The Tractability of Path-Based Inheritance

Bart Selman and Hector J. Levesque*
Dept. of Computer Science
University of Toronto
Toronto, Canada M5S 1A4

Abstract

Touretzky (1984) proposed a formalism for nonmonotonic multiple inheritance reasoning which is sound in the presence of ambiguities and redundant links. We show that Touretzky's inheritance notion is NP-hard, and thus, provided P#NP, computationally intractable. This result holds even when one only considers unambiguous, totally acyclic inheritance networks. A direct consequence of this result is that the conditioning strategy proposed by Touretzky to allow for fast parallel inference is also intractable. Therefore, it follows that nonmonotonic multiple inheritance hierarchies, although compact representations, may not allow for efficient retrieval of information as has been suggested in attempts to use such hierarchies, *e.g.,* in NETL (Fahlman 1979). We also analyze the influence of various design choices made by Touretzky. We show that all versions of downward (coupled) inheritance, *i.e.,* on-path or off-path preemption and skeptical or credulous reasoning, are intractable. However, tractability can be achieved when using upward (decoupled) inheritance.

1 Introduction

Since the early semantic networks of Quillian (1968), inheritance hierarchies have been used to provide a compact representation and efficient reasoning mechanism for certain kinds of taxonomic information. One problem that has plagued these systems is that of *exceptions* (or cancellation) in non-tree hierarchies. Early attempts to systematically deal with this issue, such as that of NETL (Fahlman 1979), were later shown to be unsound in the presence of redundant links and ambiguities (Reiter and Criscuolo 1983; Touretzky 1984). The first comprehensive definition that appeared to solve these problems was that of Touretzky (1984). Since then, other equally sound schemes have been proposed (Sandewall 1986; Ilorty *et ai* 1987). But despite a decade of study, with increasingly subtle examples and counter-examples being considered, consensus has yet to emerge regarding the proper treatment of multiple inheritance with cancellations.

From a knowledge representation standpoint, part of the problem is that there are tremendous subtleties in reasoning with propositions that admit exceptions, such as "Birds fly" (Brachman 1985). Not surprisingly, the research on inheritance systems mentioned above has not attempted to settle the larger logical and semantical issues associated with defeasible reasoning. Rather, the argument has been that inheritance systems need to conform to a set of special intuitions involving paths through hierarchies. We therefore call such systems *path-based inheritance systems* and contrast them with the more general *nonmonotonic reasoning systems* (Reiter 1987). The latter approach attempts to establish a logical account of defeasible reasoning (using, for example, autoepistemic, circumscriptive, conditional, or default logic), and somehow absorb hierarchies and inheritance as a special case. While the nonmonotonic systems tend to be more principled and semantically motivated on the whole, they have yet to be applied successfully to problems as intricate as those considered by the path-based approaches.

But if there are indeed irreducible intuitions about inheritance and paths through hierarchies, these intuitions are sometimes in conflict and can give rise to different inheritance systems (Touretzky *et ai* 1987). How then to choose among the competing systems, especially since there is no independent semantic characterization that adequately covers the phenomena in question? While we do not claim to have an answer to this question, we do propose here a *criterion* that should be taken into account when comparing systems. What we will show is that there can be a significant difference in the *compuiational tractability* of inheritance depending on fine points of the definition used. In other words, two accounts of inheritance that cover by and large the same territory, differing only in certain complex cases, may nonetheless be quite different in their overall computational demands.

The main technical result of this paper is that the definition of inheritance proposed by Touretzky[1] is inherently NP-hard, and remains so even for totally acyclic

* Fellow of The Canadian Institute for Advanced Research.

[1] When speaking of "Touretzky's definition of inheritance" or "Touretzky's inheritance system," we are referring to the system defined in Touretzky (1984).

unambiguous networks. Thus there cannot be an algorithm that correctly determines if one node is inheritable from another that runs in time that is polynomial in the size of the network.[2] An immediate consequence of this is that the conditioning of a network, which Touretzky proposes to allow for fast parallel inference, is itself computationally intractable. This suggests that a Touretzky inheritance procedure cannot run unsupervised, unless the network can be restricted in form or in size.

