ELSEVIER

Artificial Intelligence 69 (1994) 347–357

Artificial Intelligence

Research Note

Default theories that always have extensions ☆

Christos H. Papadimitriou [a,*] Martha Sideri [b]

[a] *Department of Computer Science and Engineering, University of California, 9500 Gilman Drive, La Jolla, CA 92093-0114, USA*

[b] *Department of Computer Science, Athens University of Economics and Business, Athens, Greece*

Received March 1993; revised May 1994

Abstract

Default logic is an important and influential formalization of commonsense reasoning. Determining whether a given default theory has an extension is the main computational problem pertinent to default logic, the analog of testing for validity and finding deductions in more classical logics. Substantially generalizing a result by Etherington [5] we show that all default theories that have no odd cycles (in some precise sense) have an extension, which can be found efficiently. We also give a proof that it is NP-complete to find extensions even for default theories with no prerequisites and at most two literals per default, a case substantially simpler than the NP-completeness results in the literature.

1. Introduction

Default logic, originally proposed by Reiter [13], is a powerful and influential formalization of knowledge representation and nonmonotonic reasoning. The basic knowledge representation mechanism in default logic is the *default*, an object of the form

$$\delta = \frac{\phi : \psi \,\&\, \chi}{\chi}, \tag{1}$$

☆ Research partially performed while the authors visited the Laboratoire de Réchérche en Informatique at the Université de Paris Sud, and supported by the ESPRIT Project ALCOM and the National Science Foundation.

* Corresponding author. E-mail: christos@cs.ucsd.edu.

SSDI 0004-3702(94)00038-3

where ϕ, χ, and ψ are Boolean expressions in conjunctive normal form (this class of defaults is referred to in the literature as *propositional seminormal defaults*). Intuitively, default (1) above means that *if ϕ has been established, and neither $\neg\psi$ nor $\neg\chi$ have, then we can "assume χ by default"*. The intended use of defaults is exemplified as follows:

$$\frac{\mathrm{bird}(x) : \neg\mathrm{penguin}(x) \ \&\ \mathrm{flies}(x)}{\mathrm{flies}(x)}.$$

ϕ is called the *prerequisite* of δ, denoted $P(\delta)$, ψ the *justification*, $J'(\delta)$, and χ the *consequence* $C(\delta)$.

A *(propositional seminormal) default theory* is now defined to be a pair $D = (\alpha_0, \Delta)$, where α_0 is an expression (intuitively, comprising our initial knowledge of the world), and Δ is a set of defaults $\Delta = \{\delta_1, \delta_2, \ldots, \delta_k\}$, as in (1) above.

The semantics (but also the proof theory) of a default theory is defined in terms of a powerful concept called an *extension*, defined next. Given a default theory (α_0, Δ) and a subset $A \subseteq \Delta$ of its defaults, define $\alpha[A] = \alpha_0 \wedge \bigwedge\{C(\delta) : \delta \in A\}$ (we add to α_0 all clauses in the consequences of all defaults in A). Also, if A and B are finite sets of defaults and δ a default, we say that (A, B) *activates* the default δ (written $(A, B) \models \delta$) if $\alpha[A] \models P(\delta)$ and $\alpha[B] \not\models \neg(J'(\delta) \wedge C(\delta))\}$; that is, if the prerequisite of δ is logically implied by $\alpha[A]$, and its consequence and justification are not falsified by $\alpha[B]$. If the consequence or justification of a default δ is falsified by $\alpha[B]$, then we say that B *obstructs* δ.

Next, let $(\alpha, \Delta, B)^*$ denote the limit of the following sequence of sets of defaults: $S_0 = \emptyset$, and $S_{i+1} := S_i \cup \{\delta \in \Delta : (S_i, B) \models \delta\}$—the smallest set S of defaults which contains all defaults activated by (S, B). Finally, we say that S is an *extension* of the default theory (α_0, Δ) if $S = (\alpha_0, \Delta, S)^*$ (usually in the literature the expression $\alpha[S]$, and not S itself, is called an extension; this slight twisting of standard terminology is convenient for our point of view and methodology, and otherwise inconsequential). Notice that the sought extension S appears in the definition and the iteration—in other words, the computation of extensions is no easy monotonic operator, as in Horn logic (or the inflationary semantics of negation, see [7]). Obviously the recurrence above must converge after k or fewer steps, where k is the number of defaults in Δ, but not necessarily to S; if not, S fails to be an extension. A default theory may have many, one or no extensions.

