Chapter 9
An Analysis of Defeasible Inheritance Systems

The material in this chapter is taken from [GS08e].

9.1 Introduction

9.1.1 Terminology

"Inheritance" will stand here for "nonmonotonic or defeasible inheritance". We will use indiscriminately "inheritance system", "inheritance diagram", "inheritance network", "inheritance net".

In this introduction, we first give the connection to reactive diagrams, then give the motivation, then describe in very brief terms some problems of inheritance diagrams, and mention the basic ideas of our analysis.

9.1.2 Inheritance and Reactive Diagrams

Inheritance systems or diagrams have an intuitive appeal. They seem close to human reasoning, natural, and are also implemented (see [Mor98]). Yet, they are a more procedural approach to nonmonotonic reasoning, and, to the authors' knowledge, a conceptual analysis, leading to a formal semantics, as well as a comparison to more logic-based formalisms like the systems P and R of preferential systems are lacking. We attempt to reduce the gap between the more procedural and the more analytical approaches in this particular case. This will also give indications how to modify the systems P and R to approach them more to actual human reasoning. Moreover, we establish a link to multi-valued logics and the logics of information sources (see, e.g. [ABK07] and forthcoming work of the same authors, and also [BGH95]).

An inheritance net is a directed graph with two types of connections between nodes, $x \rightarrow y$ and $x \not\rightarrow y$. Diagram 9.1.1 (p. 252) is such an example. The meaning of $x \rightarrow y$ is that x is also a y and the meaning of $x \not\rightarrow y$ is that x is not a y.

D.M. Gabbay, K. Schlechta, *Logical Tools for Handling Change in Agent-Based Systems*, Cognitive Technologies, DOI 10.1007/978-3-642-04407-6_9,

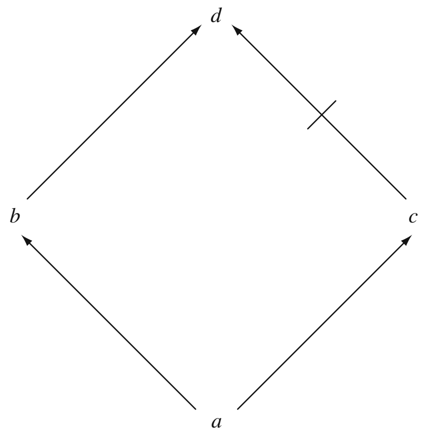


Diagram 9.1.1 The Nixon Diamond

We do not allow the combinations $x \not\to y \not\to z$ or $x \not\to y \to z$, but we do allow $x \to y \to z$ and $x \to y \not\to z$.

Given a complex diagram such as Diagram 9.1.2 (p. 252) and two points say z and y, the question we ask is to determine from the diagram whether the diagram says that

(1) z is y,
(2) z is not y,
(3) nothing to say.

Since in Diagram 9.2.2 (p. 259) there are paths to y from z either through x or through v, we need to have an algorithm to decide. Let $\mathcal{A}$ be such an algorithm.

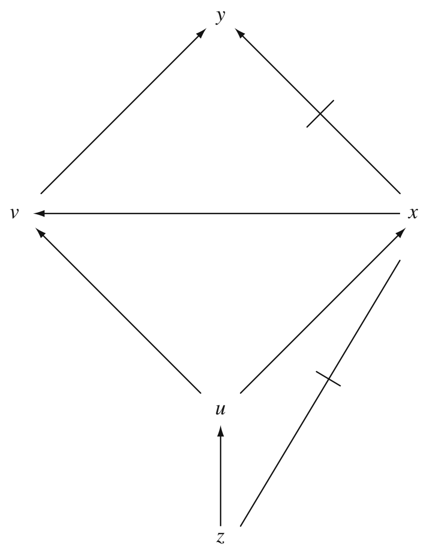


Diagram 9.1.2 The problem of downward chaining

We need $\mathcal{A}$ to decide

(1) Are there valid paths from z to y?
(2) Of the opposing paths (one which supports "z is y" and one which supports "z is not y"), which one wins (usually winning makes use of being more specific, but there are other possible options)?

So, for example, in Diagram 9.1.2 (p. 252), the connection $x \to v$ makes paths through x more specific than paths through v. The question is whether we have a valid path from z to x.

In the literature, as well as in this chapter, there are algorithms for deciding the valid paths and the relative specificity of paths. These are complex inductive algorithms, which may need the help of a computer for the case of the more complex diagrams.

It seems that for inheritance networks we cannot adopt a simple minded approach and just try to "walk" on the graph from z to y, and depending on what happens during this "walk" decide whether z is y or not. To explain what we mean, suppose we give the network a different meaning, that of fluid flow. $x \to y$ means there is an open pipe from x to y and $x \not\to y$ means there is a blocked pipe from x to y.

To the question "can fluid flow from z to y in Diagram 9.1.2" (p. 252), there is a simple answer:

> Fluid can flow iff there is a path comprising of $\to$ only (without any $\not\to$ among them).

Similarly, we can ask in the inheritance network something like (*) below:

> (*) z is (resp. is not) y according to diagram D, iff there is a path π from z to y in D such that some noninductive condition $\psi(\pi)$ holds for the path π.

Can we offer the reader such a ψ?

If we do want to help the user to "walk" the graph and get an answer, we can proceed as one of the following options:

> Option 1. Add additional annotations to paths to obtain D^* from D, so that a predicate ψ can be defined on D^* using these annotations. Of course, these annotations will be computed using the inductive algorithm in $\mathcal{A}$, i.e. we modify $\mathcal{A}$ to $\mathcal{A}^*$ which also executes the annotations.
>
> Option 2. Find a transformation τ on diagrams D to transform D to $D' = \tau(D)$, such that a predicate ψ can be found for D'. So we work on D' instead of on D.

We require a compatibility condition on options 1 and 2:

(C1) If we apply $\mathcal{A}^*$ to D^* we get D^* again.
(C2) $\tau(\tau(D)) = \tau(D)$.

We now present the tools we use for our annotations and transformation. These are the reactive double arrows.

Consider the following Diagram 9.1.3 (p. 254):

We want to walk from a to e. If we go to c, a double arrow from the arc $a \to c$ blocks the way from d to e. So the only way to go to e is through b. If we start at a' there is no such block. It is the travelling through the arc (a, c) that triggers the double arrow $(a, c) \twoheadrightarrow (d, e)$.

We want to use $\twoheadrightarrow$ in $\mathcal{A}^*$ and in τ. So in Diagram 9.1.2 (p. 259) the path $u \to x \not\to y$ is winning over the path $u \to v \to y$, because of the specificity arrow $x \to v$. However, if we start at z then the path $z \to u \to v \to y$ is valid because of $z \not\to x$. We can thus add the following double arrows to the diagram to get Diagram 9.1.4 (p. 255).

If we start from u and go to $u \to v$, then $v \to y$ is cancelled. Similarly, $u \to x \to v$ cancelled $v \to y$. So the only path is $u \to x \not\to y$.
If we start from z, then $u \to x$ is cancelled and so is the cancellation $(u, v) \twoheadrightarrow (v, y)$. Hence the path $z \to u \to v \to y$ is open.
We are not saying that τ(Diagram 9.1.3)= Diagram 9.1.4, but something effectively similar will be done by τ.

We emphasize that the construction depends on the point of departure. Consider Diagram 9.1.2 (p. 259). Starting at u, we will have to block the path uvy. Starting

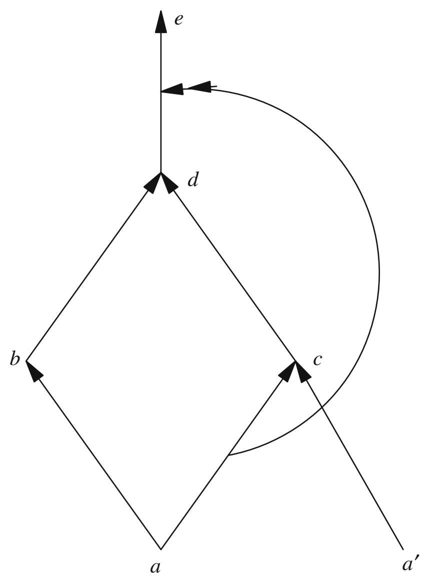


Diagram 9.1.3 Reactive graph

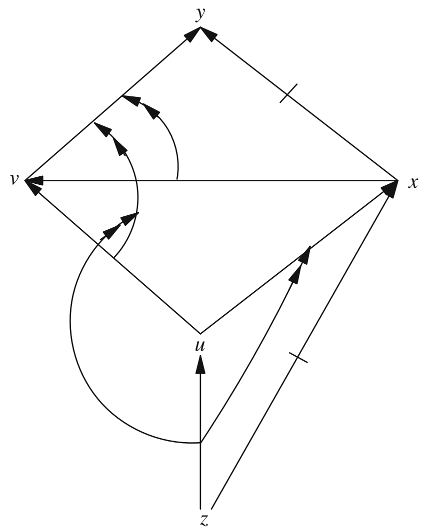


Diagram 9.1.4 Walking through the diagram

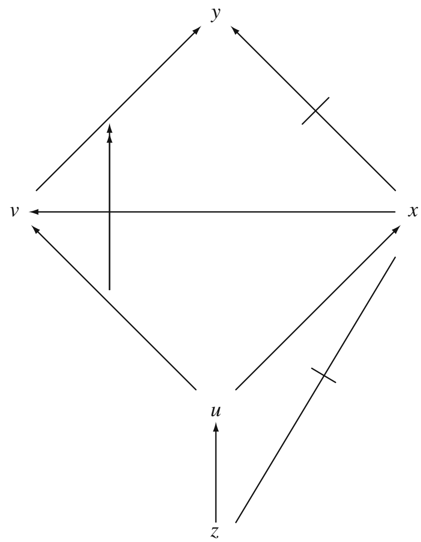


Diagram 9.1.5 The problem of downward chaining – reactive

at z, the path $zuvy$ has to be free. See Diagram 9.1.5 (p. 255). So we cannot just add a double arrow from $u \to v$ to $v \to y$, blocking $v \to y$, and leave it there when we start from z. We will have to erase it when we change the origin. At the same time, this shows an advantage over just erasing the arrow $v \to y$:

When we change the starting point, we can erase simply all double arrows, and do not have to remember the original diagram.

How do we construct the double arrows given some origin x?

First, if all possible paths are also valid, there is nothing to do. (At the same time, this shows that applying the procedure twice will not result in anything new.)

Second, remember that we have an upward chaining formalism. So if a potential path fails to be valid, it will do so at the end.

Third, suppose that we have two valid paths $\sigma : x \to y$ and $\tau : x \to y$.

If they are negative (and they are either both negative or positive, of course), then they cannot be continued. So if there is an arrow $y \to z$ or $y \not\to z$, we will block it by a double arrow from the first arrow of σ and from the first arrow of τ to $y \to z$ ($y \not\to z$, respectively).

If they are positive, and there is an arrow $y \to z$ or $y \not\to z$, both $\sigma y \to z$ and $\tau y \to z$ are potential paths (the case $y \not\to z$ is analogue). One is valid iff the other one is, as σ and τ have the same endpoint; so preclusion, if present, acts on both in the same way. If they are not valid, we block $y \to z$ by a double arrow from the first arrow of σ and from the first arrow of τ to $y \to z$. Of course, if there is only one such σ, we do the same, there is just to consider to see the justification.

We summarize: Our algorithm switches impossible continuations off by making them invisible; there is just no more arrow to concatenate. As validity in inheritance networks is not forward looking – validity of $\sigma : x \to y$ does not depend on what is beyond y – validity in the old and in the new network starting at x are the same. As we left only valid paths, applying the algorithm twice will not give anything new.

We illustrate this by considering Diagram 9.1.6 (p. 256). First, we add double arrows for starting point c, see Diagram 9.1.7 (p. 257), and then for starting point x, see Diagram 9.1.8 (p. 257).

For more information on reactive diagrams, see also [Gab08c], and an earlier version [Gab04].

