MODAL PROPOSITIONAL SEMANTICS FOR REASON MAINTENANCE SYSTEMS*

Allen L. Brown, Jr.
General Electric Corporate Research and Development
Schenectady, New York 12301
Arpanet: BrownAL @GE-CRD

ABSTRACT

Non-monotonic logics are examined and found to be inadequate as descriptions of reason maintenance systems (sometimes called truth maintenance systems). A logic is proposed that directly addresses the problem of characterizing the mental states of a reasoning agent attempting to reason with respect to some object theory. The proposed logic, propositional dynamic logic of derivation (PDLD), is given a semantics, and a sound and complete axiomatization. The descriptive power of PDLD is demonstrated by expressing various inferential control policies as PDLD formulae.

In this note I will elaborate the propositional fragment of an axiomatic semantics of reason maintenance systems (RMS's) [Do2]. The development of such a semantics stems from the desire to provide a declarative specification language for RMS's with particular emphasis on the description of the control of their reasoning processes, and to serve as a formal setting within which to compare and contrast the properties of different RMS's.

I. INTRODUCTION

There is considerable ongoing research activity in the realm of non-monotonic reasoning [Pel. The avowed aim of this research is to capture in a logical formalism some of the non-monotonic processes (e.g., default reasoning and defeasible reasoning) that are clearly part of the common sense reasoning repertoire enjoyed by humans. Implicit or explicit in many of these formalisms is the notion that the formalism in some sense describes the process carried out by the reasoning agent. McDermott and Doyle [McDDo] analyze Doyle's TMS [Dol] in terms of the non-monotonic logic that they elaborate. Their analysis suggests that the logic of TMS is a fragment of their non-monotonic logic. I believe that their analysis confuses the logic practiced by the reasoning agent (the TMS) with the particular object theory that the agent reasons about. A reasoning agent should be viewed as a finitary computing entity. The computations that it carries out have the express aim of mechanizing some object theory. Depending on the nature of the object theory or the reasoning agent's grasp of the theory, the mechanization may turn out to be imperfect. With respect to logics like that of [McD] and [Re], because there cannot be, in general, a recursive enumeration of the theorems of the object theory, a reasoning agent's mechanization of such theories is bound to be imperfect. In summary, the relation that obtains between an object theory and a reasoning agent is that the theory is an ideal object that the agent might hope to compute.

The sense in which many of the non-monotonic logics that have been studied might be descriptions of RMS's, or reasoning agents more generally, is roughly the sense in which a formalization of recursive function theory might be the description of a programming language, say PASCAL. Recursive function theory can be taken as an ideal object that a PASCAL implementation attempts to mechanize. However, recursive function theory has little to say about the actual semantics of PASCAL programs. Inevitably, a formal semantics of PASCAL would include recursive function theory, but most of the meat in axiomatizing PASCAL is the formalization of the states of the abstract machine that is interpreting PASCAL.

There are some researchers who have attempted to address the issue of describing the reasoning agent and its mental states. Weyhrauch's FOL system [We] has an explicit notion of object theory and meta-theory. (Indeed, FOL permits the construction of arbitrary hierarchies of such object/meta pairs.) FOL is an axiomatic system, specifically, a first-order system with types. From my perspective, FOL's main defect is that a FOL meta-theory, if taken as an attempt to formalize the properties of reasoning agents, has no explicit notion of the agent's mental state. I believe that an explicit notion of mental state is key to many representations and control issues.

Doyle [Do2] develops a very powerful functional semantics for theories of reasoned assumptions. His semantics, in the guise of an admissible set, has a definite notion of the mental state of a reasoning agent. He elaborates his functional semantics so as to be able give taxonomic structure to a wide range of reasoning formalisms. He focuses primarily on giving an account of what inferential theories are sanctioned by different formal notions of reasoned assumptions. My interest, in

* The research reported herein is funded in part by the Defense Advanced Research Projects Agency under contract number F30602-85-C-0033.

contrast, is in describing the behavior of a reasoning agent when constrained to adhere to particular object theories. I should also mention that I prefer axiomatic to functional specifications as I think there is much more available technology for compiling operational RMS's from axiomatic descriptions.

