Dialectics and Specificity: Conditioning in Logic-based Hypothetical Reasoning (Preliminary Report)*

David Poole
Department of Computer Science,
The University of British Columbia,
Vancouver, B.C., Canada V6T 1W5
poole@cs.ubc.ca

Abstract

In this paper we start with defaults as possible hypotheses and prediction as membership in all extensions. It is argued that this is too conservative and does not allow many intuitive answers. We show how viewing membership in all extensions as a form of dialectic, and adding a notion of conditioning can produce more intuitive answers. Defaults are possible hypotheses for a logical argument that contain a pragmatic component that is a context in which we know the default is applicable. This context is used to ignore counter arguments that follow from the context of the default. The conditioning that is presented is very close to the irrelevance that Geffner added to ϵ-semantics, and the resulting solutions turn out to be very similar.

1 Introduction

When considering default knowledge, there is a very strong notion that we should prefer more specific knowledge over more general knowledge [Touretzky, 1986, Poole, 1985, Loui, 1987, Geffner, 1988]. In probability theory this is accomplished by conditioning [Pearl, 1988]. In this paper, we show how a form of conditioning can be added to a logic-based hypothetical reasoning system. The resulting system is simple, can be easily implemented and solves many problems in a natural, straight-forward manner.

This work uses the first order predicate calculus; default reasoning is accomplished by allowing the assumption and criticism of premises in logical arguments. The use of conditioning has been inspired by probability, particularly the work of [Pearl, 1988, Geffner, 1988, Neufeld and Poole, 1988].

*This a revised version (section 5) of a paper in *Eighth Biennial Conference of the Canadian Society for Computational Studies of Intelligence (CSCSI-90)*

1.1 Logic-based Hypothetical Reasoning

Monotonicity has often been cited as a problem with using logic as a basis for commonsense reasoning. In [Poole, 1988, Poole, 1989b] it was argued that instead of deduction from our knowledge, reasoning should be viewed as a process of theory formation. In [Poole, 1988] it was shown how default reasoning can be viewed in this way by treating defaults as possible hypotheses that can be used in an explanation. In [Poole, 1989b] it was shown how membership in all extensions can form the basis for prediction and can be implemented as a process of dialectics.

1.2 Dialectics

The idea behind dialectics [Loui, 1990] is that a conclusion is reached by a process of argumentation. One agent comes up with an argument for a proposition; another agent can criticise the argument by coming up with a counter argument. In the Theorist framework all of the arguments are valid deductions; the premises are background knowledge, knowledge of the case at hand and assumptions.

One particularly appealing framework [Poole, 1989b] is where there are two agents. One agent finds arguments for a proposition. The other agent tries to either dismiss the argument out of hand (by showing it is inconsistent), or create an argument against the premises of the first agent's arguments. This idea is developed in more detail in section 2.2.

This implements membership in all extensions which is (propositionally, at least) equivalent to circumscription [Etherington, 1988]. This dialectical implementation [Poole, 1989b] provides an abstract specification of recent implementations of circumscription [Przymusinski, 1989, Ginsberg, 1989, Inoue and Helft, 1990]. In this paper it is argued that this notion of prediction is too restrictive, but is a good starting point for different argument forms.

1.3 Background and Contingent Knowledge

Consider the following example:

Example 1.1 Suppose we have as defaults "birds fly", "emus don't fly", and as facts "emus are birds" and "Tweety is an emu". There is a very strong preference for concluding "Tweety doesn't fly" based on specificity [Touretzky, 1986, Poole, 1985, Loui, 1987, Thomason and Horty, 1988]. We prefer to use the more specific knowledge about emus over the more general knowledge about birds.

The instances of the facts that are relevant to the conclusion are

$$emu(tweety) \wedge (emu(tweety) \Rightarrow bird(tweety)) \quad (1)$$

Using the same defaults, if we change the facts by swapping the role of *emu* and *bird* the answer should be, by symmetry, that Tweety does fly (i.e., the opposite of the previous conclusion). The instance of the facts used would be

$$bird(tweety) \wedge (bird(tweety) \Rightarrow emu(tweety)) \quad (2)$$

It is important to notice that formulae (1) and (2) are logically equivalent. Notice also that we have only talked about the facts and not about the defaults.

This seems to indicate that defining specificity, with logically equivalent instances of facts treated identically, is impossible. The first reaction is that these are different because the implication is an instance of a fact, and the equivalence does not hold between the facts, only between the instances of the facts. There are, however, good reasons why this is more than a syntactic distinction [Poole, 1990].

