How To Change Your Mind

Joao P. Martins, Maria R. Cravo

No&ucirc;s, Volume 25, Issue 4, Special Issue on Cognitive Science and Artificial Intelligence (Sep., 1991), 537-551.

How To Change Your Mind

JOÃO P. MARTINS and MARIA R. CRAVO

INSTITUTO SUPERIOR TÉCNICO
LISBON, PORTUGAL

ABSTRACT

In this paper, we investigate the rules that should underlie a computer program that is capable of revising its beliefs or opinions. Such a program maintains a model of its environment, which is updated to reflect perceived changes in the environment. This model is stored in a knowledge base, and the program draws logical inferences from the information in the knowledge base. All the inferences drawn are added to the knowledge base.

Among the propositions in the knowledge base, there are some in which the program believes, and there may be others in which the program does not believe. Inputs from the outside world or reasoning carried out by the program may lead to the detection of contradictions, in which case the program has to revise its beliefs in order to get rid of the contradiction and to accommodate the new information.

1. INTRODUCTION

In this paper, we investigate the rules that should underlie a computer program that is capable of revising its beliefs or opinions. Such a program is called a *Belief Revision System*, BRS for short. We assume that the BRS maintains a model of its environment, which is updated to reflect perceived changes in the environment. This model is stored in a knowledge base, and the BRS draws logical inferences from the information in the knowledge base. All the inferences drawn are added to the knowledge base. We also assume that the model of the environment is represented by logical sentences.

Among the propositions in the knowledge base, there are some in which the BRS believes, and there may be others in which the

NOÛS 25 (1991) 537-551

BRS does not believe. Inputs from the outside world or reasoning carried out by the BRS may lead to the detection of contradictions, in which case the BRS has to revise its beliefs in order to get rid of the contradiction and to accommodate the new information. This change of beliefs consists of a decision about what proposition is the culprit for the contradiction, its disbelief, and the subsequent disbelief of every proposition that depends on the selected culprit.

An essential aspect of a BRS is recording dependencies between beliefs. If it is not possible to distinguish beliefs that depend upon a hypothesis from beliefs that do not, then any belief must be questioned when the hypothesis is removed; if it is not possible to identify the hypotheses that underlie a contradiction, then every hypothesis in the knowledge base may be questioned when a contradiction is detected.

There is a rich tradition in Artificial Intelligence related to the study of systems that maintain dependencies among propositions and are able to cope with the detection of contradictions: Truth Maintenance Systems [Martins 1987, 1990]. These systems, however, merely record dependencies as told by an outside system, the problem solver, and have no reasoning capabilities. In the philosophic literature, there is also work concerning the study of changing one's mind due to the occurrence of contradictions [Stalnaker 1984, Harman 1986, Gärdenfors 1988, Gärdenfors & Makinson 1988]. This work, however, is not concerned with the computer implementation of the theories developed and, furthermore, assumes logical omniscience, i.e., all the consequences of the premises are known, which is unrealistic from a practical point of view. The work reported in this paper pertains to both approaches by defining a model for maintaining sets of beliefs, carrying out reasoning within these beliefs, and revising beliefs whenever contradictions are detected.

In our work, we take the word ''belief'' to denote *justified belief*: a proposition is believed by the BRS either because it was told so or because it can be derived from other believed propositions.

2. THE INFERENCE SYSTEM

A program capable of changing its beliefs has to keep a record of where each proposition in the knowledge base came from—the *support* of the proposition. The support is used both during the identification of the possible culprits for a contradiction and in the process of changing the system's beliefs. In this section, we are concerned with the computation of the support of propositions. In this respect, we discuss a logic, SWM* (after *S*hapiro, *W*and, and *M*artins). The SWM* system is a successor to the SWM system [Martins & Shapiro

1988], which, in turn, is a successor to the system of Shapiro & Wand 1976, which was developed to support BRSs. The interesting aspect of supporting a BRS in SWM* is that the dependencies among propositions are computed by the system itself rather than having to force the user (or an outside system) to do this, as in many existing systems.