Goodwin recently introduced [Go] a new inferential formalism, logics of current proof (LCP's). His intent is to capture the dynamic reasoning processes of finite reasoning agents. LCP's are not logics in the usual sense as they have no proof theory or model theory. Goodwin's formal account of LCP's is functional in nature. The principal appeal of LCP's is that they explicitly encode the development of the deductive process. It was in attempting to give a first-order logic account of LCP's, having models that suitably interpreted the sequence of databases in an LCP that I happened upon the idea of a dynamic logic of derivation.

The proximal technical inspiration of the dynamic logic of derivation (DLD) is the dynamic logic (DL) formalism introduced by Pratt and elaborated by Fischer, Harel, Ladner, Meyer, and others [Hal, Ha2]. DL gives axiomatic meaning to programs by means *of* a first-order language augmented with a collection of modal operators corresponding to those programs. Formulae in the language are used to characterize the states of computational processes before and after the execution of some computational step(s). DL's model theory is a collection of Kripke-style worlds [HeCr] connected by binary relations corresponding to various possible programs. Just as the worlds of DL's semantics capture the states of a computational machine, the states of a DLD model will capture the mental states of a rational agent. The approach that I shall be taking is presaged by Pratt in [Pr] where he uses variants of DL to formalize individual actions, sequences of actions (processes), and their effects. The remainder of this paper is devoted to elucidating propositional dynamic logic of derivation (PDLD).

II. SYNTAX

Let I be a first-order language equipped with functions, predicates, connectives, quantifiers, and perhaps even modalities. L has the usual formation rules for first-order languages. The details of L will not concern me very much here. Let T be a theory over the language L. T is assumed to be axiomatizable with a set of axioms and rules of inference. 'L, the language of PDLD, can to some extent be considered a metalanguage for for theories over L. Formulae over 'L will typically be used to specify how the formulae of T are actually derived from r's axioms and rules of inference. This specification will be in the form of an axiomatized theory $'T$. I will call $'T$ the mechanization of T. In effect $'T$, when so elaborated, will (partially) specify a reason maintenance system for the theory T.*

$'L$ has two sets of symbols: the atomic formulae and the atomic derivations. The atomic formulae are further subdivided into two classes, the proper atomic formulae and the reified atomic formulae. $'\Phi$ is a reified atomic formula of $'L$ if, and only if, Φ is a formula of L. I will use (possibly subscripted) ϕ, ψ, and χ to denote formula variables of L; Φ, Ψ, and χ to denote instances of formulae of L; p, q and r to denote formula variables of $'L$; P, Q, and R to denote instances of atomic formulae of $'L$; α and β to denote derivation variables; and a and b to denote instances of named atomic derivations. There is also the anonymous atomic derivation, $\vdash$. The proper atomic formulae are meant to behave like the truth value bearing constants of ordinary propositional logic. Intuitively reified atomic formulae are formulae that are asserted as deduced after some instance of a rule of inference in T has been applied.** Similarly atomic derivations are specific instances of inference rules. The PDLD-wffs and PDLD-derivations are defined by simultaneous induction:

1. an atomic formula is a PDLD-wff,
2. an atomic derivation is a PDLD-derivation,
3. for any PDLD-derivations α and β $(\alpha;\beta)$, $(\alpha \bigcup \beta)$, α^*, and α^{-1} are PDLD-derivations,
4. for any PDLD-wffs p and q and PDLD-derivation α, $\neg p$, $p \vee q$, and $<\alpha>p$ are PDLD-wffs.

I will abbreviate $\neg(\neg p \vee \neg q)$ to $p \wedge q$; $\neg p \vee q$ to $p \rightarrow q$; $(p \rightarrow q) \wedge (q \rightarrow p)$ to $p \equiv q$; $<\alpha;\alpha^{n-1}>p$ $(n>0)$ to $<\alpha^n>p$; $\neg<\alpha>\neg p$ to $[\alpha]p$; and $<\alpha^0>p$ to p.