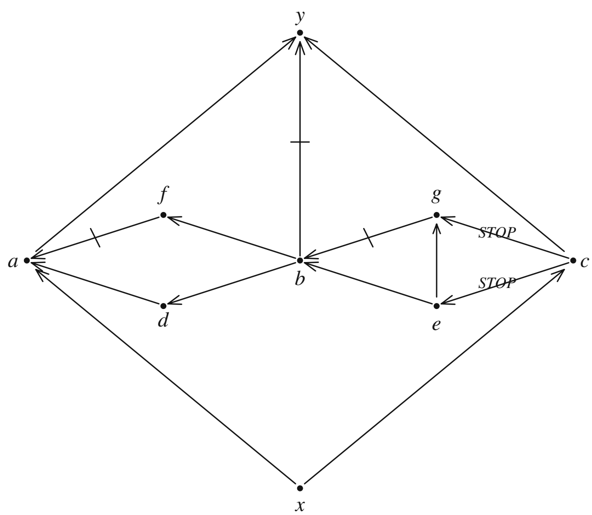


Diagram 9.1.6 STOP signs

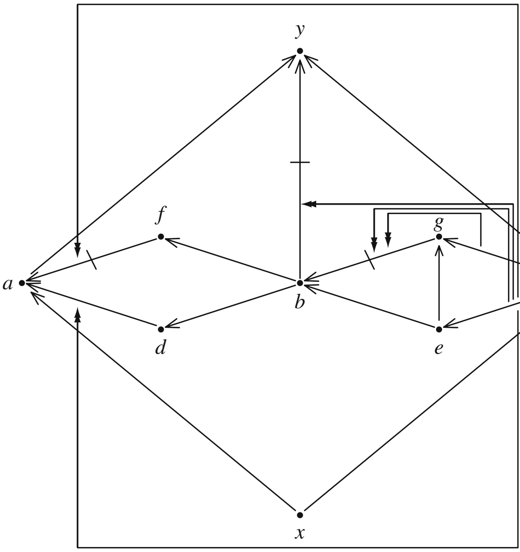


Diagram 9.1.7 Starting point c

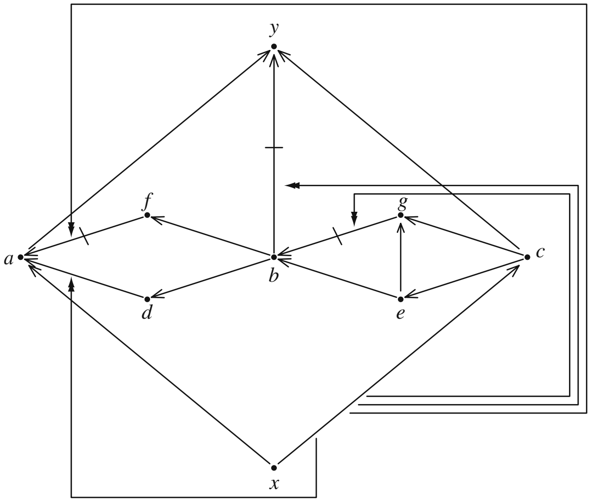


Diagram 9.1.8 Starting point x

9.1.3 Conceptual Analysis

Inheritance diagrams are deceptively simple. Their conceptually complicated nature is seen by, e.g. the fundamental difference between direct links and valid paths, and the multitude of existing formalisms, upward vs. downward chaining, intersection of extensions vs. direct scepticism, on-path vs. off-path preclusion (or pre-emption), split validity vs. total validity preclusion, etc. to name a few; see the discussion in Sect. 9.2.3 (p. 269). Such a proliferation of formalisms usually hints at deeper problems on the conceptual side, i.e. that the underlying ideas are ambiguous, and not sufficiently analysed. Therefore, any clarification and resulting reduction of possible formalisms seems a priori to make progress. Such clarification will involve conceptual decisions, which need not be shared by all, they can only be suggestions. Of course, a proof that such decisions are correct is impossible, and so is its contrary.

We will introduce into the analysis of inheritance systems a number of concepts not usually found in the field, like multiple truth values, access to information, comparison of truth values. We think that this additional conceptual burden pays off by a better comprehension and analysis of the problems behind the surface of inheritance.

We will also see that some distinctions between inheritance formalisms go far beyond questions of inheritance, and concern general problems of treating contradictory information – isolating some of these is another objective of this article.

The text is essentially self-contained, still some familiarity with the basic concepts of inheritance systems and nonmonotonic logics in general is helpful. For a presentation, the reader might look into [Sch97] and [Sch04].

The text is organized as follows. After an introduction to inheritance theory, connections with reactive diagrams in Sect. 9.3 (p. 270), and big and small subsets and the systems P and R in Sect. 9.2 (p. 258), we turn to an informal description of the fundamental differences between inheritance and the systems P and R in Sect. 9.4.2 (p. 276), give an analysis of inheritance systems in terms of information and information flow in Sect. 9.4.3 (p. 277), then in terms of reasoning with prototypes in Sect. 9.4.4 (p. 281), and conclude in Sect. 9.5 (p. 284) with a translation of inheritance into (necessarily deeply modified) coherent systems of big and small sets, respectively, logical systems P and R. One of the main modifications will be to relativize the notions of small and big, which thus become less "mathematically pure", but perhaps closer to actual use in "dirty" common sense reasoning.

9.2 Introduction to Nonmonotonic Inheritance

9.2.1 Basic Discussion

We give here an informal discussion. The reader unfamiliar with inheritance systems should consult in parallel Definition 9.2.3 (p. 265) and Definition 9.2.4 (p. 266). As there are many variants in the definitions, it seems reasonable to discuss them before a formal introduction, which, otherwise, would seem to pretend to be definite without being so.

9.2.1.1 (Defeasible or Nonmonotonic) Inheritance Networks or Diagrams

Nonmonotonic inheritance systems describe situations like "normally, birds fly", written $birds \rightarrow fly$. Exceptions are permitted, "normally penguins don't fly", $penguins \not\rightarrow fly$.

Definition 9.2.1 A nonmonotonic inheritance net is a finite DAG, directed, acyclic graph, with two types of arrows or links, $\rightarrow$ and $\not\rightarrow$, and labelled nodes. We will use Γ, etc. for such graphs, and σ, etc. for paths – the latter to be defined below.

Roughly (and to be made precise and modified below, we try to give here just a first intuition), $X \rightarrow Y$ means that "normal" elements of X are in Y, and $X \not\rightarrow Y$ means that "normal" elements of X are not in Y. In a semi-quantitative set interpretation, we will read "most" for "normal", thus "most elements of X are in Y", "most elements of X are not in Y", etc. These are by no means the only interpretations, as we will see – we will use these expressions for the moment just to help the reader's intuition. We should add immediately a word of warning: "most" is here not necessarily, but only by default, transitive, in the following sense. In the Tweety diagram, see Diagram 9.2.1 (p. 259) below, most penguins are birds, most birds fly, but it is not the case that most penguins fly. This is the problem of transfer of relative size which will be discussed extensively, especially in Sect. 9.5 (p. 284).

According to the set interpretation, we will also use informally expressions like $X \cap Y$, $X - Y$, $\boldsymbol{C}X$ – where $\boldsymbol{C}$ stands for set complement. But we will also use nodes informally as formulas, like $X \wedge Y$, $X \wedge \neg Y$, $\neg X$. All this will only be used here as an appeal to intuition.

Nodes at the beginning of an arrow can also stand for individuals, so $Tweety \not\rightarrow fly$ means something like: "Normally, Tweety will not fly". As always in nonmonotonic systems, exceptions are permitted, so the soft rules "birds fly", "penguins don't fly", and (the hard rule) "penguins are birds" can coexist in one diagram, penguins are then abnormal birds (with respect to flying). The direct link $penguins \not\rightarrow fly$ will thus be accepted, or considered valid, but not the composite path $penguins \rightarrow birds \rightarrow fly$, by specificity – see below. This is illustrated by Diagram 9.2.1 (p. 252), where a stands for Tweety, c for penguins, b for birds, and d for flying animals or objects.

(Remark: The arrows $a \rightarrow c$, $a \rightarrow b$, and $c \rightarrow b$ can also be composite paths – see below for the details.)

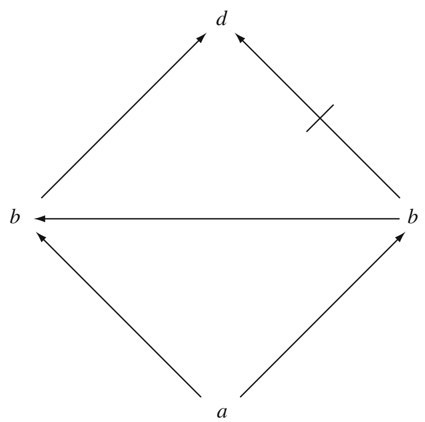


Diagram 9.2.1 Tweety diagram

(Of course, there is an analogous case for the opposite polarity, i.e. when the arrow from b to d is negative, and the one from c to d is positive.)

The main problem is to define in an intuitively acceptable way a notion of valid path, i.e. concatenations of arrows satisfying certain properties. We will write $\Gamma \models \sigma$, if σ is a valid path in the network Γ, and if x is the origin, and y the endpoint of σ, and σ is positive, we will write $\Gamma \models xy$, i.e. we will accept the conclusion that xs are ys, and analogously $\Gamma \models x\overline{y}$ for negative paths. Note that we will not accept any other conclusions, only those established by a valid path; so many questions about conclusions have a trivial negative answer: There is obviously no path from x to y. For example, there is no path from b to c in Diagram 9.2.1 (p. 256). Likewise, there are no disjunctions, conjunctions, etc. in our conclusions, and negation is present only in a strong form: "It is not the case that xs are normally ys" is not a possible conclusion, only "xs are normally not ys" is one. Also, possible contradictions are contained, there is no EFQ.

To simplify matters, we assume that for no two nodes $x, y \in \Gamma$ $x \to y$ and $x \not\to y$ are both in Γ, intuitively, that Γ is free from (hard) contradictions. This restriction is inessential for our purposes. We admit, however, soft contradictions and preclusion, which allows us to solve some soft contradictions – as we already did in the penguins example. We will also assume that all arrows stand for rules with possibly exceptions; again, this restriction is not important for our purposes. Moreover, in the abstract treatment, we will assume that all nodes stand for (nonempty) sets, though this will not be true for all examples discussed.

This might be the place for a remark on absence of cycles. Suppose we also have a positive arrow from b to c in Diagram 9.1.1 (p. 256). Then, the concept of preclusion collapses, as there are now equivalent arguments to accept $a \to b \to d$ and $a \to c \not\to d$. Thus, if we do not want to introduce new complications, we cannot rely on preclusion to decide conflicts. It seems that this would change the whole outlook on such diagrams. The interested reader will find more on the subject in [Ant97], [Ant99], [Ant05].

Inheritance networks were introduced about 20 years ago (see, e.g. [Tou84], [Tou86], [THT87]), and exist in a multitude of more or less differing formalisms; see, e.g. [Sch97] for a brief discussion. There still does not seem to exist a satisfying semantics for these networks. The authors' own attempt [Sch90] is an a posteriori semantics, which cannot explain or criticize or decide between different formalisms. We will give here a conceptual analysis, which provides also at least some building blocks for semantics, and a translation into (a modified version of) the language of small and big subsets, familiar from preferential structures; see, Sect. 3.2.2.6 (p. 58).

We will now discuss the two fundamental situations of contradictions, then give a detailed inductive definition of valid paths for a certain formalism so the reader has firm ground under his feet, and then present briefly some alternative formalisms.

As in all of nonmonotonic reasoning, the interesting questions arise in the treatment of contradictions and exceptions. The difference in quality of information is expressed by "preclusion" (or pre-emption). The basic diagram is the Tweety diagram, see, Diagram 9.2.1 (p. 256).

Unresolved contradictions give either rise to a branching into different extensions, which may roughly be seen as maximal consistent subsets, or to mutual cancellation in directly sceptical approaches. The basic diagram for the latter is the Nixon Diamond, see, Diagram 9.1.1 [(p. 257)], where $a = Nixon$, $b = Quaker$, $c = Republican$, $d = pacifist$.

In the directly sceptical approach, we will not accept any path from a to d as valid, as there is an unresolvable contradiction between the two candidates. The extensions approach can be turned into an indirectly sceptical one, by forming first all extensions, and then taking the intersection of either the sets of valid paths, or of valid conclusions, see [MS91] for a detailed discussion. See also Sect. 9.2.3 (p. 269) for more discussion on directly vs. indirectly sceptical approaches.

[We describe this now in more detail:]

9.2.1.2 Preclusion

In the above example, our intuition tells us that it is not admissible to conclude from the fact that penguins are birds, and that most birds fly that most penguins fly. The horizontal arrow $c \to b$ together with $c \not\to d$ bars this conclusion; it expresses specificity. Consequently, we have to define the conditions under which two potential paths neutralize each other, and when one is victorious. The idea is as follows: (1) We want to be sceptical, in the sense that we do not believe every potential path. We will not arbitrarily chose one either. (2) Our scepticism will be restricted, in the sense that we will often make well-defined choices for one path in the case of conflict: (a) If a compound potential path is in conflict with a direct link, the direct link wins. (b) Two conflicting paths of the same type neutralize each other, as in the Nixon Diamond, where neither potential path will be valid. (c) More specific information will win over less specific one.

(It is essential in the Tweety diagram that the arrow $c \not\to d$ is a direct link; so it is in a way stronger than compound paths.) The arrows $a \to b$, $a \to c$, $c \to b$ can also be composite paths: The path from c to b (read $c \subseteq \ldots \subseteq b$, where $\subseteq$ stands here for soft inclusion), however, tells us that the information coming from c is more specific (and thus considered more reliable), so the negative path from a to d via c will win over the positive one via b. The precise inductive definition will be given below. This concept is evidently independent of the length of the paths, $a \cdots \to c$ may be much longer than $a \cdots \to b$, so this is not shortest path reasoning (which has some nasty drawbacks, discussed in, e.g. [HTT87]).

A final remark: Obviously, in some cases, it need not be specificity, which decides conflicts. Consider the case where Tweety is a bird, but a dead animal. Obviously, Tweety will not fly, here because the predicate “dead” is very strong and overrules many normal properties. When we generalize this, we might have a hierarchy of causes, where one overrules the other, or the result may be undecided. For instance, a falling object might be attracted in a magnetic field, but a gusty wind might prevent this, sometimes, with unpredictable results. This is then additional information (strength of cause), and this problem is not addressed directly in traditional inheritance networks. We would have to introduce a subclass “dead bird” – and subclasses often have properties of "pseudo-causes", being a penguin probably is not a "cause" for not flying, nor bird for flying; still, things change from class to subclass for a reason. Before we give a formalism based on these ideas, we refine them, adopt one possibility (but indicate some modifications), and discuss alternatives later.

9.2.2 Directly Sceptical Split Validity Upward Chaining Off-Path Inheritance

Our approach will be directly sceptical, i.e. unsolvable contradictions result in the absence of valid paths, it is upward chaining, and split validity for preclusions (discussed below, in particular, in Sect. 9.2.3 (p. 269)). We will indicate modifications to make it extension based, as well as for total validity preclusion. This approach is strongly inspired by classical work in the field by Horty, Thomason, Touretzky, and others, and we claim no priority whatever. If it is new at all, it is a very minor modification of existing formalisms.

