

New Problems for Inheritance Theories

Leora Morgenstern

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

Abstract

Research in the area of inheritance has generally focussed on a small set of toy problems. This paper discusses several problems, encountered while developing a commercial application using an inheritance network, that have previously received little or no attention. These problems include inheriting rules, non-unary inheritance, and inheriting from subevents.

1 Introduction

Inheritance has long been recognized as prevalent in commonsense reasoning. It has been advocated as a practical means of implementing some common patterns of logical reasoning since the early days of AI (Quillian, 1968). Progress in inheritance has generally moved along one of two directions. On the one hand, researchers have developed general implementations of basic inheritance without exceptions (Schmolze and Lipkis, 1983; Mays et. al, 1991), concentrating on issues such as classification algorithms. On the other hand, researchers have extended the basic framework of inheritance to inheritance with exceptions and multiple inheritance. This research has focussed on the skeptical vs. credulous inheritance debate (Horty et. al, 1990), various implementations of the specificity criterion (Touretzky, 1986; Horty et. al, 1990), and complexity results for inheritance algorithms (Selman and Levesque, 1993).

For the most part, this second line of research has remained confined to theoretical, academic circles. Few non-trivial implementations exist.¹ Moreover, the benchmark problems that exist are all toy problems, giving the impression that problems such as inheritance with exceptions and multiple inheritance occur rarely in real applications. Indeed, this view has been explicitly expressed by some of the leaders in the field.² As if often the case when research in a subfield concentrates on a small set of toy problems, inheritance research sometimes appears both sterile and irrelevant.

¹The work of Evett et. al(1994) is a notable exception. However, it is worth remarking that this implementation is only an approximation to Touretzky's (1986) algorithm; for efficiency reasons, arbitrary cancellation links are not allowed.

²Horty, personal communication.

Given our intuition that some form of inheritance underlies much of commonsense reasoning, this state of affairs is somewhat puzzling. It would be expected that the careful examination of a real-world problem whose domain knowledge could naturally be represented in an inheritance network would yield several interesting non-toy problems. This paper is the result of such an examination into a commercial domain: the development of a benefit inquiry system in a medical insurance corporation.

We discuss the domain problem, describe the framework that we developed, and discuss several problems that arise when reasoning in such a network. We propose (partial) solutions to some of these problems. The purpose of this paper, however, is not primarily to provide solutions to a set of new problems. Rather, we wish to open up these problems for discussion and research. In particular, we suggest that these problems become new benchmark problems for inheritance theories, in the spirit of (Lifschitz, 1988).

This paper is structured as follows: We begin with a brief description of the benefit inquiry system. In sections 3-5, we discuss three problems for inheritance theories: inheriting rules, non-unary inheritance, and inheriting from subevents. In Section 6, we explore the semantics of rules inheritance. Finally, we discuss the applicability of these problems to other commonsense reasoning domains in which inheritance is common.

2 The Benefits Network

2.1 Domain Background

The inheritance problems discussed in this paper were discovered as a result of the development of a benefit inquiry system for a large medical insurance corporation. The medical insurance corporation has millions of members, the vast majority of whom buy insurance in groups. The company offers its members a choice of *products*. A product is a collection of benefits, services, and rules, which are related in the following manner:

A service may be *covered* 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 benefit 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?" "Can the surgery be done in an inpatient hospital?" "What will the deductible be?" "Is there a separate deductible for the anesthesiologist?" "How many days can I stay in the hospital after a standard delivery?"

An inheritance hierarchy is a natural choice to represent the underlying knowledge for this domain, since so much of the information is taxonomic in nature. For example, TMJ therapy and spinal manipulation are both types of physical therapy; physical therapy, speech therapy, and occupational therapy are all types of therapy, which is a sort of professional service. As we will discuss in Section 3, rules that apply to a node may also be inherited by its subtypes.

2.2 The Inheritance Network

Note on Methodology:

Theories of inheritance networks are either *translational* or *direct* (Horty, 1994). Translational theories, such as (Etherington and Reiter, 1983; Etherington, 1988, Ch. 4; Lifschitz, 1994, Sec. 4), specify the meaning of a network formalism in terms of a non-monotonic logic such as Default Logic (Reiter, 1980) or Circumscription (McCarthy, 1980). Direct theories, such as (Touretzky, 1986; Horty et. al. 1990; Horty, 1994), analyze and characterize interesting features and properties of semantic networks, such as consistency, and the sets of conclusions that can be derived from the premises, in terms of the network formalism itself. The focus is on paths in the network. The direct approach does not provide a model-theoretic semantics.

We follow the direct, path-based approach in the bulk of this paper. In Section 6, we suggest a translational semantics for some of the work done in previous sections of this paper.

Definition of the Network:

The semantic network consists of a set of nodes, a set of links, and a context.

Nodes A node consists of two parts: a label and a (possibly empty) set of rules or well-formed formulas in a first-order logical language. If a node $N = (L, R)$, where L is the label of the node and R is the set of rules, we will sometimes refer to R as *the set of rules at N* or *rules(N)*. If r is a member of R , we say *at(r, N)*.

The *label* of a node denotes some set of medical services. Examples of labels are Surgical, which denotes the set of surgical services and Anesthesia, which denotes the set of anesthesia services. Medical services may be identified as professional procedure services, condition services, or setting services. Professional procedure services, intuitively, are those services that are characterized by the medical professional who delivers it (e.g., surgery and physical therapy). Condi-

tion services are those services that are characterized by a particular condition (e.g., maternity and mental health). Setting services are those services that are directly linked to the setting in which they are offered: e.g., inpatient ancillary services or hospice services.

