Inheriting Well-formed Formulae in a Formula-Augmented Semantic Network

Leora Morgenstern
IBM T.J. Watson Research Center
30 Saw Mill River Road, Hawthorne, NY 10532
leora@watson.ibm.com

Abstract

We examine a commercial application, a large portion of whose domain knowledge is taxonomic in nature, and investigate the suitability of inheritance networks for the task of reasoning about knowledge in this area. We claim that standard inheritance networks are not sufficiently expressive to capture domain knowledge, and introduce *formula-augmented semantic networks* (FANs), a semantic network which attaches well-formed formulae to nodes. We investigate the concept of inheriting wffs in such a network and give a method for computing the sets of well-formed formulae that apply to a given node. Finally, we compare FANs to more familar knowledge structures and the task of inheriting wffs to more familar concepts such as reasoning about sets of default rules in non-monotonic systems.

1 OVERVIEW/MOTIVATION

Inheritance has long been recognized as useful in business domains. This paper investigates the suitability of inheritance networks for reasoning about benefits in the medical insurance domain. The goal has been to develop an expert system that can answer customers' questions about their insurance benefits. Customers typically ask whether a particular service — such as a hip replacement or a CAT scan — is covered, and what restrictions and limitations apply to this coverage (e.g., deductible, maximum number of services per benefit period).

Because so much of the information about medical services is taxonomic in nature — e.g., spinal manipulation is a type of physical therapy; physical therapy, speech therapy, and occupational therapy are all types of therapy — and because coverage and accompanying restrictions are to a large extent inherited along these taxonomic lines, it is natural to encode much of this information in an inheritance network. On the other hand, a significant chunk of information is in the nature of complex business rules specifying certain restrictions that cannot easily be mapped into a network structure. For example, rules such as

> The initial newborn exam must be performed during the mother's hospitalization for delivery

and

> If a patient has exceeded the substance abuse benefits for the year, payment for hospitalization charges may be made under the patient's inpatient hospital benefit

do not map into class-subclass or class-attribute links. Instead, it seems more natural to regard such rules as well-formed formulae, and to reason directly with such wffs.

To accomplish this, we introduce an extension of inheritance networks which we call *formula-augmented semantic networks* or *formula-augmented inheritance networks* (FANs). A FAN looks much like an inheritance network, except that a set of wffs may be associated with each node. The question that we must then answer is: can the wffs at a node be inherited by more specific nodes in the network? General wff-inheritance leads to rampant inconsistency; thus, the main focus of this paper is on how we can perform wff-inheritance in a coherent manner. In particular, we discuss criteria of specificity and explicit path preference in guiding wff-inheritance.

This paper is organized as follows. We describe the commercial domain and the benefits inquiry system. We then define formula-augmented semantic networks and specify the task of wff-inheritance. Subsequently, we explore various methods for wff-inheritance and provide a method for computing the wffs that apply to a node. Finally, we discuss the relation between FANs and other knowledge structures, and between wff-inheritance and reasoning about prioritized rules in other non-monotonic systems.

2 FORMULA-AUGMENTED SEMANTIC NETWORKS

2.1 BENEFITS INQUIRY

The need for formula-augmented semantic networks was discovered while developing a benefits inquiry system for a large medical insurance corporation. The company offers its members a choice of *products.* A product is a collection of benefits, services, and business rules, which are related in the following manner: A service may be *covered* or *excluded* by a particular benefit, and is subject to some set of rules, specifically cost-share, administrative, and medical rules, as well as (possibly) restrictions on the set of medical professionals or facilities that are available. Each product has on the order of 500–1000 rules.

A benefits inquiry system is intended to aid customer service representatives in answering customers' questions. Customers may wish to know if a particular service is covered, or the specific rules that limit coverage. Examples of typical questions are:

```
"Will my son's tonsillectomy be
covered?"
```

```
"How many days can I stay in the
hospital after a standard delivery?"
```

An inheritance hierarchy is a natural choice to represent much of the underlying knowledge for this domain, since, as mentioned in Section 1, so much of the information is taxonomic in nature or is inherited along the taxonomy.

Representing business rules within an inheritance hierarchy is a difficult task. Simple rules such as

> The copay for medical services is 20%

can easily be represented by specifying the attribute of having a 20% copayment. But many of the rules are considerably more complex. For example, consider the rule

> Medical services incurred by an organ donor are covered by the recipient's Human Organ Transplant benefit only if the donor is otherwise not covered.

There does not seem to be any natural way to represent this in a standard inheritance network. This motivates our definition of a formula-augmented semantic network in the next section. In Section 4, we consider whether FANs are truly necessary, or whether we can encode the information in a more traditional structure.

2.2 DEFINITION OF A FAN

A Formula-Augmented Semantic (or Inheritance) Network is a tuple $(\mathcal{N}, \mathcal{W}, \mathcal{B}, \mathcal{L}1, \mathcal{L}2, \mathcal{O})$, where
• $\mathcal{N}$ is a set of nodes,
• $\mathcal{W}$ is a set of wffs,

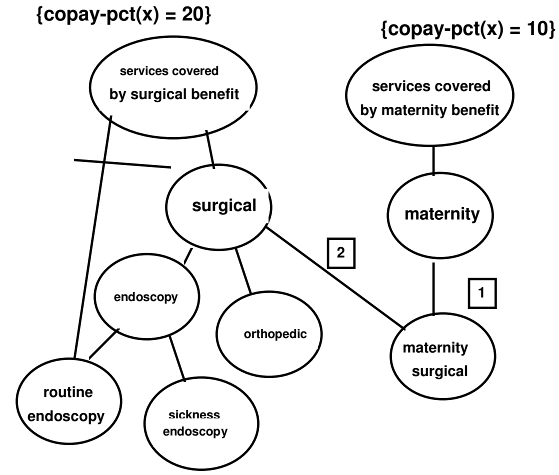

Figure 1: *A portion of the medical insurance network. Lines represent is-a links; slashed lines represent cancels links*

• $\mathcal{B}$ is the background information or background context,
• $\mathcal{L}1$ is a set of links connecting nodes — i.e. $\mathcal{L}1 \subseteq \mathcal{N} \text{ X } \mathcal{N}$,
• $\mathcal{L}2$ is a set of links connecting nodes to sets of wffs — i.e. $\mathcal{L}2 \subseteq \mathcal{N} \text{ X } 2^{\mathcal{W}}$, and
• $\mathcal{O}$ is a (possibly empty) set of partial orderings on some of the members of $\mathcal{L}1$, subject to certain restrictions which are discussed below.

Intuitively, in this domain, each instance of a FAN represents a particular product of the medical insurance corporation.

The elements of the tuple are described in detail below. The description of $\mathcal{L}1$, which is part of any standard inheritance network, may be safely skipped by those familiar with the work of (Horty et al., 1990). $\mathcal{W}$, $\mathcal{B}$, $\mathcal{L}2$, and $\mathcal{O}$, however, are unique to FANs.