Our conceptual ideas to be presented in detail in Sect. 9.4.3 (p. 277) make split validity, off-path preclusion, and upward chaining a natural choice. For the reader's convenience, we give here a very short resume of these ideas: We consider only arrows as information, e.g. $a \to b$ will be considered information b valid at or for a. Valid composed positive paths will not be considered information in our sense. They will be seen as a way to obtain information, so a valid path $\sigma : x \ldots \to a$ makes information b accessible to x, and, second, as a means of comparing information strength, so a valid path $\sigma : a \ldots . \to a'$ will make information at a stronger than information at a'. Valid negative paths have no function; we will only consider the positive initial part as discussed above, and the negative end arrow as information, but never the whole path.

Choosing direct scepticism is a decision beyond the scope of this article, and we just make it. It is a general question how to treat contradictory and absent information, and if they are equivalent or not, see the remark in Sect. 9.4.4 (p. 281). (The fundamental difference between intersection of extensions and direct scepticism for defeasible inheritance was shown in [Sch93].) See also Sect. 9.2.3 (p. 269) for more discussion.

We now turn to the announced variants as well as a finer distinction within the directly sceptical approach. Again, see also Sect. 9.2.3 (p. 269) for more discussion. Our approach generates another problem, essentially that of the treatment of a mixture of contradictory and concordant information of multiple strengths or truth values. We bundle the decision of this problem with that for direct scepticism into a "plug-in" decision, which will be used in three approaches: the conceptual ideas, the inheritance algorithm, and the choice of the reference class for subset size (and implicitly also for the treatment as a prototype theory). It is thus well encapsulated and independent from the context.

These decisions (but, perhaps to a lesser degree, (1)) concern a wider subject than only inheritance networks. Thus, it is not surprising that there are different formalisms for solving such networks, deciding one way or the other. But this multitude is not the fault of inheritance theory, it is only a symptom of a deeper question. We first give an overview for a clearer overall picture, and discuss them in detail below, as they involve sometimes quite subtle questions.

(1) Upward chaining against downward or double chaining.
(2.1) Off-path against on-path preclusion.
(2.2) Split validity preclusion against total validity preclusion.
(3) Direct scepticism against intersection of extensions.
(4) Treatment of mixed contradiction and preclusion situations, no preclusion by paths of the same polarity.

(1) When we interpret arrows as causation (in the sense that $X \rightarrow Y$ expresses that condition X usually causes condition Y to result), this can also be seen as a difference in reasoning from cause to effect vs. backward reasoning, looking for causes for an effect. (A word of warning: There is a well-known article [SL89] from which a superficial reader might conclude that upward chaining is tractable and downward chaining is not. A more careful reading reveals that, on the negative side, the authors only show that double chaining is not tractable.) We will adopt upward chaining in all our approaches. See Sect. 9.4.4 (p. 281) for more remarks.
(2.1) and (2.2) Both are consequences of our view – to be discussed below in Sect. 4.3 – to see valid paths also as an absolute comparison of truth values, independent of reachability of information. Thus, in Diagram 9.2.1 (p. 256), the comparison between the truth values "penguin" and "bird" is absolute, and does not depend on the point of view "Tweety", as it can in total validity preclusion – if we continue to view preclusion as a comparison of information strength (or truth value). This question of absoluteness transcends obviously inheritance networks. Our decision is, of course, again uniform for all our approaches.
(3) This point, too, is much more general than the problems of inheritance. It is, among other things, a question of whether only the two possible cases (positive and negative) may hold, or whether there might be still other possibilities. See Sect. 9.4.4 (p. 281).
(4) This concerns the treatment of truth values in more complicated situations, where we have a mixture of agreeing and contradictory information. Again, this problem reaches far beyond inheritance networks.

We will group (3) and (4) together in one general, "plug-in" decision, to be found in all approaches we discuss.

Definition 9.2.2 This is an informal definition of a plug-in decision:

We describe now more precisely a situation which we will meet in all contexts discussed, and whose decision goes beyond our problem – thus, we have to adopt one or several alternatives, and translate them into the approaches we will discuss. There will be one global decision, which is (and can be) adapted to the different contexts.

Suppose we have information about ϕ and ψ, where ϕ and ψ are presumed to be independent – in some adequate sense. Suppose then that we have information sources $A_i : i \in I$ and $B_j : j \in J$, where the A_i speak about ϕ (they say ϕ or $\neg\phi$) and the B_j speak about ψ in the same way. Suppose further that we have a partial, not necessarily transitive (!), ordering $<$ on the information sources A_i and B_j together. $X < Y$ will say that X is better (intuition: more specific) than Y. (The potential lack of transitivity is crucial, as valid paths do not always concatenate to valid paths – just consider the Tweety diagram.)

We also assume that there are contradictions, i.e. some A_i say ϕ, some $\neg\phi$, likewise for the B_j – otherwise, there are no problems in our context. We can now take several approaches, all taking contradictions and the order $<$ into account.

- (P1) We use the global relation $<$, and throw away all information coming from sources of minor quality, i.e. if there is X such that $X < Y$, then no information coming from Y will be taken into account. Consequently, if Y is the only source of information about ϕ, then we will have no information about ϕ. This seems an overly radical approach, as one source might be better for ϕ, but not necessarily for ψ too.

  If we adopt this radical approach, we can continue as below, and can even split in analogue ways into (P1.1) and (P1.2), as we do below for (P2.1) and (P2.2).
- (P2) We consider the information about ϕ separately from the information about ψ. Thus, we consider for ϕ only the A_i, for ψ only the B_j. Take now, e.g. ϕ and the A_i. Again, there are (at least) two alternatives.

  – (P2.1) We eliminate again all sources among the A_i for which there is a better $A_{i'}$, irrespective of whether they agree on ϕ or not.

    - (a) If the sources left are contradictory, we conclude nothing about ϕ, and accept for ϕ none of the sources. (This is a directly sceptical approach of treating unsolvable contradictions, following our general strategy.)
    - (b) If the sources left agree for ϕ, i.e. all say ϕ or all say $\neg\phi$, then we conclude ϕ (or $\neg\phi$), and accept for ϕ all the remaining sources.

  – (P2.2) We eliminate again all sources among the A_i for which there is a better $A_{i'}$, but only if A_i and $A_{i'}$ have contradictory information. Thus, more sources may survive than in approach (P2.1).

  We now continue as for (P2.1):

  - (a) If the sources left are contradictory, we conclude nothing about ϕ, and accept for ϕ none of the sources.
  - (b) If the sources left agree for ϕ, i.e. all say ϕ or all say $\neg\phi$, then we conclude ϕ (or $\neg\phi$), and accept for ϕ all the remaining sources.

The difference between (P2.1) and (P2.2) is illustrated by the following simple example. Let $A < A' < A''$, but $A \not< A''$ (recall that $<$ is not necessarily transitive), and $A \models \phi$, $A' \models \neg\phi$, $A'' \models \neg\phi$. Then (P2.1) decides for ϕ (A is the only survivor), (P2.2) does not decide, as A and A'' are contradictory, and both survive in (P2.2).

There are arguments for and against either solution: (P2.1) gives a uniform picture, more independent from ϕ, (P2.2) gives more weight to independent sources, it "adds" information sources, and thus gives potentially more weight to information from several sources. (P2.2) seems more in the tradition of inheritance networks, so we will consider it in the further development.

The reader should note that our approach is quite far from a fixed-point approach in two ways: First, fixed-point approaches seem more appropriate for extension-based approaches, as both try to collect a maximal set of uncontradictory information. Second, we eliminate information when there is better, contradicting information, even if the final result agrees with the first. This, too, contradicts in spirit the fixed-point approach.

After these preparations, we turn to a formal definition of validity of paths.

9.2.2.1 The Definition of $\models$ (i.e. of Validity of Paths)

All definitions are relative to a fixed diagram Γ. The notion of degree will be defined relative to all nodes of Γ, as we will work with split validity preclusion, so the paths to consider may have different origins. For simplicity, we consider Γ to be just a set of points and arrows, thus, e.g. $x \to y \in \Gamma$ and $x \in \Gamma$ are defined, when x is a point in Γ, and $x \to y$ an arrow in Γ. Recall that we have two types of arrows, positive and negative ones.

We first define generalized and potential paths, then the notion of degree, and finally validity of paths, written $\Gamma \models \sigma$, if σ is a path, as well as $\Gamma \models xy$, if $\Gamma \models \sigma$ and $\sigma : x \ldots . \to y$.

Definition 9.2.3 (1) Generalized paths:

A generalized path is an uninterrupted chain of positive or negative arrows pointing in the same direction; more precisely:

$x \to p \in \Gamma \to x \to p$ is a generalized path,

$x \not\to p \in \Gamma \to x \not\to p$ is a generalized path.

If $x \cdots \to p$ is a generalized path, and $p \to q \in \Gamma$, then $x \cdots \to p \to q$ is a generalized path,

if $x \cdots \to p$ is a generalized path, and $p \not\to q \in \Gamma$, then $x \cdots \to p \not\to q$ is a generalized path.

(2) Concatenation:

If σ and τ are two generalized paths, and the end point of σ is the same as the starting point of τ, then $\sigma \circ \tau$ is the concatenation of σ and τ.

(3) Potential paths (pp.):

A generalized path contains atmost one negative arrow, and this at the end, is a potential path. If the last link is positive, it is a positive potential path, if not, a negative one.

(4) Degree:
As already indicated, we shall define paths inductively. As we do not admit cycles in our systems, the arrows define a well-founded relation on the vertices. Instead of using this relation for the induction, we shall first define the auxiliary notion of degree, and do induction on the degree. Given a node x (the origin), we need a (partial) mapping f from the vertices to natural numbers such that $p \to q$ or $p \not\to q \in \Gamma$ implies $f(p) < f(q)$, and define (relative to x) :

Let σ be a generalized path from x to y, then $deg_{\Gamma,x}(\sigma) := deg_{\Gamma,x}(y) :=$ the maximal length of any generalized path parallel to σ, i.e. beginning in x and ending in y.

Definition 9.2.4 Inductive definition of $\Gamma \models \sigma$:

Let σ be a potential path.

- Case I:
  σ is a direct link in Γ. Then $\Gamma \models \sigma$
  (Recall that we have no hard contradictions in Γ.)
- Case II:
  σ is a compound potential path, $deg_{\Gamma,a}(\sigma) = n$, and $\Gamma \models \tau$ is defined for all τ with degree less than n – whatever their origin and endpoint.
- Case II.1: Let σ be a positive pp. $x \cdots \to u \to y$, let $\sigma' := x \cdots \to u$, so $\sigma = \sigma' \circ u \to y$.
  Then, informally, $\Gamma \models \sigma$ iff

  (1) σ is a candidate by upward chaining,
  (2) σ is not precluded by more specific contradicting information,
  (3) all potential contradictions are themselves precluded by information contradicting them.

  Note that (2) and (3) are the translation of (P2.2) in Definition 9.2.2 (p. 263).
  Formally, $\Gamma \models \sigma$ iff

  (1) $\Gamma \models \sigma'$ and $u \to y \in \Gamma$.
  (The initial segment must be a path, as we have an upward chaining approach. This is decided by the induction hypothesis.)
  (2) There are no v, τ, τ' such that $v \not\to y \in \Gamma$ and $\Gamma \models \tau := x \cdots \to v$ and $\Gamma \models \tau' := v \cdots \to u$. ($\tau$ may be the empty path, i.e. $x = v$.)
  (σ itself is not precluded by split validity preclusion and a contradictory link. Note that $\tau \circ v \not\to y$ need not be valid; it suffices that it is a better candidate (by τ').)
  (3) All potentially conflicting paths are precluded by information contradicting them:
  For all v and τ such that $v \not\to y \in \Gamma$ and $\Gamma \models \tau := x \cdots \to v$ (i.e. for all potentially conflicting paths $\tau \circ v \not\to y$) there is z such that $z \to y \in \Gamma$ and either
  $z = x$
  (the potentially conflicting pp. is itself precluded by a direct link, which is thus valid)

or
there are $\Gamma \models \rho := x \cdots \rightarrow z$ and $\Gamma \models \rho' := z \cdots \rightarrow v$ for suitable ρ and ρ'.

- Case II.2: The negative case, i.e. σ a negative pp. $x \cdots \rightarrow u \not\rightarrow y$, $\sigma' := x \cdots \rightarrow u$, $\sigma = \sigma' \circ u \not\rightarrow y$ is entirely symmetrical.

Remark 9.2.1 The following remarks all concern preclusion.

(1) Thus, in the case of preclusion, there is a valid path from x to z, and z is more specific than v, so $\tau \circ v \not\rightarrow y$ is precluded. Again, $\rho \circ z \rightarrow y$ need not be a valid path, but it is a better candidate than $\tau \circ v \not\rightarrow y$ is, and as $\tau \circ v \not\rightarrow y$ is in simple contradiction, this suffices.
(2) Our definition is stricter than many popular ones in the following sense: We require – according to our general picture to treat only direct links as information – that the preclusion "hits" the precluded path at the end, i.e. $v \not\rightarrow y \in \Gamma$ and ρ' hits $\tau \circ v \not\rightarrow y$ at v. In other definitions it is possible that the preclusion hits at some v', which is somewhere on the path τ, and not necessarily at its end. For instance, in the Tweety diagram, see, Diagram 9.2.1 (p. 256), if there were a node b' between b and d, we will need the path $c \rightarrow b \rightarrow b'$ to be valid, (obvious) validity of the arrow $c \rightarrow b$ will not suffice.
(3) If we allow ρ to be the empty path, then the case $z = x$ is a subcase of the present one.
(4) Our conceptual analysis has led to a very important simplification of the definition of validity. If we adopt on-path preclusion, we have to remember all paths which led to the information source to be considered: In the Tweety diagram, we have to remember that there is an arrow $a \rightarrow b$; it is not sufficient to note that we somehow came from a to b by a valid path, as the path $a \rightarrow c \rightarrow b \rightarrow d$ is precluded, but not the path $a \rightarrow b \rightarrow d$. If we adopt total validity preclusion, see also Sect. 9.2.3 (p. 269) for more discussion, we have to remember the valid path $a \rightarrow c \rightarrow b$ to see that it precludes $a \rightarrow c \rightarrow d$. If we allow preclusion to "hit" below the last node, we also have to remember the entire path which is precluded. Thus, in all those cases, whole paths (which can be very long) have to be remembered, but NOT in our definition.
   We only need to remember (consider the Tweety diagram)

   (a) we want to know if $a \rightarrow b \rightarrow d$ is valid, so we have to remember a, b, d. Note that the (valid) path from a to b can be composed and very long.
   (b) we look at possible preclusions, so we have to remember $a \rightarrow c \not\rightarrow d$; again the (valid) path from a to c can be very long.
   (c) we have to remember that the path from c to b is valid (this was decided by induction before).