But the news is not all bad. We also show that there are plausible variants of the Touretzky definition that are indeed tractable. Among these is the definition proposed by Horty *et al.* (1987), who first exhibited a polynomial algorithm for computing inheritance. We extend this work and demonstrate which parts of the definition are responsible for making inheritance tractable.

Again, we do not wish to claim that the tractability of inheritance implies the *correctness* of the definition; but it is certainly one issue among many that needs to be taken into account in resolving differences among competing accounts.

In the next section, we consider the precise definition of several forms of path-based inheritance. In the subsequent section, we examine the complexity of inheritance, and which combination of features in the definition affect the tractability of the associated reasoning. In the final section, we summarize our results and discuss directions for future research.

2 Path-Based Inheritance Systems

For our purposes, an inheritance network consists of a finite set TV of objects called *nodes* denoted with lower-case letters x, y, z, and a set T of *edges,* defined as follows:

Definition: Edge
An edge is an element of TV x {0,1} x TV, that is, any ordered pair of nodes and a 0-1 value called its *polarity.* Edges with a 0 are called *negative* and those with a 1 are called *positive.* We draw positive edges as *x*—>y and negative edges as *x+y.*

Intuitively, the nodes of a network are intended to stand for concepts or properties such as "bird," "penguin," "Tweety," or "flies." Edges, on the other hand, stand for statements: A positive edge *x*—>*y* stands for the statement "an *x* is normally a y," while the corresponding negative edge represents the statement "an *x* is normally not a y."[3] Path-based inheritance is concerned with the logic of statements of this type only.[4] The goal is to define what it means for a new edge *x*—>*y* to be inferrable from a given set of edges T. To do so, we use the notion of a path.

Definition: Path
A path is a sequence of edges from nodes x_0 to *x1* to . .. to x_{n-1} to x_n where $n \geq 1$ and the first n — 1 edges are positive.[5] The *polarity* of a path is the polarity of the final edge. The nodes x_0 and x_n are called the *start point* and *end point* of the path. The edge formed by taking the start and end points and the polarity of a path we call the *conclusion* supported by the path. We will let lower-case Greek letters stand for paths and draw them $x_0 \rightarrow x_1 \rightarrow \ldots x_n$.

It is tempting to define the set of edges that are inferrable from a network *T* directly as the set of all conclusions supported by at least one path formed by edges in T. The complication is that a path may be invalidated in one of two ways: it may be contradicted by other paths (in which case neither path wins) or it may be preempted by a more specific path (in which case the more specific path wins).

For the following definitions, we will let be any set of paths, σ be any path $x_0 \rightarrow x_1 \rightarrow \ldots x_n$, and *x* be any node.

Definition: Contradiction
a is *contradicted in* $ iff there is path in Φ with the same start and end points as *a,* but of opposite polarity.

So for example, the path xo—>x\—+X2 is contradicted by the path $x_0 \rightarrow x_3 \rightarrow x_4 \rightarrow x_5 \not\rightarrow x_2$ since the start and end points are the same but the final edge is of opposite polarity.

For preemption, the idea is that a path is preempted by an edge of opposite polarity from an intermediate of the path. We will consider two definitions of intermediate.

Definition: On-path intermediate
x is an *on-path intermediate of a in* $ iff for some $i \leq n$, either $x = x_i$ or Φ contains a positive path $\gamma = x_0 \rightarrow \ldots \rightarrow x_{i-1} \rightarrow y_0 \rightarrow \ldots \rightarrow y_m \rightarrow x_i$ with *x* = *yj* for some *j* < *m.* (Note that *a* and 7 must be identical up to node £;_i.)

Definition: Off-path intermediate
x is an *off-path intermediate of a m* Φ iff there is a positive path 7 in $ from *XQ* to x_{n}_i that contains x. (Note that cr and 7 can be completely disjoint except for the nodes xo and $x_{„}$_i.)

