McDermott, Drew; & Doyle, Jon (1980),
"Non-Monotonic Logic I", Artificial Intelligence
13: 41-72;
reprinted in:

Readings in Nonmonotonic Reasoning

Edited by
Matthew L. Ginsberg
Stanford University

MORGAN KAUFMANN PUBLISHERS, INC.
LOS ALTOS, CALIFORNIA
© 1987

3.2 Modal Approaches

The modal approach to nonmonotonic inference is probably the least explored of the three principal formalisms. The reasons, I suspect, are that the papers describing the work often have been difficult to understand, and the formal work itself has been somewhat unsatisfactory until recently.

The first paper to try this approach is McDermott and Doyle's 1980 paper, "Nonmonotonic logic I." Motivational material is presented in the first three sections, after which the authors turn to the formalities of the modal approach.

Unfortunately, the formalism they present is too weak to be of much use. As McDermott himself remarks in the sequel [McDermott, 1982], "There was not actually any relation between the truth values of p and Lp." McDermott attempted to patch this by using a modal theory based on one of the modal logics T, $S4$ or $S5$. Unfortunately, these approaches also are flawed, since only the logic $S5$ is strong enough to capture the notions McDermott is trying to formalize, but nonmonotonic $S5$ in fact collapses to a monotonic theory. So something else is needed.

Moore supplies the something else in autoepistemic logic, introduced in "Semantical considerations on nonmonotonic logic." He also begins with a variety of motivational material before turning to formal matters in Section 3. The paper is extremely clearly written, although the proofs of many formal results were missing until the appearance of his next paper on the subject, "Possible-world semantics for autoepistemic logic."

In this second paper, Moore draws a connection between autoepistemic logic and a Kripke-style approach to modal logic based on possible worlds [Kripke, 1971]. Because of the fairly constructive nature of Kripke's approach, Moore is able to prove a variety of claims made in the earlier paper about (for example) the existence of multiple extensions in certain cases. He remarks that Halpern and Moses [Halpern, Moses, 1984] obtained similar results independently; more recently, Levesque [Levesque, 1987] has explored these ideas as well.

Non-Monotonic Logic I *

Drew McDermott

Department of Computer Science, Yale University,
New Haven, CT 06520, U.S.A.

Jon Doyle

Artificial Intelligence Laboratory, Massachusetts Institute of
Technology, Cambridge, MA 02139, U.S.A.

Recommended by Nils Nilsson

ABSTRACT

'Non-monotonic' logical systems are logics in which the introduction of new axioms can invalidate old theorems. Such logics are very important in modeling the beliefs of active processes which, acting in the presence of incomplete information, must make and subsequently revise assumptions in light of new observations. We present the motivation and history of such logics. We develop model and proof theories, a proof procedure, and applications for one non-monotonic logic. In particular, we prove the completeness of the non-monotonic predicate calculus and the decidability of the non-monotonic sentential calculus. We also discuss characteristic properties of this logic and its relationship to stronger logics, logics of incomplete information, and truth maintenance systems.

1. Introduction

'Non-monotonic' logical systems are logics in which the introduction of new axioms can invalidate old theorems. Such logics are very important in modeling the beliefs of active processes which, acting in the presence of incomplete information, must make and subsequently revise assumptions in light of new observations. We present the motivation and history of such logics. We develop model and proof theories, a proof procedure, and applications for one non-monotonic logic. In particular, we prove the completeness of the non-monotonic predicate calculus and the decidability of the non-monotonic sentential calculus. We also discuss characteristic properties of this logic and its relationship to stronger logics, logics of incomplete information, and truth maintenance systems.

2. The Problem of Incomplete Knowledge

The relation between formal logic and the operation of the mind has always been unclear. On one hand, logic was originally the study of the 'laws of thought,' and has remained at least in principle a prescriptive theory of the operation of the 'ideal' mind. On the other hand, many logicians have often been unduly uninterested in ordinary 'non-ideal' mental phenomena. Some of the more striking differences between properties of formal logics and mental phenomenology occur in situations dealing with perception, ambiguity, common sense, causality and prediction. One common feature of these problems is that they seem to involve working with incomplete knowledge. Perception must account for the noticing of overlooked features, common sense ignores myriad special exceptions, assigners of blame can be misled, and plans for the future must consider never-to-be-realized contingencies. It is this apparently unavoidable making of mistakes in these cases that leads to some of the deepest problems of the formal analysis of mind.

Some studies of these problems occur in the philosophical literature, the most relevant here being Rescher's [36] analysis of counterfactual conditionals and belief-contravening hypotheses. In artificial intelligence, studies of perception, ambiguity and common sense have led to knowledge representations which explicitly and implicitly embody much information about typical cases, defaults, and methods for handling mistakes [28, 35]. Studies of problem-solving and acting have attempted representing predictive and causal knowledge so that decisions to act require only limited contemplation, and that actions, their variations, and their effects can be conveniently described and computed [6, 9, 10, 11]. Indeed, one of the original names applied to these efforts, 'heuristic programming,' stems from efficiency requirements forcing the use of methods which occasionally are wrong or which fail. The possibility of failure means that formalizations of reasoning in these areas must capture the process of revisions of perceptions, predictions, deductions, and other beliefs.

In fact, the need to revise beliefs also occurs in certain implementations of deductive systems working within traditional logics. Much work has been done on mechanized proof techniques for the first-order predicate calculus [29, 30, 38]. Incomplete information is represented in these systems as disjunctions of the several possibilities where the individual disjuncts may be independent of the axioms being used, that is, cannot be proven or contradicted by arguments from the axioms. Thus, proof procedures engage in *case-splitting*, in which disjuncts are considered in a case-by-case fashion. At any given time, the proof procedure will have some set of current assumptions, from which the current set of formulas has been derived. If failures in the proof attempt lead to investigating new splits, and so change the set of current assumptions, the current set of derived formulas must also be updated, for it is the current set of formulas on which the proof procedure bases its actions.

Classical symbolic logic lacks tools for describing how to revise a formal theory

* This research was conducted at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract number N00014–75–C–0643, and in part by NSF Grant MCS77–04828.

to deal with inconsistencies caused by new information. This lack is due to a recognition that the general problem of finding and selecting among alternate revisions is very hard. (For an attack on this problem, see Rescher [36]. Quine and Ullian [33] survey the complexities.) Although logicians have been able to ignore this problem, philosophers and researchers in artificial intelligence have been forced to face it because humans and computational models are subject to a continuous flow of new information. One important insight gained through computational experience is that there are at least two different problems involved, what might be called 'routine revision' and 'world-model reorganization.'

World-model reorganization is the very hard problem of revising a complex model of a situation when it turns out to be wrong. Much of the complexity of such models usually stems from parts of the model relying on descriptions of other parts of the model, such as inductive hypotheses, testimony, analogy, and intuition. An example of such large-scale reorganization would be the revision of a Newtonian cosmology to account for perturbations in Mercury's orbit. Less grand examples are children's revisions of their world-models as discovered by Piaget, and the revision of one's opinion of a friend upon discovering his dishonesty.

Routine revision, on the other hand, is the problem of maintaining a set of facts which, although expressed as universally true, have exceptions. For example, a program may have the belief that all animals with beaks are birds. Telling this program about a platypus will cause a contradiction, but intuitively not as serious a contradiction as those requiring total reorganization. The relative simplicity of this type of revision problem stems from the statement itself expressing what revisions are appropriate by referring to possible exceptions. Such relatively easy cases include many forms of inferences, default assumptions, and observations.

Classical logics, by lumping all contradictions together, has overlooked the possibility of handling the easy ones by expanding the notation in which rules are stated. That is, we could have avoided this problem by stating the belief as 'If something is an animal with a beak, then *unless proven otherwise*, it is a bird.' If we allow statements of this kind, the problem becomes how to coordinate sets of such rules. Each such statement may be seen as providing a piece of advice about belief revision; for this approach to make sense, all the little pieces of advice must determine a unique revision. This is the subject of this paper. Of course, if we are successful, the world-model reorganization problem will still be unsolved. But we hope factoring out the routine revision problem will make the more difficult problem clearer.

3. Approaches to Non-Monotonic Logic and the Semantical Difficulties

The study of the problem of formalizing the process of revision of beliefs has been almost completely confined to the practical side of artificial intelligence research, where much work has been done [6, 12, 26, 42]. Theoretical foundations for this work have been lacking. This paper studies the foundations of these forms of reasoning with revisions, which we term *non-monotonic* logic.

Traditional logics are called *monotonic* because the theorems of a theory are always a subset of the theorems of any extension of the theory. (This name for this property of classical logics was used, after a suggestion by Pratt, in Minsky's [28] discussion. Hayes [11] has called this the 'extension' property.) In this paper, by *theory* we will mean a set of axioms. A more precise statement of monotonicity is this: If A and B are two theories, and $A \subseteq B$, then $\mathrm{Th}(A) \subseteq \mathrm{Th}(B)$, where $\mathrm{Th}(S) = \{p: S \vdash p\}$ is the set of theorems of S. We will be even more precise about the definition of $\vdash$ in Section 5.

Monotonic logics lack the phenomenon of new information leading to a revision of old conclusions. We obtain non-monotonic logics from classical logics by extending them with a modality ('consistent') well-known in artificial intelligence circles, and show that the resulting logics have well-founded, if unusual, model and proof theories. We introduce the proposition-forming modality M (read 'consistent'). Informally, $\mathsf{M}p$ is to mean that p is consistent with everything believed. (See [24].) Thus one small theory employing this modality would be

(1) $\text{noon} \wedge \mathsf{M}[\text{sun-shining}] \supset \text{sun-shining}$
(2) noon
(3) $\text{eclipse} \supset \neg\text{sun-shining}$,

in which we can prove

(4) sun-shining.

If we add the axiom

(5) eclipse

then (4) is inconsistent, so, by (1), (4) is not a theorem of the extended theory.

The use of non-monotonic techniques has some history, but until recently the intuitions underlying these techniques were inadequate and led to difficulties involving the semantics of non-monotonic inference rules in certain cases. We mention some of the guises in which non-monotonic reasoning methods and belief revising processes have appeared.

Scriven [40, 41] proposed that explanations, and in particular historical explanations of rational actions or decisions, are based not on universal or statistical laws, but rather on truisms or more generally, what he terms 'normic' statements. Normic statements include such statements as "In delicate circumstances, rational men act cautiously," or "All murders are committed from motives of revenge, lust, jealously, hate, greed, or fear." Such statements frequently involve terms such as 'naturally,' 'normally,' 'typically,' 'tendency,' 'ought,' 'should,' and others. Normic statements provide plausible explanations of actions or situations, explanations which may be invalidated by providing exceptions, special cases, or other mediating circumstances; that is, instances of normic statements are *defeasible*. In this way, normic statements seem closely related to statements expressible in non-monotonic logic. Scriven pointed out that while in some cases normic statements could imply statistical statements and so have some predictive power, in other cases normic statements can only supply coherent explanations, cannot rule out alternative coherent arguments, and thus fail to have predictive power. De Kleer [4] amplifies this point in considering explanations of the behavior of designed artifacts. As we shall see in this paper, this circumstance is also highly suggestive of non-monotonic logic.