SWM* is loosely based on relevance logic [Anderson & Belnap 1975]. The main features of relevance logic used in SWM* are (1) the association of each wff with a set containing all hypotheses (non-derived propositions) that were *really* used in its derivation (the origin set) and (2) the statement of the rules of inference taking origin sets into account, specifying what should be the origin set of the resulting wff.

Another important issue in BRSs consists in the recording of the conditions under which contradictions may occur. This is important, because once the BRS discovers that a given set is inconsistent, it may not want to consider it again, and even if it wants to consider it, it wants to keep in mind that it is dealing with an inconsistent set.

2.1 KNOWLEDGE STATES

SWM* deals with knowledge states. A knowledge state is a pair containing a knowledge base and a set of sets known to be inconsistent. The knowledge base contains propositions (written as wffs) associated with a support (an indication of dependencies between a particular wff and other wffs in the knowledge base). The set of known inconsistent sets records all sets of hypotheses in the knowledge base that were discovered to be inconsistent. Since we do not assume logical omniscience, the knowledge base does not necessarily contain all the consequences that can be drawn from the propositions it contains. It may even happen that the knowledge base is inconsistent but that the inconsistency has not been discovered. Whenever new inconsistencies are detected, they are recorded in the known inconsistent sets.

The *knowledge base* is a set of supported wffs. A *supported wff* consists of a wff and an associated pair, its support, containing an origin tag and an origin set. For a particular supported wff, the origin tag indicates how the supported wff was placed in the knowledge base (i.e., whether it was supplied by an outside system or was generated during deduction), and the origin set indicates the dependencies of this supported wff on other wffs (hypotheses) in the knowledge base.

Supported wffs are of the form $<A,\tau,\alpha>$, where A is a wff with origin tag τ and origin set α. The pair (τ, α) is called the *support* of the supported wff $<A, \tau, \alpha>$. The support of a wff is not part of the wff itself but rather associated with a *particular occurrence of the wff.* The *origin tag* is an element of the set $\{hyp, der, ext\}$: *hyp* identifies hypotheses, *der* identifies normally derived wffs within SWM*, and *ext* identifies special wffs whose origin set was extended. A supported wff with *ext* origin tag has to be treated specially in order to avoid the introduction of irrelevancies (for a discussion of this issue, see Martins & Shapiro 1988). The *origin set* is a set of hypotheses. The origin set of a supported wff contains those hypotheses that were *actually used* in the derivation of that wff. The rules of inference of SWM* guarantee that the origin set of a supported wff contains *all and only* the hypotheses that were used in its derivation. For example, $<Man(Socrates), hyp, \{Man(Socrates)\}>$ and $<Mortal(Socrates), der, \{Man(Socrates), \forall x[Man(x) \rightarrow Mortal(x)]\}>$ are supported wffs corresponding, respectively, to a hypothesis and a derived wff.

A *knowledge state*, written [[*KB*,*KIS*]], is a pair containing a knowledge base (*KB*) and a set of known inconsistent sets (*KIS*). The *knowledge base* is a set of supported wffs; the *set of known inconsistent sets* is a set containing the sets of wffs in the *KB* 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. A knowledge state is intended to represent the knowledge that we have at a particular moment; the *KB* contains all the propositions that were received from the outside world up to that moment and the subset of their consequences that was derived so far; and the *KIS* contains information about all the sets that have been discovered to be inconsistent.

2.2 SOME INFERENCE RULES

In this section, we present some of the rules of inference of SWM*. These rules are grouped into two sets, pure logic rules and computational rules. *Pure logic rules* are like traditional rules of inference; they allow the introduction of new supported wffs into the knowledge base. *Computational rules* are rules that update the information about sets known to be inconsistent.

2.2.1 PURE LOGIC RULES

These rules correspond to traditional inference rules. They have the effect of adding new supported wffs to the *KB*. The addition of new supported wffs to the *KB* may be done in two different ways:

a new supported wff is introduced from the outside (this new supported wff is called a hypothesis) or a supported wff is derived from other supported wffs in the *KB*. Since pure logic rules add new supported wffs to the *KB*, they transform [[*KB, KIS*]] into [[*KB′,KIS*]] where $KB \subset KB'$. The following are some of the pure logic inference rules (the formal statement of all rules can be found in Martins & Shapiro 1988):

Hypothesis (Hyp). This rule enables the introduction of any supported wff as a hypothesis.