Intuitively, the equation $S = (\alpha_0, \Delta, S)^*$ means that all defaults in S can be eventually activated via the iteration above, while all other defaults fail to become eventually activated.

There is a basic computational problem suggested by default logic: "Given a default theory, find an extension, if it has one." This problem has been analyzed extensively in the past from the standpoint of computational complexity [3, 5, 8, 14]. In these works, broad subclasses of default theories for which the extension problem is tractable have been identified, most notably *ordered* default theories [5, 14]—the analog of stratified logic programs [2]. In this paper we show that *even default theories*, a more general class of default theories than the class of ordered default theories, are guaranteed to always have an extension (Theorem 5). Even default theories extend the ordered ones in the following way: Ordered default theories disallow all positive-length cycles in a weighted directed graph reflecting the structure of the default theory; even theories disallow all *positive odd-length* cycles. For example, the two defaults

$$\frac{\text{bird}(x) : \neg\text{penguin}(x)\ \&\ \text{flies}(x)}{\text{flies}(x)},$$
$$\frac{\text{bird}(x) : \text{penguin}(x)\ \&\ \neg\text{flies}(x)}{\text{penguin}(x)},$$

together with the initial database $\alpha_0 = \text{bird}(x)$, comprise a theory which is not ordered (intuitively, there is no way to order the predicates penguin(x) and flies(x) so that the earlier one does not adversely affect the truth of the latter one). Still the default theory obviously does have an extension—in fact, as it is typical in even theories, and comes out rather elegantly in our proof, it has *two mutually contradicting extensions*. The intuition behind our result is this: Even-length cycles denote chains of influences with an even number of adverse influences, *which presumably cancel*; this intuition is quite tricky to formalize and prove.

A parenthesis on logic programming is perhaps appropriate at this point. There are fascinating similarities, sometimes downright mathematical equivalence, between default theories and their extensions on the one hand, and logic programs with negation and their stable models on the other; see for example [1] for an extensive discussion of this parallel. The analog of Etherington's *ordered* default theories in the logic programming domain are the *stratified* programs of [2]. A generalization of stratified programs to ones with no odd cycles had been suggested in [9] (in fact, the result in [9] can be seen as a special case of ours, one that applies to disjunction-free defaults). Our work on even default theories has more recently inspired work on the semantics of logic programming with negation, leading to a new, more powerful semantics of logic programs with negation, and the precise characterization of *totality* (the property of a logic program to always have a fixpoint model) [12]. In other words, it is shown in [12] that "even logic programs" are in some sense the most general well-defined logic programs with negation. This throws a negative light on the possibility of identifying even more general subclasses of default theories that are guaranteed to have an extension.

The results in this paper were presented in preliminary form in [11].

2. Even default theories

We shall now define certain graphs which capture useful properties of default theories. Let $D = (\alpha_0, \Delta)$ be a default theory. Its *literal graph* $L(D)$ has the literals as nodes, and has a directed edge (x, y) if both $\neg x$ and y appear in the same clause of α_0 or of a consequence of D. Intuitively, this graph captures which literals can contribute in the proof of others in the context of D; it is analogous to a graph considered in [5]. We write $L^*(x, y)$, omitting explicit reference to D, whenever there is a path from x to y in $L(D)$ (this includes the possibility $x = y$).

$G(D) = (\Delta, E)$, is a directed graph *with the defaults as nodes*, and whose set of arcs E is partitioned into two disjoint sets $E = E_0 \cup E_1$—the *arcs with weight zero and those of weight one*, respectively. This graph is constructed from D as follows:

(a) There is an arc $(\delta, \delta') \in E_0$ if there is a literal x appearing positively in the consequence of δ, a literal y which appears also positively in the prerequisite of δ', and $L^*(x, y)$.

(b) There is an arc $(\delta, \delta') \in E_1$ if there is a literal x that appears positively in the consequence of δ, a literal y which appears negatively in the justification or in the consequence of δ', and $L^*(x, y)$.