Definition: Preemption (off-path or on-path)
a is *preempted m* $\underline{\Phi}$ ifT there is a node *x* that is an intermediate of *a* in $\Phi_{„}$ and an edge in Φ of the opposite polarity of *a* from *x* to $x_{„}$.

For example, consider the set of paths $\Phi = \{C \rightarrow re \rightarrow e, e \rightarrow g, re \not\rightarrow g\}$, where C,re,e, and *g* respectively denote "Clyde," "royal elephant," "elephant," and "gray." Figure 1(a) gives the underlying inheritance network. (The figure contains an additional node *Ae,* which will be discussed below.) The path $<C \rightarrow re \rightarrow e \rightarrow g$ would be on-path preempted in Φ by the edge $re \not\rightarrow g$, since

[2]For the purpose of this paper, and to keep the provisos to a minimum, we assume that P#NP.

[3]In the case where *x* is a property, instead of 'an x," this should read "something with property x." This also applies to *y* as a property. When *x* denotes an individual concept, instead of "an *x* is normally," the phrase "x is" should be used. The case with *y* as an individual concept never arises.

[4]Touretzky also defines "no-conclusion" edges. Such edges are not often used in practice, and therefore we will ignore them here in order to simplify our definitions. Note that these edges cannot decrease the complexity of the inheritance reasoning.

[5]So every edge in a network is a path.

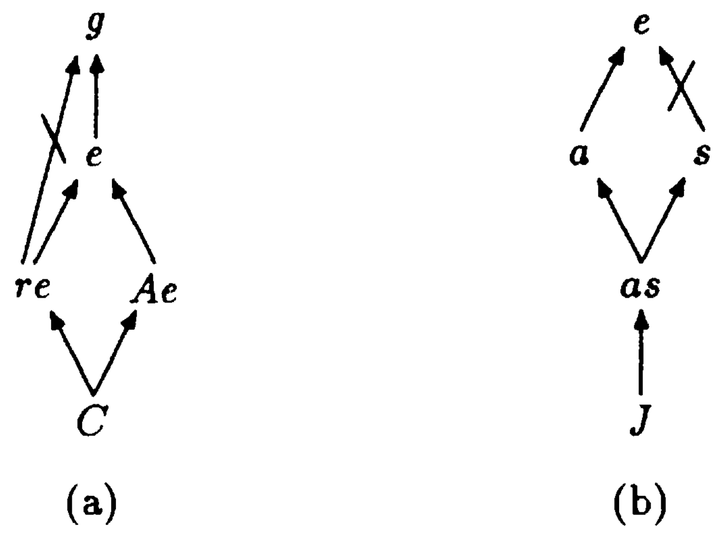

Figure 1: Examples of preemption and path concatenation. Arrows with a cross bar denote negative edges, the other arrows denote positive edges.

re is an on-path intermediate of the path.[6] It would also be off-path preempted by the same edge. On the other hand, the path $C \rightarrow Ae \rightarrow e \rightarrow g$, where *At* denotes "African elephant," would only be off-path preempted in Φ.

Contradiction and preemption tell us how a path in a network may be invalidated. But once a path is invalidated, other paths that contain it may be invalidated also. So in determining inheritable paths, we must be sure to concatenate only those paths that have not been ruled out. As it turns out, there are two ways to concatenate paths.

Definition: Upward concatenation
a is an *upward concatenation of paths in* Φ iff the last edge of *a* is in Φ and the path consisting of all but the last edge of *a* is also in Φ.

Definition: Downward concatenation
a is a *downward concatenation of paths in* Φ iff the path consisting of all but the first edge of *a* is in Φ and the path consisting of all but the last edge of *a* is also in $\blacksquare\Phi$.

The latter form of concatenation was originally called *double chaining* in Touretzky (1984). Here we opt for the more recent terminology used in Touretzky *et al.* (1987). The definition of inheritance based on upward concatenation that we will consider leads to so-called decoupled inheritance, as opposed to the coupled inheritance when using downward path concatenation.

