Encoding The ATMS In AGM Logic (Revised)

Simon Dixon and Norman Foo
Knowledge Systems Group
Basser Department of Computer Science
University of Sydney, NSW 2006
Australia

e-mail: simon@cs.su.oz.au, norman@cs.su.oz.au

ABSTRACT

The two main theories of justification are the foundational theory and the coherence theory. We choose as representatives of these two theories the ATMS [de Kleer 1986] and the AGM logic [Gärdenfors 1988a], and show that it is possible to simulate the ATMS in the AGM logic by encoding the justificational information as epistemic entrenchments. The ATMS context changes are performed by AGM expansion and contraction operations. The algorithm for calculating these entrenchments is presented, followed by a proof that the AGM logic with these entrenchments is behaviourally equivalent to the ATMS.

1. Introduction

Few would dispute the necessity of having a method of representing justificational information in a nonmonotonic reasoning system, but there is less agreement on what form these justifications should take. There are two distinct philosophical approaches to formalising the requirements for a belief to be justified by another belief or set of beliefs; these are called the *foundational* and *coherence* theories of justification [Pappas 1978, Gärdenfors 1989].

1.1 The Foundational Theory

Foundational reasoning is based on the concept that each fact or belief is accepted on the grounds of other beliefs which justify it. These beliefs are in turn justified by others, forming a chain of supports for each belief. Infinite chains of supports are disallowed, as are cycles in the chains, so that each chain of inferences starts

from a set of premises or assumptions. These are called the foundational beliefs, since they give support to the whole of the belief set, and yet are not supported themselves by any other beliefs.

A number of operational systems based on the foundational model have been developed. The first system of this type was the Truth Maintenance System (TMS) [Doyle 1979], which was followed by several other systems based on similar ideas, and classed loosely under the title "reason maintenance systems". Principal among these is the Assumption-based Truth Maintenance System (ATMS) [de Kleer 1986]. The ATMS was designed for problem solving in the field of qualitative physics, so functionality and efficiency were major concerns in its design. Hence we have in the ATMS an operational and reasonably efficient implementation of the foundational style of reasoning.

1.2 The Coherence Theory

The coherence theory of justification takes a different view of what constitutes a valid justification or reason to believe a proposition. A belief is considered valid on the basis of its coherence with all of the other beliefs, rather than having an explicit justification, as required by the foundational theory. In other words, a belief or fact need not have any explicit support for it to be included in the set of beliefs. The main criterion for the acceptance of a belief is that it is coherent with all or as many as possible of the other accepted beliefs. In this way, beliefs are able to justify each other in a circular fashion, so there is no concept of a foundational belief.

In addition to this, the coherence theory is usually accompanied by a principle of minimal change: when accepting a belief which is inconsistent with the belief set, the aim is to modify the belief set as little as possible whilst incorporating the new belief and maintaining a coherent set of beliefs.

The most well-developed system of this sort is the AGM logic of belief [Gärdenfors 1988a]. The AGM logic is defined by a set of rationality postulates, which are intended to capture the notion of rational change of belief. In addition to these postulates, there is an ordering on the beliefs, called the epistemic entrenchment [Gärdenfors 1988b], which ensures that the system can evaluate a unique solution within the constraints imposed by the postulates. One major drawback with the AGM logic is that the operations all produce closed theories, which can be exponentially large, even in the propositional case, and usually infinite in the first order case. A recently developed implementation [Fabian 1992] works with finite bases instead of the full theories, and satisfies most of the AGM postulates.

1.3 Motivation and Overview of the Paper

One desirable property of any reasoning system is a means of expressing explicit justificational information, and reasoning with it foundationally. We show that although the AGM logic is based on the coherence theory of justification, it is possible to use epistemic entrenchment to achieve foundational behaviour. We describe an algorithm for translating the justificational information stored by the ATMS into an entrenchment ordering on the beliefs in the AGM model. This algorithm uses the AGM expansion and contraction operations to perform ATMS context switches. We prove that this algorithm creates a system which is behaviourally equivalent to the ATMS, relative to the specification of the ATMS given in section 2.2.

The motivation for developing this algorithm was to show that by using epistemic entrenchment it is possible for the AGM logic to exhibit foundational behaviour, without the limitations of the ATMS. In the ATMS, the problem solver and the truth maintenance systems are separate, so that the rules are not part of the logic. Attempts have been made to formalise the logic of the ATMS [Reiter 1987, Selman 1990]. We do not attempt to formalise the logic of the ATMS in this paper, but we have formalised the functional behaviour of the ATMS in order to prove the correctness of the simulation algorithm. There is no corresponding problem with formalising the AGM logic, since the rules are part of the logic, so it is possible to reason about the system directly. Also, the ATMS's separation of rules from the data means that the rules cannot be altered without restarting the system, whereas the AGM logic allows the rules to be varied dynamically.

