonic Reasoning*, pages 260–274, Washington DC, July 1991.

[21] J. Stillman. It's Not My Default: The Complexity of Membership Problems in Restricted Propositional Default Logic. In *Proceedings AAAI-90*, pages 571–579, 1990.

[22] J. Stillman. The Complexity of Propositional Default Logic. In *Proceedings AAAI-92*, pages 794–799, 1992.