When we reason about a set of medical services, we can usually classify that set as consisting of either professional, condition, or setting services. However, this is not always the case; the classes are not disjoint. Indeed, we will often need to reason about services that are best characterized by two or more of these service types. For example (Section 4), we reason about physical therapy in an inpatient setting: this set of medical services belongs to both professional procedure services and setting services.

Services may also be classified by their coverage; examples are Services Covered by Surgical Benefit and Services Covered by Maternity Benefit (see Figure 1)³.

The set of rules R consists of well-formed formulas chosen from the first-order predicate calculus. We impose the following restriction: any rule r must be a wff ψ that can be expressed as $\forall x\phi(x)$, where ϕ is a wff that is free in x . An examples of a rule is:

$\forall x (\text{copay}(x) \leq \$75)$ (the flat co-pay for services is no more than \$75).

The set of rules at a node is constrained to be consistent.

Context The *context* of a network is a (possibly empty) set of wffs of first-order logic. Intuitively, a context represents the background information that is true; for example, the context may include a wff which states that all medical services must be provided by licensed medical professionals in order to qualify for reimbursement.⁴ The set of rules at each node in a network is constrained to be consistent with the context of the network. We denote the context of a network as \mathcal{C} .

Links The most important types of links between nodes are the positive and negative is-a 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 $x \Rightarrow y$, $x \not\Rightarrow y$, $x \rightarrow y$, or $x \not\rightarrow y$, 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 .

³All figures can be found at the end of the paper.

⁴If we wish, we may think of the wffs in a context as being included in the set of rules at each node.

- 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 *compound paths*. For ease of presentation, this paper will be concerned almost exclusively with defeasible paths; extensions to strict and compound paths are straightforward⁵.

Often, there will be both positive and negative paths between two nodes. Following Touretzky(1986) and Horty(1994), we speak of the undefeated paths as those which are *constructible* and neither *pre-empted* nor *conflicted*. Intuitively, the undefeated 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 Hypnosis 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 pre-empted by the negative path. Thus, the negative path is undefeated.

In cases where there is an undefeated path from x to y and an undefeated path from x to z (where $y \neq z$), we can allow the prioritization of a particular path. In Figure 1, there is a (positive) path between Maternity Surgical and Services Covered by Maternity Benefit and a (positive) path between Maternity Surgical and Services Covered by Surgical Benefit. The number 1 denotes that the path between Maternity Surgical and Covered by Maternity Benefit has priority over the other path(s). We may also allow such prioritization in cases of conflicting paths. Doing so would entail altering the definition of a conflicted path.

Subevents A fifth type of link is known as the *subevent* link. It is natural to view a medical service or set of medical services as an *event*⁶, that is, the event in which that medical service is provided to some in-

dividual. For example, the set of surgical services can³ be viewed as the event of some individual undergoing a surgical procedure. By extension, since the label of a node denotes a set of medical services, it is natural to consider each node as representing some event.

If we view each node as an event, we may wish to talk about the decomposition of that node into subevents. We may be interested in a complete⁷ decomposition — e.g., the node Human Organ Transplant decomposes into the subevents of (organ) acquisition, (organ) transport, (recipient) preparation, and so on — or merely in a partial decomposition which gives us some subevents.

If x is a subevent of y , we write *subevent*(x,y). If we introduce the predicate *occurs*(n,I), which is true if the event represented by node n occurs during interval I , and the predicate *sub*($I1,I2$), which is true if $I1$ is a subinterval of $I2$, then we have the following axiom on subevent:

$$\text{Axiom: } \text{subevent}(x,y) \supset (\text{occurs}(y,I) \supset \exists J (\text{sub}(J,I) \wedge \text{occurs}(x,J)))$$

For any node y , we are particularly interested in a distinguished set of subevents: those subevents x for which both *subevent*(x,y) and $x \rightarrow y$. Anesthesia is an example. It is both the case that Anesthesia is a subevent of Surgical services and that Anesthesia services are a (defeasible) subtype of Surgical services. Thus, there is an undefeated positive path between Anesthesia and Surgical services and between Anesthesia and Services Covered by Surgical Benefit. Therefore Anesthesia inherits the property of being covered by the Surgical Benefit (and is also a candidate for inheriting the rules at Surgical, such as the cost-share rule declaring that the co-pay for Surgical services is 20%; see Section 3 for further details).

If $x \rightarrow y$ and *subevent*(x,y), we write $x \hookrightarrow y$.

It is important to note that we are maintaining the conceptual distinction between the is-a link and the subevent link. However, in contrast to most other networks in the literature, we do not insist that the relations corresponding to these links be disjoint. Generally, these links are disjoint in networks that contain both is-a and decomposition links. For example, Kautz’s (1991) cooking hierarchy distinguishes between those events that are *subtypes* of the event Make-Pasta-Dish (Make-Fettucini-Alfredo, Make-Spaghetti-Pesto, and Make-Spaghetti-Marinara are examples) and the *steps* of Make-Pasta-Dish (Make-Noodles, Make-Sauce, Boil). (In our notation, $\text{Make-Fettucini-Alfredo} \rightarrow \text{Make-Pasta-Dish}$, but *subevent*($\text{Make-Noodles}, \text{Make-Pasta-Dish}$)). Make-Noodles is *not* a subtype of Make-Pasta-Dish.

In contrast, in this network, there is a significant overlap between these links. It is not the case, however, that whenever x is a subevent of y then $x \rightarrow y$; we mention a common exception below. Moreover, we are

⁵Note 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.

⁶We follow McDermott’s (1982) ontology and understand an event, intuitively, to be the set of intervals in which that event takes place. Since events are just sets of intervals, they can be arbitrarily complex. In particular, sets of events are events.

⁷Informally, a decomposition $e_1 \dots e_n$ of an event e is complete if doing all of $e_1 \dots e_n$ in some specified order entails doing e .

not arguing that the is-a and subevent relations must overlap in other networks. Perhaps it is more common for these links to be disjoint. The fact that these relations overlap here is due to a feature specific to this network: many subevents, such as Anesthesia or (organ) Acquisition, are considered bona fide medical services.

