INFORMATION AND COMPUTATION 112, 239–256 (1994)

A Graph-Theoretic Approach to Default Logic

YANNIS DIMOPOULOS* AND VANGELIS MAGIROU

Department of Informatics, Athens University of Economics and Business, 76 Patission Street, Athens 104 34, Greece

A network representation of propositional seminormal disjunction-free default theories is presented, leading to a graph-theoretic approach to their analysis. The problem of finding an extension is proved to be equivalent to that of determining a kernel for a corresponding graph, allowing stronger complexity results as well as new conditions for the existence of extensions.

1. INTRODUCTION

The ability of humans to draw conclusions that are retractable, along with the observation that standard logics are not sufficient to capture such patterns of common sense reasoning, has received much attention in AI research. This motivated an effort to formalize retractable reasoning which resulted in the development of nonmonotonic logics. A well-known such formalism introduced by Reiter is known as default logic (Reiter, 1980). A great amount of work has been done concerning both the theoretical properties of default logic as well as its representational power.

In this paper we investigate default logic using a graph-theoretic representation. The problem of determining whether a default theory has an extension is proved to be NP-complete even in the perhaps simplest possible case of propositional disjunction free seminormal default theories (denoted as DFP) without prerequisites in the default rules. This result generalizes a result by Kautz and Selman (1991), who show the NP-completeness of the problem in the case of single literal rules, but rules which in general allow prerequisites, as well as a result of Stillman (1990) concerning the NP-completeness of membership problems. Perhaps more important, the graph-theoretic approach may shed new light to the computational properties of the problem.

In Section 2, we give the definitions necessary for the development of our approach. In Section 3 we introduce our graph-oriented approach to

* Current address: Max-Planck-Institut für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany. E-mail: yannis@mpi-sb.mpg.de

0890-5401/94 $6.00

default logic. In Section 4 we focus on prerequisite-free default theories and prove some computational results. In Section 5 we generalize the results of the previous section to the case of default theories with prerequisites. In Section 6 we summarize our results and make a quick reference to an algorithm which computes the kernels of a digraph.

2. Definitions

A default theory consists of a finite set of default rules D and a set of propositions W, denoted as $\Delta = (D, W)$. The propositions in W are those that are to be believed beyond any doubt, while D is a set of rules leading to various, possibly inconsistent default beliefs. A default rule is an object of the form $a : b/c$, where a, b, and c are propositions, a is called the prerequisite, b the justification, and c the consequent. A default rule allows us to represent default assertions. For example the assertions "birds other than penguins can be assumed to fly" and "graduate students are adult, unless it is known otherwise" are expressed by the default rules "bird: fly $\wedge \neg$penguin/fly" and "graduate_student: adult/adult" respectively. Default rules provide a mechanism to "jump to conclusions" in the absence of information to the contrary. The key concept in Default Logic is that of an *extension* of a default theory, which is roughly what can be consistently believed given (D, W). The formal definition of an extension was given by (Reiter, 1980), but a more convenient description is the statement of the following theorem in the same reference.

Theorem 1. *A set of propositions E is an extension of a propositional default $\Delta = (D, W)$ iff it satisfies the equation $E = \bigcup_{i=0}^{\infty} E_i$, where $E_0 = W$ and $E_{i+1} = \mathrm{Th}(E_i) \cup \{w \mid a : b_1 \wedge \cdots \wedge b_n/w \in D,\ a \in E_i$ and $\neg b_j \notin E\}$.*

In our analysis we often assume deductive closure implicitly when talking about extensions. Unfortunately Theorem 1 does not provide a method of generating extensions, despite its seemingly constructive nature. If one is lucky to guess at an extension E, Theorem 1 can be used to verify the claim. However, if we do not know E, then the sets $E_0, E_1, \ldots$ cannot be generated, as their construction depends on E itself.

Example 1. Let $\Delta = (D, W)$ be a default theory, where $D = \{d_1 =: A \wedge \neg C/A,\ d_2 =: B \wedge \neg A/B,\ d_3 =: C \wedge \neg B/C\}$. Assume first that the given truths, W, consist only of A, i.e., $W = \{A\}$. Guessing an extension $E = \{A, C\}$ we can compute the sets $E_0, E_1, \ldots$ of Theorem 1 which are $E_0 = \{A\}$, $E_1 = \{A, C\} = E_2 = \cdots = E_n = \cdots$. Hence $E = \bigcup_{i=0}^{\infty} E_i = \{A, C\}$, which equals the initial E, verifying that $\{A, C\}$ is indeed an extension. On the other hand, if we are given that $W = \{B\}$, then it can be shown that

$E_0 = \{B\}$, $E_1 = \{A, B\} = E_2 = \cdots = E_n = \cdots$ and thus again $E = \{A, B\}$ is an extension. However, if $W = \emptyset$ it can be shown (using the results to be developed later) that no extension exists.

Given an extension E for a default theory Δ the set of generating defaults for E with respect to Δ is defined to be $\mathrm{GD}(E, \Delta) = \{a : b_1 \wedge \cdots b_n/w \in D \mid a \in E \text{ and } \neg b_i \notin E\}$.

A default theory $\Delta = (D, W)$, in which every element of W, the prerequisite, the justification, and the consequent of any default is a conjunction of literals, is called a disjunction-free default theory. A default theory in which every default rule is of the form a: $b \wedge c/b$ is called seminormal. In the rest of this paper we consider only Propositional Seminormal Disjunction Free Default Theories, abbreviated as DFP's.