To illustrate the difference between the two forms of concatenation, consider the following set of paths $\Phi = \{J \rightarrow as \rightarrow a, as \rightarrow s \not\rightarrow e, a \rightarrow e\}$, where *J,as,s,a,* and *e* respectively denote "Jill," "adult student," "student," "adult," and "employed." The underlying inheritance network is given in figure 1(b). The path $J \rightarrow as \rightarrow a \rightarrow e$ can now be formed by upward path concatenation. This path supports the conclusion $J \rightarrow e$, while $as \rightarrow s \not\rightarrow e$ in Φ supports *as—e.* So, in this case, Jill and the class of adult students differ w.r.t. the property "employed." In general, when there is no coupling between the properties of a class and the properties of its superclasses, one speaks of decoupled inheritance. But with downward path concatenation the path $J \rightarrow as \rightarrow a \rightarrow e$ cannot be obtained from the paths in Φ. To do so, one would also need *as->a—+e* which, in fact, is contradicted in Φ by $as \rightarrow s \not\rightarrow e$. So, in this case, there is a coupling between the properties that Jill can inherit and those inherited by the class of adult students.

We now define the inheritable paths.

Definition: Inheritable path (on-path and off-path, downward and upward)
σ is *inheritable in* Φ iff

(a) σ is a concatenation of paths in Φ;
(b) σ is not contradicted in Φ;
(c) σ is not preempted in Φ.

Intuitively, the paths that are inheritable in Φ are those that are inferrable from but not invalidated by Φ. But where does this set Φ come from? There are different ways of choosing a set Φ. We first consider Touretzky's definition of so called credulous inheritance reasoning. In this approach, Φ is chosen to be the least set of paths whose edges are those of *T* and closed under inheritance. We call such sets credulous grounded extensions.[7]

Definition: Credulous grounded extension
Φ is a *credulous grounded extension of a set of edges T* iff

(a) $\Gamma \subseteq \Phi$;
(b) For all σ, $\sigma \in \Phi - \Gamma$ iff σ is inheritable in Φ.

Unfortunately, there need not be a unique credulous grounded extension for a given set F, that is a network can be *ambiguous.* Inheritance reasoners allowing for multiple extensions, such as Touretzky's, are called credulous reasoners because they explore the various alternatives in different extensions. Instead of allowing for multiple extensions, Horty *et al.* (1987) propose a form of skeptical inheritance in which at most one grounded extension is generated. In this form of inheritance a unique extension is inductively constructed by including paths that are inheritable only if a path with the same start and end point but of opposite polarity could not be inherited. The induction is based on the degree of a path:

Definition: Degree
Given a set of edges F, the degree of a path with start point *x* and end point *y* is the length of the longest path in T from *x* to *y* (ignoring the polarity of the edges).

Horty *et al.* restrict their definition of skeptical inheritance to acyclic networks:

Definition: Acyclic
A set of edges T is *acyclic* iff the graph formed by the elements of *Y* is acyclic.[8]

We can now state the definition of skeptical inheritance as follows:

Royal elephant is a subclass of elephant; therefore, information associated with it should override information associated with the elephant class. This is precisely what is captured by preemption.

[7]These were called grounded expansions in Touretzky (1984).

[8]Touretzky (1984) speaks of *totally acyclic,* as distinguished from *IS-A acyclic* networks in which the graph formed by only the positive edges is required to be acyclic.

Definition: Skeptical grounded extension
Φ is a *skeptical grounded extension of a set of edges T* iff $\Phi = \bigcup_{i=1}^{\infty} \phi_i$ where ϕ_i is defined as follows:

(a) $\phi_1 = \top$;

(b) ϕ_{i+1} contains the paths in ϕ_i and each path *a* of degree *i* + 1 with the following properties:

- a can be obtained by concatenation of two paths in ϕ_i;[9]
- there neither exists an edge in øi, nor a path inheritable in ϕ_i with the same start and end point as a but of opposite polarity.