In section 2, a detailed description of the ATMS is given, including a formal specification of its behaviour. The AGM logic is outlined in section 3, followed by the algorithm for simulating the ATMS in the AGM logic in section 4, with the proof of the correctness of the algorithm in section 5. Section 6 concludes the paper with a summary of what has been achieved, and an outline of the direction of future research.

2. The Assumption-Based Truth Maintenance System

2.1 Description of the ATMS

The ATMS is an implementation of the foundational approach to modelling states of belief. Each belief must be justified by some set of beliefs, and any proposition that does not have such a justification is not accepted as a belief. The exception to this rule is the set of foundational beliefs, which require no justification at all. In the ATMS, these beliefs are called *assumptions*. For a justification to be valid, the beliefs which justify any proposition must be justified themselves, so that chains of justifications are formed.

The foundational theory places two restrictions on these chains: firstly they must be acyclic, so that no proposition can form part of its own justification, and secondly the chains must be finite, regressing to the beliefs which require no further justification. The ATMS removes the restriction that justifications be acyclic, since the "ATMS mechanism will never mistakenly use it as a basis for support" [de Kleer 1986, p.155]. Naturally, the ATMS has no way of representing infinite chains of justifications. We place one additional restriction on the ATMS: we do not allow the assumptions to be justified, in keeping with the concept of foundational beliefs. De Kleer [1986, p.147] provides a technique which "avoids ever having to justify assumptions", so this restriction does not reduce the power of the ATMS at all.

In the ATMS, each proposition is represented by a node, and is treated as atomic. Any logical relationship between nodes must be provided by the problem solver, in the form of justifications which are passed to the TMS. The justifications represent propositional Horn clauses, with the normal provability relationships, except that inconsistency is avoided by the use of *nogoods*, which have a separate data structure. Nogoods are the sets of assumptions which cannot be held simultaneously, and they are used to reduce the size of the search space, improving the efficiency of the ATMS.

The current set of assumptions is called the *environment*, and the set of facts derivable from this environment is called the *context*. An atomic proposition p is held to be true in an environment E if and only if it is a member of the current context. In this paper, ATMS provability is denoted $E \vdash_{ATMS} p$.

For a node representing a proposition to be true it must have a well-founded supporting justification, that is, a chain of justifications starting from a set of assumptions, where each of these assumptions is part of the current environment. The only other conditions under which a node is considered true are when it is an assumption, and therefore requires no justification, and when the current environment is inconsistent, in which case all nodes are accepted as true.

The implementation of the ATMS achieves a high level of efficiency by creating a static data structure in place of a theorem prover to calculate revisions of belief sets. The justifications and nogoods are "pre-compiled" into this data structure initially, and after this they are not consulted again. For each proposition p, the ATMS creates a node representing it, and associates a *label* with it, which stores the justificational information. The label represents the set of environments in which the proposition p is true. The label must be consistent, sound, complete and minimal. A label is consistent when each of its members is a consistent environment. That is, a label must not contain an environment which is a superset of any nogood set. The label for p is sound when p is derivable from each environment of the label. If every environment from which p can be derived is a

superset of some environment of its label, then the label is said to be complete. Finally, a label is minimal if no environment of the label is a superset of any other.

2.2 ATMS Functional Specification

Let E^* represent the set of all assumptions. As described above, no member of E^* may appear as the consequent of any justification. Let a and c represent atoms, and A a set of atoms. The input to the ATMS consists of a set of justifications, $J = \{(A,c) \mid (\bigwedge_{x \in A} x) \rightarrow c\}$, a set of nogood environments, $N = \{A \subseteq E^* \mid \neg \bigwedge_{x \in A} x\}$, and the current environment, $E = \{a \in E^* \mid a\}$.

$E \mid\!\!\sim_{ATMS} p$ if and only if

1. $\exists\, C \in N$ such that $C \subseteq E$, (the system is inconsistent), or
2. $p \in E$, (p is an assumption), or
3. $\exists\, (A,p) \in J$ such that $\forall\, a \in A,\ E \mid\!\!\sim_{ATMS} a$, ($p$ has a well-founded justification).

Condition 3 may be reformulated in terms of the ATMS label:

3*. $\exists\, L \in Label(p)$ such that $L \subseteq E$.

3. The AGM Logic

The AGM belief revision logic has been described at length in [Gärdenfors 1988a]. It is based on a logically closed set of beliefs, K, with the closure operator denoted *Conseq*, so that $K = Conseq(K)$. Three basic operations are central to the system: expansion, contraction and revision, denoted K_a^+, K_a^- and K_a^* respectively, where a is the sentence by which the belief set is being expanded, contracted or revised. The inconsistent belief set is denoted $K_\perp$. Expansion involves adding a new belief to the belief set, with all of its consequences. A contraction is the removal of a belief from the belief set, accompanied by the removal of sufficient other beliefs so that the belief set remains consistent and closed under logical consequence. That is, the belief being removed must not be derivable from the remaining belief set. The third operation, revision, comprises the addition of a new belief to the belief set such that any conflicting beliefs are removed from the belief set, maintaining the consistency of the system. Revision can be defined as a contraction of the negation of the belief, followed by an expansion operation; this is called the Levi identity.

