THEORETICAL FOUNDATIONS FOR BELIEF REVISION

João P. Martins
Dep. Eng. Mecânica
Inst. Superior Técnico
Av. Rovisco Pais
1000 Lisboa
Portugal

Stuart C. Shapiro
Dep. Computer Sci.
University at Buffalo
226 Bell Hall
Buffalo, N.Y. 14260
USA

ABSTRACT

Belief revision systems are AI programs that deal with contradictions. They work with a knowledge base, performing reasoning from the propositions in the knowledge base, "filtering" those propositions so that only part of the knowledge base is perceived - the set of propositions that are under consideration. This set of propositions is called the set of believed propositions. Typically, belief revision systems explore alternatives, make choices, explore the consequences of their choices, and compare results obtained when using different choices. If during this process a contradiction is detected, then the belief revision system will revise the knowledge base, "erasing" some propositions so that it gets rid of the contradiction.

In this paper, we present a logic suitable to support belief revision systems and discuss the properties that a belief revision system based on this logic will exhibit. The system we present, SWM, differs from most of the systems developed so far in two respects: First, it is based on a logic which was developed to support belief revision systems. Second, its implementation relies on the manipulation of sets of assumptions, not justifications. The first feature allows the study of the formal properties of the system independently of its implementation, and the second one enables the system to work effectively and efficiently with inconsistent information, to switch reasoning contexts without processing overhead, and to avoid most backtracking.

INTRODUCTION

The ability to reason about and adapt to a changing environment is an important aspect of intelligent behavior. Most computer programs constructed by researchers in AI maintain a model of their environment (external and/or internal), which is updated to reflect the perceived changes in the environment. The model of the environment is typically stored in a knowledge-base (containing propositions about the state of the environment) and the program manipulates the information in this knowledge base. Most of the manipulation consists of drawing inferences from information in the knowledge base. All the inferences drawn are added to the knowledge base. One reason for model updating (and thus knowledge base updating) is the detection of contradictory information about the environment. In this case the updating should be preceded by the decision of what proposition in the knowledge base is the culprit for the contradiction, its removal from the knowledge base*, and the subsequent removal from the knowledge base of every proposition that depends on the selected culprit.

Belief revision systems are AI programs that deal with contradictions. They work with a knowledge base, performing reasoning from the propositions in the knowledge base and "filtering" the propositions in the knowledge base so that only part of the knowledge base is perceived - the set of propositions that are under consideration. This set of propositions is called the set of <u>believed propositions</u>. When the belief revision system considers another of these sets, we say that it <u>changes its beliefs</u>. Typically, belief revision systems explore alternatives, make choices, explore the consequences of its choices, and compare results obtained when using different choices. If during this process a contradiction is detected (i.e., both a proposition and its negation belong to the set of believed propositions), then the belief revision system will revise the knowledge base, "erasing" some propositions so that it gets rid of the contradiction. The Truth-Maintenance System (TMS) [Doyle 79], was the first domain-independent belief revision system. TMS maintains a knowledge base in which propositions are explicitly <u>marked</u> as believed or disbelieved. When a contradiction is found, TMS revises its beliefs so that no inconsistent propositions are believed. Doyle's research triggered the development of several belief revision systems [Goodwin 82, 84; McAllester 78, 80; McDermott 82, 83; Thompson 79]. These systems share two characteristics: (1) they are mainly concerned with implementation issues, paying no special attention to the logic underlying the system; (2) each proposition is justified with the propositions that directly originated it. The first aspect does not allow the formal study of the properties of the systems independently of their implementations: in those systems, it is very difficult to define and study the properties

* Or making it inaccessible to the program.

of the underlying logic except by repeatedly running the program*. The second aspect originates systems that can only deal with one situation at a time, are not able to perform inferences in a state where a contradiction was derived, and present a large computing overhead both when switching between situations and in computing the culprit for a contradiction.

As a reaction against these problems, the early 80's saw the development of new kinds of belief revision systems, characterized by: (1) an explicit concern about the foundations of the systems, independently of their implementations [Doyle 82, 83; Martins 83; Martins and Shapiro 83] and (2) the use of a new type of justifications [Martins 83; Martins and Shapiro 83; deKleer 84, 86a, 86b].

