A Parametric Definition for a Family of Inheritance Reasoners

Carl Vogel*
Institute for Computational Linguistics,
Azenbergstr. 12, University of Stuttgart, D-70174 Stuttgart, Germany
vogel@ims.uni-stuttgart.de

Fred Popowich
School of Computing Science
Simon Fraser University, Burnaby V5A 1S6, British Columbia, Canada

March 22, 1996

This paper gives a declarative specification of a popular inheritance system and shows how simple changes to this specification can result in different path-based reasoners. This parameterized definition provides a deeper understanding of the fundamental differences between some of the more popular path-based inheritance reasoners. In particular, it allows the clarification of some of the results on the complexity of reasoning in the various systems. The uniform framework also allows definition of novel systems which constitute intermediate points in the space of possible reasoners, and facilitates perspicuous Prolog implementation.

Kewords: inheritance reasoning, default reasoning, inheritance proof theory

Contents

*The work reported here is primarily the research of Carl Vogel, with Fred Popowich being particularly involved with the initial logical specification and implementation. Thanks to Nick Cercone, who collaborated with the authors on earlier research relating to the material presented in this paper, and also to two anonymous reviewers for constructive suggestions. Vogel is particularly grateful to Robin Cooper and Jeff Pelletier for feedback and encouragement as well as to the Marshall Aid Commemoration Commision for making it possible for him to do his Ph.D. at the Centre for Cognitive Science in Edinburgh. Popowich wishes to acknowledge the support of the Natural Sciences and Engineering Research Council of Canada.

1 Introduction

Path-based inheritance reasoners provide a non-monotonic propositional system for default inference which is applicable for the analysis of human reasoning about generics like, “Birds fly.” In these systems, generics are encoded as links in directed acyclic graphs, the nodes of which represent individuals, properties, or classes, and the links of which represent statements of positive or negative defaults. So, for example, Figure 1 depicts a default inheritance network. Let the nodes of the graph labeled A, B and C represent penguins, birds and fliers, respectively. Thus, the network represents that birds fly, that penguins are birds, and that penguins do not fly.

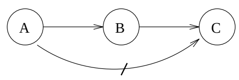


Figure 1: A Simple Inheritance Network

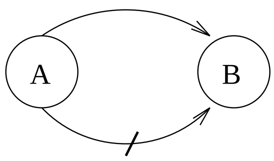


Figure 2: An Inconsistent Network

Inheritance has been invoked as a psychologically plausible logic for capturing human reasoning with generic information. Arguably, there are no contentful assertions that are strictly true under universal quantification. Instead, such truths can be seen as typicalities, statements that hold in spite of exceptions. One approach (as in conditional logic approaches) is to analyze generics as universal implications that hold as material implications in the correctly restricted set of cases (thus eliminating exceptions). The approach of inheritance reasoning, in contrast, is to take typicality as a primitive, and to develop a nonclassical propositional logic of reasoning with such statements. The approach is motivated by the ubiquity of hierarchical arrangements of information in popular and scientific organization of knowledge. However, in part because the primitive is typicality and not strict implication, the approach leads to a nonclassical system. One feature of inheritance, is for instance, the localization of inconsistent information. Networks like the one in Figure 1 contain contradictions when classically interpreted, but using typicality as the primitive, it is possible to reach an unproblematic conclusion that penguins don't fly. Even if there were directly conflicting information, such as in Figure 2, this need not propagate as in classical systems to warrant any conclusion at all.[1] The localization of inconsistency, a form of *paraconsistency*, is deemed to be a more psychological plausible notion than the classical treatment of inconsistency. Additionally, the attractive computational properties of some of these systems (polynomial time decision) even with greater expressivity than classical propositional logics[2] make inheritance reasoning a serious candidate for consideration as a psychological model of reasoning (note that the various nonmonotonic logics (McDermott & Doyle, 1980) and default logics (Reiter & Criscuolo, 1983) are undecidable).

Inheritance reasoners define methods for reaching conclusions implicit in graph representations of sentences.[3] Implicit conclusions correspond to paths through the graph that are distinguished as *permitted*. As an example, an easily stated theory of inheritance is shortest path reasoning, in which the conclusion of a graph that is not simply linear is taken to be the conclusion that corresponds to the shortest linear path through the entire graph. Touretzky (1986) has shown this form of reasoning to be formally undesirable; however, most inheritance reasoners agree with shortest path reasoning in simple cases, and determine, for instance, that the potential path in Figure 1 from A to C through the intervening node B is not permitted because it is *preempted* by the *more specific* information represented by the direct negative link from A to C. However, when it comes to more complex graph topologies, different inheritance reasoners diverge considerably on which paths should be permitted from a given graph (Touretzky, 1986; Touretzky, Horty, & Thomason, 1987; Boutilier, 1989).

A wide range of path based reasoners exist in the literature (Sandewall, 1986; Stein, 1989; Touretzky, 1986; Horty, Thomason, & Touretzky, 1990; Boutilier, 1989; Geffner & Verma, 1989) and axes of variation among many of the systems have been outlined (Touretzky et al., 1987). In this paper, the system of Horty et al. (1990) (the system they describe is refered to herein as $H90$) is taken as a basic system and a space of related reasoners is defined from that basis. The axes of variations range from syntactic properties of reasoners (on-path or off-path preemption) to more complex semantically motivated differences (restricted skepticism vs. full skepticism). This section explains these properties in informal terms through graph topologies, setting the stage for the following section to identify exactly how these properties of reasoners emerge from the reasoners' formal definitions (by providing a parameterized definition for path-based inheritance

[1] In fact, it seems that if there is ever going to be a general semantics for natural language, it will have to be in a framework that can represent local inconsistencies. No one would argue that Maugham's (1919) *The Moon and Sixpence* is meaningless because it contains the sentence, "There is no object more deserving of pity than the married bachelor" (p. 163), quite the contrary!

[2] This requires some qualification, as inheritance systems typically lack disjunction and general negation. However, since inheritance does allow typicality as a primitive and since individuals can be represented in the language, inheritance has aspects that make it more expressive than the propositional calculus. On the other hand, ignoring the interpretation of links as 'typically', inheritance can be seen as a subset of the monadic predicate calculus. In this light, the efficiency of inheritance can be seen to follow from the prohibition on cycles, and other topological features of allowable argument structures (see Niemelä & Rintanen, 1994).

[3] In what follows, sets of default sentences are often referred to in terms of structural properties of those sets when represented in graphic notation; in these terms, theories are directed acyclic graphs.

reasoners in which setting different parameters in a basic system yields each of the possibilities). This framework has been utilized in providing a general semantics for inheritance reasoning parameterized to the same proof-theoretic distinctions (Vogel, 1995b, 1996a). Various arguments about the 'correctness' of particular proof theoretic strategies are made in the literature by appeal to introspective judgements of plausibility. Hewson and Vogel (1994) attempted to identify psychological evidence which could properly adjudicate. This paper will also present a definition for a reasoner that is consistent with their findings. The result of this work is that a number of extant and novel inheritance reasoners can be clearly inter-related, and implemented (see the appendix), thus providing a deeper understanding of the relationships among competing systems. The declarative specification of the skeptical theory of inheritance reasoning given by Horty et al. (1990), as well as its Prolog implementation, was first presented by Vogel, Popowich, and Cercone (1993). That work set the foundations for parameterization of the proof theory into the uniform set of definitions given here which has not appeared before.

1.1 Some Formalities

Default inheritance networks are composed of *nodes* and default *links*. Nodes represent individuals, concepts, and properties, and links represent the classification of connected nodes. A link between two nodes in the form $A \longrightarrow B$ denotes the classification: As are typically Bs (As are Bs). A link with a slash through it, $\not\longrightarrow$, is a negative link. A *network* is a directed acyclic graph of positive and negative links, and is used to represent propositional knowledge of a "tangled" or "twisted" hierarchical nature. An *inheritance reasoner* is a set of definitions that determine the conclusions justified by a given network. Conclusions correspond to the links themselves or to chains of links (whose endpoints form an *implicit link*) that are *permitted paths.* Let π vary over paths; $lastnode(\pi)$ denotes the last node in path π, and $firstnode(\pi)$ denotes the first node in path π. If π is a positive path, then $a \overset{\sigma}{\rightsquigarrow} z$ denotes the implicit link between $firstnode(\pi) = a$ and $lastnode(\pi) = z$ through the possibly empty sequence of nodes σ. If π is a negative path, then $a \overset{\sigma}{\not\rightsquigarrow} z$ denotes the implicit link between $firstnode(\pi)$ and $lastnode(\pi)$ through the possibly empty sequence of nodes σ, such that the last link in π is a negative link. An *expansion* of a network is the set of explicit and implicit links supported by a reasoner applied to that network. An inheritance reasoner is the set of definitions that specify which paths in a network are permitted, and the permitted paths are exactly the contents of an expansion.

Individuals (as opposed to concepts or classes) may appear only as the first node of a path, but they need not necessarily occur in the path at all. Any link in a network is a path; if the link is of the form, $p \longrightarrow r$, then it is a *positive* path, and if it is of the form $p \not\longrightarrow r$, then it is a *negative* path. If a path consists of more than one link it is a compound path. Since networks are assumed to be acyclic, every path has both a first node and a last node. The *length* of a path is the number of links that it contains. The *degree* of a path is the length of the longest chain of links between its endpoints. For a given network, if π is a positive path, $lastnode(\pi) \longrightarrow r$ is a link, and r does not occur as a node in π, then $\pi \longrightarrow r$ is a positive path as well. Symmetrically, if $lastnode(\pi) \not\longrightarrow r$ is a link, and r does not occur in π, then $\pi \not\longrightarrow r$ is a negative path. Since we assume π to vary over only positive paths, this means that negative links can occur only at the end of a path. The *polarity* of a path is determined by its last link: a path whose last link is negative is called a negative path, and all other paths are positive. It is convenient to refer to awkwardly long paths with strings of node labels; using this convention, "πr" and "π / r" are abbreviations for the aforementioned paths. The metanotation $\overset{\circ}{\longrightarrow}$ indicates a link whose polarity does not matter and similarly for implicit links $\overset{\circ}{\rightsquigarrow}$.[4]

Implicit in the definition of a path's degree is the fact that there can be more than one path between any two nodes. In such cases, paths are said either to conflict or are said to be redundant

[4]Note that Touretzky (1986) incorporated "don't know" links directly into his language; however, these are not used in the literature in general nor does he make extensive use of them himself. Nonetheless, it is convenient in the metatheory to have a way of talking about something that is definitely a link, even when its polarity does not matter.

to each other. If a path π is redundant with respect to another path π', then π is a subsequence of π'.

Definition 1 (Subsequences $\subseteq$) *If π and π' are two paths of the same polarity, $\pi \subseteq \pi'$ iff each node in π also occurs in π' and for any two nodes x and y such that $x \overset{\sigma}{\leadsto} y$ in π, $x \overset{\sigma'}{\leadsto} y$ in π'*

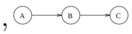
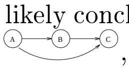
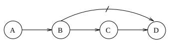
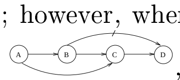
That is, the supersequence contains at least all of the nodes of the subsequence and the nodes occur in the same relative order (a subpath is a special kind of subsequence in which the nodes are in the same consecutive order).

Part of the task of an inheritance system is to define procedures for adjudicating amongst these multiple conflicting and redundant paths between two nodes. Reasoners vary when settling conflict in terms of their degree of skepticism. A credulous reasoner establishes multiple expansions when faced with conflict, and in each expansion only one of the conflicting paths is sanctioned. Skeptical reasoners always resolve to a single expansion, though they vary in the number of paths that they admit to the expansion. Skeptical reasoners vary in their degree of skepticism, the more skeptical the reasoner, the smaller the size of possible expansions. The preferred paths which are chosen are said to be *permitted*. If a path is permitted in a network, then we say that the expansion of the network contains the *implicit link* between the endpoints of the path. A reasoner is *stable* if its expansions of a network are identical with the expansions of the same network with any number of its implicit links added.

1.2 Variations

Touretzky et al. (1987) identify a number of choice points in the design of a path based inheritance reasoning system. Different decisions on these points cause different reasoners to sanction different conclusions about the same networks. The axes of variation considered herein are redundancy, preemption, ambiguity, and path construction.

1.2.1 Redundancy.

The priority of explicit links has been identified as a controversial issue with respect to stable reasoning (Boutilier, 1989). But, since Touretzky's thesis (Touretzky, 1986) it has been accepted in the literature that certain topologically identifiable links in inheritance networks are redundant since they convey no information that is not already present in longer paths, through transitivity. Essentially, certain explicit links are deemed redundant with respect to implicit links, in particular, when the explicit link expresses the same conclusion that can be drawn from a longer path. In terms of graph topology, a 'redundant' link is a direct link that connects the endpoints of another path with the same polarity. For instance, a reasoner that implements transitive inference would likely conclude from this network, , that As are normally Cs. Thus, this related network, , is deemed to contain no additional information. In a reasoner that incorporates transitivity, the conclusions implicit in both graphs are the same regardless of the information-supplying status of the redundant link; however, when graphs like these are embedded in larger networks such as, , and , a reasoner that assumes redundancy (as nearly all path-based reasoners do) will reach the same conclusions for both graphs, but reasoners that assume each direct link conveys novel information may sanction different conclusions for each graph. Touretzky (1986) argued that shortest path reasoning is unsound because of the instability that results from adding an implicit link to a network and reasoning with the result.[5] The conditional logic approach to defaults of Delgrande (1988, 1990) would find the second of the above two graphs ambiguous, and thus, Delgrande's approach is unstable (noncumulative). However, this is not a haphazard instability as in shortest path reasoning; rather, there is a strong position taken in distinguishing explicit and derived information.