$G(D)$ generalizes the graph defined in [3] (defined there for the disjunction-free case, in which L^* is the identity). Intuitively, $(\delta, \delta') \in E$ means that the activation of δ (that is, the addition of its consequence in α_i at some stage i) may possibly affect the activation of δ'. An edge in E_0 implies positive influence, whereas an edge in E_1 means adverse influence. In case (a), for example, adding δ to α_i may affect positively the addition of δ' in a future stage by satisfying a clause of its prerequisite, while in (b) the effect is negative: The activation of δ may obstruct a possible future activation of δ'. This intuition is articulated in the following lemma, where $A, B \subseteq \Delta$ and $+$ denotes union with a singleton:

Lemma 1.
(a) *Suppose that* $(A, B) \not\models \delta'$ *and* $(A + \delta, B) \models \delta'$. *Then* $(\delta, \delta') \in E_0$.
(b) *Suppose that* $(A, B) \models \delta'$ *and* $(A, B + \delta) \not\models \delta'$. *Then* $(\delta, \delta') \in E_1$.

Proof. (a) By definition we have that $\alpha[A] \not\models P(\delta')$ and $\alpha[A + \delta] \not\models P(\delta')$. Since resolution is a complete proof system for Boolean expressions in conjunctive normal form, and since δ is necessary for the activation of δ, this means that one of the clauses of $C(\delta)$ participates in a proof by resolution of a clause in $P(\delta')$. The following is a basic property of resolution proofs easily proved by induction on proof length:

If a clause C participates in a resolution proof of clause C', then for all literals x of C and x' of C' we have $L^*(x, x')$.

Hence, the resolution proof of a clause of $P(\delta')$ using a clause of $C(\delta)$ immediately implies the existence of a path in $L(D)$ from a literal of $C(\delta)$ to a literal of $P(\delta)$; but this is precisely what it means for (δ, δ') to be an edge in E_0.

(b) We have $\alpha[C] \not\models \neg(J'(\delta') \wedge C(\delta'))$ whereas $\alpha[C+\delta] \models \neg(J'(\delta') \wedge C(\delta'))$. Therefore a literal in some clause of $C(\delta)$ must figure in a resolution proof of $\neg(J'(\delta') \wedge C(\delta'))$, and thus there must be a path in $L(D)$ from that literal of $C(\delta)$ to some negated literal of $(J'(\delta') \wedge C(\delta'))$. It follows that $(\delta, \delta') \in E_1$. □

The *length* of a path or cycle P of $G(D)$ is the total number of edges of weight one in it (that is, the cardinality of $P \cap E_1$). A default theory D is called *acyclic* if there are no cycles of positive length in $G(D)$. It is called *even* if there are no cycles of odd length in $G(D)$. Acyclic theories coincide with the *ordered* theories of [5]; even theories are obviously a generalization.

Example 2. Consider the following default theory (α_0, Δ), where α_0 is the clause $(z \vee w)$, and $\Delta = \{\delta_1, \delta_2, \delta_3, \delta_4\}$. Here

$$\delta_1 = \frac{w : z \,\&\, x}{x}, \qquad \delta_2 = \frac{x : (w \vee u)}{(w \vee u)},$$
$$\delta_3 = \frac{: \neg w \,\&\, \neg z}{\neg z}, \qquad \delta_4 = \frac{u : z \,\&\, y}{y}.$$

The graph $L(G)$ consists of the two paths shown in Fig. 1(a) and the graph $G(D)$ is shown in Fig. 1(b). Notice that $G(D)$ is even. As predicted by our main result, it does have an extension: $\{\delta_3\}$.

Example 3. Recall the two conflicting defaults in theory (1), $D = (\text{bird}(x), \{\delta_1, \delta_2\})$. The graph $L(D)$ has no edges (all consequences are disjunction-free), and thus L^* is the identity. The graph $G(D)$ is shown in Fig. 1(c). As with all disjunction-free theories, all edges are of weight one. Defaults correspond to *kernels* [3,4], that is, independent dominating sets, of the graph. D has two defaults, consisting of the two singletons $\{\delta_1\}$ and $\{\delta_2\}$.

