



Revising System Specifications in Temporal Logic

Paulo T. Guerra¹ · Renata Wassermann²

Accepted: 5 July 2022 / Published online: 6 August 2022
© The Author(s), under exclusive licence to Springer Nature B.V. 2022

Abstract

Although formal system verification has been around for many years, little attention was given to the case where the specification of the system has to be changed. This may occur due to a failure in capturing the clients' requirements or due to some change in the domain (think for example of banking systems that have to adapt to different taxes being imposed). We are interested in having methods not only to verify properties, but also to suggest how the system model should be changed so that a property would be satisfied. For this purpose, we will use techniques from the area of Belief Revision, that deals with the problem of changing a knowledge base in view of new information. In the last thirty years, several authors have contributed with change operations and ways of characterizing them. However, most of the work concentrates on knowledge bases represented using classical propositional logic. In the last decade, there have been efforts to apply belief revision theory to description and modal logics. In this work, we analyze what is needed for a theory of belief revision which can be applied to the temporal logic, such as the Computation Tree Logic (CTL). In particular, we illustrate different alternatives for formalizing the concept of revision of CTL. Our interest in this particular logic comes both from practical issues, since it is used for software specification, as from theoretical issues, as it is a non-compact logic and most existing results rely on compactness. We focus here on the revision of CTL models and present a characterization result for the revision of partial models.

Keywords Belief revision · Temporal logic · Model repair

1 Introduction

System verification is a phase of the development where a system is tested against a given set of properties. These properties can describe elementary facts such as “a division by zero will never occur” and compose what we call *system specification*.

✉ Paulo T. Guerra
paulodetarso@ufc.br

¹ Federal University of Ceará, Campus Quixadá, Quixadá, Brazil

² Institute of Mathematics and Statistics, University of São Paulo, São Paulo, Brazil

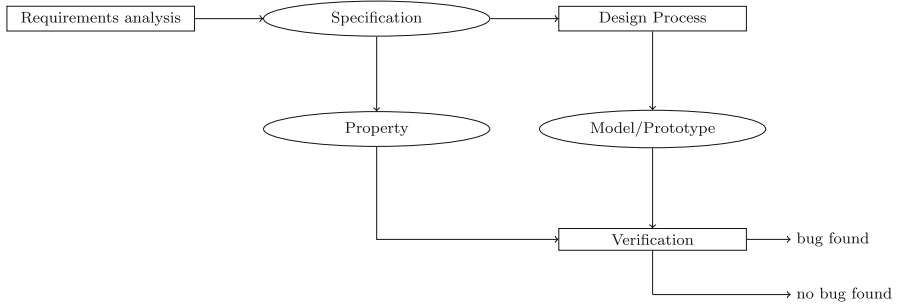


Fig. 1 Model verification mechanism. (Adapted from Baier and Katoen 2008)

A system specification establishes what a system should and should not do, being the basis of any verification activity (Baier & Katoen, 2008). We say there is an error if the system does not satisfy one or more of the properties described by the specification. The system will be *correct* if it satisfies all desired properties, otherwise it is said to be *incorrect* or *inconsistent* with respect to its specification. It is important to note that the correctness of a system is not an absolute property, always being relative to the evaluated specification.

Several formal verification methods use systems models instead of using the system concretely. This allows errors to be discovered in a preliminary development phase, before starting the development of concrete components of a system. The idea is to describe a system by mathematically precise models, capable of expressing in an unambiguous way how their behavior will be implemented. These models are then subjected to formal verification methods that will verify whether they satisfy all the properties described in the system specification (see Fig. 1). This type of approach is called *model-based verification* or simply *model checking*.

Model checking has origins in the works of Clarke and Emerson (1982); Clarke et al. (1986) and Queille and Sifakis (1982). The method consists in verifying whether a system model satisfies a given property by performing an exhaustive analysis that systematically explores all possible configurations that a system can assume according to the model, verifying that in each of these configurations the properties are indeed satisfied.

An important distinction must be made. In this work, a system specification is a description of *what* the system should do (and what it should not do). A system model is a description of *how* the system does its actions. The model check then examines whether the behavior explicitly stated by the model is according to the expected properties described by the specification. The effectiveness of model checking in predicting errors in real systems is thus as good as the model describing the system and as complete as the specification describing the intended properties.

The properties checked by the model checking have typically a qualitative nature (Baier & Katoen, 2008), such as “Is the final result is correct?”, “Will a *deadlock* ever occur?” or “Will the system complete its activities at some point?”. In formal verification, these properties need to be expressed accurately, avoiding any possible ambiguity. Since most of the properties relevant to verification of systems deal with how they behave during their execution, model verification methods are usually based

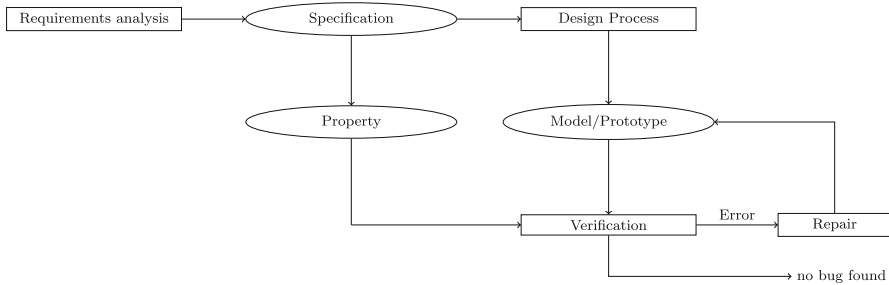


Fig. 2 Model verification mechanism with repair phase

on *temporal logics* (Clarke & Emerson, 1982; Kozen, 1983; Pnueli, 1977). These methods are capable of evaluating properties of a system specification described by formulas in these temporal formalisms.

In the last decades there has been a great improvement in the methods of model checking, with the optimization of algorithms and the design of new data structures. This allowed model verification tools in the 90's to be able to handle search spaces greater than 10^{20} states (Burch et al., 1992), against the 10^8 previously possible states.

Verification tools, when they detect an error, also usually provide information on how this error occurs. When they detect an inconsistency between model and specification, the model verification algorithms return a system execution flow that causes a violation of a property, which we call a *counterexample*. A designer can then use this information to correct the proposed model, improving the behavior of the system.

Figure 2 illustrates a more realistic development process, where the discovery of an error in the verification phase leads to changes in the original design of the system. We call this task *model repair*.

Fixing errors however is not always a simple task. The more complex the system, the more difficult it can be for a designer to correct inconsistencies. Although there are tools capable of verifying complex models, in general, these tools do not provide mechanisms to aid the task of model repair.

Belief revision (Alchourron et al., 1985) is a subarea of formal epistemology that deals with how we behave when receiving new information, especially if it is inconsistent with what we currently believe. Belief revision has applications in several areas, especially in Artificial Intelligence. It can be used, for example, to program an intelligent agent that re-plans its actions in the face of unexpected information such as an explorer robot that has a map of the environment and that plans a route based on this map. If it finds an obstacle that prohibits it from proceeding in the planned route, it can change its beliefs about the environment and trace a new route to reach its goal.