JUSTIFICATION-BASED VS. ASSUMPTION-BASED SYSTEMS

A fundamental issue in belief revision systems is to be able to identify every proposition that may have contributed to a contradiction. This is important since, on the one hand, we don't want to blame some assumption irrelevant to the contradiction as the culprit, and, on the other hand, when searching for the assumption responsible for the contradiction we don't want to leave out any assumption possibly responsible for the contradiction. In order to do this, belief revision systems have to to keep a record of where each proposition in the knowledge base came from. These records are inspected while searching for the culprit of a contradiction. Thus, associated with every proposition in the knowledge base, there will be a set, called the support of the proposition, that tells where that proposition came from.

After selecting the culprit for a contradiction, the belief revision system typically "changes its beliefs", i.e., considers another set of the propositions in the knowledge base that does not contain the culprit of the contradiction nor any proposition derived from it. Furthermore, when considering a given set of propositions, the belief revision system ignores all the other propositions that may exist in the knowledge base.

There are two different ways of recording the origin of propositions; corresponding to justification-based and assumption-based systems [deKleer 84]. In justification-based systems, the support of each proposition contains the propositions that directly originated it. This approach was taken by [Doyle 79; Goodwin 82, 84; McAllester 80; McDermott 82; Thompson

* Although there are techniques to prove properties about programs, and thus one may be tempted to use them to prove properties about these programs, without the statement of the underlying logic one does not have a clear idea of what properties to prove.

79]. In assumption-based systems, the support of each proposition contains the hypotheses (non-derived propositions) that originated it. This approach was taken by [Martins 83; deKleer 84, 86a, 86b].

Assumptions-based systems present advantages over justification-based systems, with respect to: (1) Identifying the possible culprits for a contradiction, (2) changing sets of beliefs, and (3) comparing sets of beliefs. The main advantage that justification-based systems present over assumption-based ones concerns the explanation of their reasoning. In fact, since these systems maintain a record of the history of the derivation of each proposition in the knowledge base they can explain how a given proposition was obtained. DeKleer [deKleer 84] presents an excellent discussion on these issues.

There is, however, a hidden assumption behind assumption-based systems, which is that it is possible to compute exactly which hypotheses underlie a given proposition. The obvious solution of unioning the hypotheses underlying each of the parent propositions to compute the hypotheses underlying a derived proposition won't do.* Another important issue is how to "remember" the contradictions that were derived and to avoid getting into the same contradiction twice.

In the next section we present a logic, the SWM system, that addresses these two problems. Each proposition in SWM is associated with a set (the origin set) that contains those, and only those, hypotheses used in its derivation. Each proposition in SWM is also associated with another set (the restriction set) containing the sets of hypotheses which are incompatible (produce inconsistencies) with the proposition's origin set. The SWM system defines how these sets are formed and propagated through the application of the rules of inference. Based on SWM, we define an abstract model for an assumption-based belief revision system.

THE SWM SYSTEM

In this section we introduce a logic, the SWM system (after Shapiro, Wand and Martins) that was developed to support belief revision systems. When discussing a logic, there are two aspects to consider, its syntax and its semantics.

The syntax of a logic includes a set of formation rules and a set of rules of inference. The set of formation rules determines which formulas are legal in the logic. These formulas are called well-formed formulas, wffs for short. We will assume standard formation rules for wffs with ~,

---

* This is implicitly done in some justification-based systems, e.g., with the SL-justifications of TMS [Doyle 79].

v, &, -> as connectives and ∀, E as quantifiers. See, for example, [Lemmon 78, pp.44 and 104]. The set of rules of inference (the deductive system) specifies which conclusions may be inferred from which premises. Given an argument (P,c),* we say that c is deducible from P, written P|-c, if there is a sequence of rules of inference which when applied to P produces c.

The semantics of a logic concerns the study of the conditions under which sentences are true or false. The semantics are completely determined by the specification of two things, the interpretations of the language (every possible assignment of a particular object to each particular member of the language) and the truth conditions for it (what it means for a given sentence to have a given truth value in a given interpretation). We say that the argument (P,c) is valid if there is no interpretation in which each sentence in P is true and in which c is false. If (P,c) is valid, we write P|=c.

