

Why Nonmonotonic Logic? ⁺

Steven O. KIMBROUGH *

and Fred ADAMS **

* Department of Decision Sciences, University of Pennsylvania,
Philadelphia, PA 19104, USA

** Department of Philosophy, Central Michigan University,
Mount Pleasant, MI 48859, USA

The motivation for nonmonotonic logic is to produce a machine representation for default reasoning, broadly construed. In this paper we argue that all nonmonotonic logics have (by definition) inference rules that fail to preserve truth, and this fact leads to several undesirable features. In response to these problems, but recognizing the importance of the original motivation for nonmonotonic logic, we propose an alternative to nonmonotonic logic, which achieves nonmonotonicity of reasoning without abandoning in any way truth preserving inferences. This approach is based on a possible worlds framework, which we illustrate with a small Prolog program. Motivating this approach is an important distinction, which we believe the advocates of nonmonotonic logic to be ignoring: that between inferencing and making decisions, or equivalently that between inferencing and theory construction.

Keywords: Logic, Logic Programming, Nonmonotonic Logic, Default Reasoning, Possible Worlds, Decision Support Systems.



Madison.

Steven O. Kimbrough is Assistant Professor in the Department of Decision Sciences, The Wharton School, University of Pennsylvania. His main research activities are in logic modeling – focusing on belief revision and automatic theorem proving – and in knowledge-based decision support systems. He is principal investigator for the Knowledge-Based Support Systems project with the U.S. Coast Guard. His Ph.D. and M.S. degrees are from the University of Wisconsin-

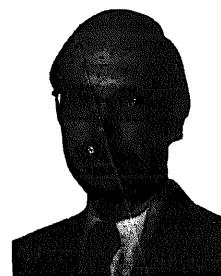
⁺ Thanks to Ronald M. Lee for reading an earlier draft and for pointing out the connection between the possible worlds framework and scenario management in DSS. Thanks also to Eric K. Clemons for reading an earlier draft and for worrying about the difference between possible worlds and context switching. This research was supported in part by the Atlantic Richfield Foundation and in part by the U.S. Coast Guard. An earlier version of this paper was presented at the 1987 Hawaii International Conference on Systems Sciences. That version may be found in the conference's proceedings.

1. Introduction

The exigencies of practical affairs require us to leap to conclusions, to go beyond the evidence, to make assumptions. In doing so, we often get things wrong, hence we need to leap back from our conclusions, undo our assumptions, revise our beliefs. This sort of reasoning is called default reasoning because in the absence of needed information we invoke defaults in order to get on with our business. It is also called defeasible reasoning because we make assumptions which may later be defeated by contravening evidence. (We shall be taking 'default reasoning' and 'defeasible reasoning' in a broad sense, not tied to any particular theory of how it is done).

The literature's standard example,¹ which we shall continue to use, is whether Tweety flies. We know, in the example, that Tweety is a bird and that all birds fly, except penguins and ostriches. Does Tweety fly? We might, if pressed by the need to act, assume that he does. We might then discover that Tweety is a penguin, at which time we need to retract our conclusion that Tweety flies.

There is no question that the Tweety example accurately captures the essentials of default reasoning. Any proposed machine representation of default reasoning should be able to handle the



Fred Adams is Assistant Professor of Philosophy at the School of Arts and Sciences at Central Michigan University. He received a B.A. in Anthropology and M.A. in Philosophy from Southern Illinois University at Edwardsville. His Ph.D. is in Philosophy from the University of Wisconsin in 1982. His research interests lie in epistemology, cognitive science, and action theory.

¹ We are referring to the artificial intelligence and management science literature. Defeasible reasoning has an extensive and older literature in philosophy, particularly in ethics. Moral imperatives – that, for example, one should not kill other people – are often thought to be defeasible by other conditions – for example, that killing is morally permitted in self defense. The philosophical literature, with few exceptions [e.g., Harman (1986)], has been little concerned with the machine implementation of defeasible reasoning.

Tweety story. How, then, should machine representation of default reasoning work? The main response to this question has been to supply some form of nonmonotonic logic [American Association for Artificial Intelligence (1984), McDermott and Doyle (1980), Reiter (1980), McCarthy (1980) Reiter (1986)]. In standard (monotonic logic),² for all statements $\gamma_1 \dots \gamma_n, \phi, \psi$ if

$$\gamma_1, \dots, \gamma_n \vdash \phi, \quad (1)$$

then, for any ψ ,

$$\gamma_1, \dots, \gamma_n, \psi \vdash \phi. \quad (2)$$

(In an equivalent, but shorter, notation we let Γ represent a collection of statements, $\gamma_1 \dots \gamma_n$, and say, e.g., that in standard (monotonic) logic, if

$$\Gamma \vdash \phi, \quad (3)$$

then, for any ψ ,

$$\Gamma, \psi \vdash \phi, \quad (4)$$

and similarly in other cases.)

A nonmonotonic logic may be defined as any logic with the following property: For some Γ, ϕ , and ψ

$$\Gamma \vdash_{NML} \phi, \quad (5)$$

but

$$\Gamma, \psi \not\vdash_{NML} \phi. \quad (6)$$

We shall call this a weak nonmonotonic logic.³ In a strong nonmonotonic logic, we should have, again for some Γ, ϕ, ψ :

$$\Gamma, \psi \vdash_{NML} \neg\phi. \quad (7)$$

The nonmonotonic approach, regardless of the particular implementation, can be criticized on a number of fronts, as we discuss in §2, thus exploring other approaches to this problem of accounting for default reasoning would be warranted (also see Kimbrough and Adams (1986), Hanks and McDermott (1986, 1987)). In what follows, we

discuss two different ways of computerizing default reasoning, neither of which is open to the problems of nonmonotonic logics. Our first method (in §3) works, but is not, we think, fundamental. It may be useful in certain applications and, more importantly, it serves to introduce a fundamental distinction between deciding and inferring. The second approach in (§4) also works and is, we think, both fundamental and fundamentally correct. It uses a possible worlds framework, which has of late been employed very successfully to deal with related problems in logic and philosophy (e.g., Lewis (1973), Lewis (1986), Stalnaker (1984)). Here, we discuss the framework briefly and motivate it. Then we present a very simple working Prolog program, which supports default reasoning on the Tweety story and which uses the possible worlds approach. The program, however, is very far from being a general system for default reasoning, and is meant only to illustrate certain aspects of our approach, which we discuss in further detail.

2. Problems with Nonmonotonic Logics

Nonmonotonic logics (as opposed to nonmonotonic reasoning) have received considerable and, we think, very effective criticisms (see Kimbrough and Adams (1986), Hanks and McDermott (1986, 1987), Nute (1986) for other problems with nonmonotonic logics). The fundamental problem, we believe, is that by definition a nonmonotonic logic must employ rules of inference, ' \vdash_{NML} ', that fail to preserve truth and this will inevitably lead to difficulties. (A rule of inference fails to preserve truth if it permits a conclusion to be drawn that might be false when the available premises are true. Recalling the definitions of nonmonotonic logic, ϕ in (5) and (6) is such a conclusion). Our purpose here is both to criticize nonmonotonic logic, and to present and explore some alternatives to it. For the remainder of this section we limit ourselves to the former task.