2.2.1 Nodes

A node represents some set of medical services. E.g., the node Surgical represents the set of surgical services; the node X-ray represents the set of x-ray services.

A node may represent a set of services that are covered by a particular benefit. For example, in Figure 1, the node Services Covered by Surgical Benefit represents the set of all medical services that are covered by the Surgical Benefit.

2.2.2 Wffs

The set of wffs $\mathcal{W}$ consists of well-formed formulae of a sorted first-order logic. An example of a rule is

$\forall$ *e1,e2, e3, m, b, i1, i2*
(delivery-event(e1,m,b) $\wedge$ *hospital-event(e2,b)*
$\wedge$ *assoc-hosp-stay(e2,e1)* $\wedge$ *newborn-exam-ev(e3,b)*
$\wedge$ *time(e2,i1)* $\wedge$ *time(e3,i2)* $\wedge\neg$ *subinterval(i2,i1))*
$\supset \neg$ *covered-as(e3,newborn-exam)*,
where *e1,e2,* and *e3* are variables ranging over events; *i1* and *i2* are variables ranging over time intervals; and *b* and *m* are variables ranging over babies and mothers, respectively.

This is the first-order translation of the business rule

> The initial newborn exam must be performed during the mother's hospitalization for delivery.

2.2.3 Background Information

The *background* $\mathcal{B}$ is a (possibly empty) set of wffs of first-order logic. Intuitively, $\mathcal{B}$ represents the background information that is true. It includes all rules that are true of all medical services and benefits. [1] For example, the background context may include a wff which states that all medical services must be provided by licensed medical professionals in order to qualify for reimbursement. $\mathcal{B}$ may also include information about particular individuals, such as a patient's medical records and the name of her primary care physician, as well as pricing charts for drugs and pay scales for medical professionals. As these examples make clear, although the information at $\mathcal{B}$ is considered to be global, in the sense that it applies to every node in the network, it may also be subject to frequent change.

2.2.4 Links

Links are relations on objects. There are two sorts of links: links joining nodes and links joining nodes to sets of wffs.

$\mathcal{L}$1: Links on nodes: The standard positive and negative is-a links relate nodes. (Negative is-a links will also be referred to as cancels links.) Positive and negative is-a links may be *strict* or *defeasible*. If x and y are nodes, we write x $\Rightarrow$ y and x $\not\Rightarrow$ y to represent, respectively, the strict positive and strict negative is-a links; we write x $\rightarrow$ y and x $\not\rightarrow$ y to represent, respectively, the defeasible positive and defeasible negative is-a links. Intuitively, x $\Rightarrow$ y means that all x's are y's; x $\not\Rightarrow$ y means that all x's are not y's (no x's are y's); x $\rightarrow$ y means that typically, x's are y's; x $\not\rightarrow$ y means that typically, x's are not y's. If x $\Rightarrow$ y or x $\rightarrow$ y, we say that x is a *child* of y or that y is a *parent* of x.

Our account of paths is based on (Horty, 1994). A path is a restricted sequence of positive and/or negative links. We may define paths recursively in the following manner:

- There is a path from x to y iff there is a positive path from x to y or there is a negative path from x to y.
- If x $\rightarrow$ y or x $\Rightarrow$ y (resp., x $\not\rightarrow$ y or x $\not\Rightarrow$ y), then there is a positive (resp. negative) path from x to y.
- If there is a positive path from x to y, and y $\rightarrow$ z or y $\Rightarrow$ z (resp. y $\not\rightarrow$ z or y $\not\Rightarrow$ z), then there is a positive (resp. negative) path from x to z.
- If there is a negative path from y to z and x $\rightarrow$ y or x $\Rightarrow$y, then there is a negative path from x to z.

Note that positive paths can have only positive is-a links; negative paths can have just one negative link at the very end of the path. Paths that contain only strict links are called *strict paths*; paths that contain only defeasible links are *defeasible paths*; paths that contain both strict and defeasible links are called *mixed paths*. For ease of presentation, this paper will be concerned almost exclusively with defeasible paths; extensions to strict and mixed paths are straightforward [2]. The notation $\pi(x, \sigma, y)$ (resp. $\pi'(x, \sigma, y)$) represents a positive (resp. negative) path from x to y through the path σ. We extend this notation so that $\pi(x, \sigma)$ (resp. $\pi'(x, \sigma)$) represents a positive (resp. negative) path from x to the last point of σ going through the path consisting of all but the last point of σ. The function *endpoint(σ)* will be used to designate the last point on the path σ.

Often, there will be both positive and negative paths between two nodes. To determine which path to choose, we follow the analysis of Touretzky(1986) and Horty(1994). Given a *context* — an inheritance network Γ and a set of paths Φ (intuitively, a set of paths arising out of the network), a path (of length greater than 1) is *inheritable* or *undefeated* if it is *constructible* and neither *pre-empted* nor *conflicted*. Briefly, a path is *constructible* in a context if it can, recursively, be built out of the paths in the network; a path is *conflicted* in a context if there is a path of opposite sign in the context with the same starting and ending points; a path is *preempted* if there is a conflicting path with more direct information about the path's endpoint (i.e., a direct link from an earlier point in the path). We also say that a path $\pi(x, \sigma, y)$ is preempted or conflicted if $\pi(x, \sigma)$ is preempted or conflicted. Intuitively, the undefeated or inheritable paths are those which "win out" over their rivals. If there is an undefeated positive (resp. negative) path between x and y, we say x $\rightsquigarrow$ y (resp. x $\not\rightsquigarrow$ y).

Examples of paths can be seen in Figure 1. There is a positive path from Orthopedic Surgery to Services Covered by Surgical Benefit. There are both positive and negative paths from Routine Endoscopy to Services Covered by Surgical Benefit; however, according to Horty et al.'s (1990) specificity criterion (or Touretzky's (1986) inferential distance criterion), the positive path is preempted by the negative path. Thus, the negative path is undefeated.

[1] If we wish, we may think of the wffs in $\mathcal{B}$ as being included in the set of rules at each node.

[2] Note also that in all examples, nodes represent sets of objects, rather than individual objects. This is done to simplify the exposition; the extension to individuals is straightforward.

$\mathcal{O}$**: Ordering on Links:** Multiple inheritance arises when there is an undefeated path from x to y, an undefeated path from x to z, and y $\neq$ z. We call any such point x a *fork point* of the network, and the paths originating from a fork point *multiple paths.* [3] Inheritance networks in the literature have traditionally considered multiple inheritance only when these multiple paths have been initial segments of conflicting paths — as is the case in Figure 1, where the multiple paths π(Routine Endoscopy, Endoscopy) π'(Routine Endoscopy, Services Covered by Surgical Benefit) are initial segments of conflicting paths. We call such cases of multiple inheritance *conflicting-path multiple inheritance.*

In this paper, we will be interested in multiple paths even where they are *not* initial segments of conflicting paths, i.e., *non-conflicting-path multiple inheritance.* Examples of non-conflicting-path multiple inheritance can be seen in Figure 1, where there are distinct non-conflicting paths between Maternity Surgical and Maternity, and between Maternity Surgical and Surgical.