There is nothing about validity in the deductive system, and there is nothing about deducibility in the semantics. Although syntax and semantics are separate parts of a logical system, and thus deducibility and validity are intensionally distinct, they must fit together properly in order for the system to make any sense. A logic is said to be sound if and only if every argument deducible in its deductive system is valid according to its semantics. A logic is said to be complete if and only if every argument valid according to its semantics is deducible in its deductive system. Given a "reasonable" semantics, a logic can be unsound due to "wrong" rules of inference; and a logic can be incomplete due to the lack of necessary rules of inference or due to rules of inference that are too constraining. The SWM system is an incomplete logic, since several arguments valid according to its semantics are not deducible in its deduction system. This fact should not be regarded as a drawback of the logic but rather as a feature that makes it attractive for its intended applications.

The first step towards formally analyzing arguments consists of providing precise meaning for everyday terms like "and", "or", "if", "if...then...", "every", "some", etc. In the process of translating an informal argument into a formal one, some of the features of the informal argument are lost. The important point is to keep in the model those features that are of interest to the modeler. Therefore, when assigning meaning to the logical terms, one should bear in mind which features of the informal arguments one wants to preserve in their formal counterparts. In our case, our main goal is to keep a record of propositional

---

* A premisse-conclusion argument is an ordered pair (P,c) in which P is a set of propositions, called premisses and c is a single proposition, called conclusion.

dependencies, and our approach adopts the meaning of the logical connectives used in classical logic and builds a deductive system that blocks some unwanted deductions (resulting in an incomplete system). Most of the blocked deductions involve the introduction of irrelevancies.

One of the fundamental problems that any logic underlying a belief revision system has to address is how to keep track of and propagate propositional dependencies. This is important, because, in the event of detection of a contradiction, one should be able to identify exactly which assumptions were used in the derivation of the contradictory propositions: We don't want to blame some assumption irrelevant to the occurrence of the contradiction as the culprit for the contradiction, and, when looking for the possible culprits for a contradiction, we don't want to leave out any assumption possibly responsible for the contradiction. In logic, the relevance logicians also want to keep track of what propositions were used to derive any given proposition. Relevance logicians have developed mechanisms to keep track of what assumptions were used in the derivation of a given proposition and to prevent the introduction of irrelevancies. One way of doing this (used in the FR system of [Anderson and Belnap 75, pp.346-348] and in the system of [Shapiro and Wand 76]) consists of associating each wff with a set, called the origin set, which references every hypothesis used its derivation. The rules of inference are stated so that all the wffs derived using a particular hypothesis will reference this hypothesis in their origin sets. Whenever a rule of inference is applied, the origin set of the resulting wff is computed from the origin sets of the parent wffs.* In order to guarantee that the origin set only contains the hypotheses actually used in the derivation of the wff, and no more hypotheses, some of the applications of the rules of inference allowed in classical logic are blocked. Most of this mechanism was adopted in the SWM system.

Besides the dependency-propagation mechanism, there is another advantage in using relevance logic, to support belief revision systems. In classical logic a contradiction implies anything; thus, in a belief revision system based on classical logic, whenever a contradiction is derived it should be discarded immediately. In a relevance-logic-based belief revision system, we may allow the existence of a contradiction in the knowledge base without the danger of filling the knowledge base with unwanted deductions. In a relevance logic-based belief revision system all a contradiction indicates is that any inference depending on every hypothesis underlying the contradiction is of no value. In this type of systems we can even perform reasoning in a knowledge base known to be inconsistent. See, for example, [Martins 83], and [Martins and Shapiro

* The resulting origin set can either be the union of the origin sets of the parent wffs or the set difference of the origin sets of the parent wffs.

86].

Another important issue in belief revision systems which will be reflected in our logic consists in the recording of the conditions under which contradictions may occur. This is important because once we discover that a given set is inconsistent,* we may not want to consider it again, and even if we do want to consider it, we want to keep in mind that we are dealing with an inconsistent set. In the SWM system, contradictions are recorded by associating each wff with a set, called the restriction set, that contains information about which sets unioned with the wff's origin set produce an inconsistent set. When new wffs are derived, their restriction sets are computed directly from the restriction sets of the parent wffs, and when contradictions are detected all the wffs whose origin set references any of the contradictory hypotheses has its restriction set updated in order to record the newly discovered contradictory set. Similarly to what happens with origin sets, we will make sure that restriction sets don't have any more information than they should.