Our main goal in this paper is to explore the application of belief revision to formal verification in order to develop techniques that can assist in the task of fixing detected errors. In particular, if we consider the model of a system as what we believe to be true, when an error is detected (i.e., a property described in the specification that is not satisfied), we can use belief revision to rationally adapt our model so that such error no longer occurs.

We can also extend the problem in order to fix specification errors, that is, errors discovered in the execution of the system and that were not foreseen in its specification.

Assuming the specification of the system as our initial beliefs, belief revision could provide mechanisms of change that imply rational changes in this specification, and consequently, in the developed system.

An important theoretical issue, however, is that the temporal logics on which most of verification methods are based do not satisfy properties expected by the classical belief revision theory. They do not satisfy, for example, the property of *compactness*, which together with *monotonicity*, ensures the correct construction of belief revision operations (Hansson & Wassermann, 2002).

1.1 Related Works

Buccafurri et al. (1999) have developed a formal framework integrating model verification and *abductive reasoning* in order to diagnose and repair errors in concurrent programs. Abductive reasoning is used to find modifications in the system so that it satisfies all the properties of a formal specification. In this approach, a system is modeled according to a Cresswell and Hughes (2012) structure and the modifications correspond to a sequence of additions or removals of state transitions that makes the model consistent with the specification.

Although their approach is successful for its purposes of repair of concurrent programs and protocols, the concept of modification adopted is somewhat narrow, limited to modifications on the transition relation of a Kripke model. Zhang and Ding (2008) proposed a framework for model updating that addresses some issues that were not addressed by Buccafurri et al., such as the addition of new states or single modifications on state labels.

Zhang and Ding's approach is based on the integration of model checking and *belief update* (Herzig & Rifi, 1999; Katsuno & Mendelzon, 1991). The authors specified a minimum change principle for model changes and then defined a concept called admissible update. The authors also describe a procedure to perform model update and analyze its semantic and computational properties.

In Guerra and Wassermann (2010), we argue that an approach based on belief update is not suitable for all cases. We propose the use of *belief revision* (Alchourron et al., 1985) as principle to guide model changes when the repair of models occurs in a static context. Despite the similarity between belief update and belief revision, the use of the incorrect approach can lead to significant loss of information.

Sousa and Wassermann (2007) addressed the practical use of belief revision for the repair of incorrect models. The authors created a tool capable of generating repair suggestions for models described in SMV specification language. However, in this work the authors do not go into the theoretical analysis of the relationship of their technique with the classical belief revision theory.

In Guerra and Wassermann (2010) we describe the concept of revision of CTL models: an approach based on belief revision for the repair of incorrect models in a static context. We explore semantic properties that relate our proposal to classic works in belief revision, as well as discussing issues related to its implementation.

In the present work we provide a more complete overview of the use of belief revision in the formalism of temporal logics. The goal is to provide theoretical foundations

to perform rational repair of inconsistencies in formal specifications, developing an approach to revise sets of temporal formulas, as well as an approach to the repair of models and partial models, applying principles of belief revision to guide structural changes in systems models.

Some recent work deals with themes related to our work. Van Zee et al. (2015) propose a time-limiting temporal logic and present a belief revision characterization in this logic. Previously, Finger and Wassermann (2008) also addressed the revision of temporal beliefs using this type of time constraint, but exploring aspects related to bounded model checking. Our goal here is to explore the revision of temporal beliefs but reasoning over infinite computations, investigating the problem with all the expressivity potential of classical temporal formalisms.

As already mentioned, Guerra and Wassermann (2010) and Zhang and Ding (2008) address the problem of repairing inconsistencies in models. These works however write a partial characterization of the rationality of their repair operations. A complete characterization of the revision of a CTL model was presented in Guerra and Wassermann (2018).

On another line, Chatzieftheriou et al. (2012), Guerra et al. (2013) and Ribeiro and Andrade (2015) address the problem of repairing partially specified models. In Chatzieftheriou et al. (2012), the authors propose a framework for the refinement of partial models that represent abstractions of concrete models with a large number of states. In Guerra et al. (2013) and Ribeiro and Andrade (2015), the authors address the problem of modifications in partial models from the point of view of the refinements that this repair generates in the set of concrete models derived from this partial model. These works also do not present a complete characterization of the rationality of their operators, having a focus on the implementation of their techniques. In this paper, we present a characterization of the problem of repair of partial models in terms of rationality postulates, especially relating it to the problem of model revision discussed in Guerra and Wassermann (2010) and Guerra and Wassermann (2018).

2 Preliminaries

In this section, we briefly introduce the concepts we use from the areas of Belief Revision and the temporal logic CTL.

2.1 Belief Revision

Belief revision deals with how to adapt a set of beliefs in order to incorporate new information, even if inconsistent with what was previously believed. Alchourron et al. (1985) proposed a set of rationality postulates in order to specify what is expected from a rational revision function, which became known as the AGM postulates.

These rationality postulates guide the revision operations through a minimal change principle, in the sense that information is valuable and should be kept whenever possible. In the AGM theory, the beliefs of an agent are represented as a belief set, a set of formulas closed under logical consequence ($K = Cn(K)$). We present below the six basic AGM postulates for revision:

- (K*1) $K * \alpha$ is a belief set.
- (K*2) $\alpha \in K * \alpha$.
- (K*3) $K * \alpha \subseteq K + \alpha$.
- (K*4) If $\neg\alpha \notin K$, then $K + \alpha \subseteq K * \alpha$.
- (K*5) $K * \alpha$ is unsatisfiable if and only if $\models \neg\alpha$.
- (K*6) If $\models \alpha \leftrightarrow \beta$, then $K * \alpha = K * \beta$.

Postulate (K*1) says that the revision of a belief set must be another belief set. (K*2) says that the revised belief set must contain the formula by which it is revised. Postulate (K*3) says that no other information besides the formula should be added. Postulate (K*4) says that if the new formula is consistent with the current beliefs no belief should be discarded. Postulate (K*5) says that the revised belief set must be consistent, unless the formula itself is inconsistent. Postulate (K*6) assures that equivalent formulas should result in the same revised belief set.

Several constructions for revision functions were proposed observing the AGM postulates. A well-known construction is called *partial meet revision* and is based on the notion of *remainders*. A *remainder set* $K \perp \alpha$ contains the maximal subsets of K that do not imply α :

Definition 1 (Alchourron et al., 1985) Let K be a belief set and α a formula. The *remainder set* $K \perp \alpha$ is a collection of sets X such that

1. $X \subseteq K$
2. $X \not\vdash \alpha$
3. For all X' such that $X \subset X' \subseteq K$, $X' \vdash \alpha$.

