A Modal Logic for Reasoning about Belief

Guo-Qiang Zhang, Chang Huang
Department of Computer Science
University of Georgia
Athens, Georgia 30602
Email: gqz@cs.uga.edu

William C. Rounds
Artificial Intelligence Laboratory
University of Michigan
Ann Arbor, Michigan 48109
Email: rounds@eecs.umich.edu

Abstract

The purpose of this paper is to integrate a number of ideas from knowledge representation, with ideas from the semantics of progamming languagse, in a basic logical system for belief. A knowledge-based system can be considered as, in a simplified way, an agent which, based on its current state of information, makes inference for purposes such as planning or taking an action that best suits the agent's goal. There are different ways to formalize such a system. The approach we are taking is model-checking: we express an agent's current state of knowledge by a semantic model, and the agent's belief or action as some logical formulas. The paradigm of model-checking deals with the issue of whether an agent's current state of knowledge supports an action or a belief. Only supported belief or action are considered as appropriate. This paradigm is, in particular, different from the traditional logical one which represents an agent's state of knowledge by a set of first-order formulas, and considers a belief or an action appropriate if it happens to be a logical consequence of the set of given formulas.

The model-checking paradigm uses a modal logic which has the following characteristics:

- *There is a modal operator to express explicit belief.*
- *The logic is partial. A formula can have three modes with respect to a model: approved (true), disapproved (false), or unknown.*
- *The specification of a model relies on techniques from domain theory, as in the semantics of programming languages.*
- *Reiter's notions of defaults and extensions are taken as concrete ways to generate a certain class of possible worlds for our model.*
- *Nonmonotonic reasoning can be modeled in this MONOTONIC logic.*
- *The logical system is well behaved – it has all the expected properties, such as monotonicity, soundness, completeness, and compactness.*

1 Introduction

There is a general consenses in the knowledge representation community that the ability to do nonmonotonic reasoning holds the key to the effectiveness of knowledge based systems. One of the well-studied formalizms for non-monotonic reasoning is default logic [15], augmenting first-order logic by certain extended rules. The basic idea of default logic is to encode common knowledge in the form of "rules of thumb", called default rules. Examples of such rules include *birds fly*, *smoke means fire*, or, in general, *events occur as planned*.

Rules of thumb, however, certainly do not have the power of logical entailment – there can be exceptions. Ommiting the enumeration of exceptions not only saves space. Conceptually, it is just not possible to list all potential exceptions. For instance, in the "Yale shooting" example, there are infinitely many possibilities for a loaded gun at time t to not fire at time $t+2$: the gun may be jammed; someone may have unloaded the gun; an earthquake may have occurred at the scene, etc. So, not all rules of thumb can be faithfully encoded as material implication in a finite form.

Reasoning in default logic is conducted by building extensions. In a simplified way, an extension of a theory is a super-set of the theory with certain conclusions from the default rules added. There are two basic modes: credulous reasoning and skeptical reasoning. Credulous reasoning considers a formula a nonmonotonic consequence of a base theory if the formula happens to be in one of the extensions of the base theory. Skeptical reasoning, on the other hand, considers a formula a nonmonotonic consequence of a base theory if the formula belongs to all the extensions of the base theory.

However, the following properties of default logic are often considered undesirable:

- There can be multiple extensions.

- Extensions may not exist.
- Even when they do exist, it can be too costly or impossible to find them.
- Default logic does not support the familiar principle of reasoning by cases [2, 10].
- Standard entailment in default logic fails to have the so-called cumulativity property [5].

The purpose of this paper is to integrate a number of ideas from knowledge representation, with ideas from the semantics of programming languages, in a basic logical system for belief. We hope that in the process of this investigation, some of these conceptual problems will be resolved, or at least relieved.

Our general perspective is this. We believe that human reasoning is in large part model-driven, in that inferences are made about mental "pictures" or "models", as convincingly demonstrated in the works of congnitive scientist Johnson-Laird [8]. The idea of model-based nonmonotonicity is that from evidence garnered from one of these partial mental models, one applies prestored "constaints" or "rules", in order to fill in information which may be missing. This can be done in a nondeterministic manner, so that the filling-in process can be completed in more than one way, resulting in new pictures (generally inconsistent with each other), about which one can then reason, plan, or act. Nonmonotonicity arises when one is forced to alter these filled-in pictures in light of new actual evidence.

We propose a new logic for belief based on this view. There have been, of course, many modal logics in the literature for the purpose of reasoning about belief (or knowledge) [6]. Our starting point differs from these in a fundamental way, in that we do not start with axioms and proof rules, but rather start with semantic models, and let the model theory guide our way to a proof system. Thus, questions like whether or not the logic is K45 are not the issue, at least at the beginning. Because the logic is entirely determined by the semantics, it is called *semantics-based*.

