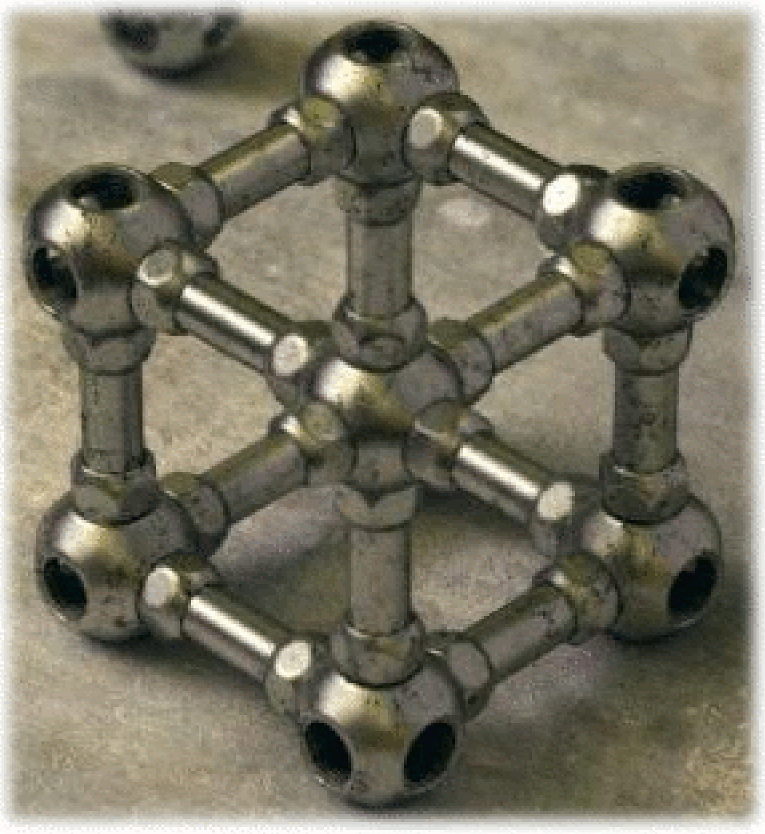

Revision of CTL Models [1]

Author(s):

Paulo T. Guerra
Renata Wassermann

[1] This work was supported by Fapesp Project LogProb, grant 2008/03995-5, São Paulo, Brazil.

Revision of CTL Models

Paulo T. Guerra and Renata Wassermann

Department of Computer Science
Institute of Mathematics and Statistics
University of São Paulo, Brazil
{paulotgo,renata}@ime.usp.br

Abstract. Model checking is one of the most robust techniques in automated system verification. But, although this technique can handle complex verifications, model checking tools usually do not give any information on how to repair inconsistent system models. In this paper, we show that approaches developed for CTL model update cannot deal with all kinds of model changes. We introduce the concept of CTL model revision: an approach based on belief revision to handle system inconsistency in a static context. We relate our proposal to classical works in belief revision and give an algorithm sketch.

1 Introduction

Model checking is an efficient method for automated verification of systems specification. Proposed by Clark and Emerson [1], this technique allows a system designer to check whether a system model satisfies (or not) its formal specification, i.e., the set of desired properties that defines the system behavior. In the last decades, many studies as [2–4] have improved model checking, resulting in fast algorithms and, consequently, in the development of efficient verification tools.

Most of these tools just return a counterexample path when they find an inconsistency. This information is useful for system designers, because they can direct the correction to a specific point of the system. However, the more complex a system is, the harder it becomes for the designer to fix it. In general, model checking tools do not have any mechanisms to assist the user in the correction task.

To address this issue, it is quite natural to think of integrating model checking and some automated change technique. The work [5] was the first in this line. Buccafurri et al. integrate model checking and abductive theory revision to perform system repairs. Although this approach can be successfully applied to their purposes, the authors themselves state that it may not be general enough to deal with other system modifications.In [6], Zhang and Ding propose a formal framework for model update through the integration of model checking and belief update [7–9]. They specify a minimal change principle for model update, and then define the concept of admissible update. They also specify a model update operator and study its semantics and computational properties.