In PLANNER [12], a programming language based on a negationless calculus, the THNOT primitive formed the basis of non-monotonic reasoning. THNOT, as a goal, succeeded only if its argument failed, and failed otherwise. Thus if the argument to THNOT was a formula to be proved, the THNOT would succeed only if the attempt to prove the embedded formula failed. In addition to the non-monotonic primitive THNOT, PLANNER employed antecedent and erasing procedures to update the data base of statements of beliefs when new deductions were made or actions taken. Unfortunately, it was up to the user of these procedures to make sure that there were no circular dependencies or mutual proofs between beliefs. Such circularities could lead to, for example, errors of groundless belief (due to two mutually supporting beliefs) or non-terminating programs (a more technical but no less irritating problem).

Two related forms of non-monotonic deductive systems are those described by McCarthy and Hayes [24] and Sandewall [39]. McCarthy and Hayes give some indications of how actions might be described using modal operators like 'normally' and 'consistent,' but present no detailed guidelines on how such operators might be carefully defined. Sandewall, in a deductive system applied to the frame problem (which is basically the problem of efficiently representing the effects of actions; see [11]) used a deductive representation of non-monotonic rules based on a primitive called UNLESS. This was used to deduce conditions of situations resulting from actions except in those cases where properties of the action changed the extant conditions. Thus one might say that things retain their color unless painted.

Sandewall's interpretation of UNLESS was in accord with then current intuitions: UNLESS (p) is true if p is not deducible from the axioms using the classical first-order inference rules. Unfortunately, this definition has several problems, as pointed out by Sandewall. One problem is that it can happen that both p and UNLESS (p) are deducible, since from a rule like "from UNLESS(C) infer D," D can be inferred, but at the same time UNLESS(D) is also deducible since D is not deducible by classical rules. These problems are partly due to the dependence of the notion of 'deducible' on the intention of deduction rules based on 'not deducible.' This question-begging definition leads to perplexing questions of beliefs when complicated relations between UNLESS statements are present. For example, given the axioms

$$\begin{aligned}&A\\ &A \wedge \mathrm{UNLESS}(B) \supset C\\ &A \wedge \mathrm{UNLESS}(C) \supset B,\end{aligned}$$

we are faced with the somewhat paradoxical situation that either B or C can be deduced, but not both simultaneously. On the other hand, in the axiom system

$$\begin{aligned}&A\\ &A \wedge \mathrm{UNLESS}(B) \supset C\\ &A \wedge \mathrm{UNLESS}(C) \supset D\\ &A \wedge \mathrm{UNLESS}(D) \supset E,\end{aligned}$$

one would expect to see A, C and E believed, and B and D not believed.

One might be tempted to dismiss these anomalous cases as uninteresting. In fact, such cases are not perverse; rather, they occur naturally in many applications. The common way they are introduced is by employing assumptions which require further assumptions to be made.

Spurred by Sandewall's presentation of the problems arising through such non-monotonic inference rules, Kramosil [17] considered sets of inference rules of the form

"From $\vdash p$, $\nvdash q$, infer $\vdash r$",

where $\vdash$ and $\nvdash$ are tokens of the meta-language and the number of antecedents can be arbitrary. Kramosil defined the set of theorems in such a system as the intersection of all subsets of the language closed under the inference rules. He noted that this set may not itself be closed under the inference rules, and showed that in the special case in which the inference rules preserve truth values (that is, are effectively monotonic) that if the set of theorems of the monotonic inference rules alone is also closed with respect to the non-monotonic inference rules, then this set is the set of non-monotonic theorems. Kramosil's conclusion was that a set of inference rules defines a formalized theory (one in which all formulas have a well-defined truth value) if and only if this same theory is that of the monotonic inference rules alone, which he interprets to mean that the non-monotonic rules are either useless or meaningless.

As we will show in this paper, Kramosil's interpretation was too pessimistic with regard to the possibility of formalizing such rules and their unusual properties. As we have argued above, the purpose of non-monotonic inference rules is not to add certain knowledge where there is none, but rather to guide the selection of tentatively held beliefs in the hope that fruitful investigations and good guesses will result. This means that one should not *a priori* expect non-monotonic rules to derive valid conclusions independent of the monotonic rules. Rather one should expect to be led to a set of beliefs which while perhaps eventually shown incorrect will meanwhile coherently guide investigations.

Non-monotonic inference rules need not appear in the explicit forms discussed by Kramosil. Many authors have described artificial intelligence programs which exhibit non-monotonic behavior only implicitly. Non-monotonicity in these systems stems typically from extra-logical devices. For example, one can effect changes in beliefs (as opposed to inferences) by using extra-logical STRIPS-like [8] operators.

Conflict resolution strategies, which use production-rule orderings and specificity criteria to determine the next system action, also induce non-monotonic behavior. ([15] and [31] in fact term this property of their systems 'non-monotonicity'.)

One class of non-monotonic inferences consists of what might be called 'minimal' inferences, in which a minimal model for some set of beliefs is assumed by assuming the set of beliefs to be a complete description of a state of affairs. Joshi and Rosenschein [16] describe a partial-matching procedure based on the operation of taking least upper bounds in a lattice of sets of beliefs. This has the effect of assuming just enough additional information to allow a desired partial match to succeed. McCarthy [25] outlines a procedure called 'circumscription', in which the current partial extension of some predicate is assumed to be the complete extension. Of course, new examples of the predication can invalidate previous completeness assumptions. Reiter [34] analyzes the related technique of assuming false all elementary predications not explicitly known true. He outlines some conditions under which data bases remain consistent under this 'closed world assumption,' and shows certain forms of data bases to be naturally consistent with this assumption. However, the closed world assumption does not seem to allow for any locality of definition of defaults, since it applies this assumption to all primitive predicates, and does not allow defaults applied to defined predicates. Circumscription, on the other hand, would seem to be applicable to any predicate whatever. Although they describe tools for non-monotonic reasoning, none of these authors discuss the problem of revision of beliefs.

These problems were mostly resolved in the Truth Maintenance System (TMS) of Doyle [6] and related systems [21, 23] in which each statement has an associated set of justifications, each of which represents a reason for holding the statement as a belief. These justifications are used to determine the set of current beliefs by examining the recorded justifications to find well-founded support (non-circular proofs) whenever possible for each belief. When hypotheses change, these justifications are again examined to update the set of current beliefs. This scheme provides a more accurate version of antecedent and erasing procedures of PLANNER without the need to explicitly check for circular proofs. The non-monotonic capability appears as a type of justification which is the static analogue of the PLANNER THNOT primitive. Part of the justification of a belief can be the lack of valid justifications for some other possible program belief. This allows, for example, belief in a statement to be justified whenever no proof of the negation of the statement is known. This representation of non-monotonic justifications, in combination with the belief revision algorithms, produced the first system capable of performing the routine revision of apparently inconsistent theories into consistent theories. Part of this revision process is a backtracking scheme called dependency-directed backtracking [42]. We will analyze this system in more detail in Section 10, but first we provide some theoretical foundations for this work.

Weyhrauch [46] presents a formal system organized about a theory/metatheory distinction in which non-monotonic rules (among many others) can be expressed. This system also makes the self-referential nature of such rules clear by allowing explicit self-reference, so that the formal system has a name for itself, and can draw conclusions about itself in the theory of itself taken as an object. However, the system is its own model of itself, so these conclusions can change the referent of its name for itself. This means that new derivations can affect old conclusions drawn from previous observations of the system's state. Much work remains to be done along these lines, one detail of which would be a formal comparison of Weyhrauch's system with non-monotonic logic.

In outline, our analysis of non-monotonic logic will proceed as follows. We first define a standard language of discourse including the non-monotonic modality M ('consistent'). The semantics of the language is based on models constructed from *fixed points* of a formalized non-monotonic proof operator. Provability in this system is then defined, and a proof of completeness for this system is presented. This is augmented by a proof procedure for a restricted class of theories and an analysis of some of the structure of models of non-monotonic theories.

4. Linguistic Preliminaries

We settle on a language $\mathscr{L}$ which will be the language of all theories mentioned in the following. $\mathscr{L}$ has an infinite number of constant letters, variable letters, predicate letters, and propositional constant letters. The formation rules of the language are as follows:

The *atomic formulas* of $\mathscr{L}$ are the propositional constant letters and the strings of the form $g(x_1, \ldots, x_n)$ for predicate letter g and variables or constants $x_1, \ldots, x_n$. The *formulas* of $\mathscr{L}$ are either atomic formulas or, for formulas p, q and variable letter x, strings of the form $\mathsf{M}p$, $\neg p$, $p \supset q$, and $\forall xp$. We use the usual abbreviations of $p \wedge q$ for $\neg[p \supset \neg q]$, $p \vee q$ for $\neg p \supset q$, $\exists xp$ for $\neg\forall x\neg p$, and abbreviate $\neg\,\mathsf{M}\neg p$ as $\mathsf{L}p$. A *statement* is a formula with no free variables. The usual criteria for determining free variables apply (see [27]). In addition, a variable x is free in $\mathsf{M}p$ if and only if x is free in p.

In this paper, the letters C, D, E and F will be used as syntactic variables ranging over propositional constant letters. The letters p, q and r will be used for formulas. Implicit quasi-quotation is used throughout. That is, if p and q are formulas, $p \supset q$ is the formula obtained by concatenating p, the implication symbol and q. This notation extends to handle finite sets of formulas in the following way: if Q is a finite set of formulas, and Q appears in a quasi-quoted context, it always stands for the conjunction of its elements. For example, $Q \supset p$ means the formula obtained by conjoining all the elements of Q and following the result with the implication symbol and p. (If Q is empty, it stands for $C \vee \neg C$). Since syntax is not a preoccupation of this paper, the presentation is not rigorous in specifying the number of arguments of predicate letters, parenthesization, etc.

The inferential system used defines a first-order theory to be a set of axioms, including the following infinite class of axioms:

For all formulas p, q and r:

(6) (i) $p \supset [q \supset p]$

(ii) $[p \supset [q \supset r]] \supset [[p \supset q] \supset [q \supset r]]$

(iii) $[\neg q \supset \neg p] \supset [[\neg q \supset p] \supset q]$

(iv) $\forall x p(x) \supset p(t)$

where $p(x)$ is a formula and t is a constant or a variable free for x in $p(x)$ and $p(t)$ denotes the result of substituting t for every free occurrence of x in $p(x)$, and

(v) $\forall x[p \supset q] \supset [p \supset \forall x q]$

if p is a formula containing no free occurrence of x. (These axioms are from [27].) These are the *logical* axioms. All other axioms are called *proper*, or non-logical axioms. (This terminology is misleading for axioms which are logical consequences of the logical axioms, but we will ignore this inelegance.) The theory with no proper axioms is called the *predicate calculus* (PC). (Note that this theory also contains strings containing the letter M, so it is actually not strict PC.) The *sentential calculus* (SC) consists of axioms which are instances of (i), (ii) and (iii) only. A theory consisting only of the sentential calculus plus a finite number of statements is called a statement theory.

In this paper, the letters A and B will be used to stand for theories.

5. Proof-Theoretic Operators

The monotonic rules of inference we will use (also from [27]) are

(7) Modus Ponens: from p and $p \supset q$, infer q
Generalization: from p, infer $\forall x p$.

If S is a set of formulas, and p follows from S and the axioms of A by the rules (7), we say $S \vdash_A p$. We abbreviate $\vdash_{\mathrm{PC}}$ by $\vdash$ alone. We define $\mathrm{Th}(S) = \{p: S \vdash p\}$.