The semantics-based approach is a new technique first used in domain logics for programming languages [1, 23]. For this approach to work, however, there has to be a well-established model theory for the special case of default reasoning. Fortunately, such a theory, called *default domain theory* (DDT), have already been developed by the first two authors over the last couple of years [18, 19, 21, 22, 17, 16]. We summarize some of its key ideas.

- Reiter's default rules can be reassigned a *semantic* role. Instead of using defaults as extended proof rules, we put them in the semantic space, as instructions for generating a certain class of possible worlds as in Kripke structures.
- Scott's information system [20] can be extended as a semantic structure for default reasoning. The notion of a "base theory" in default logic is replaced by the notion of a "base information state" (as in information systems). The notion of "extension" is retained, but now refers not to a collection of theories, but to a collection of information states, each containing more information than the base state.

We will consider a special class of default domains, called *default coherent spaces*, in this paper. It is the basis for the semantics of a modal logic, which will be shown to have the following characteristics:

- There is a modal operator to express explicit belief.
- The logic is partial. A formula can have three modes with respect to a model: approved (true), disapproved (false), or unknown.
- The specification of a model relies on techniques from domain theory, as in the semantics of programming languages.
- Reiter's notions of defaults and extensions are taken as concrete ways to generate a certain class of possible worlds for our model.
- Nonmonotonic reasoning can be modeled in this *monotonic logic.*
- The logical system is well behaved – it has all the expected properties, such as monotonicity, soundness, completeness, and compactness.

The rest of the paper is organized as follows. In Section 2 we review some background results from default domain theory, reviewing in particular the class of default coherent spaces. In Section 3 we present our modal logic for belief and its proof system. Section 4 illustrates how our semantics-based modal logic can model nonmonotonic reasoning, without getting into some of the traditional conceptual problems. Examples are discussed. Section 5 studies properties of the modal logic and establishes soundness, compactness, and completeness results, as expected of any ordinary modal logic.

2 Default coherent spaces

The logic we are concerned with has a modal operator B, for belief (We will be mostly concerned with a single agent in this paper, so the modality is not parameterized over agents). To reason about what happens in a base information state x and its extensions, we let $\mathsf{B}\varphi$ holds in x if φ holds in all extensions of x. The purpose of this section is to give a brief review of default domain theory, making precise what we mean by a "model", a "state", or an "extension". More abstract forms of default domain theory are available elsewhere [22]; this section tries to give a simple, intuitive exposition to it. It also makes the proof of compactness of our logic simpler.

Definition 2.1 *A coherent space is a pair*

$$\underline{A} = (A, \smile),$$

where A is a countable set called the tokens, and $\smile$ is the coherence relation, a reflexive, symmetric relation on A (i.e., $\smile \subseteq A \times A$ is such that for each $a \in A$, $a \smile a$, and $a \smile b$ implies $b \smile a$).

Thus, coherent spaces are exactly countable, symmetric and reflexive graphs. They are given this name in [13] because of the order-theoretic properties of these structures. To relate these structures to partial orders, there is a concept called *information states* or simply *states*. These are exactly the cliques of the graph.

Definition 2.2 *A state of a coherent space*

$$\underline{A} = (A, \smile)$$

is a subset x of A such that $a \smile b$ for every a, b in x. We write $|\underline{A}|$ for the set of states of $\underline{A}$.

Intuitively, tokens of a coherent space correspond to basic units of information. The coherence relation indicates which of the tokens are not in conflict with each other, so that they can be considered as information about a single object.

This interpretation can be convincingly demonstrated by considering a database admitting partial information. Any such a database is in fact a coherent space. Given a set L of attributes and a set V_l of values for each attribute $l \in L$, we get a coherent space $(A, \smile)$, where $A := \bigcup_{l \in L} \{l\} \times V_l$, and

$$(l_1, v_1) \smile (l_2, v_2) \iff l_1 \neq l_2 \text{ or } v_1 = v_2.$$

The intuition here is that each token (l, v) represents an atomic unit of information; two such tokens clash exactly when they assign different valuses to the same attributes. So, two tokens are coherent exactly when they are talking about different attributes, or when they do talk about the same attribute, the values are the same.

Under this interpretation, a (partial) record corresponds to an information state, and a database corrresponds to a finite set of pairwise incompatible information states. (See [4, 9] for an extensive study of the relationship between partial information in databeses and domain theory.)

Definition 2.3 *A default coherent space is a tuple*

$$\underline{A} = (A, \smile, \Delta),$$

where $(A, \smile)$ is a coherent space, Δ is a set of rules of the form $\dfrac{X : a}{a}$, with X a finite state, and a a token from A. If for all rules $\dfrac{X : a}{a}$ in Δ, X is empty, we call the default coherent space precondition-free. We write a precondition-free default rule simply as $\dfrac{: a}{a}$.

For simplicity, we consider the so called normal default rules in this paper. But our approach works for general defaults as well. A rule like $\dfrac{X : a}{a}$ is called a default rule, which intuitively means that if X is part of the current information state, and a is compatible with the current state of information, then it can be added to the information state. This vague interpretation can be made precise by the following definition, originated from Reiter [15].