[5]This property has been called the *cumulation property*: if γ is a network and l_1 is an implicit link between the endpoints of a permitted path π in γ ($\gamma \vdash \pi$ or $\gamma \vdash l_1$), then $\gamma \vdash l_2$ iff $\gamma \cup l_1 \vdash l_2$.

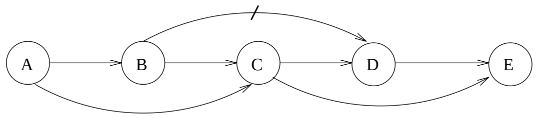


Figure 3: Stability/Redundancy

However, this alternative analysis of 'redundant' links is not the one that Boutilier (1989) adopts. Rather, he points out that the literature is not uniform in its topological identification of redundancy. For example, consider the graph in Figure 3; it is like the network just mentioned, with the exception of extension to the node E and another redundant link from $c \longrightarrow e$. Without the link $c \longrightarrow e$, H90 and Boutilier's (1989) system would conclude nothing about whether As are Es, since they do not admit chaining beyond a negative link, and specificity and redundancy for both mean that As are not Ds. However, Horty et al. (1990) define H90 so that given the network in Figure 3, it will conclude that As are in fact Es, while Boutilier's (1989) system reaches no conclusion for the same reason that it would without the link, under the assumption that the link $c \longrightarrow e$ is redundant. Both assume a topological definition of redundancy, but Boutilier's is simpler: for him a link is redundant if it spans a longer path of exactly the same polarity, but in H90, a link is redundant if it spans a longer path of the same polarity, and (if it is positive) no negative link contributes information about one of the nodes in the path with respect to which the link is redundant.[6] Clearly, H90 is not stable with respect to links that Boutilier would identify as redundant. Horty et al. (1990) find this an acceptable property since it gives their form of inheritance reasoning "a sensitivity to the structure of arguments that is difficult to achieve in deductive systems," while Boutilier finds it *ad hoc* and unintuitive.

1.2.2 On-Path/Off-Path Preemption

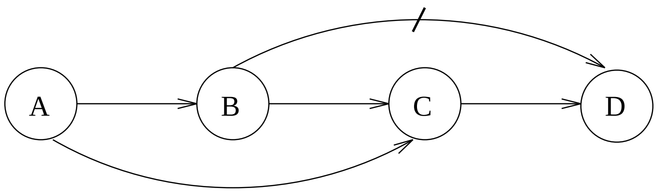


Figure 4: As are normally not Ds

Most inheritance reasoners include some form of preemption: a mechanism in which more-specific information from some path overrides information from some other path (see §2.2 for formal detail). A reasoner can be *on-path* or *off-path* preempting. In *on-path* preemption, the first node of a preempting link must occur on the preempted path, as in Figure 4 in which $b \not\longrightarrow d$ is a preempting link whose initial node occurs in the preempted path *abcd*. Because the path *acd* is deemed redundant with respect to the longer preempted path, it too is preempted, giving the conclusion that *As are typically not Ds.* However, for the related graph in Figure 5, the path *aecd* is not redundant with respect to *abcd*, and since b does not occur in *aecd*, using on-path preemption the negative path does not override the positive nor vice versa. A skeptical reasoner using on-path preemption will reach no conclusion about whether *As are Ds* or not. In a reasoner that uses off-path preemption, preemption can occur even when the preempting link does not occur on the

[6]The additional condition is not in place for the symmetric case of redundant negative links; since by the very assumption that it is a redundant negative link any positive link that would contribute conflicting information about one of the intervening nodes can at most be part of a chain that culminates in the negative link that forms the negative path that makes the other link redundant.

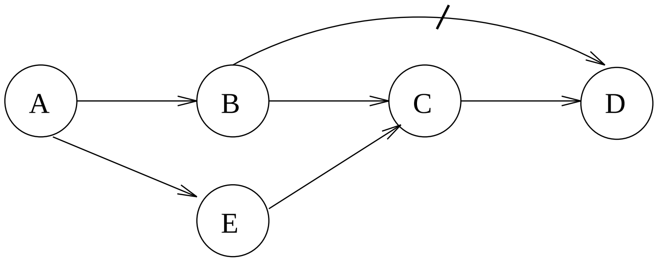


Figure 5: Ambiguous?

preempted path without the preempted path being redundant. Thus, given the network shown in Figure 5, a reasoner that uses off-path preemption will conclude that As are not Ds.

Touretzky's (1986) original system used on-path preemption, but of course there is much debate about which is most 'intuitive'. Sandewall (1986) argues that *aecd* should be considered redundant as well, and that the link $b \not\rightarrow d$ provides the most explicit information. Touretzky et al. (1987) provide an interpretation for the graph which they claim renders the conclusions of off-path preemption questionable (a = George, b = chaplain, c = man, e = marine, d = drinks beer) since George's being a marine provides the information that he isn't a typical chaplain. Nonetheless, they adopt off-path preemption in their specification of H90 (Horty et al., 1990). This debate is reconsidered by Al-Asady and Narayanan (1993) who observe that the argument against off-path preemption given by Touretzky et al. (1987) actually runs contra to the basic principle of path-based inheritance that more specific information should override more general information: information known about George by virtue of his being a man (that men typically drink beer) is more general than the information known about him by virtue of his being a chaplain. Al-Asady and Narayanan (1993) find fault with the George-network itself, and provide an alternative hierarchy in which the same problem does not emerge. Their own Exceptional Inheritance Logic (EIL) is an inheritance system that is off-path preempting. EIL separates exceptional information from environmental information not encoded in the network, and by representing both forms in the logic, they are able to restructure networks in accordance with their intuitions. They also provide unspecified mechanisms that 'condition' (cf. Touretzky, 1986) inheritance graphs by adding nodes which locate the source of exceptional information in particular locations relative to sources of 'typical' information that conflict. This idea is close to Cripps's (1987) idea of locating exceptional information explicitly within a network at a 'minimal distance' from the source of exceptional information. Unless there is an environmental property, EIL prefers explicit exceptional information

1.2.3 Degree of Skepticism

Faced with ambiguity, a reasoner can either choose a conclusion nondeterministically or can refuse to draw a conclusion. A variant of the former approach is called *credulous* reasoning and variations of the latter are *skeptical* reasoners. More exactly, a credulous reasoner creates a consistent expansion for each of the possible conclusions; thus, one path is permitted in some of the expansions and the other is permitted in the rest. This paper restricts attention to skeptical reasoners. Skeptical reasoners come in varying degrees of skepticism. A restricted skeptical reasoner reaches definite conclusions in some cases where a fully skeptical reasoner finds ambiguity. The difference is that a fully skeptical reasoner finds ambiguity in the existence of conflicting paths that have unpermitted subpaths, but a restricted skeptical reasoner requires that all subpaths of a path be permitted.

H90 is classified as a *restricted* skeptical reasoner because it does not allow ambiguities to be *cascaded.* Consider the network depicted in Figure 6. A restricted skeptical reasoner determines that the ambiguity about whether a can be classified as c is sufficient to prevent the construction

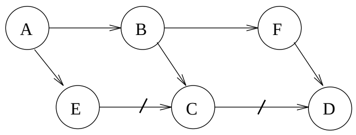


Figure 6: A Network with Potentially Cascaded Ambiguity

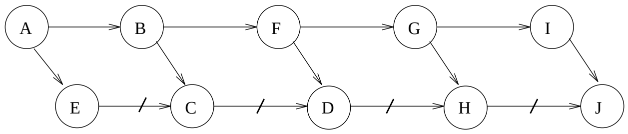


Figure 7: Parity Checking and Restricted Skepticism

of a path (abc/d) which can conflict with the conclusion that a is d. A fully skeptical reasoner (or *ideally skeptical* as defined by Stein (1989)) would conclude that the conflicting paths abc/d and $abfd$ make a's d-ness ambiguous. Thus a fully skeptical reasoner has a smaller expansion than the restricted skeptical reasoner. Stein (1989) points out the curious behavior of restricted skepticism when applied to larger networks like those in Figure 7. As before, the restricted skeptic (H90) concludes nothing about whether a is c, but does conclude that a is d. Similarly, because a is d, H90 can conclude nothing about whether a is h, and that leaves unblocked the conclusion that a is j. Thus, ambiguity does not cascade as in the fully skeptical reasoner, but it does propagate in an alternating fashion. That the alternation has an analogy in parity checking does not render the behavior remarkably intuitive. Touretzky et al. (1987) state that it is not known whether it is possible to compute the alternative fully skeptical expansion of a network without computing the intersection of all of the credulous expansions, since this is assumed to be computationally complex in the general case.

Kautz and Selman (1991) show that the membership of a literal in all expansions can be computed in $O(n^2)$ time in the size of the network for a system that does not incorporate preemption, and this means that the intersection of expansions can be computed in $O(n^3)$ time. However, for the more interesting case of skeptical reasoning with preemption, Kautz and Selman (1991) prove that computing the membership of a literal in all expansions is co-NP-complete. This is an extremely interesting result given that Stein provides a system that also computes the intersection of credulous expansions of networks with preemption and does so in polynomial time, and that H90 (Horty et al., 1990) also computes skeptical inheritance with preemption in polynomial time. The version of inheritance with specificity for which Kautz and Selman's (1991) co-NP-completeness holds uses off-path preemption, while Stein's (1989) system utilizes on-path preemption (which is intuitively easier to compute), and the off-path preempting H90 maintains tractability through its restricted rather than full skepticism.

1.2.4 Direction of Reasoning

Unfortunately, the source of complexity is not so easily diagnosed as that. Selman and Levesque (1989) argue that the decisive complexity factor in the case of H90 is that it uses upwards reasoning, and H90 has this feature in common with Stein's (1989) system. However, here we argue that their results are not decisive either.

Upwards reasoning corresponds to forward chaining with the point of origin being the queried node. Essentially, the definition given in Section 1.1 relied on forward chaining: direct links are paths; if π is a positive path, and $\textit{lastnode}(\pi) \rightarrow r$ is a link, and r does not occur as a node in π, then $\pi \rightarrow r$ is a path as well. Downwards reasoning as defined in the literature, however, relies on Touretzky's (1986) *double chaining* which can be stated as follows: direct links are paths; if $q \longrightarrow \pi$ is a path and $\pi \mathrel{-\!\!\ominus\!\!\!\rightarrow} r$ is a path then $q \longrightarrow \pi \mathrel{-\!\!\ominus\!\!\!\rightarrow} r$ is a path. The difference between these modes of reasoning can be understood with respect to the differing conclusions reached about particular graphs.

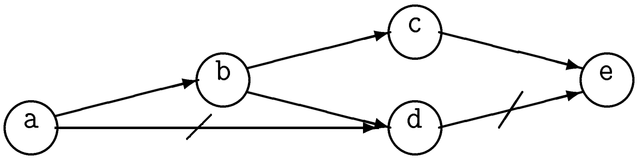


Figure 8: Preemption by $a \not\longrightarrow d$ Permits the Conclusion $a \longrightarrow e$.

Figure 8 (Touretzky et al., 1987) illustrates this point. We know that in a restricted or fully skeptical upwards reasoner a will be seen as an e, since the path abd/e, which could be a path that conflicts with $abce$, cannot be formed into a path; its initial path abd is preempted. Both the restricted and fully skeptical reasoners defined above determine that a potentially conflicting path must be unpreempted. On the other hand, a downwards skeptical[7] reasoner will begin chaining at e and will stop at the ambiguity about whether b is e. There is no way to chain beyond b to notice that the leading path is preempted. If the downwards reasoner were credulous there would be an expansion in which $abce$ is held, and one in which bd/e is held, but which contains no information at all about a's e-ness since a/d preempts bd and since $a/d/e$ is not a well-formed path. Note that the intersection of these credulous expansions is empty. If fully skeptical reasoners are defined as the intersection of upwards and downwards credulous expansions, this could be done simply by giving the definition of a downwards skeptical reasoner. Touretzky et al. point out the differences between upwards and downwards chaining (Touretzky et al., 1987) without providing a thorough explanation of why they exist.

As a preliminary step to understanding the difference, notice the topological similarity between the network depicted in Figure 8 from the last paragraph and the network shown in Figure 6 during the discussion of degrees of skepticism. The net of Figure 8 could be transformed into the net of Figure 6 by introducing an intermediate node in the link a/d (modulo node renaming). Note that the restricted and fully skeptical reasoners reach the same conclusions about Figure 8, but different conclusions about Figure 6. This is because Figure 8 represents preemption—more specific information preempts a conflicting chain of links. Based on the difference in behavior when upwards processing these two networks, we can say the following: an upwards reasoner[8] will sometimes resolve (in cases of preemption) an apparent ambiguity at some point "higher" in the network by using more specific information from lower in the network. "Higher" is just a conception of generality included in inheritance networks—individuals, the most specific entities, are at the bottom or left, and more general classifications proceed upwards or to the right.