For a default rule $\delta = a_1 \wedge \cdots a_k : b_1 \wedge \cdots b_l \wedge c_1 \cdots \wedge c_m/b_1 \wedge \cdots b_l$ we use the notation of Kautz and Selman (1991):

$$\mathrm{Prer}(\delta) = \{a_1 \cdots a_k\}$$
$$\mathrm{Cons}(\delta) = \{b_1 \cdots b_l\}$$
$$\mathrm{Just}'(\delta) = \{c_1 \cdots c_m\}$$
$$\mathrm{Just}(\delta) = \mathrm{Just}'(\delta) \bigcup \mathrm{Cons}(\delta).$$

A further useful characterization of extensions is given in Theorem 2 of (Reiter, 1980).

THEOREM 2. *Suppose E is an extension of a propositional default theory $\Delta = (D, W)$. Then $E = \mathrm{Th}(W \cup \mathrm{Cons}(GD(E, \Delta))$.*

Note that the converse of Theorem 2 is not necessarily true. That is, a set E satisfying the relation $E = \mathrm{Th}(W \cup \mathrm{Cons}(GD(E, \Delta))$ is not necessarily an extension.

3. A NETWORK REPRESENTATION OF DEFAULT THEORIES

Given a DFP theory $\Delta = (D, W)$ Kautz and Selman (and Etherington, 1988, before them) represent its structure diagrammatically in terms of implication diagrams involving the literals of the theory. We choose to represent the structure of the theory by a graph where nodes stand for rules rather than literals and use graph theoretic constructs to detect possible contradictions in Δ. We introduce a network $N = (V, E)$ consisting of a multigraph $G = (V, E)$ and a function $f: E \to \{+1, -1\}$. An edge $e \in E$ is called positive if $f(e) = 1$ and negative if $f(e) = -1$. A positive edge marks a possible derivation, while a negative one denotes inconcistencies. The vertex set is $V = \{n_0, n_1, ..., n_M\}$, where n_0 corresponds to W and $n_1, ..., n_M$

to the default rules $d_1, \ldots, d_M$. The edges are constructed as follows. For $i, j \neq 0$, introduce a negative edge from n_i to n_j if there is a literal in the consequence of d_i that negates some justification of d_j. Introduce a positive edge from d_i to d_j for every literal in the prerequisites of d_j that appears in the consequence of d_i. For n_0, introduce a negative edge to n_i if some literal in W negates some justification of d_i, and introduce a positive edge to n_i for every literal in W appearing in the prerequisites of the d_i. In the case of a prerequisite-free default rule introduce a positive edge from n_0 to the node corresponding to the rule. For a rule d_i such that some of the prerequisites do not appear anywhere in W or the consequents of another rule put a negative edge from n_0 to n_i. These last negative edges are not strictly necessary as our extension finding method will omit from consideration any part of the graph not connected to W via positive edges.

Some further structure is required in order for the network to represent accurately the default theory Δ. Consider a node n_i corresponding to d_i, and consider its positive incoming edges. If $E_{i,m}$ are the edges corresponding to the literal a_m it is obvious that just one of the edges in $E_{i,m}$ is needed to transfer a_m to n_i. Hence we refer to $E_{i,m}$ as an OR-structure. Furthermore note that we might have parallel positive edges between two nodes corresponding to different literals. Without loss of generality we can merge in a single edge all edges that are in such an OR-structure.

The edges that form an OR-structure are depicted in the network by joining them by an OR sign, as shown in Fig. 1. Any positive edge not covered by an OR sign is a trivial OR-structure consisting a single edge. We use the term head of an OR-structure to denote the node to which the corresponding edges are incoming, and the term tail to denote the set of nodes from which the edges forming the structure are outgoing. The positive edges are denoted by the arrows and the negative edges by the cross-hatched arrows.

EXAMPLE 2. Let $\Delta = (D, W)$ be a default theory with $W = \{a, b\}$ and $D = \{d_1 = a : b \wedge c/b,\ d_2 = a \wedge b : \neg c \wedge d/\neg c,\ d_3 = a \wedge d : e \wedge f/e,\ d_4 = e : g/g\}$. The corresponding network is shown in Fig. 1.

A *valuation* of a network is an assignment $u(n_i)$ of values from $\{-1, 1\}$ to the nodes which satisfies the following conditions ($u(n_i)$ is then denoted u_i): First, n_0 is assigned the value $u(n_0) = u_0 = 1$. For $i \neq 0$ a node n_i is assigned the value $u_i = 1$ if (a) for each OR-structure that has n_i as its head there is at least one of the nodes of the OR-structure's tail with value 1 and (b) there is no negative edge $e = (n_j, n_i)$ with $u_j = 1$. Finally, a node n_i is assigned the value $u_i = -1$ if the above conditions are not satisfied.

The existence of a valuation is evidence for the existence of an extension. However, one must make sure that the sequential derivation indicated by

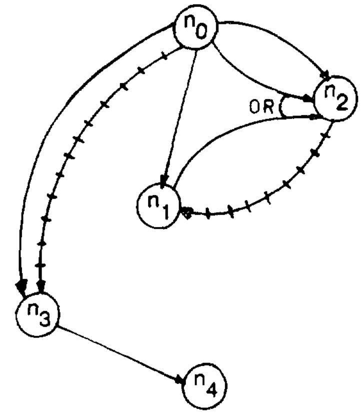


FIGURE 1

Reiter's Theorem 1 is possible for the candidate valuation. A case where this is impossible is shown in Example 3 below.