We note an important connection between a subevent of a medical service and other descendants of that medical service. If x is a subevent of y , it is also a subevent of most descendants of y . For example, since Anesthesia is a subevent of Surgical, it is also a subevent of Endoscopic and of Orthopedic surgical services; likewise, (organ) Acquisition, a subevent of Human Organ Transplant, is a subevent of Heart (transplant) and Liver (transplant), which are subtypes of Human Organ Transplant. An important exception to this rule, of course, is other subevents: if $x_1 \hookrightarrow y$ and $x_2 \hookrightarrow y$, it is typically not the case that x_1 is a subevent of x_2 .

We thus have the following axiom:

$$\text{subevent}(x,y) \wedge z \rightsquigarrow y \wedge \neg \text{subevent}(z,y) \supset \text{subevent}(x,z)$$

Note that it is not the case that $x \hookrightarrow z$. Anesthesia is a subevent of Endoscopic surgical, but we would not want to say that Anesthesia \rightarrow Endoscopic surgical. In fact, it does not inherit any attributes (or rules) from Endoscopic surgical. Interestingly, Endoscopic surgical may inherit rules from Anesthesia. We discuss this further in Section 5.

3 Inheriting Rules

3.1 The Problem

Thus far we have described a semantic network with an underlying inheritance hierarchy that seems to model ordinary inheritance with exceptions. But there are several features of this network, alluded to in section 2, that take it beyond the standard inheritance network.

The most important feature that distinguishes this network is that rules are associated with nodes. Thus, we can ask the following question: What rules are *inherited* by a node in the network? Direct theories of inheritance traditionally focus on answering 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? For example, in the Clyde-Royal Elephant network (Touretzky, 1986), the question of interest is: does the grey attribute hold of royal elephants (i.e., are royal elephants grey)?

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 rules *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: $N \uparrow \psi$ if rule ψ pertains to node N . Overloading the \uparrow symbol, we say $N \uparrow \Psi$ if $\psi \in \Psi$ iff $N \uparrow \psi$. We define $\Psi(N) = \{\psi \mid N \uparrow \psi\}$. That is, $\Psi(N)$ is the set of rules that apply to node N . Then the question — and the new problem for inheritance theories — is: given a node N in the network described in Section 2, what is the set Ψ such that $N \uparrow \Psi$?

A naive approach might suggest the following:

$$\Psi(N) = \bigcup_{N \rightarrow N_i} \text{rules}(N_i) \cup \text{rules}(N).$$

That is, the set of rules that apply to a node N can be obtained by looking at the nodes to which there is an undefeated positive path from N , taking all the rule sets at these nodes and the rules at N , and forming the union of these sets. But such an approach is obviously wrong. Consider, for example, Figure 2. There is an undefeated positive path from $N3$ to $N2$. Thus, according to the naive approach, $\Psi(N3) = \text{rules}(N3) \cup \text{rules}(N2) = \{\forall x P(x), \forall x \neg Q(x)\} \cup \{\forall x R(x), \forall x(P(x) \supset Q(x))\}$. But this rule set is obviously inconsistent; we clearly do not want to blindly take unions of rule sets.

The problem arises because of two sources of non-monotonicity. First, there are defeasible links in the network. Second, the rules at a node are best thought of as *typically* true at a node, rather than rules that are *always* true at a node (see Section 6 for an elaboration of this theme and an attempt to make this notion more precise). If all paths in a network are strict, and for any node, all rules at that node are *always* true at that node, the problem of inconsistency described here could not arise.

Since inconsistency can arise, determining $\Psi(N)$ entails doing two things:

1. Deciding if the set of rules at that node is consistent with both the set of rules at the nodes to which there is an undefeated positive path and the background context.
2. If these sets are not consistent, choosing as many rules as possible which apply to the node. That is, determining a *maximally consistent subset* (Gardenfors, 1988) of the inconsistent set formed by the union of rule sets.

In section 3.2, we discuss possible solutions to these problems. First, we discuss some of the implications of this problem.

We are suggesting that all rules at nodes can be viewed as rules that are *typically* true at a node (that is, as non-monotonic rules), and that the problem of rules inheritance can be viewed, in part, as the problem of selecting a maximally consistent subset of rules from an inconsistent set of wffs. The question then arises: does the foregoing discussion demonstrate that the problem of rules inheritance is essentially the problem of belief revision, or the problem of constructing an extension in default logic, or some other non-monotonic formalism? Have we shown that inheritance, far from being a small subarea of non-monotonic logic, involves most of the issues facing non-monotonic logic? If so, perhaps

the problem of rules inheritance is not a new problem at all, but merely an old problem in a new guise.

Indeed, we believe that the existence of the rules inheritance problem does demonstrate that the general problem of inheritance involves many difficult issues that are present in general non-monotonic reasoning. Moreover, we have demonstrated the intrinsic difficulty of inheritance, not for a contrived toy example, but for a problem that occurs naturally in an applied commercial setting. (And, by extension, we have shown once again how prevalent non-monotonic reasoning is in realistic domains and in commercial applications.)

Nevertheless, it would be foolish to simply argue that the problem of rules inheritance is subsumed by the problem of non-monotonic logic and to subsequently forget about it. The rules inheritance problem is an interesting sub-problem of non-monotonic logic that deserves to be considered in its own right as a benchmark problem for inheritance theories. Specifically, in much the same way that standard (attribute) inheritance with exceptions uses specificity information present in the structure of a network to determine whether to prefer a positive or a negative path from one node to another, rules inheritance uses information present in the structure of a network to guide the choice of a maximally consistent subset of rules. We discuss this in more detail below.