In addition, for the proper application of some rules of inference, it is important to know whether a given wff was introduced as a hypothesis or was derived from other wffs. In order to do this, we associate each wff with an identifier, called the origin tag that tells whether the wff is a hypothesis, a normally derived proposition, or a special proposition, that if treated regularly, would introduce irrelevancies into the knowledge base.**

Formally, the SWM system deals with objects called supported wffs. A supported wff consists of a wff and an associated triple containing an origin tag (OT), an origin set (OS), and a restriction set (RS). The set of all supported wffs is called the knowledge base. We write A|t,a,r to denote that A is a wff with OT t, OS a, and RS r, and we define the functions ot(A)=t, os(A)=a and rs(A)=r.

The problem of multiple derivations of the same wff is not directly addressed by the SWM system: if a proposition is derived in several different ways then it is added to the knowledge base with different OTs, OSs, and RSs. It is the computer system that interprets the knowledge base that worries about the problem of multiple derivations (see [Martins 83], and [Martins and Shapiro 84]).

---

* A set is inconsistent if a contradiction may be derived from it. A set is consistent just in case it is not inconsistent. We represent a contradiction by -><-, thus A is inconsistent if A |- -><-

** For a discussion of this latter case and the reasons that lead us to introduce this additional value for origin tags, refer to [Martins 83] or [Martins and Shapiro forthcoming].

The OS is a set of hypotheses. The OS of a supported wff contains those (and only those) hypotheses that were actually used in the derivation of that wff. The OTs range over the set {hyp, der, ext}: hyp identifies hypotheses, der identifies normally derived wffs within SWM, and ext identifies special wffs whose OS was extended. An RS is a set of sets of wffs. A wff, say A, whose RS is {R1, ... ,Rn} means that the hypotheses in os(A) added to any of the sets R1, ... ,Rn produce an inconsistent set. The RS of an extended wff will contain every set which unioned with the wff's OS will produce a set that is known to be inconsistent. Our rules of inference guarantee that the information contained in the RS is carried over to the new wffs whenever a new proposition is derived. Furthermore, the rules of inference guarantee that RSs do not contain any redundant information; i.e., given A|t,a,{R1,...,Rn}, the following types of redundancy do not arise:

1. There is no $r \in \{R1,...,Rn\}$ such that $r \cap a \neq \emptyset$.*

2. There are no $r \in \{R1,...,Rn\}$ and $s \in \{R1,...,Rn\}$, such that $r \subset s$.**

We say that the supported wff A|t,a,{R1, ... ,Rn} has a minimal RS if the following two conditions are met:

1. $\forall r \in \{R1, \ldots ,Rn\}\ (r \cap a)=\emptyset$;

2. $\forall r,s \in \{R1, \ldots ,Rn\}\ r \not\subset s$.

In [Martins 83], we prove that all the supported wffs in the knowledge base resulting from the application of the rules of inference of the SWM system have minimal RS.

To compute the RS of a wff resulting from the application rule of inference, we define the functions μ and $\int$. The function μ is used whenever a rule of inference which generates a supported wff whose OS is the union of the OSs of the parent wffs is applied. It generates the RS of the resulting wff by unioning the RSs of the parent wffs and removing from the resulting set some sets which would be redundant, namely that would violate one of the two conditions listed above. The function $\int$ is used by the rules of inference which generate a supported wff with a

* Otherwise, the set would r contain extra information, namely, all the wffs in r a.

** Otherwise, the set s could be discarded from the restriction set without any loss of information: Since r belongs to the RS of A|t,a,{R1,...,Rn}, we know that that $a \cup r$|- -><-. Also, since any set containing an inconsistent set is itself inconsistent, we could infer that $a \cup s$ is inconsistent, since $(a \cup r) \subset (a \cup s)$.

smaller OS than the parent wffs. It takes the RS of the several hypotheses in the resulting OS and computes a minimal RS from those RSs. The functions μ and $\int$ are defined as follows:

$$\mu(\{r1, \ldots ,rm\},\{o1, \ldots ,on\}) = \sigma(\Psi(r1 \cup \ldots \cup rm, o1 \cup \ldots \cup on)),$$

where

$$\Psi(R,O) = \{a \mid (a \in R \,\&\, a \cap O = \emptyset) \vee (Eb)[b \in R \,\&\, b \cap O \neq \emptyset \,\&\, a = b - O]\}$$

and

$$\sigma(R) = \{a \mid a \in R \,\&\, \sim(Eb)(b \neq a \,\&\, b \in R \,\&\, b \subset a)\}$$

and

$$\int(O) = \mu(\{r \mid EH \in O : r = \underline{rs}(H)\} , \{o \mid EH \in O : o = \underline{os}(H)\})$$