Definition 2.4 *Let $(A, \smile, \Delta)$ be a default coherent space, and x a member of $|\underline{A}|$. For any $S \in |\underline{A}|$, define $\Phi(x, S)$ to be the union $\bigcup_{i \in \omega} \phi(x, S, i)$, where*

$$\begin{aligned} &\phi(x, S, 0) = x, \\ &\phi(x, S, i+1) = \phi(x, S, i) \\ &\quad \cup \{a \mid \frac{X : a}{a} \in \Delta \ \&\ X \subseteq \phi(x, S, i) \\ &\qquad\qquad \&a \smile b \text{ for every } b \text{ in } S\}. \end{aligned}$$

Call y an extension of x if $\Phi(x, y) = y$. In this case we also write $x \epsilon_{\underline{A}} y$, with the subscript omitted most of the time.

Here are some useful properties of the extension relation on a default coherent space $(A, \smile, \Delta)$. (For a proof of these, see [18].)

Theorem 2.1

- *Extensions always exist.*

- *If $x \epsilon y$ then $y \supseteq x$.*
- *$x \epsilon y$ and $y \epsilon z$ implies $y = z$.*
- *If $x \epsilon y$ and $x \epsilon y'$ then either $y = y'$ or $a \not\smile a'$ for some $a \in y$ and $a' \in y'$.*

3 System B

We now turn to our propositional modal language for reasoning about default models and extensions. Fix U, an infinite set of atomic formulas. We assume that each atomic formula has a syntactic *dual token* $\overline{a}$, with $\overline{\overline{a}} =_{def} a$. The existence of the dual token is not crucial; but we want to call readers' attention to the possiblity of modelling different kinds of negations in our system. The language $\mathcal{L}(\mathsf{B})$ is generated by the grammar

$$\varphi ::= \mathbf{t} \mid \mathbf{f} \mid \sigma \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathsf{B}\varphi,$$

where $\sigma \in U$. Strings generated by the above grammar are called *formulas* in the language $\mathcal{L}(\mathsf{B})$. We think of members of U as propositional "variables", though in practice they could have a complex infrastructure.

Remarks on the language:

1. The dual token $\overline{a}$ is distinct in meaning from the formula $\neg a$. $\neg a$ means we don't have the piece of information a, but $\overline{a}$ means something stronger: we have the evidence that a will never be realized.
2. The symbols $\mathbf{t}$ and $\mathbf{f}$, standing for true and false, are not tokens in U.
3. As usual, by a *literal* we mean a token a or $\overline{a}$ or a negation of one of these, or $\mathbf{t}$ or $\mathbf{f}$.
4. We introduce other formulas and modalities by abbreviation: $(\varphi \to \psi) = \neg\varphi \vee \psi$ and $\mathsf{M} = \neg\mathsf{B}\neg$.

The formulas $\mathsf{B}\varphi$ and $\mathsf{M}\varphi$ are read "φ is believed" and "φ is believable" respectively. The meaning of $\mathsf{B}\varphi$ is that with respect to an information state w, φ holds in all z such that $w \epsilon z$. The meaning of $\mathsf{M}\varphi$ is $\neg\mathsf{B}\neg\varphi$, as usual. So the B and M modalities only look at the preferred supersets (extensions) of a given set.

Now we give the semantics of the modal formulas using default coherent spaces introduced in the previous section.

Definition 3.1 *Given a default coherent space $\underline{A} = (A, \smile, \Delta)$, let x and y range over information states of $(A, \smile)$. The relation $\models_{\underline{A}}$ is the smallest relation between the information states and formulas in $\mathcal{L}(\mathsf{B})$ satisfying the following (The subscript in $\models_{\underline{A}}$ will be omitted most of the time).*

- *$x \models_{\underline{A}} \mathbf{t}$ always and $\mathbf{f}$ never;*
- *$x \models_{\underline{A}} \sigma$ if the atomic formula $\sigma \in x$;*
- *$x \models_{\underline{A}} \varphi \vee \psi$ iff $x \models_{\underline{A}} \varphi$ or $x \models_{\underline{A}} \psi$;*
  Similarly for $\wedge$;
- *$x \models_{\underline{A}} \neg\varphi$ iff not $x \models_{\underline{A}} \varphi$;*
- *$x \models_{\underline{A}} \mathsf{B}\varphi$ iff for all y such that $x \epsilon y$, we have $y \models_{\underline{A}} \varphi$.*

According to the definition of the "satisfaction" relation, it is appropriate to call a pair $(x, \underline{A})$ a *model*, where $\underline{A}$ is a default coherent space, and x is an information state of it. We say a model $(x, \underline{A})$ *supports* a formula φ just in case we have $x \models_{\underline{A}} \varphi$. In this case we also say φ is true in $(x, \underline{A})$.