We call a valuation u *sequential* if there is a nonnegative integer function $l(n_i) \equiv l_i$ on the set of nodes which have positive value u_i such that

(a) $l_0 = 0$ and

(b) for $i \neq 0$, l_i must satisfy

$$l_i = \max_{\text{OR with head } n_i} \left(\min_{j \in \text{tail of OR}} l_j \right) + 1$$

Intuitively, the number l_i is the earliest step in which rule d_i can be applied, since its application must follow the application of any rules required for its prerequisites to be derived. We call such a function l a *sequencing* function. Such functions do not exist for arbitrary valuations, as shown in the next example.

EXAMPLE 3. Let $\Delta = (D, W)$, $W = \{a_1, a_2, a_3, a_4\}$, $D = \{d_1 = a_1 \wedge a_5 : a_6/a_6,\ d_2 = a_2 \wedge a_6 : a_7 \wedge \neg a_8/a_7 \wedge \neg a_8,\ d_3 = a_3 \wedge a_7 : a_5/a_5,\ d_4 = a_4 : a_8 \wedge a_5/a_5\}$. The corresponding network is shown in Fig. 2. Note that $u_0 = u_1 = u_2 = u_3 = 1$, $u_4 = -1$ is a valuation but no sequencing function exists because of cycle n_1, n_2, n_3 which consists of positive edges does not allow any node to get $l_i = 1$.

One can show that indeed the values of the sequencing function l can be interpreted as sequential steps. Namely we have the following lemma:

LEMMA 1. *The range of values of function l is an initial segment of the positive integers.*

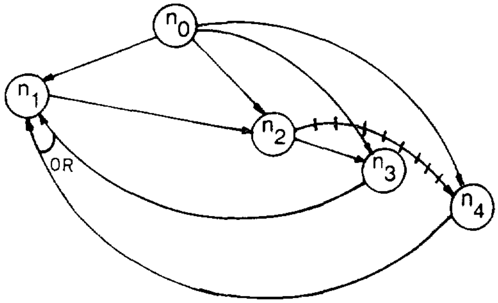


FIGURE 2

The following lemma is used further on. It is stated without proof, as it is a slight modification of that of Kautz and Selman (1991):

LEMMA 2. *Let $\Delta = (D, W)$ be a disjunction free finite propositional default theory. Then E is an extension of $\Delta = (D, W)$ iff there exists a sequence of rules $\delta = \{d_{i_1}, d_{i_2}, ..., d_{i_n}\}$ from D, which without loss of generality can be taken as $\delta = \{d_1, d_2, ..., d_n\}$, and a series of sets $E_0, E_1, ..., E_n$ such that*

(i) $E_0 = W$

(ii) $E_i = E_{i-1} \cup \text{Cons}(d_i)$

(iii) $\text{Prer}(d_i) \subseteq E_{i-1}$

(iv) *There is no $c \in \text{Just}(d_i)$ such that $\neg c \in E_n$*

(v) *There is no $d \in D - \delta$ with $\text{Prer}(d) \subseteq E_n$ such that for all literals $c \in \text{Just}(d)$ it holds that $\neg c \notin E_n$*

(vi) *E is the deductive closure of E_n.*

Furthermore, the default sequence δ can be partitioned into subsets $\{\delta_1, \delta_2, ..., \delta_k\}$ such that δ_1 includes all the defaults that have their prerequisites in E_0, δ_2 includes all the defaults with prerequisites in E_0 or in the consequences of the rules in δ_1, and so on.

The relation between extensions of some DFP theory and sequencing valuations on the corresponding network, is given in the following result:

THEOREM 3. *If a DFP default theory $\Delta = (D, W)$ has an extension E, and δ is the default sequence of Lemma 2, then the corresponding network has a sequencing valuation such that* (a) $u_0 = 1$, (b) $u_i = 1$ *if $d_i \in \delta$, and* (c) $u_i = -1$ *otherwise. Conversely, a network for which a sequencing valuation exists has an extension defined by the deductive closure of W and the consequents of the positive nodes in the valuation.*

Proof. Let $D=\{d_1, d_2, \ldots d_m\}$, and let $\delta=\{d_{i_1}, d_{i_2}, \ldots d_{i_n}\}$ be the sequence of Lemma 2. Without loss of generality we can take $\delta=\{d_1, d_2, \ldots, d_n\}$. We claim that $u_i=1$ for $i \leqslant n$ and $u_i=-1$ otherwise is a sequential valuation. To see that u is indeed a valuation note first that u_0 is 1 by construction. Furthermore, a positive node corresponds to a default which by Lemma 2 has all its prerequisites belonging to the consequences of some antecedent positive default, and furthermore none of its justifications is negated by any default in δ. Hence the valuation rules are satisfied for positive nodes. Now a negative node corresponds to some default rule for which either its prerequisites are not in the extension or some of its justifications are negated in the extension. This is again a restatement of the valuation conditions. It only remains to show the existence of a sequencing function l_i for the positive nodes. According to Lemma 2 the defaults corresponding to positive nodes can be partitioned into subsets $\delta_1, \delta_2, \ldots, \delta_k$. Let l_i be the function that assigns to the positive node $n_i \in \delta_m$ the index of the set m. It is clear that this function is a sequencing one as it corresponds to the step at which the particular rule can be applied.