To compute the OT of a wff resulting from the application of the rules of inference, we define the function $\uparrow$ as follows:

$$\uparrow(a,b) = \begin{cases} \text{ext if } a=\text{ext ot } b=\text{ext} \\ \text{der otherwise} \end{cases}$$

$$\uparrow(a,b,\ldots,c) = \uparrow(a,\uparrow(b,\ldots,c))$$

Two supported wffs are said to be <u>combinable</u> by some rule of inference if the supported wff resulting from the application of the rule of inference has an OS that is not known to be inconsistent. We define the predicate <u>Combine</u>, which decides the combinability of the supported wffs A and B:

$$\text{Combine}(A,B) = \begin{cases} \text{false} & \text{if } Er \in \underline{rs}(A) : r \subset \underline{os}(B) \\ \text{false} & \text{if } Er \in \underline{rs}(B) : r \subset \underline{os}(A) \\ \text{true} & \text{otherwise} \end{cases}$$

The rules of inference of the SWM system, guarantee that:

1. The OS of a supported wff contains <u>every</u> hypothesis that was used in its derivation.

2. The OS of a supported wff <u>only</u> contains the hypotheses that were used in its derivation.

3. The RS of a supported wff records <u>every</u> set known to be inconsistent with the wff's OS.

4. The application of rules of inference is blocked if the resulting wff would have an OS known to be inconsistent.

It is important to distinguish between a set being inconsistent and a set being known to be inconsistent. An inconsistent set is one from which a contradiction can be derived; a set known to be inconsistent is an inconsistent set from which a contradiction has been derived. The goal of adding RSs is to avoid re-considering known inconsistent sets of hypotheses.

The OT and OS of a proposition reflect the way the proposition was derived: the OS contains the hypotheses underlying that proposition, and the OT represents the relation between the proposition and its OS. The RS of a proposition reflects our current knowledge about how the hypotheses underlying that proposition relate to the other hypotheses in the knowledge base. Once a proposition is derived, its OT and OS remain constant; however, its RS changes as the knowledge about all the propositions in the knowledge base does. Again we do not address here the problem of multiple derivations of the same proposition, a fundamental problem in belief revision. In the SWM system, if the same wff is derived in several different ways, then several supported wffs are added to the knowledge base (all of them with the same wff) and thus the reason we say that the OT and OS of a wff remain constant. The program that uses the knowledge base generated by SWM treats these wffs appropriately.

The following are the rules of inference of the SWM system.*

Hypothesis (Hyp): For any wff A and sets of wffs R1...Rn ($n \geq 0$), such that $\forall r \in \{R1, \ldots, Rn\}$: $r \cap \{A\} = \emptyset$ and $\forall r,s \in \{R1, \ldots, Rn\}: r \not\subset s$, we may add the supported wff A|hyp,{A},{R1, ... ,Rn} to the knowledge base, provided that A has not already been introduced as a hypothesis.

Implication Introduction (->I): From B|der,o,r and any hypothesis H∈o, infer H->B|der,o−{H},$\mathcal{J}$(o−{H}).

Modus Ponens - Implication Elimination, Part 1 (MP): From A|t1,o1,r1, A->B|t2,o2,r2, and Combine(A,A->B), infer B|$\uparrow$(t1,t2),o1 ∪ o2,μ({r1,r2},{o1,o2}).

Modus Tollens - Implication Elimination, Part 2 (MT): From A->B|t1,o1,r1, ~B|t2,o2,r2, and Combine(A->B,~B), infer

* There is an extra connective in the SWM system, the truth-functional or, which will not be discussed in this paper. For a detailed description of this connective, refer to [Martins 83], and [Martins and Shapiro 84].

~A|↑(t1,t2),o1∪o2,μ({r1,r2},{o1,o2}).