The idea of a partial meet construction is that there is a mechanism that selects elements of the remainder set at hand:

Definition 2 (Alchourron & Makinson, 1982) Let K be a belief set and α a new belief, γ is a *selection function* for K and α if and only if:

1. $\emptyset \subset \gamma(K \perp \alpha) \subseteq K \perp \alpha$ if $K \perp \alpha \neq \emptyset$
2. $\gamma(K \perp \alpha) = \{K\}$ otherwise.

Definition 3 (Alchourron et al., 1985) Let K be a belief set, α a new belief, and γ a *selection function* for K and α . A *partial meet revision* function over K is given by

$$K * \alpha = Cn \left(\bigcap \gamma(K \perp \alpha) \cup \{\alpha\} \right)$$

Alchourrón, Gärdenfors and Makinson have proven the following result:

Theorem 1 (Alchourron et al., 1985; Hansson, 1999) *Let $*$ be a function which, given a formula α , takes a belief set K into a new belief set $K * \alpha$. For every theory K , $*$ is a partial meet revision operation over K if and only if $*$ satisfies the basic postulates (K*1)–(K*6) for revision.*¹

This result relies on certain properties of the underlying logic, such as compactness and the deduction theorem, among others. As we will see in the next subsection, an interesting logic to which we would like to apply belief revision, CTL, is not compact.

¹ Actually, Alchourrón, Gärdenfors and Makinson proved the result for another operation, *contraction*, and then later proved the relation between contraction and revision, having this theorem as a corollary.

2.2 Computation Tree Logic

Computation Tree Logic (CTL) (Clarke et al., 1986) is a temporal logic where the future is represented by a tree-like structure. Due to its branching characteristic, CTL is used to formally represent system properties, for example that every possible execution path eventually ends. CTL syntax is given by the following BNF:

$$\varphi ::= \top \mid \perp \mid p \mid (\neg\varphi) \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \rightarrow \varphi) \mid \text{EX}\varphi \mid \text{AX}\varphi \mid \text{EF}\varphi \mid \text{AF}\varphi \mid \text{EG}\varphi \mid \text{AG}\varphi \mid \text{E}[\varphi U \varphi] \mid \text{A}[\varphi U \varphi]$$

where its temporal modalities are composed by path quantifiers (E, “there is a path”, or A, “for all paths”) and state operators (X, “next state”, U, “until”, G, “globally in all states” or F, “some future state”). The diagrams in Fig. 3 represent branching sequences of states whose starting states satisfy the indicated CTL formula. As an example, consider the formula EX φ . The E stands for “there is a path”, the X stands for “in the NeXt state”. Hence, the formula is valid in the initial state if and only if there is a path starting at this state where in the next state, φ holds. If we look at formula AF φ , we have that in All paths, there is a Future state in which φ holds. The semantics of U is more complex, involving two formulas: E[$\varphi U \psi$] holds if and only if there is a path in which φ holds in all states Until the first state where ψ holds.

The CTL semantic is given through labeled transition system (LTS), described in Definition 4.

Definition 4 A labeled transition system is a tuple $\mathcal{M} = \langle AP, S, s_0, R, L \rangle$ such that:

1. AP is a countable set of propositional atoms;
2. S is a finite set of states;
3. $s_0 \in S$ is the initial state;
4. $R \subseteq S \times S$ is a transition relations over S;
5. $L : AP \rightarrow \mathcal{P}(S)$ is a labeling function of truth assignment.

Usually CTL semantics is given over Kripke structures, a special kind of LTS where the transition relation is required to be total. A graphical representation of a Kripke structure is depicted in Fig. 4.

The CTL semantics is then defined inductively as follow.

Definition 5 Let $\mathcal{M} = \langle AP, S, s_0, R, L \rangle$ be a LTS, $s \in S$ a state of \mathcal{M} and φ a CTL formula. We define $\mathcal{M}, s \models \varphi$ inductively as follows:

1. $\mathcal{M}, s \models \top$.
2. $\mathcal{M}, s \models p$ iff $s \in L(p)$.
3. $\mathcal{M}, s \models \neg\varphi$ iff $\mathcal{M}, s \not\models \varphi$.
4. $\mathcal{M}, s \models \varphi_1 \wedge \varphi_2$ iff $\mathcal{M}, s \models \varphi_1$ and $\mathcal{M}, s \models \varphi_2$.
5. $\mathcal{M}, s \models \text{EX}\varphi$ iff there is $s' \in S$ such that $(s, s') \in R$ and $\mathcal{M}, s' \models \varphi$.
6. $\mathcal{M}, s \models \text{AX}\varphi$ iff for all $s' \in S$ such that $(s, s') \in R$, $\mathcal{M}, s' \models \varphi$.
7. $\mathcal{M}, s \models \text{EF}\varphi$ iff there is a path² $\pi = [s_1, s_2, \dots]$ in \mathcal{M} such that $s_1 = s$ e $\mathcal{M}, s_i \models \varphi$ for some $i \geq 1$.

² A path $\pi = [s_1, s_2, \dots, s_{i-1}, s_i, s_{i+1}, \dots]$ is a path in $\mathcal{M} = \langle AP, S, s_0, R, L \rangle$ if and only if $\forall s \in \pi, s \in S$ and $(s_i, s_{i+1}) \in R$ for all $i \geq 1$.