3.2 Effecting Rules Inheritance

As we have argued in the previous section, determining which rules a node N1 inherits from a node N2 when $N1 \rightsquigarrow N2$ consists of the following steps:

1. Determining if $\text{rules}(N1) \cup \text{rules}(N2) \cup \mathcal{C}$ is consistent ⁸.
2. If $\text{rules}(N1) \cup \text{rules}(N2) \cup \mathcal{C}$ is not consistent, choosing a maximally consistent subset of $\text{rules}(N1) \cup \text{rules}(N2)$. ⁹

We make two brief remarks about the first 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

⁸We introduce the following notation: X is consistent wrt \mathcal{C} means that $X \cup \mathcal{C}$ is consistent

⁹More precisely, we choose a subset S that satisfies the following conditions:

1. $S \subset (\text{rules}(N1) \cup \text{rules}(N2))$.
 2. $S \cup \mathcal{C}$ is consistent.
 3. S is the largest subset satisfying conditions 1. and 2.
- We say that S is a maximally consistent subset of $\text{rules}(N1) \cup \text{rules}(N2)$ with respect to (wrt) \mathcal{C} . If the meaning is clear, we may omit the reference to \mathcal{C} .

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 rules in $S1 \cup S2$, an alternate phrasing of this question is: which rules should we delete? That is, how do we decide that some rules are more important than others? The general strategy is to articulate some *preference principles* for rule sets and to choose maximally consistent subsets in accordance with these principles. For specific guidance on these preference principles, we turn to several examples, below. Each example suggests a criterion for preferring some rules to others.

We consider first the example in Figure 3. HGH (Human Growth Hormone) \rightarrow (Prescription) Drugs. Suppose we have the following rules at these nodes:

Drugs:
 $\{ \forall x (\text{copay-pct}(x) = 10)$
 $\forall x, p (\text{Non-network}(p) \text{ and filled-at}(x, p) \supset \neg \text{reimbursable}(x)) \}$ (There is a 10% copay for all drugs and Drugs are not reimbursable if they are filled at non-network pharmacies.)

HGH:
 $\{ \forall x (\text{copay-pct}(x) = 50) \}$
 (The copay for HGH drugs is 50%). If we assume that the background context \mathcal{C} entails some basic arithmetic facts such as $\forall x, v1, v2 (v1 \neq v2) \supset (\neg (\text{copay-pct}(x) = v1 \wedge \text{copay-pct}(x) = v2))$, then the union of these two sets is inconsistent with respect to \mathcal{C} ¹⁰.

Obviously, there are two maximally consistent subsets:

1. The set of rules at the Drugs node, namely:
 $\{ \forall x (\text{copay-pct}(x) = 10)$
 $\forall x, p (\text{Non-network}(p) \text{ and filled-at}(x, p) \supset \neg \text{reimbursable}(x)) \}$ and
2. $\{ \forall x (\text{copay-pct}(x) = 50)$
 $\forall x, p (\text{Non-network}(p) \text{ and filled-at}(x, p) \supset \neg \text{reimbursable}(x)) \}$

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 rule from the HGH node to the subset that has the rule from the Drugs node. That is, we prefer subset 2 to subset 1.

In general, assume that $S1, S2$ are sets such that $S1 \cup S2$ is inconsistent wrt \mathcal{C} . Let $X1$ and $X2$ be maximally consistent subsets wrt \mathcal{C} of $S1 \cup S2$. Let $R1 = X1 - X2$; $R2 = X2 - X1$. i.e., $R1$ and $R2$ are all that distinguish

¹⁰The actual network contains the following cost-share rule at the Prescription Drugs node: $\forall x (\text{copay}(x) = \$3)$ (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, one way to deal with this problem is to include general principles forbidding ridiculous consequences in the background context. The difficulty then becomes expressing such principles and ensuring that we have got them all.

X1 from X2. If this is the case, we say that $X1 \cong X2$ with respect to $R1, R2$. We can state the following principle:

Principle of Positional Preference (specificity criterion): Let $N1 \rightsquigarrow N2$. Assume that $S1$ is the set of rules at $N1$; $S2$ is the set of rules at $N2$; and $S1 \cup S2$ is inconsistent wrt \mathcal{C} . Let $X1, X2$ be maximally consistent subsets wrt \mathcal{C} such that $X1 \cong X2$ wrt $R1, R2$, where $R1 \in S1, R2 \in S2$. Then we prefer $X1$ to $X2$.

That is, we prefer the set that has the rule associated with the more specific class. This principle thus merely formalizes the general intuition that lies behind all work on inheritance with exceptions: specific nodes have preference over general nodes.

For an example where the principle of specificity does not provide guidance toward choosing between maximally consistent subsets, consider Figure 4. Insulin Syringes \rightarrow Drugs; Insulin Syringes \rightarrow Supplies. Insulin Syringes inherits 2 cost share rules:

$\forall x (\text{copay-pct}(x) = 10)$ (There is a 10% copay for all prescription drugs) and

$\forall x (\text{copay-pct}(x) = 20)$ (There is a 20% copay for all supplies.)

The union of these rules is inconsistent wrt the background context. It is not the case that Drugs \rightsquigarrow Supplies or that Supplies \rightsquigarrow Drugs, so specificity does not help here. However, the path from Insulin Syringes to Drugs has priority over the path from Insulin Syringes to Supplies. Thus, it is reasonable to expect that we will prefer a rule whose source is the Drugs node over a rule whose source is the Supplies node. In practice, we must fine-tune the concept of a prioritized path, since a path may have priority for one type of rule, but not for another. For example, the path from Maternity Surgical to Maternity may have preference over the path from Maternity Surgical to Surgical for cost-share rules, but not for medical rules.