To show the converse, form a sequence of defaults consistent with the sequencing function and note that they satisfy the conditions of Lemma 2. ∎

The above discussion is clarified in the following example:

EXAMPLE 4. Let $\Delta=(D, W)$ be the default theory with $W=\{a_1, a_2\}$ and $D=\{d_1=a_1 : b_1 \wedge c_1/b_1,\ d_2=a_2 : b_2 \wedge c_2/b_2,\ d_3=b_1 : b_3 \wedge c_3/b_3,\ d_4=b_1 : b_4 \wedge c_1/b_4,\ d_5=b_2 : \neg c_1 \wedge c_5/\neg c_1,\ d_6=b_2 : b_5 \wedge c_6/b_5\}$. The corresponding network G is shown in Fig. 3. The assignment $u_0, u_2, u_5, u_6=1$ and $u_1, u_3, u_4=-1$ gives G a sequencing valuation with $l_0=0$, $l_2=1$, $l_5=l_6=2$, and $W \cup \mathrm{Cons}(d_2, d_5, d_6)=\{a_1, a_2, b_2, \neg c_1, b_5\}$ is an extension of Δ.

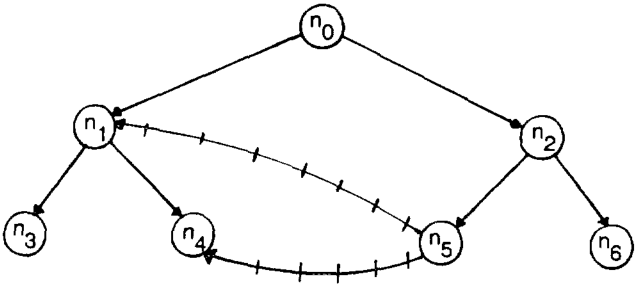


FIGURE 3

4. Default Theories without Prerequisites

Consider a particular default theory Δ such that $\text{Prer}(d_i) = \varnothing$ for each $d_i \in D$. It is clear that in the corresponding network each node n_i for $i > 0$ has an incoming positive edge from n_0 and no other positive edges are present. In addition, there exist no nontrivial OR-structures. Thus nodes are connected essentially only through negative edges. A negative connection between n_i and n_j occurs if for a literal a, $a \in \text{Just}(d_i)$ and $\neg a \in \text{Cons}(d_j)$. For this network, an assignment u_i is a sequencing valuation provided (a) no two positive nodes n_i and n_j are connected by a negative edge and (b) each negative node is connected to a positive node by some negative edge. It is clear that such a u is a valuation, and the sequencing function is simply 0 for n_0 and 1 for the positive nodes. Intuitively this means that any set of mutually consistent defaults is a sequential valuation and all conclusions can be derived in one step from the initial assumptions in W.

We can easily establish a connection between extensions and the graph-theoretic concept of a kernel, a concept that is useful when studying two person zero sum games defined on graphs. A *kernel* of a directed graph $G = (V, E)$ (Berge, 1973) is a node set $K \subseteq V$ such that no two nodes in K are joined by an edge in E and such that for every node $n' \in V - K$ there is a node $n \in K$ for which $(n, n') \in E$. (Berge's definition of a kernel stipulates that (n', n) should be in E, but this is not an essential difference, as one can consider graphs with edges of inverted directions).

The problem of finding a sequencing valuation and hence an extension in this particularly simple case of DFP default theories is equivalent to finding a kernel for the graph which consists of the nodes n_i and the negative edges of the network corresponding to the DFP. In fact, consider the network $N = (V, E)$ corresponding to a DFP without prerequisites and form a graph $\bar{G} = (V, E')$, where E' is an edge set containing only the negative edges of E. If there exists a kernel K for $\bar{G}$ then we can get a sequential valuation by assigning the value 1 to each node in K and -1 to each node in $N - K$, an assignment that satisfies the two conditions for a valuation stated in the beginning of this section, as no kernel nodes are adjacent and any nonkernel node is dominated, i.e., negated by a node in the kernel. Conversely, an extension of Δ provides a kernel of $\bar{G}$ by identifying the nodes of the valid defaults with kernel nodes. We can now state the following:

Proposition 1. *The problem of determining whether a prerequisite free DFP default theory has an extension is NP-complete in the number of default rules.*

Proof. The proof is by reduction from the problem of finding a kernel of a directed graph, which is known to be NP-complete (Garey and

Johnson, 1979, attributed to Chvatal, 1973). Let $G=(N, E)$ be a directed graph. We exhibit a prerequisite free DFP default theory with a polynomial number of default rules which has an extension iff G has a kernel. Thus the existence of a polynomial extension finding algorithm would also resolve the kernel problem.