The postulates for contraction and revision do not define a unique function for either operation, so it is necessary to add some further constraints, which come in the form of an epistemic entrenchment relation, which is an ordering on the members of the belief set. The entrenchment of a proposition p is denoted $Ent(p)$, and these are ordered by the normal relational operators: $<$, $\leq$ and $=$. There are 5

axioms which entrenchment orderings obey, plus a rule for linking the entrenchment to the contraction and revision functions. These are summarised below:

AGM Entrenchment Axioms

(EE1) If $Ent(A) \leq Ent(B)$ and $Ent(B) \leq Ent(C)$ then $Ent(A) \leq Ent(C)$
(EE2) If $A \vdash B$ then $Ent(A) \leq Ent(B)$
(EE3) For any A and B, $Ent(A) \leq Ent(A \wedge B)$ or $Ent(B) \leq Ent(A \wedge B)$
(EE4) When $K \neq K_{\perp}$, $A \notin K$ iff $Ent(A) \leq Ent(B)$ for all B
(EE5) If $Ent(B) \leq Ent(A)$ for all B then $\vdash A$

Expansion: $K_{\alpha}^{+} = Conseq(K \cup \alpha)$

(C-) Condition: $B \in K_{A}^{-}$ iff $B \in K$ and either $\vdash A$ or $Ent(A) < Ent(A \vee B)$

Levi Identity: $K_{\alpha}^{*} = (K_{\neg\alpha}^{-})_{\alpha}^{+}$

It has been shown [Gärdenfors 1988a] that a given contraction function uniquely defines a corresponding revision function and entrenchment ordering, and likewise a given revision function specifies the contraction function and entrenchment uniquely. Conversely, given an epistemic entrenchment ordering on the beliefs which satisfies the 5 entrenchment axioms, there are unique contraction and revision functions which satisfy their respective postulates and the (C-) condition.

4. Translation Algorithm

4.1 Constructing an Entrenchment Relation

To specify an entrenchment ordering completely requires an ordering on the dual atoms. The number of dual atoms is exponential in the number of atoms in the language, and this gives us reason to look for a partial ordering which contains sufficient information to define the contractions which interest us, and which is consistent with the entrenchment axioms. In doing this, we are not defining an alternative to the AGM entrenchment axioms, but instead we are allowing many of the entrenchment values to be unknown to us, as we will never need to consult these values to implement the AGM contraction operator. Thus the partial entrenchment relation which we define may be extended to any full entrenchment relation (as long as it is consistent with the AGM entrenchment axioms), without affecting our results.

Since we are modelling ATMS environment changes, the only beliefs to be added or contracted are ATMS assumptions, which are atomic. In fact, our algorithm contracts one atom at a time, and for the results we require, the only beliefs we

need to check for membership in K are also atomic (ATMS nodes). Then the (C-) condition implies that we only need to know entrenchment relations between each assumption and the disjunction of the same assumption with each other atom. For any atoms *a* and *b*, (EE2) constrains this relation to $Ent(a) \leq Ent(a \vee b)$, which allows two possibilities: $Ent(a) < Ent(a \vee b)$ or $Ent(a) = Ent(a \vee b)$. For this algorithm, it is sufficient to use only 5 distinct entrenchment values, denoted $E1$, $E2$, $E3$, $E4$ and $E5$, where $E1 < E2 < E3 < E4 < E5$. We denote the entrenchment of a proposition *p*, when the AGM system represents the ATMS in environment *E*, by $Ent_E(p)$. Then by (EE4), $Ent(p) = E1$ iff $p \notin K$, for consistent K. By (EE5), $Ent(p) = E5$ only if *p* is a tautology. Since ATMS rules cannot be altered once they are provided by the problem solver, they are entrenched at the next highest level of entrenchment, $E4$. The atoms in the current environment have entrenchment $E2$, and all disjunctions of pairs of atoms where at least one atom is an assumption are initially given the entrenchment $E3$.

4.2 Definitions

Let the set of all atoms be denoted Σ.

From the functional specification of the ATMS, it is possible to work backwards to a definition of the label of a node. Each environment in the label of a node *p* is a support set for *p*, so the label is a set of support sets, which we may define as follows:
$A \in Label(p)$ if and only if

1. $A \subseteq E^*$
2. $A \vdash_{ATMS} p$
3. $(\forall A' \subset A)[A' \nvdash_{ATMS} p]$
4. $(\neg \exists C \in N)[C \subseteq A]$

The first condition requires that the label contain assumptions only, that is, the foundational beliefs of the node. The second condition ensures soundness, the third minimality, and the fourth consistency. Completeness is achieved when the label contains all sets fulfilling the four conditions.