To formalize this principle, we must extend our notation as follows: Let $Rules(i, N)$ denote the set of rules of type i at node N . We can now speak of the priority of a path at a node *with respect to* a particular rule type. We then state the principle of prioritized path preference as follows:

Principle of Prioritized Path Preference: Assume $N1 \rightsquigarrow N2, N1 \rightsquigarrow N3$, and that it is not the case that $N2 \rightsquigarrow N3$ or that $N3 \rightsquigarrow N2$. Let $N1 \rightsquigarrow N2$ be the path with highest priority at $N1$ *with respect to rule type k*. Assume that $S2$ is of type k and is the rule set at $N2$, $S3$ is of type k and is the rule set at $N3$, and $S2 \cup S3$ is inconsistent wrt \mathcal{C} . If $X2$ and $X3$ are maximally consistent subsets wrt \mathcal{C} of $S2 \cup S3$ such that $X2 \cong X3$ wrt $R2, R3$, where $R2 \in S2$ and $R3 \in S3$, then we prefer $X2$ to $X3$.

The two preference principles thus proposed are compatible, but issues of ordering must be addressed. If both principles apply, which do we apply first? For example, assume that $N1 \rightsquigarrow N2, N1 \rightsquigarrow N3$, that $S1$ is the rule set at $N1$, $S2$ is the rule set at $N2$, and $S3$ is the rule set at $N3$. Assume further that $S1 \cup S2$ is inconsistent wrt \mathcal{C} ; that $S1 \cup S3$ is inconsistent wrt \mathcal{C} , and that

$S2 \cup S3$ is inconsistent wrt \mathcal{C} . Do we conflict-resolve $N2$ and $N3$, and then conflict-resolve $N1$ and the result? Or do we conflict-resolve $N1$ and $N2$, conflict-resolve $N1$ and $N3$, and then conflict-resolve the results? Clearly, different approaches will yield different maximally consistent subsets. It appears reasonable to first conflict-resolve $N2$ and $N3$, and then conflict-resolve $N1$ and the result. That is, apply the principle of primary path before the principle of positional preference.

The Positional Preference and Prioritized Path Preference principles both utilize information present in the structure of the network. In this sense, they are principles that are specific to the problems of rules conflict in the framework of inheritance. Other standard preference principles may also be needed here. One may wish to assign some rules a higher priority than others (as in (McCarthy, 1986)), regardless of the rule's position in the network. For example, a rule stating that any form of ambulance transport (air, land, or water) is covered if that form of transport is medically necessary could have priority over any other rule regarding ambulances no matter where these rules are positioned.

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.

Using this preference principle is computationally intensive. In general, determining if a maximally consistent set entails a proposition p is NP-hard (Nebel, 1991).

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. 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 rules into k types, subject to the following constraint:

If $Rules(i, n1) \cup Rules(j, n2)$ is inconsistent wrt \mathcal{C} , then $i = j$.

That is, rules 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 rule sets at $N1$ are of a different type than the rule 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 rules into sets that satisfy this constraint is not a trivial task. A good knowledge representation is indispensable. Finding such a representation is no longer only of academic interest; it can greatly affect the tractability of the system.

3.4 Back to Attribute Inheritance

We first noted the need to do conflict resolution and recognition when we were faced with the issue of rules inheritance. Further reflection, however, suggests that similar problems can arise even when performing standard attribute inheritance. Consider 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 context \mathcal{C} . 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.

This problem has not been discussed in the inheritance literature (to my knowledge) because inconsistencies in the examples used have always been explicit. The Nixon diamond example (Reiter and Criscuolo, 1983) is a classic example. Suppose, however, we modify Touretzky's (1986) modification of this example (see Figure 5). There is no explicit contradiction between Hawk and Pacifist. If, however, we add the background knowledge that these concepts are contradictory: $\forall x (Pacifist(x) \equiv \neg Hawk(x))$, 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 rules inheritance).

4 Non-network Non-unary Inheritance

Standard inheritance networks are characterized by the fact that each link connects exactly two nodes. This concept has been formalized by Etherington (1988). Using a translational approach to inheritance (see Section 2.2) in which an inheritance network is represented using the formalism of a non-monotonic logic, Etherington views such networks as a special case of default theories (Reiter, 1980). Specifically, network theories are defined as those default theories (D, W) in which

- W contains only literals or disjuncts of the form $(\alpha \vee \beta)$ where α and β are literals
- D contains only normal and semi-normal defaults of the form $\frac{\alpha:\beta}{\beta}$ or $\frac{\alpha:\beta \wedge \gamma_1 \wedge \dots \wedge \gamma_n}{\beta}$, where α, β , and all γ_i are literals. A special case of network theories are unary theories (Kautz and Selman, 1989): W does not contain disjuncts; D contains only normal or semi-normal defaults of the form $\frac{\alpha:\beta}{\beta}$ or $\frac{\alpha:\beta^1 \wedge \gamma}{\beta^1}$, where β is a literal, α and β^1 are positive literals and γ is a negative literal.

In network theories and unary theories, the antecedent of an implication or inference rule contains only one literal. In the terminology of the direct ap-

proach to inheritance networks, Etherington's restrictions entail that all links connect exactly two nodes, each of which expresses a literal.