In contrast, a downwards reasoner will never reach definite conclusions about a network where an upwards reasoner resolves to an ambiguity. That is, since downwards reasoners begin from general and reason towards specific, it will never resolve an apparent ambiguity using the more general information already noticed. Faced with ambiguity, it will never be able to chain further towards more specific information that could resolve the ambiguity of the whole path. In schematic terms, this can be understood from the nature of the chaining specification given at the start of this section. Essentially, when using double chaining all subpaths of a permitted path will also be permitted, but that is not necessarily so with upwards reasoning. Consider a single path

[7] It is perhaps slightly confusing to refer to generic skeptical reasoners without indicating their degrees of skepticism, but for the present purposes their degrees of skepticism are not relevant. The distinction made is between skeptical reasoners, which for arbitrary degrees of skepticism produce single expansions when faced with ambiguities rather, and credulous reasoners, which generate multiple expansions in those same circumstances.

[8] For the remainder of this section, "reasoner", in the context of ambiguity resolution, refers to a system that produces a single expansion—be it restricted or fully skeptical.

$\Pi = \pi \longrightarrow x = y \longrightarrow \omega$; Table 1 depicts the difference between upwards and downwards reasoning.

Upwards Chaining	$\underline{\quad\pi\quad}\ \underline{x}$ $\underline{y}\ \underline{\quad\omega\quad}$	$\vdash \pi x \not\supset \vdash \omega$
Downwards Chaining	$\underline{\quad\pi\quad}\ \underline{x}$ $\underline{y}\ \underline{\quad\omega\quad}$	$\vdash \pi x \supset \vdash \omega$

Table 1: Necessary Permission of Subpaths ($\Pi = \pi \longrightarrow x = y \longrightarrow \omega$.)

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

Table 2: Complexity of the various forms of defeasible inheritance.

This is in fact the key difference between complexity in upwards and downwards reasoners that Selman and Levesque (1993) identify as the source of intractability in downwards reasoners, and the reason why upwards reasoners (e.g. Stein's and H90) can be polynomial. Selman and Levesque (1989, p. 1144) summarize their results in the table reproduced in Table 2 (P stands for 'doable in polynomial time'). However, these results are not completely decisive as they depend on the particular sort of redundancy employed in H90, discussed above. It is not clear, for example, that a system based on Boutilier's (1989) notion of redundancy would still be susceptible to the same construction that Selman and Levesque (1993) use to prove NP-hardness of downward reasoning. In fact, their construction relies crucially on the conclusions made available through a link which Boutilier would consider redundant (hence making the network ambiguous), but which Horty et al. (1990) advocate as a desirable source of instability as a nonredundant link. Further, it is difficult to reconcile Stein's (1989) claim of providing a polynomial time algorithm for ideally skeptical inheritance with specificity with the further construction of Selman and Levesque (1993) which leads them to conjecture that "any reasonable preemption strategy" will make ideally skeptical inheritance intractable. This is a stronger claim than the one reported in the last section about the complexity of ideally skeptical inheritance. The reason for pointing it out here is that the stronger claim, in conjunction with Stein's claim of actually having a polynomial algorithm to compute exactly what Selman and Levesque (1993) conjecture is not possible, indicate that there is a likely interaction of other proof theoretic features of inheritance not discriminated in the Table 2. We have already mentioned the treatment of redundancy as one example, but there may be others.

1.2.5 Chaining Complexity

One aspect of downwards reasoning that Selman and Levesque (1993) cite as a source of complexity is double chaining. When using double chaining (see the depiction in Table 1), all subpaths of a permitted path will also be permitted. This is in comparison to upwards reasoning, for which that is not the case. We propose to explore an obvious alternative downwards reasoning strategy that is more symmetric to the upwards. Simple downwards is defined here as the intuitive symmetric variation: direct links are paths; if π is a path, and $r \longrightarrow firstnode(\pi)$ is a link, and r does not occur as a node in π, then $r \longrightarrow \pi$ is a path as well. Hereafter, 'downwards' reasoning will be assumed to mean 'simple downwards' reasoning, and what has been traditionally called downwards reasoning will be referred to as 'double chaining'. Table 3 shows how downwards reasoning compares to upwards reasoning and double chaining in connection to subpaths. Clearly, this form of downwards reasoning has the same property as upwards reasoning in that all subpaths of a path need not be permitted. However, this form of downwards reasoning has not been studied before in the literature.

Upwards Chaining	π x y ω	$\vdash \pi x \not\supset \vdash \omega$
Double Chaining	π x y ω	$\vdash \pi x \supset \vdash \omega$
Downwards Chaining	π x y ω	$\vdash y\omega \not\supset \vdash \pi$

Table 3: Necessary Permission of Subpaths ($\Pi = \pi \longrightarrow x = y \longrightarrow \omega$.)

To understand how downwards reasoning as defined here is different, as always, it is useful to consult an example network. Consider the network depicted in Figure 9. H90, an upwards reasoner, will conclude that the network is ambiguous about whether *As* are *Gs* because there is no permitted path to chain the link $E \longrightarrow G$ onto (because of the ambiguous subpath from B to E). Similary for double chaining. However, using downwards reasoning, although it is not possible to construct the path $BCDEG$ due to the same ambiguity, the path $CDEG$ is available, and it is possible, particularly in a system that uses a type of redundancy closely related to H90's, to chain the link $A \longrightarrow C$ to the left end of that path, yielding an uncontested conclusion that *As* are *Gs*. H90's version of redundancy actually has this link as redundant, as discussed above, but there is a strong sense in which downwards reasoning is an inverse of upwards reasoning (almost, in fact, like reasoning 'upwards' from the most general node in the related graphs in which the links go in the opposite direction), and recall from the earlier discussion that a topologically similar link was deemed non-redundant by H90 (contra Boutilier (1989)). Here it is important to point out just that *downwards* reasoning can have the property that Selman and Levesque (1993) identified as crucial to tractable path-based inheritance.

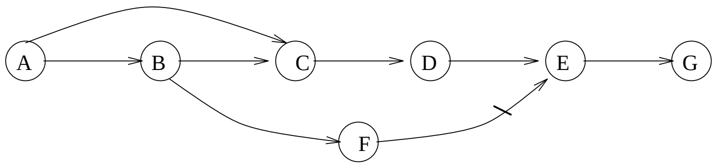


Figure 9: Downwards vs. Upwards vs. Double Chained Reasoning

1.3 The Rest of this Paper

The purpose of this work is to explain how these variations in reasoning styles can be anchored in the formal definition of a reasoner parameterized to realize these possibilities. To this end, H90, the restricted skeptical, upwards chaining, off-path preempting reasoner of Horty et al. (1990) is adopted as standard. Section 2 gives the declarative specification of this system provided by Vogel et al. (1993). Section 3 provides the parameterization of that specification which generates other reasoners, and in Section 4 complexity analysis is offered. The appendix gives a translation of the parameterized definitions into Prolog.

2 A Basic System

This section gives a logic based definition of the H90 system following Vogel et al. (1993). The plan is to define the notion of a permitted path, since it is the implicit link between the endpoints of a permitted path that expresses a conclusion of a reasoner with respect to a network.

2.1 Permission

The paths permitted in an arbitrary network by an inheritance reasoner are given here in terms of upwards construction of paths of increasing degree. It is also possible to state things in terms of downwards constructions (Touretzky et al., 1987) or double chaining (Touretzky, 1986). The formal definition follows:

Definition 2 (Permission)

1. Let π be a path.
   (a) If π is a direct link, then π is permitted.
   (b) If the degree of π is one then π is a direct link by the definition of degree, hence π is permitted.
2. Let π be a compound path of degree n. Assume that all permitted paths with degree less than n are known.
   (a) If π is a positive path then it has the form αz (i.e., $lastnode(\alpha) \longrightarrow z$ is a link in the network). The path α is positive, and the degree of α is less than n. The path π is permitted iff
      i. α is permitted,
      ii. $firstnode(\alpha) \not\longrightarrow z$ is not a direct link in the net,
      iii. All negative paths $\pi' = \alpha'/z$, where α' is a permitted positive path (with degree less than n) and $firstnode(\alpha) = firstnode(\alpha')$, are preempted.
   (b) If π is a negative path (it has the form, α/z), then π is permitted only under the conditions symmetric to those stated in (a). That is, iff:
      i. α is permitted,
      ii. $firstnode(\alpha) \longrightarrow z$ is not a direct link in the net,
      iii. All positive paths $\pi' = \alpha' z$, where α' is permitted positive path (with degree less than n) and $firstnode(\alpha) = firstnode(\alpha')$, are preempted.

Definition 2 follows the inductive structure of the path-based definition for H90 despite the change from their network notation. Consistent with the H90 definition, the above definition proceeds with upwards construction of paths of increasing degree. According to these definitions, if conflicting paths intersect only at their endpoints, they will cancel each other. Cancellation is stipulated by condition (iii) on the permission of π in Definition 2. This condition states that π is permitted only if all conflicting paths are preempted. Since π has degree n, we know of all paths which could conflict with π. By the definition of degree, none of the conflicting paths is longer than n. The degree of a path becomes significant only during the examination of compound paths for the existence of conflicting paths. Conflicting paths are handled trivially in the case of direct links. All direct links are sanctioned as paths through a network, even conflicting links. If a direct link conflicts with a compound path then the definition of preemption is satisfied and the direct link preempts the compound path. In the case of conflicting compound paths, it is known that none of the conflicting paths has a degree greater than n, and all shorter paths between the same endpoints are known, because Definition 2 proceeds on the basis of increasing degree.

2.2 Preemption

Path preemption allows (more specific) information that is contained in a direct link to override conflicting information in a (more general) compound path. This topological ordering of paths is called the *inferential distance ordering* (Touretzky, 1986).

Only direct conflicting links can preempt other paths, although a preempting link may be part of a longer path.

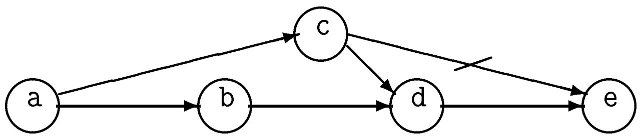


Figure 10: Path *abde* is off-path preempted by the link $c \not\rightarrow e$.

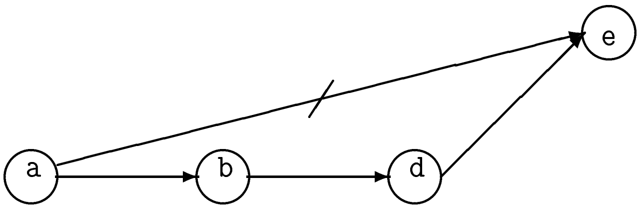


Figure 11: Link $a \not\rightarrow e$ preempts the path *abde*.

Definition 3 (Preemption) Let π and π' be positive paths, and let p and y be nodes. A positive path πy is *preempted* by a link $p \not\rightarrow y$ or a negative path π / y is *preempted* by a link $p \longrightarrow y$ if there exists a permitted path π' such that

1. $firstnode(\pi) = firstnode(\pi')$,
2. $lastnode(\pi) = lastnode(\pi')$,
3. p occurs in π', and
4. $p \neq lastnode(\pi')$.

When a path is preempted by a link $p \mathrel{-\!\!\circ\!\!\rightarrow} y$, it is also said that the *node* p preempts the path. In the case where $p = lastnode(\pi')$ (i.e. where condition 4 is violated), we would have two conflicting links $p \longrightarrow y$ and $p \not\rightarrow y$. Instead of allowing preemption, the network is said to contain an inconsistency.

An example of preemption is presented graphically in Figure 10, in which a positive path is preempted by a negative link. Matching the definition to the network in Figure 10, π corresponds to the path *abd*, π' to *acd*, p to c, and y to e. The endpoints of π and π' coincide, $p \not\rightarrow y$ is a link in the network, so *abde* is (off-path) preempted by the link $c \not\rightarrow e$. Essentially, ac/e is said to be more specific than both *acde* and *abde*. Another straightforward example of preemption is given in Figure 11: there is a similar matching between nodes and links of the network and the definition, except that in this case π and π' are identical.

Compound paths which conflict are subject to cancellation. The difference between cancellation and preemption is that neither conflicting path is permitted after cancellation, but preemption does permit one of its conflicting paths (namely, the path consisting of the direct link). Consider the network depicted in Figure 12 which has a topology similar to Figure 10. Although paths π and π' exist whose endpoints coincide, no π' exists that contains a node participating in a preempting link. Neither ac/e nor *abde* is favored over the other. In the network shown in Figure 10, preemption *resolves* an apparent ambiguity, but this does not happen in the network of Figure 12. Since this is a skeptical reasoner the paths cancel each other—neither is permitted. A credulous reasoner would resolve to two extensions from the network, one in which a's are e's and another in which a's are not e's.

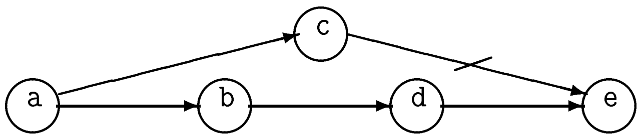


Figure 12: Paths ac/e and $abde$ cancel each other.