The above definition uses the ATMS provability relation to extract the internal data structure from the ATMS. However, an explicit definition is necessary to implement the following algorithm. The label update algorithm is performed as follows:

Update_Label (p)
For each justification $(B,p) \in J$
    If the label of any member of B is empty
        Continue with next justification
    For each choice of one environment from each member of B
        Let L be the union of these environments
        If L subsumes any nogood environment
            Continue with next choice
        Else if L is subsumed by any environment in *Label* (p)
            Continue with next choice
        Else if L subsumes any environments in *Label* (p)
            Remove those environments from *Label* (p)
        Add L to *Label* (p)
If *Label* (p) has changed
    For each justification $(B,q) \in J$ such that $p \in B$
        Update_Label (q)
End {*Update_Label*}

The foundational beliefs of an atom p in the environment E, denoted $FB(p,E)$, is the set of support sets of p which are held to be true. If no such set exists, then the atom has no well-founded justification. Formally:

$$FB(p,E) = \{ A \in Label(p) \mid A \subseteq E \}$$

The essential support set, $ES(p,E)$, is defined:

$$ES(p,E) = \bigcap_{X \in FB(p,E)} X$$

From the functional specification of the ATMS, an atom is considered true if and only if it is an assumption, or the environment is inconsistent, or it has a well-founded justification. In terms of the above definitions:

$$E \mid\!\!\sim_{ATMS} p \leftrightarrow (p \in E) \vee (\exists C \in N)\ [C \subseteq E] \vee (FB(p,E) \neq \varnothing)$$

Using these definitions, we can now describe the algorithm which calculates the entrenchments for the AGM logic to simulate the ATMS behaviour.

4.3 Algorithm ATMS_to_AGM

This algorithm calculates the epistemic entrenchments necessary to simulate ATMS environment change in the AGM logic via the AGM belief change operators described in section 3. The belief set K contains all of the current beliefs, and corresponds to the ATMS context. Obviously, since K is logically closed, it is a much larger set than the corresponding ATMS context, but we show that for all $p \in \Sigma$, $E \mid\!\!\sim_{ATMS} p \leftrightarrow p \in K$.

The algorithm encodes the current support relationships in the entrenchment relation according to the (C-) condition. That is, if p is an essential support for q, then the removal of p must force the removal of q, and this, according to the (C-) axiom, requires that the entrenchment relation contain $Ent(p) = Ent(p \vee q)$. Since $Ent(p) = E2$, this is achieved by setting $Ent(p \vee q) = E2$. Alternatively, if q is not dependent on p, then we have $Ent(p) < Ent(p \vee q)$, by setting $Ent(p \vee q) = E3$.

Initially, K contains the justifications and nogoods in their logical (Horn clause) form. That is, for each $(A,p) \in J$, $((\bigwedge_{x \in A} x) \rightarrow p) \in K$, and for each $C \in N$, $(\neg \bigwedge_{x \in C} x) \in K$. Also, for each of these formulas f, $Ent(f) = E4$. The initial value of Previous_Environment is $\varnothing$.

For each New_Environment

$\forall\, x \in (Previous_Environment - New_Environment)$

{ Remove all assumptions which are no longer true }

$K := K_x^-$ { AGM contraction }

{ Note that AGM logic implies that $Ent(x) := E1$ }

$\forall\, c$ such that $x \in \bigcup_{X \in FB(c, Previous_Environment)} X$

$\forall\, y \in (ES(c, New_Environment) - ES(c, Previous_Environment))$

{ Entrench new essential supports }

$Ent(y \vee c) := E2$ (i.e. $Ent(y) = Ent(y \vee c)$)

$\forall\, y \in (ES(c, Previous_Environment) - ES(c, New_Environment))$

{ Change entrenchments of old supports which are no longer essential }

$Ent(y \vee c) := E3$ (i.e. $Ent(y) < Ent(y \vee c)$)

$\forall\, x \in (New_Environment - Previous_Environment)$

{ Add all assumptions which have become true }

$K := K_x^+$ { AGM expansion }

$Ent(x) := E2$

$\forall\, y \in \Sigma,\ Ent(x \vee y) := E3$ { Default entrenchment }

$\forall\, c$ such that $x \in \bigcup_{X \in Label(c)} X$

$\forall\, y \in (ES(c, New_Environment) - ES(c, Previous_Environment))$

{ Entrench new essential supports }

$Ent(y \vee c) := E2$ (i.e. $Ent(y) = Ent(y \vee c)$)

$\forall\, y \in (ES(c, Previous_Environment) - ES(c, New_Environment))$

{ Change entrenchments of old supports which are no longer essential }

$Ent(y \vee c) := E3$ (i.e. $Ent(y) < Ent(y \vee c)$)

$Previous_Environment := New_Environment$

Note that the default entrenchment, $Ent(x) < Ent(x \vee y)$, for atomic x and y, encodes the fact that x and y are unrelated, since if x is removed from K, y is unaffected. It is the job of the problem solver, not the TMS, to point out any relationships between the various atoms, or if it is considered expedient, to refrain from doing so.