The particular inference rules (7) are not very important. Later in the paper, when we concentrate on statement theories, the rule of generalization will be dropped without much fanfare. All that is important is that the operator Th have the following properties, which together are called *monotonicity*:

(8) (i) $A \subseteq \mathrm{Th}(A)$

(ii) If $A \subseteq B$, then $\mathrm{Th}(A) \subseteq \mathrm{Th}(B)$,

and the property (9) of *idempotence*

(9) $\mathrm{Th}(\mathrm{Th}(A)) = \mathrm{Th}(A)$.

Clearly, any classical inference system satisfies these conditions. Condition (9) can also be viewed as a fixed point equation, stating that the set of theorems monotonically derivable from a theory is a fixed point of the operator which computes the closure of a set of formulas under the monotonic inference rules. A well-known property of the monotonic inference rules is that $\mathrm{Th}(A)$ is the smallest fixed point of this closing process; in fact, that $\mathrm{Th}(A)$ is the intersection of all S such that $A \subseteq S$ and $\mathrm{Th}(S) = S$.

In order to deal with non-monotonic logic, we need a new inference rule like this one (which we will take back immediately):

(10) "If $\nvdash_A \neg p$, then $\vdash_A \mathrm{M}p$."

That is, if a formula's negation is not derivable, it may be inferred to be consistent. As it stands, however, this rule is of no value because it is circular. 'Derivable' means 'derivable from axioms by inference rules', so we cannot define an inference rule in terms of derivability so casually.

Instead, we retain the definition of $\vdash$ as meaning monotonic derivability, and define the operator NM as follows: for any first-order theory A and any set of formulas $S \subseteq \mathscr{L}$ ($\mathscr{L}$, recall, is the entire language), let

(11) $\mathrm{NM}_A(S) = \mathrm{Th}(A \cup \mathrm{As}_A(S))$,

where $\mathrm{As}_A(S)$, the set of *assumptions* from S, is given by

(12) $\mathrm{As}_A(S) = \{\mathrm{M}q: q \in \mathscr{L} \text{ and } \neg q \notin S\} - \mathrm{Th}(A)$.

Notice that theorems of A of the form $\mathrm{M}q$ are never counted as assumptions. NM_A takes a set S and produces a new set which includes $\mathrm{Th}(A)$ but also includes much more: everything provable from the enlarged set of axioms and assumptions which is the original theory together with all assumptions not ruled out by S.

We would like to define $\mathrm{TH}(A)$, the set of theorems non-monotonically derivable from A, by analogy with the monotonic case as

(13) "$\mathrm{TH}(A)$ = the smallest fixed point of NM_A."

This 'definition' tries to capture the idea of adding the non-monotonic inference rule (10) to a first-order theory A. This is plausible, since it demands a set such that all of its elements may be proven from axioms and assumptions not wiped out by the proofs. Such fixed points $S = \mathrm{NM}_A(S)$ might be depicted as in Fig. 1. Unfortunately, there is in general no appropriate fixed point of NM_A. It can happen that a theory has no fixed point under the operator NM_A. Even if there are fixed points, there need not be a smallest fixed point.

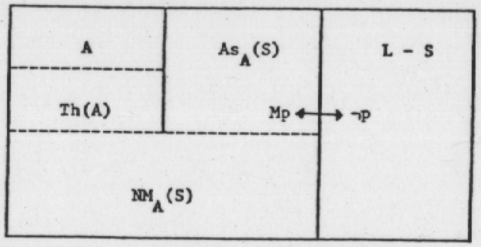

Fig. 1. The graphical conventions are intended to suggest these facts:
$A \subseteq \mathrm{Th}(A)$, $(\mathrm{Th}(A) \cup \mathrm{As}_A(S)) \subseteq \mathrm{NM}_A(S)$, $\mathrm{Th}(A) \cap \mathrm{As}_A(S) = \varnothing$, $(\mathscr{L} \cap S) - \mathrm{NM}_A(S) = \varnothing$.

For example, consider the theory T1 obtained as

(14) $\mathrm{T1} = \mathrm{PC} \cup \{\, \mathsf{M}C \supset \neg D,\ \mathsf{M}D \supset \neg C\}$,

where C and D are propositional constants. $\mathrm{NM}_{\mathrm{T1}}$ has two fixed points, which can be called F_1 and F_2. F_1 contains $\neg C$ but not $\neg D$, and F_2 contains $\neg D$ but not $\neg C$. Since $\neg D$ is not in F_1, $\mathsf{M}D$ is in F_1, and so $\neg C$ is in F_1. Similarly, the presence of $\neg D$ in F_2 keeps $\neg C$ out and $\mathsf{M}C$ in F_2. The problem is that neither $F_1 \cap F_2$ nor $F_1 \cup F_2$ is a fixed point of $\mathrm{NM}_{\mathrm{T1}}$. Since neither $\neg C$ nor $\neg D$ is in $\mathrm{F}_1 \cap \mathrm{F}_2$, $\mathsf{M}C$ and $\mathsf{M}D$ are both in $\mathrm{NM}_{\mathrm{T1}}(\mathrm{F}_1 \cap \mathrm{F}_2)$, so $\neg C$ and $\neg D$ are in $\mathrm{NM}_{\mathrm{T1}}(\mathrm{F}_1 \cap \mathrm{F}_2)$, so $\mathrm{F}_1 \cap \mathrm{F}_2 \neq \mathrm{NM}_{\mathrm{T1}}(\mathrm{F}_1 \cap \mathrm{F}_2)$. Similarly, both $\neg C$ and $\neg D$ are in $\mathrm{NM}_{\mathrm{T1}}(\mathrm{F}_1 \cup \mathrm{F}_2)$, so applying $\mathrm{NM}_{\mathrm{T1}}$ to the union results in a smaller set. So in this case there is no natural status for $\neg C$ and $\neg D$.

An example of a theory with no fixed point of the corresponding operator is the theory T2 obtained as

(15) $\mathrm{T2} = \mathrm{PC} \cup \{\, \mathsf{M}C \supset \neg C\}$.

In this case, $\mathrm{NM}_{\mathrm{T2}}$ has no fixed point, since alternate applications of the operator to any set produce new sets in which either both $\mathsf{M}C$ and $\neg C$ exist or neither exist.

Therefore, we must accept a somewhat less elegant definition of TH. Let us define TH as follows:

(16) $\mathrm{TH}(A) = \bigcap(\{\mathscr{L}\} \cup \{S\colon \mathrm{NM}_A(S) = S\})$.

That is, the set of provable formulas is the intersection of all fixed points of NM_A, or the entire language if there are no fixed points. We will use the abbreviation $A \mid\!\sim p$ to indicate that $p \in \mathrm{TH}(A)$. With this definition, neither $\mathsf{M}C$ nor $\mathsf{M}D$ is a theorem of T1 in (14), but $\mathsf{M}C \vee \mathsf{M}D$ is. In the following, we will abbreviate $\{S\colon \mathrm{NM}_A(S) = S\}$ as $\mathrm{FP}(A)$, and (somewhat abusing the terms) call the elements of this set *fixed points* of the theory A.

This definition of the provable statements is quite similar in some respects to the definition of compatibility-restricted entailment given by Rescher [36]. In that system, a set S of formulas is said to CR-entail a formula p if p follows in the standard fashion from each of one or more 'preferred' maximal consistent subsets of S. In the present case, we obtain the preferred subsets of formulas as fixed points of the operator NM_A (the 'compatible subsets'), but in contrast to normal deducibility where the empty set always suffices, there need not be any such subsets. This case produces the entire language as the set of provable formulas by vacuous fulfilment of the condition of derivability.

One unusual consequence of this definition of provability is that the deduction theorem does not hold for non-monotonic logic. For example, while $\{C\} \mid\!\sim \mathsf{M}\mathsf{L}, C$ it is not true that $\mid\!\sim C \supset \mathsf{M}\mathsf{L}C$. This failure of the deduction theorem is to be expected, however, since the non-monotonic provability of a formula depends on the completeness of the set of hypotheses, that is, on the fact that no other axioms are available. The deduction theorem, however, would if valid produce implications valid no matter what other axioms were added to the system, even if these axioms would invalidate the completeness condition used in the derivation of the implication. One should note that although the deduction theorem does not hold in general in non-monotonic logic, there are many particular cases in which it does hold. For instance, if some conclusion follows classically from some hypotheses, then the expected implication will also hold. In addition, not all properly non-monotonic theories are such that the deduction theorem fails. It is an interesting open problem to characterize the precise cases in which the deduction theorem is valid in non-monotonic theories.

So far, we have defined 'provability' without defining 'proof'. For a formula to be provable in a theory, it must have a standard proof from axioms and assumptions in each fixed point of the theory, and, as yet, we have no way of enumerating fixed points or even of describing one. It is worth note that when a theory has more than one fixed point, the fixed points are inaccessible in the sense that the sequence $\mathrm{Th}(A)$, $\mathrm{NM}_A(\mathrm{Th}(A))$, $\mathrm{NM}_A(\mathrm{NM}_A(\mathrm{Th}(A)))$, ... does not converge to a fixed point. We have a proof, which we do not present here, that if NM_A has exactly one fixed point, then the fixed point is the limit of successive applications of NM_A to the sequence of sets starting with A. We will eventually attend to defining non-monotonic proof, but first we turn our attention to the topic of semantics.

6. Model Theory

The semantics of non-monotonic logic is built on the notion of model, just like the semantics of classical logic. In fact, the definition of model for a non-monotonic theory depends directly on the usual definition.

An *interpretation* V of formulas over a language $\mathscr{L}$ is a pair $\langle X, U\rangle$, where X is a nonempty set, and U is a function which associates relations and values over the domain X with each predicate, variable, constant and propositional constant letter in the usual fashion. That is, for each n-ary predicate letter P, $U(P) \subseteq X^n$; for each variable or constant x, $U(x) \in X$; and for each propositional constant letter, C, $U(C) \in \{0, 1\}$. Using this mapping function U we define the value $V(p)$ of a formula p in the interpretation V to be an element of $\{0, 1\}$ satisfying the following conditions: For an atomic formula $p(x_1, \ldots, x_n)$, the value is 1 if $\langle U(x_1), \ldots, U(x_n)\rangle \in U(p)$, and is 0 otherwise. $V(\neg p) = 1$ if $V(p) = 0$, and is 0 otherwise. $V(p \supset q) = 1$ if either $V(p) = 0$ or $V(q) = 1$, and is 0 otherwise. $V(\forall xp) = 1$ if for all $y \in X$, $V'(p) = 1$, where $V' = \langle X, [y/x]U\rangle$, where $[y/x]U$ is the mapping derived from U by changing its value at the point x to the value y. $V(\forall xp) = 0$ otherwise. If $V(p) = 1$, we say that V *satisfies* p, and write $V \models p$.

A *monotonic model* of a set of formulas $S \subseteq \mathscr{L}$ is an interpretation V which satisfies each formula in S, that is, $V(p) = 1$ for each formula $p \in S$. A *non-monotonic model* of a theory A is a pair $\langle V, S\rangle$, where V is a monotonic model of S, and $S \in \mathrm{FP}(A)$. When the context makes the intended meaning clear, we will use the term *model* of A to mean either a non-monotonic model, a monotonic model, or an element of $\mathrm{FP}(A)$ for the theory A.