III. SEMANTICS

Let W be a non-empty universe of states, elements of which are denoted by s and t (possibly with subscripts). A PDLD interpretation determines whether or not an PDLD-wff P is true in a state s (or s satisfies P). Atomic derivations can be viewed as binary relations on W. Accordingly an interpretation is defined to be a triple $<W,\pi,m>$, where W is a non-empty set, π is a function from the atomic formulae into 2^W, and m is a function from the atomic derivations into $2^{W \times W}$. π and m provide meaning for atomic formulae and derivations, and are extended inductively to the rest of $'L$:

* I wish to distinguish PDLD (and the first-order dynamic logic of derivation) from the dynamic logics of programs investigated by Pratt et al The distinction is not grounded so much in their respective model theories or proof theories, but rather in the fact that the model-theoretic worlds of the former are related by program statements while in the latter they are related by inferential steps

** The distinction between proper and reified atomic formulae will play no role in the development of PDLD proper The distinction becomes important when the axioms that describe particular RMS's are adjoined to the axiomatization of PDLD.

$m(\alpha;\beta)$ = $\{<s,t>|$ There is a u such that $<s,u> \cup m(\alpha)$ and $<u,t> \in m(\beta)\}$,

$m(\alpha \cup \beta)$ = $m(\alpha) \cup m(\beta)$,

$m(\alpha^*)$ = the reflexive, transitive closure of $m(\alpha)$,

$m(\alpha^{-1})$ = $\{<s,t>|<t,s> \in m(\alpha)\}$,

$m(\vdash)$ $\supseteq$ $\cup_i m(a_i)$ where a_i is an atomic derivation,

$\pi(p \vee q)$ = $\pi(p) \cup \pi(q)$,

$\pi(\neg p)$ = $W - \pi(p)$,

$\pi(<a>p)$ = $\{s|$ There is a t such that $<s,t> \in m(a)$ and $t \in \pi(p)\}$,

$\pi(<\alpha;\beta>p)$ = $\{s|$ There is a t such that $<s,t> \in m(\alpha;\beta)$ and $t \in \pi(p)\}$,

$\pi(<\alpha \cup \beta>p)$ = $\{s|$ There is a t such that $<s,t> \in m(\alpha \cup \beta)$ and $t \in \pi(p)\}$,

$\pi(<\alpha^*>p)$ = $\{s|$ There is a t such that $<s,t> \in m(\alpha^*)$ and $t \in \pi(p)\}$,

$\pi(<\alpha^{-1}>p)$ = $\{s|$ There is a t such that $<s,t> \in m(\alpha^{-1})$ and $t \in \pi(p)\}$.

Denoting $s \in \pi(P)$ by $s \models P$ and $<s,t> \in m(a)$ by sat and adopting free usage of conventional logical symbols, one may write for a fixed interpretation $<W,\pi,m>$ that $s \models <a>P$ if, and only if, there is a t such that sat and $t \models P$. Given an interpretation $S = <W,\pi,m>$, a PDLD-wff P is S-valid (written $\models_S P$) if for every $s \in W$ $s \models P$. A PDLD-wff P will be said to be PDLD-valid (written $\models P$) if for every S, it is S-valid. P will be said to be S-satisfiable if there is an s such that $s \models P$. and satisfiable if there is an S such that $\models_S P$.

IV. AXIOMATIZATION

The axioms '{T*}' through '{Bew}' together with the rules '{MP}' and '{Nec}' below form the system **P**. The axioms for **P** are:

{T*} the tautologies of propositional calculus

{[]} $[\alpha](p \rightarrow q) \rightarrow ([\alpha]p \rightarrow [\alpha]q)$

{U} $[\alpha \cup \beta]p \equiv ([\alpha]p \wedge [\beta]p)$

{C} $[\alpha;\beta]p \equiv [\alpha][\beta]p$

{Step} $[\alpha^*]p \rightarrow [\alpha]p$

{Refl} $[\alpha^*]p \rightarrow p$

{Trans} $[\alpha^*]p \rightarrow [\alpha^*][\alpha^*]p$

{Con1} $p \rightarrow [\alpha]<\alpha^{-1}>p$

{Con2} $p \rightarrow [\alpha^{-1}]<\alpha>p$

{Ind} $(p \wedge [\alpha^*](p \rightarrow [\alpha]p)) \rightarrow [\alpha^*]p$

{Bew} $[\vdash^z]p \rightarrow [\alpha^z]p$ where z is an integer or "*," and α is atomic.