A. Kuri-Morales and G.R. Simari (Eds.): IBERAMIA 2010, LNAI 6433, pp. 153–162, 2010.

While belief update deals with changing information due to changes in the world, belief revision [10] deals with incoming information about a static world. Despite the similarity of the problems, the use of an incorrect type of change can lead to information loss.

In our work, we propose the use of belief revision for changing CTL specifications. Our approach addresses the repair of systems specifications in a static context. As Zhang and Ding do to update, we define a model revision operator and characterize its relationship with some classical work on belief revision.

2 CTL Model Checking

Formal verification methods check whether a system model, specified in a formal language, satisfies a set of desired properties. Model checking [11, 12] is probably the most widely used verification method. According to [11], model checking is defined as the process of computing an answer to the question whether $(M, s) \models \phi$ holds, where ϕ is a temporal formula, M is an appropriate model of the system under consideration, s is a state of that model, and $\models$ is the underlying satisfaction relation.

In this paper, we will address only a special type of model checking, the *CTL model checking*. We will assume that systems are described by Kripke models and that the desired properties are CTL formulas. We introduce both concepts below.

2.1 Computation Tree Logic

Temporal logic is a kind of modal logic where we can represent and reason about time sentences. *Computation Tree Logic* [11, 12], CTL for short, models the future as a tree-like structure. Due to its branching characteristic, CTL can quantify over several execution paths that a system can follow. In Definitions 1 and 3, we briefly show the CTL syntax and semantics, respectively (see [12] for details).

Definition 1 (Syntax). *A CTL formula has the following syntax:*

$$\varphi ::= \top \mid p \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid EX\varphi \mid EG\varphi \mid E[\varphi U \varphi]$$

where p ranges over a set of atomic formulas, $\neg$, $\wedge$ are the classical logic connectives, and the remaining elements are temporal operators.

We use $\perp$ to denote $\neg\top$, and other temporal operators can be derived from EX, EG and EU: AX $\phi = \neg$ EX $\neg\phi$; AG $\phi = \neg$ EF $\neg\phi$; AF $\phi = \neg$ EG $\neg\phi$; EF $\phi =$ E [true U ϕ]; A[ϕ U β] $= \neg$ E[$\neg\beta$ U $\neg\phi \wedge \neg\beta$] $\wedge \neg$ EG $\neg\beta$.

Each temporal operator consists of a path quantifier (E, “there is a path”, or A, “for all paths”) followed by a state operator (X, “next state”, U, “until”, G, “globally in states” or F, “some future state”).

Definition 2. *Let P be a finite set of atomic propositions, a* Kripke model *is a triple $M = (S, R, L)$, where*

1. *S is a set of states*
2. *$R \in S \times S$ is a serial relation between states, that is, $\forall s \in S$, $\exists s' \in S$ such that $(s, s') \in R$.*
3. *$L : S \to 2^P$ is a labeling function such that $\forall s \in S$, $L(s)$ determines all atomic propositions true in s.*

Definition 3 (Semantic). *Let $M = (S, R, L)$ be a Kripke model, $s \in S$ a state of M and ϕ a CTL formula. We define $(M, s) \models \phi$ inductively as follows:*

1. *$(M, s) \models \top$.*
2. *$(M, s) \models p$ iff $p \in L(s)$.*
3. *$(M, s) \models \neg\phi$ iff $(M, s) \not\models \phi$.*
4. *$(M, s) \models \phi_1 \wedge \phi_2$ iff $(M, s) \models \phi_1$ and $(M, s) \models \phi_2$.*
5. *$(M, s) \models EX\phi$ iff $\exists s' \in S$ such that $(s, s') \in R$ and $(M, s') \models \phi$.*
6. *$(M, s) \models EG\phi$ iff there is a path $\pi = [s_0, s_1, \ldots]$ in M such that $s_0 = s$ and $(M, s_i) \models \phi$ for all $i \geq 0$.*
7. *$(M, s) \models E[\phi_1 U \phi_2]$ iff there is a path $\pi = [s_0, s_1, \ldots]$ in M such that $s_0 = s$, $\exists i \geq 0$, $(M, s_i) \models \phi_2$ and $\forall j < i$, $(M, s_j) \models \phi_1$.*

Given a CTL Kripke model $M = (S, R, L)$ and its set of initial states $Init(S) \subseteq S$, we say that $M \models \phi$, if and only if $(M, s) \models \phi$ for some $s \in Init(S)$.

