Implicit Ordering of Defaults in Inheritance Systems

David S. Touretzky

Computer Science Department
Carnegie-Mellon University
Pittsburgh, PA 15213

Abstract

There is a natural partial ordering of defaults in inheritance systems that resolves ambiguities in an intuitive way. This is not the shortest-path ordering used by most existing inheritance reasoners. The flaws of the shortest-path ordering become apparent when we consider multiple inheritance. We define the correct partial ordering to use in inheritance and show how it applies to semantic network systems. Use of this ordering also simplifies the representation of inheritance in default logic.

1. Introduction

There is a natural partial ordering of defaults in inheritance systems that resolves ambiguities in an intuitive way. This ordering is defined implicitly by the hierarchical structure of the inheritance graph. Surprisingly, it is not the shortest-path ordering used by most existing inheritance systems, such as FRL [1] or NETL [2]. We define the correct ordering, called inferential distance, and show how its use results in more reasonable inheritance behavior than that of either FRL or NETL. We go on to represent inheritance systems in default logic, following the example of Etherington and Reiter [3]. Although exceptions must normally be treated explicitly in default logic, use of inferential distance allows us to handle them implicitly, which has several advantages.

2. The Inferential Distance Ordering

The intuition underlying all inheritance systems is that subclasses should override superclasses. Where inferential distance differs from the shortest-path ordering is in determining subclass/superclass relationships. The inferential distance ordering says that A is a subclass of B iff there is an inheritance path from A to B. In single (as opposed to multiple) inheritance systems, the shortest inference path always contains the inference from the most specific subclass. But under multiple inheritance, there are two cases where the shortest-path ordering disagrees with inferential distance. One involves the presence of true but redundant statements; the other involves ambiguous networks.

3. Handling True But Redundant Statements

Figure 1 illustrates a problem caused by the presence of redundant links in an inheritance graph. Let us start with the following set of assertions: "elephants are typically gray; royal elephants are elephants but are typically not gray; circus elephants are royal elephants; Clyde is a circus elephant." If subclasses override superclasses, then Clyde is not gray. But what happens when we add the explicit statement that Clyde is an elephant, as shown in figure 1? This is a redundant statement because Clyde is indisputably an elephant; he was one before we added this statement; in fact, this is one of the inferences we expect an inheritance reasoner to generate. Yet when we make the fact explicit it causes problems. In FRL Clyde will inherit properties through both Circus.Elephant and Elephant, so FRL will conclude that he both is and is not gray. In NETL, the redundant statement that Clyde is an elephant contributes an inference path to gray that is shorter than either of the two paths (one to gray, one to not-gray) which go through Circus.Elephant. NETL will therefore conclude that Clyde is gray, which contradicts the (correct) conclusion it would reach without the redundant link present.

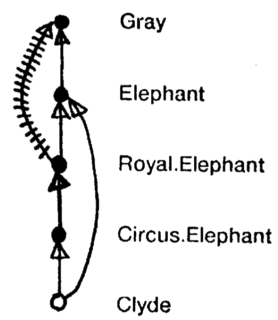

Figure 1.

Inferential distance is unaffected by redundant links. Clyde could either inherit grayness, a property of Elephant, or non-grayness, a property of Royal.Elephant. Since the network contains an inheritance path from Royal.Elephant to Elephant, according to the inferential distance ordering Royal.Elephant is a subclass of Elephant; the direct link from Clyde to Elephant does not alter this relationship. Therefore we conclude that Clyde should inherit non-grayness from Royal.Elephant rather than grayness from Elephant.

4. Ambiguous Inheritance Networks

Consider the following set of assertions, shown in NETL notation in figure 2. "Quakers are typically pacifists; pro-defense people are typically not pacifists; Republicans are typically pro-defense; Nixon is both a Quaker and a Republican." This network is ambiguous: it has two valid extensions. (An extension is the nonmonotonic or default logic equivalent of a theory [4].) In one extension Nixon is a pacifist; in the other he is not.

Most existing inheritance reasoners would not recognize this ambiguity. If pacifism were a slot that could be filled with either

---

This research was sponsored by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 3597, monitored by the Air Force Avionics Laboratory Under Contract F33615-81 K-1539. The author was partially supported by a fellowship from the Fannie and John Hertz Foundation. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US Government.

"yes" or "no," FRL would simply return both values, with no notice of the inconsistency. NETL would conclude that Nixon was a pacifist simply because the inference path to that conclusion is shorter than the path to the opposite conclusion. Yet the fact that one path is shorter than the other is irrelevant.