The rules of inference for **P** are:

{MP} if $\vdash_P p \rightarrow q$ and $\vdash_P p$ then $\vdash_P q$.

{Nec} if $\vdash_P p$ then $\vdash_P [\alpha]p$

The following two theorems are straightforward consequences of the syntax, semantics, and axiomatization above:*

Theorem: The axioms '{T*}' through '{Bew}' are PDLD-valid.

Theorem: The rules of inference '{MP}' and '{Nec}' are sound with respect to PDLD interpretations.

Parikh's completeness proof for propositional dynamic logic of programs [Pa] can be adapted to PDLD to obtain:

Theorem: Every PDLD-valid formula is in the deductive closure of the system **P**.**

V. DESCRIPTIVE POWER

A. General Considerations on Monotonic Theories

Thus far I have done nothing that connects any particular object theory T with a mechanization 'T. To make that connection and to exhibit the descriptive power of 'L, I will augment **P** with proper axioms that characterize a monotonic theory T. Assume that T includes the first-order predicate calculus. For each axiom Φ of T, there is an axiom 'Φ of 'T. Consider an instance of modus ponens in T:

{MP} if $\vdash_T P$ and $\vdash_T P \rightarrow Q$ then $\vdash_T Q$.

This suggests an axiom for 'T of the form:

$$'P \wedge '(P \rightarrow Q) \rightarrow <MP>'Q.$$***

* Proofs of theorems may be found in [Br].

** The principal technical hurdles in adapting Parikh's complex proof to **P** are in validating certain claims that Parikh makes for "pseudo-models" and "closed sets" when applied to **P**.

*** For a complete characterization of a monotonic inference rule such as modus ponens one should also add the axiom $'P \wedge '(P \rightarrow Q) \rightarrow [MP]'Q$ since the rule is entirely deterministic in its consequent.

The second observation to be made about modus ponens is that it is "belief conserving." That is, anything that is believed before the application of modus ponens should continue to be believed afterward. Conservation of belief (and non-belief) is a property inherent in monotonic rules of inference. To generalize then from the case of modus ponens, for each inference instance (of T) represented by the atomic derivation a with antecedents $\Phi_1,\ldots,\Phi_n$ and consequent Ψ there is an axiom of 'T of the form

{a} $\quad '\Phi^{\ulcorner}\wedge\cdots\wedge{}'\Phi^{n}\rightarrow<a>{}'\Psi.$

Given that T is monotonic, it seems natural to require the following frame axiom schema to enforce belief conservation relative to each atomic derivation:

{aF} $\quad '\phi\rightarrow[a]{}'\phi$ where ϕ is any L-wff,

$\neg{}'\phi\rightarrow[a]\neg{}'\phi$ where ϕ is any L-wff $\neq\Psi$.

It can be shown that $<\vdash'>'\Phi$ can be proved from 'T (keeping in mind that 'T mechanizes the first-order predicate calculus), an augmentation of **P**, whenever Φ is a theorem of T. Indeed, **P** augmented with a axioms corresponding to an object theory T together with derivation and frame axioms as above will be termed the natural mechanization of T. This leads to asserting that a PDLD theory 'T completely mechanizes T just in case

$\vdash_T\Phi$ if, and only if,

$\vdash_{'T}<\vdash'>('\Phi\wedge[\vdash']{}'\Phi)$.*

Needless to say, if the object theory T to be mechanized happened to be the pure first-order predicate calculus, formulae such as $\neg<\vdash'>'\Phi$ cannot generally be proven in the natural mechanizing theory 'T [Bo]. This observation has important consequences *vis a' vis* the proof theory of non-monotonic theories [McD] and their mechanizations (see below).