3 Belief Revision

Belief revision deals with the problem of adapting an agent's set of beliefs in order to incorporate new information, possibly inconsistent with what the agent used to believe. In [10], Alchourrón, Gärdenfors and Makinson proposed a set of postulates that belief change should obey, as well as some constructions for revision that follow these postulates. This became known as the AGM paradigm.

In the AGM Paradigm, beliefs are represented as *belief sets*, that is, a set of formulas K such that $Cn(K) = K$, where Cn is a supraclassical consequence operator. Two other belief representation are important to our work: *belief bases* and *possible worlds*. When we use *belief bases* we assume that belief sets can be inferred from a small set of beliefs. A belief base B_K represents the belief set K such that $K = Cn(B_K)$. In a *possible worlds* representation, we do not represent the beliefs of the agent by sets formulas, but by all the models (worlds) that satisfy the beliefs. We denote by $[K]$ (some times $Mod(K)$) the set of models where every formula of a belief set K holds.

The AGM postulates guide revision operations through a minimal change principle, that is, nothing should be added or removed from a belief set if it is not strictly necessary to the success of revision. Given a belief set K, a belief α, $K + \alpha$ is the result of expanding K with α, usually defined as $K + \alpha = Cn(K \cup \{\alpha\})$. The operation of revision, denoted by $K * \alpha$ is not uniquely defined, but constrained by rationality postulates. Here, we will follow the notation adopted in [13], where the authors suggested a reformulation of AGM postulates in which a belief set is represented by a single sentence ψ and the result of revising ψ by ϕ is denoted by $\psi \circ \phi$.

(R1) $\psi \circ \phi \vDash \phi$.
(R2) If $\psi \wedge \phi$ is satisfiable, then $\psi \circ \phi \equiv \psi \wedge \phi$
(R3) If ϕ is satisfiable, then $\psi \circ \phi$ is satisfiable.
(R4) If $\psi_1 \equiv \psi_2$ and $\phi_1 \equiv \phi_2$, then $\psi_1 \circ \phi_1 \equiv \psi_2 \circ \phi_2$.
(R5) $(\psi \circ \phi) \wedge \mu \vDash \psi \circ (\phi \wedge \mu)$
(R6) If $(\psi \circ \phi) \wedge \mu$ is satisfiable, then $\psi \circ (\phi \wedge \mu) \vDash (\psi \circ \phi) \wedge \mu$

4 Belief Update

The purpose of *belief update*, as *belief revision*, is to preserve the consistency of belief sets when new information is added. While *belief revision* deals with static worlds, belief update acts on dynamic worlds, where the new information describes changes that occur in that world. This is the fundamental distinction between them: the nature of the change.

In belief revision, the new information means a refinement of beliefs, or even that the current belief set can not describe the world correctly. But in belief update, the new belief shows a change, it does not say anything about what the world was, just about what it looks like after the change.

Suppose we want to change the belief base ψ by a formula ϕ. Revision operators will select from the models of ϕ those that are in some sense closest to the models of ψ. On the other hand, update operators select, for each model M of ψ, the models of ϕ that are closest to M. Example 1 illustrates this distinction between revision and update.

Example 1. Suppose there is a room containing three objects: a book, a magazine and a table. The proposition b means "the book is on the table" and the proposition m means "the magazine is on the table". We believe that $\psi \leftrightarrow (b \wedge \neg m) \vee (\neg b \wedge m)$, that is, either the book or the magazine are on the table, but not both. There are only two models: I, b is true and m is not; or J, where m is true and b is not.