In the study of logic, one is interested in effective ways to generate all "theorems" – formulas which are true in all models. An effective method is often presented in terms of a collection of "axioms" and "inference rules", forming a proof system. A proof system is *sound* if any formula generated (provable) from the system is a theorem. It is called *complete* if all theorems can be generated from the system.

Proof systems can be presented in different styles, such as Hilbert's axioms, Gentzen's sequences, or the semantic tableu. We present our proof system in the equational style (similar to Boolean algebra; although we are dealing with a modal logic instead of a propositional logic). The equational axioms/rules include $B1 - B10$ given below.

The standard Boolean equational laws;
$B1$. $\mathsf{B}\mathbf{f} = \mathbf{f}$, $\mathsf{B}\mathbf{t} = \mathbf{t}$, $\overline{a} = \overline{a} \wedge \neg a$;
$B2$. $\varphi = \varphi \wedge \mathsf{B}\varphi$, where φ is negation-free ;
$B3$. $\mathsf{B}\mathsf{B}\varphi = \mathsf{B}\varphi, \mathsf{M}\mathsf{M}\varphi = \mathsf{M}\varphi, \mathsf{B}\mathsf{M}\varphi = \mathsf{B}\varphi, \mathsf{M}\mathsf{B}\varphi = \mathsf{M}\varphi$;
$B4$. $\mathsf{B}(\varphi \vee (\psi \wedge \mathsf{B}\rho)) = \mathsf{B}(\varphi \vee (\psi \wedge \rho))$;
$B5$. $\mathsf{B}(\varphi \vee (\psi \wedge \neg\mathsf{B}\rho)) = \mathsf{B}(\varphi \vee (\psi \wedge \neg\rho))$;
$B6$. $\mathsf{B}(\bigwedge \varphi_i) = \bigwedge \mathsf{B}\varphi_i$;
$B7$. $\mathsf{M}(\bigvee \varphi_i) = \bigvee \mathsf{M}\varphi_i$;
$B8$. $\mathsf{M}\varphi \wedge \mathsf{B}\psi = \mathsf{M}(\varphi \wedge \psi) \wedge \mathsf{B}\psi$;
$B9$. $\mathsf{B}(\varphi \vee \psi) = \mathsf{B}(\varphi \vee \psi) \wedge (\mathsf{M}\varphi \vee \mathsf{B}\psi)$.

A formula $\varphi = \psi$ reads: φ is *equivalent* to ψ. Recall also that we regard M as abbreviating $\neg\mathsf{B}\neg$. In addition to the above equational laws, we also have a metarule $B10$, as part of $\mathbf{B}$, which allows the substitution of a formula by an equivalent one in any context. An often used instance of this metarule is $\varphi = \mathbf{t}$ if and only if $\neg\varphi = \mathbf{f}$.

The axioms in **B** are not independent of each other. In fact, it can be shown that $B7$ and $B9$ are derivable from the rest. With respect to these equational axioms, a formula φ is a theorem if we can prove that $\varphi = \mathbf{t}$.

It is easy to demonstrate that **B** is different from modal logics such as S_4 or S_5 for necessity and possibility. Our modal operator B does not satisfy axioms such as $\mathsf{B}\varphi \rightarrow \varphi$. It is also different from any logics for reasoning about beliefs we have seen so far. Other systems for beliefs, at least those about which we are aware, tend to equate B with L, an operator for knowledge. One exception is Meyer and van der Hoek [11] who introduce a P (preferred) operator. However, this operator is accorded a $K45$ Kripke semantics, in which, if $\neg\varphi$ is true, then it is believed. So for any φ in these systems, $\varphi \rightarrow \mathsf{B}\varphi$ is an axiom, while we only allow negation-free φ's in $B2$. This reflects our interpretation of the negation $\neg\varphi$ as "it is not the case that φ is known", or "it is not the case that we have knowledge about φ", according to our current state of affairs. Therefore, "$\neg\varphi$" by no means invites us to believe $\neg\varphi$, because in the absence of real knowledge about φ, we may well choose to believe φ.

4 Comparison

We are now in a position to discuss some of the special characteristics of our logic.

4.1 Reasoning by cases

To illustrate the advantage of our logical system, especially how default reasoning is carried out within the system, we apply our framework to a typical (toy) example for which *reasoning by cases* fails in Reiter's default logic.

Consider a circuit wired in such a way that

- The light will normally go on if switch s_1 is closed.
- The light will normally go on if switch s_2 is closed.

We can use default rules $\frac{s_1 : \mathtt{light}}{\mathtt{light}}$ and $\frac{s_2 : \mathtt{light}}{\mathtt{light}}$ to express these. What if we know only that s_1 or s_2 is closed? Using Reiter's default logic, from $s_1 \vee s_2$ one cannot prove s_1, and one cannot prove s_2 either. So one cannot apply either of the default rules.