Negation Introduction (¬I). This rule states that from the hypotheses underlying a contradiction, i.e., the origin set of a proposition corresponding to a contradiction, we can conclude that the conjunction of any number of them must be false under the assumption of the others.

Implication Introduction (→I). This rule states that if the wff C was derived assuming the hypothesis H, then $H \rightarrow C$ can be derived under the assumption of the remaining hypotheses underlying that derivation of C.

Modus Ponens—Implication Elimination (MP): This rule states that if we have A and $A \rightarrow B$, then we can conclude B, B will depend on any hypotheses that either A or $A \rightarrow B$ depends on.

And Introduction (∧I). This rule states that if we have A and B, then we can derive $A \wedge B$; $A \wedge B$ will depend on any hypothesis that either A or B depends on. The resulting supported wff will have either a *der* or an *ext* origin tag, depending on whether or not A and B have the same origin set.

And Elimination (∧E). This rule enables the elimination of conjunctions. It states that if we have $A \wedge B$, then we can derive either A or B, provided that $A \wedge B$ is not "contaminated" (i.e., has no *ext* tag). Either A or B will depend on any hypothesis that $A \wedge B$ depends on.

2.2.2 COMPUTATIONAL RULES

These rules are triggered upon the discovery of inconsistent sets. They are obligatorily applied whenever a new inconsistent set is discovered. When this happens, no supported wffs are added to *KB* (as happens with pure logic rules), but rather a new set is added to *KIS*. These rules transform [[*KB,KIS*]] into [[*KB,KIS′*]] in which $KIS \subset KIS'$.

Updating of Inconsistent Sets (UIS). This rule is obligatorily applied whenever a contradiction is detected. Its effect is to update the information about the sets known to be inconsistent.

Derived Hypothesis (DH). This rule is obligatorily applied when a supported wff is derived such that there is already a hypothesis in the *KB* with the same wff and that hypothesis belongs to a known inconsistent set. The effect of this rule is to record that the hypotheses underlying the derivation of this new wff together with the remaining hypotheses in that known inconsistent set are a set known to be inconsistent.

2.3 SUMMARY

SWM* works with knowledge states, which are of the form $[[KB, KIS]]$ in which: (1) *KB* is a set of supported wffs; (2) *KIS* is a set of sets of wffs. The origin set of every supported wff in the *KB* contains wffs that correspond to hypotheses existing in the *KB*. For every wff appearing in a known inconsistent set, there is a corresponding hypothesis in the *KB*.

We define derivability within SWM* ($\vdash_{SWM^*}$) as follows: Given $[[KB, KIS]]$, we write

$$[[KB, KIS]] \vdash_{SWM^*} [[KB', KIS']]$$

if and only if there is a sequence of rules of inference of SWM* that transforms the knowledge state $[[KB, KIS]]$ into the knowledge state $[[KB', KIS']]$.

3. NONMONOTONICITY

Most computer systems only have an incomplete description of the world. In such cases, we would like to draw some conclusions that, although not entailed by the available information, are considered plausible. These plausible conclusions are suggested (not entailed) by rules that are not universal, i.e., rules that have exceptions, for example, ''Birds normally fly''. These rules are called *default rules.* A plausible conclusion drawn by using a default rule may have to be retracted in the face of new information. This kind of reasoning is called *nonmonotonic reasoning.*

SWM* does not address the nonmonotonicity problem, but there is an extension of it, SWMC (after *S*hapiro, *W*and, *M*artins, and *C*ravo), which does. In this section, we give a very brief description of the main features of SWMC. (A detailed description of SWMC can be found in Cravo & Martins 1990a, 1990b.)

One crucial feature of formalisms that allow for nonmonotonic reasoning is the ability to express default rules and exceptions to these rules. To cope with this issue, the language of SWMC is an extension of the language of SWM*. It contains a new quantifier, the default quantifier, denoted by ∇, which allows it to express default rules. It also has a distinguished 2-place predicate, *Applicable*, that, among other things, allows it to express exceptions to default rules.