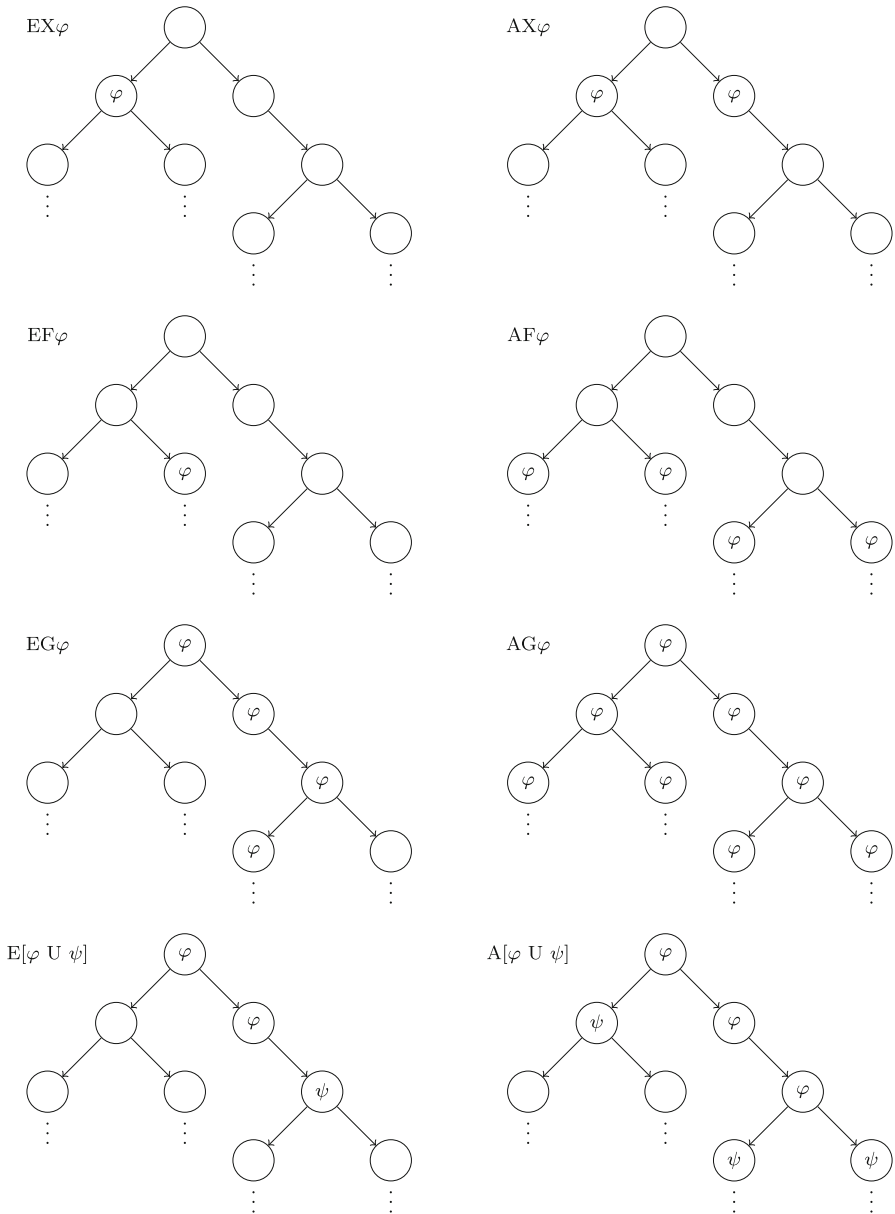
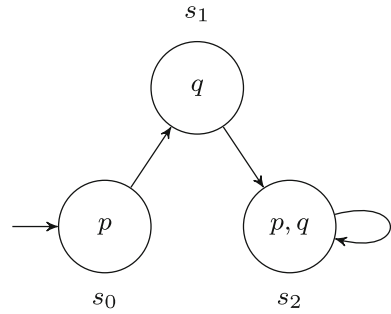


Fig. 3 Example of CTL formulas

8. $\mathcal{M}, s \models \text{AF}\varphi$ iff for all paths $\pi = [s_1, s_2, \dots]$ in \mathcal{M} such that $s_1 = s$, $\mathcal{M}, s_i \models \varphi$ for some $i \geq 1$.
9. $\mathcal{M}, s \models \text{EG}\varphi$ iff there is a path $\pi = [s_1, s_2, \dots]$ in \mathcal{M} such that $s_1 = s$ and $\mathcal{M}, s_i \models \varphi$ for all $i \geq 1$.
10. $\mathcal{M}, s \models \text{AG}\varphi$ iff for all paths $\pi = [s_1, s_2, \dots]$ in \mathcal{M} such that $s_1 = s$, $\mathcal{M}, s_i \models \varphi$ for all $i \geq 1$.

Fig. 4 Example of Kripke structure



- 11. $\mathcal{M}, s \models E[\varphi_1 \cup \varphi_2]$ iff there is a path $\pi = [s_1, s_2, \dots]$ in \mathcal{M} such that $s_1 = s$, $\exists i \geq 1, \mathcal{M}, s_i \models \varphi_2$ and $\forall j < i, \mathcal{M}, s_j \models \varphi_1$.
- 12. $\mathcal{M}, s \models A[\varphi_1 \cup \varphi_2]$ iff for all paths $\pi = [s_1, s_2, \dots]$ in \mathcal{M} such that $s_1 = s$, $\exists i \geq 1, \mathcal{M}, s_i \models \varphi_2$ and $\forall j < i, \mathcal{M}, s_j \models \varphi_1$.

We say that $\mathcal{M} \models \varphi$ if $\mathcal{M}, s_0 \models \varphi$.

3 Two Approaches for CTL Belief Revision

Belief revision can be used with Computation Tree Logic to compose a framework capable of managing the consistency of system’s behaviors. By assuming a representation of a system as the initial beliefs, when an inconsistency is detected between our current belief and some desired CTL property, we can use belief revision principles to minimally adapt our beliefs in order to accommodate this new belief.

There are two main approaches to CTL belief revision, depending on how the set of beliefs is represented. In the first approach called *specification revision*, the system behavior is described by a set of CTL formulas, each one describing a desired property about the intended evolution of the system through the time. In the second approach called *model revision*, the system behavior is described by one or more possible models (labeled transition system), where the states and possible transitions among them are explicitly described.

3.1 Revision of Specifications

On the *specification revision* approach, the system behavior is described by a set of CTL formulas representing system properties. These properties may describe things like liveness, safety, absence of deadlocks, etc. As in classical belief revision, a new piece of information may be inconsistent with our set of beliefs and we must adapt our specification with this goal.

Let K be a set of CTL formulas and φ a single CTL formula, we say that K is *consistent* with φ if there is a model that satisfies all formulas in $K \cup \{\varphi\}$. For example, let $K = Cn(\{EFp, AG(p \rightarrow q)\})$, $\varphi_1 \equiv q$ and $\varphi_2 \equiv AG\neg q$, we have that K is consistent with φ_1 , but not with φ_2 . Every model that has the initial state labeled with p and q satisfies both K and φ_1 . However, in every model of K , p eventually holds (EFp) and so does q ($AG(p \rightarrow q)$), hence φ_2 cannot be satisfied.

The aim of specification revision is to maximally preserve the original beliefs. The revision here has a direct correspondence with the classical *partial meet* construction, being based on maximal consistent sets.

In the example above, the remainder set $K \perp \neg\varphi_2$ is a collection of sets $X, X_0, X_1, X_2, X_3, \dots$, such that:

1. $X = Cn(\{AG(p \rightarrow q)\})$,
2. $X_0 = Cn(\{EFp\} \cup \{AX^n(p \rightarrow q) \mid n > 0\})$,
3. $X_i = Cn(\{EFp\} \cup \{AX^n(p \rightarrow q) \mid n \neq i\} \cup \{EX^i(p \rightarrow q)\})$.

where $AX^n\psi$ is an alias to $\underbrace{AXAX \dots AX}_n\psi$.

Based on $K \perp \neg\varphi_2$, the partial meet revision $K * \varphi_2$ may produce as result

$$\begin{aligned}
 K * \varphi_2 &= Cn(\{AG\neg q, AG(p \rightarrow q)\}), \text{ or} \\
 K * \varphi_2 &= Cn(\{AG\neg q, AX(p \rightarrow q), AXAX(p \rightarrow q), \dots\}), \text{ or} \\
 K * \varphi_2 &= Cn(\{AG\neg q\})
 \end{aligned}$$

or several others possibilities depending of the choice of the selection function γ .

When trying to apply AGM-style belief revision for CTL, we see that several theoretical results cannot be applied due to the absence of compactness. Partial-meet constructions depend on computing the remainder set, and we have shown in Guerra and Wassermann (2017) that it is not always feasible. Recently, Ribeiro et al. (2018) have proposed an alternative construction that does not depend on compactness.