In such cases, we may allow the prioritization of a particular path. We do this by first specifying a partial order on certain links of $\mathcal{L}1$. Specifically, if x $\rightarrow$ y_1, x $\rightarrow$ y_2, ..., x $\rightarrow$ y_n are elements of $\mathcal{L}1$ and are not initial segments of conflicting paths, we may place a partial order on the x $\rightarrow y_i$'s; this partial order is an element of $\mathcal{O}$.

The partial order on paths can then be defined recursively as follows:

•$x \rightarrow y$ is preferred to $x \rightarrow z$ if $((x, y), (x, z)) \in \mathcal{O}$', where $\mathcal{O}$' is an element of $\mathcal{O}$.
•$\pi(x, \sigma, y)$ is preferred to $\pi(x, \tau, z)$ if $\pi(x, endpoint(\sigma))$ is preferred to $\pi(x, endpoint(\tau))$.
•$\pi(x, \sigma, y)$ is preferred to $\pi(x, \sigma, z)$ if $\pi(endpoint(\sigma), y)$ is preferred to $\pi(endpoint(\sigma), z)$.

If there are p fork points in the network, there are at most p partial orders (elements) in $\mathcal{O}$. The number of partial orders in $\mathcal{O}$, and the elements of these partial orders may be further restricted; in particular, we may not wish to specify priorities on links that are initial segments of conflicting paths, or may insist that the priorities be placed in a particular way. We discuss this further in Section 3.2.4.

$\mathcal{L}2$**: Links between nodes and wff sets:** Let N be a node, and W a set of wffs. $N \rightarrow_w W$ means, intuitively, that each w of W is typically true at node N. $N \Rightarrow_w W$ means that each wff w of W is true at N; $N \not\rightarrow_w W$ means that each wff w of W is typically false at node N; $N \not\Rightarrow_w W$ means that each wff w of W is false at N. In practice, we rarely use the $\Rightarrow_w$ link. Intuitively, this link is used only when a formula is true at all nodes Ni such that $Ni \rightsquigarrow N$; i.e., we assume that there are no exceptions to N. This happens only rarely.

[3] As opposed to a *branch point,* which is downward.

Since W can be a singleton, we allow the overloading of the $\rightarrow_w$, $\Rightarrow_w$, $\not\rightarrow_w$ and $\not\Rightarrow_w$ links so that they can refer to individual wffs as well as sets of wffs. If $N \rightarrow_w W$ or $N \Rightarrow_w W$, we will sometimes refer to W as the set of wffs at N or wffs(N). If w is a member of W, we say $at(w, N)$.

The set of rules at each node in a network is constrained to be consistent with the background information $\mathcal{B}$.

2.3 FANs with Added Link Types

The informal description of an insurance product in Section 2.1 might lead one to expect the network to have nodes representing benefits as well as nodes representing sets of services. We would then need covers and excludes links between benefit nodes and service nodes, as well as the usual taxonomic links connecting service nodes to service nodes (and benefit nodes to benefit nodes).

Although this representation is in some sense more natural than then one given in this paper (and is in fact the representation used in the implemented benefits inquiry system), we have opted against it. Because we wish to model this network as much as possible as a standard inheritance network, we choose to reify a node representing a benefit x as a node representing the services covered by benefit x (as in Figure 1). This allows us to use standard is-a and cancels links instead of covers and excludes links. This decision is not integral to the concept of a FAN. We could easily augment the definition of a FAN to include links such as covers and excludes, and specify the interaction between these links and the taxonomic links.

Although we do not explore this option further in this paper, we do note here an interesting feature of the FAN with covers and excludes links (called a modified FAN for the purposes of this discussion) for the insurance application under discussion. The only taxonomic link needed is the standard (strict) is-a link. There is no need for defeasible links, and thus no need for cancel links. Intuitively, the reason for this is that in the insurance application, most of the taxonomic links are genuine and not defeasible subset relations. The only place the defeasible link is needed is between a node and a "services-covered-by-benefit-x" node. But this link is replaced by the covers or excludes nodes in a modified FAN.

Indeed, this discussion highlights an interesting feature of FANs in this domain: cancel links always go directly up to the root node. Due to this feature, the algorithm to identify undefeated paths is much simpler than the Horty et al. (1990) or Stein (1992) algorithm, and in particular, requires only a simple upward traversal from a node until one reaches either a cancels link or a root. While this feature has been exploited in the implementation, this paper discusses the more general case of FANs in which general inheritance with exceptions is allowed.

It should also be noted that the problem of wff-inheritance discussed in this paper applies not only to FANs but to modified FANs. This shows that the wff-inheritance problem can arise in a strict taxonomy as well as in an inheritance hierarchy with exceptions.

3 INHERITING WELL-FORMED FORMULAE

3.1 THE PROBLEM

One of the defining features of FANs (as opposed to standard semantic networks) is that wffs are associated with nodes. This feature prompts the following question:
What wffs are *inherited* by a node in a FAN? In a standard inheritance network, we focus primarily on the question: is there an undefeated positive path between A and B? Since most nodes in a network represent sets, and since the top node in an inheritance hierarchy often represents some attribute, the most intuitive interpretation of this question is: does some attribute hold of set A? Such questions are of interest in this inheritance network as well: we wish to know whether a particular medical service is covered by some benefit. In addition, however, we are interested in determining which business rules or wffs *apply* or *pertain* to that service (equivalently, to the node representing that service).

Formally, we pose the question as follows. We introduce the following notation: N$\uparrow w$ if wff w pertains or applies to node N. We overload the $\uparrow$ symbol for sets, and say N$\uparrow$ W if $\forall$ w $w \in W$ iff N$\uparrow w$. We define Ψ(N) = $\{w \mid$ N$\uparrow$ w $\}$. That is, Ψ(N) is the set of wffs that apply to node N. Then the question is: given a node N in a FAN, what is the set Ψ such that N$\uparrow$ Ψ?

A naive approach might suggest taking the union of the wff set attached to N, together with the wff sets attached to all ancestors of N — or more precisely, of the wff sets attached to all those nodes to which there is an undefeated positive path from N. That is,
$\Psi(\mathrm{N}) = \bigcup_{N\rightsquigarrow\ N_i}$ wffs(N_i)$\cup$ wffs(N).
Such an approach, however, is obviously wrong. Consider, for example, Figure 2. There is an undefeated positive path from N3 to N1. Thus, according to the naive approach,
Ψ(N3) = wffs(N3) $\cup$ wffs(N1) =
$\{$ P $\neg$ Q $\}$ $\cup$ $\{$ R, P $\supset$ Q $\}$. But this set of wffs is obviously inconsistent. Note also that the naive approach fails to correctly compute Ψ(N2). Although wffs(N1) $\cup$ wffs(N2) is consistent, it is not consistent with respect to the background information $\mathcal{B}$ = $\{$ $\neg$ R $\vee \neg$ S $\}$. Clearly, we do not want to blindly take unions of wff sets.