3 Other Systems

The formal definitions of path permission can be altered to provide alternative styles of inheritance reasoning. First it is helpful to restate Definition 2 in more schematic terms:

Definition 4 (Permission)

1. Let π be a path.
   (a) If π is a direct link, then π is permitted.
   (b) If the degree of π is one then π is a direct link by the definition of degree, hence π is permitted.
2. Let π be a compound path of degree n. Assume that all permitted paths with degree less than n are known.
   (a) If π is a positive path then it has the form αz. The path α is positive. The path π is permitted iff
      i. α is permitted,
      ii. $firstnode(\alpha) \not\longrightarrow z$ is not a direct link in the net,
      iii. no conflicting path matters.
   (b) If π is a negative path (it has the form, α/z), then π is permitted iff:
      i. α is permitted,
      ii. $firstnode(\alpha) \longrightarrow z$ is not a direct link in the net,
      iii. no conflicting path matters.

Definition 5 (Conflicts that Matter) Restricted Skepticism, Off-Path Preemption
No path conflicting with π matters iff

1. *if π has the form, αz, and all negative paths $\pi' = \alpha'/z$, where α' is a permitted positive path and $firstnode(\alpha) = firstnode(\alpha')$, are Off-Path preempted.*
2. *if π has the form, α/z, and all positive paths $\pi' = \alpha' z$, where α' is a permitted positive path and $firstnode(\alpha) = firstnode(\alpha')$, are Off-Path preempted.*

Off-path preemption is exactly that sort which was specified in Definition 3. In the sections that follow we will modify this schematic definition to illustrate the relationships among the various proof theories for inheritance under consideration. The relationships among these styles of reasoning have not before been made explicit. Later in the paper, a Prolog implementation is offered which has the advantage of perspicuous encoding since there is a direct textual relationship between the definitions given here and the Prolog code. The Prolog implementation of the Off-Path preempting, restricted skeptical reasoner (H90) was first presented by Vogel et al. (1993)

3.1 On-Path Preemption

Using on-path instead of off-path preemption, the nodes of the preempting link must occur on the path that is being preempted (as is the case depicted in Figure 11). On-path preemption is rather directly related to off-path preemption as just a more restricted form of preemption. Without changing the structure of Definition 3 we can define on-path preemption by adding the additional restriction that $\pi' \subseteq \pi$. Note that Vogel et al. (1993) erroneously state that to obtain on-path preemption, the relationship between π' and π should be equality. However, as discussed below, that restriction actually implements an alternative treatment of redundant links.

Definition 6 (On-Path Preemption) Let π and π' be positive paths, and let p and y be nodes. A positive path πy is *preempted* by a link $p \not\longrightarrow y$ or a negative path π/y is *preempted* by a link $p \longrightarrow y$ if there exists a permitted path π' such that

1. $\pi \subseteq \pi'$
2. $firstnode(\pi) = firstnode(\pi')$,
3. $lastnode(\pi) = lastnode(\pi')$,
4. p occurs in π', and
5. $p \neq lastnode(\pi')$.

Recall that reasoners which use off-path preemption conclude from the network in Figure 10 that *As* are not *Es*. On-path preemption leads to ambiguity in that network because c, the preempting node, does not occur on the path *abde*; thus that positive path conflicts with the negative path. Because preemption is about letting paths of opposite polarity override each other, using on-path preemption yields ambiguity where off-path preemption allows a definite conclusion.

3.2 Redundancy

3.2.1 Redundancy and Preemption

The literature assumes that certain topologically defined links are redundant and that a path through a set of redundant links should be preempted if the longer path with respect to which redundancy is determined is also preempted.

Definition 7 (Redundancy) *A path π is redundant with respect to a longer path π' if and only if they have the same polarity and:*

1. $\pi \subseteq \pi'$;
2. firstnode(π) = firstnode(π')
3. lastnode(π) = lastnode(π')

Intuitively, Definition 7 says that a path π is redundant with respect to π' if π is a subsequence of π' and they have the same endpoints. This just means that π contains direct links where π' has compound paths, so it is assumed that the direct links each contain no more information than is already in the corresponding longer path.

Thus, Definition 6 could have been given as follows:

Definition 8 (On-Path Preemption) Let π and π' be positive paths, and let p and y be nodes. A positive path πy is *preempted* by a link $p \not\longrightarrow y$ or a negative path π/y is *preempted* by a link $p \longrightarrow y$ if there exists a permitted path π' such that

1. π is redundant with respect to π'
2. p occurs in π', and

3. $p \neq lastnode(\pi')$.

However, under the assumption that each link in a network conveys first-class information, preemption can still occur. Obviously, in a system without redundancy there is no need to stipulate preemption for paths that are redundant with respect to preempted paths. This means, in the above definition that π and π' are identical. Thus, on-path preemption for inheritance without redundancy reduces to the following:

Definition 9 (On-Path Preemption without Redundancy) Let π be a positive path, and let p and y be nodes. A positive path πy is *preempted* by a link $p \not\rightarrow y$ or a negative path π/y is *preempted* by a link $p \longrightarrow y$ iff:

1. p occurs in π', and
2. $p \neq lastnode(\pi')$.

Using this definition, it is possible to conclude of the network in Figure 13 that *As* are not *Ds*, but not for the network in Figure 14. That is, the path *abcd* is still preempted, but the path *acd* is

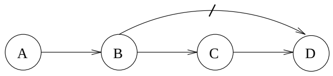


Figure 13: A Simple Inheritance Network

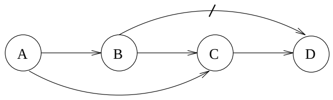


Figure 14: The Link $a \longrightarrow c$ Is Redundant

not since b, the preemptor, does not occur in *acd*. Since *acd* and ab/d conflict, there is ambiguity, therefore it cannot be concluded that *As* are not *Ds*.

It is not coherent to define off-path preemption without redundancy since redundancy is actually just a special case of an off-path preemption, as the last example illustrates. This entails that in terms of relative numbers of paths deemed preempted, there is a total ordering such that off-path preemption is strictly more preemptive than on-path preemption which is in turn strictly more preemptive than on-path preemption without the assumption of redundancy. Eliminating preemption altogether is the only form of reasoning less preemptive than on-path preemption without redundancy. Thus, there are four possible forms of preemption that can be slotted into the definition of 'conflicts that matter'.

3.2.2 Redundancy and Permission

A final remark on redundancy is necessary in order to parameterize the definition of permission fully with respect to redundancy. Recall from Section 1.2.1 that there are conflicting views (Boutilier, 1989) about how uniformly stable inheritance reasoners should be with respect to redundant links. For example, given the network in Figure 15, H90 would conclude *As are typically Es*, although if the network did not contain $c \longrightarrow e$, H90 would reach no conclusion about whether *As* are *Es* because in the smaller network the subpath *abcd* is not permitted since $a \overset{b}{\not\rightarrow} d$, and its paths are assumed not to have nonfinal negative links. Nonetheless by Definition 7, $c \longrightarrow e$ is redundant with respect to the path *cde*. Boutilier (1989) argues that because $c \longrightarrow e$ is redundant, there should be no conclusion about whether As are Es, just as $a \longrightarrow c$ does not affect the conclusion that As are not Ds. He asserts that the presence of neither link should change the allowable conclusions. The other possibility, of course, is to accept the argument that both links be treated uniformly but to treat them both as non-redundant.

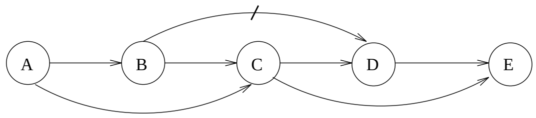


Figure 15: Are both $a \longrightarrow c$ and $c \longrightarrow e$ Redundant?

Definition 10 (Informational Redundancy) *A path π in a network Γ is informationally redundant with respect to a longer path π' if and only if they have the same polarity and:*

1. $\pi \subseteq \pi'$;
2. $\mathrm{firstnode}(\pi) = \mathrm{firstnode}(\pi')$;
3. $\mathrm{lastnode}(\pi) = \mathrm{lastnode}(\pi')$;
4. $\forall \langle a, b \rangle$ *such that* $a \overset{\circ}{\rightarrow} b \in \pi, \exists \sigma$ *such that* $a \overset{\sigma}{\leadsto} b \in (\Gamma - (\pi \setminus \pi'))$

The definition of informational redundancy is more complicated than simple topological redundancy (Definition 7) in having an extra final condition which basically states that all of the links in the redundant path correspond to permitted paths (the implicit links) in the network even when reasoning is performed on the network minus those links. Using this definition and considering the network in Figure 15 again, it is clear that *ace* is redundant with respect to *abcde* (since both $a \overset{\emptyset}{\leadsto} c$ and $c \overset{\emptyset}{\leadsto} e$ are implicit links, permitted even when the network doesn't contain them explicitly). Using H90 on the network shown in Figure 16, the path fae is informationally redundant with respect to both $face$ and $fabcde$. However, using Boutilier's (1989) system fae is informationally redundant with respect to neither $face$ nor $fabcde$; this is because his system does not sanction a permitted path between a and e. Therefore in the network without the explicit link $a \longrightarrow e$ there will not be an implicit link between the two nodes either. Clearly, informational redundancy ignores fewer paths than redundancy simpliciter.

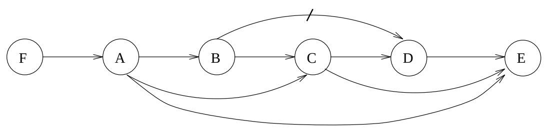


Figure 16: Is $a \longrightarrow e$ Redundant?

It is possible to amend the definition of conflicts that matter when chaining a path together to determine if it is permitted. Definition 11 incorporates the alternative view of redundancy of Boutilier (1989), but casts his system into the terms used within this paper rather than utilizing the double-chaining mechanism that he used to achieve the same effect.

Definition 11 (Conflicts that Matter: Boutilier (1989)) *No path conflicting with π matters iff*

1. *if π has the form, αz and there is no unpermitted path with respect to which αz is informationally redundant, then all negative paths $\pi' = \alpha'/z$, where α' is a permitted positive path and $firstnode(\alpha) = firstnode(\alpha')$, are preempted.*

2. *if π has the form, α/z and there is no unpermitted path with respect to which α/z is informationally redundant, then all positive paths $\pi' = \alpha' z$, where α' is a permitted positive path and $firstnode(\alpha) = firstnode(\alpha')$, are preempted.*

The apparent circularity between the definitions of informational redundancy and the conflicts clause of the preemption definition is not vicious because the appeal to preemption in the informational redundancy definition is with respect to a smaller graph which has links under scrutiny removed. Informational redundancy is still a topological definition — it employs only a very weak form of information and is sensitive to the other constraints on reasoning within some system. That is, it basically asserts that a link which is redundant simpliciter may not be redundant when considered by a reasoner which would not have actually licensed an implicit link between the endpoints of the longer path. If the longer path is not permitted, then a link which spans its endpoints actually does add information to the network. However, this is still quite a weak form of information to consider. Given the network in Figure 15, both $a \longrightarrow c$ and $c \longrightarrow e$ are informationally redundant, but since H90 differentiates them, ignoring $a \longrightarrow c$ to conclude $a \overset{b}{\not\rightsquigarrow} d$ but regarding $c \longrightarrow e$ as evidence for the conclusion $a \overset{bc}{\rightsquigarrow} e$, there is reason to feel there should be an intuitive way of defining redundancy such that $a \longrightarrow c$ is redundant but $c \longrightarrow e$ is not. As it stands, both links are informationally redundant, and certain topological configurations that the redundant link can exist in have an impact on whether the reasoner is stable.[9]

It should be clear that there is no point in integrating informational redundancy into off-path preemption, because the notions are antithetical (the basic idea of off-path preemption being that even compound paths, essentially 'redundant links' with intervening nodes can be redundant). Integrating informational redundancy into on-path preemption, as in Definition 12 yields a quite skeptical system. Essentially, there will not even be preemption as in Figure 10 since π and π' are identical in that case, and a lone path cannot be informationally redundant (if its links were removed there would be no other path to license them as implicit links). While paths like *acd* in Figure 14 are preempted using this definition, by virtue of the fact that *acd* is informationally redundant with respect to *abcd*, the path *abcd* is not preempted (and therefore the network is ambiguous). The difference between this and the previous application of informational redundancy is that here it is integrated into the definition of preemption, but there it was just used as a check on allowable conclusions. Perhaps it seems futile to define a version of on-path preemption which does not actually admit preemptions in the usual sense. However, it is useful to have this system as a point in the spectrum between considering all paths as containing first class information (as in some scheme in which numbers of paths of each polarity determine the final conclusion) and other systems in which preemptions occur without regard to the number of conflicts. That is, the systems defined so far have used preemption to reach definite conclusions and assumed that some paths can be discounted, disregarding the numbers of each; the system just defined still discounts some paths as well as numbers in conflicting polarities, but it does not preempt anything to reach a definite conclusion; systems defined later will not discount any path, will consider relative cardinalities and will reach definite conclusions where possible.

Definition 12 (Preemption with Informational Redundancy) Let π and π' be positive paths, and let p and y be nodes. A positive path πy is *preempted* by a link $p \not\longrightarrow y$ or a negative path π/y is *preempted* by a link $p \longrightarrow y$ if there exists a permitted path π' such that

1. π is informationally redundant with respect to π'

2. p occurs in π', and

3. $p \neq lastnode(\pi')$.