Example 4. Consider the following eight defaults (α_0 is true).

$$\delta_0 = \frac{: a \,\&\, \neg c}{a}, \quad \delta_1 = \frac{: b \,\&\, \neg a}{b}, \quad \delta_2 = \frac{: \neg e \,\&\, \neg b}{\neg b}, \quad \delta_3 = \frac{: d \,\&\, b}{d},$$
$$\delta_4 = \frac{e \,\&\, \neg d}{e}, \quad \delta_5 = \frac{: \neg g \,\&\, \neg b}{\neg b}, \quad \delta_6 = \frac{: f \,\&\, b}{f}, \quad \delta_7 = \frac{: g \,\&\, \neg f}{g}.$$

They are again disjunction-free, and thus $L(D)$ is empty. $G(D)$ is shown in Fig. 1(d); it does have odd cycles, and thus existence of defaults is not guaranteed by our main result below. As the reader is invited to check, there is only one extension: $\{\delta_1, \delta_3, \delta_6\}$. Thus, if other considerations force us to include 0 in the extension (see the proof of Theorem 9 below), then there is no extension.

Our main result, stated next, generalizes that in [5]:

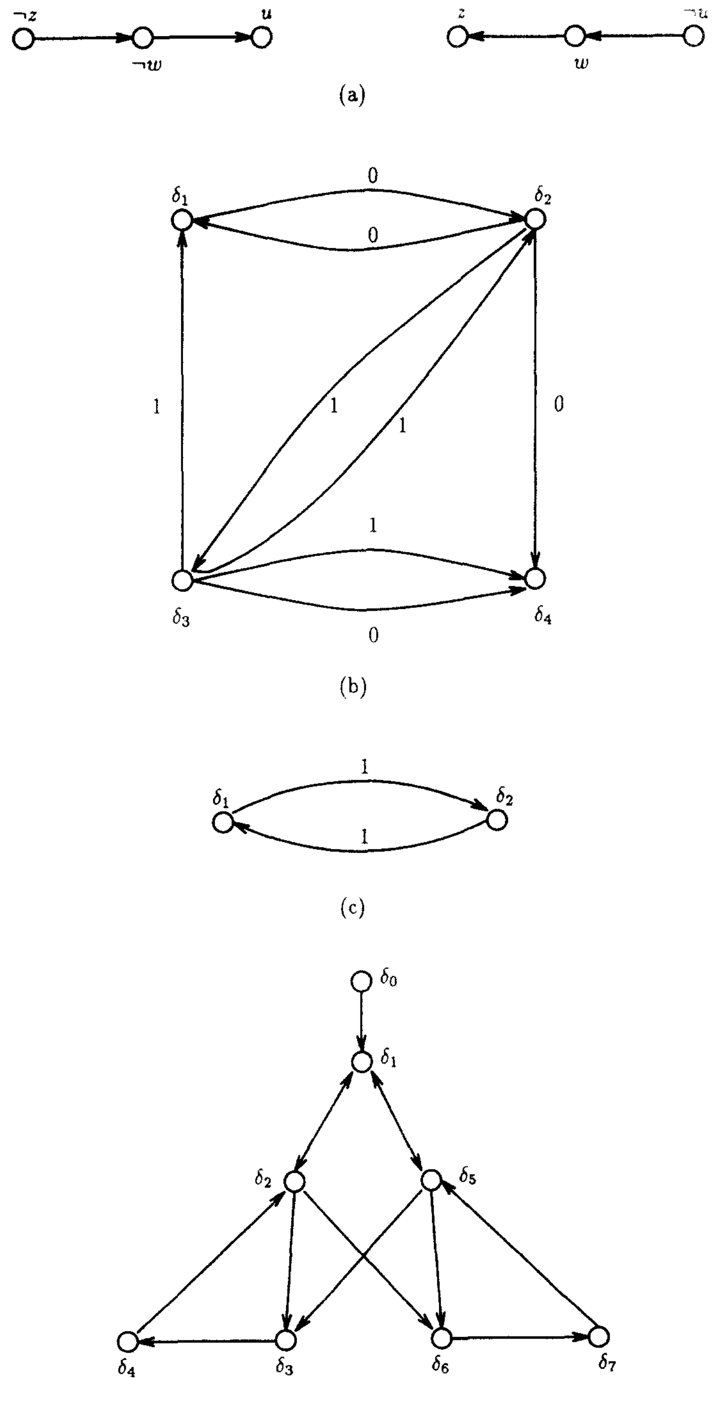

Fig. 1. Defaults theories and graphs.