Although unorthodox, this definition provides a meaning for formulas $\mathsf{M}p$ which reflects the proof-theoretic property that 'p is consistent with what is believed.' This notion is made precise by including in the model a set of 'current assumptions' (namely, $\mathrm{As}_A(S)$). A model for a theory must assign 1 to all of these assumptions, so the effect is that $\mathsf{M}p$ is assigned 1 in a model if $\neg p$ is not derivable and $\neg\, \mathsf{M}p$ is not derivable from the current assumptions and the original theory, that is, if p is consistent with what is 'believed' in the model. Unfortunately, $\mathsf{M}p$ may be assigned 1 in some model even when $\neg p$ is derivable (for example, when no axiom mentions $\mathsf{M}p$ at all). This indicates that the logic is too weak.

We view the weakness of this semantics as the worst defect of our logic. We think we can improve it, but it appears that the semantics of non-monotonic systems will always be somewhat unorthodox. The problem is the failure of a kind of locality in semantic definitions. In conventional logics, the meaning of a composite expression can always be defined in terms of the meanings of its parts, even if these meanings are infinite things like functions on sets of possible worlds. In non-monotonic logic, this is impossible. The meaning of $\mathsf{M}p$ can never be purely a function of the meaning of p, because it depends on the available axioms as well. This 'holistic' property seems inescapable. The best we can do is to try to express this dependence semantically, in terms of models of p plus the axioms. In the semantics just presented, this is not achieved, but we have hope that it can be achieved for a slightly stronger logic.

With this definition of model, we can justify the definition of provability.

Theorem 1 (Soundness). *If $A \mid\!\sim p$, then $V \vDash p$ for all models $\langle V, S\rangle$ of A.*

Proof. Assume $A \mid\!\sim p$. If there are no models of A, the theorem follows trivially. Otherwise, p is a member of every fixed point of A. But since every model of A is a monotonic model of a fixed point of A, every model assigns 1 to p.

Theorem 2 (Completeness). *If $V \vDash p$ for all models $\langle V, S\rangle$ of A, then $A \mid\!\sim p$.*

Proof. Assume that it is not true that $A \mid\!\sim p$. Thus there is a fixed point S of NM_A which does not contain p. Now $\mathrm{Th}(S) = S$ by idempotence, so $S \nvdash p$. But the predicate calculus is complete, so some monotonic model V of S has $V(p) = 0$.

It is not surprising that we have completeness, since the definition of truth makes reference to provability. The proof was for first-order theories, but it can easily be generalized to any complete formal logic. For example, if we take care not to confuse M with the S5 operator 'possibly,' we can easily get a complete non-monotonic extension of S5. However, none of these observations are very interesting unless we have some assurance that provability is decidable. We will shortly present a proof procedure for non-monotonic statement theories.

7. Fixed Points of Theories

This section will try to analyze the structure of fixed points for non-monotonic theories. We investigate the number of fixed points of theories, and their relation to the provable statements.

Non-monotonic theories may have varying number of fixed points. Classically inconsistent theories have just one fixed point (the entire language $\mathscr{L}$) and thus no models. The theory T2 in (15) also has no models due to the lack of a fixed point. Theories formulated in strictly classical language have exactly one fixed point, as does the theory

(17) $\mathrm{T3} = \mathrm{PC} \cup \{\, \mathsf{M}C \supset C\}$.

Some theories have several fixed points, e.g. T1 in (14). It is also possible for a theory to have an infinite number of fixed points. This is exemplified (we assume equality and an infinite domain of unequal constants) by

(18) $\mathrm{T4} = \mathrm{PC} \cup \{\forall x[\, \mathsf{M}p(x) \supset [p(x) \wedge \forall y[x \neq y \supset \neg p(y)]]]\}$.

Even in theories having only one fixed point, the non-monotonically provable statements need not coincide with the classically provable statements. Theory T3 above is an example, for $C \in \mathrm{TH(T3)}$, but $C \notin \mathrm{Th(T3)}$. Some statements will be provable in theories with multiple fixed points, but will have different proofs in each fixed point. For example, $\mathsf{M}C \vee \mathsf{M}D \in \mathrm{TH(T1)}$ and $\exists x\, \mathsf{M}p(x) \in \mathrm{TH(T4)}$.

The classical results concerning truth and provability for logical languages are that, for a given theory A, a formula is *valid* in A (true in all models of A) if and only if it is *provable* in A, and that the theory has a model if and only if it is *consistent* (cannot be used to derive a contradiction). In non-monotonic logic, somewhat different circumstances obtain. As Theorems 1 and 2 have shown, validity in a theory remains equivalent to provability. However, from the definition of models of non-monotonic theories, it follows that a non-monotonic theory A has a model only if the operator NM_A has a classically consistent fixed point. Non-monotonic theories can lack fixed points (e.g. the theory T1), but we have defined such theories to be inconsistent.

The basic structure theorem states that all fixed points of a non-monotonic theory A are (set inclusion-) minimal fixed points.

Theorem 3. *If $S_1, S_2 \in \mathrm{FP}(A)$ and $S_1 \subseteq S_2$, then $S_1 = S_2$.*

Proof. If $S_1 \subseteq S_2$, then $\mathrm{As}_A(S_2) \subseteq \mathrm{As}_A(S_1)$, so by the monotonicity of Th, $\mathrm{NM}_A(S_2) \subseteq \mathrm{NM}_A(S_1)$. But since S_1 and S_2 are fixed points of this operator, $S_2 \subseteq S_1$, so $S_1 = S_2$.

This result suggests that strict set-theoretic minimality is not a particularly interesting distinction among fixed points. In the following sections we will make steps towards more interesting classifications, but without a fully satisfactory solution. Important applications of this theorem are the following two corollaries.

Corollary 4. *If $\mathscr{L}$ is a fixed point of A, then it is the only fixed point of A.*

Proof. If $S \in \mathrm{FP}(A)$, then $S \subseteq \mathscr{L}$, so $S = \mathscr{L}$ by Theorem 3.

Note that if $\mathscr{L}$ is a fixed point of A, then A is classically inconsistent, that is, $\mathrm{Th}(A) = \mathscr{L}$.

Corollary 5. *If p, $\neg p \in \mathrm{TH}(A)$, then* $\mathrm{TH}(A) = \mathscr{L}$.

Proof. If A has no fixed points, the theorem follows by definition. If both p and $\neg p$ are members of a fixed point S of A, then since fixed points are closed under monotonic deduction, $S = \mathscr{L}$. But then $\mathrm{FP}(A) = \{\mathscr{L}\}$, so $\mathrm{TH}(A) = \mathscr{L}$.

With these results, we can study the notion dual to provability in non-monotonic theories. We say that a formula p is *arguable* from A if $p \in \cup\, \mathrm{FP}(A)$, that is, if some fixed point of A contains p. Clearly, all provable formulas are arguable. Our next theorem shows that in consistent theories, provability and arguability are almost dual notions.

Theorem 6. *If A is consistent and p is provable in A, then $\neg p$ is not arguable.*

Proof. If p is provable in a consistent theory A, then any $S \in \mathrm{FP}(A)$ containing $\neg p$ would be inconsistent, which is impossible by Corollary 4.

Unfortunately, the converse of this theorem is not true. For example, in the theory with no proper axioms, $\neg C$ is not arguable, but C is not provable. We will term the notion dual to provability *conceivability*, so that C is conceivable if it is not provably false. Thus all arguable formulas are conceivable, but not vice versa. We say *doubtless* p if and only if $\neg p$ is not arguable. In PC, C is doubtless yet not arguable, and in the theory

(19) $\mathrm{T5} = \mathrm{PC} \cup \{ \mathsf{M}C \supset C,\ \mathsf{M}\neg C \supset \neg C\}$

C is arguable yet not doubtless. Fig. 2 summarizes these classifications.

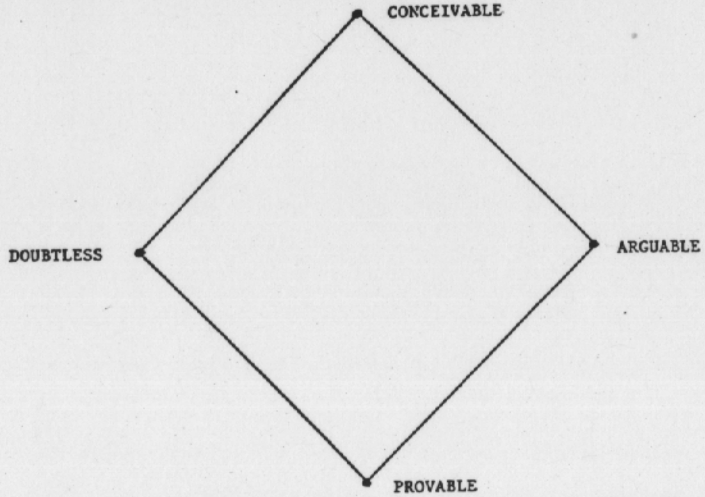

Fig. 2. Diagram of four classifications of formulas. Each line connects a set of formulas with a superset. Incomparable sets are not connected in this way.

It is worthy of note that the provable and arguable statements of a consistent theory cannot be classified as the monotonic theorems of the theory augmented by some set of assumptions. That is, the set of arguable statements may be inconsistent yet not sum to the entire language $\mathscr{L}$, and the set of provable statements may involve assumptions that vary from fixed point to fixed point, as in the theory T2 above, where neither the assumption $\mathsf{M}C$ nor the assumption $\mathsf{M}D$ is present in both fixed points.

Another natural classification is that of 'decision'. We say that p is *decided* by a consistent theory A if and only if for all $S \in \mathrm{FP}(A)$, either $p \in S$ or $\neg p \in S$. The dual to this notion is just its negation. In this case we say that A is *ambivalent* about p if p is not decided by A.

Corollary 7. *If p is doubtless yet decided by A, p is provable.*

Proof. For each $S \in \mathrm{FP}(A)$, either $p \in S$ or $\neg p \in S$; yet $\neg p \notin S$, so $p \in S$.

8. The Evolution of Theories

We now turn to analyzing inter-theory relationships. These are important in describing the effects of incremental changes in the set of axioms, and this is the task of practical systems like the TMS [6], which has the task of maintaining a description of a model of a changing set of axioms. As we shall see, there are many unusual phenomena which occur when theories change. The most striking result shows that the analogue of the compactness theorem of classical model theory does not hold for non-monotonic theories. This has important repercussions on the methods useful in constructing 'models' of theories incrementally.

Theorem 8. *There exists a consistent theory with an inconsistent subtheory.*

Proof. Consider the consistent theory

(20) $\mathrm{T6} = \mathrm{PC} \cup \{ \mathsf{M}C \supset \neg C, \neg C\}$.

The subtheory $\mathrm{PC} \cup \{ \mathsf{M}C \supset \neg C\}$ is inconsistent.

Note, however, that the theory T6 in (20) has as a thesis the formula $\neg\, \mathsf{M}C$, which makes it quite different than some previously considered theories. We will comment further on this type of theory in Section 11.

In many cases, the changes in fixed points induced by changes in theories is less drastic than those apparent in the previous theorem. The simplest cases are as follows.

Theorem 9. *If A is consistent, and p is arguable in A, then $A' = A \cup \{p\}$ is consistent, and $\mathrm{FP}(A') \cap \mathrm{FP}(A) \neq \varnothing$.*

Proof. Since p is arguable, there is some $S \in \mathrm{FP}(A)$ such that $p \in S$. But clearly, S is then also a fixed point $\mathrm{NM}_{A'}$.

FIG. 2. Diagram of four classifications of formulas. ...
superset. Incomparable sets are not connected in this way.

is then also a fixed point $NM_{A'}$.