2.1. Wimpy Modus Ponens

Modus ponens is a rule of inference sanctioned by common sense and standard logics, including the familiar version of sentence logic and predi-

² Hacking (1979) goes so far as to say that the nonmonotonicity property, stated slightly more generally in his formulation, is one of four defining conditions for deductive logic: reflexivity ($\alpha \vdash \alpha$), transitivity (If $\alpha \vdash \beta$ and $\beta \vdash \gamma$, then $\alpha \vdash \gamma$), and or-introduction (If $\gamma \vdash \Phi$ then $\Gamma \vdash \psi \vee \Phi$).

³ Note also that the turnstile, ' \vdash ', and the nonmonotonic logic turnstile, ' \vdash_{NML} ', are proof-theoretic symbols. The statements characterizing nonmonotonic logics could be reformulated equivalently with the double turnstile, ' \models ' or ' \vDash_{NML} ', which are model-theoretic symbols.

cate logic. The rule may be expressed as follows:
Given $\phi \rightarrow \psi$ and ϕ , infer ψ .

What we shall call the rule of wimpy modus ponens (WMP) may be expressed as follows:

Given $\phi \rightarrow \psi$ and ϕ , do not infer that ψ .

WMP can be thought of as articulating a very conservative approach to life. Not so with perverse modus ponens (PMP), which may be expressed as follows:

Given $\phi \rightarrow \psi$ and ϕ , infer $\neg\psi$.

Now, on very mild assumptions⁴ it is a simple matter to see that all weak nonmonotonic logics support WMP. To see this, recall the definition of a weak nonmonotonic logic, in §1. Assume we have, for some particular Γ and ϕ :

$$\Gamma \vdash_{NML} \phi. \quad (8)$$

But, in sentence logic (and any system containing it), if

$$\Gamma \vdash \phi, \quad (9)$$

then

$$\Gamma \vdash (\phi \rightarrow \phi), \quad (10)$$

for any well formed formula, ϕ , whatsoever. In particular, it is true of ψ (see the above definition of a weak nonmonotonic logic, (5) and (6)). So, since nonmonotonic logics are (here) assumed to support all inferences sanctioned by standard logic, it follows that all weak nonmonotonic logics support the WMP inference rule. Using the same line of reasoning, it is clear that all strong nonmonotonic logics support the PMP inference rule.

2.2. The Lack of a Deduction Theorem

The deduction theorem in standard (monotonic) logic states that for all collections of statements, Γ , and for every statement, ϕ ,

If $\Gamma \vdash \phi$, then $\vdash (\Gamma \rightarrow \phi)$

(cf. Kleene (1967)).⁵ In other words, the deduction theorem makes the seemingly obvious statement that if you can derive ϕ from a collection of axioms, Γ , then Γ materially implies ϕ ; further that $(\Gamma \rightarrow \phi)$ is a logical truth. Our purpose in this

⁴ Specifically, that any inference sanctioned by \vdash or by \vDash is also sanctioned by the nonmonotonic analogs.

⁵ This is shorthand for: If $A_1, \dots, A_n \vdash \phi$ then $\vdash ((A_1, \dots, A_n) \rightarrow \phi)$.

subsection is to prove that any nonmonotonic logic in which the deduction theorem holds is inconsistent.

We will carry out the proof in sentence logic. Our presumption is that nonmonotonic logics are extensions of classical (monotonic) logic. In particular, we assume that these extensions add axioms or rules of inference (or transformation) to standard sentence logic, but do not add to the class of well formed formulas in standard sentence logic.⁶ We will also use a particular formulation of sentence logic, although this is not essential. The formulation we will use is called PM, for Principia Mathematica. PM is the formulation of sentence logic made by Russell and Whitehead, but modified to eliminate a redundant axiom (cf. Hughes and Cresswell (1967)). PM has four axioms and two transformation rules, as follows:

1. $((P \vee P) \rightarrow P)$,
2. $(Q \rightarrow (P \vee Q))$,
3. $((P \vee Q) \rightarrow (Q \vee P))$,
4. $((Q \rightarrow R) \rightarrow ((P \vee Q) \rightarrow (P \vee R)))$.

TR1. The result of uniformly replacing any variable in a thesis by any well-formed formula is itself a thesis. (A thesis is either an axiom or a theorem. This is called the rule of uniform substitution).

TR2. If ϕ and $(\phi \rightarrow \psi)$ are theses, so is ψ . (This is called the rule of modus ponens and may also be stated as: If $\Gamma \vdash \phi$ and $\Gamma \vdash (\phi \rightarrow \psi)$, then $\Gamma \vdash \psi$.)

In appendix A, we give the proof that any nonmonotonic logic, in which the deduction theorem holds, is inconsistent. Here we shall discuss an example, for the purpose of illustrating the import of the theorem.

A main way of creating a nonmonotonic logic has been to introduce rules of inference, called default rules, into a system of standard logic (e.g., Reiter (1980), Genesereth and Nilsson (1987, p. 152)). These default rules are commonly written in the form

$$\frac{\alpha(x) : \beta(x)}{\gamma(x)}, \quad (11)$$

where $\alpha(x)$, $\beta(x)$, and $\gamma(x)$ stand for formulas.⁷

⁶ If axioms or rules of inference are also removed from sentence logic by a nonmonotonic logic, then a large class of new problems arises, but discussing these would take us too far afield.

⁷ An equivalent expression for (11) is: $(\alpha(x) : \beta(x)) / \gamma(x)$.

For example, the following is a particular default rule in this form

$$\frac{\text{Bird}(x) : \text{Flies}(x)}{\text{Flies}(x)},$$

which is to be interpreted as ‘For all x , if x is a bird and that x flies is consistent with the assumptions (axioms) at hand, then infer that x flies’. More generally, a default rule schema (11, above) would be interpreted as ‘For all x , if $\alpha(x)$ and that $\beta(x)$ is consistent with the assumptions (axioms) at hand, then infer that $\gamma(x)$ ’. To rework an example presently in the literature [de Kleer (1986, p. 132)], suppose we were to add two default rules to the standard system, PM, discussed earlier:

$$\text{DR1: } :A/A,$$

$$\text{DR2: } :C/C.$$

Both DR1 and DR2 are unconditioned default rules, the $\alpha(x)$ condition in (11) being absent. DR1 is to be interpreted ‘For all x , if that A is consistent with the assumptions (axioms) at hand, then infer that A ’, and similarly for DR2. Given these rules, then we have the following sequent in nonmonotonic logic R .⁸

$$\begin{aligned} & (A \rightarrow B), (C \rightarrow D), \\ & ((A \wedge C) \rightarrow E) \vdash_R (A \wedge B \wedge C \wedge D), \end{aligned} \quad (12)$$

but not in PM proper. If, however, we add to the premises the statement $\neg(B \wedge D)$, then the conclusion of (12) is no longer derivable. Thus, we have a nonmonotonic logic. We shall now see that if this system includes the deduction theorem, then a contradiction can be deduced. From (12), it should be apparent that

$$(A \rightarrow B) \vdash_R B, \quad (13)$$

for by applying DR1 we get A as a thesis, and by applying TR2 (modus ponens) to A and $(A \rightarrow B)$ we get B as a thesis.⁹ Step 1 of our argument is, then, formula (13). Starting here the derivation of a contradiction is as follows.