Theorem 5. *Every even default theory has an extension.*

The following is the basic graph-theoretic fact on which our construction and proof is based:

Lemma 6. *If M is a strongly connected subgraph of $G(D)$ with no odd cycles, then the nodes of M can be partitioned into two disjoint subsets $M = K \cup L$ such that (a) all zero-weight edges are either within K or within L, and (b) all unit-weight edges are from K to L or back.*

The even graph $G(D)$ in Fig. 1(b) has a strongly connected component $\{\delta_1, \delta_2, \delta_3\}$. The two "subcomponents" predicted by Lemma 6 are $\{\delta_1, \delta_2\}$ and $\{\delta_3\}$. In the even graph in Fig. 1(c) the two subcomponents are the two singletons.

Proof. We shall first prove that M has no odd *pseudocycles*, that is, odd cycles of the underlying undirected graph. A pseudocycle has an even number k of reversals of direction; we show the result by induction on k. It certainly holds when k is zero (we have a cycle). For the induction step, consider an odd pseudocycle Ψ with $k > 0$ reversals, and consider a maximal path P from u to v on this pseudocycle. Since M is strongly connected, there is a path P' from v to u. Since M has no odd cycles, P' has the same parity as P. Replacing P by P' in Ψ, we obtain an odd pseudocycle Ψ' with $k - 2$ reversals, contradicting the induction hypothesis.

Hence the underlying graph of M has no odd cycles. Fix a node k of M, and define K to be the set of nodes that are an even distance away from k in the underlying undirected graph, and L the set of nodes that are an odd distance away from k. These sets obviously exhaust the nodes of M. We claim that they are also disjoint. In proof, if m is in both K and L, then there are paths of even *and* odd length from k to m, and these two paths define an odd cycle. It is easy to verify that this partition of the nodes of M satisfies the required conditions. □

Proof of Theorem 5. Let $D = (\alpha, \Delta)$ be an even default theory, and $G(D)$ its graph. We construct two nondecreasing sequences of subsets of Δ, $V_0 = S_0 = \emptyset$, $S_1 \subseteq V_1$, $S_2 \subseteq V_2, \ldots$. Informally, V_i are the defaults that we have considered so far, while S_i are the defaults that we have decided to include in the extension under construction. $G|_S$ denotes the subgraph of $G(D)$ induced by the node set S.

For $i = 0, 1, \ldots$, consider $G_i = G(D)|_{\Delta - V_i}$ (that is, the graph of the not-yet-considered defaults), and its strongly connected components. Choose a strongly connected component with no incoming edge, say $S = (M, F_0 \cup F_1)$, where F_0 contains the arcs of weight zero, and F_1 those of weight one. M can be partitioned into two sets K and L, as prescribed in Lemma 6. There are three cases:

Case 1. There is a default $\delta \in M$ (without loss of generality $\delta \in K$) such that $(V_i, V_i \cup L) \models \delta$. In this case we set $V_{i+1} = V_i + \delta$ and $S_{i+1} = S_i + \delta$. That is, if a default would be activated even if all of its obstructions (the defaults in L) are activated and none of its supporting defaults (those in K) is activated, then we must add it to our extension.

Case 2. There is a default $\delta \in M$ (again, without loss of generality $\delta \in K$) such that $\delta \notin (\alpha[S_i], S_i \cup K, S_i)^*$. In this case we set $V_{i+1} = V_i \cup \{\delta\}$ and $S_{i+1} = S_i$. That is, if a default would not be eventually activated even if none of its obstructions in M is activated, then we simply delete it from further consideration.

Case 3. Finally, if no such defaults exist, we let $V_i = V_{i-1} \cup K \cup L$ and $S_i = S_{i-1} \cup K$. That is, we pick any one of the two sides of M (without loss of generality, K) and activate all of its defaults, deleting the rest of M from further consideration.

Obviously this process terminates at some step m, where $V_m = \Delta$. We claim that S_m is an extension of $D = (\alpha_0, \Delta)$. In fact, we shall prove by induction on i that S_i is an extension of the default theory (α_0, V_i). The basis $i = 0$ is trivial.