However, in the model-checking paradigm using our belief system, the relevant default coherent space has three tokens: s_1, s_2 and `light`, all coherent with one another. The default set Δ contains two rules, as mentioned above. For any information state x in this coherent space, one can check that it is true that

$$x \models [(s_1 \vee s_2) \rightarrow \mathsf{B}\,\mathtt{light}],$$

which means no matter what state x is, one can believe the light will go on from the fact that one of the two switches is closed. This matches our intuition well.

The difference between model-checking and theorem proving boils down to difference between the "satisfaction" relation and the "logical consequence" relation. From $x \models \varphi$, we cannot derive the truth of $\bigwedge x \rightarrow \varphi$, even when x is finite. A simple example can illustrate this. While we have

$$\{\mathtt{Bird(Tweety)}\} \models \neg\mathtt{Dead(Fred)},$$

meaning that having "Tweety is a bird" as a fact does not support the claim that "Fred is dead", it is absurd to have

$$\mathtt{Bird(Tweety)} \rightarrow \neg\mathtt{Dead(Fred)}.$$

4.2 Possible worlds

Here is how our approach gets around the problem of multiple extensions. When extensions are regarded as partial (possible) worlds, the extension relation is similar to the accessibility relation in Kripke structures. A default rule functions here not as an extended proof rule, but as part of a procedure for building an agent's preferred worlds extending the current one. There can be many different worlds reachable from the current world. So the possibility of multiple extensions becomes a feature, not a bug – Kripke structures would be rather uninteresting if there were only one world accessible from the current one.

When defaults are regarded as a constructive method for building worlds, we can investigate different model building procedures. Reiter's extension operator, when phrased model-theoretically, remains one of the key "algorithms" for building preferred worlds. However, extensions may not exist in some reasonable cases. To cope with this, we have introduced the notion of a "dilation" [18], a robust generalization of the notion of an extension. Dilations exist in all reasonable cases.

4.3 Monotonicity versus nonmonotonicity

Clearly, the set-up of our logic completely follows the traditional Tarskian style. There is nothing abnormal about this logic: it is sound, complete, and compact, as will be established in the next section.

Still, nonmonotonicity can appear in our framework in two possible ways. One is that if a model $(x, \underline{A})$ supports a formula φ, it does not mean that in a model $(y, \underline{A})$ with more information (i.e. $x \subseteq y$), the formula will still be supported. Consider, for example, the default coherent space $(\{a, s, d\}, \smile, \Delta)$, where $a \smile s$, $s \smile d$, but $a \not\smile d$, and $\Delta = \{\frac{s : a}{a}\}$. We can interpret s as, with respect to a particular flight, "scheduled to arrive at noon"; a as "arriving at noon"; and d as "delayed". We have $\{s\} \models \mathsf{B}\, a$, since a belongs to $\{s, a\}$, the only extension of $\{s\}$. But we do not have $\{s, d\} \models \mathsf{B}\, a$, since a does not belong to $\{s, d\}$, the only extension of itself. This makes perfect sense: if a flight is scheduled to arrive at noon, we can tentatively believe that it will. However, with the additional information that the flight is delayed (we might not have had this information earlier), we can no longer believe that the flight will arrive at noon.

More abstract notions of nonmonotonic entailment can also be studied in our framework. In [21], for example, we considered basic properties of nonmonotonic entailment. Let φ and ψ be formulas. With respect to a given default structure, a formula φ nonmonotonically entails a formula ψ, if ψ is satisfied in every extension of any informationally minimal partial model of φ. We discovered that, cautious monotony, due to Gabbay [12], fails in general for our notion of default nonmonotonic entailment. However, there are various sufficient conditions which guarantee the property of cautious monotony.

5 Soundness, completeness, and compactness

In this section we establish some basic properties of System **B**, such as soundness, completeness, and compactness. No new proof principles are introduced. However, since modeling nonmonotonic reasoning is not traditionally a key motivation behind classical logical systems, it is important to show that our logical framework retains all the expected properties.

5.1 Soundness

Some terminology is now in order. If $\varphi = \psi$ is provable in **B**, we write $\vdash \varphi = \psi$. A theorem of **B** is a formula φ such that $\vdash \varphi = \mathbf{t}$. In this case we also write $\vdash \varphi$. Let $[\![\varphi]\!]$ denotes the set (class) $\{(x, \underline{A}) \mid x \models_{\underline{A}} \varphi\}$. A formula φ is satisfiable if $[\![\varphi]\!] \neq \emptyset$. Otherwise φ is unsatisfiable. A formula $\varphi = \psi$ is valid if $[\![\varphi]\!] = [\![\psi]\!]$.

Proposition 5.1 *$\varphi = \psi$ is valid iff*

$$[\![(\varphi \to \psi) \wedge (\psi \to \varphi)]\!] = [\![\mathbf{t}]\!].$$

$\varphi \to \psi$ is valid iff $\varphi = \varphi \wedge \psi$ is valid, which is in turn equivalent to saying that $\varphi \vee \psi = \psi$ is valid.