1. $(A \rightarrow B) \vdash_R B$; DR1, TR2,
2. $\vdash_R ((A \rightarrow B) \rightarrow B)$; 1, deduction theorem,
3. $\vdash_R ((\neg A \vee B) \rightarrow B)$; 2, PM,

⁸ Named for Reiter-type default logics. See Reiter (1980).

⁹ Inferences of this form (13) had a special name before they were rediscovered by artificial intelligence: affirming the consequent.

4. $\vdash_R (\neg(\neg A \vee B) \vee B)$; 3, PM,
5. $\vdash_R ((A \wedge \neg B) \vee B)$; 4, PM,
6. $\vdash_R ((A \vee B) \wedge (\neg B \vee B))$; 5, PM,
7. $\vdash_R (A \vee B)$; 6, PM,
8. $\vdash_R (A \vee \neg B)$; 7, PM (TR1),
9. $\vdash_R A$; 7, 8, PM (resolution).
10. $\vdash_R \neg A$; 9, PM (TR1).

This proof illustrates the more general proof is given in the appendix. (note that the expression in step 6 is in conjunctive normal form. See the appendix. Also, ‘PM’ indicates that the inference is sanctioned by the system PM, i.e., by standard propositional logic.)

2.3. The Inconsistency of Unconditional Default Theories

There is a third problem, afflicting not all nonmonotonic logics but a broad variety of them. In the example default system of §2.2, two particular default rules were used, DR1 and DR2, both of which are, as noted earlier, unconditional default rules. Rules of this sort are frequently encountered in default logics, especially in the form

$$\frac{: \beta(x)}{\gamma(x)}. \quad (14)$$

(See Genesereth and Nilsson (1987, p. 153).) Subject to our usual qualification of taking nonmonotonic logics as extensions of, as including, standard logic, all such logics with unconditional default rules are inconsistent. The proof is simple. Let (14) be the schema, or form, of such a rule and assume we also have PM.¹⁰ Let ‘ \vdash_{UD} ’ represent the turnstile for an arbitrary default theory with an unconditional default rule with the form of (14).

1. $\vdash_{UD} \gamma(x)$; (14)
2. $\vdash_{UD} \neg \gamma(x)$; 1, PM(TR1, $\neg \gamma(x)$ for $\gamma(x)$).

Recalling the example proof from §2.2, this result can be thought of as a corollary to the theorem showing the absence of the deduction theorem for nonmonotonic logic.

¹⁰ The only restriction on (14) is that it be a genuine default rule; the inference it permits is not valid in standard logic.

2.4. Whither Semantics?

So far, except for a footnote, we have discussed nonmonotonic logic from the point of view of proof theory. This is also the usual practice in the literature. In this section, however, we shall briefly discuss nonmonotonic logic from a semantical point of view, beginning with the introduction of some basic semantic concepts.

Consider a syntactic system, S . S might be, for example, the syntactic system underlying predicate logic, but here we are taking a much broader view; S may be any formal syntactic system at all. In order to work with S we shall want a way of interpreting it, of saying what an expression in S means. The standard way of doing this is to introduce an interpretation function, I , that maps from expressions in S to the members of a set, M (the interpretation, or meaning, of S). Suppose M is the set $\{T, F\}$ and I maps at least one expression in S into M , then I is said to be a valuation function for S . A formal (logical) language, L , consists of a syntactic system, S , and a non-empty set of valuation functions for S , called the admissible valuations of L . We can then make the following definitions.

Definition. Given a formal language, L , a set of expressions, X , where $X \subseteq L$, and an admissible valuation function, v , then X is satisfied (by v on L) if and only if $v(A) = T$ for all $A \in X$.

Definition. The set X , such that $X \subseteq L$, semantically entails A (or, in symbols, $X \models A$) if and only if every admissible value function that satisfies X also satisfies A .

All this is standard stuff in formal semantics (and follows closely the treatment in (van Fraasen (1971, pp. 31–33)). The underlying idea is that the valuation function gives the interpretation or meaning to expressions in the language, and that an expression is valid if it is true under all interpretations. Given this approach and these definitions, we have the following result.

Theorem. Monotonicity of Semantic Entailment. If $X \subseteq Y$ and $X \models A$, then $Y \models A$. (See van Fraasen (1971, p. 32).)

The proof of this theorem is trivial. Every admissi-

ble valuation that satisfies the set Y must also satisfy every subset of Y , hence every such valuation satisfies X . But we are given that every admissible valuation that satisfies X also satisfies A . Thus, every admissible valuation that satisfies y also satisfies A , or $Y \models A$, in symbols.

This theorem, and the associated concepts and definitions, apply directly to standard, monotonic logic. They also have much to say about non-monotonic logic. Without some sort of coherent semantics (i.e., some coherent interpretation function) it is simply not clear what such a logic might mean (and if we do not know what it means we are in no position to say that it works in practice). But the import of this theorem in the present context is that if a logic is to be nonmonotonic, then either there is no valuation function available or the definitions of satisfiability and semantic entailment need to be revised and alternative concepts used. Although there is no way of showing that any new definitions must fail in providing an adequate semantics for nonmonotonic logic, it does seem that the second alternative is quite unpromising; the definitions make good intuitive sense and have been used extensively with success.

There remains the possibility of developing a nonmonotonic logic semantics based on an alternative valuation function. The first thing to note about this strategy is that even if it succeeds *none* of the expressions in a resulting nonmonotonic language can be interpreted as either true or false, for since a valuation function maps at least one expression to either truth or falsity, an interpretation function that is not a valuation function can map no expression to either truth or falsity. The second thing to note is that the above theorem is proved for valuation functions that map to the set $\{T, F\}$. The fact that we interpret T as truth and F as falsity is immaterial for the theorem, which is really about interpretation functions that map to any set with two elements (or indeed more than two elements, since an interpretation function that maps at least one expression to some member of a set with n elements ($n > 2$) also maps that expression to some member of a set with 2 elements). The third thing to note is that normally we think of reasoning as aimed at finding statements that are true, or likely to be true, or are believable, or can be taken as an adequate basis for action. But given the first two points, no nonmonotonic semantics can interpret expressions

in any such manner. In short, even if a nonmonotonic semantics were to be had, it would not allow us to say the sorts of things we need to say in a logic, nonmonotonic or not.

These features of nonmonotonic logic – support for wimpy modus ponens, absence of the deduction theorem, and the inconsistency of default theories with unconditioned default rules, as well as the semantic difficulties and the problems discovered by others – are, if not devastating, at least unhappy and undesirable. We conclude that entirely different approaches to nonmonotonic reasoning ought to be investigated. Our discussion on that topic begins in §3.

3. Defaulting with Rules of Action

In general when we work with models, there is an important distinction between what the model says and what we do with the model, although there is a close relation between the two. To illustrate, if we have modeled a particular system with a mathematical program and if the model is solved successfully, then the model tells us what the optimum arrangement of decision variables is (on the assumption that the model is itself a correct representation of the system modeled). That, the optimum given the problem representation, is one thing and what we do with it is another. Of course, the beauty of optimization models is that they give us the best settings for the decision variables, so the decision rule (or rule of action) we use with them is simple, when we are convinced of the appropriateness of the representation: set the variables at what the (solved) model says is the optimum arrangement. In any event, there are these two quite distinct elements of the modeling process: what the model says is the case and what we do with that information.¹¹

The same point can be made with regard to logic (or inferential) models: there is a distinction between the conclusion of a valid (or even sound) argument and what it is we do with the conclusion (see Carroll (1895) for a similar point). For example, the argument ‘All humans are mortal. You are a human. Therefore, you are mortal’, is a sound one, but its soundness is perhaps less interesting than what it is we do in response to the conclu-

sion. Should we buy life insurance? Should we be happy or sad or neither? What difference should this fact make in our lives? It is these questions that are, in this case, the interesting ones. In effect, what is interesting is less the conclusion (because it seems incontrovertible) and more the rules of action appropriate for this conclusion.