So in all cases (the last one is even simpler), we need only to remember the starting node, a (or c), the last node of the valid paths, b (or c), and the information $b \rightarrow d$ or $c \not\rightarrow d$ – i.e. the size of what has to be recalled is ≤ 3. (Of course, there may be many possible preclusions, but in all cases we have to look at a very limited situation, and not arbitrarily long paths.)

We take a fast look forward to Sect. 4.3, where we describe diagrams as information and its transfer, and nodes also as truth values. In these terms – and the reader is asked to excuse the digression – we may note above point (a) as $a \Rightarrow_b d$ – expressing that, seen from a, d holds with truth value b, (b) as $a \Rightarrow_c \neg d$, (c) as $c \Rightarrow_c b$ – and this is all we need to know.

We indicate here some modifications of the definition without discussion, which is to be found below.

(1) For on-path preclusion only: Modify condition (2) in Case II.1 to: (2′) There is no v on the path σ (i.e. $\sigma : x \cdots \to v \cdots \to u$) such that $v \not\to y \in \Gamma$.
(2) For total validity preclusion: Modify condition (2) in Case II.1 to: (2′) There are no v, τ, τ' such that $v \not\to y \in \Gamma$ and $\tau := x \cdots \to v$ and $\tau' := v \cdots \to u$ such that $\Gamma \models \tau \circ \tau'$.
(3) For extension-based approaches: Modify condition (3) in Case II.1 as follows: (3′) If there are conflicting paths, which are not precluded themselves by contradictory information, then we branch recursively (i.e. for all such situations) into two extensions: one, where the positive non-precluded paths are valid second; where the negative non precluded paths are valid.

Definition 9.2.5 Finally, define $\Gamma \models xy$ iff there is $\sigma : x \to y$ s.t. $\Gamma \models \sigma$, likewise for $x\overline{y}$ and $\sigma : x \cdots \not\to y$.

Diagram 9.2.2 (p. 257) shows the most complicated situation for the positive case.

We have to show now that the above approach corresponds to the preceding discussion.

Fact 9.2.1 The above definition and the informal one outlined in Definition 9.2.2 (p. 263) correspond, when we consider valid positive paths as access to information and comparison of information strength, as indicated at the beginning of Sect. 9.2.2 (p. 262) and elaborated in Sect. 9.4.3 (p. 277).

Proof As Definition 9.2.2 (p. 263) is informal, this cannot be a formal proof, but it is obvious how to transform it into one.

We argue for the result, the argument for valid paths is similar. Consider then case (P2.2) in Definition 9.2.2 (p. 263), and start from some x.

Case 1: Direct links, $x \to z$ or $x \not\to z$.

By comparison of strength via preclusion, as a direct link starts at x, the information z or $\neg z$ is stronger than all other accessible information. Thus, the link and the information will be valid in both approaches. Note that we assumed Γ free from hard contradictions.

Case 2: Composite paths.

In both approaches, the initial segment has to be valid, as information will otherwise not be accessible. Also, in both approaches, information will have the form of direct links from the accessible source. Thus, condition (1) in Case II.1 corresponds to condition (1) in Definition 9.2.2 (p. 263).

In both approaches, information contradicted by a stronger source (preclusion) is discarded, as well as information which is contradicted by other, not precluded sources; so (P2.2) in Definition 9.2.2 (p. 263) and II.1 (2) + (3) correspond. Note that variant (P2.1) of Definition 9.2.2 (p. 263) would give a different result – which we could, of course, also imitate in a modified inheritance approach.

Case 3: Other information.

Inheritance nets give no other information, as valid information is deduced only through valid paths by Definition 9.2.5 (p. 268), and we did not add any other information either in the approach in Definition 9.2.2 (p. 263). But as is obvious in Case 2, valid paths coincide in both cases.

Thus, both approaches are equivalent. □

9.2.3 Review of Other Approaches and Problems

We now discuss shortly in more detail some of the differences between various major definitions of inheritance formalisms. Diagram 6.8, p. 179, in [Sch97-1] (which is probably due to folklore of the field) shows requiring downward chaining would be wrong. We repeat it here, see, Diagram 9.1.2 (p. 259).

Preclusions valid above (here at u) can be invalid at lower points (here at z), as part of the relevant information is no longer accessible (or becomes accessible). We have $u \rightarrow x \not\rightarrow y$ valid, by downward chaining. Any valid path $z \rightarrow u \ldots . y$ has to have a valid final segment $u \ldots . y$, which can only be $u \rightarrow x \not\rightarrow y$, but intuition says that $z \rightarrow u \rightarrow v \rightarrow y$ should be valid. Downward chaining prevents such changes, and thus seems inadequate, so we decide for upward chaining. (Already preclusion itself underlines upward chaining: In the Tweety diagram, we have to know that the path from bottom up to penguins is valid. So at least some initial subpaths have to be known – we need upward chaining.) (The rejection of downward chaining seems at first sight to be contrary to the intuitions carried by the word "inheritance".) See also the remarks in Sect. 9.4.4 (p. 281).

9.2.3.1 Extension-Based vs. Directly Sceptical Definitions

As this distinction has already received detailed discussion in the literature, we shall be very brief here. An extension of a net is essentially a maximally consistent and in some appropriate sense reasonable subset of all its potential paths. This can, of course, be presented either as a liberal conception (focussing on individual extensions) or as a sceptical one (focusing on their intersection – or, the intersection of their conclusion sets). The seminal presentation is that of [Tou86], as refined by [San86]. The directly sceptical approach seeks to obtain a notion of sceptically accepted path and conclusion, but without detouring through extensions. Its classic presentation is that of [HTT87]. Even while still searching for fully adequate definitions of either kind, we may use the former approach as a useful "control" on the latter. For if we can find an intuitively possible and reasonable extension supporting a conclusion $x\overline{y}$, whilst a proposed definition for a directly sceptical notion of legitimate inference yields xy as a conclusion, then the counterexemplary extension seems to call into question the adequacy of the directly sceptical construction, more readily than inversely. It has been shown in [Sch93] that the intersection of extensions is fundamentally different from the directly sceptical approach. See also the remark in Sect. 9.4.4 (p. 281).

From now on, all definitions considered shall be (at least) upward chaining.

9.2.3.2 On-Path vs. Off-Path Preclusion

This is a rather technical distinction discussed in [THT87]. Briefly, a path $\sigma: x \to \ldots \to y \to \ldots \to z$ and a direct link $y \not\to u$ is an off-path preclusion of $\tau: x \to \ldots \to z \to \ldots \to u$, but an on-path preclusion only iff all nodes of τ between x and z lie on the path σ.

For instance, in the Tweety diagram, the arrow $c \not\to d$ is an on-path preclusion of the path $a \to c \to b \to d$, but the paths $a \to c$ and $c \to b$, together with $c \not\to d$, is an (split validity) off-path preclusion of the path $a \to b \to d$.

9.2.3.3 Split Validity vs. Total Validity Preclusion

Consider again a preclusion $\sigma : u \to \ldots \to x \to \ldots \to v$ and $x \not\to y$ of $\tau : u \to \ldots \to v \to \ldots \to y$. Most definitions demand for the preclusion to be effective – i.e. to prevent τ from being accepted – that the total path σ is valid. Some ([GV89], [KK89], [KKW89a], [KKW89b]) content themselves with the combinatorially simpler separate (split) validity of the lower and upper parts of σ: $\sigma' : u \to \ldots \to x$ and $\sigma'' : x \to \ldots \to v$. In Diagram 9.2.3 (p. 271), taken from [Sch97-1], the path $x \to w \to v$ is valid, so is $u \to x$, but not the whole preclusion path $u \to x \to w \to v$.

Thus, split validity preclusion will give here the definite result $u\overline{y}$. With total validity preclusion, the diagram has essentially the form of a Nixon Diamond.

9.3 Defeasible Inheritance and Reactive Diagrams

Before we discuss the relationship in detail, we first summarize our algorithm.

9.3.1 Summary of Our Algorithm

We look for valid paths from x to y.

(1) Direct arrows are valid paths.

Diagram 9.2.2 The complicated case

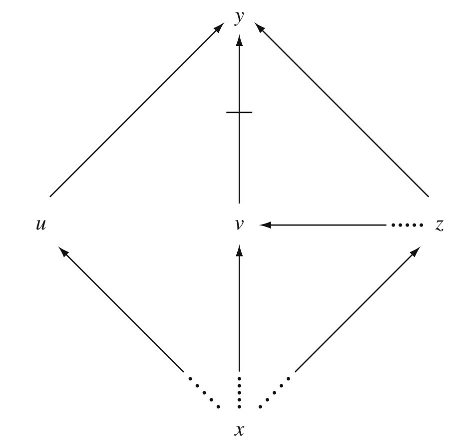
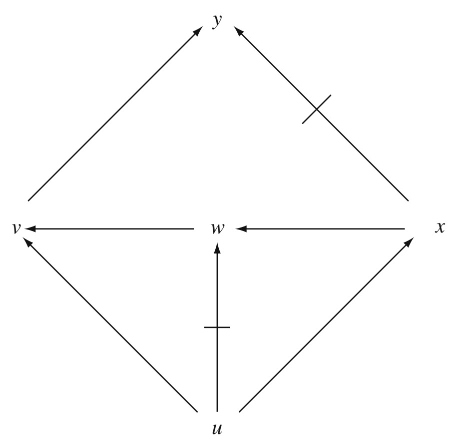


Diagram 9.2.3 Split vs. total validity preclusion



(2) Consider the set C of all direct predecessors of y, i.e. all c such that there is a direct link from c to y.

(2.1) Eliminate all those to which there is no valid positive path from x (found by induction); let the new set be $C' \subseteq C$.
If the existence of a valid path has not yet been decided, we have to wait.

(2.2) Eliminate from C' all c such that there is $c' \in C'$ and a valid positive path from c' to c (found by induction) – unless the arrows from c and from c' to y are of the same type. Let the new set be $C'' \subseteq C'$ (this handles preclusion).
If the existence of such valid paths has not yet been decided, we have to wait.

(2.3) If the arrows from all elements of C'' to y have same polarity, we have a valid path from x to y, if not, we have an unresolved contradiction, and there is no such path.

Note that we were a bit sloppy here. It can be debated whether preclusion by some c' such that c and c' have the same type of arrow to y should be accepted. As we are basically only interested whether there is a valid path, but not in its details, this does not matter.

9.3.2 Overview

There are several ways to use reactive graphs to help us solve inheritance diagrams – but also to go beyond them.

(1) We use them to record results found by adding suitable arrows to the graph.
(2) We go deeper into the calculating mechanism, and use new arrows not only as memory, but also for calculation.
(3) We can put up "signposts" to mark dead ends in the following sense: If we memorize valid paths from x to y, then, anytime we are at a branching point u coming from x, trying to go to y, and there is valid path through an arrow leaving u, we can put up a signpost saying "no valid path from x to y through this arrow".
    Note that we have to state destination y (of course), but also outset, x: There might be a valid path from u to y, which may be precluded or contradicted by some path coming from x.
(4) We can remember preclusion in the following sense: If we found a valid positive path from a to b, and there are contradicting arrows from a and b to c, then we can create an arrow from a to the arrow from b to c. So, if, from x, we can reach both a and b, the arrow from b to c will be incapacitated.

Before we discuss the first three possibilities in detail, we shortly discuss the more general picture (in rough outline).

(1) Replacing labels by arrows and vice versa.
    As we can switch arrows on and off, an arrow carries a binary value – even without any label. So the idea is obvious:
    If an arrow has one label with n possible values, we can replace it with n parallel arrows (i.e. same source and destination), where we switch exactly one of them on – this is the label.
    Conversely, we can replace n parallel arrows without labels, where exactly one is active, by one arrow with n labels.
    We can also code labels of a node x by an arrow $\alpha : x \to x$, which has the same labels.
(2) Coding logical formulas and truth in a model.
    We take two arrows for each propositional variable, one stands for true, the other for false. Negation blocks the positive arrow, enables the negative one. Conjunction is solved by concatenation, disjunction by parallel paths. If a variable occurs more than once, we make copies, which are "switched" by the "master arrow".

We come back to the first three ways to treat inheritance by reactive graphs, and also mention a way to go beyond usual inheritance.

9.3.3 Compilation and Memorization

When we take a look at the algorithm deciding which potential paths are valid, we see that, with one exception, we only need the results already obtained, i.e. whether there is a valid positive/negative path from a to b, and not the actual paths themselves. (This is, of course, due to our split validity definition of preclusion.) The exception is that preclusion works "in the upper part" with direct links. But this is a local problem: We only have to look at the direct predecessors of a node.