The amount of work performed by the algorithm suggests that simulating the ATMS in AGM logic is not an easy task. The reason for this is that foundational reasoning is independent of history - for example the ATMS context depends only on the current environment, and is not affected by any previous environment, whereas the AGM logic relies on a principle of minimal change when moving from one theory to the next. Thus extensive revisions of the entrenchment relation are often necessary to keep the system "effectively independent" of its previous state.

5. Theorem and Proof of Correctness

The ATMS and AGM systems are equivalent if and only if:

Theorem 1: $(\forall p \in \Sigma)\ [\ E \mid\!\!\sim_{ATMS} p \ \leftrightarrow\ p \in K\ ]$

We will prove this theorem by induction on the number of expansion and contraction operations. Firstly, it is necessary to prove two important lemmas which establish the relationship between entrenchment and the notion of essential support, and that between essential support and ATMS provability.

5.1 Lemma 1

$$Ent_E(a) \ = \ Ent_E(a \vee b) \ \leftrightarrow \ a \in ES(b,E),\ \forall a \in E,\ \forall b \in \Sigma$$

This lemma will also be proved by induction on the number of expansions and contractions. Let E represent the current environment (Previous_Environment partially updated k times), and E' the same environment updated $k+1$ times.

Initial case

Initially, $E = \varnothing$, and hence the lemma is trivially satisfied.

Inductive Hypothesis

Assume that the lemma holds for the first k expansions and contractions. Then let the $(k+1)$ st operation expand or contract E by x, giving the new environment E'.

Case (1): Expansion

Part (A): Suppose $a \in ES(b,E)$.

Then from the inductive hypothesis we have $Ent_E(a) \ = \ Ent_E(a \vee b)$.

Then if $x \notin \bigcup_{X \in Label(b)} X$, $FB(b,E) = FB(b,E')$, and so $ES(b,E) = ES(b,E')$, and $a \in ES(b,E)$. Also, the entrenchments of the disjunctions are unchanged, and hence $Ent_{E'}(a) \ = \ Ent_{E'}(a \vee b)$.

Alternatively, if $x \in \bigcup_{X \in Label(b)} X$, then we must consider two possibilities. Firstly, if $a \in ES(b,E')$ then $a \notin (ES(b,E) - ES(B,E'))$, and the entrenchments are unchanged. Hence, from the inductive hypothesis, $Ent_{E'}(a) \ = \ Ent_{E'}(a \vee b)$. Secondly, if $a \notin ES(b,E')$, then $a \in (ES(b,E) - ES(B,E'))$, and the entrenchment is changed to $Ent_{E'}(a \vee b) \ := \ E3$, giving $Ent_{E'}(a) \ < \ Ent_{E'}(a \vee b)$.

Part (B): Now consider the case when $a \notin ES(b,E)$.

Then from the inductive hypothesis we have $Ent_E(a) < Ent_E(a \vee b)$.

Then if $x \notin \bigcup_{X \in Label(b)} X$, $FB(b,E) = FB(b,E')$, and so $ES(b,E) = ES(b,E')$, and $a \notin ES(b,E)$. As above, the entrenchments are unchanged, and hence $Ent_{E'}(a) < Ent_{E'}(a \vee b)$.

Alternatively, if $x \in \bigcup_{X \in Label(b)} X$, then once more we must consider two possibilities. Firstly, if $a \in ES(b,E')$ then $a \in (ES(b,E') - ES(B,E))$, and the entrenchment of $a \vee b$ becomes $E2$, so $Ent_{E'}(a) = Ent_{E'}(a \vee b)$. Secondly, if $a \notin ES(b,E')$, then $a \notin (ES(b,E) - ES(B,E'))$, and the entrenchments are unchanged. Hence, from the inductive hypothesis, $Ent_{E'}(a) < Ent_{E'}(a \vee b)$.

Case (2): Contraction

Part (A): Suppose $a \in ES(b,E)$.

Then from the inductive hypothesis we have $Ent_E(a) = Ent_E(a \vee b)$.

Then if $x \notin \bigcup_{X \in FB(b)} X$, $FB(b,E) = FB(b,E')$, and so $ES(b,E) = ES(b,E')$, and $a \in ES(b,E)$. Also, the entrenchments involving a and $a \vee b$ are unchanged, and hence $Ent_{E'}(a) = Ent_{E'}(a \vee b)$.

Alternatively, if $x \in \bigcup_{X \in FB(b)} X$, then we must consider two possibilities. Firstly, if $a \in ES(b,E')$ then $a \notin (ES(b,E) - ES(B,E'))$, and the entrenchments are unchanged. Hence, from the inductive hypothesis, $Ent_{E'}(a) = Ent_{E'}(a \vee b)$. Secondly, if $a \notin ES(b,E')$, then $a \in (ES(b,E) - ES(B,E'))$, and $a \vee b$ is entrenched at $E3$, so that $Ent_{E'}(a) < Ent_{E'}(a \vee b)$.