What we ultimately care about is the conclusions (that is, the edges defined by the start and end points) supported by the paths in a grounded extension of a network. So, we define:

Definition: Conclusion set
A set of edges is a *conclusion set* for *T* iff for some grounded extension ø, the edges are all the conclusions supported by the elements of ϕ.

The inheritance problem, then, is this:

Definition: Inheritance problem
Given an acyclic network T, find a conclusion set of *T.*

Note that we are really talking about 8 inheritance problems here according to whether we consider off-path or on-path preemption, upward or downward path concatenation, and skeptical or credulous reasoning. Touretzky's definition, for example, would be classified as on-path, downward, and credulous, while Horty's version is off-path, upward, and skeptical.

We now consider the computational difficulty of the inheritance problem.

3 Computational Complexity

The following theorem shows that for Touretzky's inheritance notion there is no polynomial algorithm that takes as input an acyclic network F and finds a conclusion set of T.

Theorem 1 *The inheritance problem for on-path, downward, credulous inheritance (Touretzky 1984) is NP-hard.*

The proof of this theorem is based on a reduction from the NP-complete decision problem "path with forbidden pairs" (or PWFP) defined by Gabow, Maheshwari, and Osterweil (1976). An instance of PWFP consists of a directed graph G = *(V,E),* specified vertices *s,t* *V,* and a collection *C* = {(a_1,b_1),..., (a_n, b_n)} of pairs of vertices from *V*. The question is: does there exist a path from *s* to *t* in G that contains at most one vertex from each pair in C? This problem remains NP-complete even if we only consider acyclic graphs and all pairs in *C* are disjoint. Consider an instance of this restricted version of PWFP. Without loss of generality, we may assume that each forbidden pair (a,, bi) is such that there does not exist a path from bi, to ai in *C.*

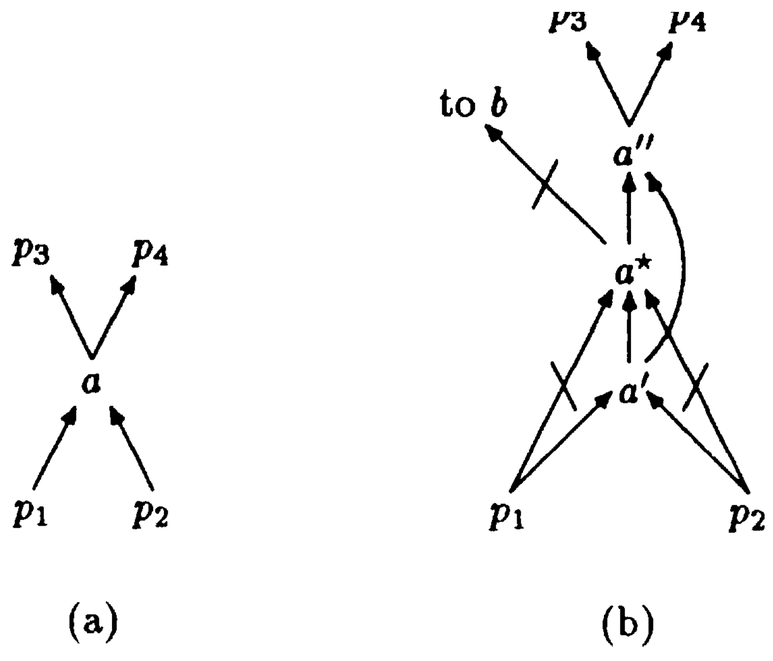

Figure 2: Construction used in proof of theorem 1.