Unfortunately, this theorem cannot be strengthened to conclude that FP(A') is contained in FP(A), since in the theory

$$(21)\quad T7 = PC \cup \{ MC \supset \neg D,\ MD \supset \neg C, \neg C \supset E\}$$

there are two fixed points, call them F_1 and F_2, with $\neg C \in F_1$, $E \in F_1$ and $\neg D \in F_2$, $E \notin F_2$. Extending this theory by adding the axiom E produces a theory also with two fixed points, one of which is F_1. but the other fixed point F_3 differs from F_2 in that $E \in F_3$ and $M\neg E \notin F_3$.

Theorem 10. *If A and $A' = A \cup \{p\}$ are consistent and* $FP(A) \cap FP(A') \neq \varnothing$, *then p is arguable in A.*

Proof. Since $p \in A'$, $p \in S$ for every $S \in FP(A')$. Thus $p \in S$ for some $S \in FP(A)$.

Theorem 11. *If A and $A' = A \cup \{p\}$ are consistent, then p is provable in A if and only if* $FP(A') = FP(A)$.

Proof. If p is provable in A, $p \in S$ for every $S \in FP(A)$, so each member of FP(A) is also a member of FP(A'). If $FP(A) = FP(A')$, then since $p \in S$ for each $S \in FP(A')$, $p \in S$ for each $S \in FP(A)$, so p is provable in A.

The import of these theorems is that if a new axiom is already implicit in the current axioms, either no change of fixed point is necessary, or a simple shift to a different fixed point of the previous axioms is allowable. When considering changes which delete axioms from theories, the basic problem is the non-compactness result mentioned above. Other interesting questions are of the form 'how few axioms must be added or removed to remove p.' Answers to these questions will in general depend on the specific theory in question.

Another important phenomenon is the 'hierarchy of assumptions' [6], in which some non-monotonic choices depend on others. This manifests itself in terms of fixed points as the addition of new axioms increasing the number of fixed points of the theory. For example, adding the axiom E to the theory

$$(22)\quad T8 = PC \cup \{[E \wedge MC] \supset \neg D, [E \wedge MD] \supset \neg C\}$$

increases the number of fixed points from one to two. In this case, E can be interpreted as the reason for choosing between $\neg C$ and $\neg D$. Since E itself might have been an assumption, there might therefore be a hierarchical structure of assumptions in each fixed point.

To get a global view of theory evolution, we consider the set of all consistent theories containing a consistent theory A as a subtheory. For a formula p, we can consider the evolution of the properties of p of being arguable, provable, or decided over sequences of extensions of the theory A. The evolution of arguability is mainly a question of control structures; this is the point of the encoding of control primitives in non-monotonic dependency relationships given by Doyle [6]. We have at present no way of describing the evolution of decision. However, analysis of the relationships between the theories and their extensions will shed light on how our semantics for Mp matches the intuitive notion of 'p can be added consistently to the theory.'

We say that p is *assumable* in a consistent theory A if the theory $A \cup \{p\}$ is also consistent. We name the dual notion by saying that p is *uncontroversial* in a theory if $\neg p$ is not assumable in the theory. The matching of the semantics of non-monotonic logic with this more standard notion of consistency will be apparent upon examining the correlation between assumability of p and the arguability of Mp in a theory, since this latter condition would seem to say there is a coherent interpretation of the axioms in which p is consistent. Our logic is weak, however, and so this correlation is weak. As an approximation, we note that Mp is arguable if p is arguable, and so instead attempt to correlate arguability of p with assumability of p. This correlation is as follows. By Theorem 9 the assumable formulas include the arguable formulas, but not vice versa since C is assumable but not arguable in PC. The assumable formulas are incomparable with the conceivable formulas, since C is conceivable but not assumable in

$$(23)\quad T9 = PC \cup \{C \supset [D \wedge [MD \supset \neg D]]\},$$

and $\neg C$ is assumable but not conceivable in the theory T3 of (17). Also, the assumable formulas are incomparable with the uncontroversial formulas, since C is assumable but not uncontroversial in PC, and C is uncontroversial but not assumable in

$$(24)\quad T10 = PC \cup \{C \supset [D \wedge [MD \supset \neg D]], \neg C \supset [E \wedge [ME \supset \neg E]]\}.$$

We specify another classification by saying that a formula p is *safe* in a consistent theory A if and only if $p \in TH(A')$ for all consistent A' such that $A \subseteq A'$, and that p is *forseeable* if and only if $\neg p$ is not safe. Let $\text{Safe}(A) = \{p: p \text{ is safe in } A\}$. We then can characterize the set Safe(A) as follows.

Theorem 12. *If A is consistent, then* Safe(A) *is the least set such that the following three conditions hold:*

(i) $A \subseteq \text{Safe}(A)$
(ii) $\text{Th}(\text{Safe}(A)) = \text{Safe}(A)$
(iii) *If* $p \in \text{Safe}(A)$, *then* $Mp \in \text{Safe}(A)$.

Proof. The first two cases are correct because all formulas classically deducible from safe formulas (in particular the axioms) will remain classically deducible when the set of axioms is enlarged. The case of interest is (iii), which declares that 'covered' assumptions are safe. That is, if $p \in \text{Safe}(A)$, then $\neg p$ cannot be a member of any consistent extension of A, so Mp will be a member of every consistent extension; thus Mp is safe.

It is clear that all safe formulas are both assumable and uncontroversial, and that these inclusions are proper. Elementary considerations show further that the forseeable formulas include the assumable and uncontroversial formulas, but again, not vice versa. Also, the provable formulas properly include the safe formulas with theory T3 in (17) as the example, and the forseeable formulas properly include the conceivable formulas via the same example.

A weakened version of assumability is produced by saying that p is *realizable* in a consistent theory A if there is some consistent theory A' such that $A \subseteq A'$ and $p \in A'$. We also say that p is *undeniable* if and only if $\neg p$ is not realizable. Clearly, the realizable formulas include the assumable formulas, but the converse does not hold as $\mathsf{M}C \supset \neg C$ is not assumable in PC but is an axiom of the consistent theory T6 in (20). The forseeable formulas obviously include the realizable formulas, but not vice versa since C is forseeable but not realizable in the theory T9 of (23). Also, the realizable formulas are incomparable with the conceivable formulas, since C is conceivable but not realizable in T9 of (23), and $\neg C$ is realizable but not conceivable in T3 of (17). The example of T10 in (24) provides an example of what following Kripke might be called the *paradoxical* formulas of a theory, formulas (in this case C) such that neither they nor their negations are realizable. The example of T9 in (23) provides an example of what might be called the *intrinsic* formulas of a theory, formulas (in this case $\neg C$) which are realizable and undeniable.

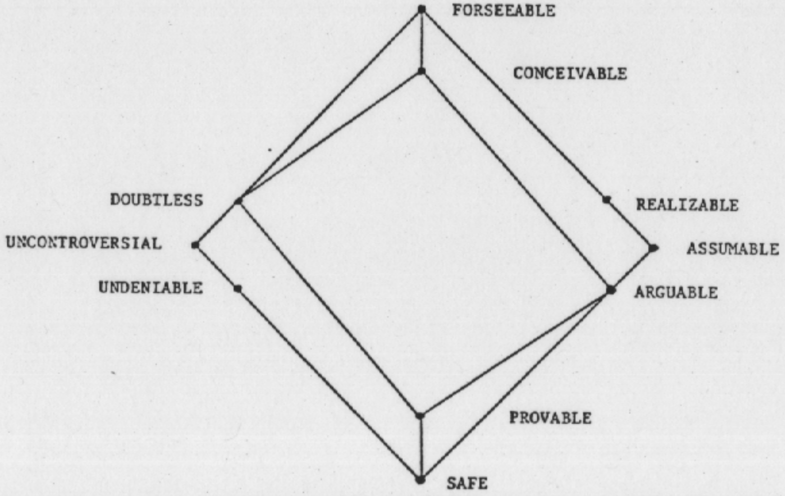

Fig. 3. Diagram of the classifications of formulas. Each line connects a set of formulas with a superset.

Figure 3 summarizes all these observations. These results illustrate the distinction between arguability and assumability, that arguability does not completely capture the notion of assumability. This is probably to be expected from the Gödel–Tarski results on the indescribability of consistency within consistent theories. It would be interesting to see a more careful analysis of this situation. One goal of such an analysis might be to connect the logic of incomplete information implicit in non-monotonic logic to other logics of incomplete information, such as the S4 interpretation of the intuitionistic predicate calculus [13, 18], Kripke's theory of truth [19; cf. 22, 44], and Lipski's theory of incomplete models [20; cf. 37, 45]. The S4 interpretation of IPC tries to describe the gradual accumulation of mathematical truths, and seems closely related to our notion of safety. Kripke's theory of truth has strong similarities to the current theory, for it develops models for the truth of self-referential and theory-referential statements which are fixed points of a certain operator on partially defined truth-predicates. Since the acceptable models of truth are restricted to be fixed points of this operator, there can be never-decided paradoxical statements. The logic of the natural notions of possibility and necessity thus are not dual, but instead form a diamond relationship similar to the case for non-monotonic assumability and safety. Lipski's theory of incomplete models is considerably simpler and stronger than either Kripke's theory or non-monotonic logic, for his incomplete models can be constructed from any partial extensions of the predicates of the language, thus producing a certain completeness in the set of possible models. This logic is particularly interesting in that it allows for the truth value of formulas to change arbitrarily often upon successive extensions of models.

In the above we have been concerned only with ways of describing the evolution of theories upon the addition of new axioms. One might also define descriptors for the case of removing axioms, or for the past history of provability. For example, assuming that all subtheories of A are consistent, we might say that p is *untested* if p is conveivable in every subtheory of A (cf. [13, p. 115]. Are there other interesting descriptors of this kind? If so, what are their properties? We have not investigated these questions, but suspect they may be fruitful.

9. A Proof Procedure for Non-Monotonic Statement Theories

In this section, we demonstrate a proof procedure for the non-monotonic statement logic. This procedure is based on the semantic tableau method for the ordinary sentential calculus [1]. In this method, a systematic attempt is made to find a falsifying interpretation for a formula under test. The formula is labeled 'false' or '0', and semantic rules guide further labeling in an obvious way. For example, to show

$$[C \supset D] \supset [\neg C \vee D],$$

start by labeling the formula false:

$$[C \supset D] \underset{0}{\supset} [\neg C \vee D]$$

the notion of assumability. This is probably to be expected from the Gödel–Tarski

For it to be false, its antecedent must be true and its consequent false:

$$\begin{array}{cccccccccccc} [C & \supset & D] & \supset & [\neg & C & \vee & D] \\ & 1 & & 0 & & & 0 & \end{array}$$

and similarly for disjunction and negation. In order to proceed further, the tableau must split into two cases to handle the embedded implication:

Case 1.
$$\begin{array}{cccccccc} [C & \supset & D] & \supset & [\neg & C & \vee & D] \\ 0 & 1 & & 0 & 0 & 1 & 0 & \end{array}$$

Case 2.
$$\begin{array}{cccccccc} [C & \supset & D] & \supset & [\neg & C & \vee & D] \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \end{array}$$

In case 1, C is labeled both 1 and 0. In case 2, D is labeled both 1 and 0. Thus there is no falsifying model, and the formula is valid.