For the induction step, suppose that indeed S_i is an extension of the default theory (α_0, V_i); we claim that S_{i+1} is an extension of (α_0, V_{i+1}). We must establish that $(\alpha_0, V_i, S_i)^* = S_i$ implies $(\alpha_0, V_{i+1}, S_{i+1})^* = S_{i+1}$; that is, S_{i+1} activates the same defaults in V_i as S_i did, and furthermore, from among the defaults in $V_{i+1} - V_i$, it activates precisely the defaults in $S_{i-1} - S_i$. This is certainly true if $S_{i+1} = S_i$ (Case 2). Otherwise, there are new defaults in $S_{i+1} - S_i$. Suppose that these defaults affect the status (activated or not) of a default $\delta \in V_i$ considered at some previous stage, say $j \leqslant i$. By Lemma 1, this implies that there is an edge in $G(D)$ from a default in $S_{i+1} - S_i$ to δ. Since δ belongs to a strongly connected component of G_j with no incoming edges, this means that the defaults in $S_{i+1} - S_i$ are in the same strongly connected component of G_j as δ. Now, if δ was added to V_j by Case 1, then we know that none of its predecessors can obstruct it (by the condition of Case 1); so the induction step holds. Finally, if δ was added or deleted under Case 3, then all of its predecessors were either added or deleted at the same step, and so δ has no predecessors in $S_{i+1} - S_i$. We conclude that newly activated defaults in $S_{i+1} - S_i$ (if any) cannot change the status (activated or not) of the defaults in V_i, that is, $V_i \cap (\alpha_0, V_{i+1}, S_{i+1})^* = S_i$.

Finally, consider any default $\delta \in V_{i+1} - V_i$. We claim that S_{i+1} activates δ if and only if $\delta \in S_{i+1}$. If δ was introduced under Case 1, then this is certainly true, as S_i already activated δ. Similarly for Case 2, since $S_{i+1} = S_i$, and S_i failed to activate δ. If δ was in the set K of Case 3, then S_{i+1} will eventually activate δ because, otherwise, Case 2 would have obtained. Finally, if δ was in the set L of Case 3, then S_{i+1} fails to activate δ because, otherwise, Case 1 would be applicable. The induction is complete. □

This proof has certain positive computational implications. We start with one of mostly theoretical interest. The general problem of finding extensions of default theories is known to be complete for the Σ_2^p level of the polynomial hierarchy. Σ_2^p, or $\mathrm{NP}^{\mathrm{NP}}$, is the class of languages recognized in nondeterministic polynomial time with the help of an NP oracle [10]. Completeness for this class is considered an even bleaker status than NP-completeness. This result was proved independently by Stillman [15], Gottlob [6], and the authors [11].

In the case of even default theories we are at least a little lower:

Corollary 7. *Finding extensions of an even default theory is in* $\Delta_2^p = \mathrm{P}^{\mathrm{NP}}$.

We conjecture that the problem of finding an extension in a general even theory is in fact Δ_2^p-complete. Perhaps more interestingly, further special cases of the extension problem can be solved in polynomial time:

Corollary 8. *An extension of a default theory* $D = (\alpha_0, \Delta)$ *with even* $G(D)$, *where, for all* $\delta \in \Delta$, $C(\delta)$ *and* $J'(\delta)$ *are conjunctions of Horn clauses can be found in polynomial time. Similarly for Krom clauses (clauses with at most two literals).*

Proof. In this case all clauses of the expressions $\alpha[S_i]$ and all $J'(\delta)$'s are Horn (respectively, Krom) (although the expression $P(\delta)$ may be still in general conjunctive normal form), and thus the inferences required for carrying out Cases 1, 2, and 3 can be done in linear time. It is clear that otherwise the algorithm implicit in the proof of Theorem 5 is polynomial. Naturally, this result covers also the disjunction-free case. □

Furthermore, the idea in the proof of Theorem 5 (and Lemma 6) were used in [12] to define the *tie-breaking semantics* of logic programs with negations, a generalization of the well-founded semantics and, in some provable sense, the ultimate semantics of negation in Datalog, see [12].

Another useful genre of problems related to default theories is the one that asks whether a property holds of all extensions—"cautious" reasoning. Interestingly, recent results [4] reveal that the graph-theoretic technique used in Theorem 5 may be less helpful in this regard. Our technique handles a specific kind of extension that are guaranteed to exist in even graphs. However, default theories with even graphs may also have other defaults, besides the "standard" ones—and telling whether they do is a hard problem, see [4] for a discussion of the disjunction-free case.