3.2 Revision of Models

Another way to describe the behavior of a system is structurally, explicit representing states and the transitions between them, which is a natural perspective for system designers.

Based on Zhang and Ding’s work (2008), we have introduced model revision in Guerra and Wassermann (2010). The *model revision* approach consists in, given structural models of systems as beliefs, repair these models changing minimally their structure in order to preserve the information given initially.

This minimality criterion may be an important factor for system designers. In many applications, the addition of new states may represent the need to develop new components, or new transitions that may mean a significant increase in the complexity of the concrete system.

The idea of CTL model revision is to use principles of belief revision theory to rationally choose minimal model modifications considering an application on static contexts. A modification is a composition of ground primitive update operations originally proposed in Zhang and Ding (2008):

- 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.

Models are then compared by their structural similarity: difference of states, transitions, labeling function. Given two CTL Models $\mathcal{M} = \langle AP, S, s_0, R, L \rangle$ and $\mathcal{M}' = \langle AP', S', s_0', R', L' \rangle$, we denote by $Diff_{PUI}(\mathcal{M}, \mathcal{M}')$, for each $PUI(i = 1, \dots, 5)$, the difference between \mathcal{M} and \mathcal{M}' , where

- $Diff_{PU1}(\mathcal{M}, \mathcal{M}') = R' - R$ (the set of pairs added to the relation).
- $Diff_{PU2}(\mathcal{M}, \mathcal{M}') = R - R'$ (the set of pairs removed from the relation).
- $Diff_{PU3}(\mathcal{M}, \mathcal{M}') = \{s \in S \cup S' \mid s \in L(p) - L'(p) \text{ or } s \in L'(p) - L(p) \text{ for some } p \in AP\}$ (the set of states whose labeling function has changed).
- $Diff_{PU4}(\mathcal{M}, \mathcal{M}') = S' - S$ (the set of added states).
- $Diff_{PU5}(\mathcal{M}, \mathcal{M}') = S - S'$ (the set of removed states).

In order to be admissible, a modification must be minimal in relation to all possible changes, i.e., if for some modification that transforms a model from inconsistent to consistent with some desired property, there is another modification that also produces this consistence, but with fewer structural changes w.r.t the original model, the later must be chosen. This leads to the following ordering definition:

Definition 6 (Guerra & Wassermann, 2010) Let \mathcal{W} be a set of CTL models and $\mathcal{M}_1 = \langle AP, S_1, s_1, R_1, L_1 \rangle$ and $\mathcal{M}_2 = \langle AP, S_2, s_2, R_2, L_2 \rangle$ two CTL models, we say that \mathcal{M}_1 is at least as near to \mathcal{W} as \mathcal{M}_2 , denoted to $\mathcal{M}_1 \leq_{\mathcal{W}} \mathcal{M}_2$, if and only if for every composition of primitive operations PU1–PU5 that transforms a model $\mathcal{M}' \in \mathcal{W}$ in \mathcal{M}_2 there is a composition that transforms a model $\mathcal{M} \in \mathcal{W}$ in \mathcal{M}_1 such that

1. For each i ($1 \leq i \leq 5$), $Diff_{PUI}(\mathcal{M}, \mathcal{M}_1) \subseteq Diff_{PUI}(\mathcal{M}', \mathcal{M}_2)$
2. If $Diff_{PU3}(\mathcal{M}, \mathcal{M}_1) = Diff_{PU3}(\mathcal{M}', \mathcal{M}_2)$, then for every s in $Diff_{PU3}(\mathcal{M}, \mathcal{M}_1)$, $\{p \in AP \mid s \in L_1(p)\} \subseteq \{p \in AP \mid s \in L_2(p)\}$

We denote by $\mathcal{M}_1 <_{\mathcal{W}} \mathcal{M}_2$ if $\mathcal{M}_1 \leq_{\mathcal{W}} \mathcal{M}_2$ and $\mathcal{M}_2 \not\leq_{\mathcal{W}} \mathcal{M}_1$.

In this way, we can order a set of models according to the structural similarity that they have with respect to a given referential model. We say that \mathcal{M}_1 is closer to \mathcal{M} than \mathcal{M}_2 , if (1) \mathcal{M}_1 is obtained from \mathcal{M} by applying primitive operations that cause fewer changes than those used to obtain \mathcal{M}_2 and (2) if the same states were affected by PU3 operations both in \mathcal{M}_1 and in \mathcal{M}_2 , then there were fewer changes on the set of propositional atoms of \mathcal{M}_1 states.

Based on this closeness order, Zhang and Ding define the notion of *admissible update* for model repair: a minimum change criterion by comparing the results of possible structural modifications in the models.

Definition 7 (Zhang & Ding, 2008) Let $\mathcal{M} = \langle AP, S, s_0, R, L \rangle$ be a model and φ a CTL formula that should hold in \mathcal{M} . A modification is called *admissible*, denoted by $Update(\mathcal{M}, \varphi)$, if and only if it produces a model that satisfies:

1. $Update(\mathcal{M}, \varphi) = \mathcal{M}'$ and $\mathcal{M}' \models \varphi$;
2. There is no \mathcal{M}'' such that $\mathcal{M}'' \models \varphi$ and $\mathcal{M}'' <_{\mathcal{M}} \mathcal{M}'$.

We denote by $Poss(Update(\mathcal{M}, \varphi))$ the set of all models that could be obtained from \mathcal{M} by admissible modifications.

Based on the closeness ordering defined by Zhang and Ding (2008), Guerra and Wassermann (2010) define a change operator \circ_c using a belief revision perspective.

Definition 8 (Guerra & Wassermann, 2010) Given two CTL formulas ψ and φ , we denote by $\psi \circ_c \varphi$ the result of a revision whose models are defined as

$$\text{Mod}(\psi \circ_c \varphi) = \text{Min}_{\text{Mod}(\psi)}(\text{Mod}(\varphi))$$

where $\text{Min}_{\mathcal{B}}(\mathcal{A})$ denotes the set of all minimal models of \mathcal{A} with respect to $\leq_{\mathcal{B}}$.

3.3 Differences Between Model and Specification Revision

For classical propositional logic, belief revision produces equivalent results if applied over sets of formulas or over propositional assignments (models) (Grove, 1988). This means that it is possible to define revision of belief sets in terms of relations of proximity of their possible models, and *vice-versa*. However, for CTL, as we show by means of an example, this interdefinability does not hold. If it did, it would be possible to define revision of temporal formulas in terms of current works on revision of models.

Let p_1 , p_2 and p_3 be the only propositional symbols in our system. Suppose that we believe that in our system the following properties are true: p_1 holds globally; if p_1 does not hold in some next state, a path where p_2 holds globally initiates in some of the next states; the later also holds to p_3 ; and that p_1 , p_2 and p_3 are mutually exclusive, i.e., at most one of them can hold in any state.