This distinction, between the conclusion of an inferential system and the rules of action we apply to the conclusion, is an elementary, even trivial, one. We think it is a useful distinction, however, and one that has been forgotten in much of the discussion of nonmonotonic logic. In fact, keeping this distinction in mind will allow us to construct a device that obviates the need for nonmonotonic logic and allows us to remain within the familiar confines of standard (i.e., truth-preserving) logics. To see why this is so, let us rehearse yet again the story that normally concludes with a call for nonmonotonic logic.

Again, the standard story goes as follows. We require software that can reason; that can draw conclusions; that can look beyond the evidence for general patterns; that can leap to conclusions; and that can retract these conclusions in the face of conflicting evidence received later. But, according to the story, while (standard) logic is good for drawing inferences it is monotonic and is no good for withdrawing inferences. What is needed is a new kind of logic, a nonmonotonic logic.

The standard story then usually continues with an example. Suppose we observed Tweety and notice that Tweety is a bird. Can Tweety fly? Well, most birds can fly, unless they are penguins or ostriches. But most birds are not penguins or ostriches, so we would like to conclude that Tweety can fly, until such time as we find that Tweety is a penguin or ostrich. Then we will not want to conclude that Tweety can fly. This is, plausibly, an example of the pattern of reasoning ‘In the absence of evidence to the contrary, assume A’ [Reiter (1986)].

Several schemes have been proposed for formalizing such nonmonotonic inferencin (see references cited in §1, above), but in fact none is needed. The essential flaw in the standard story is the assumption that we need to conclude that Tweety can fly, if we are to realize our goal of assuming that any bird can fly unless it is known to be a particular non-flier. In fact, if we make good use of the distinction between the conclusion

¹¹ Harman (1986) makes much use of this distinction.

of an argument and the rules of action associated with the conclusion, we can tell a version of the standard story without abandoning standard logic. In short, we can take action without sanctioning an invalid inference.

Here is the revised story. A knowledge base tells us that Tweety is a bird, that most birds fly, and that every bird flies except penguins or ostriches. Formally, we have

1. $\text{Bird}(\text{tweety}),$
 2. $\forall(x)(\text{Bird}(x) \rightarrow \text{Likely-flier}(x)),$
 3. $\forall(x)((\text{Bird}(x) \wedge \neg\text{Penguin}(x)$
 $\quad \wedge \neg\text{Ostrich}(x)) \rightarrow \text{Flies}(x)),$
 4. $\forall(x)(\text{Penguin}(x) \vee \text{Ostrich}(x)$
 $\quad \rightarrow \neg\text{Flies}(x)).$
- (15)

From these three statements it does not follow that Tweety flies, although it does follow that Tweety is in the category of things that are likely fliers. Suppose we need to conclude that Tweety flies. We can't. But we do not need to conclude that Tweety flies (nor do we need a special rule telling us to expect Tweety to fly). To see this, consider why we might want to make such a conclusion: to take action. If, for example, we are looking for Tweety, then it matters greatly whether Tweety can fly. If Tweety can fly, we shall look in a certain manner; and if Tweety cannot fly, we shall look in some other manner. More generally, assume that we have two rules of the form:

1. For all x , if $\text{Flies}(x)$, then take action $\phi(x).$
2. For all x , if $\neg\text{Flies}(x)$, then take action $\psi(x).$

where $\phi(x)$ and $\psi(x)$ may be arbitrarily complex expressions. So, if we could conclude that Tweety flies, we could derive that we are to take action $\phi(\text{tweety}).$

But these are our rules. We make them so we can change them. In particular, we might have

1. For all x , if $\text{Likely-flier}(x),$
 then take action $\phi(x).$
2. For all x , if $\neg\text{Likely-flier}(x),$
 then take action $\psi(x).$

But this will not quite work, for if we later find that Tweety is a penguin, we do not want to conclude at that time that we should take action $\phi.$ So here is what we might do. We might have

several rules of action, numbered sequentially, and a meta-rule that says to use the lowest-numbered applicable rule. For example, in the present case our rules of action might be

1. For all x , if $\text{Flies}(x)$, then do $A.$
 2. For all x , if $\neg\text{Flies}(x)$, then do $B.$
 3. For all x , if $\text{Likely-flier}(x)$, then do $C.$
- (16)

Suppose we had the statements in (15), above, in our knowledge base and we wanted to know what to do. By (16), above, we would do C , whatever that was, since we can neither conclude that Tweety flies or that he does not fly. If we subsequently learned that Tweety was a penguin, then (16) would direct us to do $B.$ The important thing to note is that we can use this scheme to take the place of a default assumption that Tweety flies, in the absence of information to the contrary, by making A and C identical.

It is worth noting that this move is quite general. Whenever we have a default rule we can reformulate it in a similar fashion. Suppose that α (a possibly complex statement) is defaultable, i.e., if we know that α we will conclude that β ; if we know that $\neg\alpha$ we will conclude that γ ; and if we know neither that α nor that $\neg\alpha$ we will assume that $\beta.$ Let $\text{Action}(\beta)$ and $\text{Action}(\gamma)$ be, respectively, the actions we would take if β or γ were known to be true. We may, then, lay out our rules of action as follows:

1. If α , then do $\text{Action}(\beta).$
2. If $\neg\alpha$, then do $\text{Action}(\gamma).$
3. If true, then do $\text{Action}(\beta).$

where our meta-rule is to employ the lowest numbered rule available (having a true condition). More complex situations would merely call for more complex antecedent conditions.

The difference between these two cases (non-monotonic logic and the method just presented) is that in the latter case we make the assumption that we should act as if Tweety were a flier if Tweety is a likely flier, while in the former case we make the assumption that Tweety is a flier on the grounds that Tweety is a likely flier. The difference in terms of our actions is nil, but in the second case we get a form of nonmonotonic reasoning without embracing nonmonotonic logic.

There is much more to be said about this method, but the point has been made that the

distinction between inferences and rules of action can be used to represent default reasoning without nonmonotonic inferencing (deduction), and that suffices for our present purposes.

4. Defaulting with Possible Worlds

So far, we have applauded the motivation for nonmonotonic logic, criticized the move to nonmonotonic logic (mainly on the grounds that its transformation rules fail to preserve truth), and suggested an alternate approach to reasoning with defaults. We distinguished between transformation rules (rules of inference) and decision rules (rules of action). By keeping the two distinct in a computer program we can create a default reasoning effect without invoking logical transformation rules that fail to preserve truth, and this allows us to represent in our programs what leaping to a conclusion really is: a decision, not an inference. (This is also what retracting an assumption is.)