[9]Recall that stability is the property of a reasoner reaching the same conclusions with or without redundant links explicitly present.

3.2.3 Summary

This section has outlined three forms of redundancy which exist in a decreasing order of dismissiveness of paths. Redundancy in H90 effectively includes certain compound paths through the way off-path preemption is specified. On-path preemption utilizes a slightly tighter form that labels paths as redundant solely if they contain redundant links, where redundant links are just those that span a longer path of the same polarity. Informational redundancy is a more restricted version in which the paths spanned by redundant links must be permitted. Incorporating informational redundancy into preemption yields a form of preemption which preempts only redundant paths but not the longer paths with respect to which the smaller ones are redundant. Because preemption is invoked only in contexts where there are paths of conflicting polarity, this implies that using preemption with informational redundancy will yield fewer definite conclusions than preemption based on strictly topologically defined preemption. Preemption without the assumption of redundancy leads to a similar state of affairs. Figure 17 illustrates the consequences of fitting these various definitions together in terms of the paths preempted by the various systems. H90 preempts the most paths, and hence more often reaches definite conclusions (this of course is dependent upon the network reasoned about, for some networks all four will reach the same conclusions). Using on-path preemption instead of off-path preemption in H90 yields a system in which only direct links (and the paths comprised of them) can be redundant—such a system preempts strictly more paths than either on-path preemption without the assumption of redundant links and on-path preemption with informational redundancy. The last two systems are not in a subsumption relation as has been seen in the examples above: the former still allows paths to be preempted when there are no 'redundant links' but yields ambiguity in other cases, and the latter preempts paths comprised of redundant links, but does not allow paths to be preempted that do not contain redundant links, yielding ambiguity.

A path π in a network Γ is redundant with respect to a longer path π' if and only if they have the same polarity and:	
Path/Link Redundancy:	$\mathit{firstnode}(\pi) = \mathit{firstnode}(\pi')$, and $\mathit{lastnode}(\pi) = \mathit{lastnode}(\pi')$;
Link Redundancy:	$\pi \subseteq \pi'$, and $\mathit{firstnode}(\pi) = \mathit{firstnode}(\pi')$, and $\mathit{lastnode}(\pi) = \mathit{lastnode}(\pi')$;
Informational Redundancy:	$\pi \subseteq \pi'$, and $\mathit{firstnode}(\pi) = \mathit{firstnode}(\pi')$, and $\mathit{lastnode}(\pi) = \mathit{lastnode}(\pi')$, and $\forall \langle a, b \rangle$ such that $a \multimap\!\!\rightarrow b \in \pi, \exists \sigma$ such that $a \overset{\sigma}{\rightsquigarrow} b \in (\Gamma - (\pi \setminus \pi'))$.

Recapitulating Redundancy.

3.3 Skepticism

3.3.1 Fully Skeptical Reasoning

Figure 6 depicted a network which differentiates restricted skeptical and fully skeptical reasoners. For convenience, this figure is reproduced in this section as Figure 18.

The reasoner defined in the previous section was a restricted skeptical reasoner: ambiguities do not cascade through a network. Restricted skepticism is independent of a choice of preemption methods. In the network of Figure 18 we can conclude that a is f because there is no path permitted which conflicts with $abef$. The chain of links abd/f, which appears as if it could conflict, is not actually permitted because abd is not permitted, and it is impossible to build a path by extending a chain that is not itself a permitted path. In Definition 2, this was stipulated inside condition (iii) which determines what it means for a conflicting path to matter (Definition 5): α', part of the conflicting path π' must be a variable over positive paths.

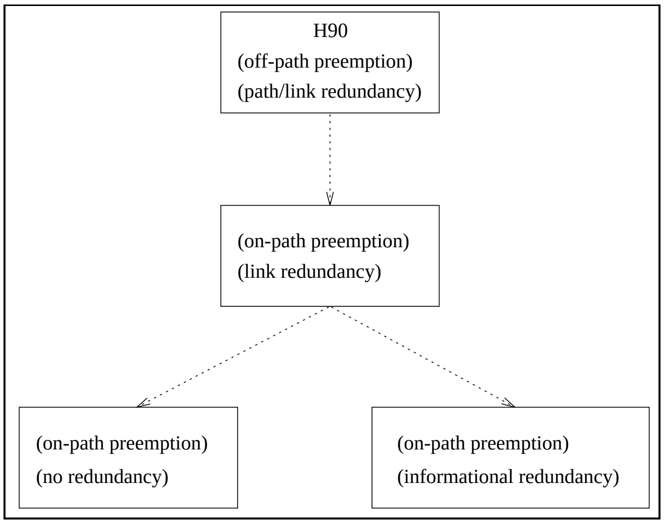


Figure 17: Paths Preempted by Various Systems

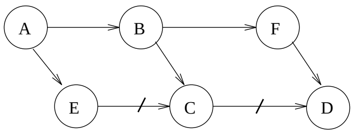


Figure 18: A Network with Potentially Cascaded Ambiguity

A fully skeptical reasoner demands that ambiguities cascade through a network. Thus, a fully skeptical reasoner should reach no conclusion about a's f-ness, because a non-preempted negative path conflicts with the positive path $abef$. In the definition, α' is a variable over positive paths without regard for whether α' is actually permitted, so long as it is not preempted by more specific information.

Definition 13 (Conflicts that Matter: Subpath Credulity) *No path conflicting with π matters iff*

1. *if π has the form, αz, and all negative paths $\pi' = \alpha'/z$, where α' is a positive path and $firstnode(\alpha) = firstnode(\alpha')$, are preempted.*

2. *if π has the form, α/z, and all positive paths $\pi' = \alpha' z$, where α' is a positive path and $firstnode(\alpha) = firstnode(\alpha')$, are preempted.*

This difference is orthogonal to the question of which form of preemption gets used, therefore the specification of which conflicts matter is referred to as *subpath credulity.* That is, the restricted skeptical reasoner is skeptical about ambiguous subpaths, and ignores them. A reasoner that uses subpath credulity will assume that ambiguous subpaths are actually positive when considering the potential ambiguity of longer paths. The effect is that ambiguity cascades through the network to the longer path, and in the end, it's the subpath credulous system that concludes indeterminacy more than the restricted skeptical reasoner. For want of a better term this is called fully skeptical reasoning, but since either form of preemption can be used with subpath credulity and since on-path preemption will result in more indeterminacy than off-path preemption, there is strong motivation for calling a fully skeptical reasoner that uses on-path preemption even more skeptical than one that uses off-path preemption.

3.3.2 Ideally Skeptical Reasoning

One good reason for calling fully skeptical reasoning, in which ambiguities cascade, something other than "fully skeptical" is that it is not coextensive with ideally skeptical reasoning, even though it seems to be at first glance. Ideally skeptical reasoning is taken to be the intersection of credulous expansions (see Stein, 1989). However, there can be elements of the intersection of all credulous expansions of a network that are licensed by unique paths which do not occur in all credulous expansions. For example, Stein (1989) provides a network and intepretation as depicted in Figure 19. Stein (1989, p.1156) points out, "Whether a seedless grape vine is a fruit plant or an arbor plant, it is certainly a plant!" That is, $a \rightsquigarrow h$ is an element of each credulous expansion. However, it is by virtue of $a \overset{bef}{\rightsquigarrow} h$ in the expansions in which seedless grape vines are not fruit plants (ac/d is the permitted path, hence abd/f isn't available to conflict with the subpath $abef$), and by virtue of $a \overset{bdg}{\rightsquigarrow} h$ in the expansions in which seedless grape vines are fruit plants. Restricted skepticism does get the desired conclusion here, but does not correspond to ideally skeptical reasoning as discussed in Section 1.2.3 (because it licenses some conclusions that ideally skeptical reasoning would not). Surprisingly at first glance, fully skeptical reasoning does not capture ideally skeptical reasoning either, as can be seen from the fully skeptical expansion of the network shown in Figure 19: because the subpath abd is not permitted (due to ambiguity), the path $abdgh$ is not permitted; because the subpath $abef$ is not permitted (due to ambiguity), the path $abefh$ is not permitted. Thus, there is not a way using the fully skeptical reasoner to build a path that licenses the implicit link $a \rightsquigarrow h$.

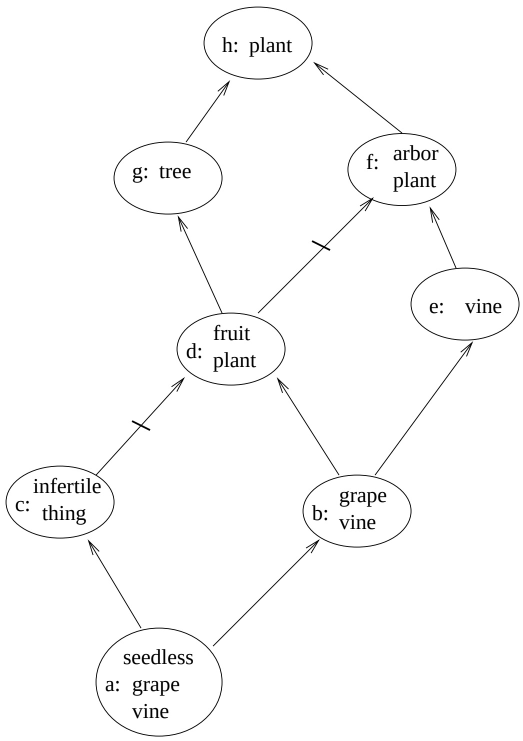


Figure 19: Ideally Skeptical Reasoning: $a \rightsquigarrow h$

Stein (1989, p.1156) concludes:

> There are facts which are true in all credulous extensions, but which have no justification in the intersection of those extensions. This is why we cannot generate a "skeptical extension"—no *particular* set of edges of Γ from *seedless grape vine* to *plant*

is in every credulous extension, so no such path can be in the "skeptical extension." Thus every path-based approach to skeptical inheritance will always be either unsound or incomplete with respect to ideally skeptical inheritance. We can only compute the always-true inferences by, in effect, reasoning about all of the credulous extensions. Fortunately, in acyclic hierarchies, such reasoning is tractable.

Stein (1989) proceeds to build a truth-maintenance system which does reason about all of the credulous extensions.

Note that in spite of the subpath ambiguity it is possible even in the fully skeptical reasoner to conclude $e \overset{f}{\rightsquigarrow} h$ and $d \overset{g}{\rightsquigarrow} h$. That is, the property of being a plant inherits all the way down to node b, regardless of the ambiguity between b and f. The only reason it doesn't inherit all the way to a, of course, is because of the subpath ambiguity between a and d. However, note that from the vantage point of node b, if h is inherited all the way to d and to e, then even if there is a subpath ambiguity about whether $b \rightsquigarrow f$ holds it is nonetheless reasonable to conclude that $b \rightsquigarrow h$, since b can get to h through either d or e. Contrast this with the network in Figure 20, which is just a subset of the one from Figure 19. The smaller network does license $e \overset{f}{\rightsquigarrow} h$, however there is no other information about h to reason with in the credulous expansion that contains $b \overset{d}{\not\rightsquigarrow} f$. Essentially, all that is required is an inheritance version of reasoning by cases to justify $b \rightsquigarrow h$ in the first example, and that principle's inapplicability in the second case distinguishes the two. In the absence of an explicit preemption between a and h and given that $b \rightsquigarrow h$ and $a \overset{\emptyset}{\rightsquigarrow} b$, it is reasonable to conclude $a \rightsquigarrow h$. Moreover, it is reasonable to even associate that conclusion more directly with a path: $a \overset{bef}{\rightsquigarrow} h$. This is because while $abdgh$ suffers from subpath ambiguity, the fact that $e \overset{f}{\rightsquigarrow} h$ still gets information from h to b, therefore $abefh$ is the empowering path. Admittedly, this is a weak relationship, since it too suffers subpath ambiguity, but that ambiguity is already overridden by the case-reasoning.