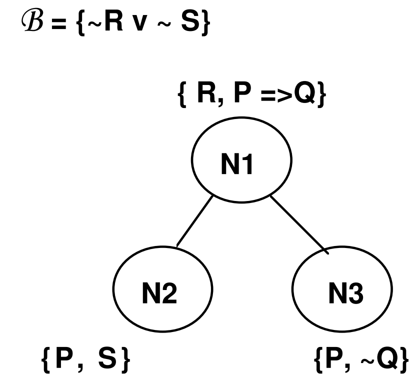

Figure 2: *Taking the union of wffs at nodes yields inconsistency.*

The problem arises because the wffs at a node are usually *typically* true at a node rather than always true at a node. In particular, a set of wffs W may be typically true at a node N, but may not be typically true at all subclasses of N. We discuss ways of dealing with this inconsistency in Section 3.2.

3.2 EFFECTING INHERITANCE OF WFFS

The previous section demonstrated that a naive approach to wff inheritance — namely, taking the union of wff sets at all nodes to which there is a positive undefeated path — leads to inconsistency. A proper approach to wff-inheritance must *recognize* potential inconsistency in inheriting wffs and *resolve* these inconsistencies.

Thus, one could describe the problem of determining which wffs apply to a focus node N as consisting of the following steps:

1. Calculating all the nodes Ni such that N $\rightsquigarrow$ Ni

2. Taking the union of the wffs at N, the wffs at all Ni calculated above, and the background information, and determining if this set is consistent.

3. If this union is not consistent, choosing a maximally consistent subset of wffs(N) $\cup \bigcup_{N\rightsquigarrow Ni}$ wffs(Ni). More precisely, we choose a subset S that satisfies the following conditions:
a. $S \subset$ (wffs(N) $\cup \bigcup_{N\rightsquigarrow Ni}$ wffs(Ni)).
b. $S \cup \mathcal{B}$ is consistent.
c. S is the largest such subset; that is, there does not exist S' satisfying conditions a. and b. such that $S' \subset S$.
We say that S is a maximally consistent subset of wffs(N) $\cup \bigcup_{N\rightsquigarrow Ni}$ wffs(Ni) with respect to (wrt) $\mathcal{B}$. If the meaning is clear, we may omit the reference to $\mathcal{B}$.

We make two brief remarks about the second step:

First, deciding whether a set of rules is consistent is only semi-decidable in general; however, we can restrict our attention to certain subsets of sentences such as those without existential quantifiers or self-embedding function symbols. Such sets are decidable. Second, in decidable cases, deciding whether or not a set of sentences is consistent is intractable; we discuss ways of handling this problem in Section 3.3.

Most of the conceptual difficulties arise in the second task: choosing a maximally consistent subset. In general, there is more than one maximally consistent subset of rules. The question is: which maximally consistent set should we choose? Since each maximally consistent subset is formed by deleting some of the wffs in an inconsistent set, an alternate phrasing of this question is: which wffs should we delete? That is, how do we decide that some wffs are more important than others? The general strategy is to articulate some *preference principles* for sets of wffs and to choose maximally consistent subsets in accordance with these principles.

The particular strategy developed in this paper is to examine preference principles that are based on the structure of the inheritance network. That is, we exploit as much as possible the structure of the inheritance network while we are constructing and choosing maximally consistent subsets. We focus on specificity and multiple paths, both non-conflicting and conflicting. We turn to several examples to illustrate this approach.

Two remarks on these examples: First, some examples deal with paths of length 1; however, we generalize to paths of arbitrary length in the discussion and specification of the wff-inheritance procedure. Second, to simplify and shorten the exposition, we have chosen very simple wffs; the inconsistencies are typically arithmetic in nature. As noted in Sections 1 and 2, rules are typically much more complex, and contradictions between these complex wffs are rampant.

3.2.1 Specificity

Consider the example in Figure 3. HGH (Human Growth Hormone) $\rightarrow$ $\mathrm{R_x}$ Drugs. Suppose we have the following wffs at these nodes: (All variables are assumed to be universally quantified unless otherwise specified.)
Drugs:
{ copay-pct(x) = 10
(Non-network(p) $\wedge$ filled(x,p)) $\supset$ penalty(x) = \$15}
(There is a 10% copay for all drugs and there is a \$15 penalty for prescriptions if they are filled at non-network pharmacies.)
HGH:
{ copay-pct(x) = 50 }
(The copay for HGH drugs is 50%).
If we assume that the background context $\mathcal{B}$ entails some basic arithmetic facts such as
(v1 $\neq$ v2) $\supset$
(¬ (copay-pct(x) = v1 $\wedge$ copay-pct(x) = v2)) —
that is, a service cannot have two co-pay percentages, then the union of these two sets is inconsistent with respect to $\mathcal{B}$ [4].

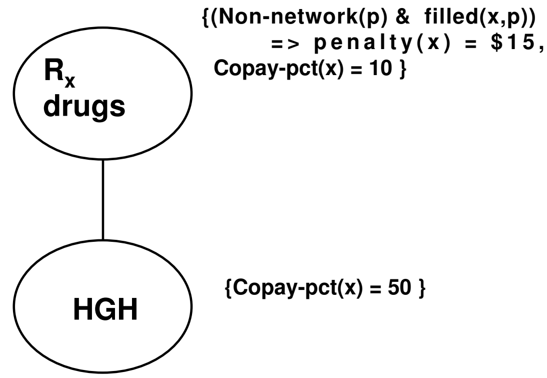

Figure 3: *The wffs at HGH are more specific than the wffs at* $\mathrm{R_x}$ *Drugs*

Obviously, there are two maximally consistent subsets:
1. The set of wffs at the Drugs node, namely:
{ copay-pct(x) = 10
(Non-network(p) $\wedge$ filled(x,p)) $\supset$ penalty(x) = \$15 }
and
2. { copay-pct(x) = 50
(Non-network(p) $\wedge$ filled(x,p)) $\supset$ penalty(x) = \$15 }

The choice of which maximally consistent subset to prefer is clear. The HGH node is more specific than the Drugs node; thus we prefer the subset that has the wff from the HGH node to the subset that has the wff from the Drugs node. That is, we prefer subset 2 to subset 1.

We say that subset 2 is a *preferred maximally consistent subset (pmcs)* relative to the sets of wffs at HGH and wffs at Drugs, with the set of wffs at HGH preferred over the set of wffs at Drugs. We define this concept formally below:

[4] The actual network contains the following cost-share rule at the Prescription Drugs node: copay(x) = \$3 (There is a \$3 copay for all prescription drugs). The union of the two sets is then not, strictly speaking, inconsistent, as long as HGH drugs cost \$6. In other words, the union of these sets entail that HGH drugs cost \$6. Detecting unreasonable consequences such as this is an interesting problem in its own right, but beyond the scope of this paper. Presumably, there are several ways to deal with this problem: one can include price tables in the background information, or include general principles forbidding ridiculous consequences in the background information. The difficulty in the latter strategy then is a matter of expressing such principles and ensuring that we have got them all. (I am grateful to Ernie Davis for pointing out this problem.)