Notice that for an object theory T and mechanizing theory 'T, I have been implicitly taking $<\vdash'>'\Phi$ to mean that 'T "believes" 'Φ to be a consequence of believing the object theory T. Suppose **P** were taken as the object theory of 'T.** 'T can be constructed in such a way that Φ is a theorem of 'T if, and only if, $<a_1;\ldots;a_n>'\Phi$ is a theorem of 'T for some sequence $a_1,\ldots,a_n$ of (reified) atomic derivations. On the other hand, it can also be demonstrated for 'T that Φ is not a theorem of 'T if, and only if, $<a_1;\ldots;a_n>'\Phi$ is *not* a theorem of 'T for any sequence $a_1,\ldots,a_n$ of atomic derivations. In fact, Φ is not a theorem of 'T if, and only if, $\neg'\Phi\rightarrow<a_1;\ldots;a_n>\neg'\Phi$ is a theorem of 'T for every sequence $a_1,\ldots,a_n$ of reified atomic derivations. The situation that appears to obtain in 'T then is the PDLD analogue of what Moore [Mo] calls autoepistemic stability of an ideally rational agent. Loosely speaking, Φ is a theorem of 'T if, and only if, from every mental state (wherein Φ may or may not believed) there is a derivation leading to a mental state in which Φ is believed. Conversely, Φ is not a theorem of 'T if, and only if, (dis-)belief in Φ is invariant under derivation.

B. Specifying Breadth-first Search

An explicit derivation of Φ is a formula of the form $<a_1;\ldots;a_k>'\Phi$. 'T enumerates the theorems of T in a breadth-first fashion if and only if

1. for each theorem Φ of T, there is an explicit derivation of Φ that is a theorem of 'T,
2. the sequence of named atomic derivations that appears in the prefix of 'Φ corresponds to the sequence of inference rules applied in the proofs of the theorems of T when enumerating them in breadth-first order,
3. if Ψ_1 precedes Ψ_2 in the breadth-first ordering, then the derivation of Ψ_2 cannot be proved as a theorem of 'T until Ψ_1 has been proved.

A formula T is said to be of rank n if the shortest proof of that formula is of length n. Then axioms of T are of rank 0. Let Δ_n be an ordered list of the last atomic derivations applied in the proofs of each of the formulae of rank n.* Breadth-first enumeration is achieved by replacing axioms '$\{a\}$' above with '$\{a_{n,m}\}$' below:

$\{a_{n,m}\}$ $\quad C_{1,1}$

$C_{n,m}\wedge{}'\Phi_{n,m,1}\wedge\cdots\wedge{}'\Phi_{n,m,k}\rightarrow$

$<a_{n,m}>'\Psi_{n,m}\wedge D_{n,m}$

and adding boundary conditions

$\{Ba_{n,m}\}$ $\quad D_{n,m}\rightarrow[a_{n,m}]C_{n,m+1}$ if there exists $a_{n,m+1}$

$D_{n,m}\rightarrow[a_{n,m}]C_{n+1,1}$, otherwise

where the $a_{n,m}$ is the m'th atomic derivation on the list Δ_n, and the $\Phi_{n,m}$'s and $\Psi_{n,m}$ are, respectively, the antecedents and consequent of the atomic derivation $a_{n,m}$.

---

* The assertion $<\vdash'>'\Phi$ does not suffice on the right hand side of the "if, and only if" as T might be a non-monotonic theory. The second clause is necessary in order to assure that once 'T derives 'Φ it "sticks" and that 'T does not oscillate, believing and disbelieving 'Φ, owing to some belief revision policy.

** To see how such a thing is possible, let 'L be the language L together with the formula 'Φ whenever Φ is a formula of L. Consider '**P**, the system **P** taken over 'L together with an axiom 'Φ whenever Φ is an axiom of **P**, the natural axiomatic encodings of the rules of inference (modus ponens and necessitation) of **P**, and the frame axioms for those rules of inference. In the same spirit as reified atomic formulae, atomic derivations that are instances of modus ponens and necessitation of the system **P** will be called reified.

* It could be that the formulae of rank n are infinite in number. In that case the enumeration will never get beyond the formulae of rank n.