There seems to be a qualitative difference between the facts "emus are birds" and "tweety is an emu". The first is a fact that we would not like to consider being false (we would not consider the question "what if emus were not birds"), the second is one we may consider being false (e.g., we could conceive of the situation where Tweety was a sparrow).

This indicates that we should partition the facts into *background facts* and the *contingent facts* [Poole, 1985, Delgrande, 1988, Geffner, 1988]. This distinction is similar to the distinction between the network and markers in marker passing systems such as NETL [Fahlman, 1979], to the difference between the probabilistic knowledge (such as $p(A|B) = 0.345$) and the conditioning knowledge (the B in the preceding equation) in probability theory [Pearl, 1988], and to the difference between background knowledge and observations in abduction [Popl, 1973, Poole, 1989b].

1.4 Conditioning and Contexts

The final piece of the jigsaw is the notion of conditioning. If all we know about Tweety is that Tweety is an emu (given that we do not also have "emus do fly"), there is a very strong tendency to want to conclude that Tweety doesn't fly from the default "emus don't fly".

The intuition behind conditioning that will be used is that if "p's are q's" is a default and if we know $p(c)$, then all of the objections that could be raised about $q(c)$ that follow from $p(c)$ have already been taken into account when building the knowledge base. We only consider arguments against the conclusion $q(c)$ that do not already follow from $p(c)$.

This conditioning is accomplished by associating a *context* with each default, in which we know the default is applicable. Arguments against a default can be ignored if they are also arguments against the default given only the context of the default.

The notion of "context" is used, rather than, for example making a default into a pair, for a number of reasons. The first is the reluctance to invent any new connectives; part of the Theorist research is to see how far we can get inventing as little as possible. It may be the case that the appropriate context for a default is not the same as the precondition for the default (see section 5). Contexts seem to reflect a natural intuition.

The use of contexts is similar to an automatic prioritisation or cancelling of defaults, but, we will see that it has considerable advantages. One important advantage is that the sort of knowledge required to build the knowledge base is local, and so the knowledge base should be able to be built incrementally.

2 Formal Framework

2.1 Theorist Framework

Theorist is a simple framework for hypothetical reasoning.

We assume we are given a standard first order language over a countable alphabet. By a formula we mean a well formed formula in this language. By an instance of a formula we mean a substitution of terms in this language for free variables in the formula. In this paper the Prolog convention of variables starting with an upper case letter is used.

The basic definitions of Theorist are in terms of a set of closed formulae A (given as true) and a set of (possibly open) formulae H (the "possible hypotheses"). A **scenario** of (A, H) is a set D of ground instances of elements of H such that $D \cup A$ is consistent. If g is a closed formula, an **explanation** of g from (A, H)

is a scenario of (A, H) which, together with A, implies g. An **extension** of (A, H) is the set of logical consequences of A together with a maximal (with respect to set inclusion) scenario of (A, H).

In [Poole, 1988] it was shown how to avoid having complex formulae as defaults by "naming" complicated defaults (similar to the use of abnormality[McCarthy, 1986]), using the name as the default and have the name implying the formula as a fact. This is done in the examples in this paper.

2.2 Membership in all extensions

It can be argued [Poole, 1989b] that predicting what is in all extensions (i.e., can be explained even if an adversary chooses the defaults) provides more satisfactory results than, for example, predicting what is in one extension. Etherington [1988] has shown that this notion of prediction corresponds (propositionally at least) to circumscription [McCarthy, 1986].

The following theorem was proved in [Poole, 1989b, theorem 2.6]:

Theorem 2.1 *g is in every extension of (A, H) if and only if there is a set $\mathcal{E}$ of (finite) explanations of g from (A, H) such that there is no scenario S of (A, H) inconsistent with every element of $\mathcal{E}$.*

Theorem 2.1 leads to the following dialectical view of membership in every extension[1][Poole, 1989b].

There are two processes $\mathcal{Y}$ and $\mathcal{N}$ that are having an argument as to whether g should be predicted. Process $\mathcal{Y}$ tries to find explanations of g. Process $\mathcal{N}$ tries to find a scenario inconsistent with all of $\mathcal{Y}$'s explanations.

In general $\mathcal{Y}$ has a set of explanations Φ (initially Φ is empty). $\mathcal{N}$ tries to find a scenario S which is inconsistent with all members of Φ (i.e., explains the conjunction of the negation of the elements of Φ). When $\mathcal{N}$ finds such a scenario S, $\mathcal{Y}$ must find an explanation of g from (S, H). Whichever process, using a complete proof procedure, gives up first loses:

- If $\mathcal{Y}$ cannot come up with an explanation based on $\mathcal{N}$'s scenario S, then g is not in all extensions (in particular g is not in any extension of S).
- If $\mathcal{N}$ cannot come up with a scenario inconsistent with all of $\mathcal{Y}$'s arguments, every extension contains at least one of $\mathcal{Y}$'s arguments, and so g is in every extension.