Def. (Preferred Maximally Consistent Subset): *Assume that S1, S2 are sets such that S1 $\cup$ S2 is inconsistent with respect to $\mathcal{B}$. Let X1 and X2 be maximally consistent subsets of S1 $\cup$ S2 wrt $\mathcal{B}$. Let R1 be a subset of S1; let R2 be a subset of S2, such that R1 = X1 - X2; R2 = X2 - X1. That is, R1 and R2 are all that distinguish X1 from X2. If this is the case, we say that X1 $\cong$ X2 with respect to R1,R2. Then if S1 is preferred to S2, X1 is a preferred maximally consistent subset of S1 $\cup$ S2. Otherwise, if S1 $\cup$ S2 is consistent with respect to $\mathcal{B}$, S1 $\cup$ S2 is a (in this case* the*) preferred maximally consistent subset of S1 and S2.*

Note that there is not necessarily a unique preferred maximally consistent subset. We introduce the function PMCS where PMCS(S1,S2) returns the *set* of all preferred maximally consistent subsets of S1 and S2, with S1 preferred to S2. By a slight abuse of notation, we will use the notation pmcs(S1,S2) to mean the (random or nondeterministic) function that returns one of the elements of PMCS(S1,S2). In addition, we extend pmcs to n sets, so that pmcs(S1, ..., Sn) = pmcs(pmcs(S1, ..., Sn-1), Sn).

3.2.2 Multiple Paths (Non-conflicting)

Specificity is clearly not the only criterion one can use in determining a preferred maximally consistent subset. Consider again Figure 1. In this figure, the node Maternity Surgical has links to both Maternity and Surgical and thus inherits wffs from both nodes. The union of the wffs at these nodes { copay-pct(x) = 10, copay-pct(x) = 20} is obviously inconsistent with respect to the background constraint mentioned above. There are obviously two maximally consistent subsets:
{copay-pct(x) = 10}
{copay-pct(x) = 20}.
Although specificity does not help here — Maternity is neither more nor less specific than Surgical — it is clear which subset we should prefer. Since the link between Maternity Surgical and Maternity has preference over the link between Maternity Surgical and Surgical, it is reasonable to prefer the first subset, since this keeps the wff of the Maternity node. Thus, one can compute the rules that apply to Maternity by computing pmcs(wffs(Maternity), wffs(Surgical)). In fact, if there were wffs at Maternity Surgical as well, these wffs might contradict either the wffs at Maternity, the wffs at Surgical, or both. Since the wffs at more specific nodes are preferred, the computation of the rules that apply to Maternity Surgical would then be
pmcs(wffs(Maternity Surgical),
pmcs(wffs(Maternity), wffs(Surgical))).

3.2.3 A Procedure for General Paths

The examples discussed thus far have dealt only with paths of length 1. Even in such simple cases, however, one may need to consider wffs at more than 2 nodes; the order in which one takes preferred maximally consistent subsets is obviously crucial. We now discuss this issue for more general paths.

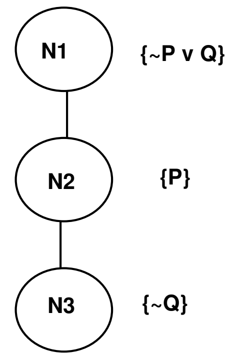

Figure 4: *Doing a simple downward traversal will give the wrong answer for wff-inheritance*

Consider first a node N in a network where there is one path from N to a root node (and thus, no forking points), and consider how to compute Ψ(N). It is clear what we do *not* want to do. We do not want to first take the union of all sets of wffs at the nodes on the path from N to the root, then take maximally consistent subsets of this large set, and finally choose preferred maximally consistent subsets relative to the specificity criterion. It is hard to beat this method for inefficiency. At the very least, the procedure to compute Ψ(N) ought to iteratively traverse the path, computing Ψ(N) as one goes along.

Upward vs. Downward Traversal: The obvious question to ask is whether to traverse the path upward or downward. A downward traversal might seem natural for the following reasons: First, a downward traversal emphasizes the natural analogy between wff-inheritance and (iterated) belief revision (Gardenfors, 1988; Darwiche and Pearl, 1994). Specifically, one might think of inheriting wffs as the process of revising the beliefs at a more general node by the wffs at a more specific node; this translates to the downward traversal of a path in the network. Second, one might expect downward traversal to have the side benefit of computing the sets of wffs that apply to intermediate nodes on the path.

However, straightforward downward traversal — a simple recursive procedure in which one sets Ψ(root) = wffs(root), and for each non-root node x, Ψ(x) = pmcs(wffs(x), Ψ(parent(x)) — turns out to be incorrect. Consider computing Ψ(N3) in Figure 4. It is clear that the expected answer is {P, $\neg$ Q }; the union of all wff sets is inconsistent, and one discards the wff at N1 since that is the least specific. However, if one traverses the path downward, one gets
Ψ(N1) = {P $\supset$ Q},
Ψ(N2) = {P $\supset$ Q, P}.

When one reaches N3, one gets
$\Psi(N3) = pmcs(\{\neg Q\}, \{P \supset Q, P\})$.
But there are two preferred maximally consistent subsets: $\{P, \neg Q\}$ and $\{P \supset Q, \neg Q\}$ — contrary to expectations. The problem is that during the downward traversal, we have lost the information that P comes from a node that is more specific than the node from which $P \supset Q$ comes.

There are clearly ways to fix this problem: we can, for example, keep track of the source node of each of the wffs that has been collected so far.

If we wish to avoid this sort of bookkeeping, however, the simplest method for computing $\Psi(N)$ that is consistent with the specificity constraint is based on an upward traversal of the network. Specifically, one begins at the focus node N, taking wffs(N) as the starting set. One then proceeds up the path, at each node taking a preferred maximally consistent subset of the set computed so far and the wffs at the current node. This process will ensure that the specificity constraint is obeyed.

We must also ensure that path ordering is respected in case of forking paths. This is done by examining all links at each point in the path, ordering them, and recursively proceeding up the more preferred links before the less preferred links.

In addition, we must ensure that we do not collect rules from nodes that are only on conflicted or preempted paths. The simplest way to avoid this problem is to preprocess the FAN to remove the preempted and conflicted links. We may do this by using an extension/modification of the procedure given in (Stein, 1992), which computes the *specificity extension* at a focus node.