By induction on the axioms and rules, we can establish the soundness of System **B**. However, note that the proof depends on the (nontrivial) properties of extensions mentioned at the end of Section 2.

Theorem 5.1 (Soundness) *If $\vdash \varphi = \psi$, then*

$$[\![\varphi]\!] = [\![\psi]\!].$$

5.2 Completeness

As an illustration of how to use the rules of System **B**, we present the following proposition.

Proposition 5.2

$$\mathsf{B}(\varphi \to \psi) \to (\mathsf{B}\varphi \to \mathsf{B}\psi);$$
$$\mathsf{B}(\varphi_1 \vee \varphi_2 \vee \cdots \vee \varphi_n) \to \mathsf{M}\varphi_1 \vee \mathsf{M}\varphi_2 \vee \cdots \vee \mathsf{M}\varphi_n;$$
$$(a_1 \vee a_2 \vee \cdots \vee a_n) \to \mathsf{B}(a_1 \vee a_2 \vee \cdots \vee a_n)$$

are theorems of B, where the a_i's are tokens.

We now prove a normal form theorem for System **B**, which is the first step toward the completeness of the system.

Lemma 5.1 (Modal Depth) *Every formula φ is equivalent to a formula of modal depth ≤ 1, where the modal depth of a formula is the maximal number of nestings of modalities in the formula.*

As far as complexity is concerned, one can, in linear time, simply delete all the modal operators B inside the scope of a modal operator B and the resulting formula will be equivalent to the original one. The reason is that the procedure of bringing to disjunctive normal form can be reversed, to bring a disjunctive normal form back to the original formula. After deleting the B's and applying the reverse transformation, the only change is that some B's have disappeared.

Definition 5.1 *A formula is* pseudo-atomic *if it is a literal, or $\mathsf{B}(\bigvee a_i)$, where each a_i is a literal, or $\mathsf{M}(\bigwedge b_j)$, where each b_j is a literal.*

Lemma 5.2 (Pseudo-DNF)
Every formula is equivalent to a disjunction of conjunctions of pseudo-atomic formulas.

Let φ be a conjunction of literals. We write φ^+ for the set of tokens that appear as conjuncts of φ. For example, if φ is $a \wedge b \wedge \neg c \wedge \neg d$, where a, b, c, d are tokens, then $\varphi^+ = \{a, b\}$.

Lemma 5.3 (Main Satisfiability Lemma) *Let*

$$\mathsf{M}\varphi_1 \wedge \mathsf{M}\varphi_2 \wedge \cdots \wedge \mathsf{M}\varphi_n$$

be a formula such that each φ_i is a satisfiable conjunction of literals. Then there is a pre-condition free default structure with exactly n extensions y_i $(1 \leq i \leq n)$ of $\emptyset$, such that for each extension y_i, $y_i \models \varphi_i$. As a consequence, $\mathsf{M}\varphi_1 \wedge \mathsf{M}\varphi_2 \wedge \cdots \wedge \mathsf{M}\varphi_n$ is satisfiable.

Theorem 5.2 (Decidability and Completeness) *The satisfiability of a formula is decidable. Moreover, if $\varphi = \psi$ is valid, then it is a theorem of System* **B**.

To decide the satisfiability of any formula ψ in System **B**, we first bring ψ to pseudo-disjunctive normal form, where each disjunct looks like

$$\varphi_1 =_{def} \bigwedge a_i \wedge \bigwedge \neg b_j \wedge \bigwedge \mathsf{B}\alpha_m \wedge \bigwedge \mathsf{M}\beta_n,$$

with each α_m a disjunction of literals, and β_n a conjunction of literals. Without loss of generality, we may assume that for every pair (i, j) there is no clash between the literals a_i and $\neg b_j$; otherwise φ_1 can immediately be reduced to $\mathbf{f}$. Also without loss, we may assume that there is at least one $\mathsf{M}\beta_k$ occurring; otherwise conjoin the formula $\mathsf{M}\mathbf{t}$ (which is equivalent to $\mathbf{t}$) to φ_1.

The second step is the key to the procedure. We make use of the equivalence $B8$

$$\mathsf{B}\alpha \wedge \mathsf{M}\beta = \mathsf{B}\alpha \wedge \mathsf{M}(\alpha \wedge \beta)$$

and the equivalence B2 for negation-free φ to bring φ_1 to an equivalent

$$\varphi_2 =_{def} \bigwedge a_i \wedge \bigwedge \neg b_j \wedge \bigwedge \mathsf{B}\alpha_m \wedge \bigwedge \mathsf{M}\ [\beta_n \wedge \bigwedge a_i \wedge \bigwedge \alpha_m].$$

One can further simplify each $\beta_n \wedge \bigwedge a_i \wedge \bigwedge \alpha_m$ inside M by transforming to a propositional disjunctive normal form.