Part (B): Now consider the case when $a \notin ES(b,E)$.

Then from the inductive hypothesis we have $Ent_E(a) < Ent_E(a \vee b)$.

Then if $x \notin \bigcup_{X \in FB(b)} X$, $FB(b,E) = FB(b,E')$, and so $ES(b,E) = ES(b,E')$, and $a \notin ES(b,E)$. As above, the entrenchments are unchanged, and hence $Ent_{E'}(a) < Ent_{E'}(a \vee b)$.

Alternatively, if $x \in \bigcup_{X \in FB(b)} X$, then we must consider two possibilities. Firstly, if $a \in ES(b,E')$ then $a \in (ES(b,E') - ES(B,E))$, and the entrenchment $Ent_{E'}(a \vee b) := E2$ is added, giving $Ent_{E'}(a) = Ent_{E'}(a \vee b)$. Secondly, if $a \notin ES(b,E')$, then $a \notin (ES(b,E) - ES(B,E'))$, and the entrenchments are unchanged. Hence, from the inductive hypothesis, $Ent_{E'}(a) < Ent_{E'}(a \vee b)$.

We have shown that for all cases, $Ent_{E'}(a) = Ent_{E'}(a \vee b) \leftrightarrow a \in ES(b,E')$, and hence by induction, lemma 1 holds for all assumption sets E.

5.2 Lemma 2

If E is a consistent environment and $a \neq b$ then

$$a \in ES(b,E) \quad \leftrightarrow \quad (E \models_{ATMS} b) \ \wedge \ ((E - \{a\}) \not\models_{ATMS} b)$$

Proof

Let $E' = E - \{a\}$

Case (1): $a \in ES(b,E)$

Then $FB(b,E) \neq \varnothing$, and hence $E \models_{ATMS} b$.

Also, $(\forall A \in FB(b,E))\ [a \in A]$.

Hence $(\forall A \in FB(b,E))\ [A \notin FB(b,E')]$.

Therefore $FB(b,E') = \varnothing$, and $E' \not\models_{ATMS} b$.

Case (2): $a \notin ES(b,E)$

Suppose $E \models_{ATMS} b$.

Then either $b \in E$ or $FB(b,E) \neq \varnothing$.

If $b \in E$ then $E' \models_{ATMS} b$, since $a \neq b$.

Otherwise, $\exists A \in FB(b,E)$ such that $a \notin A$, since $a \notin \bigcap_{X \in FB(b,E)} X$. Now $A \in FB(b,E')$, so $FB(b,E') \neq \varnothing$ and hence $E' \models_{ATMS} b$.

We have shown that if $a \notin ES(b,E)$, then either $E' \models_{ATMS} b$, or else $E \not\models_{ATMS} b$. Combining with case (1) completes the result.

5.3 Proof of Theorem 1

Initial case

Before any operations are performed, $E = \varnothing$. In this case, $E \models_{ATMS} a \ \leftrightarrow \exists (B,a) \in J$ such that $(\forall\, b \in B)\ [E \models_{ATMS} b]$. Since no assumptions are true, the chains of justifications must end in formulae of the form $(\varnothing,x)$, which are called *premises*.

Part (1): Suppose $E \models_{ATMS} a$. We show by induction on the length (n) of the longest chain of justifications that $a \in K$.

When $n = 1$, we have $(\varnothing,a) \in J$, which in Horn clause form is the formula a, and hence $a \in K$.

Inductive hypothesis: assume that if $E \models_{ATMS} a$ then $a \in K$ for all $n \leq k$.

Consider a proof of a with maximum length $k+1$ justifications. Then $\exists (B,a) \in J$ such that $\forall\, b \in B$, $E \models_{ATMS} b$ in at most k steps. Then by the inductive hypothesis, each b is in K. Since K is logically closed, we also have $(\bigwedge_{x \in B} x) \in K$. Combining this with $(\bigwedge_{x \in B} x) \rightarrow a$, which is also in K, since $(B,a) \in J$, we can conclude by modus ponens that $a \in K$.

We conclude by induction that for all n, $a \in K$.

Part (2): Conversely, suppose $a \in K$. Then there exists a sequence of resolutions of clauses in K from $J \cup N$ such that the final step produces the clause a. We know that no clauses from N are used in the resolution, because when one of the resolving clauses has no positive literals, the resulting clause will have no positive literals, so no such clause can be involved in a sequence of resolutions which produces a positive literal. We now show by induction on the number (n) of resolution steps that $E \mid\!\!\sim_{ATMS} a$.

Initial case: when $n = 0$, then $(\varnothing, a) \in J$, and hence $E \mid\!\!\sim_{ATMS} a$.