We thus have the following procedure to compute $\Psi(N)$ in a FAN:

```
PROCEDURE COMPUTE Ψ(N)
Preprocess the FAN to remove
  preempted, conflicted links
Ψ(N) := wffs(N)
FAN-traverse(N)

PROCEDURE FAN-TRAVERSE(x)
if x has not been visited
 mark x as visited
 Ψ(N) := pmcs(Ψ(N),wffs(x))
 If x is a leaf
  then return
  else
   Determine all nodes yi such that x →yi
   Do topological sort of all k links x →yi
   for i = 1 to k do
    FAN-traverse(yi)
```

Note that since we are traversing a dag, as opposed to a tree, some nodes may be visited twice. To avoid re-computing preferred maximally consistent subsets (an expensive operation), we mark nodes as we visit them.

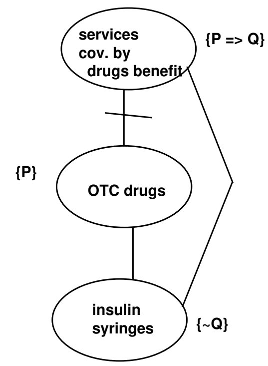

Figure 5: *Which node gets preference? Does preemption make a difference?*

It will be noted that the computation of $\Psi(N)$ does not necessarily produce a unique set. This is because, as noted, the function pmcs does not yield a unique preferred maximally consistent subset. The existence of multiple maximally consistent subsets has always been something of a problem for nonmonotonic theories. In practice, however, the existence of multiple preferred maximally consistent subsets is not at all problematic for this domain. Indeed, it is perfectly legitimate to reason with any preferred maximally consistent subset of wffs that obeys the criteria of specificity and path preference. (In particular, members who are insured by some product often "win a case" by demonstrating that there is a coherent argument — that, is a preferred maximally consistent set of business rules — that supports their claim.) This phenomenon is at odds with the distaste that AI researchers usually have for credulous reasoning, but is worthy of further investigation.

3.2.4 An Open Question: The Case of Preempting Paths

Consider now — in contrast to the previous examples — inheriting wffs in a network with conflicting multiple paths. Figure 5 shows a portion of the drugs network. Although OTC Drugs (over-the-counter drugs) is a subtype of Drugs, it is not a subtype of the class of services covered by the Drugs benefit. On the other hand, a subtype of OTC Drugs, Insulin Syringes, is covered by the Drugs benefit. This is a classic case of preemption. In fact, this case is known as positive preemption, since a positive path — the direct link from Insulin Syringes to Services Covered by Drugs Benefit — preempts a negative path — the path from Insulin Syringes to OTC Drugs to Services Covered by Drugs Benefit.

Clearly, there are positive undefeated paths from Insulin Syringes to Services Covered by Drugs Benefit and from Insulin Syringes to OTC Drugs. Thus, In-

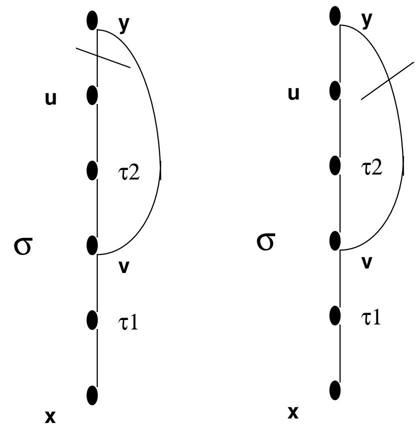

Figure 6:
a) positive path preempts negative path
b) negative path preempts positive path

sulin Syringes stands to inherit wffs from both nodes. The question is: suppose the set of wffs at OTC Drugs is inconsistent with the set of wffs at Services Covered by Drugs Benefit (either absolutely, or with respect to the background information and/or the set of wffs at Insulin Syringes). Which set of wffs do we prefer? Do we prefer the wffs from Services Covered by Drugs Benefit because that node is on the preempting path? Or do we refrain from preferring either of the paths?

There are arguments both for and against preferring nodes from the preempting path. To formally state the arguments, we use Horty's (1994) notation for preempting paths. A positive path $\pi(x, \sigma, u) \rightarrow y$ is *preempted* ((in the context (Γ, Φ)) iff there is a node v such that (i) either $v = x$ or there is a path of the form $\pi(x, \tau 1, v, \tau 2, u) \in \Phi$, and (ii) $v \not\rightarrow y \in \Gamma$. A negative path $\pi(x, \sigma, u) \not\rightarrow y$ is preempted (in the context (Γ, Φ)) iff there is a node v such that (i) either $v = x$ or there is a path of the form $\pi(x, \tau 1, v, \tau 2, u) \in \Phi$, and (ii) $v \rightarrow y \in \Gamma$. (See Figure 6.) The question is: Is the path $\pi(v, y)$ preferred to the path $\pi(v, \tau 2, u)$?

The argument for preferring a positive preempting path over the path it preempts runs as follows: Clearly, the positive path $\pi(v, y)$ is in some sense preferred to the negative path $\pi'(v, \tau 2, u, y)$ (that is the reason that we conclude y). Thus, preemption seems to offer some evidence that the preempting path is stronger than the preempted path. Then presumably, one of the links in the negative path $\pi'(v, \tau 2, u, y)$ is not as strong as the positive path between v and y. It is possible that the weak link is the negative link between u and y — but this is by no means definite, because we do, after all, conclude that u's are not y's. Thus, it seems as likely that one of the links in the path between v and u is weak. (Indeed, we know that at least in one respect v's are not typical u's — unlike typical u's, they are y's.) Thus, we ought to prefer $\pi(v, y)$ to $\pi(v, \tau 2, u)$.

On the other hand, the argument for *not* preferring a positive preempting path over the path it preempts rests on the fact that there is a positive undefeated path between v and u in the same way that there is a positive undefeated path between v and y. Thus, they both should have the same status; neither should be preferred.

If one does decide to prefer preempting over preempted paths, another question arises. Consider paths forking off from nodes in preempted paths (e.g., consider another link from u to some node z.) Ought they also to have lower priority than preempting paths? The rationale given for preferring preempting to preempted paths would seem to hold here as well, but the intuition becomes increasingly weak.

Indeed, the lack of examples makes honing intuitions particularly difficult. There are few examples in this domain of positive preemption [5] and fewer still where there are conflicting wff sets in these portions of the network; thus, it is difficult to guess what the correct behavior ought to be.

The procedure to compute the wffs that apply to a node, outlined in the previous section, takes a neutral stand on the issue. It does not state a preference for preempting paths over the paths that are preempted, but it allows that preference to be incorporated. One can incorporate such a preference in two ways: First, one can place a condition on $\mathcal{O}$ stating that preempting paths are always strictly greater than their associated preempted paths. (One may in fact wish to insist that no paths come between the preempting and preempted path.) Second, one can modify the procedure so that, when one identifies the forking paths at a fork point, one groups preempting paths with the paths they preempt, and subsequently collects wffs along preempting paths before collecting wffs along the associated preempted paths.

The difficulty is not a technical one. The hard part here is getting the intuition right. The careful examination of other domains should shed some light on the question.