Consequently, we can do most of the calculation just once, in an induction of increasing "path distance", memorize valid positive (the negative ones cannot be used) paths with special arrows which we activated once their validity is established, and work now as follows with the new arrows:

Suppose we want to know if there is a valid path from x to y. We look backwards at all predecessors b of y (using a simple backward pointer), and look whether there is a valid positive path from x to b, using the new arrows. We then look at all arrows going from such b to y. If they all agree (i.e. all are positive or all are negative), we need not look further, and have the result. If they disagree, we have to look at possible comparisons by specificity. For this, we see whether there are new arrows between the bs. All such b to which there is a new arrow from another b are out of consideration. If the remaining agree, we have a valid path (and activate a new arrow from x to y if the path is positive), if not, there is no such path. (Depending on the technical details of the induction, it can be useful to note this also by activating a special arrow.)

9.3.4 Executing the Algorithm

Consider any two points x, y. There can be no path, a positive potential path, a negative potential path, both potential paths, a valid positive path, or a valid negative path (from x to y). Once a valid path is found, we can forget potential paths; so we can code the possibilities by $\{*, p+, p-, p+-, v+, v-\}$ in above order. We saw that we can either work with labels, or with a multitude of parallel arrows; we choose the first possibility. We create for each pair of nodes a new arrow, (x, y), which we intialize with label $*$.

First, we look for potential paths. If there is a direct link from x to y, we change the value $*$ of (x, y) directly to $v+$ or $v-$. If (x, y') has value $p+$, or $p+-$, or $v+$, and there is a direct link from y' to y, we change the value of (x, y) from $*$ to $p+$ or $p-$, depending on the link from y' to y, from $p+$ or $p-$ to $p+-$ if adequate (we found both possibilities), and leave the value unchanged otherwise. This determines all potential paths.

We make for each pair x, y at y a list of its predecessors, i.e. of all c s.t. there is a direct link from c to y. We do this for all x, so we can work in parallel. A list is, of course, a concatenation of suitable arrows. Suppose we want to know if there is valid path from x to y.

First, there might be a direct link, and we are done. Next, we look at the list of predecessors of y for x. If one c in the list has value $*$, $p-$, v$-$ for (x, c), it is eliminated from the list. If one c has $p+$ or $p+-$, we have to wait. We do this until all (x, c) have $*$, $v+$, or $v-$, so those remaining in the list will have $v+$. We look at all pairs in the list. While at least one (c, c') has $p+$ or $p+-$, we have to wait. Finally, all pairs will have $*$, $p-$, $v+$ or $v-$. Eliminate all c' s.t. there is (c, c') with value $v+$ – unless the arrows from c and c' to y are of the same type. Finally, we look at the list of the remaining predecessors, if they all have the same link to y. We set (x, y) to $v+$ or $v-$, otherwise to $*$. All such operations can be done by suitable operations with arrows, but it is very lengthy and tedious to write down the details.

9.3.5 Signposts

Putting up signposts requires memorizing all valid paths, as leaving one valid path does not necessarily mean that there is no alternative valid path. The easiest thing to do is probably to put up a warning post everywhere, and collect the wrong ones going backwards through the valid paths.

We illustrate this with Diagram 9.3.5 (p. 256):

There are the following potential paths from x to y : xcy, $xceb$-y, $xcebday$, xay.

The paths xc, xa, and $xceb$ are valid. The latter, xce is valid. $xceb$ is in competition with $xcg-b$ and $xceg-b$, but both are precluded by the arrow (valid path) eg. ys predecessors on those paths are a, b, c. a and b are not comparable, as there is no valid path from b to a, as bda is contradicted by $bf-a$. None is more specific than the other one. b and c are comparable, as the path ceb is valid, since $cg-b$ and $ceg-b$ are precluded by the valid path (arrow) eg. So c is more specific than b. Thus, b is out of consideration, and we are left with a and c. They agree, so there is positive valid path from x to y; more precisely, one through a, one through c. We have put STOP signs on the arrows ce and cg, as we cannot continue via them to y.

9.3.6 Beyond Inheritance

We can also go beyond usual inheritance networks.

Consider the following scenario:

- Museum airplanes usually will not fly, but usual airplanes will fly.
- Penguins don't fly, but birds do.
- Nonflying birds usually have fat wings.

- Nonflying aircraft usually are rusty.
- But it is not true that usually nonflying things are rusty and have fat wings.

We can model this with higher-order arrows as follows:

- Penguins $\to$ birds, museum airplanes $\to$ airplanes, birds $\to$ fly, airplanes $\to$ fly.
- Penguins $\not\to$ fly, museum airplanes $\not\to$ fly.
- Flying objects $\not\to$ rusty, flying objects $\not\to$ have fat wings.
- We allow concatenation of two negative arrows:

  For example, coming from penguins, we want to concatenate penguins $\not\to$ fly and fly $\not\to$ fat wings. Coming from museum aircraft, we want to concatenate museum airfcraft $\not\to$ fly and fly $\not\to$ rusty.

  We can enable this as follows: We introduce a new arrow α : (penguin $\not\to$ fly) $\to$ (fly $\not\to$ fat wings), which when traversing penguin $\not\to$ fly enables the algorithm to concatenate with the arrow it points to, using the rule "$- * - = +$", giving the result that penguins usually have fat wings.

See [Gab08d] for deeper discussion.

9.4 Interpretations

9.4.1 Introduction

We will discuss in this section three interpretations of inheritance nets.

First, we will indicate fundamental differences between inheritance and the systems P and R. They will be elaborated in Sect. 9.5 (p. 284), where an interpretation in terms of small sets will be tried nonetheless, and its limitations are explored.

Second, we will interpret inheritance nets as systems of information and information flow.

Third, we will interpret inheritance nets as systems of prototypes.

Inheritance nets present many intuitively attractive properties; thus, it is not surprising that we can interpret them in several ways. Similarly, preferential structures can be used as a semantics of deontic and of nonmonotonic logic; they express a common idea: choosing a subset of models by a binary relation. Thus, such an ambiguity need not be a sign for a basic flaw.

9.4.2 Informal Comparison of Inheritance with the Systems P and R

9.4.2.1 The Main Issues

In the authors' opinion, the following two properties of inheritance diagrams show the deepest difference to preferential and similar semantics, and the first even to classical logic. They have to be taken seriously, as they are at the core of inheritance systems, are independent of the particular formalism, and show that there is a fundamental difference between the former and the latter. Consequently, any attempt at translation will have to stretch one or both sides perhaps beyond the breaking point.

(1) Relevance and
(2) subideal situations or relative normality.

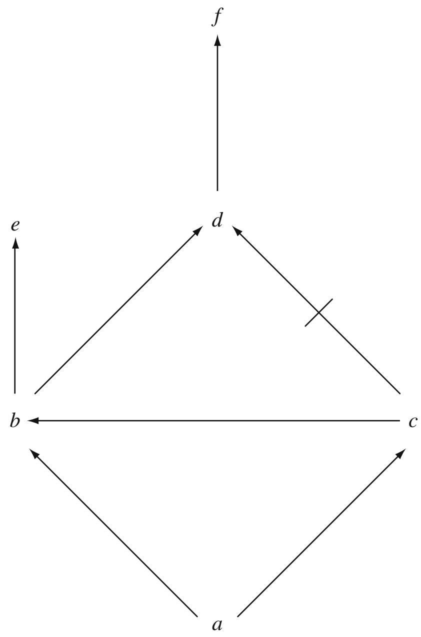
Both (and more) can be illustrated by the following simple Diagram 9.4.1 (p. 277) (which also shows conflict resolution by specificity).

(1) Relevance: As there is no monotonous path whatever between e and d, the question whether es are ds, or not, or vice versa, does not even arise. For the same reason, there is no question whether bs are cs, or not. (As a matter of fact, we will see below in Fact 9.5.1 (p. 286) that bs are *non*-cs in system P – see Definition 2.3 (p. 35).) In upward chaining formalisms, as there is no valid positive path from a to d, there is no question either whether as are fs or not. Of course, in classical logic, all information is relevant to the rest, so we can say, e.g. that es are ds, or es are *non*-ds, or some are ds, some are not, but there is a connection. As preferential models are based on classical logic, the same argument applies to them.
(2) In our diagram, as are bs, but not ideal bs, as they are not ds, the more specific information from c wins. But they are es, as ideal bs are. So they are not perfectly ideal bs, but as ideal bs as possible. Thus, we have graded ideality, which does not exist in preferential and similar structures. In those structures, if an element is an ideal element, it has all properties as such; if one such property is lacking, it is not ideal, and we can't say anything anymore. Here, however, we sacrifice as little normality as possible; it is thus a minimal change formalism.

In comparison, questions of information transfer and strength of information seem lesser differences. Already systems P and R (see Definition 2.3 (p. 35)) differ on information transfer. In both cases, transfer is based on the same notion of smallness, which describes ideal situations. But, as said in Remark 3.2.1 (p. 58), this is conceptually very different from the use of smallness, describing normal situations. Thus, it can be considered also on this level an independent question, and we can imagine systems based on absolutely ideal situations for normality, but with a totally different transfer mechanism.

For these reasons, extending preferential and related semantics to cover inheritance nets seems to stretch them to the breaking point. Thus, we should also look

Diagram 9.4.1 Information transfer



for other interpretations. (The term "interpretation" is used here in a nontechnical sense.) In particular, it seems worthwhile to connect inheritance systems to other problems, and see whether there are similarities there. This is what we do now. We come back to the systems P and R in Sect. 9.5 (p. 284).

Note that Reiter defaults behave much more like inheritance nets than like preferential logics.

9.4.3 Inheritance as Information Transfer

An informal argument showing parallel ideas common to inheritance with an upward chaining formalism and information transfer is as follows: First, arrows represent certainly some kind of information, of the kind "most as are bs" or so. (See, Diagram 9.4.1 (p. 277).) Second, to be able to use information, e.g. "ds are fs" at a, we have to be able to connect from a to d by a valid path. This information has to be made accessible to a, or, in other terms, a working information channel from a to d has to be established. Third, specificity (when present) decides conflicts (we take the split validity approach). This can be done procedurally, or, perhaps simpler and certainly in a more transparent way, by assigning a comparison of information strength to valid paths. Now, information strength may also be called truth value (to use a term familiar in logic) and the natural entity at hand is the node itself – this is just a cheap formal trick without any conceptual meaning.

When we adopt this view, nodes, arrows, and valid paths have multiple functions, and it may seem that we overload the (deceptively) simple picture. But, it is perhaps the charm and the utility and naturalness of inheritance systems that they are not "clean", and hide many complications under a simple surface, as human common sense reasoning often does, too.

In a certain way, this is a poor man's interpretation, as it does not base inheritance on another formalism, but gives only an intuitive reading. Yet, it gives a connection to other branches of reasoning, and is as such already justified – in the authors' opinion. Moreover, our analysis makes a clear distinction between arrows and composite valid paths. This distinction is implicit in inheritance formalisms, and we make it explicit through our concepts. But this interpretation is by no means the only one, and can only be suggested as a possibility.

We will now first give the details, and then discuss our interpretation.

9.4.3.1 Information

Direct positive or negative arrows represent information, valid for their source. Thus, in a set reading, if there is an arrow $A \rightarrow B$ in the diagram, most elements of A will be in B. In short: "most As are Bs" – and $A \not\rightarrow B$ will mean that most As are not Bs.

9.4.3.2 Information Sources and Flow

Nodes are information sources. If $A \rightarrow B$ is in the diagram, A is the source of the information "most As are Bs". A valid, composed or atomic positive path σ from U to A makes the information of source A accessible to U. One can also say that As information becomes relevant to U. Otherwise, information is considered independent – only (valid) positive paths create the dependencies.

(If we want to conform to inheritance, we must not add trivialities like "xs are xs", as this would require $x \rightarrow x$ in the corresponding net, which, of course, will not be there in an acyclic net.)

9.4.3.3 Information Strength

A valid, composed or atomic positive path σ from A' to A allows us to compare the strength of information source A' with that of A: A' is stronger than A. (In the set reading, this comparison is the result of specificity: more specific information is considered more reliable.) If there is no such valid path, we cannot resolve contradictions between information from A and A'. This interpretation results in split validity preclusion: the comparison between information sources A' and A is absolute, and does NOT depend on the U from which both may be accessible – as can be the case with total validity preclusion. Of course, if desired, we can also adopt the much more complicated idea of relative comparison.

Nodes are also truth values. They are the strength of the information whose source they are. This might seem an abuse of nodes, but we already have them, so why not use them?

9.4.3.4 Discussion

Considering direct arrows as information meets probably with little objection. The conclusion of a valid path (e.g. if $\sigma : a \ldots \to b$ is valid, then its conclusion is "*a*s are *b*s") is certainly also information, but it has a status different from the information of a direct link, so we should distinguish it clearly. At least in upward chaining formalisms, using the path itself as some channel through which information flows, and not the conclusion, seems more natural. The conclusion says little about the inner structure of the path, which is very important in inheritance networks, e.g. for preclusion. When calculating validity of paths, we look at (sub- and other) paths, but not just their results, and should also express this clearly.

Once we accept this picture of valid positive paths as information channels, it is natural to see their upper ends as information sources. Our interpretation supports upward chaining, and vice versa, upward chaining supports our interpretation.

One of the central ideas of inheritance is preclusion, which, in the case of split validity preclusion, works by an absolute comparison between nodes. Thus, if we accept split validity preclusion, it is natural to see valid positive paths as comparisons between information of different strengths. Conversely, if we accept absolute comparison of information, we should also accept split validity preclusion – these interpretations support each other.

Whatever type of preclusion we accept, preclusion clearly compares information strength, and allows us to decide for the stronger one. We can see this procedurally, or by giving different values to different information, depending on their sources, which we can call truth values to connect our picture to other areas of logic. It is then natural – as we have it already – to use the source node itself as truth value, with comparison via valid positive paths.