[1]This algorithm corresponds to an abstract specification of algorithms for computing circumscription [Ginsberg, 1989, Przymusinski, 1989]. These algorithms find all of Y's arguments and then fail on N's counter arguments [Inoue and Helft, 1990].

One further refinement of theorem 2.1 can be easily proven. This corollary says that $\mathcal{N}$ only needs to choose one default from each of $\mathcal{Y}$'s explanations.

Corollary 2.2 g is in all extensions of (A, H) if and only if there is a set $\mathcal{E}$ of explanations of g from (A, H) such that there does not exist a counter argument. Scenario S of (A, H) is a counter argument if $\forall \phi \in \mathcal{E}\ \exists d \in \phi$ such that

1. $A \wedge d \models \neg S$.

The following example shows how restricted this notion of prediction is.

Example 2.3 Suppose we have the fact that emus are birds, and the defaults "birds fly", "emu's don't fly", and "if something looks like an emu, it is an emu".

This can be represented as[2]:

$$
\begin{aligned}
K = \{ \quad & \forall X\ emu(X) \Rightarrow bird(X), \\
& \forall X\ ll\_emu(X) \wedge mbe(X) \Rightarrow emu(X), \\
& \forall X\ bird(X) \wedge bf(X) \Rightarrow flies(X), \\
& \forall X\ emu(X) \wedge enf(X) \Rightarrow \neg flies(X) \\
H = \quad & \{bf(X), \\
& enf(X), \\
& mbe(X)\}
\end{aligned}
$$

Using membership in all extensions as a basis for prediction, we do not predict $\neg flies(tweety)$ from

$$(K \cup \{emu(tweety)\}, H).$$

This is because of the counter argument $\{bf(tweety)\}$.

This seems like a peculiar objection to $enf(tweety)$ as it is a counter argument for any emu.

Similarly we do not predict $emu(tweety)$ from $(K \cup \{ll\_emu(tweety)\}, H)$, due to the counter argument

$$\{bf(tweety), enf(tweety)\}$$

which again, always holds whenever the default is applicable.

The objection to the conclusion of $\neg flies(tweety)$ from $(K \cup \{ll\_emu(tweety)\}, H)$, namely $\{bf(tweety)\}$, is also a peculiar objection.

The proposed "solutions" to such problems, namely using cancellation axioms [McCarthy, 1986, Poole, 1988] and providing global priorities [McCarthy, 1986], are unsatisfactory for a number of reasons (see section 6). In this paper an alternate solution is advanced.

[2]$ll\_emu(X)$ is intended to mean "X looks like an emu".

3 Forcing Conditioning

If we have "emus don't fly" as a default, we want it to be used if all we know about an object is that it is an emu. Although there may be counter arguments (e.g., because it is a bird, it flies), we have taken these into account when building the knowledge base. The idea is to ignore counter arguments to "emu's don't fly" that follow just from the object being an emu. We still take into account other arguments as to why the emu should fly.

We assume that we are given the following sets

K a set of closed formulae; the "background knowledge". The knowledge that we know is always true.

G a set of closed formulae; the given knowledge about the situation being considered.

H a set of open formulae; the "possible hypotheses".

We associate with each possible hypothesis a context. The intention is that given just the context associated with a hypothesis we know the hypothesis is applicable (if consistent). A counter argument can be ignored if it is a counter argument when given only the context of the default.

The *context* of possible hypothesis h, written $\mathcal{C}(h)$ is a formula with free variables amongst the free variables of h.

The basic idea that we exploit is that some counter-arguments will be over-ridden by specificity. If S is an argument against d, i,e,

$$K \wedge G \wedge d \models \neg S$$

then S can be ignored due to specificity if

$$K \wedge \mathcal{C}(d) \wedge d \models \neg S$$

Definition 3.1 We **predict**$_1$ g if there is a set $\mathcal{E}$ of explanations of g from $(K \cup G, H)$ such that there does not exist a counter argument. Scenario S of $(K \cup G, H)$ is a counter argument if $\forall \phi \in \mathcal{E}\ \exists d \in \phi$ such that

1. $K \wedge G \wedge d \models \neg S$ and
2. $K \wedge \mathcal{C}(d) \wedge d \not\models \neg S$.

Note that prediction in this definition is a strict superset of membership in all extensions. If some formula is in all extensions, then it is predicted.