The complete set of formation rules for the language of SWMC can be found in Cravo & Martins 1990a. Here we only give an example of how a default rule with an exception is expressed in SWMC. The wffs $\nabla x[Bird(x) \rightarrow Flies(x)]$ and $\forall x[Penguin(x) \rightarrow \neg Applicable(\nabla x[Bird(x) \rightarrow Flies(x)],x)]$ are intended to mean: Birds normally fly; Penguins are an exception to the previous rule.

Because SWMC is intended to support BRSs, it also works with supported wffs, and the rules of inference of SWMC also associate with each derived wff the wffs underlying its derivation. To record the fact that a derived wff is not a sound conclusion of some set of wffs, but is only a plausible conclusion of that set of wffs, there is a new type of wff, called *assumptions*, which are of the form *Applicable*(D, c), where D is a default rule and c is an individual symbol. Supported wffs corresponding to assumptions have a special origin tag, *asp*. For example, from the supported wffs $<\nabla x\ [Bird(x) \rightarrow Flies(x)], hyp, \{\nabla x[Bird(x) \rightarrow Flies(x)]\}>$ and $<Bird(Tweety), hyp, \{Bird(Tweety)\}>$, the rules of inference of SWMC allow us to infer the following supported wffs:

$$<Applicable(\nabla x[Bird(x) \rightarrow Flies(x)], Tweety), asp, \alpha>$$

$$<Flies(Tweety), der, \alpha>$$

where $\alpha = \{\nabla x\ [Bird(x) \rightarrow Flies(x)], Bird(Tweety), Applicable(\nabla x[Bird(x) \rightarrow Flies(x)], Tweety)\}$. This means that the wff *Flies(Tweety)* was derived using not only the given hypotheses, but also the assumption that the default rule is applicable to the particular individual *Tweety*. This issue is crucial when dealing with contradictions. If we now add the hypothesis

$$<\neg Flies(Tweety), hyp, \{\neg Flies(Tweety)\}>$$

we wouldn't want to conclude that the set

$$\{\nabla x[Bird(x) \rightarrow Flies(x)],\ Bird(Tweety), \neg Flies(Tweety)\}$$

is inconsistent (which would happen if we didn't associate the assumption with the wff *Flies*(*Tweety*)). All we want to conclude is that we must withdraw this conclusion in the face of the new piece of information: we can no longer assume that the default rule is applicable to *Tweety*, because we now know that *Tweety* doesn't fly.

In SWMC, three notions of consequence between a set of wffs Δ and a wff C are defined: (1) *Sound consequence*, denoted $\Delta \vdash C$, corresponds to the classical notion of consequence. (2) *Plausible con-*

sequence, denoted $\Delta \vdash_P C$, means that, given Δ, there are reasons to suppose C and no reasons against it. (3) *Conceivable consequence*, denoted $\Delta \vdash_C C$, means that, given Δ, there are reasons to suppose C but there are also reasons against it.

A semantics has been defined for SWMC, based on the classical notion of model, and a relation of preference among sets of models [Cravo & Martins 1990b]. Although several other nonmonotonic logics have been defined ([Lukaszewicz 1990] is a good overview), SWMC is, to the best of our knowledge, the only one that keeps dependencies among propositions, thus making it suitable for applications in BRS.

4. EXTERNAL BEHAVIOR

As we said at the outset, among the propositions in the knowledge base, there are some in which the BRS believes and there may be some others in which the BRS does not believe. Inputs from the outside world or reasoning carried out by the BRS may lead to the detection of contradictions, in which case the BRS has to revise its beliefs in order to get rid of the contradiction and to accommodate the new information. Up to now, we have been concerned with the definition of the rules of reasoning of a BRS. In this section, we address the issues of defining the beliefs of a BRS based on SWM* and of defining how the BRS fails to believe the consequences of a proposition that is disbelieved.

There are two approaches to defining the beliefs of a computational system, corresponding to label-based systems and context-based systems. In a *label-based system* (for example, Doyle 1979), beliefs are defined by labeling the propositions that should be considered. These labels are typically IN for believed propositions and OUT for disbelieved propositions. When a proposition is disbelieved, the BRS has to go through the knowledge base deciding what the consequences of the removal are and re-labeling propositions. In *context-based systems* (for example, de Kleer 1986), the knowledge-base retrieval function has to know which hypotheses are under consideration whenever it performs a knowledge-base retrieval operation. In context-based systems, propositions are labeled with the hypotheses underlying their derivation; it is the knowledge-base retrieval function that decides dynamically (every time it performs a knowledge-base retrieval) which propositions should be considered.