We now construct an acyclic network *T* from this instance of PWFP. First, we include every edge of *G* as a positive edge of *T.* Then for every node *a* that is the first element of a forbidden pair in C, we replace *a* in *T* by the network shown in figure 2. Figure 2(a) shows a node *a* from the forbidden pair (a, b,) with all of its its neighbors in G, and 2(b) shows the structure in *T* that replaces a: two additional nodes (for a total of three) and *n* + 4 additional edges must be included, where *n* is the number of edges pointing to *a* in G. Note that of the three new nodes, the middle one a* is linked to 6 by a negative edge. Since G is acyclic and the forbidden pairs are such that there is no path from bi to ai for any pair in C, it follows that *T* is an acyclic network. Moreover, F is such that given an arbitrary on-path, downward, credulous conclusion set *C* of T, there is a path in G from *s* to *t* containing at most one vertex from each forbidden pair iff the conclusion *s*—>*t* is in C.[10]

Now, consider an algorithm that takes as input an instance of PWFP, constructs (in polynomial time) an acyclic network *T* as outlined above, then finds an on-path, downward, credulous conclusion set *C* of T, and, finally, returns "yes" if *s*—>*t*øG, and and "no" otherwise. Such an algorithm returns "yes" iff there exists a path from *s* to *t* in *G* containing at most one node from each forbidden pair; but if finding a conclusion set can be done in polynomial time, the overall algorithm will run in polynomial time. Since PWFP is NP-hard, the inheritance problem for on-path, downward, credulous reasoning must be NP-hard too.

The construction shown in figure 2 exploits various properties of inheritance reasoning in general, and Touretzky's notion specifically. Firstly, we rely on preemption since preemption prevents inheritance of a path from *s* to *t* going through *a"*and *b* (corresponding to a

[9]The perhaps more natural condition "*a* is inheritable in øi" leads to a different notion of skeptical inheritance. We use the condition given above for compatibility with Horty *et al.* (1987). Under this definition, the credulous grounded extension of an unambiguous inheritance network need not be identical to its skeptical grounded extension (Selrnan and Levesque 1989).

[10]We prove this property of *T* in Selman and Levesque (1989).

path through *a* and 6 in *G)* because of the negative edge between a* and 6.[11] Secondly, we use *quasi-redundant* edges (Touretzky 1984, p. 10; Horty *et ai* 1987, Techn. Report, p. 21). The edge *a'→a"* is an example of a quasi-redundant edge, since this edge is strictly redundant for concluding *a'→a";* however, to be able to conclude, for example, p_1 —>a", the edge is essential (so it allows us to have a path from *s to t* through *p1* and a" in a grounded extension, corresponding to a path from *s to t* via *p1* and *a* in *G).* Thirdly, the reduction relies on coupled inheritance; we will see below that decoupled (upward) reasoning allows for polynomial time inheritance.

A direct consequence of theorem 1 is that the conditioning of a network as proposed in Touretzky (1984), to enable fast parallel inference, is also intractable. Conditioning is a process of adding edges to an inheritance network in such a way that a parallel marker-passing procedure can subsequently be used to draw conclusions in time proportional to the height of the inheritance network.

We show in Selman and Levesque (1989) that the inheritance network *T* used in the above reduction is unambiguous. Thus, it follows that even when we restrict ourselves to unambiguous acyclic networks the inheritance problem based on Touretzky's notion of inheritance is intractable. Note that it also follows that determining whether an unambiguous acyclic network supports a particular conclusion is NP-hard.

Recently, Geffner and Verma (1989) used a variation of our reduction to show that reasoning based on their definition of defeasible inheritance is NP-hard. Thus, the technique given above may prove to be useful in determining the complexity of future proposals for path-based inheritance reasoners.

We will now consider the influence of the various design choices made in Touretzky (1984; Horty *et ai* 1987) on the computational complexity of the inheritance reasoning. Our results are summarized in the following theorem:

Theorem 2 *The computational complexity of the inheritance problem for the various choices between off-path or on-path, upward or downward path concatenation, and skeptical or credulous reasoning is given in the following table (P stands for doable in polynomial time):*

	skeptical		credulous	
	off-path	on-path	off-path	on-path
down	*NP-hard*	*NP-hard*	*NP-hard*	*NP-hard*
up	*P*	*P*	*P*	*P*

Clearly the choice between off-path or on-path preemption and between skeptical or credulous inheritance does not change the complexity of inheritance reasoning. However, when we consider upward (decoupled) inheritance, we do obtain tractability. The latter finding generalizes the tractability result obtained by Horty *et al.* (1987) for their off-path, upward, skeptical inheritance reasoner. The NP-hardness results are proved by showing that the various design choices do not affect the correctness of the above reduction when dealing with

Downward concatenation is also relevant here.

downward path concatenation. Details are given in Selman and Levesque (1989) which also contains polynomial algorithms for the tractable cases.