Inductive step: assume for all $n \leq k$ that $E \mid\!\!\sim_{ATMS} a$.
Consider a sequence of $k + 1$ resolution steps ending with the clause a. Then there exists in this sequence a clause containing the positive literal a, corresponding to some justification $(B, a) \in J$. Let this clause be $\neg b_1 \vee \cdots \vee \neg b_m \vee a$. (i.e. $B = \{b_1, \ldots, b_m\}$). Then for $1 \leq i \leq m$, $b_i \in K$, and the resolution proof for each b_i takes $\leq k$ steps, so by the inductive hypothesis $E \mid\!\!\sim_{ATMS} b_i$. But $(\{b_1, \ldots, b_m\}, a) \in J$, so by the definition of the ATMS, $E \mid\!\!\sim_{ATMS} a$.

By induction, we have shown that for all n, if $a \in K$ then $E \mid\!\!\sim_{ATMS} a$. This completes the initial case of the proof of the theorem.

Inductive step

Assume that after m expansions and contractions that $p \in K \leftrightarrow E \mid\!\!\sim_{ATMS} p$, for all $p \in \Sigma$. Consider the $(m+1)$st operation.

Case (1): Expansion by a

Let $E' = E \cup \{a\}$.

Suppose $E' \mid\!\!\sim_{ATMS} b$. Then one of the following 4 cases must hold:

(a) $E \mid\!\!\sim_{ATMS} b$

(b) $E \not\mid\!\!\sim_{ATMS} b$ and $\exists\, C \in N$ such that $C \subseteq E'$

(c) $E \not\mid\!\!\sim_{ATMS} b$ and $a = b$

(d) $E \not\mid\!\!\sim_{ATMS} b$ and $\exists\, (A, b) \in J$ such that $\forall\, x \in A,\ E' \mid\!\!\sim_{ATMS} x$

Case (a): By the inductive hypothesis, $b \in K$. Using AGM axiom (K^+3), $K \subseteq K_a^+$, we have $b \in K_a^+$.

Case (b): $E \subseteq K$, from the inductive hypothesis, since $\forall\, x \in E,\ E \mid\!\!\sim_{ATMS} x$. Also $a \in K_a^+$, by AGM axiom (K^+1). Since $K \subseteq K_a^+$ (AGM axiom (K^+3)) and $C \subseteq E'$, we have $C \subseteq K_a^+$. But since $C \in N$, $(\neg \bigwedge_{x \in C} x) \in K \subseteq K_a^+$, and by logical closure $K_a^+ = K_\perp$. Thus $b \in K_a^+$.

Case (c): By AGM axiom (K^+1), $b \in K_b^+$.

Case (d): We show that $b \in K_a^+$ by induction on the number (n) of justifications used in proving $E' \mid\!\!\sim_{ATMS} b$.

Initial case: when $n = 0$, by Cases (a)-(c), we have $b \in K_a^+$.

Inductive step: assume that for all $n \le k$ we have $b \in K_a^+$. Consider a proof using $k+1$ justifications, including a justification for b, $(A,b) \in J$. Then for all $x \in A$, $E' \vdash_{\overline{ATMS}} x$. Now if Case (a), (b) or (c) applies, we have shown $x \in K_a^+$. Alternatively, if Case (d) applies, the proof of x contains $\le k$ justifications, and hence by the inductive hypothesis $x \in K_a^+$.

Also, since $(A,b) \in J$, $[(\bigwedge_{x \in A} x) \rightarrow b\,] \in K \subseteq K_a^+$, so by closure we have $b \in K_a^+$.

Hence, by induction $b \in K_a^+$ for all n.

Conversely, assume that $b \in K_a^+$. That is, $b \in Conseq(K \cup \{a\})$.
We show by induction on the number (n) of resolutions involved in proving b from $K \cup \{a\}$ that $E' \vdash_{\overline{ATMS}} b$.

Initial case: when $n = 0$, either $b \in K$ or $b = a$. If $b \in K$, then by the inductive hypothesis of the main proof, $E \vdash_{\overline{ATMS}} b$, and hence since $E \subseteq E'$, we have $E' \vdash_{\overline{ATMS}} b$. Otherwise, if $b = a$, then $b \in E'$ and thus $E' \vdash_{\overline{ATMS}} b$.

Inductive step: assume for all $n \le k$ that $E' \vdash_{\overline{ATMS}} b$. Now consider a resolution proof of b requiring $k+1$ steps. This procedure uses a formula from J of the form $\neg r_1 \vee \cdots \vee \neg r_m \vee b$, where $r_i \in K_a^+$, $1 \le i \le m$. For this to resolve to b, there are resolution proofs for all r_i, $1 \le i \le m$, with each of these procedures requiring $\le k$ steps. Hence by the inductive hypothesis $E' \vdash_{\overline{ATMS}} r_i$, $1 \le i \le m$. Also $(\{r_1, \ldots, r_m\}, b) \in J$, so by the definition of the ATMS, $E' \vdash_{\overline{ATMS}} b$.