Negation Introduction (~I):
From A|t1,o,r, ~A|t2,o,r, and any set {H1, ... ,Hn}⊂o, infer ~(H1&...&Hn) | ↑(t1,t2), o-{H1, ... ,Hn}, ʃ(o-{H1, ... ,Hn}).
From A|t1,o1,r1, ~A|t2,o2,r2, o1≠o2, Combine(A,~A), and any set {H1, ... ,Hn}⊂(o1 o2), infer ~(H1&...&Hn) | ext, (o1∪o2)-{H1, ... ,Hn}, ʃ((o1 o2)-{H1, ... ,Hn}).

Negation Elimination (~E): From ~~A|t,o,r, infer A|↑(t,t),o,r.

Updating of Restriction Sets (URS): From A|t1,o1,r1, and ~A|t2,o2,r2, we must replace each hypothesis H|hyp,{H},R such that H∈(o1∪o2) by H|hyp,{H}, σ(R ∪{(o1∪o2)-{H}}). Furthermore, we must also replace every supported wff F|t,o,r (t=der or t=ext) such that o∩(o1∪o2)≠∅ by F|t,o, σ(r ∪{(o1∪o2)-o}).

And Introduction (&I):
From A|t1,o,r and B|t2,o,r, infer A&B|↑(t1,t2),o,r.
From A|t1,o1,r1, B|t2,o2,r2, o1≠o2, and Combine(A,B), infer A&B|ext,o1 o2,u({r1,r2},{o1,o2}).

And Elimination (&E): From A&B|t,o,r, and t≠ext, infer either A|der,o,r or B|der,o,r or both.

Or Introduction (vI): From ~A->B|t1,o,r and ~B->A|t2,o,r, infer AvB|↑(t1,t2),o,r.

Or Elimination (vE):
From AvB|t1,o1,r1, ~A|t2,o2,r2, and Combine(AvB,~A), infer B|↑(t1,t2),o1∪o2,μ({r1,r2},{o1,o2}).
From AvB|t1,o1,r1, ~B|t2,o2,r2, and Combine(AvB,~B), infer A|↑(t1,t2),o1∪o2,μ({r1,r2},{o1,o2}).
From AvB|t1,o1,r1, A->C|t2,o2,r2, B->C|t3,o2,r2, and Combine(AvB,A->C), infer C | ↑(t1,t2,t3),o1∪o2,μ({r1,r2},{o1,o2}).

∀ introduction (∀I): From B(t) | der,o∪{A(t)},r, in which A(t) is a hypothesis which uses a term (t) never used in the system prior to A's introduction, infer ∀(x)[A(x)->B(x)] | der,o, ʃ(o).*

* According to this rule of inference, the universal quantifier can only be introduced in the context of an implication. This is not a drawback, as may seem at first, since the role of the antecedent of the implication (A(x)) is to define the type of object that are being quantified. This is sometimes called relativized quantification.

∀ elimination - Universal Instantiation (∀E):
From the supported wffs ∀(x)[A(x)->B(x)]|t1,o1,r1, A(c)|t2,o2,r2 and Combine(∀(x)[A(x)->B(x)], A(c)), in which c is any individual symbol, infer A(c)->B(c) | ↑(t1,t2),o1 ∪ o2,μ({r1,r2},{o1,o2});

E introduction (EI): From A(c) | t,o,r in which c is an individual constant, infer E(x)[A(x)] | ↑(t,t),o,r;

E elimination (EE): From E(x)[A(x)] | t,o,r and any individual constant c which was never used before, infer A(c) | ↑(t,t),o,r.

Among others, the following theorems hold for SWM (their proof can be found in [Martins 83]):

Theorem: All the supported wffs in the knowledge base resulting from the application of the rules of inference of SWM have minimal RS.

Theorem: In the knowledge base resulting from the application of the rules of inference of SWM, if two supported wffs have the same OS, then they have the same RS as well.

Theorem: Every OS has recorded with it every known inconsistent set.

A CONTEXTUAL INTERPRETATION FOR SWM

We now discuss how a program using SWM should interpret SWM's wffs. We provide a contextual interpretation for SWM. We use the word "contextual interpretation" instead of just "interpretation" for the following two reasons: On the one hand, we want to stress that we are not providing an interpretation for SWM in the logician's sense of the word; on the other hand, we want to emphasize that our definition of truth depends on the notion of context. This contextual interpretation defines the behavior of an abstract revision system (i.e., not tied to any particular implementation), which we call MBR (Multiple Belief Reasoner).