For each node $n_i \in N$ construct a default rule d_i for which $\text{Cons}(d_i)=\{a_i\}$ (let a_i denote a literal) and $\text{Just}'(d_i)=\{\neg a_j \mid (n_j, n_i) \in E\}$ and hence $\text{Just}(d_i)=\{a_i\} \cup \text{Just}'(d_i)$. Consider the prerequisite free DFP $\Delta=(\varnothing, \{d_1, d_2 \cdots d_m\})$, with m being the number of nodes in G. It is obvious that the graph $\bar{G}$ obtained by deleting the positive edges of the network obtained from Δ is exactly the original graph G. Hence a solution of the extension problem for Δ would solve the kernel problem for G. Furthermore, the number of defaults of Δ is equal to the number of nodes of G. ▮

The above proposition extends the complexity results of Kautz and Selman (1991), as well as those of Stillman (1990), who also shows that the membership problem (i.e., does there exist an extension of Δ containing a given literal q?) is NP-complete even for prerequisite free default logics. It is worth noticing that Marek and Truszczynski (1991) independently proved that the problem of the existence of an expansion for a simple class of Autoepistemic Theories is NP-complete by constructing a directed graph based on the literals of the theory and using a reduction from the kernel existence problem.

EXAMPLE 5. The graph G of Fig. 4 corresponds to a default theory $\Delta=(D, \varnothing)$, with $D=\{d_1 =: a_1/a_1,\ d_2 =: a_2 \wedge \neg a_1/a_2,\ d_3 =: a_3 \wedge \neg a_1 \wedge \neg a_4/a_3,\ d_4 =: a_4 \wedge \neg a_2 \wedge \neg a_6/a_4,\ d_5 =: a_5 \wedge \neg a_3/a_5,\ d_6 =: a_6 \wedge \neg a_5/a_6\}$. Note that $E=\text{Th}(W \cup \text{Cons}\{d_1, d_4, d_5\})$ is an extension for Δ and $K=\{n_1, n_4, n_5)$ is a kernel for G.

The relation between kernels and extensions can be of further theoretical use. For instance, according to the well known Richardson theorem on the

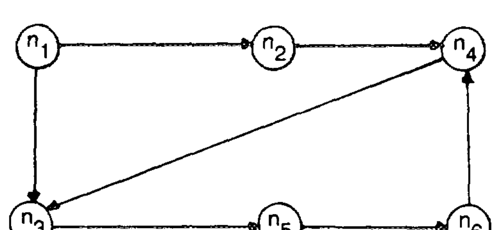


FIGURE 4

existence of kernels, a prerequisite free DFP has an extension provided there are no cycles of odd length in the corresponding graph (see Berge, 1973). Hence an immediate consequence of this theorem is the following proposition:

PROPOSITION 2. *Every DFP theory without prerequisites, for which the corresponding graph has no odd cycles, has an extension.*

The nonexistence of cycles is a special case of Richardson's theorem, and thus is the analog of several known results about the existence of extensions when some sort of order and thus no cycles are present in the defaults. Thus Proposition 2 extends the results of Etherington (1988) on ordered default theories which in our case correspond to acyclic graphs. An interesting question is whether the above "no odd cycle" property translates into a new useful class of logic programs, possibly generalizing the notion of stratification; also whether stronger graph theoretic techniques on kernels determination (e.g., Grundy functions) have any meaningful implication about default logic. The proof of Richardson's theorem actually gives a polynomial time algorithm to construct a kernel. It starts by determining an arbitrary basis of the graph (a basis is a minimal independent set of nodes from which all nodes are reachable). It then deletes all nodes adjacent to those in the basis and essentially repeats the basis selection process. Determining a basis can be done in polynomial time as a byproduct of the determination of strongly connected components, and hence the whole kernel finding process is polynomial. Then we get the following two propositions:

PROPOSITION 3. *If the corresponding graph of a DFP theory Δ without prerequisites is strongly connected and contains no odd cycles, then for every rule $d_i \in D$ there is an extension E such that $d_i \in \mathrm{GD}(E, \Delta)$ (i.e., d_i is applicable) and another extension E' such that $d_i \notin \mathrm{GD}(E', \Delta)$.*

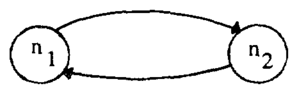
Proof. In a strongly connected graph any node d is a basis and hence can be used as the first step in Richardson's procedure to construct E. On the other hand, selecting a node d' adjacent to d as a basis gives rise to an extension E' that excludes d. ∎

PROPOSITION 4. *Given a DFP theory without prerequisites whose graph has no odd cycles, there exists a polynomial algorithm that identifies some extension and determines if the extension is unique.*

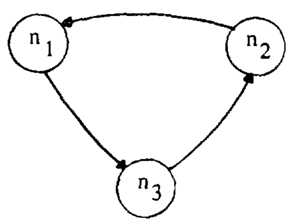
Proof. The application of Richardson's procedure determines an extension. The extension is unique iff the relevant connected components are singletons. ∎

We illustrate these results in the following example:

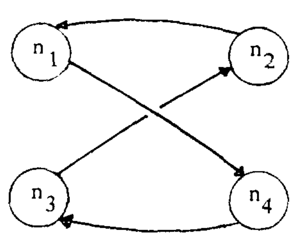
EXAMPLE 6. Let $\Delta = (D, \varnothing)$, $D = \{:a \wedge \neg b/a,\ :b \wedge \neg a/b\}$, which has two extensions. Note that the corresponding graph (G1 of Fig. 5) has an even cycle and is strongly connected. If $D = \{:a \wedge \neg b/a,\ :b \wedge \neg c/b,\ :c \wedge \neg a/c\}$ the graph (G2 of Fig. 5) has an odd cycle and no extension exists. To break this cycle it suffices to put in W one of the propositions a, b, or c or one of their negations and produce an extension. If $D = \{:a \wedge \neg b/a,\ :b \wedge \neg c/b,\ :c \wedge \neg d/c,\ :d \wedge \neg a/d\}$, the corresponding graph (G3 of Fig. 5) has an even cycle, is strongly connected and possesses two extension. Observe that for every $d_i \in D$ and for every $p \in \text{Cons}(d_i)$ there is an extension E that contains p and an extension E' that does not contain p.