4.1 CONTEXTS AND BELIEF SPACES

We now define the behavior of an abstract context-based BRS (i.e., not tied to any particular implementation) called MBR (for Multiple Belief Reasoner): A *context* is a set of hypotheses; a context determines a *belief space*, which is the set of all hypotheses defining the context and all the wffs in the *KB* that were derived exclusively from them. A belief space is represented by $\ll[[KB,KIS]],C\gg$, in which C (a context) is a set of hypotheses, that is, a set of wffs such that for every $H \in C$ there exists $<H,hyp,\{H\}> \in KB$.

Within the SWM* formalism, the wffs in a belief space are characterized by the existence of a supported wff in the *KB* with an origin set that is contained in the context. The belief space determined by a context is the subset of all the wffs existing in the *KB* that were derived (according to the rules of inference of SWM*) from the supported wffs corresponding to the hypotheses in the context. It contains those wffs that *have been derived* in the *KB* among all possible *derivable* wffs, which, again, stresses that we are not assuming logical omniscience.

Any operation performed by MBR (query, addition, etc.) is associated with a context. We refer to the context under consideration, i.e., the context associated with the operation currently being performed, as the *current context*. While the operation is being carried out, the only propositions that will be considered are the propositions in the belief space defined by the current context. This belief space is called the *current belief space*. A proposition is said to be *believed* if it belongs to the current belief space.

4.2 DETECTION OF CONTRADICTIONS

Let us now consider how MBR acts when a contradiction is detected. We discuss two kinds of contradiction detection: contradictions within the current belief space and contradictions within a belief space strictly containing the current belief space. The main difference between them is that the former may require changes in the current context and allows the deduction of new supported wffs, while the latter leaves this context unchanged and does not allow the addition of new wffs to the knowledge base.

Suppose that we are working in the belief space $\ll[[KB,KIS]],C\gg$ and the *KB* contains the supported wff $<A \wedge \neg A,\tau,\alpha>$. Suppose, furthermore, that α does not contain any member of *KIS* (that is, α is not known to be inconsistent). In this case, one of two things will happen:

1. *The contradictory wff does not belong to the current belief space* ($\alpha \not\subset C$). In this case, the contradiction is recorded (through the application of UIS), but nothing more happens. The effect of doing so is to record that α is now known to be inconsistent.
2. *The contradictory wff belongs to the current belief space* ($\alpha \subset C$). In

this case, UIS is applied, resulting in the updating of the sets known to be inconsistent. The rule of negation introduction can be applied (generating new supported wffs in the knowledge base), and a revision of beliefs should be performed if we want to work within a consistent belief space. Since MBR only considers wffs in the current belief space, a decrease in the current context entails the removal of wffs from the current belief space. The resolution of a contradiction in the current belief space entails a *contraction* in Gärdenfors and Makinson's sense [Gärdenfors & Makinson 1988]. This contraction is performed through a family of functions R^{-}_{H}, indexed by the wff, H, to be removed:

$$R^{-}_{H}(\ll [[KB, KIS]], C \gg) = \ll [[KB, KIS]], C - \{H\} \gg.$$

From SWM*'s standpoint, after the discovery of the inconsistent set α, the removal of *any one* of the hypotheses in α is *guaranteed* to remove this contradiction from the current belief space and restore unknown inconsistency to the current context if it was not known to be inconsistent before discovery of this contradiction.

5. THE REVISION OF BELIEFS

The revision of beliefs is the ultimate task for which a belief revision system is designed. It uses all the previously discussed features in deciding about the possible culprits for a contradiction, in "removing" one of them from the knowledge base, and in changing its beliefs accordingly. The revision of beliefs is carried out through a function R^* from belief spaces into belief spaces.

$$R^*(\ll [[KB, KIS]], C \gg) = \ll [[KB, KIS]], C' \gg.$$

The effect of this function will be to remove one or more hypotheses from the context C (the culprits for the contradiction) and possibly to add some new hypotheses to the context C, generating another context, C', that is not known to be inconsistent.