Figure 1: A diagram of the knowledge in example 3.2. Thick lines are facts, thin lines are (named) defaults.

Example 3.2 Consider the following "knowledge" about birds (see figure 1):

$$\begin{aligned}
K = \{\ & \forall X\ emu(X) \Rightarrow bird(X) \\
& \forall X\ ostrich(X) \Rightarrow bird(X) \\
& \forall X\ \neg(emu(X) \wedge ostrich(X)) \\
& \forall X\ ll_emu(X) \wedge mbe(X) \Rightarrow emu(X) \\
& \forall X\ ll_ostrich(X) \wedge mbo(X) \Rightarrow ostrich(X) \\
& \forall X\ bird(X) \wedge bf(X) \Rightarrow flies(X) \\
& \forall X\ in_cage(X) \wedge llb(X) \Rightarrow bird(X) \\
& \forall X\ emu(X) \wedge enf(X) \Rightarrow \neg flies(X) \\
& \forall X\ ostrich(X) \wedge onf(X) \Rightarrow \neg flies(X) \\
& \forall X\ flies(X) \wedge nf(X) \Rightarrow in_air(X)\} \\
H = \{\ & bf(X), enf(X), onf(X), mbe(X), mbo(X), \\
& llb(X), nf(X)\}
\end{aligned}$$

The context information can be represented as

$$\begin{aligned}
\mathcal{C}(bf(X)) &= bird(X) \\
\mathcal{C}(enf(X)) &= emu(X) \\
\mathcal{C}(onf(X)) &= ostrich(X) \\
\mathcal{C}(mbe(X)) &= ll_emu(X) \\
\mathcal{C}(mbo(X)) &= ll_ostrich(X) \\
\mathcal{C}(llb(X)) &= in_bird_cage(X) \\
\mathcal{C}(nf(X)) &= flies(X)
\end{aligned}$$

Example 3.3 Suppose we are given

$$G = \{\ emu(tweety),$$

$$in\_bird\_cage(tweety),$$
$$bird(polly),$$
$$in\_bird\_cage(polly)\}$$

We predict$_1$ that tweety does not fly, as there is an explanation of $\neg flies(tweety)$, namely $\{enf(tweety)\}$. The only potential counter argument (i.e., explanation of $\neg enf(tweety)$) is $\{bf(tweety)\}$. This explanation is ignored due to specificity as

$$K \wedge enf(tweety) \wedge \mathcal{C}(enf(tweety)) \models \neg bf(tweety)$$

We do not predict$_1$ that tweety flies. There is an explanation of $flies(tweety)$, namely $\{bf(tweety)\}$, however the explanation of $\neg bf(tweety)$, $(\{enf(tweety)\})$ is not an explanation of $\neg bf(tweety)$ from the context of $bf(tweety)$, namely $bird(tweety)$.

The knowledge $in\_bird\_cage(tweety)$ provided no evidence for the flying ability of Tweety. It could be safely ignored as it was irrelevant to the conclusion.

We predict that Polly flies, as there is an argument for the flying of Polly, and no reason to doubt that argument. There is no argument for Polly not flying.

Example 3.4 Consider how example 2.3 is handled. Suppose we have the knowledge base of example 3.2, and are given

$$G = \{ll\_emu(tweety)\}.$$

There is an explanation for $emu(tweety)$, namely $mbe(tweety)$. There is one counter-argument for this, namely

$$\{bf(tweety), enf(tweety)\}$$

however, this argument follows from $\mathcal{C}(mbe(tweety))$, and so can be ignored. Thus we predict $emu(tweety)$.

We predict$_1$ $\neg flies(tweety)$. There is an explanation, namely

$$\{mbe(tweety), enf(tweety)\}$$

The same counter argument exists for $mbe(tweety)$, and can be ignored for the same reason as above. There is one explanation of $\neg enf(tweety)$, namely

$$\{mbe(tweety), bf(tweety)\}$$

This can be ignored as $bf(tweety)$ is an argument against $enf(tweety)$ given $\mathcal{C}(enf(tweety))$.

We do not predict$_1$ $flies(tweety)$. There is an explanation for $flies(tweety)$, namely

$$\{mbe(tweety), bf(tweety).\}$$

There is a counter argument (an explanation of $\neg bf(tweety)$):

$$\{mbe(tweety), enf(tweety)\}$$

which cannot be ignored as it does not follow from $\mathcal{C}(bf(tweety))$.