On the other hand, consider the tableau for $[C \vee D] \supset [C \wedge D]$:

(25)
$$\begin{array}{llllllll} [C & \vee & D] & \supset & [C & \wedge & D] & \\ & 1 & & 0 & & 0 & & \\ [C & \vee & D] & \supset & [C & \wedge & D] & \\ 1 & 1 & & 0 & & 0 & & \\ [C & \vee & D] & \supset & [C & \wedge & D] & \text{CLOSED} \\ 1 & 1 & & 0 & 0 & 0 & & \\ [C & \vee & D] & \supset & [C & \wedge & D] & \text{OPEN} \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & \\ [C & \vee & D] & \supset & [C & \wedge & D] & \\ & 1 & 1 & 0 & & 0 & & \\ [C & \vee & D] & \supset & [C & \wedge & D] & \text{OPEN} \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & \\ [C & \vee & D] & \supset & [C & \wedge & D] & \text{CLOSED} \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & \end{array}$$

This tableau has been split twice, for a total of four *branches*. Two branches are *closed* as before, that is, some formula is labeled both true (1) and false (0). But two are *open*, that is, there is an exhaustive consistent labeling of formulas. This means that there are two falsifying models, so the formula is not valid. (Notice that we could have been more clever in labeling the lines of this tableau. In the second line, for instance, we could have labeled both C's at once, forcing the D's to be labeled 0, and arriving at an open branch immediately.)

We will extend this procedure to handle non-monotonic statement theories. Without going into details, we assume an implementation of the algorithm just alluded to, which takes a *goal* and generates the complete tableau for it. (E.g., the goal of (25) is $[C \vee D] \supset [C \wedge D]$.) A tableau has several branches, each a consistent labeling of subformulas if one exists (when the branch is open), else a partial labeling (when it is closed). The tableau is the result of applying all rules to the goal. Two tableaux are equal if and only if they have the same goal. The tableau of a formula is obviously computable, since the number of branches is no greater than 2^N, where N is the number of subformulas of its goal.

We state without proof the following properties of the tableau method:

The procedure is *complete* in the sense that a formula is provable if and only if its tableau has all closed branches.

The procedure is *exhaustive* in the following sense: if X and Y are sets of formulas such that $X \subseteq Y$ and $Y \vdash_{SC} p$ but $X \nvdash_{SC} p$, then in the tableau for p, in every open branch there is some element of $Y - X$ labeled 0.

For non-monotonic logic, we need to generalize to tableau structures. If A is a statement theory, and p is a formula whose provability is to be tested, then $\langle A, p, t, X\rangle$ is an A-tableau structure if and only if t is the tableau with goal $A \supset p$, and X is the smallest set such that $t \in X$, and if $t' \in X$, then if $\mathsf{M}q$ appears labeled 0 in some branch b of t', then $t'' \in X$, where t'' is the tableau with goal $A \supset \neg q$. In this last situation, we say that t' *mentions* t'' in branch b.

In the classical procedure, a tableau is closed if all its branches are, and this can be determined unambiguously. In the case of a tableau structure, we can't tell whether a tableau is closed until we have determined the status of the tableaux it mentions, and there may be loops to contend with.

Therefore we introduce the notion of an *admissible labeling* of a tableau structure, an assignment of one label, either OPEN or CLOSED, to each tableau in the structure, such that:

(a) If the tableau with goal $A \supset \neg q$ is labeled OPEN, then every occurrence of $\mathsf{M}q$ is labeled 1 in every tableau, and

(b) A branch is labeled CLOSED if and only if some formula is labeled both 0 and 1 in that branch.

The proof procedure creates tableau structures and labels them, as follows. Given A and p, the first step is to construct the tableau with goal $A \supset p$. All other tableaux needed are then constructed. That is, if some constructed tableau has a formula $\mathsf{M}q$ labeled 0 in an open branch, then construct the tableau with goal $A \supset \neg q$ if that tableau was not previously constructed. The tableau structure is then checked for admissible labelings by examining all possible labelings of the tableaux for labelings satisfying the admissibility test. This test consists of first labeling with 1 each occurrence of $\mathsf{M}q$ in the tableau structure provided that the structure contains the tableau with goal $A \supset \neg q$ labeled OPEN. Then the labeling is admissible if all tableaux labeled OPEN have some open branch, and all tableaux labeled CLOSED have every branch closed. If in all admissible labelings the initial tableau with goal $A \supset p$ is labeled CLOSED, then p is provable, and otherwise is unprovable. We will shortly prove the correctness of this algorithm.

We first present some examples. In the theory

(26) $\text{T11} = \text{SC} \cup \{\, \mathsf{M}C \supset \neg D,\ \mathsf{M}D \supset \neg E,\ \mathsf{M}E \supset \neg F\}$

(see [39]) the T11-tableau structure for $\neg F$ has only one admissible labeling:

$$\begin{array}{rl|l|l|l}
\text{T11} = & \mathsf{M}C \supset \neg D & t \quad \neg F & t' \quad \neg E & t'' \quad \neg D \quad | \quad t''' \quad \neg C \\
& 1 & \quad\;\; 0\ 1 & \quad\;\; 0\ 1 & \quad\;\; 0\ 1 \quad\quad\quad\;\; 0\ 1 \\
& \mathsf{M}D \supset \neg E & \quad\;\; \mathsf{M}E & \quad\;\; \mathsf{M}D & \quad\;\; \mathsf{M}C \\
& 1 & \quad\;\; 0 & \quad\;\; 0 & \quad\;\; 0 \\
& \mathsf{M}E \supset \neg F & & & \\
& 1 & \text{CLOSED} & \text{OPEN} & \text{CLOSED} \quad\quad \text{OPEN}
\end{array}$$

Notice that we don't bother to copy the axioms in each tableau, but only those parts that become relevant. The tableau structure shows that $\neg F \in \text{TH(T11)}$, but $\neg C \notin \text{TH(T11)}$.

Another example is the T12-tableau structure for $\neg C$, where

(27) $\text{T12} = \text{SC} \cup \{\, \mathsf{M}C \supset \neg D, \mathsf{M}D \supset \neg C\}$.

$$\begin{array}{rl|l|l}
\text{T12} = & \mathsf{M}C \supset \neg D & t \quad \neg C & t' \quad \neg D \\
& 1 & \quad\;\; 0\ 1 & \quad\;\; 0\ 1 \\
& \mathsf{M}D \supset \neg C & \quad\;\; \mathsf{M}D & \quad\;\; \mathsf{M}C \\
& 1 & \quad\;\; 0 & \quad\;\; 0
\end{array}$$

This tableau structure has two admissible labelings. If t' is labeled OPEN, t is labeled CLOSED, and vice versa. So there is an admissible labeling in which t is labeled OPEN, and $\neg C$ is not provable.

On the other hand, the T12-tableau structure for $\mathsf{M}C \vee \mathsf{M}D$ looks like this:

$$\begin{array}{rl|l|l|l}
\text{T12} = & \mathsf{M}C \supset \neg D & t \quad \mathsf{M}C \vee \mathsf{M}D & t' \quad \neg C & t'' \quad \neg D \\
& 1 & \quad\;\; 0 \quad 0\ 0 & \quad\;\; 0\ 1 & \quad\;\; 0\ 1 \\
& \mathsf{M}D \supset \neg C & & \quad\;\; \mathsf{M}D & \quad\;\; \mathsf{M}C \\
& 1 & & \quad\;\; 0 & \quad\;\; 0
\end{array}$$

Again, there are two admissible labelings, but in both of them t is labeled CLOSED, so $\mathsf{M}C \vee \mathsf{M}D$ is a theorem of T12.

(The tableau structures just given are not really complete. It is left as an exercise for the reader to show that using the axioms to split each tableau into branches will not change the outcome.)

Theorem 13. *The proof procedure always halts and finds all admissible labelings of the tableau structure for its goal.*

Proof. The theorem is easily seen true by noting that because the set of proper axioms of the theory is finite, only a finite set of tableaux can be constructed. Once this is done, there are only finitely many labelings to cycle through, with trivial checks for admissibility and provability.

The next two lemmas guarantee the correctness of the approach.

Lemma 14. *If S is a fixed point of NM_A, there is an admissible labeling of the tableau structure for $A \supset p$ such that $p \in S$ if and only if the tableau is labeled* CLOSED *in that labeling.*

Proof. Let $S \in \text{FP}(A)$. We will construct the admissible labeling. In the tableau structure for $A \supset p$, label a tableau OPEN if the goal of the tableau is $A \supset q$ and $q \notin S$. Consider one of the remaining tableaux, with goal $A \supset r$ There must be a minimal set of elements $X = \{\, \mathsf{M}q_1, \ldots, \mathsf{M}q_n\}$, such that $X \subseteq \text{As}_A(S)$ and $X \vdash_A r$. If $X = \varnothing$, then the tableau for $A \supset r$ is closed no matter how assumptions are labeled. Otherwise, by exhaustiveness, every branch of the tableau has some $\mathsf{M}q_i \in X$ labeled 0. So there will be a tableau for each such $A \supset \neg q_i$. But these tableaux will be labeled OPEN (because $\neg q_i \notin S$), so the corresponding branch of the tableau for $A \supset r$ will be CLOSED. So the whole tableau for $A \supset r$ will be CLOSED. Further, no open tableau will be labeled CLOSED, because then there would be a proof of its goal from assumptions. Thus, if the tableau for $A \supset p$ is labeled CLOSED, it can be proved from assumptions in $\text{As}_A(S)$, so $p \in S$. If it is OPEN, $p \notin S$ by construction.

Lemma 15. *If there is an admissible labeling for the tableau structure for $A \supset p$, there is a fixed point S of NM_A such that, for every tableau with goal $A \supset q$, the tableau is labeled* CLOSED *if and only if $q \in S$.*

Proof. We construct S from the labeling. Let R_0 be the set of formulas $\mathsf{M}q$ such that the tableau for $A \supset \neg q$ is labeled OPEN. Let $S_0 = \text{Th}(A \cup R_0)$, and let $\mathsf{M}q_1, \mathsf{M}q_2, \ldots$ be an enumeration of all the formulas of the form $\mathsf{M}q$ in $\mathscr{L} - R_0$, with the property that if $\mathsf{M}q_i$ is a subexpression of $\mathsf{M}q_j$, then $i \leqslant j$. (E.g., $\mathsf{M}C$ is a subexpression of $\mathsf{M}\neg\,\mathsf{M}C$.)

Define R_{i+1} and S_{i+1}, for $i = 0, 1, \ldots$ as follows:

$$R_{i+1} = R_i \quad \text{if } \neg q_{i+1} \in S_i, \quad \text{else } R_i = R_i \cup \{\, \mathsf{M}q_{i+1}\}, \text{ and}$$

$$S_{i+1} = \text{Th}(A \cup R_{i+1}).$$

Now let $S = \bigcup_{i=0}^{\infty} S_i$, and $R = \bigcup_{i=0}^{\infty} R_i$. Clearly, $S_i \subseteq S_{i+1}$ and $S = \text{Th}(A \cup R)$. Since $\text{NM}_A(S) = \text{Th}(A \cup \text{As}_A(S))$, we can show that $\text{NM}_A(S) = S$ by showing that $R = \text{As}_A(S)$.

First, to show $\text{As}_A(S) \subseteq R$. Let $\neg q \notin S$. We will show $\mathsf{M}q \in R$. If $\mathsf{M}q \in R_0$, then since $R_0 \subseteq R$, $\mathsf{M}q \in R$. Otherwise q must be some q_i. If $\neg q \notin S$, then $\neg q \notin S_{i-1}$, so $\mathsf{M}q \in R_i$, so $\mathsf{M}q \in R$.