In the present section, we take this basic idea, that default reasoning includes a large portion of decision making as well as inferencing, and develop it further. We do this by sketching a philosophical view of inquiry and action. This view is controversial and currently under much debate [see Stalnaker (1984)] for a sympathetic discussion). Nonetheless, we think it is essentially correct. We now present the framework, which we shall label the possible worlds approach to default reasoning.

The world we live in is but one of many possible worlds. In the actual world, Reagan beat Mondale for president in 1984, but there was (most would agree) the possibility that Mondale would beat Reagan. That is, in some possible worlds Reagan lost. Similarly, we can look to the future and say that in some possible worlds the Democratic nominee wins for president in the year 2000 and in other possible worlds the Republican nominee wins. We do not know, of course, whether the Democrats or the Republicans will win the presidential election in 2000. That is to say, we do not fully know which possible world is (the one and only) actual world; given an array of possible worlds to choose from, we often find the actual world undiscernible. Further, since there are a large (infinite?) number of possible worlds in which

the Republicans win and a large number in which the Democrats win, we do not even know which category of possible world we live in (at least with regard to who wins the 2000 presidential election).

But we have a need to know. We are agents with both an interest in knowing and an interest in taking action, preferably (but not necessarily) on the basis of knowledge. In our current jargon, we have an interest in determining which possible world is the actual world (or which category of possible world the actual world is in). In our more ordinary way of speaking, we would (assume for the sake of the argument) like to know whether the Republicans or the Democrats will win in 2000. The problem, of course, is that we do not have this knowledge (we do not know which possible world we are in) and (assume for the sake of the argument) we need to take action predicated on who wins the election.

All, including the advocates of nonmonotonic logic, would agree that this is simply another circumstance in which we shall have to stick our necks out and leap to a conclusion. As noted above, the advocates of nonmonotonic logic would have us do the leaping with logical transformation rules that fail to preserve truth. Our idea is that leaping to a conclusion is better represented as making an assumption, i.e., as tentatively constructing an hypothesis about which possible world (or category of possible worlds) we are in to use as our basis for action. In the argot of possible worlds, we have our rules of action and the prospect of many possible worlds before us. We make a decision, which may be the result of a very complex deductive process, about which possible world to assume in actual as input to our action rules. Then, we do as the rules dictate. Later after more information has come in and we are faced with another decision about which action rules to invoke, our list of possible worlds has changed as, perhaps, have our knowledge bases about them. In any case, we reason again as to what should be done, and we act

The overall structure, even simplified for the sake of discussion, is complex, but that is not particularly surprising. Reasoning is complex and our structure is meant to be a good, early approximation to that process. Let us now look at an example, the usual one about Tweety, and a simple implementation (in Prolog) of a possible worlds based system for default reasoning.

The Prolog program, whynml_a.pl, found in appendix B, supports the following interactions, thereby making it a program that supports non-monotonic reasoning.

```
?- do_this (find(tweety)).
using world: 1
Look in the sky.
    yes
?- discover(world(1,p(penguin(tweety),true))).
    yes
?- do_this (find(tweety)).
Using world: 2
Look in the barnyard.
    yes
?- undiscover(world(1,p(penguin(tweety),true))).
yes
?- do_this (find(tweety)).
Using world: 1
Look in the sky.
    yes
```

The interpretation here is that we begin by giving the program a goal (*do_this*) of locating Tweety (*find(tweety)*). On the basis of its current information, the program concludes that the possible world (world 1) in which Tweety can fly is the best possible world to use in satisfying the goal, so the action rules are applied on the assumption that world 1 is the actual world. There are three action rules. The first directs us, for all *x*, to look in the sky if the goal is to find *x* and *x* files and *x* was last seen in the barnyard (which we are assuming of Tweety).

Next, in the above interaction with the program, we record our discovery that (in world 1) Tweety is in fact a penguin. The program accepts our information and eliminates world 1 as a possible world under consideration, because it is inconsistent with our Tweety being a penguin. That leaves world 2 as the possible world under consideration with the highest utility. In that world Tweety is a penguin. When we then ask the program to find Tweety, it assumes that world 2 is actual, finds that Tweety is not a flier, invokes an appropriate action rule, and advises us to look in barnyard.

Several comments are in order about this possible worlds approach to default reasoning. First, and most important, is that although the program

displays nonmonotonicity, its inferences remain within the familiar confines of standard (truth-preserving) logic. Nonmonotonicity is achieved by implementing decisions (e.g., by altering the list of possible worlds under consideration and by altering the descriptions of the possible worlds), rather than by altering standard inferencing procedures.

A second point is that the idea of possible worlds is more than a nice way of thinking about certain things. There is a well-developed branch of logic, called modal logic, that is an apt modeling tool for representing possible worlds.¹² For a large number of modal logic systems both the proof theory and the model theory (semantics) have been successfully worked out. In fact, the semantics commonly used for modal systems are called possible worlds semantics in the literature. Thus, by taking a possible worlds approach to default reasoning we become heirs to some well-established logic modeling tools.

A third point is that this system (and approach) need not be unidirectional. The *discover/1* predicate/procedure can be inverted so as to remove a descriptor and to recall possible worlds that had been taken out of consideration. With an *undiscover/1* predicate available, for inverting the *discover/1* predicate, it is a simple matter to produce a what-if capability on this system of logic models.

```
what_if(Goal,Statement) :-  
    discover(Statement),  
    do_this(Goal),  
    undiscover(Statement).
```

Thus, whynml_a.pl contains a simple truth maintenance system. Assumptions may be added and removed and the program can maintain a collection of considered possible worlds each of which is internally consistent with the added and removed assumptions. (We note that the algorithm used in whynml_a.pl is a heuristic, albeit a sensible one.)

A fourth point is that the possible worlds framework (modal logic) is used by logicians and philosophers because of its semantic power for representing certain concepts, many of which are of much interest for management applications.

¹² Modal concepts may, of course, be expressed by relativization in standard predicate logic [Lewis (1967), Lewis (1986), Forbes (1985)]. When we speak here of modal logic we mean to include the logical representation of modal concepts in predicate logic as well.

For example, a logical truth is true in all possible worlds, as are necessary truths [Burks (1977), Plantinga (1974)]. Laws of nature (scientific laws) can be interpreted so as to constrain possible worlds [Burks (1977)]. Counterfactual conditionals and causal locutions, which have proved so difficult to represent [Kowalski et al. (1986)], have a natural interpretation in the possible worlds framework. For example, ‘If Mondale had beaten Reagan in 1984 we would not be funding the anti-government forces in Nicaragua’ is true if and only if in every possible world in which Mondale beat Reagan we are not funding the anti-government forces in Nicaragua.¹³ Representation of these, and other concepts (e.g., possibility, necessity, subjunctive expressions), requires correct representation of the logical relations (accessibility) among possible worlds.

A fifth point has to do with representation, in a program, of possible worlds. We shall limn three different approaches.¹⁴ Our discussion here is meant to be introductory. Fuller development of these ideas will be the subject of future research. In the first approach, which we call the scenario approach, each possible world under consideration is represented explicitly and distinct from the other representations of worlds. There is no common information shared by the different world representations. A control program is responsible for picking which world to employ when queries are made to the system. By designing the control program to pick different scenarios according to new information it receives, a form of nonmonotonic reasoning can be achieved. This sort of context switching is workable in many circumstances and is in fact a common programming technique. The main difficulty with this approach is that all the knowledge (or information) about the various scenarios must be explicitly encoded in each scenario in the program. If, for example, several of the scenarios make assumptions in common, these assumptions must be explicitly repeated in each world representation, and if a common assumption is retracted it must be retracted from each world representation. The

scenario approach is used in whynml_a.pl. Each possible world under consideration is fully described. Thus, for example, laws that hold in each possible world (e.g., that ostriches do not fly) must be explicitly represented as holding in each world under consideration.