Example 3.5 Suppose we have the knowledge base of example 3.2 and are given that Tweety looks like an emu and also looks like an ostrich (they do look similar):

$$G = \{ll\_emu(tweety) \wedge ll\_ostrich(tweety)\}$$

There are two explanations of $bird(tweety)$, namely:

$$\{mbe(tweety)\}$$
$$\{mbo(tweety)\}.$$

There is a counter argument to each of these explanations (they are, in fact, counter arguments to each other), but there is only one potential counter argument to both explanations, namely

$$\{bf(tweety), enf(tweety), onf(tweety)\}$$

This, however is also a counter in the contexts of each default and so can be ignored.

We thus predict $bird(tweety)$. We also predict $\neg flies(tweety)$.

Example 3.6 As an interesting variation to the previous example, suppose we are given that Tweety either looks like an emu or looks like an ostrich:

$$G = \{ll\_emu(tweety) \vee ll\_ostrich(tweety)\}$$

There is one explanation of $bird(tweety)$, namely:

$$\{mbe(tweety), mbo(tweety)\}.$$

There are potential counter arguments to this explanation, namely

$$\{bf(tweety), enf(tweety)\}$$
$$\{bf(tweety), onf(tweety)\}$$

These, however can be ignored due to specificity. We thus predict $bird(tweety)$. We also predict $\neg flies(tweety)$.

These examples show the robustness of the definition of specificity.

It is interesting to consider how this definition handles the qualitative lottery paradox [Poole, 1989a] that is problematic for many systems. In [Poole, 1989a] it was shown that there is a conflict between the "one step default property" (conditioning in this paper) and conjunctive closure. It was argued that conjunctive closure was the less intuitive property.

Example 3.7 The general form of the qualitative lottery paradox given in [Poole, 1989a] can be expressed as:

$$\begin{aligned} K &= \{ \quad \forall X\ b(X) \wedge d_i(X) \Rightarrow c_i(X), \text{ for } i = 1..n, \\ &\qquad \forall X\ \neg(c_1(X) \wedge \ldots \wedge c_n(X))\} \\ H &= \{ \quad d_i(X), \text{ for } i = 1..n\} \end{aligned}$$

$$\mathcal{C}(d_i(X)) \quad = \quad b(X), \text{ for } i = 1..n$$

Given $b(t)$, we can predict$_1$ $d_i(t)$ (and so $c_i(t)$) for any i. We predict the conjunctions of these conclusions while they are consistent. For example, we predict$_1$

$$c_1(t) \wedge \ldots \wedge c_{j-1}(t) \wedge c_{j+1}(t) \wedge \ldots \wedge c_n(t)$$

for each j. The reason is that the only argument against each $d_i(t)$ is

$$\{d_1(t), \ldots, d_{i-1}(t), d_{i+1}(t), \ldots, d_n(t)$$

and this is an argument against d_i given only the context of the default.

We do not however predict the conjunction of all of the $c_i(t)$, as this is inconsistent and so cannot even be explained.

4 Refinement of Conditioning

Example 4.1 Consider the following facts and defaults:

$$\begin{aligned} K &= \{ \quad uni\_student(X) \wedge usa(X) \Rightarrow adult(X), \\ &\qquad uni\_student(X) \wedge usne(X) \Rightarrow \neg employed(X), \\ &\qquad adult(X) \wedge ae(X) \Rightarrow employed(X)\} \\ H &= \{ \quad usa(X), usne(X), ae(X)\} \end{aligned}$$

$$\begin{aligned} \mathcal{C}(usa(X)) &= uni\_student(X) \\ \mathcal{C}(usne(X)) &= uni\_student(X) \\ \mathcal{C}(ae(X)) &= adult(X) \end{aligned}$$

Using the previous definition of prediction, given $uni\_student(fred)$, we predict

$$adult(fred) \wedge \neg employed(fred)$$

However, given

$$uni\_student(fred) \wedge adult(fred)$$

we do not predict $\neg employed(fred)$. The counter argument, $ae(fred)$ cannot be ignored. While we cannot *prove* $\neg ae(fred)$ from any default and its context, we can *predict* $\neg ae(fred)$ from the context of either default.

Example 4.2 Suppose we are given the background knowledge of example 3.2, and the contingent knowledge,

$$G = ll\_emu(tweety) \wedge \neg in\_air(tweety)$$

There is an explanation of $emu(tweety)$, namely by assuming

$$\{mbe(tweety)\}$$

There is an explanation of $\neg emu(tweety)$, by assuming

$$\{bf(tweety), nf(tweety)\}$$

The negation of this counter argument is not *proven* from the context of any default and that default, but $\neg bf(tweety)$ is *predicted* from the context of $mbe(tweety)$.

This leads us to the next definition of prediction which allows us to predict even more. The idea is to extend the definition so that a counter argument needs to just predict the negation of the defaults. This is defined recursively to ensure that the definition is well-grounded.