Suppose now that we send a robot to put the book on the table. After the success of the operation, we want to incorporate to our belief base the new belief $\phi \leftrightarrow b$. This belief also has two possible models: I and a new one K, where both, book and magazine, are on the table. We will select those models of ϕ which are closest to the models of ψ. We will assume as distance relation the number of proposition with different true value. The model I is at distance 0 of itself and at distance 2 of J (both propositions change). On the other hand, the model K is a distance 1 of both I and J (matching exactly one object location).

A revision operator will select just model I because it is the closest model to the original set. But intuitively this should not be the case. After the robot performs its task, all we know is that the book is on the table, but we know nothing about the magazine location.

Katsuno and Mendelzon [9] proposed a set of postulates to characterize rational update operations. Let ψ represent a finite belief base, ϕ the new belief and $\psi \diamond \phi$ the result of the update, the eight update postulates defined by Katsuno and Mendelzon are:

(U1) $\psi \diamond \phi \vDash \phi$.
(U2) If $\psi \vDash \phi$, then $\psi \diamond \phi \equiv \psi$.
(U3) If both ψ and ϕ are satisfiable, then $\psi \diamond \phi$ is also satisfiable.
(U4) If $\psi_1 \equiv \psi_2$ and $\phi_1 \equiv \phi_2$, then $\psi_1 \diamond \phi_1 \equiv \psi_2 \diamond \phi_2$.
(U5) $(\psi \diamond \phi) \wedge \mu \vDash \psi \diamond (\phi \wedge \mu)$
(U6) If $\psi \diamond \phi_1 \vDash \phi_2$ and $\psi \diamond \phi_2 \vDash \phi_1$, then $\psi \diamond \phi_1 \equiv \psi \diamond \phi_2$
(U7) If ψ is complete (i.e has a unique model) then $(\psi \diamond \phi_1) \wedge (\psi \diamond \phi_2) \vDash \psi \diamond (\phi_1 \vee \phi_2)$
(U8) If $(\psi_1 \vee \psi_2) \diamond \phi \equiv (\psi_1 \diamond \phi) \vee (\psi_2 \diamond \phi)$

Herzig [14] shows that, more than distinct, revision and update are mutually incompatible, since there is not an operator able to satisfies both revision and update postulates.

5 CTL Model Change

Let us now depart from propositional logic and see what happens when we want to change formal specifications, where models are Kripke models and the input is a CTL formula. First we will look at changing a single model $M = (S, R, L)$. Zhang and Ding [6] defined five basic operations where all possible changes on CTL models could be achieved.

PU1 : Adding one pair to the relation R
PU2 : Removing one pair from the relation R
PU3 : Changing the labeling function on one state
PU4 : Adding one state
PU5 : Removing one isolated state

Given two CTL Models $M = (S, R, L)$ and $M' = (S', R', L')$, for each PUi $(i = 1, \ldots, 5)$, we denote by $\mathit{Diff}_{PUi}(M, M')$ the difference between M and M', where M' is an updated model from M, where

1. $\mathit{Diff}_{PU1}(M, M') = R' - R$ (the set of pairs added to the relation).
2. $\mathit{Diff}_{PU2}(M, M') = R - R'$ (the set of pairs removed from the relation).
3. $\mathit{Diff}_{PU3}(M, M') = \{s \mid s \in S \cap S' \text{ and } L(s) \neq L'(s)\}$ (the set of states whose labeling function has changed).
4. $\mathit{Diff}_{PU4}(M, M') = S' - S$ (the set of added states).
5. $\mathit{Diff}_{PU5}(M, M') = S - S'$. (the set of removed states).

Based on these metrics, Zhang and Ding define the ordering $\leq_M$: a measure on the difference between two CTL models with respect to a given model.