G1



G2



G3

FIGURE 5

643/112/2-7

On the negative side, the following lemma shows that any characterization of the kernels is hard even for graphs without odd cycles.

LEMMA 3. *Given a directed graph without odd cycles, the problem of determining whether there is a kernel including (or excluding) a given node is NP-complete.*

Proof. Consider a 3-SAT formula $F = \prod_{i=1}^{M} c^i$ in clauses $c^i = y_1^i + y_2^i + y_3^i$ where y_j^i is a literal in the variables $x_1, ..., x_N$. We construct a graph without odd cycles as follows (see Fig. 6). For each clause c^i construct a 7-node subgraph in nodes $u_1^i, ..., u_7^i$ as in Fig. 6. Furthermore, construct $2N$ nodes w_j, $\bar{w}_j$, corresponding to the variables x_j and their

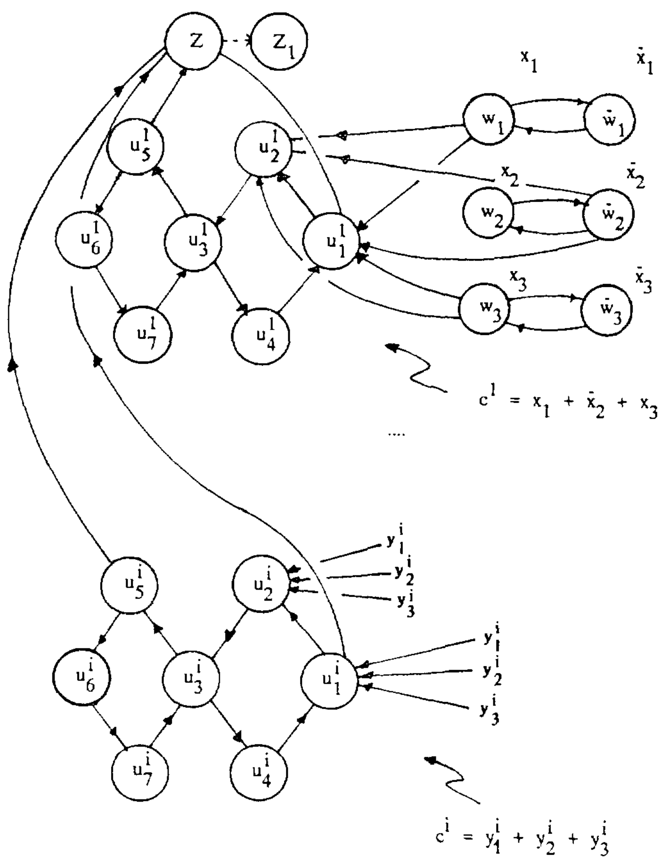


FIGURE 6

negations, with edges $(w_j, \bar{w}_j)$ and $(\bar{w}_j, w_j)$. Introduce also an extra node Z. For each clause c^i add edges (w_j, u_1^i) and (w_j, u_2^i) or $(\bar{w}_j, u_1^i)$ and $(\bar{w}_j, u_2^i)$, depending on whether x_j or $\bar{x}_j$, appears in c^i. Finally introduce edges (u_1^i, Z) and (u_5^i, Z) for all i. The graph constructed this way has only even cycles. Now if F is satisfiable, a kernel exists which includes Z and consists of Z, the w_j, $\bar{w}_j$'s that correspond to the x_j's that are true, and u_3^i, u_6^i for all i. Conversely, a kernel that includes Z induces a truth value assignment that satisfies F. Hence, determining a kernel that includes a specific node is NP-complete even in the case of even-cycled graphs. For the problem of a kernel excluding a given node, consider the same graph with an additional node Z_1 and the edge (Z, Z_1). If Z_1 is not allowed to be in the kernel, Z must be included, and hence the previous argument remains valid. ∎

The previous lemma in conjunction with the construction in the proof of Proposition 1 gives us the following extension of Stillman's (1990) results:

PROPOSITION 5. *The membership problem is NP-complete even in the case of DFPs without prerequisites and without odd cycles.*

5. DEFAULT THEORIES WITH PREREQUISITES

Since the problem of finding the extensions of DFP propositional default theories without prerequisites is equivalent to that of finding a kernel in the corresponding graph, we investigate next the possibility of generalizing the approach to arbitrary DFP default theories with prerequisites. This task is complicated by the fact that nontrivial positive edges have a different meaning than negative edges in the network corresponding to a default theory. Positive edges show a dependence of a default rule from a previous one, while negative edges show incompatibility between rules. Thus the concept of a kernel is not quite appropriate in the presence of positive edges. One can, however, remove positive edges from a network by adding further nodes corresponding to negation of literals plus further negative edges. This is essentially an application of De Morgan's laws to the network.