Definition 4.3 We **predict**$_i$ g given G if there is a set $\mathcal{E}$ of explanations of g from $(K \wedge G, H)$ such that there does not exist a counter argument. Scenario S of $(K \wedge G, H)$ is a counter argument if $\forall \phi \in \mathcal{E}\ \exists d \in \phi$ such that

1. $K \wedge G \wedge d \models \neg S$ and
2. we do not predict$_{i-1}$ $\neg S$ given $\mathcal{C}(d) \wedge d$.

We **predict**$_0$ g given A if $K \wedge A \models g$.

Definition 4.4 We **predict** g given G if there is some i such that we predict$_i$ g given G.

In this definition predict$_1$ is the same as the previous definition; each higher integer allows us to predict more.

In example 4.1 we predict$_2$ $\neg employed(fred)$ given

$$uni\_student(fred) \wedge adult(fred)$$

In example 4.2 we predict$_2$ $emu(tweety)$ given

$$ll\_emu(tweety) \wedge \neg in\_air(tweety)$$

5 Pragmatics

Contexts are intended to be the cases under which we know the assumption is applicable. The "normal case" is where the default "p's are q's" is represented as the fact

$$\forall X\ p(X) \wedge d(X) \Rightarrow q(X)$$

with the default $d(X)$ and the context information

$$\mathcal{C}(d(X)) = p(X)$$

There is nothing in the theory to force this use of contexts. There are two extremes of contexts that are interesting. If $\mathcal{C}(d)$ is uniformly $false$, prediction becomes equivalent to membership in one extension (as all counter arguments are ignored). If $\mathcal{C}(d)$ is uniformly $true$, prediction is equivalent to membership in all extensions.

One pragmatic idea is that if "a's are c's" and "b's are not c's", then we have a conflict if we know something is both an a and a b. If we prefer the first default over the second, we want to say that the second default is not applicable if a is true. This can be done by:

$$\begin{aligned} K &= \{ \quad a \wedge d_1 \Rightarrow c \\ &\qquad\; b \wedge d_2 \Rightarrow \neg c \\ &\qquad\; a \Rightarrow \neg d_2\} \\ H &= \{ \quad d_1, d_2\} \\ \mathcal{C}(d_1) &= a \\ \mathcal{C}(d_2) &= b \end{aligned}$$

If we are given a, we predict c. If we are given b, we predict $\neg c$. If we are given $a \wedge b$ we predict c, using assumption d_1. Notice that if we are given nothing, then we predict $\neg a$ (assuming d_2). This is reasonable as because $b's$ are not $c's$ we are implicitly assuming $\neg a$. If we are given a, we do not predict $\neg b$, which is again reasonable as we are not making any implicit assumptions about b (given a we predict c whether or not b is true).

This may be a simplistic way to handle causal reasoning (see [Geffner, 1989] for a more sophisticated theory), but is good enough, with specificity, to handle some tricky examples:

Example 5.1 (Geffner, 1989) Suppose we get up in the morning and find that we have left the lights on in the car and want to determine whether the car will start. We are given that the car normally starts if we turn the key, and normally does not start if the battery was flat (even if we turn the key), and that the battery is flat, by default if the lights were on. Following the above methodology, this can be stated as

$$\begin{aligned} K &= \{ \quad turn\_key \wedge key\_starts \Rightarrow starts \\ &\qquad\; batt\_flat \wedge batt\_prevents \Rightarrow \neg starts \\ &\qquad\; batt\_flat \Rightarrow \neg key\_starts \\ &\qquad\; lights\_were\_on \wedge drained \Rightarrow batt\_flat\} \\ H &= \{ \quad key\_starts, batt\_prevents, drained\} \\ \mathcal{C}(key\_starts) &= turn\_key \\ \mathcal{C}(batt\_prevents) &= batt\_flat \\ \mathcal{C}(drained) &= lights\_were\_on \end{aligned}$$

If we are given just $turn\_key$, we predict

$$starts \wedge \neg batt\_flat \wedge \neg lights\_were\_on$$

as there is only one extension and no counter arguments.

If we were given

$$turn\_key \wedge lights\_were\_on$$

we predict $batt\_flat$ and $\neg starts$. The only potential counter argument to $drained$ is $\{key\_starts\}$ which can be ignored due to specificity; we thus derive $batt\_flat$. We also predict $\neg s$, using explanation

$$drained, batt\_prevents$$

The only counter arguments contain key_starts which is ignored by specificity.