Definition 4 (Closeness ordering). *Let $M = (S, R, L)$, $M_1 = (S_1, R_1, L_1)$ and $M_2 = (S_2, R_2, L_2)$ be three CTL Kripke models. We say that M_1 is at least as close to M as M_2, denoted as $M_1 \leq_M M_2$, if and only if for each set of PU1-PU5 operations that transform M in M_2, there exists a set of PU1-PU5 operations that transform M in M_1 such that the following conditions hold:*

1. *for each i ($i = 1, \ldots, 5$), $Diff_{PUi}(M, M_1) \subseteq Diff_{PUi}(M, M_2)$, and*
2. *if $Diff_{PU3}(M, M_1) = Diff_{PU3}(M, M_2)$, then for all $s \in Diff_{PU3}(M, M_1)$, $diff(L(s), L_1(s)) \subseteq diff(L(s), L_2(s))$, where $diff(A, B) = (A - B) \cup (B - A)$ for any two sets A and B.*

We denote $M_1 <_M M_2$ if $M_1 \leq_M M_2$ and $M_2 \not\leq_M M_1$.

Having the ordering relation specified in Definition 4, the authors formally specify in Definition 5 what an admissible model update is:

Definition 5 (Admissible update). *Given a CTL model $M = (S, R, L)$, $\mathcal{M} = (M, s_0)$ where $s_0 \in S$, and a CTL formula ϕ, a CTL Kripke model $Update(\mathcal{M}, \phi)$ is called an admissible model (or admissible update model) if the following conditions hold:*

1. *$Update(\mathcal{M}, \phi) = (M', s'_0)$, $(M', s'_0) \models \phi$, where $M' = (S', R', L')$ and $s'_0 \in S'$.*
2. *there does not exist another updated model $M'' = (S'', R'', L'')$ and $s''_0 \in S''$ such that $(M'', s''_0) \models \phi$ and $M'' <_M M'$.*

$Poss(Update(\mathcal{M}, \phi))$ denotes the set of all possible admissible models of updating $\mathcal{M}$ to satisfy ϕ.

The main contribution of Zhang and Ding's paper is perhaps the semantical analysis of CTL model update. They analyze their approach with respect to classical belief update approaches, identifying common issues between them. Although they address essentially different problems, the notion of minimal change in CTL model update is closely related to what occurs in classical approaches.

To make possible a comparison between CTL model update and traditional update postulates, Zhang and Ding define an operator $\diamond_c$ based on Winslett's Possible Model Approach [15], but designed for CTL model update. The formal definition of $\diamond_c$ is given in Definition 6.

Definition 6. *Let $M = (S, R, L)$ be a CTL Kripke model, $Init(S) \subseteq S$ the set of initial states of M and ϕ a CTL formula. M is called a model of ϕ if and only if $(M, s) \models \phi$ for some $s \in Init(S)$. Denote by $Mod(\phi)$ the set of all models of ϕ. The model update operator $\diamond_c$ is defined as:*

$$Mod(\psi \diamond_c \phi) = \bigcup_{(M,s) \in Mod(\psi)} Poss(Update((M, s), \phi))$$

Zhang and Ding ([6], Theorem 1) proved that operator $\diamond_c$ satisfies all Katsuno and Mendelzon update postulates. They show that KM postulates can characterize a wider scope of update operations than just its traditional application on propositional logic. In this sense, Zhang and Ding show that KM postulates might be essential for any model based update approach.

6 CTL Model Revision

The need for CTL model revision is motivated through the fundamental difference between belief revision and belief update: the type of problems that each one is suitable for. As seen in Section 4, belief update is used to modify a belief set in order to accommodate a new information about a dynamic world, while belief revision is used when the new information is about a world that has not changed.

To illustrate a situation where belief revision is more adequate, let us go back to Example 1. In this example we believe that, in a given room, either a book is on a table (b) or a magazine is (m), but not both, that is, $\psi \leftrightarrow (b \wedge \neg m) \vee (\neg b \wedge m)$ where ψ represents our knowledge base. Then, we ask a robot to put the book on the table. In other words, we want to incorporate to our beliefs a new information, $\mu \leftrightarrow b$. An update operator generates a new knowledge base $\psi' \leftrightarrow b$, while a revision operator generates the knowledge base $\psi'' \leftrightarrow (b \wedge \neg m)$. Intuitively, the update result has a more rational behavior, because the whole situation does not give us any information about the magazine location.