In the third step, we apply $B7$ to bring the disjunctions outside M. We now check if there is any disjunct ρ of the form

$$\bigwedge a_i \wedge \bigwedge \neg b_j \wedge \bigwedge \mathsf{B}\alpha_m \wedge \bigwedge \mathsf{M}\beta'_n,$$

such that each β'_n's are satisfiable. (This is a simple task because each β'_n is a conjunction of literals.) If there is such a disjunct ρ, then by the Main Satisfiability Lemma it is satisfiable. (Note that we now have to replace the empty situation by $\{a_i\}$ in the statement and proof of the Main Satisfiability Lemma) Therefore $\varphi_2 = \varphi_1$ is satisfiable, and so the original formula ψ is satisfiable. Otherwise there is no such ρ, and we can reduce φ_1 to $\mathbf{f}$ in one step. We then repeat the above for all the other disjuncts in the pseudo-DNF of ψ.

This completes the description of the decision procedure.

For completeness, suppose $\varphi = \psi$ is valid. Then the negation of $(\varphi \to \psi) \wedge (\psi \to \varphi)$ must be unsatisfiable. The decision procedure shows we have

$$\vdash \neg[(\varphi \to \psi) \wedge (\psi \to \varphi)] = \mathbf{f}.$$

Therefore $\varphi = \psi$ is a theorem of System **B**.

5.3 Compactness

A set of formula Σ is said to be satisfiable if there exists a model $(x, \underline{A})$ such that $x \models_{\underline{A}} \varphi$ for every φ in Σ. Compactness is the property which states that for any formula set Σ, it is satisfiable exactly when all finite subsets of it are satisfiable. Therefore, to prove compactness, we need to show that Σ has a model, if all finite subsets of it has one. Note that the fact that each finite subset of Σ has a model does not mean all the models are the same; we cannot even assume that they are based on the same default coherent space. So, the trick is to "unify" the possibly distinct models into a single one. This notion of "unification" can be made formal by the *sub-structure* relation $\prec$.

Definition 5.2 *Let*

$$\underline{A}_i = (A_i, \smile_i, \Delta_i)$$

$(i = 1, 2)$ be default coherent spaces. Define $\underline{A}_1 \prec \underline{A}_2$ if

1. $A_1 \subseteq A_2$,
2. $\Delta_1 \subseteq \Delta_2$,
3. $a \smile_1 b \Leftrightarrow a, b \in A_1$ & $a \smile_2 b$.

Observe that if $\underline{A}_0 \prec \underline{A}_2 \prec \cdots \prec \underline{A}_i \prec \cdots$ is a chain of default coherent spaces, then the least upper bound of this chain exists (with respect to the order $\prec$). Moreover, the least upper bound is just the componentwise union:

$$\bigsqcup \underline{A}_i = (\bigcup A_i, \bigcup \smile_i, \bigcup \Delta_i).$$

Lemma 5.4 *Let Σ be a set of pseudo-atomic formulas. If every finite subset of Σ is satisfiable, then Σ is satisfiable.*

We outline the main ideas of the proof and refer the reader to [7] for more details. First, rewrite Σ as the union of four disjoint components:

$$\Sigma = P \cup N \cup B \cup M,$$

where

$P = \{a_1, a_2, \ldots\}$, the set of atomic formulas contained in Σ;

$N = \{\neg b_1, \neg b_2, \ldots\}$, the set of negated atomic formulas contained in Σ;

$B = \{\mathsf{B}(\theta_1), \mathsf{B}(\theta_2), \ldots\}$, where each θ_i is a disjunction of literals; and

$M = \{\mathsf{M}(\lambda_1), \mathsf{M}(\lambda_2), \ldots\}$, where each λ_i is a conjunction of literals.

We construct a finite branching, infinite tree as follows. Each node of the tree is labeled by a default coherent space. If one node $\underline{A}$ is the parent of another node, labeled by $\underline{B}$, then we ensure that $\underline{A} \prec \underline{B}$. Let $x = P$. This x will be fixed for all the structures constructed below.

Let the root of the tree be the structure $(P, \smile, \emptyset)$, where $a \smile b$ for each a, b in P. (For later reference, we name each component as A_0, $\smile_0$, and Δ_0, respectively.) With respect to this structure, we clearly have $x \models p$ for every p in P, and $x \models \neg q$, for every $\neg q$ in N.

We now extend the tree to the next generation of nodes, so that each node is labeled by a default coherent space satisfying all formulas in the set $P \cup N \cup \{\mathsf{B}(\theta_1)\}$. The next generation of nodes are labeled by default coherent spaces satisfying all formulas in the set $P \cup N \cup \{\mathsf{B}(\theta_1), \mathsf{M}(\lambda_1)\}$ with the additional property that for each node $\underline{A}$ at the current level, there exists a node $\underline{B}$ at the previous level such that $\underline{B} \prec \underline{A}$.