Second, to show $R \subseteq \text{As}_A(S)$, that is, if $\mathsf{M}q \in R$, then $\neg q \notin S$. There are two cases. If $\mathsf{M}q \in R_0$, then there is an OPEN tableau for $A \supset \neg q$. Assume that $\neg q \in S$. Then there must be a $k \geqslant 1$ such that $\neg q \in S_k$ and $\neg q \notin S_{k-1}$. So $R_k \vdash_A \neg q$ and $R_{k-1} \nvdash_A \neg q$. But then by exhaustiveness, $\mathsf{M}q_k$ is labeled 0 in the tableau for $A \supset \neg q$. So there is also a tableau for $A \supset \neg q_k$. If this tableau is OPEN, then $\mathsf{M}q_k \in R_0$. If this tableau is CLOSED, $\mathsf{M}q_k \in S_0$, and hence $\mathsf{M}q_k \in S_{k-1}$. Either way, $R_k = R_{k-1}$, which is impossible.

In the other case, q will be some q_i, so $\mathsf{M}q \in R_i$, and $\neg q \notin S_{i-1}$. Assume that $\neg q \in S$, that is, $\neg q$ is an element of some S_k, $k \geqslant i$, and $\neg q \notin S_{k-1}$. Then $R_k \vdash_A \neg q$ but $R_{k-1} \nvdash_A \neg q$, so $\{\mathsf{M}q_k\} \cup R_{k-1} \vdash_A \neg q$.

Now, $\mathsf{M}q_k$ does not occur as a subexpression of $q = q_i$ (since $k \geqslant i$), so $\mathsf{M}q_k$ must occur in the axioms A. So in some branch of the tableau for $A \supset \neg q$, $\mathsf{M}q_k$ must be labeled 0. But this means that $\mathsf{M}q_k$ must be labeled 0 in some branch of the tableau for $A \supset p$, for any p. So any tableau structure must have a tableau for $A \supset \neg q_k$. This tableau must be OPEN, or $\neg q_k$ would be a member of S_0, and hence a member of S_{k-1}. So $\mathsf{M}q_k \in R_0$, so $R_k = R_{k-1}$, which is a contradiction.

It remains to show that the labels agree with the fixed point. If the tableau for $A \supset \neg q$ is OPEN, then $\mathsf{M}q \in S$ by construction. If it is CLOSED, there is a proof of $\neg q$ from R_0, so $\neg q \in S_0$. But $S_0 \subseteq S$, so the final labeling agrees as well.

Theorem 16. *If A is a statement theory (a finite extension of the sentential calculus), then non-monotonic provability in A is decidable.*

Proof. Let $\langle A, p, t, X\rangle$ be the tableau structure for a formula p. If the procedure labels t CLOSED in every admissible labeling, then there is no fixed point of NM_A which does not contain p, since there then would be an OPEN labeling. So p is in all fixed points, and hence provable. If the procedure labels t OPEN in some admissible labeling, there is a fixed point of NM_A which does not contain p, so p is unprovable.

The proof procedure extends a previous procedure due to Hewitt [12], and embodied in MICRO-PLANNER [43], a computer programming language for (among other things) mechanical theorem proving. A practical implementation of this procedure would interleave the building and labeling of tableaux, and would avoid building a complete tableau structure when unnecessary. We invite you to compare this procedure with, for instance, the tableau-structure method for S5 [14]. One difference between these procedures is that the present procedure splits tableaux into branches before generating alternatives, while the S5 procedure splits the whole set of alternatives into branches.

10. The Truth Maintenance System

The only known adequate solutions to the handling of non-monotonic proofs are Doyle's [6] TMS program and its relatives [21, 23]. With our theoretical results in hand, we can present a description of what this program does. The TMS has two basic responsibilities:

(a) It maintains a data base of proofs of formulas generated by an independent proof procedure or perceptual program. In our terms its goal is to avoid the presence of both $\neg q$ and $\mathsf{M}q$ in the data base simultaneously.

(b) It detects inconsistencies, and adds axioms to a theory in order to eliminate them.

The TMS keeps track, for each formula in the data base, of the formula's *justifications*. A justification of a formula p is a set $\{p_1, \ldots, p_n\}$ of formulas which entail p. Such a justification may be viewed as a fragment of the tableau for $A \supset p$; that is, for each branch of p's tableau, the justification contains a formula p_i labeled 0 in that branch.

The basic TMS algorithm searches for a labeling of formulas involved in justifications. It obeys two principles; p is labeled 1 if and only if all the formulas in some justification of p are labeled 1, and $\mathsf{M}p$ is labeled 1 if and only if $\neg p$ is labeled 0. When the TMS finds a labeling satisfying these conditions, it arranges the data base so that only formulas labeled 1 are 'visible' to the higher-level proof procedure or program. Thus, from the point of view of a program using the TMS, it chooses a subset of formulas to 'believe.' These formulas are said to be *in*; the other formulas are *out*.

This is reminiscent of our proof procedure's search for admissible tableau labelings, but there are some important differences. The TMS operates on partial sets of tableau fragments, so its decisions may require revision as new fragments (justifications) are discovered. But there is a more striking difference between our proof procedure and the TMS. The TMS searches for just one admissible labeling of its tableau fragments, not all such labelings. The most it can hope to find is one fixed point (actually, a finite subset of one), not all of them. In the terminology we developed earlier, it finds some of the arguable formulas rather than the provable formulas. For example, consider its behavior on the theory T12 in (27). In this theory, $\mathsf{M}C$ is arguable, and so is $\mathsf{M}D$, but neither is provable (only $\mathsf{M}C \vee \mathsf{M}D$ is provable). Nonetheless, the TMS, given the justifications $\{\mathsf{M}C\}$ of $\neg D$ and $\{\mathsf{M}D\}$ of $\neg C$, will pick one of $\{\mathsf{M}C, \mathsf{M}D\}$ to be *in*, and the other to be *out*.

There are several reasons for such jumping to conclusions. One is that since all arguable formulas are also assumable, these decisions may at worst lead to later shifts in fixed points. That is, since arguable formulas might be added consistently later on, it cannot hurt much to act on the assumption that they will be added. A more pressing rationale for this behavior is that the program of proof procedure using the TMS typically depends on beliefs of certain types to decide what to do, and cannot abide by suspended judgement; even if there is a choice of possible circumstances, the program expects the TMS to decide on one so that action may be taken.

Of course, jumping to conclusions in this manner introduces the problem of having to choose between fixed points of the theory. In many cases this problem solves itself because of the way the TMS is typically used. Usually a program using the TMS is attempting to discover which fixed point of a theory corresponds to the real world. The best way to do this is to pick one model and stick with it until trouble arises, and then salvage as much information as possible by making as few changes as necessary. 'Trouble' can take the form of new information or new deductions from old information conflicting with old information or assumptions. Either way, the response is the same; to switch to a new fixed point. Programs frequently try to organize their use of the TMS so as to ensure the case of a single fixed point being the usual case. However, it is not possible to completely determine in this fashion how the TMS should decide between alternate fixed points. One way even more information of this sort might be used would be to employ Rescher's [36] suggestion of modal categories as a method for selecting among the various fixed points generated by a theory. That is, suppose the formulas of the language are segmented into $n+1$ modal categories $\mathscr{L} = \mathscr{M}_0 \cup \cdots \cup \mathscr{M}_n$. Then given fixed points of a theory A as $S_1, \ldots, S_m$, with corresponding sets of assumptions $A_1, \ldots, A_m$, we can segment the A_i into components $A_{1,0}, \ldots, A_{1,n}, \ldots, A_{m,0}, \ldots, A_{m,n}$ in concordance with the modal categories. We can then rank the fixed points by schemes involving orderings on the vectors of assumption components. Adding such devices to TMS-like systems is an interesting topic for future research.

The two goals of the TMS, to prevent both $\neg p$ and Mp being *in*, and to prevent both p and $\neg p$ being *in*, give rise to two different types of activity. In the first case, when a new justification is discovered for some formula which then invalidates some current assumption, the TMS must reexamine the current labeling to find a new labeling consonant with the enlarged set of justifications. This process is fairly straightforward, although there are important special cases concerning circular proofs which require special care. This process thus takes on the appearance of a relaxation procedure for finding an acceptable labeling, and then determination of non-circular proofs for all formulas labeled 1.

The second type of inconsistency handled by the TMS, that of p and $\neg p$ being *in*, requires somewhat different treatment. In the first type of process just described, the TMS uses justifications in a unidirectional manner, determining labelings of formulas from the labelings of the formulas of their justifications, and not vice versa. In the second case, the TMS must traverse these justifications in the opposite direction, seeking the assumptions underlying the conflicting formulas. To resolve the inconsistency of these assumptions, the TMS converts the problem to one of the first type by producing a new justification for the denial of one of the assumptions in terms of the other assumptions. This might be viewed as the TMS sharing the weakness of our logic; it cannot rule out an assumption Mp by deriving $\neg$ Mp, but must instead produce a derivation of $\neg p$. This second process is called dependency-directed backtracking [42].

For example, the existing theory may be $\{ \mathrm{M}C \supset E\}$ in which both MC and E are believed. Adding the axiom M$D \supset \neg E$ leads to an inconsistent theory, as MD is assumed (there being no proof of $\neg D$), which leads to proving $\neg E$. The dependency-directed backtracking process would trace the proofs of E and $\neg E$, find that two assumptions, MC and MD, were responsible. Just concluding $\neg$ M$C \vee \neg$ MD does no good, since this does not rule out any assumptions, so the TMS adds the new axiom $E \supset \neg D$ which invalidates the assumption MD and so restores consistency. There are many subtleties involved, as discussed in [6].

Of course, in non-monotonic logic there is also another kind of inconsistency, that due to there being no fixed point at all. The TMS can loop forever in some cases when presented with a theory containing an inconsistent subtheory of this sort. Normally, however, this does not happen. [3] describes some conditions on the structure of the set of justifications under which a simplified variant of the TMS can be shown to always halt. McAllester's [23] TMS, through a different organization, apparently avoids these problems.

11. Discussion

In contrast to classical logics, the non-monotonic logics examined in this paper have the property that extending a theory does not always leave all theorems of the original theory intact. Such logics are of great practical interest in artificial intelligence research, but have suffered from foundational weakness. We have tried to repair this weakness by providing analyses of non-monotonic provability and semantics. Our definitions lead to proofs of the completeness of non-monotonic logic and the decidability of the non-monotonic sentential calculus.

The area of non-monotonic logic is ripe for further research. Some open problems have been mentioned in the preceding sections. In the following, we list some further interesting topics.

The major problem for non-monotonic logic is deciding provability for more general cases than statement theories. Unlike classical logic, it appears that the non-monotonic predicate calculus is not even semi-decidable. That is, there seems to be no procedure which will tell you when something is a theorem. If there were, then we could use it to decide whether p was a theorem of number theory by trying to prove p and M$\neg p$ simultaneously, since one of these must be a theorem (as there is only one non-monotonic fixed point of number theory).