No system has addressed the problem of selecting *the* culprit from the set of possible culprits for a contradiction, although some proposals have been made by Doyle 1979, Martins 1983, and others. In the actual implementation of MBR, the revision of beliefs is done by an outside system (a human) that picks the culprit(s) for the contradiction and generates the new context.

In this section, we propose an architecture whose goal is to select *the* culprit for a contradiction detected during reasoning. This task will be handled by a component that we call the *belief reviser*. We envisage the task of the belief reviser as being carried out by an organized set of communicating agents or *critics*. Each of them has expertise about a specific class of problems and supplies a tentative solution based on its own knowledge (i.e., blames the fault on a particular hypothesis). The possible set of solutions is then given to a referee that can take one of the following actions: (1) Select one of the hypotheses as the culprit, based on the suggestions received and the possible hierarchy among the critics who supplied them. This hierarchy may change according to the class of problems at hand. (2) Ask a critic the reason why some hypothesis is being selected as the culprit. This action may be used to inform the user of the system about the reason why a particular hypothesis was deleted. (3) Report failure to the user of the system and ask for help in the task of culprit selection.

We envisage each of the critics as being held responsible for its recommendations. If a decision is made, at the suggestion of some critic, to drop a hypothesis that is later on recognized as not having been responsible for the contradiction, then this critic is penalized and its future suggestions will be less important; likewise, a critic whose suggestion turns out to be profitable is rewarded, and its suggestions become more important. Details of this proposal can be found in Martins & Cravo 1989.

6. AN EXAMPLE: THE RUSSELL SET

Let us consider the hypothesis that there is a set, *s*, that contains all the sets that are not members of themselves:

$$\textit{swff}1\text{: } <\exists s\forall x[\neg(x \in x)\rightarrow x \in s)\wedge(x \in s\rightarrow \neg(x \in x))],\textit{hyp},\{\textit{wff}1\}\}>$$

We use the notation *swff*1: $<\exists s\forall x[(\neg(x \in x)\rightarrow x \in s)\wedge(x \in s\rightarrow \neg(x \in x))],\textit{hyp},\{\textit{wff}1\}\}>$ to denote that $<\exists s\forall x[(\neg(x \in x)\rightarrow x \in s)\wedge(x \in s\rightarrow \neg(x \in x))],\textit{hyp},\{\textit{wff}1\}\}>$ is a supported wff called *swff*1 and that the wff $\exists s\forall x[(\neg(x \in x)\rightarrow x \in s)\wedge(x \in s\rightarrow \neg(x \in x))]$, is represented by *wff*1.

Suppose we have a knowledge state containing just *swff*1:

$$[[\{<\exists s\forall x[(\neg(x \in x)\rightarrow x \in s)\wedge(x \in s\rightarrow\neg(x \in x))],\textit{hyp},\{\textit{wff}1\}\}>,\{\}]].$$

In the belief space defined by the context $\{\textit{wff}1\}$, we can derive *swff*2 (using the rule of existential elimination) and *swff*3 (using the rule of universal elimination—these rules were not discussed in this paper, but their use is obvious; see Martins & Shapiro 1988), where "*S*" is *the* set containing precisely those sets that do not contain themselves:

$$\textit{swff}2\text{: } <\forall x[(\neg(x \in x)\rightarrow x \in S)\wedge(x \in S\rightarrow \neg(x \in x))],\textit{der},\{\textit{wff}1\}>$$

$$\textit{swff}3\text{: } <(\neg(S \in S)\rightarrow S \in S)\wedge(S \in S\rightarrow \neg(S \in S)),\textit{der},\{\textit{wff}1\}>$$

If we now add to the knowledge base the proposition that states that "*S*" is contained in itself (*swff*4) and perform reasoning in the belief space defined by the context $\{wff1, wff4\}$, we can obtain *swff*5, *swff*6, and *swff*7:

*swff*4: $<S \in S, hyp, \{wff4\}>$

*swff*5: $<(S \in S \rightarrow \neg(S \in S)), der, \{wff1\}>$

*swff*6: $< \neg(S \in S), der, \{wff1, wff4\}>$

*swff*7: $<(S \in S \wedge \neg(S \in S)), ext, \{wff1, wff4\}>$

At this point, a contradiction is detected (*swff*7), triggering the application of UIS, which produces the knowledge state:

$[[\{swff1, swff2, swff3, swff4, swff5, swff6, swff7,\}, \{\{wff1, wff4,\}\}]]$

We can apply the rule of $\neg$I to *swff*7 to infer *swff*8:

$$swff8\text{: } < \neg(S \in S), ext, \{wff1,\}>$$

We now revise the system's beliefs by applying the following contraction:

$$\begin{aligned}R -_{wff4}([[\ll\{&swff1, swff2, swff3, swff4, swff5, swff6, swff7, swff8,\},\\ &\{\{wff1, wff4,\}\}]],\\ &\{wff1, wff4,\}\gg) =\end{aligned}$$

$$\begin{aligned}\ll[[\{&swff1, swff2, swff3, swff4, swff5, swff6, swff7, swff8,\},\\ &\{\{wff1, wff4,\}\}]],\\ &\{wff1,\}\gg\end{aligned}$$

We can perform further reasoning, generating *swff*9, by $\wedge$E applied to *swff*3; *swff*10 by MP applied to *swff*8 and *swff*9; and *swff*11 by $\wedge$I applied to *swff*8 and *swff*10:

*swff*9: $<(\neg(S \in S) \rightarrow S \in S), der, \{wff1\}>$

*swff*10: $<S \in S, ext, \{wff1\}>$

*swff*11: $<(S \in S \wedge \neg(S \in S)), ext, \{wff1\}>$

Again, UIS is applied to *swff*11, resulting in the knowledge state:

$$\begin{aligned}[[\{&swff1, swff2, swff3, swff4, swff5, swff6, swff7, swff8,\\ &swff9, swff10, swff11\}, \{\{wff1, wff4\}, \{wff1\}\}]].\end{aligned}$$

If further reasoning is to be performed in a consistent belief space, then *wff*1 (which is itself inconsistent) must be removed from the current context. In this case, the rule of $\neg$I allows us to derive the following supported wff:

*swff*12: $< \neg(\exists s \forall x[(\neg(x \in x) \rightarrow x \in s) \vee (x \in s \rightarrow \neg(x \in x))], der, \{\}>$

The supported wff *swff*12 states that there is no set, *s*, that contains all the sets that are not members of themselves. Notice that the origin set of this supported wff is the empty set, which means that it does not depend on any other wff; that is, it is a *universal truth.*

7. CONCLUDING REMARKS

Although much work has been carried out, both in AI and in philosophy, regarding how to model changes of mind or attitudes, the approaches followed in both fields present several drawbacks: Most of the work in AI, falling under the general area of Truth Maintenance Systems, merely concerns the recording of dependencies between propositions as given by an outside system, the problem solver, and has no reasoning capabilities. The work carried out by philosophers is not concerned with the computer implementation of the theories developed and, furthermore, assumes logical omniscience, i.e., all the consequences of the premises are known, which is unrealistic from a practical point of view.

We address problems relevant to both approaches: on the one hand, we want to be able to record dependencies among propositions (as with TMSs), but we want the system to be able to reason with the propositions it believes and automatically record dependencies for the new propositions it deduces. On the other hand, we want to study mechanisms for changing one's mind upon the detection of contradictions (as with the philosophical approach), but we are aware that believing all consequences of believed propositions is both unrealistic (humans don't behave that way) and impractical from a computational point of view.

The belief revision system presented here is based on a logic specifically conceived to support belief revision systems. We discussed the properties of the system independently of its implementation. An implementation of MBR is part of the SNePS system [Shapiro 1979, Shapiro & Rapaport 1987, Shapiro & Martins 1990], written in Common Lisp and running on Explorer and Symbolics Lisp Machines, Sun Stations, Macintoshes, and VAX systems at the Department of Computer Science, State University of New York at Buffalo, and at the Instituto Superior Técnico (School of Engineering of the Technical University of Lisbon, Portugal).

ACKNOWLEDGMENTS

Stuart C. Shapiro contributed the Russell Set example. Many thanks also go to John Corcoran, Ernesto Morgado, Nuno J. Mamede, J. Terry Nutter, Carlos Pinto-Ferreira, William J. Rapaport, and members of the SNePS Research Group for their criticisms and suggestions.