To summarize, theorem 2 clearly shows that the type of path concatenation (upward or downward) is the determining factor in the complexity of the reasoning. This distinction corresponds to the difference between decoupled and coupled reasoning. It should be noted though, that in formalisms with substantially different definitions of preemption or grounded extension, other factors may also influence the complexity of the reasoning.

A further understanding of the complexity issues underlying inheritance reasoning can be obtained by considering the polynomial algorithms for upward inheritance. These algorithms iteratively construct a conclusion set. The main difficulty in adding a new conclusion to the set arises from having to determine whether the path that supports this conclusion is preempted or not. So, we have to search for intermediates, which requires access to the set of *paths* in the underlying grounded extension. At worst, one would have to keep track of the full extension obtained so far (which can be of exponential size). But for upward inheritance it is sufficient to keep track of the current conclusion set, since one can rederive in polynomial time any particular path in the underlying extension, and thus, search for intermediates efficiently.

As a final topic, we will consider *goal-directed* inheritance. Note that so far we have considered the complexity of finding any arbitrary conclusion set given an inheritance hierarchy. In some situations, however, it might be wasteful to generate the entire conclusion set, and moreover, one might be interested in an extension that supports some particular conclusion (or set of conclusions). We therefore define the problem of goal-directed inheritance reasoning: given an inheritance network *T* and a conclusion *x→y* (or *x+y),* does there exists an extension of *T* supporting *x→y* (or *x-f+y)!*

When a network is unambiguous or when skeptical reasoning is desired, the computational complexity of this inference task is essentially the same as that of searching for a conclusion set, since, on the one hand, after finding the unique conclusion set, it a trivial to determine whether it contains a certain conclusion, and, on the other hand, after at most a polynomial number of queries, one can determine the conclusion set (for a network containing *n* concepts one has to consider at most 2n(n — 1) possible conclusions). Moreover, our reduction from PWFP shows that goal-directed inheritance for on-path and off-path, credulous, downward inheritance is NP-hard, just like the problem of finding an extension. (Consider the query: does the constructed network have an extension that supports the conclusion *s—>t?)* The following theorem, however, shows that goal-directed reasoning is strictly more difficult than searching for an arbitrary conclusion set:[12]

[12]Kautz and Selman (1989) show that goal-directed reasoning for default logic is also strictly harder than generating an arbitrary extension.

Theorem 3 *Goal-directed, on-path, upward, credulous inheritance is NP-hard.*

The proof of this theorem is again based on a reduction from PWFP (Selman and Levesque 1989). For this reduction, it is essential for the constructed network to be ambiguous. This theorem reveals some of the extra difficulties in inheritance reasoning due to ambiguities, although these difficulties do not arise when searching for an arbitrary conclusion set, as shown by theorem 2. The complexity of goal-directed, off-path, upward, credulous inheritance remains an open problem.

4 Conclusions

We have shown that path-based inheritance reasoning as defined in Touretzky (1984) is NP-hard, even when restricted to acyclic unambiguous networks. Moreover, the versions of this form of inheritance that use skeptical reasoning and off-path preemption are also intractable. Thus, while Touretzky (1984) showed that inheritance networks can be conditioned to allow for correct and efficient retrieval of information (time *0(log(n))* for n concepts) such as in NETL (Fahlman 1979), our results demonstrate that this conditioning procedure itself is intractable when based on downward (coupled) inheritance. However, our other complexity results, generalizing the tractability result obtained by Horty *ct ai* (1987), also suggest that the various forms of upward (decoupled) inheritance can be used to achieve tractability.