Virtually all examples in the inheritance literature are of network theories; most of the best known, indeed, are of unary theories. The benefits semantic network, however, is highly non-network and thus highly non-unary.¹¹ Consider, for example, the therapy portion of the network. Physical Therapy (PT), Occupational Therapy (OT), and Speech Therapy (ST) are all subtypes of the Therapy service, and are covered by the Therapy Benefit. However, coverage — as well as the rules that apply — depends also on the setting in which therapy is delivered and the condition of the patient receiving therapy. PT is covered by the Therapy Benefit if delivered in the home or office; however, it is considered a Hospice Ancillary service and is covered by the Hospice Benefit if it is delivered in a hospice setting. It is considered an Inpatient Hospital Ancillary service and is covered by the Inpatient Hospital Benefit if it is delivered in an inpatient hospital setting. Moreover, if PT is delivered in an inpatient hospital setting while the patient is recovering from a cardiac condition or a stroke, it is covered under the Physical Medicine and Rehabilitation Benefit. Like PT, OT and ST are covered under the Inpatient Hospital Benefit if delivered in an inpatient setting; however, they are not covered at all if delivered in a hospice setting.

Assuming that each of these benefits, services, and settings is represented as a single node, it is clear that these facts cannot be represented as links between two nodes. We can, however, represent these facts by the following default theory: (We use the following abbreviations: IP = inpatient hospital, IA = Inpatient Ancillary, H = hospice, HA = Hospice Ancillary, PM&R = Physical Medicine and Rehabilitation)

$$\frac{PT:Therapy \quad OT:Therapy \quad ST:Therapy \quad PT \wedge IP:IA \quad PT \wedge H:HA}{Therapy \quad Therapy \quad Therapy \quad IA \quad HA} \quad \frac{PT \wedge IP \wedge (cardiac \vee stroke):PM\&R}{PM\&R} \quad \frac{(ST \vee OT) \wedge IP:IP}{IA} \quad \frac{(ST \vee OT) \wedge H:\neg Covered}{Therapy:TherapyBenefit} \quad \frac{IA:IPBenefit \quad PM\&R:PM\&RBenefit}{IPBenefit \quad PM\&RBenefit} \quad \frac{TherapyBenefit}{TherapyBenefit}$$

This is clearly a non-network, non-unary theory. The question — and the second of the new problems for inheritance theories — is: how can we represent this theory and perform inferences in an inheritance network?

Two approaches are possible. First, we can associate nodes with boolean expressions. We can do this by either dropping the assumption that nodes represent literals or introducing new literals that are defined as equivalent to boolean expressions (e.g., introducing a literal PTIP where PTIP is defined as $PT \wedge IP$). In

¹¹The term *non-network* is odd because we are after all discussing inheritance in a semantic network. Moreover, as we see below, Horty and Thomason (1990) have shown how to extend the graphical notation to general boolean expressions. In what follows, therefore, we will usually refer to these theories as non-unary.

either case, we call this approach *reifying boolean expressions as nodes*, or *reifying booleans*. Applying this approach to the example above, we have, for example, $PT \wedge IP$ as a node which is a child of PT ; $PT \wedge IP \wedge (\text{cardiac} \vee \text{stroke})$ is a node which is a child of $PT \wedge IP$, and so on. We then draw links from these new nodes to the nodes representing the conclusion. The semantic network for the therapy example can be seen in Figure 6.

We have chosen this approach for the development of the benefit inquiry tool. But the approach has a serious disadvantage: it can lead to an inordinate number of new nodes. (Note that in general, reified nodes can be children not only of leaf nodes, but also of any internal nodes in the network.) This profligacy is undesirable from the computational, intuitive, and presentational points of view. The computational problems are obvious. The intuitive problem stems from the fact that we lose the information that certain nodes should be grouped together. For example, the fact that the hospice setting treats OT and PT differently, or that OT and ST are treated similarly, is clear only after we search the OT, PT, and ST subtrees. The presentational problems are inherent in the fact that the large number of nodes that result from reification can make portions of the network difficult to display either on a page or a standard size screen.

In practice, we deal with these issues as follows: the benefit inquiry tool that we have developed does not draw trees below the PT, OT, and ST levels. Instead, we provide functions called *association links*. An *associated setting link* is a function mapping a (non-setting service) in the network together with a setting (such as home or inpatient hospital) to another node in the network. For example, the associated setting link would map the pair (PT, inpatient hospital) to Inpatient Ancillary. The associated condition-setting link maps the domain associated with the associated setting function together with a condition to a node in the network; for example ((PT, inpatient hospital), cardiac) maps to PM&R. The associated condition and setting-condition links are defined analogously. Users of the benefit inquiry system may call up these functions by clicking on a node, and subsequently clicking on the relative condition and/or setting in order to get to the parent of the reified node. This is all rather far from the spirit of inheritance, although necessary at this pass of the implementation.

A more promising approach is one based on preliminary work of Horty and Thomason (1990). Horty and Thomason have extended the framework of the standard inheritance network to accommodate general boolean expressions such as those that occur in the example above. Nodes that represent boolean operators are provided; see Figure 7 to see, for example, how the boolean expression $a \wedge b \supset c$ is represented. Analogous to Touretzky's (1986) definition of a path that is constructible but neither pre-empted nor conflicted, Horty and Thomason have defined the set of *arguments* that

are neither pre-empted nor conflicted.

The Horty-Thomason approach does not yield a particularly simple network (see Figure 8 for a rendering, using this approach, of a portion of the therapy graph depicted in Figure 6). It is easy to see that the number of nodes in such a graph, if we count boolean connectives as nodes, can easily surpass even the large number of nodes used in the reifying booleans approach. However, the network does show the connections between boolean expressions and their components, and thus facilitates reasoning about parts of boolean expressions in a way not possible using the reifying booleans approach.

This approach is preliminary and needs to be developed further. In particular, it is not clear how one can translate the definition of undefeated arguments, which has a heavily proof-theoretic flavor, into reasonable algorithms¹². An improved scheme for presentation of boolean expressions in network form is also needed.