But suppose that instead of asking the robot to put the book on table, we ask it to just enter in the room and tell us where the book is. Suppose now that when the robot comes back it informs us that the book is on the table. Note that the new belief remains the same, $\mu \leftrightarrow b$, and the results of update and revision operations will be identical to the previous case. But, in this new situation, the result of revision seems more appropriated.

According to the new information $\mu \leftrightarrow b$, two room configurations are possible: (1) the book is on the table and the magazine is on the floor ($b \wedge \neg m$) or (2) both, book and magazine, are on the table ($b \wedge m$). In the last case, we used to believe that book and magazine are not in the same location and we know that nothing in the room has changed, so there is no reason to us to have doubts about where the magazine is. We can see that revision produces a more informative result.

Note that, in both cases, the knowledge base ψ and the new information μ are identical, what changes is how a individual interacts with the world, and, therefore, the way that the agent should realize it. We believe that the same distinction problem between static and dynamic worlds should be applied to system specifications.

There are two main reasons to modify a system specification: (1) when it fails to satisfy a given requisite and (2) when it needs to be adapted to a new requisite. When we modify a specification motivated by (1), we believe that a repair approach based on belief revision could generate more informative results than that produced by the CTL Model Update approach, in a similar way to what occurs between belief revision and belief update.

An approach based on belief revision for handling inconsistencies was presented in [16, 17]. The authors enrich the NuSMV model checker [4] with principles of belief revision, which results in *BrNuSMV*[1], a tool capable of suggesting modifications to a given system specification such that some desired CTL property becomes true.

Perhaps because BrNuSMV is a initial step into the integration of model checking and belief revision principles, there are some gaps: it is not possible to deal with all CTL formulas; the revision process is done state by state, with restricted set of beliefs; there is no formal analysis of the belief revision process. In our work, we intend to fill some of these gaps, doing a formal analysis of a revision method applied to generic CTL models and studying its relation with classical belief revision approaches.

[1] Acronym for Belief Revision NuSMV.

We specify in Definition 7 our revision operator $\circ_c$ over CTL formula.

Definition 7. *Given two CTL formulas ψ and ϕ, we define that $\psi \circ_c \phi$ results in a CTL formula whose models are defined as*

$$Mod(\psi \circ_c \phi) = Min_{Mod(\psi)}(Mod(\phi))$$

where $Min_{\mathcal{B}}(\mathcal{A})$ denotes the set of all minimal models of $\mathcal{A}$ with respect to all orderings $\leq_M$ such that M is a model of $\mathcal{B}$.

Theorem 1. *Operator $\circ_c$ satisfies revision postulates (R1)-(R6).*[2]

In Algorithm 1, we present an algorithm sketch for CTL model revision. Given a belief base ψ and a new belief ϕ, both represented by CTL formulas, the revision algorithm computes a set $\mathcal{S}$ of CTL Kripke models such that for all $M \in \mathcal{S}, M \in Mod(\psi \circ_c \phi)$.

Algorithm 1. CTLModelRevision(ψ, ϕ)

```
Input: Two CTL formulas $\psi$ and $\phi$, representing the belief base and the new belief,
    respectively.
Output: A set of CTL Kripke models $\mathcal{S}$ such that $\mathcal{S} \subseteq Mod(\psi \circ_c \phi)$.
 1: $\mathcal{S} \leftarrow \emptyset$;
 2: for all models $M = (S, R, L)$ such that $M \in Mod(\psi)$ do
 3:    if $M \models \phi$ holds then
 4:       $\mathcal{S} \leftarrow \mathcal{S} \cup \{M\}$;
 5:    else
 6:       repeat
 7:          $\mathcal{S} \leftarrow \mathcal{S} \cup CTLUpdate((M, s), \phi)$, where $s \in Init(S)$;
 8:       until there is no change in $\mathcal{S}$.
 9: for all models $M = (S, R, L)$ such that $M \in Mod(\psi)$ do
10:    $\mathcal{S} \leftarrow \mathcal{S} - \{M' | \{M', M''\} \subseteq \mathcal{S}$ and $M'' <_M M'\}$;
11: return $\mathcal{S}$.
```