In the second approach, which we call the worlds explicit approach, different (classes of) possible worlds under consideration are represented explicitly, but shared assumptions are possible. Certain statements could be assumed to hold in the actual world (e.g., `world(actual,p(bird(tweety),true))`) in whynml_a.pl). The decision problem for the program is first, to determine which of the possible worlds under consideration could possibly be the actual world (given the assumed (partial) description of the actual world), and second, to determine which of these it is best tentatively to assume is a description of the actual world. With such an approach, the d/1 meta-interpreter (in whynml_a.pl) would be modified so that its deductions could employ both the axioms describing the actual world and the axioms describing the possible world chosen for the occasion. The logical effect of this is to treat axioms describing the actual world as necessary truths – true in all the possible worlds. An advantage of this approach is that world descriptions can be shortened because assumptions common to all can be attributed to the actual world and the program can automatically manage the relevant inferencing.¹⁵

In the third approach, which we call the worlds implicit approach, only the actual world is represented explicitly. An indefinite number of possible worlds are represented implicitly through a series of axioms about what might be. Returning to Tweety, in whynml_a.pl we say that in world 2 Tweety is a penguin, in world 3 he is an ostrich, and so on. Without specifying any possible world explicitly, we might simply state that certain things are possible (although they may not all be possible in the same world, or even in the actual world, for this is the sort of thing the program is to deduce). Consider a simple example to illustrate the point. Switching from Prolog to logic, let our axioms be

¹³ The authoritative work on counterfactuals is Lewis (1973).

¹⁴ A fourth approach would be to develop a logic that could be used directly to effect a switch among possible worlds under consideration. This approach is explored by [Belzer and Lower (1988), Nute (1986), Nute (1988)].

¹⁵ A hierarchy of assumptions common to subsets of worlds under consideration could be implemented as well.

as follows.

1. $\forall x(Bird(x) \rightarrow \Diamond Ostrich(x))$,
2. $\forall x(Bird(x) \rightarrow \Diamond Penguin(x))$,
3. $\forall x(Bird(x) \rightarrow \Diamond(\neg Penguin(x) \wedge \neg Ostrich(x)))$, (17)
4. $\forall x(Bird(x) \wedge \neg Ostrich(x) \rightarrow Flies(x))$.

(The operator, \Diamond , for ‘It is possible that...’, is modal. The first axiom is translated roughly as ‘Any bird might be an ostrich.’) In a worlds implicit representation the program’s decision problem is first, to determine which possibilities (if any) could be actual and if so would permit a deduced answer to the query at hand, and second, to determine the best of these possibilities and to make appropriate assumptions. To illustrate, a meta-interpreter for determining whether Tweety flies would collect the statements that are possibly true but not known actually to be true and that were they actually true would permit a deduced answer to the question at hand. In the present case that would be $\Diamond \neg Ostrich(tweety)$, which is deducible from the above axioms. Then, instead of defaulting and inferring that what is possible is actual, a meta-interpreter could consider the consequences of making various assumptions, could choose an assumption based on a preference model, and could record the assumption for later use.

These three methods each have their various advantages and drawbacks, detailed discussion of which we leave for future research. What is most significant is that each allows us to separate decisions (about what to assume) from inferences (from what has been found or assumed), thus permitting nonmonotonicity of reasoning without having to resort to inference that is not deductively valid. Further, these approaches permit default policies of essentially any sort to be stated (and modified and examined) explicitly. In whynml.apl, the available world with the highest utility is the one assumed for the purposes at hand. Although a trivial utility function was used (take the lowest-numbered world), the point is that an arbitrary utility function could be employed for determining a default policy. Finally, we note that the computational costs introduced by a possible worlds approach to nonmonotonic reasoning lie largely in selecting which possible world will be used for the decision problem at hand. Depending

upon our rules for selecting possible worlds, this cost may well be (as it is in the present example) quite low.

In short, although there are many questions to be investigated with respect to the possible worlds approach, it is inherently both an intuitively plausible and a very powerful framework for representing default reasoning.

5. Summary and Conclusion

The motivation for nonmonotonic logic is to produce a machine representation for default reasoning, broadly construed. But, we have argued, all nonmonotonic logics have (by definition) inference rules that fail to preserve truth, and this fact leads to several undesirable features. In response to these problems, but recognizing the importance of the original motivation for nonmonotonic logic, we have proposed an alternative to nonmonotonic logic, which achieves nonmonotonicity of reasoning without abandoning in any way truth preserving inferences. This approach is based on a possible worlds framework, which we have illustrated in the previous section. Motivating this approach is an important distinction, which we believe the advocates of nonmonotonic logic to be ignoring: that between inferencing and making decisions, or equivalently that between recognizing what follows from a theory and theory construction.

The possible worlds framework is extremely powerful, not only for supporting default reasoning, but for representing concepts that otherwise cannot be captured easily or at all. We have only hinted at the possibilities, but with this framework at hand we are confident that the way is open for exciting progress.

Finally, we want to note that we are only claiming that the possible worlds approach to default reasoning is a promising one. There are, in fact, deep and difficult questions to be investigated. Reiter (1986) notes that there are two parts of a formal theory of default reasoning:

- (i) A method for representing default assumptions.
- (ii) A theory of ‘truth maintenance’, i.e., of how a system is to be maintained when assumptions and inferences have to be retracted.

On (i) there lies ahead the problem of generating possible world representations, including how many should be represented in the system and what their characteristics should be.¹⁶ On (ii) the accessibility relation between possible worlds will greatly complicate the truth maintenance problem. But we leave these issues for future work.

6. Bibliographic Note

The literature on nonmonotonic logic and nonmonotonic reasoning has been growing rapidly. Three seminal papers are McDermott and Doyle (1980), Reiter (1980) and McCarthy (1980). In 1984 the AAAI held a workshop on nonmonotonic reasoning. The proceedings are available from the AAAI [American Association for Artificial Intelligence (1984)] and contain a comprehensive bibliography through mid-1984. Besides the papers cited above, the following are useful: Nute (1986), Brweka (1986), Morris and Nado (1986), Winslett (1986). The idea of possible worlds is a metaphysical notion, originally due to Leibniz. It has been recently revived among philosophers and logicians and used for a variety of purposes [e.g., Stalnaker (1984), Lewis (1983), Lewis (1985), Burks (1977), Plantinga (1974)]. Possible world concepts have been very closely associated with recent work in modal logic. An elementary introduction to the subject is available [Bradley and Swartz (1979)] as are more advanced textbooks [e.g., Hughes and Cresswell (1968), Chellas (1980), Forbes (1985)]. The collection of essays edited by Loux (1979) contains many useful research papers and Loux's introduction is particularly clear.

Appendix A

The purpose of this appendix is to prove in sentence logic (but without loss of generality) that any nonmonotonic logic in which the deduction theorem holds is inconsistent.