The construction continues by alternatively adding a formula from B and a formula from M, with the tree structure preserved. Since this process can continue indefinitely, and since at each next generation there are a finite number of new nodes, we derive a finite branching, infinite tree. By K'onig's lemma, this tree has an infinite branch. The least upper bounds of all the default coherent spaces along this branch is again a default coherent space. Moreover, it is a model of all formulas in Σ. Therefore, Σ itself is satisfiable.

Once this lemma is established, we can use a similar method to prove that compactness property.

Theorem 5.3 (Compactness) *Let Σ be a set of formulas from System* **B**. *If every finite subset of Σ is satisfiable, then Σ is satisfiable.*

6 Conclusion

We have introduced a modal logic for belief, developed hand in hand with a well-motivated semantic structure. It is possible to model nonmonotonic reasoning in this framework; but the logic itself has the normal properties, such as soundess, completeness, and compactness.

The general framework we have outlined for building logics and models is well-justified on theoretical grounds (the methodology of domain theory) and by much of the work on modal non-monotonic logic, notably including the original work of McDermott and Doyle [14]. This original work, together with that of Reiter, perhaps went astray by relying too heavily on proof-theoretic ideas. This was very understandable at the time, given the lack of model-theoretic techniques appropriate for the task at hand. For us, the key to the successful integration of default reasoning and a modal logic for belief has been domain theory. We think that this theory, and the very much philosophically allied situation theory [3], may have a lot to offer to the knowledge representation community.

References

[1] S. Abramsky. Domain theory in logical form. *Annals of Pure and Applied Logic* 51, 1991.

[2] F. Baader and B. Hollunder. Embedding Defaults into Terminological Knowledge Representation Formalisms. In *Proceedings of Third Annual Conference on Knowledge Representation*, Morgan Kaufmann, 1992.

[3] Jon Barwise. *The Situation in Logic.* 17. Center for Study of Language and Information, Stanford, California, 1989.

[4] P. Buneman, A. Jung, and A. Ohori. Using powerdomains to generalize relational data bases. *Theoretical Computer Science*, 91:23-35, 1991.

[5] P. Gärdenfors and D. Makinson. Nonmonotonic inference based on expectations. *Artificial Intelligence* 65, 197-245, 1994.

[6] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge.* MIT Press, Cambridge, MA, 1995.

[7] C. Huang. *Results on a Modal Logic for Belief with Default Domain Models.* M.S. thesis, Department of Computer Science, University of Georgia, Athens, Georgia, 1996.

[8] P. Johnson-Laird. *Mental Models.* Harvard University Press, 1983.

[9] L. Libkin. *Aspects of Partial Information in Databases.* Ph.D. thesis, University of Pennsylvania, 1994.

[10] V. Lifschitz. On open defaults. In *Proceedings of the Symposium on Computational Logics*, Brussels, 1990.

[11] J. J. ch. Meyer and W. van der Hoek. A modal logic for nonmonotonic reasoning. In *Non-Monotonic Reasoning and Partial Semantics*, chapter 3. Ellis Horwood, 1992.

[12] D. Gabbay. Theoretical foundations for non-monotonic reasoning in expert systems. In K. R. Apt, editor, *Proceedings of NATO Advanced Study Institute on Logics and Models of Concurrent Systems*, pages 439–457. Springer Verlag, 1985.

[13] Jean-Yves Girard, Linear logic, *Theoretical Computer Science* **50** (1987) 1–102.

[14] D. McDermott and J. Doyle. Non-monotonic logic I. *Artificial Intelligence*, 13:41–72, 1980.

[15] Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[16] W. Rounds and G.-Q. Zhang. Attunement to constraints in nonmonotonic reasoning. In *Logic, Language and Computation vol 1*, J. Seligman and D. Westerstahl (eds), Lecture Notes No. 58, CSLI Publications, Stanford University, pp. 479-494, 1996.

[17] W. Rounds and G.-Q. Zhang. Logical considerations on default semantics. To appear in *Annals of Mathematics and Artificial Intelligence.*

[18] W. Rounds and G.-Q. Zhang. Domain theory meets default logic. *Journal of Logic and Computation, vol 5*, no. 1, pp. 1-25, 1995.

[19] W. Rounds and G.-Q. Zhang. Modal logics for default domains, 1992. Manuscript.

[20] Dana S. Scott. Domains for denotational semantics. In *Lecture Notes in Computer Science 140*, 1982.

[21] G.-Q. Zhang and W. Rounds. Nonmonotonic consequences in default domain theory. To appear in *Annals of Mathematics and Artificial Intelligence.*

[22] G.-Q. Zhang and W. Rounds. Defaults in domain theory. To appear in *Theoretical Computer Science.*

[23] Guo-Qiang Zhang. *Logic of Domains.* Birkhauser, Boston, 1991.