First, we initialize $\mathcal{S}$ as empty. In lines 2-8, we iterates over ψ's models to find models that satisfies ϕ. If a model M of ψ satisfies ϕ, then M is added to $\mathcal{S}$, otherwise, in lines 6-8, we look for a set of possible models that satisfy ϕ and are close to M. For this purpose, we make successive calls to $CTLUpdate((M, s), \phi)$, a function described in [6]. In lines 9-10, we remove from $\mathcal{S}$ all models that are not minimal in relation to ψ's models. After this step, all models in $\mathcal{S}$ satisfy ϕ and are minimal with respect to $\leq_M$. Finally, we return $\mathcal{S}$.

Zhang and Ding [6], Theorem 8, proved that $CTLUpdate((M, s), \phi)$ always generates a model M' that satisfies ϕ and is a closest model of M according to Definition 4. However, there are no guarantees that $CTLUpdate((M, s), \phi)$ generates all possible minimal models, so we can not ensure that our algorithm can obtain $Mod(\psi \circ_c \phi)$ as result.

Zhang and Ding [6] do not define a CTL model update algorithm. In order to compare the revision and update model approaches, we specify in Algorithm 2 a

[2] The proof is omitted due to space limitation.

model update method based in their $\diamond_c$ update operator. The difference between these two approaches is illustrated in Example 2.

Algorithm 2. CTLModelUpdate(ψ, ϕ)

```
Input: Two CTL formulas $\psi$ and $\phi$, representing the belief base and the new belief,
  respectively.
Output: A set of CTL Kripke models $\mathcal{S}$ such that $\mathcal{S} \subseteq Mod(\psi \diamond_c \phi)$.
  $\mathcal{S} \leftarrow \emptyset$;
  for all models $M = (S, R, L)$ such that $M \in Mod(\psi)$ do
    if $M \models \phi$ holds then
      $\mathcal{S} \leftarrow \mathcal{S} \cup \{M\}$;
    else
      repeat
        $\mathcal{S} \leftarrow \mathcal{S} \cup CTLUpdate((M, s), \phi)$, where $s \in Init(S)$;
      until there is no change in $\mathcal{S}$.
  return $\mathcal{S}$.
```

Example 2. Let p and q be two propositional atoms and $\psi \leftrightarrow EXAGp$ our belief base. Suppose we need to modify ψ in order to accept $\phi \leftrightarrow E[pUq]$. Suppose also that ϕ does not describe a change in the world, just a new information about it. We know that $M_1 = (\{s_0, s_1\}, \{(s_0, s_1), (s_1, s_1)\}, \{L(s_0) = \{p\}, L(s_1) = \{p\}\}$ and $M_2 = (\{s_0, s_1\}, \{(s_0, s_1), (s_1, s_1)\}, \{L(s_0) = \{q\}, L(s_1) = \{p\}\}$ belong to $Mod(\psi)$. When we submit ψ and ϕ to CTL model update, we obtain a model $M_1' = (\{s_0, s_1, s_2\}, \{(s_0, s_1), (s_1, s_1), (s_1, s_2)\}, \{L(s_0) = \{p\}, L(s_1) = \{p\}, L(s_2) = \{q\}\}$, which does not belong to the output of the CTL model revision algorithm. Although M_1' is minimal in relation to M_1, M_1' it is not minimal in relation to all ψ's models (M_2 itself satisfies ϕ and it is $M_2 <_{M_2} M_1'$), so M_1' is not in $Mod(\psi \circ_c \phi)$, and it is removed in last loop of the revision algorithm.

7 Conclusion

We have shown that belief revision can be used to handle inconsistencies in formal system specifications when they are specified as CTL Kripke models. Based on the well-known difference between belief revision and belief update, we argue that our approach can be more adequate than Zhang and Ding's CTL model update when applied to system modifications in a static context. We have defined a revision operator for CTL and shown that our operator obeys the (R1)-(R6) revision postulates which shows that our operator performs revision in a rational way. Based on this operator definition, we developed an algorithm sketch for our CTL model revision approach.