According to the above properties, let K be our initial belief base defined as

$$K = \text{Cn}(\{\text{AG}p_1, \text{EX}\neg p_1 \rightarrow \text{EXAG}p_2, \text{EX}\neg p_1 \rightarrow \text{EXAG}p_3, \varphi_{\text{mutex}}\})$$

where *mutex* stands for *mutually exclusive*, and φ_{mutex} expresses the fact that in no state in the future we have p_i and p_j both true, with $i \neq j$:

$$\varphi_{\text{mutex}} \equiv \neg\text{EF}(p_1 \wedge p_2) \wedge \neg\text{EF}(p_1 \wedge p_3) \wedge \neg\text{EF}(p_2 \wedge p_3).$$

Suppose that we realize that p_2 or p_3 must hold in some next state

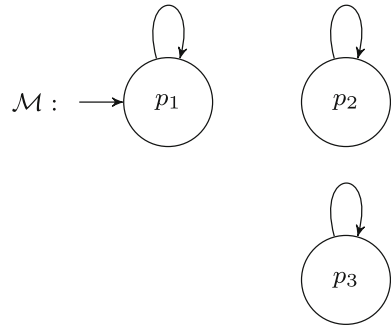
$$\text{EX}(p_2 \vee p_3).$$

This new piece of information is inconsistent with our current belief base K , then we need to revise K by $\text{EX}(p_2 \vee p_3)$. In this example we also assume as a fundamental property the mutual exclusion between the propositions, thus our goal is to ensure

$$\psi \equiv \text{EX}(p_2 \vee p_3) \wedge \varphi_{\text{mutex}}$$

For the specification revision $K * \psi$, we need to find all maximal subsets of K that are consistent with the new property ψ . In this example, $\text{EX}(p_2 \vee p_3)$ is inconsistent

Fig. 5 Model of the initial belief set



with AXp_1 , inferred from AGp_1 , so we must give up these beliefs. The only maximal consistent subset $X \in K \perp \neg\psi$ is given by

$$X = Cn(\{p_1, EXp_1, AXAXp_1, AXAXAXp_1, \dots, \varphi_{mutex}, EX\neg p_1 \rightarrow EXAGp_2, EX\neg p_1 \rightarrow EXAGp_3\})$$

Thus, as $K \perp \neg\psi$ is a singleton set, there is just one possible choice for γ and the only revision result is given by

$$K * \psi = Cn(X \cup \{EX(p_2 \vee p_3) \wedge \varphi_{mutex}\})$$

From the model revision perspective, our beliefs are now composed by the set \mathcal{K} of all models that satisfy the belief set K . Due to φ_{mutex} restriction, every model has states labeled by at most one of p_1, p_2 or p_3 . Due to AGp_1 , all possible models for our initial beliefs are bisimilar to \mathcal{M} in Fig. 5.

According to the model revision framework, we need to find all minimal structural modifications on the models of \mathcal{K} in order to produce models that satisfy $EX(p_2 \vee p_3) \wedge \varphi_{mutex}$. This results in the following revised set of models:

$$\mathcal{K}' = \mathcal{K} \circ_c (EX(p_2 \vee p_3) \wedge \varphi_{mutex})$$

Every model $M \in \mathcal{K}$ could be repaired by just one addition of transition (PU1) between a state p_1 (in an initial p_1 -loop) to a state p_2 or p_3 (in a p_2 -loop or p_3 -loop, respectively). Any additional modification is considered redundant according to the model revision framework. Figure 6 shows examples of model repairs.

The revision result \mathcal{K}' contains only those models that are structurally close to our original set of models.

Both approaches—revising sets of formulas and revising sets of models—rely on different notions of minimal change. On one side, there is no model in \mathcal{K}' that satisfies all formulas in specification revision K' . Every model in \mathcal{K}' can only satisfy one of $EX\neg p_1 \rightarrow EXAGp_2$ or $EX\neg p_1 \rightarrow EXAGp_3$, since only one transition was added. This results in a non-minimal change with respect to the original set of formulas.

On the other hand, there is no model of K' that could be used as a model revision result. In order to satisfy $EX\neg p_1 \rightarrow EXAGp_2$ and $EX\neg p_1 \rightarrow EXAGp_3$, all models

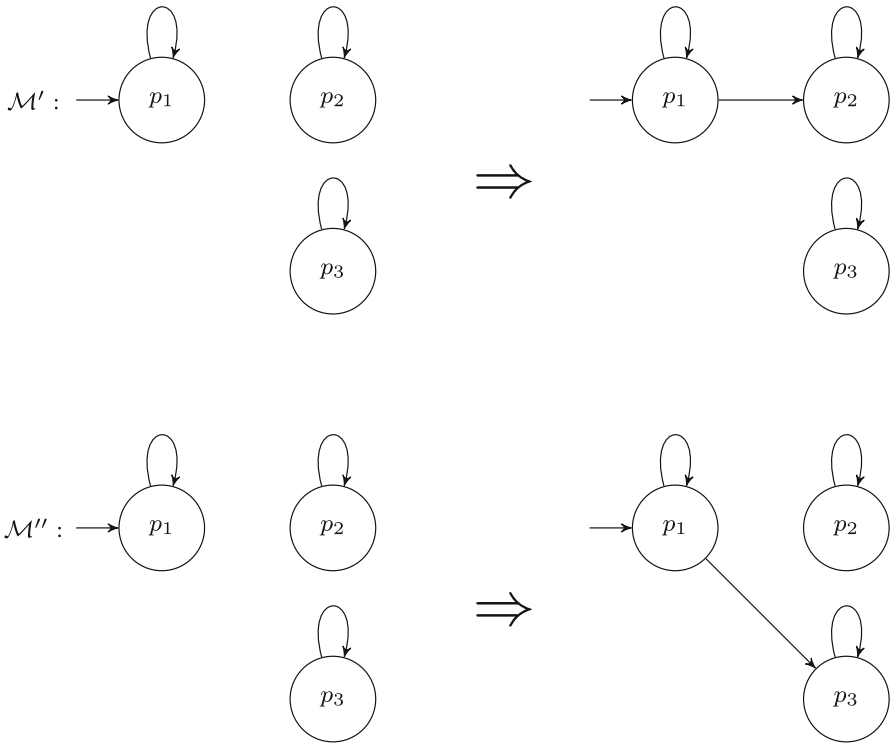
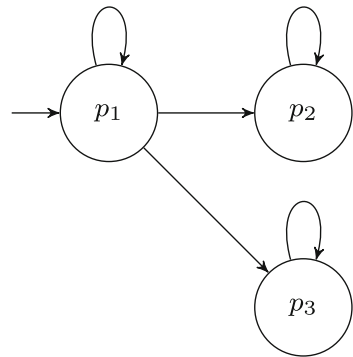


Fig. 6 Solutions to repair $M \in \mathcal{K}$

Fig. 7 A non minimal solution to model revision



of K' contain at least two new transitions with respect to \mathcal{K} . Figure 7 shows an example of a model of K' that is not minimal, since it is possible to satisfy the property with fewer modifications (see Fig. 6).