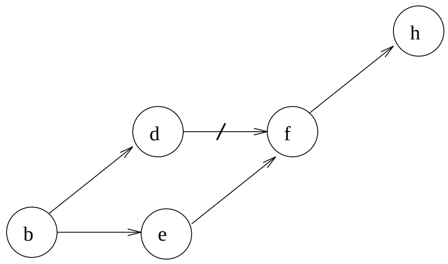


Figure 20: Ideally Skeptical Reasoning: $\neg\exists(b \rightsquigarrow h)$

Definition 14 (Conflicts that Matter: Subpath Credulity + Cases) *No path conflicting with π matters iff*

1. *if π has the form, αz, and all negative paths $\pi' = \alpha'/z$, where α' is a positive path and $firstnode(\alpha) = firstnode(\alpha')$ are (i.) preempted or (ii.) for each node i in α where $i \neq$ firstnode(α), $\exists\omega$, $i \overset{\omega}{\rightsquigarrow} z$.*

2. *if π has the form, α/z, and all positive paths $\pi' = \alpha' z$, where α' is a positive path and $firstnode(\alpha) = firstnode(\alpha')$, are (i.) preempted or (ii.) for each node i in α where $i \neq$ firstnode(α), $\exists\omega$, $i \overset{\omega}{\rightsquigarrow} z$.*

Definition 14 specifies this new form of skeptical reasoning, which implements reasoning with cases, here in the context of cascaded ambiguity. Essentially, the only difference is that for all paths that conflict with the subpath under extension (into a longer chain), each node but the first one already permits the endpoint of the chain. Thus, this definition incorporates reasoning by cases into skeptical path-based reasoning, and we argue that it is complete with respect to ideally skeptical reasoning. Stein's point, quoted above, still stands, since by sanctioning paths that yield the implicit link as permitted, it licenses paths that are not contained in all credulous expansions, and is thus not sound with respect to ideally skeptical reasoning. It would be possible to generalize the system along the lines of Veltman's (1994) Update Semantics to distinguish syntactically between explicit and inferred links, for instance by incorporating abstraction over paths in between endpoints ($\rightsquigarrow$). Thus far we have been using the symbolism $i \rightsquigarrow j$, without explicit mention of a path as in $i \overset{\pi}{\rightsquigarrow} j$, as an abbreviation for "a path permitted between i and j". Definition 15 formalizes this, but notice that it does not have an 'only-if' condition: a definition of *weak permission* (see Definition 16) supplies the other conditions on the weak permission of an implicit link between the endpoints of a path that is not itself permitted. If this sort of distinction is made use of in defining inheritance, then it is straightforward to license the conclusion corresponding to the endpoints of the path without actually requiring support of the path itself, as in Definition 16. This achieves soundness with respect to ideally skeptical reasoning as well, although it is in a system slightly more expressive than path-based reasoning.

Definition 15 (Implicit Link Abstraction) *Given a network* Γ *and a reasoner* $\mathcal{R}$*:*

- $x \rightsquigarrow y$ *if* $\exists \pi,\ x \overset{\pi}{\rightsquigarrow} y$
- $x \not\rightsquigarrow y$ *if* $\exists \pi,\ x \overset{\pi}{\not\rightsquigarrow} y$

Definition 16 (Ideally Skeptical Permission ($\rightsquigarrow$))

1. Let π be a path.
   (a) If π is a direct link, then π is permitted ($x \overset{\emptyset}{\rightsquigarrow} y$, where $firstnode(\pi) = x$ and $lastnode(\pi) = y$).
   (b) If the degree of π is one then π is a direct link by the definition of degree, hence π is permitted ($x \overset{\emptyset}{\rightsquigarrow} y$, where $firstnode(\pi) = x$ and $lastnode(\pi) = y$).
2. Let π be a compound path of degree n, with $firstnode(\pi) = x$. Assume that all permitted paths with degree less than n are known.
   (a) If π is a positive path then it has the form αz. The path α is positive. The path π is permitted ($x \overset{\alpha}{\rightsquigarrow} z$) iff
      i. α is permitted,
      ii. $firstnode(\alpha) \not\longrightarrow z$ is not a direct link in the net,
      iii. no conflicting path matters (in the fully skeptical sense).
   (b) If π is a negative path (it has the form, α/z), then π is permitted ($x \overset{\alpha}{\not\rightsquigarrow} z$) iff:
      i. α is permitted,
      ii. $firstnode(\alpha) \longrightarrow z$ is not a direct link in the net,
      iii. no conflicting path matters (in the fully skeptical sense).
   (c) If π is a positive path then it has the form αz. The path α is positive. The path π is weakly permitted ($x \rightsquigarrow z$) iff
      i. α is at least weakly permitted ($x \rightsquigarrow lastnode(\alpha)$),
      ii. $firstnode(\alpha) \not\longrightarrow z$ is not a direct link in the net,
      iii. no conflicting path matters (in the extended skeptical sense).
   (d) If π is a negative path (it has the form, α/z), then π is weakly permitted ($x \overset{\alpha}{\not\rightsquigarrow} z$) iff:
      i. α is at least weakly permitted ($x \rightsquigarrow lastnode(\alpha)$),

ii. $firstnode(\alpha) \longrightarrow z$ is not a direct link in the net,

iii. no conflicting path matters (in the extended skeptical sense).

While Definition 16 does capture ideal reasoning, it does so by utilizing abstractions over paths (using the network in Figure 19 as an example, it does not license $a \overset{bef}{\rightsquigarrow} h$, but it does license $a \rightsquigarrow h$), and is therefore not strictly a path-based inheritance reasoner, yet it does this without reasoning about all credulous expansions. However, we have argued that there is reason to feel the paths licensed by the extended skepticism (with both subpath credulity and reasoning with cases) are in fact supportable, even though they do not occur in all credulous expansions (note, for example that it only licenses some of the paths whose endpoints are represented in the implicit link). For the remainder of this paper we will focus on strictly path based accounts. We will not provide an implementation of ideally skeptical reasoning, but do implement full skepticism extended with reasoning by cases (see the appendix).

A reasoner that uses subpath credulity and reasoning by cases will not cascade ambiguity to the same extent that the unextended skepticism will, however it will in many cases reduce to exactly the same thing. Just as in skeptical reasoning with subpath credulity, adding reasoning about cases is compatible with both forms of preemption. However, because it licenses conclusions about paths (which in ideal skepticism we can call weakly permitted) that have ambiguous subpaths, the extended skepticism yields less ambiguity than fully skeptical reasoning.

3.3.3 Summary

This section has presented a series of skeptical reasoners with different degrees of skepticism. The literature makes its sharpest distinction between credulous reasoning in which ambiguities are resolved one way or another in two or more expansions and skeptical reasoning in which there is always just one expansion (and faced with a genuine ambiguity, no determinate conclusion is made). Clearly, though, there is more than one kind of skepticism to work with. To fit with traditional notions of skepticism, it makes sense to consider the possibilities in terms of their relative capacity for classifying a network as ambiguous. H90 reaches a definite conclusion when at all possible. Allowing ambiguities to cascade (requiring subpath credulity) gives the opposite end of the spectrum in which a network is more likely to be considered ambiguous. The two possibilities in between are subpath credulity augmented by reasoning with cases or ideally skeptical reasoning. The preceding section described how ideally skeptical reasoning licenses conclusions that correspond to no path that exists in the intersection of all credulous expansions. The other system introduced here, cascaded ambiguities with case reasoning, approximates ideally skeptical reasoning, but is not identical since it licenses paths as well as the implicit links they justify. A summary of these relationships is depicted in Figure 21. The variation discussed in the next section, downwards reasoning, is orthogonal to skepticism associated with paths and subpaths, and for each combination of the preceding systems yields a system that is more likely to find ambiguity.

3.4 Direction of Reasoning

Recall from Section 1.2.4 that the direction of chaining also affects the conclusions that are sanctioned by a reasoner. That section surveyed a few approaches to direction of chaining, and pointed out that the negative complexity results are attached to a form of double chaining. H90, a polynomial system, utilizes forward single chaining, but both Touretzky (1986) and Boutilier (1989) rely on double chaining. The complexity results of Selman and Levesque (1989, 1993) for downwards reasoning actually apply to double chaining. Actually, downward reasoning need be no more complex than upwards chaining, which fits with the observation also pointed out in Section 1.2.4 that a downwards reasoner will never resolve to a single expansion where an upwards reasoner would find ambiguity (ambiguities are easier to identify than definite conclusions).

Definition 17 (Downwards Permission)

1. Let π be a path.

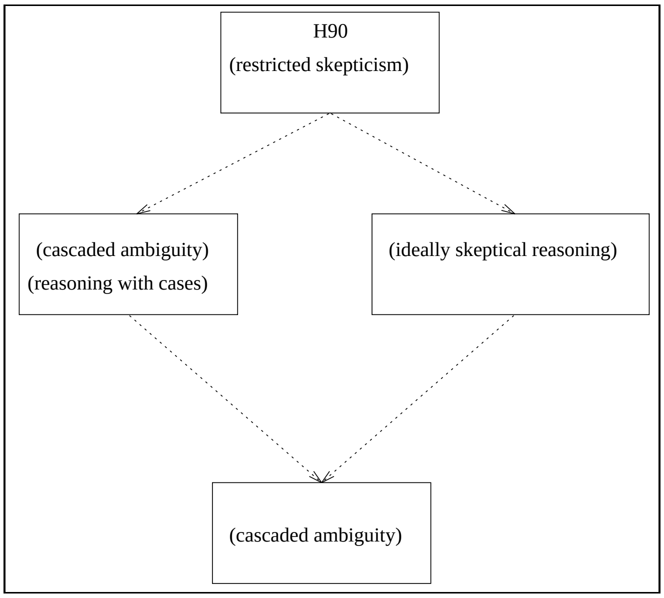


Figure 21: Paths Preempted by Various Systems

   (a) If π is a direct link, then π is permitted.
   (b) If the degree of π is one then π is a direct link by the definition of degree, hence π is permitted.

2. Let π be a compound path of degree n. Assume that all permitted paths with degree less than n are known.
   (a) If π is a path then it has the form $x \longrightarrow \alpha$. The path α may be positive or negative. The path π is permitted iff
      i. α is permitted,
      ii. $x \not\longrightarrow \mathit{lastnode}(\alpha)$ is not a direct conflicting link in the net,
      iii. no conflicting path matters.

Note that this definition assumes the standard definition of path construction: a negative link can occur only as the last link of a path. This makes it easier to collapse the specification of permission of positive and negative paths. Just as in the forward chaining definition of permission, the path being chained onto has to be well-formed (potentially permitted), it's just that in this case it's the back half of the path rather than the front half. Everything else remains the same. This is sufficient to obtain the difference in reasoning about the network depicted in Figure 22: even though the path abd/e is preempted in an upwards reasoner, thus allowing the conclusion $a \overset{bc}{\rightsquigarrow} e$, for a downwards reasoner it is not possible to chain past the ambiguity at b about whether $b \overset{c}{\rightsquigarrow} e$ or $b \overset{d}{\not\rightsquigarrow} e$, and therefore no conclusion is reached about whether there is an implicit link between a and e. The various forms of preemption can also be slotted into downwards reasoning.

3.5 What People Really Use

At this point, definitions have been provided for twenty-one reasoners,[10] although they are not all distinct. The possible combinations are indicated in Table 4; there are three classes of redundancy (link, informational, none) for on-path preemption, and four classes (the above three plus

[10]Actually, forty-two if you include upwards and downwards chaining.

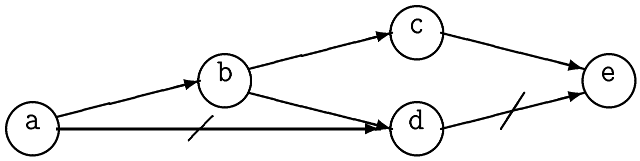


Figure 22: Preemption by $a \not\rightarrow d$ Permits the Conclusion $a \overset{bc}{\leadsto} e$.

path/link) for off-path preemption. Each of these seven possibilities has three possible values for degree of skepticism which yields twenty-one reasoners. There is overlap among the twenty-one: for instance, off-path preemption with only link redundancy is just on-path preemption. The advantage of setting the reasoners out in these terms is that it lays the foundation for inspecting the relationships among them so that essential equivalences and differences can be identified.

		Degree of Skepticism		
Preemption		Restricted	Fully	Extended
On-Path				
Redundancy		✓	✓	✓
path/link	×			
link	✓			
informational	✓			
none	✓			
Off-Path				
Redundancy		✓	✓	✓
path/link	✓			
link	✓			
informational	✓			
none	✓			

Table 4: A Space of Possible Default Inheritance Reasoners

Hewson and Vogel (1994) and Vogel (1995b) tested people for patterns of response to sets of sentences and compared the observed responses with the predictions of various reasoners. It turned out that none of the reasoners did a perfect job of it. Even though people did tend in the majority to classify problems as ambiguous, that is insufficient evidence to opt for the most skeptical possible system (presumably, downwards reasoning without even preemption) since there may be other factors, like problem complexity, that interact with people's classifications. The concentrations of responses to problems after the indeterminate classifications are set aside yield evidence for less skeptical systems, or at least a stronger foothold from which to explore the factors that affect their classification. It is equally interesting to define reasoners which yield a closer approximation to observed behavior.

In general terms, people's behavior was in accord with transitivity for simple problems. However, a definition of complexity was not provided, although it seems that whatever definition is provided interacts with at least: the number of links in network, and the number of paths through a network. These two factors are not inter-reducible, though they tended to be related in the problems tested. Essentially, with greater complexity there was an increased trend to classify the network as indeterminate, which, interestingly, implies that people reason more closely to the way predicted by full skepticism than either extended skepticism or ideal skepticism. People rarely behaved as would have been expected if reasoning were guided by preemption. Rather than 'more-direct' links obtaining priority for one path over another, they seemed to provide conflicting information which rendered the whole problem indeterminate. Recall that this is the net effect of basing the definition of preemption on something like informational redundancy, although that definition does not capture the issues at stake (since it does still license disregarding some paths as redundant). A definition which comes closer in effect and classification would be one that assumes no link redundant, and rules out preemption as well in most cases. Shortest path reasoning (with which most theories of preemption agree on *certain* networks) was not supported at all by observation, although there was a tendency to accept direct preempting links as decisive; this supports on-path preemption without redundant links. However, that support as well also falls away with more complex problems. There did seem to be some support as well for conclusions based on counting the number of paths of each polarity, more support with respect to complex problems than was available for most of the other strategies, in fact. However, that strategy itself only holds if a rather fundamental assumption of the inheritance literature is set aside, in particular, the assumption that paths can contain negative links only as their last links. Observations suggest at least three possible alternatives, the first two seeming to fit better than the last.

