

A Theory of Nonmonotonic Inheritance Based on Annotated Logic

Krishnaprasad Thirunarayan *
Department of Computer Science and Engineering
Wright State University
Dayton, Ohio 45435

Michael Kifer†
Department of Computer Science
SUNY at Stony Brook
Stony Brook, NY 11794

April 20, 1995

Abstract

We propose a logical language for representing networks with nonmonotonic multiple inheritance. The language is based on a variant of annotated logic studied in [5, 6, 17, 18, 19, 20, 21]. The use of annotated logic provides a rich setting that allows to disambiguate networks whose topology does not provide enough information to decide how properties are to be inherited. The proposed formalism handles inheritance via strict as well as defeasible links. We provide a formal account of the language, describe its semantics, and show how a unique intended model can be associated with every inheritance specification written in the language. Finally, we present an algorithm that correctly propagates inherited properties according to the given semantics. The algorithm is also complete in the sense that it computes the set of all properties that must be inherited by any given individual object, and then terminates.

*Work supported in part by NSF grant IRI-9009587.

†Work supported in part by NSF grant IRI-8903507.

1 Preliminaries

It is customary to depict inheritance networks using diagrams where *properties* and *individual objects* are represented as nodes of the diagram. Object and property nodes are depicted as labeled rectangles and ovals, respectively. Each property node defines a *class* of individual objects that possess this property. Each link in the diagram represents one of the following four types of relationship between the nodes:

- A solid link going from an individual object o to a property node p represents an *instance-of* relationship; it says that o is a member of the class defined by p .
- A slashed link from an individual node o to a property node p represents a *non-instance* relationship; it says that o is *not* a member of the class defined by p .
- A solid link from p_1 to p_2 , where p_1 and p_2 are property nodes, denotes an *is-a* relationship; roughly, it says that the class defined by p_1 is a subset of the class of objects defined by p_2 .
- A slashed link from p_1 to p_2 denotes an *is-not-a* relationship between these property nodes; it is a statement to the effect that elements of the class defined by p_1 “usually” do not have property p_2 .

The is-a and is-not-a relationships come in two flavors: *strict* and *defeasible*; “instance-of” and “non-instance” relationships are always strict. Strict “instance-of”, “non-instance”, and “is-a” links have simple mathematical meaning based on the usual set-theoretic relations \in , \notin , and \subseteq . The strict “is-not-a” link from p_1 to p_2 says that the classes defined by p_1 and p_2 are disjoint. We will depict strict “is-a” and “is-not-a” links using bold solid and bold slashed arrows, respectively. For “instance-of” and “non-instance” links we will use plain (solid or slashed) arrows, even though these links are always strict.

The defeasible flavor of the above links is not that well understood, and there are several different views of what they should mean. The common intuition is that such a link between p_1 and p_2 expresses the fact that under normal circumstances the members of the class of p_1 are or are not members of the class of p_2 , depending on whether the relationship is “is-a” or “is-not-a”. However, certain “atypical” members of p_1 may be “excused” from being members of the class of p_2 . Defeasible links will be depicted using plain (solid or slashed) arrows.

Consider the “Penguin Triangle” example shown in Figure 1. Intuitively, the individual object Donald there should inherit the property *fly* by virtue of being a bird. However, for Tweety things are not that clear-cut. On one hand, it should fly because it is a bird. On the other hand, as a penguin, Tweety might as well inherit $\neg fly$. Thus, on the surface, the network appears to be ambiguous about whether Tweety is a flying bird.

Several works tried to address this and related problems from various angles. The main advantage of our formalism is that it is based on a well-studied logic, has model theory and a sound and complete algorithm for computing inheritance. We also argue that our