Based on this example, we can see that there is no model for K' that preserves the structural similarity as intended by the model revision. Likewise, there is no model in \mathcal{K}' that satisfies a subset of the specification that maximally preserves the initial properties, as intended by specification revision.

This example shows that, unlike belief revision over classical logic, revision over CTL in the syntactic or semantic levels may lead to different results. There are cases

where the two approaches coincide. If we remove the φ_{mutex} restriction from the our example, the model revision approach will produce at least one model where the specification result is satisfied (by labeling the initial state with all propositions).

The revision of sets of CTL formulas can be seen as a special case of the framework developed by Ribeiro et al. (2018) for non-compact logics. In that paper, a different version of the partial meet construction is proposed and then fully characterized by AGM-style postulates. In the rest of this paper, we will focus on the other side of the problem, characterizing model revision.

4 Revising Complete Models

In Guerra and Wassermann (2018), we propose two characterizations of the model repair operation: one based on AGM-like postulates where belief states are represented by sets of temporal formulas; and one based on postulates of rationality over structural changes on models. Our focus in this section is in the later characterization and how it is applied to describe the minimal change principle in model repair.

In this characterization, a repair operator is represented by one or more compositions of *atomic modifications*:

Definition 9 An *atomic modification* is a pair $\langle \mathcal{O}, \mathcal{D} \rangle$ such that $\mathcal{O} \in \{PU1, \dots, PU5\}$ denotes a primitive update and

1. if $\mathcal{O} = PU1$ or $\mathcal{O} = PU2$, then $\mathcal{D} \in S \times S$, indicating the transition to be added or removed;
2. if $\mathcal{O} = PU3$, then $\mathcal{D} \in S \times AP$, indicating a change of a label of a state; or
3. if $\mathcal{O} = PU4$ or $\mathcal{O} = PU5$, then $\mathcal{D} \in S$, indicating the state to be added or removed.

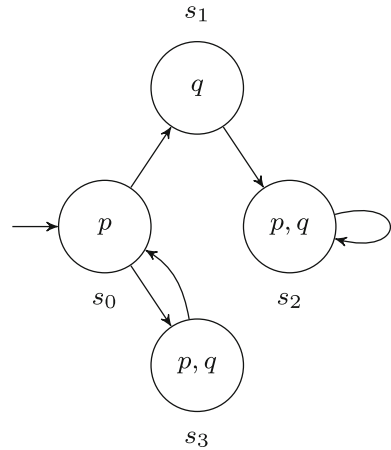
For example, $\langle PU2, (s_0, s_1) \rangle$ is an atomic modification that represents the removal of a relation between the s_0 and s_1 in a given model.

Definition 10 Let $\mathcal{M} = \langle AP, S, s_0, R, L \rangle$ be a Kripke structure and a an atomic modification. The *application* of a to \mathcal{M} results in a model $\mathcal{M}[a]$ such that:

1. if $a = \langle PU1, (s_i, s_j) \rangle$ and $s_i, s_j \in S$, then $\mathcal{M}[a] = \langle AP, S, s_0, R \cup \{(s_i, s_j)\}, L \rangle$;
2. if $a = \langle PU2, (s_i, s_j) \rangle$ and $(s_i, s_j) \in R$, then $\mathcal{M}[a] = \langle AP, S, s_0, R - \{(s_i, s_j)\}, L \rangle$;
3. if $a = \langle PU3, (s, p) \rangle$ and $s \notin L(p)$, then $\mathcal{M}[a] = \langle AP, S, s_0, R, L' \rangle$, where $L' = L$ except for $L'(p) = L(p) \cup \{s\}$;
4. if $a = \langle PU3, (s, p) \rangle$ and $s \in L(p)$, then $\mathcal{M}[a] = \langle AP, S, s_0, R, L' \rangle$, where $L' = L$ except for $L'(p) = L(p) - \{s\}$;
5. if $a = \langle PU4, (s) \rangle$, $\mathcal{M}[a] = \langle AP, S \cup \{s\}, s_0, R, L \rangle$;
6. if $a = \langle PU5, (s) \rangle$ and for all $(s_i, s_j) \in R$, $s \neq s_i$ and $s \neq s_j$, then $\mathcal{M}[a] = \langle AP, S - \{s\}, s_0, R, L \rangle$.
7. In all other cases, $\mathcal{M}[a] = \mathcal{M}$.

Definition 11 Let \mathcal{M} be a model, a *modification* Δ in \mathcal{M} is a finite sequence of atomic modifications $\Delta = \langle a_1, a_2, \dots, a_n \rangle$. We represent by $\mathcal{M}[\Delta]$ the model resulting from the application of Δ to \mathcal{M} , i.e., $\mathcal{M}[\Delta] = \mathcal{M}[a_1][a_2] \dots [a_n]$. In the case where $\Delta = \emptyset$ or that the application of Δ do not preserve Kripke models properties, as the serial transition relation over states, we have $\mathcal{M}[\Delta] = \mathcal{M}$.

Fig. 8 Model after the application of a modification



A modification Δ then represents a composition of sequence of atomic modifications. For example, let δ be the following sequence of modifications

$$\Delta = \{ \langle \text{PU4}, s_3 \rangle, \langle \text{PU3}, (s_3, p) \rangle, \langle \text{PU3}, (s_3, q) \rangle, \langle \text{PU2}, (s_0, s_2) \rangle, \langle \text{PUI}, (s_0, s_3) \rangle, \langle \text{PUI}, (s_3, s_0) \rangle \}$$

and \mathcal{M} be the model depicted in Fig. 4. The modification δ applied in \mathcal{M} generates the model $\mathcal{M}[\delta]$ depicted in Fig. 8. Note that when δ is applied in \mathcal{M} , the atomic modification $\langle \text{PU2}, (s_0, s_2) \rangle$ has no effect in the final result.

Let \mathcal{M} be a model, α a temporal formula, and $\mathcal{R}(\mathcal{M}, \alpha)$ a set of modifications given as a solution to the repair of \mathcal{M} given α . In Guerra and Wassermann (2018), we proposed the following postulates to define the expected rationality of $\mathcal{R}(\mathcal{M}, \alpha)$:

- (R*1) $\mathcal{R}(\mathcal{M}, \alpha) = \emptyset$ if and only if $\models \neg \alpha$
- (R*2) For all $\Delta \in \mathcal{R}(\mathcal{M}, \alpha)$, $\mathcal{M}[\Delta] \models \alpha$
- (R*3) If $\mathcal{M} \models \alpha$, then $\mathcal{R}(\mathcal{M}, \alpha) = \{\emptyset\}$
- (R*4) For all $\Delta \in \mathcal{R}(\mathcal{M}, \alpha)$, if $\Delta' \subset \Delta$, then $\mathcal{M}[\Delta'] \not\models \alpha$
- (R*5) For all $\Delta \in \mathcal{R}(\mathcal{M}, \alpha)$, there is Δ' such that $\mathcal{M}[\Delta][\Delta'] = \mathcal{M}$.