The interaction of the *Cs* and D's prevents $\psi_{n,m}$ from being derived before $\psi_{n,m+1}$ is derived. Indeed, no formula of rank *n* is derived before every formula of lesser rank is derived. The ψ's are thereby forced to be produced in breadth-first order. Of course it must be verified that a theory ' *T* that mechanizes *T* completely, when modified with the breadth-first axioms, continues to mechanize *T* completely. To that end the following holds:

Theorem: If 'T is the natural mechanization of *T* with axioms '{a} and if 'T' is the breadth-first mechanization of *T* with axioms '$\{a_{n,m}\}$' and '$\{Ba_{n,m}\}$' replacing '{a}', and if $\vdash_T < \vdash^* > \Phi$, then $\vdash_{T'} < \vdash^* > \Phi$

With a different set of boundary conditions, a depth-first enumeration of the theorems of *T* could have been achieved. That is, there is a set of boundary conditions such that

1. for each theorem of *T* there is an explicit derivation of φ that is a theorem of ' T,
2. the sequence of named atomic derivations that appears in the prefix of 'φ corresponds to the sequence of inference rules applied in the proofs of the theorems of *T* when enumerating them in depth-first order,
3. if precedes ψ_2 in the depth-first ordering, then the derivation of ψ_2 cannot be proved as a theorem of '*T* until ψ_1 has been proved.

The interaction between the axioms '$\{a_{n,m}\}$' and boundary conditions suggests a general "programming" methodology for controlling the application of derivations. The propositional constants $D_{n,m}$ and $C_{n,m}$ should be viewed as "enabling" and "completion" flags for the firing of the atomic derivation $a_{n,m}$. These constants indicate respectively that a derivation can be used and that a derivation has been used. Programming then consists of designing systems of boundary conditions to achieve the desired sequencing of inferences by suitably controlling the truth values of enabling flags in various mental states.

Goodwin [Go] (and McDermott before him in [McD]) cites a number of problems in using deduction to control deduction. He remarks that attempts at controlling inferences by deductive methods have typically resulted in invalidating particular inferences altogether, or alternatively resulted in RMS states that assert that some proposition has been proven if and only if it has not been proven. It should be clear from the discussion of programming above that atomic derivations are enabled with respect to particular states. As a consequence, an inference can be temporarily en-(dis-)abled, and there is no problem whatsoever in having some proposition 'ψ be derived by some derivation that has since become disabled. The axioms '$\{Ba_{n,m}\}$' could just as well have been written

$\{Ba_{n,m}\}$ $D_{n,m} \rightarrow [a_{n,m}] C_{n,m+1} \wedge \neg D_{n,m}$ if there exists $a_{n,m+1}$

$D_{n,m} \rightarrow [a_{n,m}] C_{n+1,1} \wedge \neg D_{n,m}$ otherwise

which have the effect of disabling each of the '$\{a_{n,m,}\}$ after use.

C. Finite Reasoning Agents

At the outset of this note I proclaimed PDLD as a mechanism for describing the behavior of finite reasoning agents. Careful scrutiny of PDLD interpretations will reveal that PDLD theories admit interpretations which are at odds with any reasonable notion of a finite agent. Consider the following observations: If one thinks of a sequence of mental states related by various atomic derivations as corresponding to the flow of some sort of mental time, then time can extend infinitely into the past and future. Moreover, a mental state can be immediately preceded by multiple states. Finally, states can be "dense." That is, PDLD interpretations can be such that for an atomic derivation *a* whenever <s,t >€m(a) there is a *u* such that *<s,u>€m(a)* and *<u,t>€m(a).*

As it turns out, all of these anomalies can be legislated away with appropriate axioms. Tense logics [RU] that impose various topologies on the ordering of time provide much of what is needed. To focus on one of the anomalies, consider the infinite extension into the past. This can be eliminated with:

$$< (\vdash^{-1})^* > \neg < \vdash^{-1} > p \vee \neg p.$$

This last formula says that every state either is, or is preceded by, a state which is *not* immediately preceded by a state that satisfies $p \vee \neg p$. But since every state satisfies $p \vee \neg p$, this formula can be satisfied if, and only if, every state is either immediately preceded by no state at all, or is preceded by some state which is in turn preceded by no state. This axiom prevents infinitely long (receding) chains of states. On the other hand, it does not prevent interpretations having a particular state from which there is a receding chain of any given finite length. More axiomatic machinery still is required to prevent that.