5 Inheritance from Subevents

Standard (AI) inheritance is parent-oriented with scarcely a thought for siblings and other relations. This is not true of the current system. Rules can be inherited not only from parent nodes, but also from a distinguished set of sibling (or cousin) nodes, the subevents of that node. Consider, for example, a simplified version of the Human Organ Transplant (HOT) of the network (Figure 9): Acquisition, Transport, Storage, Preparation, and Transplant are all subevents of HOT. Rules applying to Acquisition are relevant whether one is performing a heart transplant or a liver transplant. Thus, the rules attached to the Acquisition node (and to the other subevent nodes as well) must be accessed while answering questions or adjudicating a claim about a heart or liver transplant.

This changes several features of inheritance as we understand it. First, we must modify the inheritance algorithm used in standard inheritance networks. Neither a strictly top-down nor a strictly bottom-up algorithm will suffice in this case; modification is needed. If we are performing bottom-up inheritance (the efficient way, as Selman and Levesque (1993) have demonstrated), we can modify the algorithm as follows: when reasoning about a node, we must find any subevents of each ancestor of that node. (We can streamline this search by storing with each node a boolean indicating if there are subevents descending from that node, as well as other information.)

In addition, we must also consider the following questions: If a node's rule set is inconsistent with those of a subevent, which rules have priority? Can we consider the rules of the subevent to be less specific than those of the node? If all nodes need consider the rules applicable to all subevents, why not place those rules

¹² "Reasonable" is a relative term here; as Horty and Thomason have shown, reasoning in such a network is NP-complete.

at the root service node? (The answer seems to be that since subevents may be subtypes in their own right, they deserve for both logistical and conceptual reasons to be nodes in their own right — and thus to have rule sets.) Finally, what are the semantics of subevent inheritance? What does inheriting *from* subevents mean? Answering this last question will probably entail delving into previous work on temporal reasoning (e.g., Shoham, 1988).¹³

6 Suggestions Toward a Translational Theory and Semantics

The bulk of this paper has described an inheritance network using a direct, path-based approach rather than a translational approach (see Section 2.2). A translational representation of the network, in which the network is specified in terms of a non-monotonic formalism, would be desirable as well. Specifically, it would help provide a semantics for the extended inheritance network described in this paper.

Providing a translational formalism in toto is beyond the scope of this paper. However, in this section we sketch a translational approach for an inheritance network with rules inheritance. The account is based on a classic translational approach using circumscription for standard inheritance networks (see, e.g., Lifschitz, 1994), summarized below:

- A strict positive link $p \Rightarrow q$ corresponds to the axiom $\forall x (p(x) \supset q(x))$.
- A strict negative link $p \not\Rightarrow q$ corresponds to the axiom $\forall x (p(x) \supset \neg q(x))$.
- A defeasible positive link $p \rightarrow q$ corresponds to an axiom of the form $\forall x (p(x) \wedge \neg ab_{p,q}(x) \supset q(x))$.
- A defeasible negative link $p \not\rightarrow q$ corresponds to an axiom of the form $\forall x (p(x) \wedge \neg ab_{p,q}(x) \supset \neg q(x))$.
- In addition, if there are two axioms of the form: $\forall x (p1(x) \wedge \neg ab_1(x) \supset q(x))$ and $\forall x (p2(x) \wedge \neg ab_2(x) \supset \neg q(x))$, where $\forall x (p2(x) [\wedge \neg ab_3(x)] \supset p1(x))$, we add a *cancellation axiom* $\forall x (p2(x) \supset ab_1(x))$ which ensures that the specificity criterion of standard attribute inheritance is enforced.

For example, in Figure 4, $drugs \rightarrow cov\text{-}by\text{-}drugs\text{-}benefit$ can be represented as $\forall x (drugs(x) \wedge \neg ab_{drugs, drugben}(x) \supset cov\text{-}by\text{-}drugs\text{-}benefit(x))$;

$otc\text{-}drugs \rightarrow drugs$ can be represented as $\forall x (otc\text{-}drugs(x) \wedge \neg ab_{otc, drugs}(x) \supset drugs(x))$;

$otc\text{-}drugs \not\rightarrow cov\text{-}by\text{-}drugs\text{-}benefits$ is represented as $\forall x (otc\text{-}drugs(x) \wedge \neg ab_{otc\text{-}drugs, drugben}(x) \supset \neg cov\text{-}by\text{-}drug\text{-}benefit(x))$.

In addition, we have the cancellation axiom

¹³Note that properties inherited by subevents are *not* “liquid” in Shoham’s sense. When a property is liquid, if it holds over an interval, then it holds over all subintervals; here, we merely claim that some properties are inherited by the subevents explicitly mentioned in the network, which are not, of course, all possible subevents or subintervals.

$\forall x (otc\text{-}drugs(x) \supset ab_1(x))$.

We can extend this account to rules inheritance as follows: we have already argued that a rule at a node is best thought of as a wff that is typically true at a node. Thus, for any wff ψ of the form $\forall x \phi(x)$ that is associated with node N_j , we have the corresponding statement:

$\forall x (N_j(x) \wedge \neg ab_{N_j, ind}(x) \supset \phi(x))$. (*ind* is an index; there may be several rules at N_j and each may have its own *ab* predicate associated with it.) Consider, for example, Figure 3. We assume for simplicity that the drugs node has only one rule associated with it: the cost-share rule. Then we translate the rules as follows: $\forall x (HG H(x) \wedge \neg ab_{HG H}(x) \supset copay\text{-}pct(x) = 50)$ $\forall x (drugs(x) \wedge \neg ab_{drugs}(x) \supset copay\text{-}pct(x) = 10)$

The \rightarrow link between HG H and drugs is translated as: $\forall x (HG H(x) \wedge \neg ab_{HG H, drugs}(x) \supset drugs(x))$

We also have the following fact entailed by the background context \mathcal{C} :

$\forall x (\neg (copay\text{-}pct(x) = 10 \wedge copay\text{-}pct(x) = 50))$.