Example 5.2 (Hanks and McDermott, 1986) Consider the celebrated "Yale Shooting Problem"; we follow the methodology given above:

$$\begin{aligned} K &= \{ \quad \forall T\ loaded(T) \wedge lp(A,T) \Rightarrow loaded(do(A,T)), \\ &\qquad\; \forall T\ alive(T) \wedge ap(A,T) \Rightarrow alive(do(A,T)), \\ &\qquad\; \forall loaded(T) \Rightarrow \neg alive(do(shoot,T)), \\ &\qquad\; \forall loaded(T) \Rightarrow \neg ap(shoot,T)\} \\ H &= \{ \quad lp(A,T), ap(A,T)\} \\ \mathcal{C}(lp(A,T)) &= loaded(T) \\ \mathcal{C}(ap(A,T)) &= alive(T) \\ G &= loaded(0) \wedge alive(0) \end{aligned}$$

The only thing "tricky" thing here is to cancel the persistence of alive when we shoot with the gun loaded.

We can explain $\neg alive(do(shoot, do(wait, 0)))$, with $\{lp(wait, 0)\}$. The only counter arguments to $lp(wait, 0)$ is $\{ap(shoot, do(wait, 0))\}$, which can be ignored due to specificity as its negation follows from $lp(wait, 0) \wedge \mathcal{C}(lp(wait, 0))$. We do not predict $alive(do(shoot, do(wait, 0)))$, as there is a valid counter argument to the assumption $ap(shoot, do(wait, 0))$.

While the above methodology works for these examples, there are some problems for which it does not work (see [Geffner, 1989]). There is a strong feeling that the main problems in default reasoning have to do with properly characterising specificity and causation.

6 Comparison with other systems

One of the main goals of this research is to draw a bridge between those systems that treat defaults as statements of conditionals [Geffner, 1988, Delgrande, 1988], and those that treat defaults as propositional assumptions [McCarthy, 1986, Poole, 1988]. The former have nice properties with respect to specificity, but need a form of irrelevance to allow chaining and ignoring irrelevant properties. The latter ignore irrelevant details and allow chaining, but do not handle specificity well. This paper is an attempt to consider what needs to be added to the assumption based systems to allow the natural specification of specificity. The solution to the problems of specificity is also much more natural than the solution of using global priorities, particularly as no one is prepared to say where such global priorities come from or what they mean. This sort of conditioning knowledge seems like the sort of knowledge one would have about a default.

The most interesting comparison of this work is with the addition of irrelevance to ϵ-semantics. The definition of ignoring in predict$_1$ is almost identical to the definition of irrelevance in [Geffner, 1988]. Both of theses systems fail for example 4.2, and the ignoring for the general definition of prediction in this paper is almost identical to the irrelevance of [Geffner and Pearl, 1989]. The resulting systems are, however, different. For example, because we are using normal logical connectives, we can use the contrapositive of defaults. The two systems get the same result on Geffner's examples (for example the "solution" to the Yale shooting problem in example 5.2 follows a similar idea to the solution presented in [Geffner, 1988]). It seems as though there is something important about the irrelevance that is independent of the underlying probability theory.

The use of conditioning can be motivated in a similar manner to the notion of "all I know" of Levesque [Levesque, 1990]. They are, however very different. Levesque makes no distinction between background and contingent knowledge. If someone just tells us that "Tweety is an emu" we can use that as our contingent knowledge and say that this is all we know (contingently) about Tweety. As part of what Levesque "only knows" about Tweety includes all tautologies about Tweety, instances of general information (such as "$square(tweety) \Rightarrow rectangle(tweety)$") and derived information (such as $bird(tweety)$). Levesque makes no attempt to automatically use specificity.

This work should also be contrasted to the work in inheritance systems [Touretzky, 1986, Thomason and Horty, 1988, Stein, 1989]. We are trying to add a notion of specificity to a general logic system, and want the non-defeasible statement "emus are birds" to be exactly the logical statement $\forall X\ emu(X) \Rightarrow bird(X)$. This work is most closely related to the sceptical inheritance of [Stein, 1989]; both allow for membership in all extensions with a notion of specificity. This work allows for a much more expressive language than the networks used for the inheritance theory.

This work has many similarities and differences to [Poole, 1985]. In that work the important context was the context of the more general default, whereas, in this paper the important context is the one of the more specific default. The main problem with that paper was in the underlying reasoning paradigm in which the specificity was added; this problem has recently been addressed [Simari and Loui, 1990]. In [Poole, 1985], the user was not required to specify the context of the defaults, as they are in the system described in this paper. It seems to be an advantage rather than a disadvantage to be able to specify a context in which a default is known to be applicable. As shown in the previous section, this extra pragmatic knowledge can be used to advantage in many cases.