¹⁶ There are, in general, infinitely many possible worlds, and certainly too many to represent explicitly in any software system. See Kimbrough (1982) for a discussion of the claim that in everyday reasoning pragmatic criteria are employed to limit the possible worlds under consideration to a tractable, and finite, number.

To begin, assume that ψ is a formula that can be deduced in an arbitrary nonmonotonic logic from a specific collection of axioms, and that cannot be deduced in monotonic logic from that collection of axioms. ψ 's existence is guaranteed by the definition of nonmonotonic logic, above. Specifically, we have

$$\gamma \vdash_{NML} \psi,$$

but not

$$\Gamma \vdash \phi.$$

On the assumption that the deduction theorem is available we would also have

$$\vdash_{NML} (\Gamma \rightarrow \psi)$$

as a thesis. Now, $(\Gamma \rightarrow \psi)$ is equivalent (in sentence logic) to some formula, call it Δ , that is in conjunctive normal form, so we may derive Δ .

A word about conjunctive normal form. A formula in conjunctive normal form is a conjunction of disjuncts of literals. A literal is a sentence letter or the denial of a sentence letter, so a formula in conjunctive normal form looks like this:

$$((\phi_{11} \vee \dots \vee \phi_{1,m}) \wedge \dots \wedge (\phi_{n,1} \vee \dots \vee \phi_{n,p})),$$

with the $\phi_{i,j}$'s (the literals) such that $i, j \geq 1$, and for all i, j, k , $\phi_{i,k}$ is different than $\phi_{i,j}$ if $k \neq j$. In sentence logic any formula can be converted to conjunctive normal form and any thesis in conjunctive normal form will be a disjunction of some literal and its denial (and possibly other literals as well). To illustrate, the conjunctive normal form of axiom 2 of PM is $(\neg Q \vee P \vee Q)$, and that for axiom 3 is $((\neg P \vee Q \vee P) \wedge (\neg Q \vee Q \vee P))$.

Returning now to Δ , which is in conjunctive normal form, we know that at least one conjunct in Δ does not contain both a sentence letter and its denial. Were this not the case, Δ would be a thesis of sentence logic, contrary to our assumption that ϕ could not be deduced in sentence logic from the axioms present. That conjunct is also a thesis (if $(\phi \wedge \psi)$ is a thesis, then so is ϕ) and it has the following form:

$$(\phi_1 \vee \dots \vee \phi_n),$$

with $n \geq 1$. If $n = 1$, then ϕ_1 is both a literal (either a sentence letter or the denial of one) and a thesis, i.e., we have $\vdash_{NML} \phi_1$. From TR1 of MP (uniform substitution rule), the result of uniformly substituting any well-formed formula for another well-formed formula in a thesis is also a thesis. In

the thesis ϕ_1 substitute $\neg\phi_1$. This gives us both ϕ_1 and $\neg\phi_1$ as theses, and it gives us a contradiction for the special case that $n \geq 1$.

In the general case of n being greater than one, apply TR1 to our thesis

$$(\phi_1 \vee \dots \vee \phi_n), \quad (\text{A.1})$$

with $n > 1$, substituting $\neg\phi_i$ for all ϕ , with i greater than one, yielding the thesis

$$(\phi_1 \vee \dots \neg\phi_2 \vee \dots \vee \neg\phi_n). \quad (\text{A.2})$$

Given (A.1) and (A.2), ϕ_1 follows and is a thesis, thus reducing the problem to the case, discussed above, in which $n = 1$. So we have a contradiction here as well.

In sum, the deduction theorem holds in sentence logic, and any nonmonotonic logic in which sentence logic is contained and for which the deduction theorem holds is an inconsistent non-monotonic logic.

Appendix B

```
/* ***** WHYNML_A.PL **** */
/* Steven O. Kimbrough */
/* Last revision: November 3, 1987 */

/* ***** An example of worlds explicit **** */
/* ***** possible worlds under consideration **** */

/* specific worlds */

worlds (1,in).
worlds (2,in).
worlds (3,in).

/* generic clauses for worlds under consideration */

world(1,p(flies(X),(bird(X),nota(penguin(X)),nota(ostrich(X)) ))).
world(1,p(nota(flies(X)),ostrich(X))).
world(1,p(nota(flies(X)),penguin(X))).
world(2,p(flies(X), ( bird(X),nota(penguin(X)),nota(ostrich(X)) ))).
world(2,p(nota(flies(X)),ostrich(X))).
world(2,p(nota(flies(X)),penguin(X))).
world(3,p(flies(X), ( bird(X),nota(penguin(X)),nota(ostrich(X)) ))).
world(3,p(nota(flies(X)),ostrich(X))).
world(3,p(nota(flies(X)),penguin(X))).
world(1,p(bird(tweety),true)).
world(2,p(bird(tweety),true)).
world(3,p(bird(tweety),true)).
world(2,p(last_seen(tweety,'in the barnyard'),true)).

/* specific clauses for worlds under consideration */

world(1,p(nota(penguin(tweety)),true)).
world(1,p(nota(ostrich(tweety)),true)).
world(2,p(penguin(tweety),true)).
world(2,p(nota(ostrich(tweety)),true)).
```

```
world(3,p(ostrich(tweety),true)).
world(3,p(nota(penguin(tweety)),true)).
```

```
/* ***** * Main controlling predicates. * * * * * */
```

```
do_this(X) :-
  X = find(Y),
  best_world(W),
  action_rule(X,W,R),
  report(W,R).
```

```
action_rule(find(X),W,R) :-
```

```
  d(flies(X),W),
```

```
  R = 'the sky'.
```

```
action_rule(find(X),W,R) :-
```

```
  d(nota(flies(X)),W),
```

```
  d(last_seen(X,R),W).
```

```
action_rule(_, _, 'No more advice.').
```

```
d(true,W) :-!.
```

```
d(X,W) :-
```

- world(*W,p(X,true)*),
- !.

```
d(X,Y,W) :-
```

- d(*X,W*),

- d(*Y,W*),

- !.

```
d(X,W) :-
```

- world(*W,p(X,Y)*),

- not(*Y = true*),

- d(*Y,W*),

- !.

```
/* *** Utility predicates in alphabetical order *** */
```

```
asserted([]).
```

```
asserted_this(X) :-
```

- asserted(*Xs*),

- retract(asserted(*Xs*)),

- assert(asserted([*X* | *Xs*])).

```
assert_this(X) :-
```

- asserted(*Xs*),

- retract(asserted(*Xs*)),

- assert(asserted([*X* | *Xs*])),

- assert(*X*).

```
best_world(X) :-
```

- worlds(*X,in*).

```

cleanup :-  

    asserted(Ls),  

    retract_list(Ls),  

    retract(asserted(Ls)),  

    assert(asserted([])).  
  

/* discover #1: The world is already out put for other reasons. */  

discover(world(W,p(H,B))) :-  

    clause(worlds(W,out,[Ps]),[true]),  

    not(member(world(W,p(H,B)),Ps)),  

    retract_this(worlds(W,out,[Ps])),  

    assert_this(worlds(W,out,[world(W,p(H,B))|Ps])).  

/* discover #2: The world is in, but so is the discovery. */  

discover(world(W,p(H,B))) :-  

    clause(worlds(W,in),[true]),  

    world(W,p(H,B)),  

    asserted_this(world(W,p(H,B))).  

/* discover #3: The world is in, but the addition would create an inconsistency. */  

discover(world(W,p(H,B))) :-  

    clause(worlds(W,in),[true]),  

    d((nota(H),B),W),  

    retract_this(worlds(W,in)),  

    assert_this(worlds(W,out,[world(W,p(H,B))])),  

    assert_this(world(W,p(H,B))).  

/* discover #4: Ignoring possible redundancies, we assert the finding. */  

discover(world(W,p(H,B))) :-  

    clause(worlds(W,in),[true]),  

    assert_this(world(W,p(H,B))).  

discover(_).  
  

remove(X,[],[]).  

remove(X,[X|Xs],Y) :-  

    remove(X,Xs,Y).  

remove(X,[Y|Xs],[Y|Zs]) :-  

    not(X = Y),  

    remove(X,Xs,Zs).  
  

report(W,R) :-  

    nl,  

    write('Using world: '),
    write(W),
    nl,
    ((R = 'No more advice.', write(R));
     (write('Look in '),
      write(R),
      write('.'))),
    nl.  
  

retract_this(X) :-  

    asserted(Xs),  

    remove(X,Xs,Ys),