If we add the fact that $HG H(c)$ (where c is a constant), we get the following:

$ab_{HG H, drugs}(c) \vee ab_{HG H}(c) \vee ab_{drugs}(c)$.

This gives us no hint as to which rule to delete. Do we assume that the link between HG H and drugs does not hold for c , that the rule at drugs does not hold, or that the rule at HG H does not hold? We can implement the specificity criterion by assigning a higher priority to $ab_{HG H}$ than to ab_{drugs} . Moreover, we will wish in general to assume that the \rightarrow link between nodes is stronger than the associations between rules and nodes, so we assign a higher priority to $ab_{HG H, drugs}$ than to $ab_{HG H}$. Thus, our circumscription policy is: We circumscribe $ab_{HG H, drugs}$, $ab_{HG H}$, and ab_{drugs} , with $ab_{HG H, drugs} \succ ab_{HG H} \succ ab_{drugs}$. This will entail that $ab_{drugs}(c)$ in the example above. Thus, $copay\text{-}pct(c) = 50$.

We make the following conjecture, based on the observation of several examples: In general, to implement the specificity criterion for rules inheritance (Principle of Positional Preference), translate all links as described in (Lifschitz, 1994). For each wff ψ of the form $\forall x \phi(x)$ such that $At(n_i, \psi)$, have an axiom $\forall x (n_i(x) \wedge \neg ab_{n_i}(x) \supset \phi(x))$ with the following prioritization: If $n_i \rightsquigarrow n_j$, then the abnormality associated with a rule at n_i receives a higher priority than the abnormality associated with a rule at n_j . The abnormalities associated with defeasible links receive higher priorities than abnormalities associated with rules.

Formalizing this conjecture more precisely and proving that it delivers the desired behavior is a subject for future work. It is likely that a simple extension of this prioritization policy will suffice for the principle of Prioritized Path Preference. Preferring one extension over another, as discussed at the end of Section 3.2, is a more complicated matter. However, as Groszof (1992) has shown, this preference principle can also be represented using prioritized circumscription. The development of the circumscription policy will entail deter-

mining how these preference principles interact.

7 Conclusion

Our investigation of an inheritance network in an applied setting has turned up several problems for inheritance theories, specifically inheriting rule sets, non-unity inheritance, and inheritance from subevents. Hopefully, research into these problems will alter the image of inheritance as a very narrow subfield of non-monotonic reasoning.

These problems were discovered in the medical insurance domain but most likely occur in many other settings. In particular, they are likely to occur in any domain in which concepts are structured hierarchically (thus suggesting the need for a semantic network) and there is a need to use and reason with regulations (which can be represented as rules or well-formed formulas in a first-order logic). Examples of such domains include other parts of the insurance industry, such as property, car, and life insurance; and enterprise modelling, particularly in hierarchically organized bureaucracies which have many regulations. In addition, the legal domain appears to be a rich area for investigation, since legal concepts are often organized hierarchically, and the relation between laws and well-formed formulas is obvious.

Acknowledgements: Thanks to Ernie Davis, Benjamin Grosf, and Daniel Lehmann for helpful discussions and suggestions, to Bill Wynne for his assistance in populating the benefits semantic network, and to the referees for their comments on an earlier version of this paper.

REFERENCES

- D. Etherington(1988): *Reasoning with Incomplete Information*, Morgan Kaufmann, Los Altos
- M.P. Evett, J.A. Hendler, and L. Specter (1994): Parallel Knowledge Representation on the Connection Machine, *Journal of Parallel and Distributed Computing*, 2,22:168-184
- P. Gardenfors (1988): *Knowledge in Flux*, MIT Press, Cambridge
- B. Grosf (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 and R. Thomason(1990): Boolean Extensions of Inheritance Networks, *AAAI 1990*, Morgan Kaufmann, Los Altos, 663-669
- J. Horty, R. Thomason and D. Touretzky (1990): A Skeptical Theory of Inheritance in Nonmonotonic Semantic Networks, *Artificial Intelligence 42* 311-349
- H. Kautz(1991): A Formal Theory of Plan Recognition and its Implementation in J. Allen, H. Kautz, R. Pelavin, and J. Tenenber: *Reasoning About Plans*, Morgan Kaufmann, San Mateo, 69-125
- 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 Kaufmann, San Mateo, 189-197
- V. Lifschitz (1988): Benchmark Problems for Formal Non-monotonic Reasoning, Version 2.0. In M. Reinfrank, J. de Kleer, M. Ginsberg, and E. Sandewall (eds.): *Non-monotonic Reasoning: Proceedings of the 2nd International Workshop, Lecture Notes in AI #346*, Springer-Verlag, 202-219
- E. Mays, R. Dionne, and R. Weida (1991): K-REP System Overview, *SIGART Bulletin*, 2:3
- 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
- D. McDermott (1982): A Temporal Logic for Reasoning About Processes and Plans, *Cognitive Science 6*, 101-155
- B. Nebel (1989): A Knowledge Level Analysis of Belief Revision, in R. Brachman, H. Levesque and R. Reiter (eds): *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann, San Mateo 301-311
- B. Nebel (1991): Belief Revision and Default Reasoning: Syntax Based Approaches, 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 417-428
- M. Quillian (1968): Semantic Memory in M. Minsky (ed): *Semantic Information Processing*, MIT Press, Cambridge
- R. Reiter(1980) A Logic for Default Reasoning, *Artificial Intelligence 13*
- R. Reiter and G. Criscuolo (1983): On Interacting Defaults, *IJCAI 1981*, 270-276
- J. Schmolze and T. Lipkis (1983): Classification in the KL-ONE Representation System, *IJCAI 1983*
- 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
- D. Touretzky (1986): *The Mathematics of Inheritance Systems*, Morgan Kaufmann, Los Altos