One possible direction for future research is to consider further restrictions on the form of inheritance networks that would allow for a polynomial inference mechanism based on downward inheritance. One candidate we have already begun to explore is inheritance restricted to completely balanced hierarchies. Such hierarchies have a maximum depth of *0(log(n)),* where *n* is the total number of concepts in the hierarchy. Such a restriction seems quite reasonable, given that taxonomic hierarchies will often be "shallow." However, we have found a reduction from the small-clique problem[13] to downward inheritance reasoning with such networks. This result indicates that Touretzky's inheritance notion restricted to balanced hierarchies is most likely still intractable (i.e., not polynomial). See Selman and Levesque (1989) for a more detailed discussion of these issues.

Acknowledgments

This work was supported in part by a Government of Ontario Scholarship to the first author, and a grant to the second from the Natural Sciences and Engineering Research Council of Canada. We would like to thank David Etherington for pointing out to us that the complexity of inferential distance was unknown. We also would like to thank Sue Becker, Jim des Rivieres, Russ Greiner, Jeff Horty, Gerhard Lakemeyer, Peter Patel-Schneider, Rich Thomason, David Touretzky, Wlodek Zadrozny, and the anonymous referees for providing useful comments.

[13]Given a graph *G* with *n* vertices, does *G* contain a clique of size *log(n)?* (Karchmer 1989; Megiddo and Vishkin 1988)

References

Brachman, R.J. (1985). I Lied About the Trees (or, Defaults and Definitions in Knowledge Representation). *The AI magazine* 6 (3), 1985, 80-93.

Fahlman, S.E. (1979). NETL: *A System for Representing and Using Real-World Knowledge.* Cambridge, MA: MIT Press, 1979.

Gabow, H.N., Maheshwari S.N, and Osterweil L. (1976). On Two Problems in the Generation of Program Test Paths. *IEEE Trans. Software Engineering,* 1976, 227-231.

Geffner, H. and Verma, T. (1989). Inheritance = Chaining + Defeat. Technical report TR-129, Dept. of Computer Science, University of California, Los Angeles, CA, 1989.

Horty, J.F., Thomason, R.H., and Touretzky, D.S. (1987). A Skeptical Theory of Inheritance in Nonmonotonic Semantic Nets. *Proceedings of AAAI-87,*1987. More complete version: technical report CMU-CS-87-175, Dept. of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, 1987.

Karchmer, M. (1989). Personal communication.

Kautz, H.A. and Selman, B. (1989) Hard Problems for Simple Default Logics. *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning ,* Toronto, Ont., Canada, 1989.

Megiddo, N. and Vishkin, U. (1988). On Finding a Minimum Dominating Set in a Tournament. *Theoretical Computer Science,* 61, 1988, 307 - 316.

Quillian, M. (1968). Semantic Memory. In *Semantic Information Processing,* ed. M. Minsky, 216-70. Cambridge: MIT Press, 1968.

Reiter R. (1987). Nonmonotonic Reasoning. *Annual Reviews of Computer Science,* 1987.

Reiter, R. and Criscuolo, G. (1983). Some Representational Issues in Default Reasoning, *Computers & Mathematics with Applications,* (Special Issue on Computational Linguistics), 9 (1), 1983, 1-13.

Sandewall, E. (1986). Nonmonotonic Inference Rules for Multiple Inheritance with Exceptions. *Proceedings of the IEEE,* vol. 74, 1986, 81-132.

Selman, B. and Levesque, H. J. (1989). The Tractability of Path-Based Inheritance. Technical Report, Dept. of Computer Science, University of Toronto, Toronto, Ont., Canada, forthcoming.

Touretzky, D.S. (1984). The Mathematics of Inheritance Systems. Report CMU-CS-84-136, Dept. of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, 1984; also London, UK: Pitman, 1986.

Touretzky, D.S., Horty, J.F., and Thomason, R.H.(1987). A Clash of Intuitions: The Current State of Nonmonotonic Multiple Inheritance Systems. *Proceedings IJCAI-87,* Milan, Italy, 1987, 476-482.