7 Conclusion

In this paper we analysed some problems that arise from prediction based on membership in all extensions. This problem was diagnosed as being due to peculiar counter arguments. A solution was proposed that is based on a very simple idea of conditioning. This is particularly nice, as the conditioning knowledge required is local to a default, and seems to be very natural (as opposed to other solutions based on cancellation or global priorities).

Acknowledgements

Thanks to Hector Geffner and Andrew Csinger for valuable discussions on the topic of this paper. This research was supported under NSERC grant OP-POO44121.

References

[Delgrande, 1988] J. P. Delgrande, "An approach to default reasoning based on first-order conditional logic: revised report", *Artificial Intelligence*, 36(1) 63-90.

[Etherington, 1988] D. Etherington, *Reasoning with Incomplete Information*, Pitman, Morgan Kaufmann.

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

[Geffner, 1988] H. Geffner, "On the Logic of Defaults", *Proc. AAAI-88*, 449-454.

[Geffner, 1989] H. Geffner, *Default Reasoning: Causal and Conditioning Theories*, Ph.D. thesis, Computer Science, UCLA.

[Geffner and Pearl, 1989] H. Geffner and J. Pearl, "A Framework to reason with Defaults", to appear *Defeasible Reasoning and Knowledge Representation*, Kluwer Publisher.

[Ginsberg, 1989] M. Ginsberg, "A circumscriptive theorem prover", *Artificial Intelligence*, 39 209-230.

[Hanks and McDermott, 1986] S. Hanks and D. McDermott, "Default reasoning, non-monotonic logics, and the frame problem", *Proc. AAAI-86*, 328-333.

[Inoue and Helft, 1990] K. Inoue and N. Helft, *Theorem Provers for Circumscription*, *Proc. CSCSI-90.*

[Levesque, 1990] H. Levesque, "All I Know: A Study in Autoepistemic Logic", to appear *Artificial Intelligence.*

[Loui, 1987] R. P. Loui, "Defeat among arguments: a system of defeasible inference", *Computational Intelligence*, 3(2) 100-106.

[Loui, 1990] R. P. Loui, "Ampliative Inference, Computation and Dialectic", in J. Pollock and R. Cummins (Eds.) *AI and Philosophy*, M.I.T. Press.

[McCarthy, 1986] J. McCarthy, "Applications of Circumscription to Formalising Common Sense Knowledge", *Artificial Intelligence*, 28(1) 89-116.

[Neufeld and Poole, 1988] E. M. Neufeld and D. L. Poole, "Probabilistic Semantics and Defaults", *Proceedings of the Fourth Workshop Uncertainty in Artificial Intelligence*, University of Minnesota, 275-282.

[Pearl, 1988] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.

[Poole, 1985] D. L. Poole, "On the Comparison of Theories: Preferring the Most Specific Explanation", *Proc. IJCAI-85*, 144-147.

[Poole, 1988] D. L. Poole, "A Logical Framework for Default Reasoning", *Artificial Intelligence*, 36(1), 27-47.

[Poole, 1989a] D. Poole, "What the lottery paradox tells us about default reasoning", *Proceedings of the First International Conference on the Principles of Knowledge Representation and Reasoning*, Toronto, 333-340.

[Poole, 1989b] D. Poole, "Explanation and Prediction: An Architecture for Default and Abductive Reasoning", *Computational Intelligence*, 5(2) 97-110.

[Poole, 1990] D. Poole, "The effect of knowledge on belief: conditioning, specificity and the lottery paradox in default reasoning", Technical Report, Computer Science, University of British Columbia.

[Popl, 1973] H. Popl, "On the mechanisation of Abductive Logic", *Proc. IJCAI-73*, 147-152.

[Przymusinski, 1989] T. C. Przymusinski, "An algorithm to compute circumscription", *Artificial Intelligence*, 38(1) 49-73.

[Simari and Loui, 1990] G. R. Simari and R. P. Loui, "Confluence of argument systems: Poole's rules revisited", to appear, *3rd Workshop on Nonmonotonic Reasoning*, Lake Tahoe, June 1990.

[Stein, 1989] L. A. Stein, "Skeptical Inheritance: Computing the Intersection of Credulous Extensions", *Proc. IJCAI-89*, 1153-1158.

[Thomason and Horty, 1988] R. H. Thomason and J. F. Horty, "Logics for Inheritance Theory", *Proc. Second International Workshop on Non-Monotonic Reasoning.*

[Touretzky, 1986] D. S. Touretzky, *The Mathematics of Inheritance Systems*, Morgan Kaufmann.