Hence for all n, $E' \vdash_{\overline{ATMS}} b$, and thus we have shown that $E' \vdash_{\overline{ATMS}} b \leftrightarrow b \in K_a^+$.

Case (2): Contraction by a

Let $E' = E - \{a\}$.
Suppose $E' \vdash_{\overline{ATMS}} b$. Then $E \vdash_{\overline{ATMS}} b$, since $E' \subseteq E$, and hence by the inductive hypothesis, $b \in K$. Then by Lemma 2, $a \notin ES(b,E)$. From (EE2) we have $Ent_E(a) \le Ent_E(a \vee b)$, and using Lemma 1, this result strengthens to $Ent_E(a) < Ent_E(a \vee b)$. Finally, from the (C-) axiom, we have $b \in K_a^-$.

Conversely, suppose that $b \in K_a^-$. Then by the (C-) axiom, $Ent_E(a) < Ent_E(a \vee b)$, and by Lemma 1, $a \notin ES(b,E)$. We also know that $b \in K$, since $K_a^- \subseteq K$. So $E \vdash_{\overline{ATMS}} b$, by the inductive hypothesis, and then by Lemma 2 we conclude that $E' \vdash_{\overline{ATMS}} b$.

Thus we have shown that $\forall b \in \Sigma$, $E' \vdash_{\overline{ATMS}} b \leftrightarrow b \in K_a^-$, after the $(m+1)$st operation.

Therefore, by induction, $p \in K \leftrightarrow E \mid\!\!\sim_{ATMS} p, \forall p \in \Sigma$, and the algorithm correctly chooses an entrenchment so that the behaviour of the AGM system is equivalent to that of the ATMS.

6. Conclusions and Further Research

We have shown that the ATMS can be simulated in the AGM logic by using a suitable epistemic entrenchment relation to encode the foundational nature of the beliefs, and using contraction and expansion operations to perform environment changes. The ATMS encoding does not specify a complete entrenchment relation, but calculates the class of entrenchments for which the AGM's behaviour with respect to atoms is equivalent to the ATMS. Thus the coherence-based AGM logic is able to mimic the behaviour of a foundational system, the ATMS. This illustrates an advantage of the AGM logic with entrenchment over the many purely foundational systems: it is possible to express all types of justificational information using epistemic entrenchment. Current research involves using epistemic entrenchment to model non-foundational justifications, and we are examining the applications of both types of reasoning. The research also includes the implementation of an AGM belief revision system.

There are several advantages of a system based on this approach. The rule base, which is static in the ATMS, is part of the logic in the AGM system, so it is possible to allow rule revision. In order to simulate the behaviour of the ATMS, the rules were all placed at the highest allowable entrenchment, immediately below the tautologies. But there is no reason why the rules should not be given lower entrenchments so that they could be altered if necessary. One possible application of rule revision would be to model default rules and exceptions, by creating a stratified rule base, with various sets of rules being placed at different levels of entrenchment. Another advantage is that the AGM logic is a formal system, so it is possible to reason about the system, which is not true of the ATMS. Another extension of this work under current research is a coherence-based truth maintenance system, which may be implemented elegantly in the AGM logic.

7. References

[de Kleer 1986]
de Kleer, J. (1986) *An Assumption-Based TMS*, Artificial Intelligence 28, pp. 127-162.

[Doyle 1979]
Doyle, J. (1979) *A Truth Maintenance System*, Artificial Intelligence 12, pp. 231-272.

[Fabian 1992]
Fabian, R.A. (1992) Personal communication.

[Gärdenfors 1988a]
Gärdenfors, P. (1988) *Knowledge in Flux: Modeling the Dynamics of Epistemic States*, Bradford Books, MIT Press, Cambridge, MA.

[Gärdenfors 1988b]
Gärdenfors, P. and Makinson, D. (1988) *Revisions of Knowledge Systems Using Epistemic Entrenchment*, Proc. of Second Conf. on Theoretical Aspects of Reasoning About Knowledge, pp. 83-95, ed. M.Y. Vardi, Morgan Kaufmann.

[Gärdenfors 1989]
Gärdenfors, P. (1989, first draft) *The Dynamics of Belief Systems: Foundations vs. Coherence Theories*, Department of Philosophy, Lund University.

[Pappas 1978]
Pappas, G.S. and Swain M. (1978) *Essays on Knowledge and Justification*, pp. 30-35 and 184-308, Cornell University Press.

[Reiter 1987]
Reiter, R. and de Kleer, J. (1987) *Foundations of Assumption-Based Truth Maintenance Systems: Preliminary Report*, AAAI-87, pp. 183-188, Morgan Kaufmann.

[Selman 1990]
Selman, B. and Levesque H.J. (1990) *Abductive and Default Reasoning: A Computational Core*, AAAI-90, MIT Press.