MBR works with a knowledge base containing propositions that are associated with an OT, OS, and RS (in SWM's sense). Propositions are added to the knowledge base according to the rules of inference of SWM. We define a context to be a set of hypotheses. A context determines a Belief Space (BS), which is the set of all the hypotheses defining the context and all the propositions that were derived exclusively from them. Within the SWM formalism, the wffs in a given BS are characterized by having an OS that is contained in the context. The set of contexts represented in the knowledge base is the power set of the set of hypotheses existing in the knowledge base.

Any operation performed within the knowledge base (query, addition, deletion, etc.) will be associated with a context. We will refer to this context as the current context. While the operation is being carried out, the only propositions that will be considered are the propositions in the BS defined by the current context. This BS will be called the current BS. A proposition is said to be believed if it belongs to the current BS. We can look at contexts as delimiting smaller knowledge bases (the Belief Spaces) within the knowledge base. The only propositions that are retrievable are those propositions that belong to the current BS.

A common goal of belief revision systems is to stay away from contradictions. Taking this into account, it would seem natural to constrain contexts to be consistent sets of hypotheses, not just any sets of hypotheses. However, it may be the case that one desires to perform reasoning within the BS defined by an inconsistent context (in SWM, the existence of contradictions is not as damaging as in classical logic, in which anything can be derived from a contradiction) and thus the condition that a context is not known to be inconsistent will not be compulsory but rather advisable if one doesn't explicitly want to perform reasoning in a BS that is known to be inconsistent. The reason why it is advisable is that within a BS defined by a context not known to be inconsistent some simplification can be considered during the application of the rules of inference (for details refer to [Martins 83]).

Let us now consider how MBR acts when a contradiction is detected. SWM has two rules of inference to handle contradictions: negation introduction and updating of restriction sets. When a contradiction is detected, one of two things will happen:

1. Only one of the contradictory wffs belongs to the current BS:* the contradiction is recorded (through the application of URS), but nothing more happens. The effect of doing so is to record that some set of hypotheses, properly containing the current context, is now known to be inconsistent. This results in what we call belief revision within a context properly containing the current context.

2. Both contradictory wffs belong to the current BS: URS is applied, resulting in the updating of the RSs of the propositions in the knowledge base, and, in addition, the rule of ~I may also be applied. This results in what we call belief revision within the current context, normally originating the disbelief (removal from the current context) of some of its hypotheses.

Examples of these types of belief revision can be found in [Martins 83].

---

* Note that at least one of the contradictory wffs belongs to the current BS, since a contradiction is detected whenever some newly derived wff contradicts some existing one, and newly derived wffs always belong to the current BS.

CONCLUDING REMARKS

In this paper, we discussed an important class of AI programs, belief revision systems. Belief revision is important whenever reasoning is performed with a knowledge base that may contain contradictory information. Belief revision systems are capable of considering only part of the knowledge base (the set of believed propositions), perform inferences from this set, and, if a contradiction is detected, replace this set by another one (change their beliefs), and afterwards disregards every proposition that does not belong to the new set. To obtain this behavior, belief revision systems have to maintain a record of where each proposition in the knowledge base came from. We discussed two ways of keeping these records, corresponding to assumption-based and justification-based systems.

In order to build an assumption-based belief revision system, we developed a formalism that associates each proposition in the knowledge base with the set of hypotheses used in its derivation. We presented a logic (SWM) loosely based on relevance logic that captures the notion of propositional dependency and is able to deal with contradictions. SWM associates two sets with each proposition: the origin set contains every hypothesis used in the derivation of the proposition; the restriction set contains those sets of hypotheses that are incompatible with the proposition's origin set.

Each proposition generated by the rules of inference of SWM has a minimal restriction set, in the sense that restriction sets are free from some kinds of redundancies. Each such proposition has a maximal restriction set in the sense that its restriction set records all inconsistent sets known so far. Every proposition with the same origin set has the same restriction set, reflecting the fact that restriction sets are both minimal and maximal.