Definition 18 (negative path threshold)
Let intensity$(\pi) = \begin{cases} 1 \textit{ iff it contains at least one negative link} \\ 0 \textit{ otherwise} \end{cases}$

Definition 19 (negative path intensity)
Let intensity(π) = *the number of negative links in* π

Definition 20 (negative path algebra)
Let intensity$(\pi) = \begin{cases} 0 \textit{ iff it contains an even number of negative links} \\ 1 \textit{ otherwise} \end{cases}$

Definition 21 (psychologically plausible inheritance (path-counting))

1. Let π be a path.
   (a) If π is a direct link, then π is permitted.
   (b) If the degree of π is one then π is a direct link by the definition of degree, hence π is permitted.
2. Let π be a compound path of degree n. Assume that all permitted paths with degree less than n are known.
   (a) If π is a positive path, then π is permitted iff $|\,\textit{firstnode}(\pi) \longrightarrow \omega \longrightarrow \textit{lastnode}(\pi)| \geq |\pi'|$ for which $\textit{firstnode}(\pi) = \textit{firstnode}(\pi')$ and $\textit{lastnode}(\pi) = \textit{lastnode}(\pi')$ where for each π' $\textit{intensity}(\pi') \geq 1$
   (b) If π is a negative path, then π is permitted iff $|\,\textit{firstnode}(\pi) \not\longrightarrow \omega \not\longrightarrow \textit{lastnode}(\pi)|$ (where $\textit{intensity}(\pi') \geq 1) \geq |\pi'|$ for which $\textit{firstnode}(\pi) = \textit{firstnode}(\pi')$ and $\textit{lastnode}(\pi) = \textit{lastnode}(\pi')$.

Vogel (1995a) and (Vogel, 1996b) examine consistent within-subject patterns of response (the reasonsers defined above are based on between-subject systematicity on individual problems, not on subject-internal consistency on sets of related networks) and identify additional reasoning strategies. These have yet to be related in formal detail to the systems presented in this section and the full family of reasoners discussed in this paper as a whole.

4 Discussion

Due to the proliferation of inheritance reasoners, it is important to determine precisely how different classes of reasoners are related. This paper outlines a declarative specification of inheritance reasoning which makes it easier to see the relationships among various systems along axes distinct from the features that define them (for instance, downwards reasoning without preemption being the most skeptical sort of system possible). The relationships among various reasoners have been illustrated by defining the various reasoners through modifications to a basic set of definitions. To our knowledge, the space of reasoners outlined here (which is larger than that considered by the "Clash of Intuitions" paper (Touretzky et al., 1987)), has never been presented within such a uniform framework. One advantage of this presentation is that it sets the stage for future work to provide rigorous analysis of exactly what equivalences exist among these systems. A second more immediate advantage is that it leads to an easier implementation of the space of reasoners by parameterizing code for the various substitutions required in the definitions.

This section concludes with some observations about the complexity of downwards reasoning as proposed here. It is useful to first give the basic algorithm:

Input: a direct acyclic graph of positive and negative default inheritance links

Query: For all nodes x, y and through arbitrary paths π, π'..., does the network support $x \overset{\pi}{\rightsquigarrow} y$ or $x \overset{\pi}{\not\rightsquigarrow} y$.

Optimized Algorithm for Downwards Permission:

1. Let π be a path.
   - If π is a direct link, then π is permitted ($\mathcal{O}(c)$).
   - Record $x \overset{\emptyset}{\rightsquigarrow} y$ or $x \overset{\emptyset}{\not\rightsquigarrow} y$ as appropriate ($\mathcal{O}(c)$).
2. Let π be a compound path of degree n. Assume that all permitted paths with degree less than n are known.
   (a) If π is a path then it has the form $x \longrightarrow \alpha$. The path α may be positive or negative. The path π is permitted (so, record $x \overset{y\omega}{\rightsquigarrow} z$ or $x \overset{y\omega}{\not\rightsquigarrow} z$ as appropriate) iff
      i. α is recorded as permitted (*firstnode*$(\alpha) = y$, *lastnode*$(\alpha) = z$, $y \overset{\omega}{\rightsquigarrow} z$; $\mathcal{O}(n^2)$),
      ii. $x \not\longrightarrow$ *lastnode*(α) is not a direct conflicting link in the net ($\mathcal{O}(c)$),
      iii. no conflicting path matters.

From the above, it is clear that the most expensive fixed step is the test on whether the subpath is itself permitted and therefore extendible into a longer path. The reason this costs $\mathcal{O}(n^2)$ ($n =$ number of links in the network), is that the length of the list of recorded implicit links is bounded by n^2. Similarly, the number of times step two gets invoked is bounded by: $\frac{(n-1)^2+(n-1)}{2}$, which is also $\mathcal{O}(n^2)$. Thus, over the whole algorithm, the fixed steps cost $\mathcal{O}(n^4)$. The cost of determining whether conflicting paths matter is not fixed, and depends on which version is used. Consider the following:

Input: a direct acyclic graph of positive and negative default inheritance links, a list of permitted implicit links, a putative implicit link in question: either $x \overset{\pi}{\rightsquigarrow} y$ or $x \overset{\pi}{\not\rightsquigarrow} y$.

Query: are there any paths $x\omega/y$ (which conflict with $x\pi y$, if that is the input) or $x\omega y$ (which conflicts with $x\pi/y$, if that is the input) which are not preempted?

Conflicts that Matter: Restricted Skepticism No path conflicting with π matters iff

1. if π has the form, αz, and all negative paths $\pi' = \alpha'/z$, where α' is a permitted positive path ($\mathcal{O}(n^2)$) and $firstnode(\alpha) = firstnode(\alpha')$, are preempted.

2. if π has the form, α/z, and all positive paths $\pi' = \alpha' z$, where α' is a permitted positive path ($\mathcal{O}(n^2)$) and $firstnode(\alpha) = firstnode(\alpha')$, are preempted.

Assume that the version of preemption in place is on-path.

Input: a direct acyclic graph of positive and negative default inheritance links, a list of permitted implicit links, a putative implicit link in question: either $x \overset{\pi}{\rightsquigarrow} y$ or $x \overset{\pi}{\not\rightsquigarrow} y$.

Query: is $x\pi y$ or $x\pi/y$ (as determined by the input) preempted?

On-Path Preemption Let π and π' be positive paths, and let p and y be nodes. A positive path πy is *preempted* by a link $p \not\longrightarrow y$ or a negative path π/y is *preempted* by a link $p \longrightarrow y$ if there exists a permitted path π' such that

1. $\pi \subseteq \pi'$ ($\mathcal{O}(n)$)
2. $firstnode(\pi) = firstnode(\pi')$ ($\mathcal{O}(c)$),
3. $lastnode(\pi) = lastnode(\pi')$ ($\mathcal{O}(c)$),
4. p occurs in π' ($\mathcal{O}(n)$), and
5. $p \neq lastnode(\pi')$ ($\mathcal{O}(c)$).

Thus, on-path preemption contributes in the worst case $\mathcal{O}(n)$ to the complexity of determining whether any conflicting paths matter (using restricted skepticism), which does not overshadow the cost of determining subpath permission. Since determining whether conflicting paths matter iterates over known permitted paths, and that is bounded by $\mathcal{O}(n^2)$, the overall cost of determining whether conflicting paths matter is $\mathcal{O}(n^4)$. This means that for downwards restricted skeptical reasoning with on-path preemption, the time complexity is $\mathcal{O}(n^6)$. Note that for this worst case analysis it makes no difference to substitute off-path preemption. However, utilizing ambiguity cascading skepticism, there is a slight improvement to $\mathcal{O}(n^5)$.

Input: a direct acyclic graph of positive and negative default inheritance links, a list of permitted implicit links, a putative implicit link in question: either $x \overset{\pi}{\rightsquigarrow} y$ or $x \overset{\pi}{\not\rightsquigarrow} y$.

Query: are there any paths $x\omega/y$ (which conflict with $x\pi y$, if that is the input) or $x\omega y$ (which conflicts with $x\pi/y$, if that is the input) which are not preempted?

Conflicts that Matter: Subpath Credulity No path conflicting with π matters iff

1. if π has the form, αz, and all negative paths $\pi' = \alpha'/z$, where α' is a positive path ($\mathcal{O}(n)$) and $firstnode(\alpha) = firstnode(\alpha')$, are preempted.

2. if π has the form, α/z, and all positive paths $\pi' = \alpha' z$, where α' is a positive path ($\mathcal{O}(n)$) and $firstnode(\alpha) = firstnode(\alpha')$, are preempted.

Tighter bounds are almost certainly possible, but the present results are sufficient to satisfy the intuition that downwards reasoning need not be intractable if chaining is defined symmetrically to the forward chaining case. In particular, if H90 inferences, a restricted skeptical, off-path preempting, upwards chaining reasoner are polynomial computable, then so should a downwards version. The importance of this is to be rated in light of the intractability of downwards reasoning reported by Selman and Levesque (1989, 1993): however it should be emphasized that a polynomial algorithm for downwards inheritance as defined here does not contradict their results; rather it emphasizes the asymmetry they assume between the definitions of upwards and downwards reasoning. The downward reasoning they consider is actually double chaining. It also turns out that downwards fully skeptical reasoners are more efficient than the restricted skeptical version. This makes it worth examining whether the same is true for the upwards case.

We do not analyze that possibility in this paper; simply, we point out again that the declarative framework in which the reasoners are defined here makes it quite possible for such modular analyses to be performed. An advantage of this work is that it makes it easier to provide a uniform semantics to inheritance reasoning, parameterized to the proof-theoretic variations considered here in this paper. Vogel (1996a) provides a semantics for H90, and Vogel (1995b) demonstrates how this can be parameterized to provide a semantics to corresponding inheritance reasoners. This makes even clearer the advantage of specifying the definitions of the various inheritance reasoners within the same framework. Until now, there have been semantics for individual inheritance systems (Touretzky, 1986; Boutilier, 1989; Delgrande, 1990; Veltman, 1994), but never before has a single semantic framework able to be attuned to the variety of inheritance reasoners been offered.

A Implementation

Another advantage of spelling out the definitions of various inheritance reasoners in common terms is that it facilitates implementation as well as just abstract interclassification. The first order specifications translate fairly directly into logic programs, offering a further advantage in making the relationship between the definitions and their implementation more clear. This section details a Prolog implementation of the various systems from the preceding section. The basic system was first presented by Vogel et al. (1993).

A.1 H90: upwards chaining, off-path preemption, restricted skepticism

A.1.1 Permission

A restatement of Definition 2 in Prolog is shown in Figure 23. This restatement is slightly different in structure from Definition 2; the Prolog definition is not stated explicitly in terms of increasing degree. The Prolog definitions consider only links that are relevant to the query which participate in some path sharing an endpoint with one of the queried nodes. Degree is just the length of the longest generalized path between those nodes; in the present system that concept can be accessed directly by referring to those paths directly as paths rather than indirectly via their lengths.

The base case (shown below) is the first `permitted` clause which states that all direct links in a network are permitted. The empty list in the middle position of the term for the basis indicates that no intermediate nodes lie on the path. An additional base clause to permit paths of degree one is unnecessary since the set of paths whose degree is one is a subset of the set of paths that are direct links. The second clause of the Prolog program below defines permission in the general case where π is the path from `From` to `To`. The relation `complement` is used in defining this clause; thus, the same clause stipulates the permission of both positive and negative paths. The path from `From` through `SubPath` to `Last` maps to α in the formal definition of permission, and `To` maps to z (uppercase tokens are assumed to be variables). The relations, *link* and *lastnode*, verify that *lastnode*$(a) \mathrel{-\!\!\ominus\!\!\rightarrow} z$ is a link contained in the network. Note that if a directly conflicting link is contained in the network, then due to the presence of the restriction `not(link(From,NotTo))`, the path from `From` to `To` is not allowed (according to condition (ii) from Definition 2). The reference to `permitted` specifies that the subpath α must itself be permitted. Finally, the relation `no_other` holds when the path specified as input through its arguments is actually a chain of links through the network which is not itself preempted. Thus, the specification `not(unpreempted)` stipulates that no conflicting, unpreempted paths exist. This is equivalent to the specification in the formal definition that is expressed: for all paths that conflict with $a \mathrel{-\!\!\ominus\!\!\rightarrow} z$, there exists some path which

```
permitted(From,[],To) :-                        Basis
        link(From,To).

permitted(From, Path, To) :-                    π is a compound path...
        nonempty(Path),                         ... from From through Path to To
        path(From, Path, To),                   for α
        lastnode(SubPath, Last, Path),          for π'
        complement(To, NotTo),                  (ii), no direct link conflicts
        not(link(From,NotTo)),                  (i), α is permitted
        permitted(From, SubPath, Last),         (iii), just give endpoints of π'
        no_other(From, NotTo).

no_other(From, To) :-                           All π' (compound conflicts) are preempted
        not(unpreempted(From, _, To)).

unpreempted(From,[],To) :-                      A link isn't preempted
        link(From,To).

unpreempted(From,Path,To) :-                    Is a compound path π' not preempted?
        nonempty(Path),                         It must be compound...
        path(From, Path, To),                   It must be a path (π' = α' + z)
        lastnode(SubPath, Last, Path),          Such that α' exists,
        positive_path(From, SubPath, Last),     where α' is a positive path
        allowed(From, SubPath, Last),           that is allowed (ie. permitted).
        not(preempted(From,Path,To,By)).        The well-formed π' is unpreempted if it
                                                is not preempted.
allowed(From, Thru, To) :-
        permitted(From, Thru, To).
```

Figure 23: The Definition of Permission in Prolog.

```
link(a,b).          link(a,c).          link(b,d).
link(c,d).          link(c,not(e)).     link(d,e).
```

Figure 24: A Prolog Translation of Figure 10.

preempts each conflicting path. The clauses which make up the definitions of `unpreempted` and `permitted` implement Definition 2 even though a different ordering is stated on those constraints. The order of the restrictions stated in the Prolog definition is guided by efficiency considerations in limiting the search space.