The following procedure shows how this can be accomplished. For each OR-structure of any default rule node n_i we make the transformation shown in Fig. 7. It amounts to removing all positive edges of the OR-structure, introducing an extra node that corresponds to the literal a that caused the dependence among the rules and adding negative edges (a) from a to n_i and (b) from the predecessor nodes n_j of the OR-structure to a. After this process has been done for all positive edges, one must coalesce all extra nodes corresponding to the same literal a. Negative edges remain unchanged.

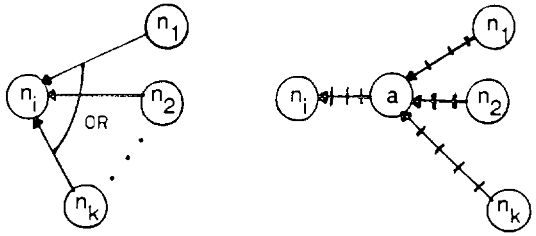


FIGURE 7

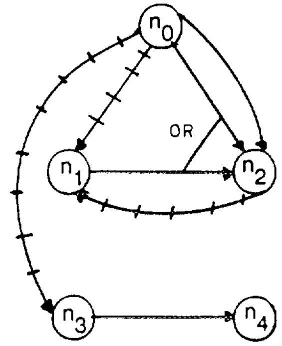


FIGURE 8

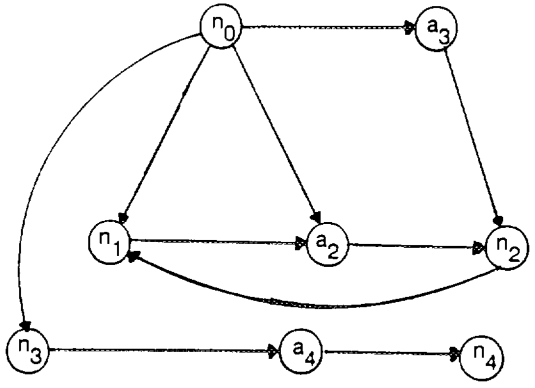


FIGURE 9

EXAMPLE 7. Consider the network N of Fig. 8. The corresponding graph with only negative edges is shown in Fig. 9.

The problem with the above construction is that an arbitrary kernel need not correspond to an extension although an extension will definitely induce a kernel. This is caused by the fact that the set of default rules of an extension are deduced in a sequential fashion in addition to being compatible. A kernel ensures compatibility but not sequential derivation, as noted in the section on valuations, where the notion of sequencing was essential. The graph-theoretic counterpart of the sequencing requirement can be stated as follows.

Let a graph $G=(V \cup V', E)$, where V corresponds to the original nodes and V' to the additional literal nodes. The edge set E consists of negative edges only. A kernel K of G is called *layered* iff there is a sequence of subsets $K_0, K_1, ..., K_m$ of $K \cap V$ with $K_m = K \cap V$ and a sequence of subsets $L_0, L_1, ..., L_m$ of V' defined as follows:

(i) $K_0 = \{n_0\}$.

(ii) For $i>0$, $K_i = K_{i-1} \cup \{n_j \mid n_j \in V - K_{i-1}, \text{adj}^-(n_j) \cap V' \subseteq L_{i-1}\}$.

(iii) For $i \geqslant 0$, $L_i = \text{adj}^+(K_i) \cap V'$,

where $\text{adj}^+(A) = \{n_k \mid n_i \in A, (n_i, n_k) \in E\}$ and $\text{adj}^-(A) = \{n_k \mid n_i \in A, (n_k, n_i) \in E\}$.

The above conditions state essentially that the default nodes that belong to a layered kernel must be obtained sequentially from the previous ones through the validity of the corresponding literals in the intervening sets L_i. Note that if a literal node α belongs to the kernel, this implies that α is false. The equivalence between layered kernels, sequential valuations, and hence extensions is established in the next two theorems.

THEOREM 4. *If a network $N=(V, E')$ has a sequential valuation with $A = \{n_i \in V \mid u_i = 1\}$ being the set of positive nodes, then the corresponding graph $G=(V \cup V', E)$ has a layered kernel $K = A \cup B$, where $B = \{a_i \mid$ there exists no edge (n_i, a_i) with $n_i \in A$, $a_i \in V'\}$.*

Proof. Let l be the sequencing function defined on the positive nodes. We write l_i for $l(n_i)$ for positive nodes n_i. Let the set K_j consist of all positive nodes whose sequencing is less or equal to j, i.e., $K_j = \{n_i \mid l_i \leqslant j\}$ with the corresponding L_j, defined as in condition (iii) above. We claim that K_m, the last K_i, along with the literal nodes L not in any L_j, forms a layered kernel. Indeed the K_j satisfy conditions (i)–(iii) above and thus it remains to show that $K_m \cup L$ is a kernel. Indeed any default node n_i not included in the kernel must have an incoming negative edge originating from a positive node or from a false literal. Furthermore, the literals in L

are not dominated by any positive node and hence are properly included in the kernel. ▮

THEOREM 5. *If a graph $G = (V \cup V', E)$ of a network $N = (V, E')$, has a layered kernel K then N has a sequencing valuation.*

Proof. We claim that a sequencing valuation is defined by $u_i = 1$ if $n_i \in B$ and $u_i = -1$ otherwise, where $B = K \cap V$, i.e., B are the nodes of the original kernel corresponding to default rules. It is obvious that u forms a valuation by the definition of the kernel. To exhibit a sequencing function let a positive node n_i belong to $K_j - K_{j-1}$, the j being unique. The sequencing function l is defined as $l(n_i) = j$. It is straightforward to check that this function is a sequencing one. ▮

It is easy to see that for theories with prerequisites whose graphs have no cycles that involve only positive edges, the complications requiring the concept of a layered kernel do not arise. Hence the results obtained for prerequisite-free theories remain valid for theories with prerequisites that lack positive cycles.