We defined the behavior of an abstract program based on SWM, the Multiple Belief Reasoner (MBR). In MBR, a context is any set of hypotheses. A context determines a belief space (BS), which is the set of all propositions whose origin set is contained in the context. A BS contains all the propositions that depend exclusively on the hypotheses defining the context. Given any context, the only propositions whose truth value is known are those propositions that belong to the BS defined by the context. The truth value of all the other propositions is unknown. By a proposition having an unknown truth value, we mean that in order to compute its truth value one has to carry out further deduction, and it may even be possible that its truth value is not computable from the hypotheses under consideration. At any moment, the only propositions that are believed (and thus retrievable from the knowledge base) are the ones that belong to the BS under consideration.

MBR only considers the propositions in the BS under consideration and thus, when a contradiction is detected and, after selecting some hypotheses as the culprit for the contradiction, in order to make inaccessible to the belief revision system all the propositions that were previously derived from such hypotheses, all one has to do in MBR is remove the selected hypotheses from the context under consideration. Afterwards, all the propositions derived from the selected hypotheses are no longer in the BS under consideration and consequently are not retrievable by the deduction system.

Aknowledgements

Many thanks to John Corcoran, Jon Doyle, Donald McKay, Ernesto Morgado, J. Terry Nutter, William J. Rapaport and members of the SNePS Research Group for their criticisms and suggestions.

This work was partially supported by the National Science Foundation under Grant MCS80-06314 and by the Instituto Nacional de Investigação Científica (Portugal), under Grant No. 20536; Preparation of this paper was supported in part by the Air Force Systems Command, Rome Air Development Center, Griffiss Air Force Base, New York 13441-5700, and the Air Force Office of Scientific Research, Bolling AFB DC 20332 under contract No. F30602-85-C-0008.

References

Anderson A. and Belnap N., Entailment: The Logic of Relevance and Necessity, Vol.1, Princeton University Press, 1975.

de Kleer J., "Choices without Backtracking", Proc. AAAI-84, pp. 79-85.

de Kleer J., "An Assumption-Based Truth Maintenance System", Artificial Intelligence, Vol.28, No.1, 1986a.

de Kleer J., "Problem Solving with the ATMS", Artificial Intelligence, Vol.28, No.1, 1986b.

Doyle J., "A Truth Maintenance System", Artificial Intelligence, Vol.12 No.3, pp.231-272, 1979.

Doyle J., "Some Theories of Reasoned Assumptions: An Essay in Rational Psychology", Department of Computer Science, Carnegie-Mellon University, 1982.

Doyle J., "The Ins and Outs of Reason Maintenance", Proc. IJCAI-83, pp.349-351.

Goodwin J. W., "A Improved Algorithm for Non-Monotonic Dependency Net Update", Research Report LiTH-MAT-R-82-23, Software Systems Research Center, Linkoping Institute of Technology, Sweden, August 1982.

Goodwin J.W., "WATSON: A Dependency Directed Inference System", Proc. Non-Monotonic Reasoning Workshop, pp.103-114, 1984.

Lemmon E.J., Beginning Logic, Hackett Publishing Company, 178.

Martins J., "Reasoning in Multiple Belief Spaces", Ph.D. Dissertation, Technical Report 203, Department of Computer Science, State University of New York at Buffalo, 1983.

Martins J. and Shapiro S., "Reasoning in Multiple Belief Spaces", Proc. IJCAI-83, pp.370-373.

Martins J. and Shapiro S., "A Model for Belief Revision", Proc. Non-Monotonic Reasoning Workshop, pp.241-294, American Association for Artificial Intelligence, 1984.

Martins J. and Shapiro S., "Hypothetical Reasoning", Proc. Applications of Artificial Intelligence to Engineering Problems, 1986.

Martins J. and Shapiro S., "A Logic for Belief Revision", forthcoming.

McAllester D., "A Three-valued Truth Maintenance System", Technical Report Memo 473, Massachussets Institute of Technology, AI Lab., 1978.

McAllester D., "An Outlook on Truth-Maintenance", AI Memo 551, Massachussets Institute of Technology, AI Lab., 1980.

McDermott D., "Contexts and Data Dependencies: A Synthesis", Department of Computer Science, Yale University, 1982.

McDermott D., "Data Dependencies on Inequalities", Proc. AAAI-83, pp.266-269.

Shapiro S. and Wand M., "The Relevance of Relevance", Technical Report No.46, Computer Science Department, Indiana University, 1976.

Thompson A., "Network Truth-Maintenance for Deduction and Modelling", Proc. IJCAI-79, pp.877-879.