Nixon can inherit either pacifism, a property of Quaker, or non-pacifism, a property of Pro.Defense. Since there is no inheritance path from Quaker to Pro.Defense, nor vice versa, the inferential distance ordering provides no justification for viewing either class as a subclass of the other. Thus an inheritance reasoner based on inferential distance would be forced to recognize the ambiguity with regard to Nixon's pacifism.

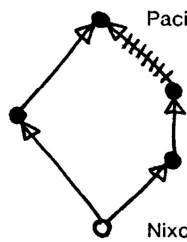

Figure 2.

5. Inferential Distance in Semantic Networks

TINA (for Topological Inheritance Architecture) is a recently implemented inheritance reasoner based on inferential distance [5]. TINA constructs the extensions of unambiguous inheritance networks by incrementally generating inheritance paths and weeding out those that violate the inferential distance ordering. This method also allows TINA to detect and report ambiguities in networks with multiple extensions. Since TINA does not use the shortest-path approach to inheritance, it is not misled by redundant links in the inheritance graph.

Another part of TINA, called the *conditioner*, can be used to correct certain problems with inheritance in NETL reported in [6]. These problems are due to NETL's implementation as a set of parallel marker propagation algorithms based on shortest-path reasoning. TINA's conditioner modifies the topology of a NETL network (after the extension has been computed) so as to force marker propagation scans to produce results in agreement with the correct extension, as defined by inferential distance. This technique can also be applied to other semantic network systems (incuding parallel systems) to speed up their inheritance search, since once a network has been conditioned we can search it using a shortest-path inheritance algorithm. Shortest-path algorithms are simpler and more effcient than inferential distance algorithms. One drawback is that any changes to the network will require at least a portion of it to be reconditioned.

6. Representing Inheritance in Default Logic

In default logic, a default inference rule is written in the form

$$\frac{\alpha(x) : \beta(x)}{\gamma(x)}$$

where $\alpha(x)$, $\beta(x)$, and $\gamma(x)$ are well-formed formulae called the *prerequisite*, the *justification*, and the *consequent* of the default, respectively [7]. The interpretation of this rule is: if $\alpha(x)$ is known, and $\beta(x)$ is consistent with what is known, then $\gamma(x)$ may be concluded. A default is said to be *normal* if the consequent is the entire justification, *i.e.* $\beta(x)$ and $\gamma(x)$ are identical. A default is said to be *semi-normal* if it is of form

$$\frac{\alpha(x) : \beta(x) \wedge \gamma(x)}{\beta(x)}$$

Etherington and Reiter use semi-normal defaults to represent inheritance systems in default logic [3]. To see why semi-normal defaults are necessary, consider the example in figure 3.

Figure 3.

This figure could be represented as the set of normal defaults D1-D3 below, plus the assertion Ostrich(Henry). We represent "ostriches are birds" (rule D2) in this example as a default rather than as a strict implication mainly for uniformity; this decision is not critical to the example. Another reason, though, is that in NETL, which we are trying to model, all statements are defeasible.

(D1) $$\frac{\text{Bird}(x) : \text{Can.Fly}(x)}{\text{Can.Fly}(x)}$$

(D2) $$\frac{\text{Ostrich}(x) : \text{Bird}(x)}{\text{Bird}(x)}$$

(D3) $$\frac{\text{Ostrich}(x) : \neg\text{Can.Fly}(x)}{\neg\text{Can.Fly}(x)}$$

Ostrich(Henry)

Using D1-D3, the assertion Ostrich(Henry) generates not one but two extensions. In one extension, Henry can't fly because he is an ostrich. But in the other, Henry can fly because he is a bird. This problem of "interacting defaults" was noted by Reiter and Criscuolo [8]. To solve it, they would replace the normal default D1 with the semi-normal version D1':

(D1') $$\frac{\text{Bird}(x) : \neg\text{Ostrich}(x) \wedge \text{Can.Fly}(x)}{\text{Can.Fly}(x)}$$

In D1', the restriction that ostriches should not be inferred to fly is incorporated into the default rule that birds fly. If we add two more types of non-flying birds, say penguins and dodos, then D1' would have to be replaced by another default that mentions all three exceptions.

There are three problems with handling exceptions explicitly using semi-normal defaults. First, as information is added to a knowledge base, existing default rules must continually be replaced with new ones that take the new exceptions into account. Second, the complexity of each individual default increases as the knowledge base grows, because more exceptions must be mentioned. Third, in any given inheritance network, the translation of one link cannot be determined independently of that of the others. For example, an IS-A link between Bird and Can.Fly might be represented as the normal default D1, yet in some networks the exact same link must be represented by D1'. Syntactically, every link in an inheritance network is a normal default, since the network formalism makes no explicit reference to exceptions. The problem with representing inheritance assertions as *semi*-normal defaults can be summarized by saying that it lacks what Woods calls "notational efficacy," a term that encompasses such properties as conciseness of representation and ease of modification [9].