4. CONCLUSIONS

In this paper we gave a graph representation of disjunction free propositional default theories, which are interesting because of their relation to inheritance systems. Our approach reveals that a source of the computational burden is the graph-theoretic way in which the default rules interact. This interaction is depicted in the graph through edges between the default rules. What we have in a default theory is a set of contradictory default rules, and our problem is to find a collection of rules that are not mutually exclusive but that can exclude the application of any other default rule in the theory.

Our graph-theoretic approach is related to some already known theoretical and computational results. The presence of cycles in a graph that depicts the interaction between defaults is the culprit for the hardness of most of default logic's computational problems. To be more precise, if a directed graph is acyclic or if it has only cycles with an even number of nodes, it is guaranteed to have a kernel which is computable in polynomial time. On the other hand, if the graph contains odd cycles the problem of determining a kernel is intractable. In this direction an algorithm has been developed elsewhere (Dimopoulos *et al.*, 1993) to find all kernels of a directed graph. The algorithm starts with the heuristic determination of a feedback node set $\{n_1, ..., n_k\}$, i.e., a node set that includes a node from every cycle. For a given kernel, a node n_i either belongs or does not belong to it. However, given a putative assignment of the u_i's to a kernel, the remaining nodes of the kernel can be uniquely determined, since the graph resulting from the deletion of the u_i's is acyclic. If the extra kernel nodes thus obtained are consistent with the putative assignement, a kernel exists; otherwise the assignment does not produce a kernel. The above idea can be implemented in a branch-bound setup and will require at most 2^k steps, k being the cardinality of the feedback node set. Of course the best results are obtained if the feedback set is minimal, but this is another NP-complete problem. In its present form the algorithm is not even Polynomial Total Time in the sense of Johnson *et al.* (1988), i.e., $O(p(n)C)$, where p is a polynomial and C the number of kernels. Thus an interesting problem is to develop such a total polynomial time algorithm and perhaps an incremental polynomial time. However, our computational experience shows that the algorithm is very efficient for sparse graphs. For example, for a graph with 100 nodes and density 8%, the average time is about 7 seconds on a VAX 8810 in Pascal running under VMS(V.5-4). This time increases with the density until a density of about 20% is reached where the average running time for a graph with 100 nodes reaches 69 seconds and then decreases for higher densities.

Normal propositional disjunction free default theories are known to have extensions. If we consider the case of such theories without prerequisites it is obvious that the corresponding graph is an undirected one. For undirected graphs, kernels coincide with maximal independent node sets. These can be generated efficiently in lexicographic order with only polynomial delay between the generation of two successive maximal independent sets, using the algorithm in Johnson *et al.* (1988). Of course it can take exponential time to list all maximal independent sets.

Several directions exist for further research on the relation between default logic and graph theoretic concepts. Among them, the development of graph motivated algorithms for default problems (skeptical reasoning, membership problem) seems particularly promising.

Acknowledgments

We thank an anonymous referee for numerous incisive comments that not only improved the presentation of the paper but also led us to examine further aspects of the problem which are stated as Proposition 5 and sharpened the characterization of the kernel algorithms.

RECEIVED July 18, 1991; FINAL MANUSCRIPT RECEIVED April 10, 1992

References

Berge, C. (1973), "Graphs and Hypergraphs," North-Holland, Amsterdam.

Chvatal, V. (1973), "On the Computational Complexity of Finding a Kernel," Report CRM-300, Centre de Recherches Mathématiques, Université de Montréal.

Dimopoulos, Y., Magirou, V., and Papadimitriou, C. (1993), "On Kernels, Defaults and Even Graphs," Technical Report MPI-I-93-264, Max-Planck-Institut für Informatik.

Etherington, D. (1988), "Reasoning with Incomplete Information," Pitman, New York.

Etherington, D. (1987), Formalizing nonmonotonic reasoning systems, *Artificial Intelligence* **31**.

Johnson, D., Yannakakis, M., and Papadimitriou, C. (1988), On generating all maximal independent sets, *Inform. Process. Lett.* **27**.

Kautz, H. A., and Selman, B. (1991), Hard problems for simple default logics, *Artificial Intelligence* **49**.

Marek, W., and M. Truszczynski, (1991), Autoepistemic Logic, *J. Assoc. Comput. Mach.* **38**, No. 3.

Reiter, R. (1980), A logic for default reasoning, *Artificial Intelligence* **13**.

Selman, B., and Kautz, H. A. (1990), The complexity of model-preference default theories, *Artificial Intelligence* **45**.

Stillman, J. (1990), It's not my default: The complexity of membership problems in restricted propositional default logics, *in* "AAAI-90."