Are there special cases in which provability is decidable or semi-decidable? We conjecture that many theories of interest to artificial intelligence are *asymptotically decidable*, in the following sense: there is a procedure which is allowed to change its answer an indefinite number of times about whether a formula is provable, but changes its answer only a finite number of times on each particular formula. (See for example the problem solving procedures given in [5]. Note also that classical first-order provability is asymptotically decidable by a procedure that changes its answer only once; answer 'unprovable' and then call any complete proof procedure, changing the answer if the proof procedure succeeds.) Asymptotic decidability is a fairly weak property of a predicate, but it isn't vacuous since there are predicates (such as totality) which are not decidable even in this sense. Furthermore, a procedure of this kind could be useful in spite of the provisional nature of its outputs, since a robot always has to act on the basis of incomplete cogitation.

Unfortunately, it appears that even for some finite first-order theories, provability is not asymptotically decidable. We must look for useful special cases.

We have presented a formalization of non-monotonic logic which, although very weak, captures most of the important properties desired, especially with regard to the structure of models of non-monotonic theories and their behavior upon extension by new axioms. The logic seems to be adequate for describing the TMS, an ability following naturally from the structure and evolution properties just mentioned. The logic also admits a proof of completeness and a proof procedure for the case of statement theories.

Unfortunately, the weakness of the logic manifests itself in some disconcerting exceptional cases which, while essentially irrelevant to the structure and evolution properties, indicate that the logic fails to capture a coherent notion of consistency. For example, the theory

$$(28) \quad \mathrm{T13} = \mathrm{PC} \cup \{ \mathsf{M}C \supset D, \neg D\}$$

is inconsistent in our logic because although $\neg \mathsf{M}C$ follows from $\neg D$ and $\mathsf{M}C \supset D$, $\neg C$ does not follow, thus allowing $\mathsf{M}C$ to be assumed; and so the theory fails to have a fixed point. This can be remedied by extending the theory to include $\neg C$, the approach taken by the TMS, but this extension seems arbitrary to the casual observer. As it happens, axioms like $\mathsf{M}C \supset D$ are much less common in applications than the unproblematic $\mathsf{M}C \supset C$, but it would be nice to get rid of this problem. Another incoherence of our logic is that consistency is not distributive; $\mathsf{M}C$ does not follow from $\mathsf{M}[C \wedge D]$. Our logic tolerates axioms which force an incoherent notion of consistency, as in

$$(29) \quad \mathrm{T14} = \mathrm{PC} \cup \{ \mathsf{M}C, \neg C\}.$$

A stronger logic might not allow this by forcing such theories to be inconsistent. We think that such logics are available, and will report their details in a forthcoming paper. Boolos's [2] System G gives some hints about the form of logics of provability.

There are several problems of a mathematical nature raised by non-monotonic logic. What are the details of the relationship between non-monotonic logic and the logics of incomplete information? What are the effects of different rules of inference on the construction of non-monotonic models? What are the details of the evolution of the properties of decision and provability? Are there interpretations of non-monotonic logic within classical logics? Are there connections between non-monotonic logic and logics with statements of infinite length? Is there a topological interpretation of non-monotonic logic in analogy to the topological interpretation of the intuitionistic calculus?

There are also a number of more speculative and long range topics for investigation raised by non-monotonic logic. The revision of beliefs performed by artificial intelligence programs can be viewed as a microscopic version of the process of change of scientific theories. (For a figurative description of such processes which is very close to a true description of non-monotonic logic and the TMS, see the beginning of Section 6 of Quine's *Two Dogmas of Empiricism* [32].) Can the ideas captured in non-monotonic logic be used to describe the general process of scientific discovery, or pragmatic behavior in general? How are the holistic semantics of non-monotonic logic related to changes in meanings? (Cf. particularly [7].) What are the trade-offs involved in jumping to conclusions? How costly is the suspension of judgement? Can non-monotonic logic be used to effectively describe and reason about actions, commands, conterfactuals, and causality?

ACKNOWLEDGEMENTS

We wish to thank Gerald Jay Sussman, Rohit Parikh, David Harel, Mitchell Marcus, Lucia Vaina, and Vaughan Pratt for helpful criticisms and comments. Drew McDermott acknowledges the support of a Josiah Willard Gibbs instructorship. Jon Doyle thanks the Fannie and John Hertz Foundation for supporting his research with a graduate fellowship.

REFERENCES

1. Beth, E. W., On machines which prove theorems, *Simon Stevin* (*Wis- en Natuurkundig Tijdschrift*) 32 (1958) 49.
2. Boolos, G., *The Unprovability of Consistency* (Cambridge University Press, Cambridge, 1979).
3. Charniak, E., Riesbeck, C. and McDermott, D., *Artificial Intelligence Programming* (Lawrence Erlbaum, Hillsdale, NJ, 1980).
4. de Kleer, J., Causal and teleological reasoning in circuit recognition, Ph.D. thesis, MIT Department of Electrical Engineering and Computer Science (1979).
5. de Kleer, J., Doyle, J., Steele, G. L. Jr. and Sussman, G. J., Explicit control of reasoning, MIT Artificial Intelligence Laboratory, Memo 427 (1977).
6. Doyle, J., A truth maintenance system, *Artificial Intelligence* 12 (1979), 231–272.
7. Dummett, M. A. E., The justification of deduction, *Proc. British Academy* LIX (1973).
8. Fikes, R. E. and Nilsson, N. J., STRIPS: A new approach to the application of theorem, proving to problem solving. *Artificial Intelligence* 2 (1971) 189–208.
9. Hayes, P. J., Robotologic, in Meltzer, B. and Michie, D. (Eds), *Machine Intelligence* 5 (American Elsevier, New York, 1970) 533–554.
10. Hayes, P. J., A logic of actions, in Meltzer, B. and Michie, D. (Eds), *Machine Intelligence* 6 (American Elsevier, New York, 1971), 495–520.
11. Hayes, P. J., The frame problem and related problems in artificial intelligence, in: Elithorn, A. and Jones, D. (Eds), *Artificial and Human Thinking* (Josey–Bass, San Francisco, 1973).
12. Hewitt, C. E., Description and theoretical analysis (using schemata) of PLANNER: a language for proving theorems and manipulating models in a robot, MIT Artificial Intelligence Laboratory TR–258 (1972).
13. Heyting, A., *Intuitionism: An Introduction* (North-Holland, Amsterdam, 1956).
14. Hughes, G. E. and Cresswell, M. J., *An Introduction to Model Logic* (Methuen and Co., London, 1972).
15. Joshi, A. K., Some extensions of a system for inferencing on partial information, in: Hayes-Roth, F. and Waterman, D. (Eds), *Pattern Directed Inference Systems* (Academic Press, New York, 1978).

16. Joshi, A. K. and Rosenschein, S. J., A formalism for relating lexical and pragmatic information: its relevance to recognition and generation, *Proc. Workshop on Theoretical Issues in Natural Language Processing* (Cambridge, MA, 1975) 79–83.
17. Kramosil, I., A note on deduction rules with negative premises, *Proc. Fourth International Joint Conference on Artificial Intelligence* (1975) 53–56.
18. Kripke, S. A., Semantical analysis of intuitionistic logic I, in: Crossley, J. N. and Dummett, M. A. E. (Eds), *Formal Systems and Recursive Functions* (North-Holland, Amsterdam, 1965) 92–130.
19. Kripke, S. A., Outline of a theory of truth, *J. Philosophy* 72 (19) (1975) 690–716.
20. Lipski, W. Jr., On the logic of incomplete information, in: Goos, G. and Hartmanis, J. (Eds), *Proc. Symp. on Mathematical Foundations of Computer Science 1977* (Springer, Berlin, 1977) 374–381.
21. London, P. E., Dependency networks as a representation for modelling in general problem solvers, Department of Computer Science TR–698, University of Maryland (1978).
22. Martin, R. L. and Woodruff, P. W., On representing 'True-in-*L*' in *L*, in: Kasher, A. (Ed.), *Language in Focus: Foundations, Methods and Systems* (D. Reidel, Dordrecht, 1976) 113–117.
23. McAllester, D. A., A three-valued truth maintenance system, MIT Artificial Intelligence Laboratory, Memo 473 (1978).
24. McCarthy, J. and Hayes, P. J., Some philosophical problems from the standpoint of artificial intelligence, in: Meltzer, B. and Michie, D. *Machine Intelligence* 4 (American Elsevier, New York, 1969) 463–502.
25. McCarthy, J., Epistemological problems of artificial intelligence, *Proc. Fifth International Joint Conference on Artificial Intelligence* (1977) 1038–1044.
26. McDermott, D., Assimilation of new information by a natural language-understanding System, MIT Artificial Intelligence Laboratory, AI–TR–291 (1974).
27. Mendelson, E., *Introduction to Mathematical Logic* (Van Nostrand Reinhold, New York, 1964).
28. Minsky, M., A framework for representing knowledge, MIT Artificial Intelligence Laboratory, AI Memo 306 (1974), reprinted without appendix in: Winston, P. (Ed.), *The Psychology of Computer Vision* (McGraw-Hill, New York, 1975).
29. Moore, R. C., Reasoning from incomplete knowledge in a procedural deduction system, MIT Artificial Intelligence Laboratory, AI–TR–347 (1975).
30. Nevins, A. J., A human-oriented logic for automatic theorem proving, *J. Assoc. Comput. Machinery* 21 (1974) 606–621.
31. Pratt, V. R., The competence/performance dichotomy in programming, MIT Artificial Intelligence Laboratory, Memo 400 (1977).
32. Quine, W. V., *From a Logical Point of View* (Harvard University Press, Cambridge, 1953).
33. Quine, W. V. and Ullian, J. S., *The Web of Belief*, second edition (Random House, New York, 1978).
34. Reiter, R., On closed world data bases, Department of Computer Science, University of British Columbia, TR–77–14 (1977).
35. Reiter, R., On reasoning by default, *Proc. Second Symp. on TINLAP*, Urbana, IL. (1978).
36. Rescher, N., *Hypothetical Reasoning* (North-Holland, Amsterdam, 1964).
37. Robinson, A., Formalism 64, in: Bar-Hillel, Y. (Ed.), *Logic, Methodology and Philosophy of Science* (North-Holland, Amsterdam, 1965) 228–246.
38. Robinson, J. A., A machine-oriented logic based on the resolution principle, *J. Assoc. Comput Machinery* 12 (1965) 23–41.
39. Sandewall, E., An approach to the frame problem, and its implementation, in: Meltzer, B. and Michie, D. (Eds), *Machine Intelligence* 7 (Wiley, New York, 1972) 195–204.
40. Scriven, M., Truisms as the grounds for historical explanations, in: Gardiner, P. (Ed.), *Theories of History* (Free Press, New York, 1959).
41. Scriven, M., New issues in the logic of explanation, in: Hook, S. (Ed.), *Philosophy and History* (New York University Press, New York, 1963).
42. Stallman, R. M. and Sussman, G. J., Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit analysis, *Artificial Intelligence* 9 (2) (1977) 135–196.
43. Sussman, G. J., Winograd, T. and Charniak, E., MICRO-PLANNER reference manual, MIT Artificial Intelligence Laboratory, AI Memo 203a (1971).
44. Takeuti, G., Formalization principle, in: Van Rootselaar, B. and Staal, J. F. (Eds), *Logic Methodology and Philosophy of Science III*, (North-Holland, Amsterdam, 1968) 105–118.
45. Van Frassen, B., Singular terms, truth-value gaps and free logic, *J. Phil.* LXIII (17) (1966) 481–495.
46. Weyhrauch, R. Prolegomena to a theory of formal reasoning, Stanford Artificial Intelligence Laboratory, Memo AIM–315 (1978).