The complete study of semantic properties demands the analysis of whether all constructions that obey the postulates (R1)-(R6) can be represented as our definition of CTL model revision. This will ensure the adequate rationality of our revision operation. Another issue is the analysis of the computational properties of CTL model revision, which is essential before we can think of practical applications.

Acknowledgments. The first author was supported by the Brazilian National Research Council (CNPq). The second author was partially supported by CNPq. This work is part of FAPESP project LogProb, grant number 2008/03995-5.

References

1. Clarke, E.M., Emerson, E.A., Sistla, A.P.: Automatic verification of finite-state concurrent systems using temporal logic specifications. ACM Transactions on Programming Languages and Systems (TOPLAS) 8(2), 244–263 (1986)
2. Amla, N., Du, X., Kuehlmann, A., Kurshan, R.P., McMillan, K.L.: An analysis of SAT-based model checking techniques in an industrial environment. In: Borrione, D., Paul, W. (eds.) CHARME 2005. LNCS, vol. 3725, pp. 254–268. Springer, Heidelberg (2005)
3. Chauhan, P., Clarke, E.M., Kukula, J.H., Sapra, S., Veith, H., Wang, D.: Automated abstraction refinement for model checking large state spaces using sat based conflict analysis. In: Aagaard, M.D., O'Leary, J.W. (eds.) FMCAD 2002. LNCS, vol. 2517, pp. 33–51. Springer, Heidelberg (2002)
4. Cimatti, A., Clarke, E.M., Giunchiglia, F., Roveri, M.: NuSMV: A new symbolic model verifier. In: Halbwachs, N., Peled, D.A. (eds.) CAV 1999. LNCS, vol. 1633, pp. 495–499. Springer, Heidelberg (1999)
5. Buccafurri, F., Eiter, T., Gottlob, G., Leone, N.: Enhancing model checking in verification by AI techniques. Artificial Intelligence 112(1-2), 57–104 (1999)
6. Zhang, Y., Ding, Y.: CTL model update for system modifications. Journal of Artificial Intelligence Research 31(1), 113–155 (2008)
7. Herzig, A., Rifi, O.: Propositional belief base update and minimal change. Artificial Intelligence 115(1), 107–138 (1999)
8. Winslett, M.: Reasoning about action using a possible models approach. In: Proc. of AAAI, pp. 89–93. Morgan Kaufmann, San Francisco (1988)
9. Katsuno, H., Mendelzon, A.O.: On the difference between updating a knowledge base and revising it. In: Proc. of KR, pp. 387–395. Morgan Kaufmann, San Francisco (1991)
10. Alchourron, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. J. Symb. Logic 50(2), 510–530 (1985)
11. Clarke, E.M., Grumberg, O., Peled, D.A.: Model checking. Springer, Heidelberg (1999)
12. Huth, M., Ryan, M.: Logic in Computer Science: Modelling and reasoning about systems. Cambridge University Press, Cambridge (2004)
13. Katsuno, H., Mendelzon, A.O.: A unified view of propositional knowledge base updates. In: Proc. of IJCAI 1989, pp. 1413–1419. Morgan Kaufmann, San Francisco (1989)
14. Herzig, A.: Logics for belief base updating. In: Dubois, D., Prade, H. (eds.) Handbook of Defeasible Reasoning and Uncertainty Management. Vol. Belief Change, pp. 189–231. Kluwer Academic, Dordrecht (1998)
15. Winslett, M.: Updating logical databases. Cambridge University Press, Cambridge (1990)
16. Sousa, T.C.: Revisão de modelos formais de sistemas de estados finitos. Master's thesis, Universidade de São Paulo (2007)
17. Sousa, T.C., Wassermann, R.: Handling inconsistencies in CTL model-checking using belief revision. In: Proc. of the Brazilian Symposium on Formal Methods (2007)