```

```

retract(asserted(Xs)),
assert(asserted(Ys)),
retract(X).

retract_list([]).
retract_list([L | Ls]) :-
    retract(L),
    retract_list(Ls).
/* undiscover #1: The world is in already. */
undiscover(world(W,p(H,B))) :-
    asserted(Ps),
    member(world(W,p(H,B)),Ps),
    clause(worlds(W,in),[true]),
    world(W,p(H,B)),
    retract_this(world(W,p(H,B))).
/* undiscover #2: The world is out just for this reason. */
undiscover(world(W,p(H,B))) :-
    asserted(Ps),
    member(worlds(W,out),[world(W,p(H,B))]),Ps),
    retract_this(worlds(W,out,[world(W,p(H,B))])),Ps),
    asserta(worlds(W,in)),
    retract(world(W,p(H,B))).
/* undiscover #3: The world is out for other reasons as well. */
undiscover(world(W,p(H,B))) :-
    asserted(Ps),
    member(worlds(W,out,PPs),Ps),
    remove(worlds(W,out,PPs),Ps,Pss),
    retract(asserted(Ps)),
    remove(world(W,p(H,B)),PPs,NewList),
    assert(asserted([worlds(W,out,NewList) | Pss])),
    retract(world(W,p(H,B))).

```

References

- American Association for Artificial Intelligence, Nonmonotonic Reasoning Workshop (Oct. 17–19, 1984).
- M. Belzer and B. Loewer, A Conditional Logic for Defeasible Beliefs, *Decision Support Systems* 4, nr. 1 (1988).
- Raymond Bradley and Norman Swartz, *Possible Worlds: An Introduction to Logic and Its Philosophy* (Hackett, Indianapolis, 1979).
- Gerhard Brweka, Tweety – Still Flying, Proceedings AAAI-86, Philadelphia, PA (1986) 8–12.
- Arthur W. Burks, *Chance, Cause, Reason* (The University of Chicago Press, Chicago, 1977).
- Lewis, Carroll, What the Tortoise Said to Achilles, *Mind* (1895) 278–280.
- Brian F. Chellas, *Modal Logic: An Introduction* (Cambridge University Press, Cambridge, 1980).
- Johan de Kleer, An Assumption-based TMS, *Artificial Intelligence* 28, no. 2 (1986a) 127–162.
- Johan de Kleer, Extending the ATMS, *Artificial Intelligence* 28, no. 2 (1986b) 163–196.

- Johan de Kleer, Problem Solving with the ATMS, *Artificial Intelligence* 28, no. 2 (1986c) 197–224.
- Graeme Forbes, *The Metaphysics of Modality* (Oxford University Press, Oxford, 1985).
- Ian Hacking, What Is Logic? *Journal of Philosophy* LXXVI, no. 6 (1979) 285–319.
- Steve Hanks and Drew McDermott, Default Reasoning, Nonmonotonic Logics, and the Frame Problem, *Proc. AAAI-86* (1986) 328–333.
- Steve Hanks and Drew McDermott, Nonmonotonic Logic and Temporal Projection, *Artificial Intelligence* 33, no. 3 (1987) 379–412.
- Gilbert Harman, *Change in View* (The MIT Press, Cambridge, MA, 1987).
- G.E. Hughes and M.J. Cresswell, *An Introduction to Modal Logic* (Methuen, London, 1968).
- Steven O. Kimbrough, Circumventing the Problems of Induction: A Theory of Rational Hypothesis Choice, University of Wisconsin – Madison, unpublished Ph.D. thesis (1982).
- Steven O. Kimbrough and Fred Adams, Alternatives to Nonmonotonic Logic, *Proc. 19th Hawaii Int'l Conf. System Sciences*, vol. IA, Yoahan Chu et al., eds. (1986) 589–594.

- Stephen Cole Kleene, *Mathematical Logic* (Wiley, New York, 1967).
- David Lewis, Counterpart Theory and Quantified Modal Logic, *Journal of Philosophy* 65 (1968) 113–26; reprinted with added postscripts in Lewis (1983).
- David Lewis, *Collected Papers*, volume I (Oxford University Press, Oxford, 1983).
- David Lewis, *On the Plurality of Worlds* (Basil Blackwell, Oxford, 1986).
- Michael J. Loux, ed., *The Possible and the Actual: Readings in the Metaphysics of Modality* (Cornell University Press, Ithaca, NY, 1979).
- J. McCarthy, Circumscription – A Form of Nonmonotonic Reasoning, *Artificial Intelligence* 13, nos. 1 & 2 (1980) 27–39.
- D. McDermott and J. Doyle, Nonmonotonic Logic I, *Artificial Intelligence* 13, nos. 1 & 2 (1980) 41–72.
- Marvin Minsky, A Framework for Representing Knowledge, Massachusetts Institute of Technology, Memo 3306 AI Lab (1981).
- Morris, Paul H. and Robert A. Nado, Representing Actions with an Assumption-Based Truth Maintenance System, *Proc. AAAI-86* (1986) 13–17.
- Donald Nute, A Nonmonotonic Logic Based on Conditional Logic, *Advanced Computational Methods Center, The University of Georgia, Athens, GA, ACMC Research Report 01-0007* (1986).
- Donald Nute, Defeasible Reasoning and Decision Support Systems, *Decisions Support Systems, Decision Support Systems* 4, nr. 1 (1988).
- Alvin Plantinga, *The Nature of Necessity* (Oxford University Press, London, 1974).
- R. Reiter, A Logic for Default Reasoning, *Artificial Intelligence* 13 nos. 1 & 2 (1980) 81–132.
- R. Reiter, Survey: On Default Reasoning, presentation at AAAI-86 (1986).
- M.J. Sergot, F. Sadri, R.A. Kowalski, F. Kriwaczek, P. Hammond and H.T. Cory, The British Nationality Act as a Logic Program, *Communications of the ACM* 29, no. 5 (May, 1986) 370–386.
- Robert Stalnaker, *Inquiry* (The MIT Press, Cambridge, MA, 1984).
- Bas C. van Fraassen, *Formal Semantics and Logic* (Macmillan, New York, 1971).
- Marianne Winslett, Is Belief Revision Harder Than You Thought? *Proc. AAAI-86* (1986) 421–427.