This work was partially supported by Junta Nacional de Investigação Científica e Tecnológica (JNICT), under Grant 87-107 and by the Instituto Nacional de Investigação Científica.

References

Anderson, Alan Ross, & Belnap, Nuel

1975 *Entailment: The Logic of Relevance and Necessity*, Vol. 1, Princeton, NJ: Princeton University Press.

Cravo, Maria R., & Martins, João P.

1990a "Defaults and Belief Revision: A Syntactical Approach", Technical Report GIA 90/02, Lisbon, Portugal: Instituto Superior Técnico, Technical University of Lisbon.

1990b "A Semantics for SWMC", Technical Report GIA 90/03, Lisbon, Portugal: Instituto Superior Técnico, Technical University of Lisbon.

de Kleer, Johan

1986 "An Assumption-Based Truth Maintenance System", *Artificial Intelligence* 28, pp. 127-162.

Doyle, Jon

1979 "A Truth Maintenance System", *Artificial Intelligence* 12, pp. 231-272.

Gärdenfors, Peter

1988 *Knowledge in Flux: Modeling the Dynamics of Epistemic States*, Cambridge, MA: MIT Press.

Gärdenfors, P., & Makison, D.

1988 "Revisions of Knowledge Systems Using Epistemic Entrenchment", *Proc. of the Second Conference on Theoretical Aspects of Reasoning about Knowledge*, M. Vardi (ed.), Los Altos, CA: Morgan Kaufmann, pp. 83-95.

Harman, Gilbert

1986 *Change in View: Principles of Reasoning*, Cambridge, MA: MIT Press.

Lukaszewicz, W.

1990 *Non-Monotonic Reasoning: Formalization of Commonsense Reasoning*, Chichester, UK: Ellis Horwood.

Martins, João P.

1983 "Reasoning in Multiple Belief Spaces", Technical Report 203, Buffalo: Department of.Computer Science, State University of New York at Buffalo.

1987 "Belief Revision", in *Encyclopedia of Artificial Intelligence*, S. C. Shapiro (ed.), pp. 58-62, New York: John Wiley and Sons.

1990 "The Truth, The Whole Truth, and Nothing But The Truth: An Indexed Bibliography to the Literature on Truth Maintenance Systems", *AI Magazine* Vol. 11, No. 5, pp. 7-25.

Martins, João P., & Cravo, Maria R.

1989 "Revising Beliefs Through Communicating Critics", Technical Report GIA 89/08, Lisbon, Portugal: Instituto Superior Técnico, Technical University of Lisbon.

Martins, João P., & Shapiro, Stuart C.

1983 "Reasoning in Multiple Belief Spaces", *Proc. of the Eighth International Joint Conference on Artificial Intelligence*, pp. 370-373, Los Altos, CA: Morgan Kaufmann.

1988 "A Model for Belief Revision", *Artificial Intelligence* 35, pp. 25-79.

Shapiro, Stuart C.

1979 "The SNePS Semantic Network Processing System", in *Associative Networks: Representation and Use of Knowledge by Computers*, N. Findler (ed.), New York: Academic Press, pp. 179-203.

Shapiro, Stuart C., & Martins, João P.

1990 "Recent Advances and Developments: The SNePS 2.1 Report", *Current Trends in SNePS—Semantic Network Processing System: Proc. of the First Annual Workshop*, D. Kumar (ed.), pp. 1-13, Lecture Notes on Artificial Intelligence 437, Berlin: Springer-Verlag.

Shapiro, Stuart C., & Rapaport, William J.
1987 "SNePS Considered as a Fully Intensional Propositional Semantic Network", in *The Knowledge Frontier: Essays in the Representation of Knowledge*, N. Cercone & G. McCalla (eds.), New York: Springer-Verlag, pp. 262-315.

Shapiro, Stuart C., & Wand, Mitchell
1976 "The Relevance of Relevance", Technical Report No. 46, Bloomington, IN: Computer Science Department, Indiana University.

Stalnaker, R.
1984 *Inquiry*, Cambridge, MA: MIT Press.