Etherington and Reiter suggest that NETL treats some types of exceptions explicitly (*i.e.* its rules are semi-normal) because two types of exception link were proposed in [6]. These links were to be added to the network automatically, in a preprocessing step, to force NETL's marker propagation algorithms to produce the desired results. In order to add these exception links one must have a specification for the correct interpretation of the network. When one creates a NETL network, then, the meaning must already be determined, whether or not the network is subsequently annotated with exception links. The NETL formalism itself does not require that exceptions be treated explicitly. Exception links were later abandoned as a marker propagation device.

In FRL, an explicit mechanism for noting exceptions has never even been proposed. If we wish to translate inheritance networks into default logic using semi-normal rules, how are we to derive these rules from the syntactically normal ones the inheritance system contains? This question was left unanswered by earlier work on nonmonotonic inheritance. The inferential distance ordering provides an answer.

7. A Formal Analysis of Inferential Distance

By representing inheritance in default logic, Etherington and Reiter were able to give a formal semantics to inheritance systems along with a provably correct inference procedure. However, since their representation does not include the notion that subclasses should override superclasses, it does not fully express the meaning of inheritance. In [5] I present a formal analysis of inheritance under the inferential distance ordering. Some of the major theorems are:

- Every acyclic inheritance network has a constructible extension. (A similar result was proved in [7].)

- Every extension of an acyclic inheritance network is finite.

- An extension is inconsistent iff the network itself is inconsistent. (We use an expanded notion of inconsistency in which the rules "typically birds can fly" and "typically birds cannot fly" are mutually inconsistent. They would not be in default logic.)

- The union of any two distinct extensions is inconsistent.

- A network is ambiguous (has multiple extensions) iff it has an unstable extension. Instability is a property defined in [5]. A necessary condition for instability is that the network contain a subgraph of the form shown in figure 4.

- Every extension of an ambiguous network is unstable. Corollary: we can determine whether a network is ambiguous by constructing one of its extensions and checking it for stability.

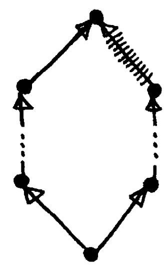

Figure 4.

- Every inheritance network is conditionable. That is, given a network and one of its extensions, we can always adjust the topology of the network so that a shortest path reasoner will produce results in agreement with the chosen extension.

- Additive conditioning (*i.e.* adding but never subtracting links) is sufficient.

8. Implementing Inferential Distance in Default Logic

Consider a subset of default logic corresponding to a family of acyclic inheritance graphs. We can represent an IS-A or IS-NOT-A link between a class P and a class Q as a normal default in the obvious way, *viz.*:

$$\frac{P(x) : Q(x)}{Q(x)} \qquad\qquad \frac{P(x) : \neg Q(x)}{\neg Q(x)}$$

Let P(x) be the prerequisite of a rule Di and Q(x) the prerequisite of a rule Dj. We define Di < Dj to mean that either there exists a default with prerequisite P(x) and conclusion Q(x), or there exists a default Dk such that Di < Dk and Dk < Dj. Returning to the ostrich example, note that D3 < D1 and D2 < D1 by this definition. D2 and D3 are unordered with respect to each other since their prerequisites are the same. The < relation is clearly a partial ordering.

The equivalent of an inheritance path in default logic is a proof sequence. The example involving Henry the ostrich, when represented by the normal defaults D1-D3, generates a pair of conflicting proof sequences S1 and S2. The arrows in these sequences indicate the defaults that justify each inference.

(S1) Ostrich(Henry) **--D3-->** ¬Can.Fly(Henry)

(S2) Ostrich(Henry) **--D2-->** Bird(Henry)
--D1--> Can.Fly(Henry)

Note that if Di precedes Dj in some proof sequence, then Di < Dj. If we order proof sequences by comparing the ordering of the maximal rules used in each proof, we see that S1 < S2 because D3 < D1. To apply inferential distance to default logic, we use the ordering on proof sequences as a filter over the set of possible extensions. (This idea was suggested by David Etherington.) Basically, we reject as invalid any extension in which a conclusion depends on a proof sequence Si such that there is a contradictory sequence Sj < Si. Thus, the extension in which Henry can fly would be rejected, since that conclusion depends on proof sequence S2 but there is a contradictory proof sequence S1 < S2.