Defaults such as the prerequisite-free, disjunction-free ones in Example 4 are called *Krom defaults* (in analogy with Krom clauses, those that contain two literals). Krom defaults are perhaps the simplest family of defaults whose extension problem is not completely trivial. We can show the following:

Theorem 9. *It is* NP*-complete to tell whether a Krom default theory has an extension.*

Theorem 9 is quite surprising, as one would hope that the well-known graph-theoretic and other techniques used to solve 2-SAT would render Krom defaults tractable. It seems to suggest that even severe "local" restrictions on the syntax of the defaults fail to tame the complexity of the extension problem; only "global" properties, such as orderability and "evenness", appear to make a difference. It strengthens NP-completeness results for more general classes [3, 14]. Theorem 9 had apparently been known to researchers in the area, as it follows easily from the work of Stillman, and also from NP-completeness results in autoepistemic logic (many thanks to Stillman for pointing this out to us). Stillman's proof was essentially identical to the simple construction which we include below for completeness.

Proof (Sketch). Reduction from SATISFIABILITY. For each variable x we have the two defaults $\frac{:x}{x}$ and $\frac{:\neg x}{\neg x}$. For each clause c and each literal $\lambda \in c$ we add the default $\frac{:\lambda \& c}{c}$. Finally, for each clause c we add "copies" of the eight defaults in Example 4, with a different set of variables a through g for each clause. This completes the construction of the default theory (α_0 is again true).

Any extension of this default theory must contain either $\frac{:x}{x}$ or $\frac{:\neg x}{\neg x}$ for all x, and thus defines a truth assignment. If a clause c is not satisfied by this truth assignment, then default 0 of the copy that corresponds to that clause must be in the extension (recall Fig. 1(d)), thus contradicting the existence of a default (recall our discussion of Fig. 1(d)). □

Acknowledgment

We are grateful to Yannis Dimopoulos, Nicole Bidoit, Christine Froidevaux, and Bart Selman for many interesting comments on the ideas and results in this paper.

References

[1] N. Bidoit and C. Froidevaux, General logical databases and programs: default logic semantics and stratification, *Inf. Comput.* **91** (1) (1991) 15–54.
[2] A.K. Chandra and D. Harel, Horn clause queries and generalizations, *J. Logic Programm.* **2** (1) (1985) 1–15.
[3] Y. Dimopoulos and V. Magirou, A graph theoretic approach to default logic, *Inf. Comput.*, (to appear).
[4] Y. Dimopoulos, V. Magirou and C.H. Papadimitriou, Computing the kernels of a graph, Technical Report, Max-Planck Institute, Saarbrücken, Germany (1993).
[5] D.W. Etherington, Formalizing nonmonotonic reasoning systems, *Artif. Intell.* **31** (1987) 41–85.
[6] G. Gottlob, Complexity results in non-monotonic logics, CD–TR 91/24, T.U. Wien (1991).
[7] P. Kolaitis and C. Papadimitriou, Why not negation by fixpoint?, in: *Proceedings Seventh ACM Symposium on Principles of Database Systems* (1988); also: *J. Comput. Syst. Sci.* (1991).

[8] H. Kautz and B. Selman, Hard problems for simple default logics, *Artif. Intell.* **49** (1991) 243–279.
[9] K. Kunen, Signed data dependencies in logic programs, *J. Logic Programm.* 7 (1989) 231–245.
[10] C.H. Papadimitriou, *Computational Complexity* (Addison-Wesley, Reading, MA, 1993).
[11] C.H. Papadimitriou and M. Sideri, On finding extensions of default theories, in: *Proceedings Fourth International Conference on Database Theory*, Lecture Notes in Computer Science, (Springer, Berlin, 1992) 276-281.
[12] C.H. Papadimitriou and M. Yannakakis, Tie-breaking semantics and structural totality of logic programs with negation, in: *Proceedings 11th ACM Symposium on Principles of Database Systems*, San Diego, CA (1992); also: *J. Comput. Syst. Sci.* (to appear).
[13] R. Reiter, A logic for default reasoning, *Artif. Intell.* **13** (1980) 81–132.
[14] B. Selman, Tractable default reasoning, Ph.D. Thesis, University of Toronto, Toronto, Ont. (1990).
[15] J. Stillman, The complexity of propositional default logics, in: *Proceedings AAAI-92*, San Jose, CA (1992) 794–799.