3.2.5 Other Preference Criteria

The computation of $\Psi(N)$ has thus far focussed on using information present in the structure of the network. Other standard preference principles may also be needed here. One may wish to assign some wffs a higher priority than others (as in (McCarthy, 1986)), regardless of the rule's position in the network. For ex-

[5] The problem does not arise at all in cases of negative preemption — when a negative path preempts a positive path. That is because one never inherits from the last node of a negative path. In Figure 6b, one would never wish to inherit from node y — and inheriting along the path from node x to node u is simply standard wff-inheritance using the specificity criterion.

ample, it might always be the case that medical rules have higher priorities than administrative rules. Likewise, it is also reasonable to prefer a particular subset of rules based on the results that this subset entails. This is equivalent to preferring one extension, or model, over the other (as in (Shoham, 1988)). For example, we may prefer extensions in which a claim gets paid to one in which the claim does not get paid.

3.3 COMPUTATIONAL ISSUES

Inheriting rules immediately transforms the problem of inheritance from a tractable problem (at least in the case of upwards inheritance: see (Selman and Levesque, 1993)) to one that is badly intractable. Wff-inheritance in this domain will often be done in a static setting as opposed to a dynamic one, [6] and this takes some of the sting out of the fact that computing preferred maximally consistent subsets is NP-hard. Nevertheless, intractability is clearly an unpleasant issue that we must handle in some form.

In practice, we have discovered that we can deal with the complexity issue by using a divide-and-conquer strategy. The trick is to divide the set of wffs into k types, subject to the following constraint:
If Wffs(i,N1) $\cup$ *Wffs(j,N2) is inconsistent wrt* $\mathcal{B}$*, then* $i = j$.[7]
That is, wffs are constrained so that sets can contradict one another only within their own type. This cuts down on much consistency checking (since often when N1 $\rightsquigarrow$ N2, the wff sets at N1 are of a different type than the wff sets at N2) and greatly reduces the time needed for consistency checking and maximal subset construction and choice. Obviously, the greater k is, the more this strategy helps.

Finding a division of the wffs into sets that satisfy this constraint is not a trivial task. The wffs in this domain are taken from business rules that were originally categorized by employees of the insurance corporation into cost-share, access, administrative, and medical rules. Certain of these groups had the desired property — e.g., cost-share rules did not contradict rules of any other type — but others did not: medical rules often contradict administrative rules. On the other hand, it was possible to identify some types of administrative rules that were consistent with all rules outside of their own type. In any case, it is hardly surprising that a categorization of rules developed from a business viewpoint is not optimal in terms of efficiency.

Finding the proper knowledge representation is clearly very important. Doing so is no longer only of academic interest; it can greatly affect the tractability of the system.

[6] Since the rules that apply to a medical service can be computed off-line. On the other hand, employees of the insurance corporation who modify the network may wish to see in real time how adding or changing rules at points in the network changes the rules that apply to a service.

[7] The notation Wffs(i,N) should be read: the set of wffs at node N of type i.

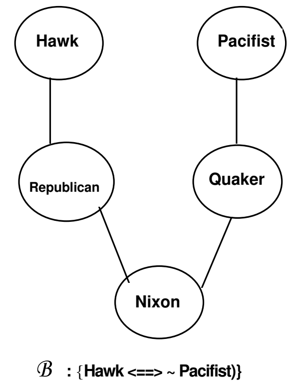

Figure 7: *The modified Nixon diamond is inconsistent, but only implicitly.*

3.4 BACK TO ATTRIBUTE INHERITANCE

We first noted the need to do conflict resolution and recognition for the purposes of wff inheritance. Further reflection, however, suggests that similar problems can arise even when performing standard attribute inheritance. Consider again Figure 1. Maternity Surgical $\rightsquigarrow$ Covered by Surgical Benefit; Maternity Surgical $\rightsquigarrow$ Covered by Maternity Benefit.

In fact, however, Maternity Surgical cannot be covered by both benefits: there is a constraint that services are covered by at most one benefit. This constraint is not explicit in the structure of the network; it is entailed by background domain knowledge; i.e., the background $\mathcal{B}$. The point of this example is not in enforcing this constraint — implementation is quite simple — but in recognizing the inconsistency. In general, when some of the knowledge of the inheritance network is present as background knowledge, inheriting attributes from multiple parents has the potential for leading to inconsistency, even if this is not explicit.

Oddly, this problem has not been discussed in the inheritance literature, perhaps because inconsistencies in the examples used have always been explicit. The Nixon diamond example (Reiter and Criscuolo, 1981) is a classic example of such an explicit inconsistency. Suppose, however, we modify Touretzky's (1986) modification of this example. (See Figure 7.) (Nixon $\rightarrow$Republican; Nixon $\rightarrow$Quaker; Quaker $\rightarrow$Pacifist; Republican $\rightarrow$Hawk). There is no explicit contradiction between Hawk and Pacifist, but if we add a statement to the background theory stating that the two concepts are contradictory, *Hawk* $\equiv \neg$ *Pacifist*, then there is an inconsistency that must be resolved. Depending on the amount and form of the background knowledge, detecting and resolving this inconsistency can be arbitrarily difficult (that is, as hard as the problem of wff-inheritance).

4 DISCUSSION AND RELATED WORK

4.1 COMPARING FANS TO OTHER KNOWLEDGE STRUCTURES

FANs are standard inheritance networks with wffs attached to nodes and in the background. The trend away from ontological promiscuity (the wanton creation of new knowledge structures) prompts the questions: In what way are FANs different from existing knowledge structures? Are they truly necessary or can we get by with more familiar mechanisms? This section argues that they are particularly natural for the domain at hand, and that existing structures have their own pitfalls.

FANs are extremely natural for any commercial domain in which general taxonomic information is best viewed separately from the bulk of the domain's business rules. This is precisely the situation in the medical insurance domain. The taxonomic information represents the structure of medical services by the medical profession (as well as the broad benefit categories that are standard in the insurance industry). This is a relatively static structure across products and corporations. The wffs represent the industry's business rules which change rapidly, across products and corporations. In fact, the network has proved to be especially easy for customer service representatives to master. At the same time, representing the business rules as wffs at nodes has kept the size of the network manageable (250–300 nodes).

We now examine alternatives to FANs. There are two obvious choices: representing everything in an inheritance network, or doing this entirely in a nonmonotonic logic.

Alternative I: Putting everything in the inheritance network.

As mentioned in the introduction, only some wffs can be easily represented as links between classes and attributes. However, we can capture all wffs simply by reifying each wff as a node. For example, a node $\forall x \phi(x)$ could be reified as the node representing the property of satisfying ϕ. There are several objections to this alternative.

First, many such nodes would be very unnatural, (consider the sample wffs in the first 2 sections) and quite outside the spirit of a semantic network. That is, nodes in the network, instead of representing concrete entities such as services and benefits, would represent ontologically dubious creatures.