Now let us try expressing figure 2 in default logic:

(D4) $$\frac{\text{Quaker}(x) : \text{Pacifist}(x)}{\text{Pacifist}(x)}$$

(D5) $$\frac{\text{Republican}(x) : \text{Pro.Defense}(x)}{\text{Pro.Defense}(x)}$$

(D6) $$\frac{\text{Pro.Defense}(x) : \neg\text{Pacifist}(x)}{\neg\text{Pacifist}(x)}$$

$$\text{Quaker(Nixon)} \wedge \text{Republican(Nixon)}$$

The inference paths we generate about Nixon are:

(S3) Quaker(Nixon) --D4--> Pacifist(Nixon)

(S4) Republican(Nixon) --D5--> Pro.Defense(Nixon) --D6--> ¬Pacifist(Nixon)

The only ordering relation among these defaults is D5 < D6. Since D4 and D6 are unordered, the proof sequences S3 and S4 are unordered, so of the two extensions we obtain, one relying on S3 and one on S4, neither is to be preferred over the other.

At this point the reader should have no trouble translating figure 1 into a set of normal defaults and verifying that under the inferential distance ordering, only the desired extension is produced.

9. The Significance of Hierarchy

Brachman, in his discussion of what IS-A is and isn't, suggests that "to the extent inheritance is a useful property, it is strictly implementational and bears no weight in any discussion of the expressive or communicative superiority of semantic nets" [10]. When an inheritance system is devoid of exceptions he is clearly right. But in nonmonotonic inheritance systems, which provide for a simple form of default reasoning, the basic assumption that classes are structured hierarchically makes implicit handling of exceptions possible. In contrast, exceptions must normally be handled *explicitly* in default logic, since default logic contains no notion of hierarchy.

Implicit handling of exceptions is possible when we are restricted to hierarchical domains with simple forms of defaults, but default logic admits more intricate sorts of theories. Unrestricted semi-normal theories cannot be represented by normal ones using the ordering defined here. Default logic is clearly a more powerful formalism than inheritance for representing knowledge, but the latter remains important due to its conceptual simplicity and efficient inference algorithms.

10. Conclusions

The intuition underlying all inheritance systems is that subclasses should override superclasses. Inferential distance is a partial ordering on defaults that implements this intuition. The inferential distance ordering differs from the shortest-path ordering used by most inheritance reasoners in cases where the network is ambiguous or contains true but redundant statements. In these cases, the shortest-path ordering fails to ensure that subclasses (and only subclasses) override superclasses.

Applying inferential distance to the default logic representation of inheritance systems allows us to faithfully represent these systems with no loss of notational efficacy. Under inferential distance, default rules need not be discarded as more information is added to the knowledge base; individual rules do not become more complex as exceptions accumulate; and the translation of any one link in an inheritance network into a default is independent of that of any other.

Acknowledgements

I am grateful to David Etherington, Jon Doyle, and Scott Fahlman for many insightful discussions, and to the referees for sugggestions which led to the restructuring of this paper.

References

[1] Roberts, R. B., and I. P. Goldstein. *The FRL Manual.* MIT AI Memo 409, MIT, Cambridge, MA, 1977.

[2] Fahlman, S. E. *NETL: A System for Representing and Using Real-World Knowledege.* MIT Press, Cambridge, MA, 1979.

[3] Etherington, D. W., and R. Reiter. "On Inheritance Hierarchies With Exceptions," *Proc. AAAI-83*, August, 1983, pp. 104-108.

[4] Reiter, R. "A Logic for Default Reasoning," *Artificial Intelligence* Vol. 13, No. 1-2, April 1980, pp. 81-132.

[5] Touretzky, D. S. *The Mathematics of Inheritance Systems.* Doctoral dissertation, Computer Science Dept., Carnegie-Mellon University, Pittsburgh, PA, 1984.

[6] Fahlman, S. E., D. S. Touretzky, and W. van Roggen. "Cancellation in a Parallel Semantic Network," *Proc. IJCAI-81*, August, 1981, pp. 257-263.

[7] Etherington, D. W. *Formalizing Non-Monotonic Reasoning Systems.* Technical report 83-1, Dept. of Computer Science, University of British Columbia, Vancouver, BC, Canada, 1983.

[8] Reiter, R., and G. Criscuolo. "On Interacting Defaults," *Proc. IJCAI-81*, August, 1981, pp. 270-276.

[9] Woods, W. A. "What's Important About Knowledge Representation?" *Computer*, Vol. 16, No. 10, October 1983, pp. 22-27.

[10] Brachman, R. J. "What IS-A Is and Isn't," *Computer*, Vol 16, No. 10, October, 1983, pp. 30-36.