D. Non-monotonic Theories

In considering the descriptive power of PDLD with respect to non-monotonic theories it should first be noted that the intuitive statement of the rule of possibilitation introduced in [McDDo] is directly expressible in PDLD. Recall that McDermott and Doyle first gave an informal definition of their non-monotonic rule of inference which stated that if a proposition were not provable in a theory T then the negation of the proposition is provably possible. Though the intent of this rule is clear, it is unfortunately circular. McDermott and Doyle had to appeal to an indirect technical device to capture possibilitation. In the PDLD mechanization of T, however, their original notion of possibilitation can be expressed as:

$$\{Pos\} \qquad \neg\langle\vdash^{\bullet}\rangle\,'\neg\phi \rightarrow \langle\vdash\rangle\,'\Diamond\phi$$

where '$\Diamond$' is the consistency modality of [McDDo, McD]. Possibilitation is well defined but, unfortunately, not effectively computable in general. Since there is no magic, a non-monotonic theory *T* that is not recursively enumerable, cannot have a complete mechanization that is recursively enumerable. If a (partial) mechanization *"T* is to remain r.e., such mechanizations cannot in general have the formulae $\neg\langle\vdash^{\bullet}\rangle\,'\neg\phi$ (on the antecedent side of '(Pos)') as theorems.

The whole point of a non-monotonic logic is to formalize the default and defeasible inferences that are evident in common sense reasoning and practiced by various RMS's. It should be evident that PDLD provides a mechanism for directly formalizing such reasoning without necessarily resorting to the sorts of infinitary processes implicit in McDermott and Doyle's rule of possibilitation. In order to realize defeasible inferences, a PDLD theory cannot have the general frame axioms *'{aF}',* not all atomic derivations will be belief conserving. A default introducing axiom scheme might be:

$$\{Hyp\} \qquad \neg\,'\neg\phi \rightarrow \langle\vdash\rangle\,'\phi$$

which says that if $\neg\phi$ is not currently believed then φ can be believed. Of course, it might be the case that $\vdash_{T}\langle\vdash^{\bullet}\rangle\,'\neg\phi$. Thus, the simple notion of default reasoning supported by *'[HypY* would admit states to *\nterpretations of 'T* that *sanctioned inconsistent* beliefs. Now for ' *T* to have inconsistent beliefs is not the same as ' T's being inconsistent. On the other hand, states that have $'\phi\wedge'\neg\phi$ true are irrational, and to have $\langle\vdash^{\bullet}\rangle\,'\phi\wedge'\neg\phi$ as a theorem of ' T makes *"T* irrational. RMS's generally have backtracking mechanisms to revise the set of current beliefs so that consistency of beliefs is restored. Although PDLD as presented here is not expressive enough to describe all the details of those mechanisms, it can describe the general policies that are typically enforced by those mechanisms. A weak policy might be:

$$('\phi\wedge'\neg\phi)\rightarrow[\vdash]\neg('\phi\wedge'\neg\phi)$$

which says that if the reasoning agent is in a state that is irrational with respect to a particular formula φ, all states immediately reachable from that state should be rationalized. A much stronger (and typically unenforceable by effective computation) policy is stated by:

$$('\phi\wedge'\neg\phi)\rightarrow[\vdash]\neg\langle\vdash^{\bullet}\rangle('\phi\wedge'\neg\phi)$$

This formula says that if the reasoning agent is in a state that is irrational with respect to a particular formula φ, the agent should do something (e.g., withdraw sufficient premises or hypotheses in which the irrational state is grounded) such that at no future time can the agent be in a state irrational with respect to ϕ. These examples of deduction and premise control policies seem to respond directly to McAllester's [McA] objections to non-standard logics:

> *The problem with non-monotonic logics is that they bring in non-traditional formalisms too early, muddying deduction, justifications, and backtracking. The aspect of truth maintenance which cannot be formalized in a traditional framework is premise control.,.*

Dynamic logics *of* derivation offer an opportunity to make the various issues explicit.