Second, the problem of wff-inheritance has now been recast as the problem of determining if two or more paths conflict. In particular, assume wffs $\phi_1 \ldots \phi_n$ such that $\phi_1 \wedge \ldots \wedge \phi_n$ are mutually inconsistent. Let $y_1 \ldots y_n$ be the nodes associated with these wffs. It is clear that the paths $x \rightsquigarrow y_1 \ldots x \rightsquigarrow y_n$ conflict in the intuitive sense of the term. But this conflict cannot even be expressed directly in a standard Hortian or Touretzkian framework. Even if we increase expressivity, there is no obvious way to recognize and resolve this conflict. In contrast, determining maximally consistent subsets of wffs is a well-understood problem. [8]

Alternative II: Doing everything in a nonmonotonic theory That is, translate the inheritance network into a nonmonotonic theory, such as autoepistemic logic (Moore, 1985) or circumscription (McCarthy, 1980). [9]

There are several objections to this strategy. First, the distinction between static taxonomic rules and more volatile business rules has been erased. Second, this representation is much less accessible to users of the expert system, who find browsing through an inheritance network easy to learn and understand. Either they must live with this hardship or we must keep the FAN *as well as* the nonmonotonic theory. We must then either construct a mapping between the two — a non-trivial task — or be sure to always update the two structures simultaneously, an unrealistic constraint in the business world. Third, such a strategy may be computationally wasteful in that in translational theories, one must generally recompute priorities and/or specificities in a more tedious manner than the algorithms available for determining specificity in an inheritance network.

4.2 RELATED WORK

There is much work in the literature on determining maximally consistent sets of defaults (Brewka, 1989, Geffner, 1990, Grosof, 1991, Delgrande and Schaub, 1994, among others). Such work is obviously very close in spirit to ours. In particular, there is an emphasis in

[8] As mentioned in the previous section, the general problem of determining if paths conflict already exists in many networks, such as the modified Nixon diamond of Figure 7. While we see that these problems can beset any network, we nevertheless believe that flooding networks with this problem, which would happen if wffs were reified as nodes, is not desirable. This is particularly true since the problem of determining if arbitrary paths in a network are inconsistent with respect to one another has not been addressed.

[9] There are a variety of methods for doing this such as (Gelfond, 1990). However, see also Horty (1994), who argues that specifying a translational semantics for a path-based inheritance network is a problem that is not yet completely solved. In any case, it is certainly possible, given a *particular* network, to give an equivalent nonmonotonic theory.

these works on considerations of specificity. This paper differs from these works in a number of respects (such as the consideration of non-conflicting-path multiple inheritance). Most importantly, what distinguishes this paper from these works is the emphasis on the semantic network structure which we take as our starting point. The other works take some nonmonotonic system as their starting point. The difference is analogous to the distinction that Horty (1994) points out between path-based and translational theories of inheritance; the latter specify the meaning of a network in terms of a nonmonotonic formalism; the former specify the meaning in terms of the paths themselves. Just as Horty argued for the naturalness and intuitiveness of path-based approaches for standard inheritance, we argue for the ease, naturalness, and intuitiveness of the path-based approach to wff-inheritance. In particular, note the discussion in (Horty, 1994), which argues that researchers using the translational approaches have not yet satisfactorily formalized the concept of specificity, which is easy to formalize within a path-based approach.

We do not, however, believe that these systems should be seen as competing. Rather, we believe that the following is the case: A major task in knowledge representation is the articulation of the structure that is most natural and useful for a particular application. Another major task is demonstrating, if possible, that this structure is equivalent to more familiar structures, or if that is not possible, demonstrating the inequivalence. In this spirit, we suggest that formula-augmented semantic networks have a useful role to play in domains in which taxonomic knowledge is a major, but not sole, component of the domain knowledge. (We are currently using FANs in life insurance and property & casualty applications). At the same time, we hope that future research will shed light on the equivalences between FANs and the systems of Geffner, Grosof, and others.

Acknowledgements: Thanks to Ernie Davis, David Etherington, Benjamin Grosof, Daniel Lehmann, Moninder Singh, and Rich Thomason for helpful discussions and suggestions, and to Bill Wynne for his assistance in populating the benefits semantic network.

REFERENCES

G. Brewka (1989): Preferred Subtheories: An Extended Logical Framework for Default Reasoning, *IJCAI 1989*, 1043-1048

A. Darwiche and J. Pearl (1994): On the Logic of Iterated Belief Revision, in R. Fagin (ed): *Proceedings of the Fifth Conference of Theoretical Aspects of Reasoning About Knowledge*, Morgan Kaufmann, San Mateo, 5-23

J. Delgrande and T. Schaub: A General Approach to Specificity in Default Reasoning, *KR 94*, 146-157

D. Etherington(1988): *Reasoning with Incomplete Information*, Morgan Kaufmann, Los Altos

P. Gardenfors (1988): *Knowledge in Flux*, MIT Press, Cambridge

H. Geffner (1990): *Default Reasoning: Casual And Conditional Theories*, MIT Press, Cambridge

M. Gelfond and H. Prszymusinska (1990): Formalization of Inheritance Reasoning in Autoepistemic Logic, *Fundamenta Informaticae 13*, 1990, 403-443

B. Grosof (1991): Generalizing Prioritization, in J. Allen, R. Fikes, and E. Sandewall (eds): *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann, San Mateo, 289-300

J. Horty (1994): Some Direct Theories of Nonmonotonic Inheritance in D. Gabbay, C. Hogger, and J. Robinson, eds: *Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. 3: Nonmonotonic Reasoning and Uncertain Reasoning*, Oxford University Press, Oxford, 111-187

J. Horty, R. Thomason and D. Touretzky (1990): A Skeptical Theory of Inheritance in Nonmonotonic Semantic Networks, *Artificial Intelligence 42*, 311-349

H. Kautz and B. Selman (1989): Hard Problems for Simple Default Logics in R. Brachman, H. Levesque, and R. Reiter (eds): *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, Morgan Kaufman, San Mateo, 189-197

J. McCarthy (1980): Circumscription – A Form of Non-Monotonic Reasoning, *Artificial Intelligence 13*, 27-39

J. McCarthy (1986): Applications of Circumscription to Formalizing Common-sense Knowledge, *Artificial Intelligence 28*, 86-116

R. Moore (1985): Semantical Considerations on Nonmonotonic Logic, *Artificial Intelligence*, 25(1), 75-94

R. Reiter and G. Criscuolo (1981): On Interacting Defaults, *IJCAI 1981*, 270-276

B. Selman and H. Levesque (1993): The Complexity of Path-Based Defeasible Inheritance, *Artificial Intelligence 62:2*, 303-340

Y. Shoham (1988): *Reasoning About Change: Time and Causation from the Standpoint of Artificial Intelligence*, MIT Press, Cambridge

L. Stein (1992): Resolving Ambiguity in Nonmonotonic Inheritance Hierarchies, *Artificial Intelligence 55*, 259-310

D. Touretzky (1986): *The Mathematics of Inheritance Systems*, Morgan Kaufmann, Los Altos