9.4.3.5 Illustration

Thus, in a given node U, information from A is accessible iff there is a valid positive path from U to A, and if information from A' is also accessible, and there is a valid positive path from A' to A, then, in case of conflict, information from A' wins over that from A, as A' has a better truth value. In the Tweety diagram, see Diagram 9.2.1 (p. 256), Tweety has access to penguins and birds, the horizontal link from penguin to bird compares the strengths, and the fly/not fly arrows are the information we are interested in.

Note that negative links and (valid) paths have much less function in our picture than positive links and valid paths. In a way, this asymmetry is not surprising, as there are no negative nodes (which would correspond to something like the set complement or negation). To summarize: A negative direct link can only be information. A positive direct link is information at its source, but it can also be a comparison of truth values, or it can give access from its source to information at its end. A valid, positive, composed path can only be comparison of truth values, or give access to information; it is NOT information itself in the sense of direct links. This distinction is very important, and corresponds to the different treatment of direct arrows and valid paths in inheritance, as it appears, e.g. in the definition of preclusion. A valid, negative, composed path has no function, only its parts have.

We obtain automatically that direct information is stronger than any other information: If A has information ϕ, and there is a valid path from A to B, making Bs information accessible to A, then this same path also compares strength, and As information is stronger than Bs information.

Inheritance diagrams in this interpretation represent not only reasoning with many truth values, but also reasoning *about* those truth values: Their comparison is done by the same underlying mechanism.

We should perhaps note in this context a connection to an area currently en vogue: The problem of trust, especially in the context of web information. We can see our truth values as the degrees of trust we put into information coming from this node, and, we not only use, but also reason about them.

9.4.3.6 Further Comments

Our reading also covers enriched diagrams, where arbitrary information can be "appended" to a node. An alternative way to see a source of information is to see it as a reason to believe the information it gives. U needs a reason to believe something, i.e. a valid path from U to the source of the information, and also a reason to disbelieve, i.e. if U' is below U, and U believes and U' does NOT believe some information of A, then either U' has stronger information to the contrary, or there is not a valid path to A anymore (and neither to any other possible source of this information). ("Reason", a concept very important in this context, was introduced by Bochman into the discussion.)

The restriction that negative links can only be information applies to traditional inheritance networks, and the authors make no claim whatever that it should also hold for modified such systems, or in still other contexts. One of the reasons why we do not have "negative nodes", and thus negated arrows also in the middle of paths might be the following (with $\boldsymbol{C}$ complementation): If, for some X, we also have a node for $\boldsymbol{C}X$, then we should have $X \not\to \boldsymbol{C}X$ and $\boldsymbol{C}X \not\to X$, thus a cycle, and arrows from Y to X should be accompanied by their opposite to $\boldsymbol{C}X$, etc.

We translate the analysis and decision of Definition 9.2.2 (p. 263) now into the picture of information sources, accessibility, and comparison via valid paths. This is straightforward:

(1) We have that information from A_i, $i \in I$, about B is accessible from U, i.e. there are valid positive paths from U to all A_i. Some A_i may say $\neg B$, some B.
(2) If information from A_i is comparable with information from A_j (i.e. there is a valid positive path from A_i to A_j or the other way around), and A_i contradicts A_j with respect to B, then the weaker information is discarded.

(3) There remains a (nonempty, by lack of cycles) set of A_i such that for no such A_i there is A_j with better contradictory information about B. If the information from this remaining set is contradictory, we accept none (and none of the paths either), if not, we accept the common conclusion and all these paths.

We now continue Remark 9.2.1 (p. 267), (4), and turn this into a formal system. Fix a diagram Γ, and do an induction as in Definition 9.2.2 (p. 263).

Definition 9.4.1 (1) We distinguish $a \Rightarrow b$ and $a \Rightarrow_x b$, where the intuition of $a \Rightarrow_x b$ is: we know with strength x that as are bs, and of $a \Rightarrow b$ that it has been decided taking all information into consideration that $a \Rightarrow b$ holds.
(We introduce this notation to point informally to our idea of information strength, and beyond, to logical systems with varying strengths of implications.)
(2) $a \to b$ implies $a \Rightarrow_a b$, likewise $a \not\to b$ implies $a \Rightarrow_a \neg b$.
(3) $a \Rightarrow_a b$ implies $a \Rightarrow b$, likewise $a \Rightarrow_a \neg b$ implies $a \Rightarrow \neg b$. This expresses the fact that direct arrows are uncontested.
(4) $a \Rightarrow b$ and $b \Rightarrow_b c$ imply $a \Rightarrow_b c$, likewise for $b \Rightarrow_b \neg c$. This expresses concatenation – but without deciding if it is accepted! Note that we cannot make ($a \Rightarrow b$ and $b \Rightarrow c$ imply $a \Rightarrow_b c$) a rule, as this would make concatenation of two composed paths possible.
(5) We decide acceptance of composed paths as in Definition 9.2.3 (p. 265), where preclusion uses accepted paths for deciding.

Note that we reason in this system not only with, but also about relative strength of truth values, which are just nodes. This is then, of course, used in the acceptance condition, in preclusion more precisely.

9.4.4 Inheritance as Reasoning with Prototypes

Some of the issues we discuss here apply also to the more general picture of information and its transfer. We present them here for motivational reasons: it seems easier to discuss them in the (somewhat!) more concrete setting of prototypes than in the very general situation of information handling. These issues will be indicated.

It seems natural to see information in inheritance networks as information about prototypes. (We do not claim that our use of the word "prototype" has more than a vague relation to the use in psychology. We do not try to explain the usefulness of prototypes either, one possibility is that there are reasons why birds fly, and why penguins don't, etc.) In the Tweety diagram, we will thus say that prototypical birds will fly, prototypical penguins will not fly. More precisely, the property "fly" is part of the bird prototype, is the property "$\neg fly$" is part of the penguin prototype. Thus, the information is given for some node, which defines its application or domain (bird or penguin in our example) – beyond this node, the property is not defined (unless inherited, of course). It might very well be that no element of the domain has ALL the properties of the prototype; every bird may be exceptional in some sense. This again shows that we are very far from the ideal picture of small and big subsets as used in systems P and R. (This, of course, goes beyond the problem of prototypes.)

Of course, we will want to "inherit" properties of prototypes; for instance, in Diagram 9.4.1 (p. 277), a "should" inherit the property e from b, and the property $\neg d$ from c. Informally, we will argue as follows: Prototypical as have property b and prototypical bs have property e; so it seems reasonable to assume that prototypical as also have property e – unless there is better information to the contrary. A plausible solution is then to use upward chaining inheritance as described above to find all relevant information, and then compose the prototype.

We now discuss three points whose importance goes beyond the treatment of prototypes:

(1) Using upward chaining has an additional intuitive appeal: We consider information at a the best, so we begin with b (and c), and only then, tentatively, add information e from b. Thus, we begin with strongest information, and add weaker information successively – this seems good reasoning policy.
(2) In upward chaining, we also collect information at the source (the end of the path), and do not use information which was already filtered by going down – thus the information we collect has no history, and we cannot encounter problems of iterated revision, which are problems of history of change. (In downward chaining, we only store the reasons why something holds, but not why something does not hold, so we cannot erase this negative information when the reason is not valid anymore. This is an asymmetry apparently not much noted before. Consider Diagram 9.1.2 (p. 259). Here, the reason why u does not accept y as information, but $\neg y$, is the preclusion via x. But from z, this preclusion is not valid anymore, so the reason why y was rejected is not valid anymore, and y can now be accepted.)
(3) We come back to the question of extensions vs. direct scepticism. Consider the Nixon Diamond, Diagram 9.1.1 (p. 257). Suppose Nixon were a subclass of Republican and Quaker. Then the extensions approach reasons as follows: Either the Nixon class prototype has the pacifist property, or the hawk property, and we consider these two possibilities. But this is not sufficient: The Nixon class prototype might have neither property – they are normally neither pacifists, nor hawks, but some are this, some are that. So the conceptual basis for the extensions approach does not hold: "Tertium non datur" just simply does not hold – as in intuitionist logic, where we may have a proof neither for ϕ, nor for $\neg\phi$.

Once we fixed this decision, i.e. how to find the relevant information, we can still look upward or downward in the net and investigate the changes between the prototypes in going upward or downward, as follows: For example, in the above example, we can look at the node a and its prototype, and then at the change going from a to b, or, conversely, look at b and its prototype, and then at the change going from b to a. The problem of finding the information, and this dynamics of information change have to be clearly separated.

In both cases, we see the following:

(1) The language is kept small, and thus efficient.
For instance, when we go from a to b, information about c is lost, and "c" does not figure anymore in the language, but f is added. When we go from b to a, f is lost, and c is won. In our simple picture, information is independent, and contradictions are always between two bits of information.

(2) Changes are kept small and need a reason to be effective. Contradictory, stronger information will override the old one, but no other information, except in the following case: Making new information (in-) accessible will cause indirect changes, i.e. information now made (in-) accessible via the new node. This is similar to formalisms of causation: if a reason is not there anymore, its effects vanish too.

It is perhaps more natural when going downward also to consider "subsets" as follows: Consider Diagram 9.4.1 (p. 277). bs are ds, and cs are $\neg d$s, and cs are also bs. So it seems plausible to go beyond the language of inheritance nets, and conclude that bs which are not cs will be ds, in short to consider $(b-c)$s. It is obvious which such subsets to consider, and how to handle them: For instance, loosely speaking, in $b \cap d$ e will hold, in $b \cap c \cap d$ $\neg f$ will hold, in $b \cap d \cap \boldsymbol{C}c$ f will hold, etc. This is just putting the bits of information together.

We turn to another consideration, which will also transcend the prototype situation, and we will (partly) use the intuition that nodes stand for sets and arrows for (soft, i.e. possibly with exceptions) inclusion in a set or its complement.

In this reading, specificity stands for soft, i.e. possibly with exceptions, set inclusion. So, if b and c are visible from a, and there is a valid path from c to b (as in Diagram 9.4.1 (p. 277)), then a is a subset of both b and c, and c a subset of b, so $a \subseteq c \subseteq b$ (softly). But then a is closer to c than a is to b. Automatically, a will be closest to itself. This results in a partial, and not necessarily transitive relation between these distances.

When we go now from b to c, we lose information d and f, win information $\neg d$, but keep information e. Thus, this is minimal change: We give up (and win) only the necessary information, but keep the rest. As our language is very simple, we can use the Hamming distance between formula sets here. (We will make a remark on more general situations just below.)

When we look now again from a, we take the set-closest class (c), and use the information of c, which was won by minimal change (i.e. the Hamming closest) from information of b. So, we have the interplay of two distances, where the set distance certainly is not symmetrical, as we need valid paths for access and comparison. If there is no such valid path, it is reasonable to make the distance infinite.

We make now the promised remark on more general situations: In richer languages, we cannot count formulas to determine the Hamming distance between two situations (i.e. models or model sets), but have to take the difference in propositional variables. Consider, e.g. the language with two variables, p and q. The models (described by) $p \wedge q$ and $p \wedge \neg q$ have distance 1, whereas $p \wedge q$ and $\neg p \wedge \neg q$ have distance 2. Note that this distance is NOT robust under redefinition of the language. Let p' stand for $(p \wedge q) \vee (\neg p \wedge \neg q)$ and q' for q. Of course, p' and q' are equivalent descriptions of the same model set, as we can define all the old singletons also in the new language. Then the situations $p \wedge q$ and $\neg p \wedge \neg q$ have now distance 1, as one corresponds to $p' \wedge q'$, the other to $p' \wedge \neg q'$.

There might be misunderstandings about the use of the word "distance" here. The authors are fully aware that inheritance networks cannot be captured by distance semantics in the sense of preferential structures. But we do NOT think here of distances from one fixed ideal point, but of relativized distances: Every prototype is the origin of measurements. For example, the bird prototype is defined by "flying, laying eggs, having feathers ...". So, we presume that all birds have these properties of the prototype, i.e. distance 0 from the prototype. When we see that penguins do not fly, we move as little as possible from the bird prototype, so we give up "flying", but not the rest. Thus, penguins (better: the penguin prototype) will have distance 1 from the bird prototype (just one property has changed). So there is a new prototype for penguins, and considering penguins, we will not measure from the bird prototype, but from the penguin prototype, so the point of reference changes. This is exactly as in distance semantics for theory revision, introduced in [LMS01], only the point of reference is not the old theory T, but the old prototype, and the distance is a very special one, counting properties assumed to be independent. (The picture is a little bit more complicated, as the loss of one property (flying) may cause other modifications, but the simple picture suffices for this informal argument.)

We conclude this section with a remark on prototypes. Realistic prototypical reasoning will probably neither always be upward nor always be downward. A medical doctor will not begin with the patient's most specific category (name and birthday or so), nor will he begin with all he knows about general objects. Therefore, it seems reasonable to investigate upward and downward reasoning here.

9.5 Detailed Translation of Inheritance to Modified Systems of Small Sets

For background material on abstract size semantics, the reader is referred to Chap. 3 (p. 53).