VI. CONCLUSIONS

In the foregoing I have developed the syntax and semantics of the propositional dynamic logic of derivation, and presented a complete axiomatization for the logic. By way of examples I have illustrated some of the expressive power available in PDLD for specifying and analyzing the behavior of reason maintenance systems. Finally I have offered dynamic logic as an alternative to the sorts of non-monotonic logics investigated heretofore as a means for giving a formal account of some aspects of common sense reasoning.

PDLD obviously cannot be completely expressive of all properties that might be ascribed to an RMS. For that, one requires the first-order dynamic logic of derivation [Br]. In the latter formalism one can not only give a complete first-order account of control protocols, but also of the collateral data structures (viz. "no-good" lists, hypothesis contexts, dependency relations, etc.) that RMS's utilize in the belief revision process. Between PDLD without the $\vdash^{\bullet}$ derivation and full first-order dynamic logic of derivation there are many alternative logics having different powers of expressiveness. The analogous dynamic logics of programs have been extensively investigated. I believe that those investigations will offer a good starting point for developing an RMS specification logic which is suitably expressive, while being deductively tractable.

REFERENCES

[Bo] Boolos, G., *The unprovability of consistency: an essay in modal logic.* Cambridge: Cambridge University Press, 1979.

[Br] Brown, A.L. (forthcoming), *Dynamic logic of derivation: considerations on the semantics of reason maintenance systems,* General Electric Research and Development Center technical report. Schenectady, New York.

[Dol] Doyle, J., "A truth maintenance system," *Artificial Intelligence* 12:0979)231-72.

[Do2] Doyle, J., *Some theories of reasoned assumptions,* Carnegie Mellon University Computer Science Department technical report no. CMU CS-83-125. Pittsburgh, 1983.

[Go] Goodwin, J.W., "WATSON. A dependency directed inference system," *Proceedings of the AAA! workshop on non-monotonic reasoning,* ed. R. Reiteretal., pp. 103-14, 1984.

[Hal] Harel, D., First-order dynamic logic, Lecture Notes in Computer Science, vol. 68. Berlin: Springer- Verlag, 1979.

[Ha2] Harel, D., "Dynamic logic," Handbook of philosophical logic, volume II: extensions of classical logic, eds. D. Gabbay and F. Guenthner, pp. 497-604. Dordrecht: D. Reidel Publishing Company, 1984.

[HeCr] Hughes, G.E. and M.J. Cresswell, An introduction to modal logic, London: Methuen, 1968.

[McA] McAllester, D.A., An outlook on truth maintenance, MIT Artificial Intelligence Laboratory memorandum no. 551. Cambridge, Massachusetts, 1980.

[McD] McDermott, D.V., "Non-monotonic logic II: non-monotonic modal theories," Journal of the Association for Computing Machinery 29:0982)33-57.

[McDDo] McDermott, D.V., and J. Doyle, "Non-monotonic logic I," Artificial Intelligence 13:0980)133-70.

[Mo] Moore, R.C., Semantical considerations on nonmonotonic logic, Proceedings of the eighth international joint conference on artificial intelligence, ed. A. Bundy, pp. 272-9, 1983.

[Pa] Parikh, R., A completeness result for a propositional dynamic logic. MIT Laboratory for Computer Science technical memorandum no. 106. Cambridge, Massachusetts, 1978.

[Pe] Pedis, D., Bibliography of literature of non-monotonic reasoning, Proceedings of the AAA1 workshop on non-monotonic reasoning, ed. R. Reiteretal., pp. 396-401, 1984.

[Pr] Pratt, V.R., Six lectures on dynamic logic, MIT Laboratory for Computer Science technical memorandum no. 117. Cambridge, Massachusetts, 1978.

[Re] Reiter, R., "A logic for default reasoning," Artificial Intelligence, 13:0980)81-132.

[RU] Rescher, N. and A. Urquhart, Temporal logic. Vienna: Springer-Verlag, 1971.

[We] Weyhrauch, R.W., "Prolegomena to a theory of mechanized formal reasoning," Artificial Intelligence 13.(1980)133-10.