Figure 24 depicts a Prolog translation of the network given in Figure 10. Links are encoded as two place relations. The polarity of a link is indicated in the second argument. A positive link $x \longrightarrow y$ is encoded as `link(x,y)`, and the negative link $x \not\longrightarrow y$ is encoded as `link(x,not(y))`. Note that the predicate can be invoked with all variables to list the entire set of permitted paths, not just those pertaining to particular nodes. This is an advantage of specification in the logic programming paradigm for investigating the implications of the network. A more important advantage of the logic specification is its immediate mutability into other instances in the space of possible inheritance reasoners.

A.1.2 Off-Path Preemption

Figure 25 depicts a restatement the definition of off-path preemption as a Prolog relation between a path (specified in the arguments `From`, `Path`, `To`) and a preempting link from `Preemptor` to `To`. While `From`, `Path`, and `To` are all variables, it is assumed that `From` and `To` vary over nodes, and that `Path` varies over lists of nodes with positive, not negative, links implicitly connecting

```
preempted(From,Path,To,Preemptor) :-
        path(From, Path, To),                   Define π
        lastnode(SubPath, Last, Path),
        complement(To, NotTo),
        link(Preemptor, NotTo),                 Preempting link
        relation(OtherPath, SubPath),           On or Off path preemption
        positive_path(From, OtherPath, Last),   Define π'
        permitted(From,OtherPath,Last),         π' is permitted
        member(Preemptor,[From|OtherPath]).     P is on π' (but not Last)
```

Figure 25: The Definition of Off-Path Preemption in Prolog.

```
relation(X, Y) :- sublist(Y,X).    OtherPath and Path are the same or
                                   Path is redundant to OtherPath
```

Figure 26: The Definition of On-Path Preemption in Prolog.

those nodes in the order of occurrence in a given list. The path from `From` to `Last` through `SubPath` corresponds to π in Definition 3, since the `lastnode` of `Path` is stipulated to be `Last`. The `preempted` relation can hold only if the chain of links described by the path $\mathit{From} \longrightarrow \mathit{Through} \not\rightarrow \mathit{To}$ is actually a chain of links in the network. The relations `complement` and `link` determine whether there is a conflicting path terminating at the node `To`. The path from `From` to `Last` through `OtherPath` corresponds to π' in Definition 3—π and π' have the same first node `First` and the same last node `Last`. Off-path preemption obtains if `relation` does not impose any restrictions on its arguments. The relation `lastnode` verifies that $\mathit{lastnode}(\pi) = \mathit{lastnode}(\pi')$, and the invocation of `permitted` (defined above) verifies that π' is actually permitted. Finally, the call to `member` is used to determine if some node p (`Preemptor`) other than $\mathit{LastNode}(\pi')$ (`Last`) participates in the conflicting link.

A.2 Other Systems

A.2.1 On-Path Preemption

Above, off-path preemption was implemented through the `relation` predicate which articulated the relationship, apart from coincidence at the endpoints, of the paths under consideration (a path and its potential preempting path). Recall that the difference between off-path and on-path preemption is that in on-path preemption one path must be a subpath of the other—the preempted path cannot contain nodes that the preempting path does not contain, although the preempted path can contain fewer nodes. This is implemented just by imposing stronger constraints on the `relation` predicate, as shown in Figure 26.

A.2.2 No Redundancy

Obtaining on-path preemption under the assumption that no link is redundant requires the trivial re-definition of the `relation` predicate such that it requires both paths to be identical forcing on path preemption (`relation(X,X)`). We do not supply an implementation for Boutilier's (1989) system using informational redundancy rather than the usual topologically defined notion in the check on whether there are any conflicting paths that matter, nor do we supply an implementation for inheritance using informational redundancy behind its notion of permission.

A.2.3 Skepticism

Fully Skeptical Reasoning The difference between restricted skeptical reasoning and fully skeptical reasoning is in whether subpaths of permitted paths are required to be unambiguous. Restricted skeptical reasoners do not require subpaths to be permitted in all cases, as discussed above. The difference is encoded in the specification of the relation `allowed`, which implements part of condition (iii) from the main definition of permission. It is actually invoked in considering paths that conflict with the path under construction (`unpreempted`). Thus, the usual case requires that conflicting paths be permitted up to their penultimate node.

```
allowed(From, Thru, To) :- permitted(From, Thru, To).
```

While the fully skeptical reasoner just requires that the chain of links up to the second to last node not be preempted.

```
allowed(From, Thru, To) :- unpreempted(From, Thru, To).
```

This completes the modification required to obtain a fully skeptical reasoner. The non-preemption of α'/z must still be verified, as in the definition, because otherwise we disregard more specific information provided by a preempting link.

Extended Skeptical Reasoning Extended skeptical reasoning, as introduced above, employs subpath credulity and adds reasoning with cases in order to generate conclusions allowed in the intersection of credulous expansions, without having to compute that intersection directly. That required introducing a second clause into the stipulation of what conflicting paths matter. In the Prolog implementation, that multiplies out into a second clause for permission, different solely in the last line which has a new way of classifying conflicting paths irrelevant which requires more information about the path under question. Of course, the parameterization would be considerably cleaner if this was handled by expanding the arity of `no_other`, but that would just be a trivial manipulation of the present implementation which does do the job.

```
permitted(From, Path, To) :-
        nonempty(Path),                          Distinguish from base case
        path(From, Path, To),                    Define α
        lastnode(SubPath, Last, Path),
        complement(To, NotTo),
        not(link(From,NotTo)),                   No conflicting direct link exists
        not(preempted(From, SubPath, Last,P)),   α is not preempted
        conflicts_irrelevant(From,SubPath,Last,To).
```

The new predicate, `conflicts_irrelevant`, as stipulated by the definition establishes whether a conflicting chain of links exists and requires each intermediate node to support the inference to the endpoint `To` of the path under extension (`From` $\longrightarrow$ `Subpath` $\longrightarrow$ `Last` which is tested for extendibility to `To`), even if there would be conflicts on subpaths.

```
conflicts_irrelevant(From,Subpath,Last,To) :-
        conflict(From,Subpath,Last,BadPath,not(Bad)),
        all_permitted([Bad|BadPath],To).

all_permitted([],_).
all_permitted([Node|Path],To) :-
        permitted(Node,_,To),
        all_permitted(Path,To).

conflict(From,Subpath,Last,BadPath,Bad) :-
        path(From,BadPath,Bad),
        complement(Bad,Good),
        member(Good,[Last|Subpath]).
```

A.2.4 Downwards Reasoning

An initial proposal for turning the Prolog definitions into a specification of downwards reasoning might involve utilizing the reversibility of logical relations with a query like `permitted(X,Y,n)`—for some node `n`: what paths through the nodes as ordered in `Y` and beginning at `X` lead to

n. However, this is not a correct intuition since the definition of the relation is for an upwards reasoner—with all of the variables instantiated the reasoner will attempt to prove the goal stated in the query as an upwards reasoner would. But, upwards and downwards reasoners behave differently (Touretzky et al., 1987). The alterations to the code for the basic algorithm are much the same as the changes to the definition of permission given earlier. First, it is necessary to specify path construction as in:

```
path(From,[],To) :- link(From,To).
path(From,[Mid|Through],To) :-
        link(Last,To),
        path(From,Front,Last),
        lastnode(Front,Last,[Mid|Through]).
```

And then, just supply an alternative main-case definition for permission. In fact, this one requires a bit less computation since it does not require invocation of the lastnode relation in order to find the last element of the path being extended. Permission of the extended path is tested against the back end of the path rather than the front end.

```
permitted(From, [First|SubPath], To) :-
        path(From,[First|SubPath], To),        Define α
        %lastnode(SubPath, Last, Path),
        complement(To, NotTo),
        not(link(From,NotTo)),                 No conflicting direct links
        permitted(First, SubPath, To),         α is permitted
        no_other(From, NotTo).
```

The versions of no_other and its (potentially, depending on what no_other invokes itself) nested call to allowed) can be substituted as with the forward chaining system.

References

Al-Asady, R. & Narayanan, A. (1993). More Notes on 'A Clash of Intuitions'. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pp. 682–7. Chambery, France.

Boutilier, C. (1989). A Semantical Approach to Stable Inheritance Reasoning. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pp. 1134–39. Detroit, Michigan.

Cripps, A. (1987). Aspects of the Inheritance Problem. MSc Thesis, Department of Computer Science, University of Victoria.

Delgrande, J. (1988). An Approach to Default Reasoning Based on a First-Order Conditional Logic: Revised Report. *Artificial Intelligence*, *36*(1), 63–90.

Delgrande, J. (1990). A Semantics for a Class of Inheritance Networks. In *Proceedings of the 8th Biennial Conference of the Canadian Society for Computational Studies of Intelligence*. Ottawa, Ontario.

Geffner, H. & Verma, T. (1989). Inheritance = Chaining + Defeat. In *CSD-890039*. Computer Science Department, University of California, Los Angeles.

Hewson, C. & Vogel, C. (1994). Psychological Evidence for Assumptions of Path-Based Inheritance Reasoning. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, pp. 409–14. Atlanta, Georgia.

Horty, J., Thomason, R., & Touretzky, D. (1990). A Skeptical Theory of Inheritance in Nonmonotonic Semantic Networks. *Artificial Intelligence*, *42*(2-3), 311–48.

Kautz, H. A. & Selman, B. (1991). Hard Problems for Simple Default Logics. *Artificial Intelligence*, *49*, 243–79.

Maugham, S. (1919). *The Moon and Sixpence*. London: Pan Books Ltd. Pan Edition, 1974.

McDermott, D. & Doyle, J. (1980). Non-Monotonic Logic I. *Artificial Intelligence*, *13*(1-2), 41–72.

Niemelä, I. & Rintanen, J. (1994). On the Impact of Stratification on the Complexity of Nonmonotonic Reasoning. *Journal of Applied Non-Classical Logics*, *4*(2), 141–79.

Reiter, R. & Criscuolo, G. (1983). Some Representational Issues in Default Reasoning. *Computers and Mathematics with Applications*, *9*(1), 15–27.

Sandewall, E. (1986). Nonmonotonic Inference Rules for Multiple Inheritance with Exceptions. *Proceedings of IEEE*, *74*(10), 1345–53.

Selman, B. & Levesque, H. (1989). The Tractability of Path-Based Inheritance. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pp. 102–9. Detroit, Michigan.

Selman, B. & Levesque, H. (1993). The Complexity of Path-Based Defeasible Inheritance. *Artificial Intelligence*, *62*, 303–39.

Stein, L. A. (1989). Skeptical Inheritance: Computing the Intersection of Credulous Extensions. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pp. 1153–60. Detroit, Michigan.

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

Touretzky, D., Horty, J., & Thomason, R. (1987). A Clash of Intuitions: The Current State of Non-Monotonic Inheritance Systems. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, pp. 476–82. Milan, Italy.

Veltman, F. (1994). Defaults in Update Semantics. Unpublished ms.; Institute for Language, Logic and Information, University of Amsterdam.

Vogel, C., Popowich, F., & Cercone, N. (1993). Logic-Based Inheritance Reasoning. In *Proceedings of the 9th Bi-Annual Conference of the Society for the Study of Artificial Intelligence and the Simulation of Behavior*, pp. 179–88. AISB'93, Birmingham, England. March 29–April 2, 1993. Appears in, A. Sloman, et al. (eds.), *Prospects for Artificial Intelligence*. Amsterdam: IOS Press. 1993.

Vogel, C. M. (1995a). Defining Psychologically Plausible Default Inheritance Reasoners. Unpublished manuscript, Institute for Computational Linguistics, University of Stuttgart.

Vogel, C. M. (1995b). *Inheritance Reasoning: Psychological Plausibility, Proof Theory and Semantics.* Ph.D. thesis, Centre for Cognitive Science, University of Edinburgh.

Vogel, C. M. (1996a). A Generalizable Semantics for Inheritance Reasoning. To appear in the CSLI Lecture Notes Volume, *The 1st Tbilisi Symposium on Logic, Language and Computation*.

Vogel, C. M. (1996b). Human Reasoning with Negative Defaults. To be presented at the *International Conference on Formal and Applied Practical Reasoning*, proceedings to be published by Springer-Verlag.