9.5.1 Normality

As we saw already in Sect. 9.4.2 (p. 276), normality in inheritance (and Reiter defaults, etc.) is relative, and as much normality as possible is preserved. There is no set of absolute normal cases of X, which we might denote $N(X)$, but only for ϕ a set $N(X, \phi)$, elements of X, which behave normally with respect to ϕ. Moreover, $N(X, \phi)$ might be defined, but not $N(X, \psi)$ for different ϕ and ψ. Normality in the sense of preferential structures is absolute: If x is not in $N(X)$ ($= \mu(X)$ in preferential reading), we do not know anything beyond classical logic. This is the dark Swedes' problem: Even dark Swedes should probably be tall. Inheritance systems are different: If birds usually lay eggs, then penguins, though abnormal with respect to flying, will still usually lay eggs. Penguins are fly-abnormal birds, but will continue to be egg-normal birds – unless we have again information to the contrary. So the absolute, simple $N(X)$ of preferential structures splits up into many, by default independent, normalities. This corresponds to intuition: There are no absolutely normal birds, each one is particular in some sense, so $\bigcap\{N(X,\phi) : \phi \in \mathcal{L}\}$ may well be empty, even if each single $N(X,\phi)$ is almost all birds.

What are the laws of relative normality? $N(X,\phi)$ and $N(X,\psi)$ will be largely independent (except for trivial situations, where $\phi \leftrightarrow \psi$, ϕ is a tautology, etc.). $N(X,\phi)$ might be defined and $N(X,\psi)$ not. Connections between the different normalities will be established only by valid paths. Thus, if there is no arrow, or no path, between X and Y, then $N(X,Y)$ and $N(Y,X)$ – where X, Y are also properties – need not be defined. This will get rid of the unwanted connections found with absolute normalities, as illustrated by Fact 9.5.1 (p. 286).

We interpret now "normal" by "big set", i.e. essentially "ϕ holds normally in X" iff "there is a big subset of X, where ϕ holds". This will, of course, be modified.

9.5.2 Small Sets

The main interest of this section is perhaps to show the adaptations of the concept of small and big subsets necessary for a more "real-life" situation, where we have to relativize. The amount of changes illustrates the problems and what can be done, but also perhaps what should not be done, as the concept is stretched too far. For more background, see Chap. 3 (p. 53).

As said, the usual informal way of speaking about inheritance networks (plus other considerations) motivates an interpretation by sets and soft set inclusion – $A \rightarrow B$ means that "most As are Bs". Just as with normality, the "most" will have to be relativized, i.e. there is a B-normal part of A, and a B-abnormal one, and the first is B-bigger than the second – where "bigger" is relative to B, too. A further motivation for this set interpretation is the often evoked specificity argument for preclusion. Thus, we will now translate our remarks about normality into the language of big and small subsets.

Consider now the system P (with cumulativity), see Definition 2.3 (p. 35). Recall from Remark 3.2.1 (p. 58) that small sets (see Sect. 3.2.2.6 (p. 58)) are used in two conceptually very distinct ways: $\alpha \mid\!\sim \beta$ iff the set of $\alpha \wedge \neg\beta$-cases is a small subset (in the absolute sense, there is just one system of big subsets of the α-cases) of the set of α-cases. The second use is in information transfer, used in cumulativity, or cautious monotony, more precisely: If the set of $\alpha \wedge \neg\gamma$-cases is a small subset of the set of α-cases, then $\alpha \mid\!\sim \beta$ carries over to $\alpha\wedge\gamma : \alpha\wedge\gamma \mid\!\sim \beta$. (See Chap. 3 (p. 53) and also the discussion in [Sch04], (p. 86), after Definition 2.3.6.) It is this transfer which we will consider here, and not things like AND, which connect different $N(X,\phi)$ for different ϕ.

Before we go into details, we will show that, e.g. the system P is too strong to model inheritance systems, and that, e.g. the system R is too weak for this purpose. Thus, preferential systems are really quite different from inheritance systems.

Fact 9.5.1

(a) System P is too strong to capture inheritance.
(b) System R is too weak to capture inheritance.

Proof (a) Consider the Tweety diagram, Diagram 9.2.1 (p. 256). $c \to b \to d$, $c \not\to d$. There is no arrow $b \not\to c$, and we will see that P forces one to be there. For this, we take the natural translation, i.e. $X \to Y$ will be "$X \cap Y$ is a big subset of X". We show that $c \cap b$ is a small subset of b, which we write $c \cap b < b$. $c \cap b = (c \cap b \cap d) \cup (c \cap b \cap \boldsymbol{C}d)$. $c \cap b \cap \boldsymbol{C}d \subseteq b \cap \boldsymbol{C}d < b$, the latter by $b \to d$, thus $c \cap b \cap \boldsymbol{C}d < b$, essentially by right weakening. Set now $X := c \cap b \cap d$. As $c \not\to d$, $X := c \cap b \cap d \subseteq c \cap d < c$, and by the same reasoning as above $X < c$. It remains to show $X < b$. We use now $c \to b$. As $c \cap \boldsymbol{C}b < c$ and $c \cap X < c$, by cumulativity $X = c \cap X \cap b < c \cap b$; so essentially by OR $X = c \cap X \cap b < b$. Using the filter property, we see that $c \cap b < b$.

(b) Second, even R is too weak: In the diagram $X \to Y \to Z$, we want to conclude that most of X is in Z, but as X might also be a small subset of Y, we cannot transfer the information "most Ys are in Z" to X. □

We have to distinguish direct information or arrows from inherited information or valid paths. In the language of big and small sets, it is easy to do this by two types of big subsets: big ones and very big ones. We will denote the first big, the second BIG. This corresponds to the distinction between $a \Rightarrow b$ and $a \Rightarrow_a b$ in Definition 9.4.1 (p. 281).

We will have the implications $BIG \to big$ and $SMALL \to small$, so we have nested systems. Such systems were discussed in [Sch95-1], see also [Sch97-1]. This distinction seems to be necessary to prevent arbitrary concatenation of valid paths to valid paths, which would lead to contradictions. Consider, e.g. $a \to b \to c \to d$, $a \to e \not\to d$, $e \to c$. Then concatenating $a \to b$ with $b \to c \to d$, both valid, would lead to a simple contradiction with $a \to e \not\to d$, and not to preclusion, as it should be – see below.

For the situation $X \to Y \to Z$, we will then conclude that:

If $Y \cap Z$ is a Z-BIG subset of Y and $X \cap Y$ is a Y-big subset of X, then $X \cap Z$ is a Z-big subset of X. (We generalize already to the case where there is a valid path from X to Y.)

We call this procedure information transfer.

$Y \to Z$ expresses the direct information in this context, so $Y \cap Z$ has to be a Z-BIG subset of Y. $X \to Y$ can be direct information, but it is used here as channel of information flow. In particular, it might be a composite valid path, so in our context, $X \cap Y$ is a Y-big subset of X. $X \cap Z$ is a Z-big subset of X: this can only be big, and not BIG, as we have a composite path.

The translation into big and small subsets and their modifications is now quite complicated: We seem to have to relativize, and we seem to need two types of big and small. This casts, of course, a doubt on the enterprise of translation. The future will tell if any of the ideas can be used in other contexts.

We investigate this situation now in more detail, first without conflicts. The way we cut the problem is not the only possible one. We were guided by the idea that we should stay close to usual argumentation about big and small sets, should proceed carefully, i.e. step by step, and should take a general approach.

Note that we start without any X-big subsets defined, so X is not even a X-big subset of itself.

(A) The simple case of two arrows, and no conflicts.

(In slight generalization:) If information ϕ is appended at Y, and Y is accessible from X (and there is no better information about ϕ available), ϕ will be valid at X. For simplicity, suppose there is a direct positive link from X to Y, written sloppily $X \to Y \models \phi$. In the big subset reading, we will interpret this as: $Y \wedge \phi$ is a ϕ-BIG subset of Y. It is important that this is now direct information, so we have "BIG" and not "big". We read now $X \to Y$ also as: $X \cap Y$ is a Y-big subset of X – this is the channel, so just "big". We want to conclude by transfer that $X \cap \phi$ is a ϕ-big subset of X.

We do this in two steps: First, we conclude that $X \cap Y \cap \phi$ is a ϕ-big subset of $X \cap Y$, and then, as $X \cap Y$ is a Y-big subset of X, $X \cap \phi$ itself is a ϕ-big subset of X. We do NOT conclude that $(X - Y) \cap \phi$ is a ϕ-big subset of $X - Y$. This is very important, as we want to preserve the reason of being ϕ-big subsets – and this goes via Y! The transition from "BIG" to "big" should be at the first step, where we conclude that $X \cap Y \cap \phi$ is a ϕ-big (and not ϕ-BIG) subset of $X \cap Y$, as it is really here where things happen, i.e. transfer of information from Y to arbitrary subsets $X \cap Y$.

We summarize the two steps in a slightly modified notation, corresponding to the diagram $X \to Y \to Z$:

(1) If $Y \cap Z$ is a Z-BIG subset of Y (by $Y \to Z$) and $X \cap Y$ is a Y-big subset of X (by $X \to Y$), then $X \cap Y \cap Z$ is a Z-big subset of $X \cap Y$.

(2) If $X \cap Y \cap Z$ is a Z-big subset of $X \cap Y$ and $X \cap Y$ is a Y-big subset of X (by $X \to Y$) again, then $X \cap Z$ is a Z-big subset of X, so $X \ldots \to Z$.

Note that (1) is very different from cumulativity or even rational monotony, as we do not say anything about X in comparison to Y: X need not be any big or medium size subset of Y.

Seen as strict rules, this will not work, as it would result in transitivity, and thus monotony: We have to admit exceptions, as there might just be a negative arrow $X \not\to Z$ in the diagram. We will discuss such situations below in (C), where we will modify our approach slightly, and obtain a clean analysis.

(Here and in what follows, we are very cautious and relativize all normalities. We could perhaps obtain our objective with a more daring approach, using absolute normality here and there. But this would be a purely technical trick (interesting in its own right), and we look here more for a conceptual analysis, and, as long as we do not find good conceptual reasons why to be absolute here and not there, we will just be relative everywhere.)

We now try to give justifications for the two (defeasible) rules. They will be philosophical and can certainly be contested and/or improved.

For (1): We look at Y. By $X \to Y$, Ys information is accessible at X, so, as Z-BIG is defined for Y, Z-big will be defined for $Y \cap X$. Moreover, there is a priori nothing which prevents X from being independent from Y, i.e. $Y \cap X$ to behave like Y with respect to Z – by default: of course, there could be a negative arrow $X \not\to Z$, which would prevent this. Thus, as $Y \cap Z$ is a Z-BIG subset of Y, $Y \cap X \cap Z$ should be a Z-big subset of $Y \cap X$. By the same argument (independence), we should also conclude that $(Y - X) \cap Z$ is a Z-big subset of $Y - X$. The definition of Z-big for $Y - X$ seems, however, less clear.

To summarize, $Y \cap X$ and $Y - X$ behave by default with respect to Z as Y does, i.e. $Y \cap X \cap Z$ is a Z-big subset of $Y \cap X$ and $(Y - X) \cap Z$ is a Z-big subset of $Y - X$. The reasoning is downward, from supersets to subsets, and symmetrical to $Y \cap X$ and $Y - X$. If the default is violated, we need a reason for it. This default is an assumption about the adequacy of the language. Things do not change wildly from one concept to another (or, better: from Y to $Y \wedge X$), they might change, but then we are told so – by a corresponding negative link in the case of diagrams.

For (2): By $X \to Y$, X and Y are related, and we assume that X behaves as $Y \cap X$ does with respect to Z. This is upward reasoning, from subset to superset and it is NOT symmetrical: There is no reason to suppose that $X - Y$ behaves the same way as X or $Y \cap X$ do with respect to Z, as the only reason for Z we have, Y, does not apply. Note that putting relativity aside (which can also be considered as being big and small in various, per-default independent dimensions) this is close to the reasoning with absolutely big and small sets: $X \cap Y - (X \cap Y \cap Z)$ is small in $X \cap Y$, so a fortiori small in X, and $X - (X \cap Y)$ is small in X, so $(X - (X \cap Y)) \cup (X \cap Y - (X \cap Y \cap Z))$ is small in X by the filter property, so $X \cap Y \cap Z$ is big in X, so a fortiori $X \cap Z$ is big in X.

Thus, in summary, we conclude by default that

(3) if $Y \cap Z$ is a Z-BIG subset of Y and $X \cap Y$ is a Y-big subset of X, then $X \cap Z$ is a Z-big subset of X.

(B) The case with longer valid paths, but without conflicts.
Treatment of longer paths: Suppose we have a valid, composed path from X to Y, $X \ldots \to Y$, and not any longer a direct link $X \to Y$. By induction, i.e.

upward chaining, we argue – using directly (3) – that $X \cap Y$ is a Y-big subset of X, and conclude by (3) again that $X \cap Z$ is a Z-big subset of X.

(C) Treatment of multiple and perhaps conflicting information.

Consider Diagram 9.5.1 (p. 291).

We want to analyse the situation and argue that, e.g. X is mostly not in Z.

First, all arguments about X and Z go via the Ys. The arrows from X to Ys, and from Y' to Y could also be valid paths. We look at information which concerns Z (thus U is not considered), and which is accessible (thus Y'' is not considered). We can be slightly more general and consider all possible combinations of accessible information, not only those used in the diagram by X. Instead of arguing on the level of X, we will argue one level above, on the Ys and their intersections, respecting specificity and unresolved conflicts.

(Note that in more general situations, with arbitrary information appended, the problem is more complicated, as we have to check which information is relevant for some ϕ – conclusions can be arrived at by complicated means, just as in ordinary logic. In such cases, it might be better to look first at all accessible information for a fixed X, then at the truth values and their relation, and calculate closure of the remaining information.)

We then have (using the obvious language: "most As are Bs" for "$A \cap B$ is a big subset of A", and "MOST As are Bs" for "$A \cap B$ is a BIG subset of A"):

In Y, Y'', and $Y \cap Y''$, we have that MOST cases are in Z. In Y' and $Y \cap Y'$, we have that MOST cases are not in Z (= are in $\boldsymbol{C}Z$). In $Y' \cap Y''$ and $Y \cap Y' \cap Y''$, we are UNDECIDED about Z.

Thus

$Y \cap Z$ will be a Z-BIG subset of Y, $Y'' \cap Z$ will be a Z-BIG subset of Y'', $Y \cap Y'' \cap Z$ will be a Z-BIG subset of $Y \cap Y''$.

$Y' \cap \boldsymbol{C}Z$ will be a Z-BIG subset of Y', $Y \cap Y' \cap \boldsymbol{C}Z$ will be a Z-BIG subset of $Y \cap Y'$.

$Y' \cap Y'' \cap Z$ will be a Z-MEDIUM subset of $Y' \cap Y''$, $Y \cap Y' \cap Y'' \cap Z$ will be a Z-MEDIUM subset of $Y \cap Y' \cap Y''$.

This is just simple arithmetic of truth values, using specificity and unresolved conflicts, and the nonmonotonicity is pushed into the fact that subsets need not preserve the properties of supersets.

In more complicated situations, we implement, e.g. the general principle (P2.2) from Definition 9.2.2 (p. 263), to calculate the truth values. This will use in our case specificity for conflict resolution, but it is an abstract procedure, based on an arbitrary relation $<$.

This will result in the "correct" truth value for the intersections, i.e. the one corresponding to the other approaches.

It remains to do two things: (C.1) We have to assure that X "sees" the correct information, i.e. the correct intersection and, (C.2), that X "sees" the accepted Ys, i.e. those through which valid paths go, in order to construct not only the result, but also the correct paths.

(Note that by split validity preclusion, if there is valid path from A through B to C, $\sigma : A \cdots \to B$, $B \to C$, and $\sigma' : A \cdots \to B$ is another valid path from A to B, then $\sigma' \circ B \to C$ will also be a valid path. Proof: If not, then $\sigma' \circ B \to C$ is precluded, but the same preclusion will also preclude $\sigma \circ B \to C$ by split validity preclusion, or it is contradicted, and a similar argument applies again. This is the same argument as the one for the simplified definition of preclusion – see Remark 9.2.1 (p. 267), (4).)

(C.1) Finding and inheriting the correct information:

X has access to Z-information from Y and Y', so we have to consider them. Most of X is in Y, most of X is in Y', i.e. $X \cap Y$ is a Y-big subset of X, $X \cap Y'$ is a Y'-big subset of X, so $X \cap Y \cap Y'$ is a $Y \cap Y'$-big subset of X; thus most of X is in $Y \cap Y'$.

We thus have Y, Y', and $Y \cap Y'$ as possible reference classes, and use specificity to choose $Y \cap Y'$ as reference class. We do not know anything, e.g. about $Y \cap Y' \cap Y''$; so this is not a possible reference class.

Thus, we use specificity twice, on the Y's-level (to decide that $Y \cap Y'$ is mostly not in Z), and on X's-level (the choice of the reference class), but this is good policy, as, after all, much of nonmonotonicity is about specificity.

We should emphasize that nonmonotonicity lies in the behaviour of the subsets, determined by truth values and comparisons thereof, and the choice of the reference class by specificity. But both are straightforward now and local procedures, using information already decided before. There is no complicated issue here like determining extensions.

We now use above argument, described in the simple case, but with more detail, speaking in particular about the most specific reference class for information about Z, $Y \cap Y'$ in our example – this is used essentially in (1.4), where the "real" information transfer happens, and where we go from BIG to big.

(1.1) By $X \to Y$ and $X \to Y'$ (and there are no other Z-relevant information sources), we have to consider $Y \cap Y'$ as reference class.

(1.2) $X \cap Y$ is a Y-big subset of X (by $X \to Y$) (it is even Y-BIG, but we are immediately more general to treat valid paths), $X \cap Y'$ is a Y'-big subset of X (by $X \to Y'$). So $X \cap Y \cap Y'$ is a $Y \cap Y'$-big subset of X.

(1.3) $Y \cap Z$ is a Z-BIG subset of Y (by $Y \to Z$), $Y' \cap \boldsymbol{C}Z$ is a Z-BIG subset of Y' (by $Y' \not\to Z$), so by preclusion $Y \cap Y' \cap \boldsymbol{C}Z$ is a Z-BIG subset of $Y \cap Y'$.

(1.4) $Y \cap Y' \cap \boldsymbol{C}Z$ is a Z-BIG subset of $Y \cap Y'$, and $X \cap Y \cap Y'$ is a $Y \cap Y'$-big subset of X, so $X \cap Y \cap Y' \cap \boldsymbol{C}Z$ is a Z-big subset of $X \cap Y \cap Y'$.

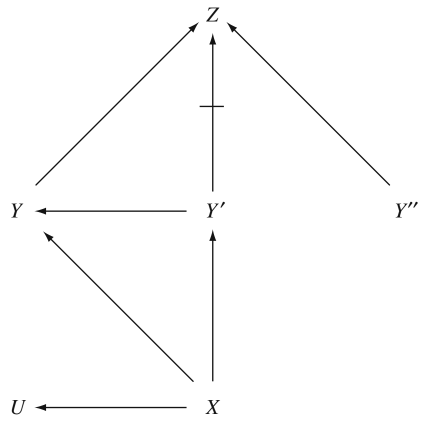
This cannot be a strict rule without the reference class, as it would then apply to $Y \cap Z$, too, leading to a contradiction.

(2) If $X \cap Y \cap Y' \cap \boldsymbol{C}Z$ is a Z-big subset of $X \cap Y \cap Y'$, and $X \cap Y \cap Y'$ is a $Y \cap Y'$-big subset of X, so $X \cap \boldsymbol{C}Z$ is a Z-big subset of X.

We make this now more formal.

We define for all nodes X, Y two sets: $B(X, Y)$ and $b(X, Y)$, where $B(X, Y)$ is the set of Y-BIG subsets of X and $b(X, Y)$ is the set of Y-big subsets of X. (To distinguish undefined from medium/MEDIUM-size, we will also have to define $M(X, Y)$ and $m(X, Y)$, but we omit this here for simplicity.)

Diagram 9.5.1 Multiple and conflicting information



The translations are then:

(1.2′) $X \cap Y \in b(X, Y)$ and $X \cap Y' \in b(X, Y') \Rightarrow X \cap Y \cap Y' \in b(X, Y \cap Y')$;
(1.3′) $Y \cap Z \in B(Y, Z)$ and $Y' \cap \boldsymbol{C}Z \in B(Y', Z) \Rightarrow Y \cap Y' \cap \boldsymbol{C}Z \in B(Y \cap Y', Z)$ by preclusion;
(1.4′) $Y \cap Y' \cap \boldsymbol{C}Z \in B(Y \cap Y', Z)$ and $X \cap Y \cap Y' \in b(X, Y \cap Y') \Rightarrow X \cap Y \cap Y' \cap \boldsymbol{C}Z \in b(X \cap Y \cap Y', Z)$ as $Y \cap Y'$ is the most specific reference class;
(2′) $X \cap Y \cap Y' \cap \boldsymbol{C}Z \in b(X \cap Y \cap Y', Z)$ and $X \cap Y \cap Y' \in b(X, Y \cap Y') \Rightarrow X \cap \boldsymbol{C}Z \in b(X, Z)$.
Finally,
(3′) $A \in B(X, Y) \rightarrow A \in b(X, Y)$, etc.

Note that we used, in addition to the set rules, preclusion, and the correct choice of the reference class.

(C.2) Finding the correct paths:

Idea:

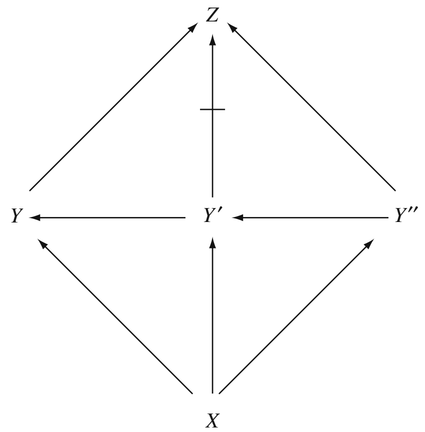
(1) If we come to no conclusion, then no path is valid, this is trivial.
(2) If we have a conclusion:
(2.1) All contradictory paths are out: e.g. $Y \cap Z$ will be Z-big, but $Y \cap Y' \cap \boldsymbol{C}Z$ will be Z-big. So there is no valid path via Y.
(2.2) Thus, not all paths supporting the same conclusion are valid.

Consider the following Diagram 9.5.2 (p. 292):

There might be a positive path through Y, a negative one through Y', a positive one through Y'' again, with $Y'' \rightarrow Y' \rightarrow Y$, so Y will be out, and only Y'' in. We can see this, as there is a subset, $\{Y, Y'\}$ which shows a change: $Y' \cap Z$ is Z-BIG, $Y' \cap \boldsymbol{C}Z$ is Z-BIG, $Y'' \cap Z$ is Z-BIG, and $Y \cap Y' \cap \boldsymbol{C}Z$ is Z-BIG, and the latter can only happen if there is a preclusion between Y' and Y, where Y looses. Thus, we can see this situation by looking only at the sets.

We now show equivalence with the inheritance formalism given in Definition 9.2.3 (p. 265).

Diagram 9.5.2 Valid paths or conclusions



Fact 9.5.2 The above definition and the one outlined in Definition 9.2.3 (p. 265) correspond.

Proof By induction on the length of the deduction that $X \cap Z$ (or $X \cap \boldsymbol{C}Z$) is a Z-big subset of X. (Outline)

It is a corollary of the proof that we have to consider only subpaths and information of all generalized paths between X and Z. Make all sets (i.e. one for every node) sufficiently different, i.e. all sets and Boolean combinations of sets differ by infinitely many elements, e.g. $A \cap B \cap C$ will have infinitely many less elements than $A \cap B$, etc. (Infinite is far too many, we just choose it by laziness to have room for the $B(X, Y)$ and the $b(X, Y)$. Put in $X \cap Y \in B(X, Y)$ for all $X \rightarrow Y$, and $X \cap \boldsymbol{C}Y \in B(X, Y)$ for all $X \not\rightarrow Y$ as base theory.

Length $= 1$: Then big must be BIG, and, if $X \cap Z$ is a Z-BIG subset of X, then $X \rightarrow Z$, likewise for $X \cap \boldsymbol{C}Z$.

We stay close now to above Diagram 9.5.1 (p. 292), so we argue for the negative case. Suppose that we have deduced $X \cap \boldsymbol{C}Z \in b(X, Z)$, we show that there must be a valid negative path from X to Z. (The other direction is easy.) Suppose for simplicity that there is no negative link from X to Z – otherwise we are finished. As we can distinguish intersections from elementary sets (by the starting hypothesis about sizes), this can only be deduced using (2′). So there must be some suitable $\{Y_i : i \in I\}$ and we must have deduced $X \cap \bigcap Y_i \in b(X, \bigcap Y_i)$, the second hypothesis of (2′). If I is a singleton, then we have the induction hypothesis, so there is a valid path from X to Y. So suppose I is not a singleton. Then the deduction of $X \cap \bigcap Y_i \in b(X, \bigcap Y_i)$ can only be done by (1.2′), as this is the only rule having in the conclusion an elementary set on the left in $b(\ldots)$, and a true intersection on the right. Going back along (1.2′), we find $X \cap Y_i \in b(X, Y_i)$, and by the induction hypothesis, there are valid paths from X to Y_i.

The first hypothesis of (2′), $X \cap \bigcap Y_i \cap \boldsymbol{C}Z \in b(X \cap \bigcap Y_i, Z)$ can be obtained by (1.3′) or (1.4′). If it was obtained by (1.3′), then X is one of the Y_i, but then there is a direct link from X to Z (due to the "B", BIG). As a direct link always wins by specificity, the link must be negative, and we have a valid, negative path from X to Z. If it was obtained by $(1.4')$, then its first hypothesis $\bigcap Y_i \cap \boldsymbol{C}Z \in B(\bigcap Y_i, Z)$ must have been deduced, which can only be by $(1.3')$, but the set of Y_i there was chosen to take all Y_i into account for which there is a valid path from X to Y_i and arrows from Y_i to Z (the rule was only present for the most specific reference class with respect to X and Z!), and we are done by the definition of valid paths in Sect. 9.2 (p. 258). □

We summarize our ingredients.

Inheritance was done essentially by (1) and (2) of (A) above and its elaborations $(1.i)$, (2) and $(1.i')$, $(2')$. It consisted of a mixture of bold and careful (in comparison to systems P and R) manipulation of big subsets. We had to be bolder than the systems P and R are, as we have to transfer information also to perhaps small subsets. We had to be more careful, as P and R would have introduced far more connections than are present. We also saw that we are forced to loose the paradise of absolute small and big subsets, and have to work with relative size.

We then have a plug-in decision what to do with contradictions. This is a plug-in, as it is one (among many possible) solutions to the much more general question of how to deal with contradictory information in the presence of a (partial, not necessarily transitive) relation which compares strength. At the same place in our procedure, we can plug-in other solutions, so our approach is truly modular in this aspect. The comparing relation is defined by the existence of valid paths, i.e. by specificity. This decision is inherited downward using again the specificity criterion.

Perhaps, the deepest part of the analysis can be described as follows: Relevance is coded by positive arrows, and valid positive paths, and thus is similar to Kripke structures for modality, where the arrows code dependencies between the situations for the evaluation of the modal quantifiers. In our picture, information at A can become relevant only to node B iff there is a valid positive path from B to A. But, relevance (in this reading, which is closely related to causation) is profoundly non-monotonic, and any purely monotonic treatment of relevance would be insufficient. This seems to correspond to intuition. Relevance is then expressed in our translation to small and big sets formally by the possibility of combining different small and big sets in information transfer. This is, of course, a special form of relevance, and there might be other forms.