Postulate (R*1) states that the lack of a repair only occurs when α is unsatisfiable. Postulate (R*2) ensures the *success* of a repair by stating that every modification in $\mathcal{R}(\mathcal{M}, \alpha)$ might lead to a model that satisfies α . Postulate (R*3) states that in the case where \mathcal{M} satisfies α , its structure must be preserved. Postulate (R*4) is related to the *relevance* of modifications and states that every possible modification must contains only relevant atomic modifications to satisfy α . Finally, Postulate (R*5) states the reversibility of each modification in $\mathcal{R}(\mathcal{M}, \alpha)$, such that it is possible to recover the original model. Postulate (R*5) is a parallel to the AGM recovery postulate.³

Theorem 2 states that the described postulates indeed capture the rationality expected for a model repair operator.

³ In fact, it is trivially satisfied by operations PU1–PU5. However (R*1)–(R*5) were intended to be applied for any set of primitive operations

Theorem 2 (Guerra & Wassermann, 2018) *Let \mathcal{M} be a model and α a temporal formula, $\mathcal{M}' \in \text{Update}(\mathcal{M}, \alpha)$ if and only if there is a set of modifications $\mathcal{R}(\mathcal{M}, \alpha)$ that satisfies (R*1)–(R*5) and $\mathcal{M}' = \mathcal{M}[\Delta]$, for some $\Delta \in \mathcal{R}(\mathcal{M}, \alpha)$.*

In this section, we have focused on the repair of a single model. However, we do not always have a complete specification of a system. In the next section, we generalize the characterization of model repair in order to deal with partial models, where uncertainty about parts of the model can be represented.

5 Revising Partial Models

In the previous section, we presented the repair of a single model. In Guerra and Wassermann (2010), we have presented an approach that generalizes this operation to deal with sets of inconsistent models. The motivation is that a system designer could describe the main behaviors of a system, but not all of them, leaving some possibilities open for the system. For example, a semaphore system could be defined by saying that *a green light must occur after a red light*, without mentioning whether a yellow light should or not occur between them. This modelling leads to a set of possible models for the system, that we may need to deal with while performing a model repair. This kind of description is called *partial modeling*.

Larsen (1990) and Larsen and Thomsen (1988) propose to perform this kind of system modeling by using a type of structures that allow the distinction between the required and admissible behavior.

The models proposed by Larsen and Thomsen are a variant of labeled transitions systems that allow us to express the necessity or the possibility of each transition, and thus determine obligatory and admissible behaviors, respectively. The transitions are thus divided into two types: *must* transitions (required) and *may* transitions (possible). In their model, a necessary behavior corresponds to that behavior that makes use of only must transitions. In turn, an admissible behavior corresponds to that which makes use of both may and must transitions.

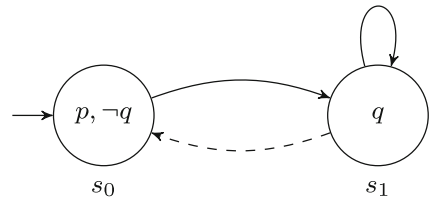
In Huth et al. (2001), Huth, Jagadeesan and Schmidt propose an extension of these models to incorporate modalities also in the state labeling. Thus, there are three possible truth values for a proposition in each state: true, false, or indeterminate. The models of Huth et al. (2001) are called *Kripke Modal Transition Systems* (KMTS).

Definition 12 (Huth et al., 2001) A Kripke Modal Transition System (KMTS) is a tuple $\mathcal{M} = \langle AP, S, s_0, R^+, R^-, L^+, L^- \rangle$ such that:

1. AP is a countable set of propositional atoms;
2. S is a finite set of states;
3. s_0 is the initial state;
4. $R^+ \subseteq R^- \subseteq S \times S$ are serial transition relations over S ;
5. $L^+ : AP \rightarrow \mathcal{P}(S)$ and $L^- : AP \rightarrow \mathcal{P}(S)$ are state labeling functions such that $L^+(p) \subseteq L^-(p)$, for all $p \in AP$.

In Definition 12, R^+ and R^- define must and may transitions, while L^+ and L^- define must and may labeling functions, respectively.

Fig. 9 Example of KMTS



Example 1 A computer server can be modelled into two main states: *idle* and *performing a required task*. It is desired that while in idle state the server keep listening for requests and that after receiving a request it immediately proceed to perform the required task. Observe that this information just partially describe the full system behavior. It is not said for example whether the system should return to the idle state after performing a task. It is also unclear whether it should keep listening for requests while performing tasks or not. These two behavior are admissible although none of them are initially required.

Figure 9 depicts a KMTS that models the partially specified behavior given in Example 1. In this KMTS, p, q are propositional atoms that represent respectively *the server is listening for requests* and *the server is performing a task*. By convention, full arrows represent must transitions, while dashed arrows represent transitions that belong exclusively to the set R^- . The may transition from s_1 to s_0 represents the possibility of the server to return to its idle state. Note that unlike LTSs, when an atom does not appear in a state, it is not interpreted as false, but as indeterminate. In state s_1 it might or not be the case whether the server keep listening for requests. This KMTS is formally defined by $\mathcal{M} = \langle \{p, q\}, \{s_0, s_1\}, s_0, \{(s_0, s_1), (s_1, s_1)\}, \{(s_0, s_1), (s_1, s_1), (s_1, s_0)\}, L^+, L^- \rangle$, where $L^+(p) = \{s_0\}$, $L^+(q) = \{s_1\}$, $L^-(p) = \{s_0, s_1\}$, $L^-(q) = \{s_1\}$.

In this work we adopt KMTS as the standard formalism for describing partial systems models. In this sense, partial models can be seen as a representation of the set of possible candidate models to implement the actual system behavior. As in Guerra et al. (2013), we use a KMTS as a compact representation of a set of Kripke models, building over it a characterization of the model revision operation defined by Guerra and Wassermann (2010).

We can obtain from a KMTS a set of possible concrete models of a system. We call this an *expansion* of a KMTS model (Guerra et al., 2013).

Definition 13 (Guerra et al., 2013) Let $\mathcal{M} = \langle AP, S, s_0, R^+, R^-, L^+, L^- \rangle$ be a KMTS, the *expansion* of \mathcal{M} into Kripke models is the set $\mathbb{K}(\mathcal{M})$ of all models $\mathcal{M}' = \langle AP', S', s'_0, R', L' \rangle$ such that $AP' = AP$, $S' = S$, $s'_0 = s_0$, $R^+ \subseteq R' \subseteq R^-$ and $L^+(p) \subseteq L'(p) \subseteq L^-(p)$, for all $p \in AP$.

The KMTS expansion produces a Kripke model for each indeterminacy in its transitions and labels. This set thus contains all possible models that may come to describe the final intended behavior of the system. Figure 10 depicts the expansion of the KMTS described in Fig. 9.

KMTS provides a compact representation of sets of Kripke models, thus modifying directly the structure of a KMTS can be more efficient than modifying each model in

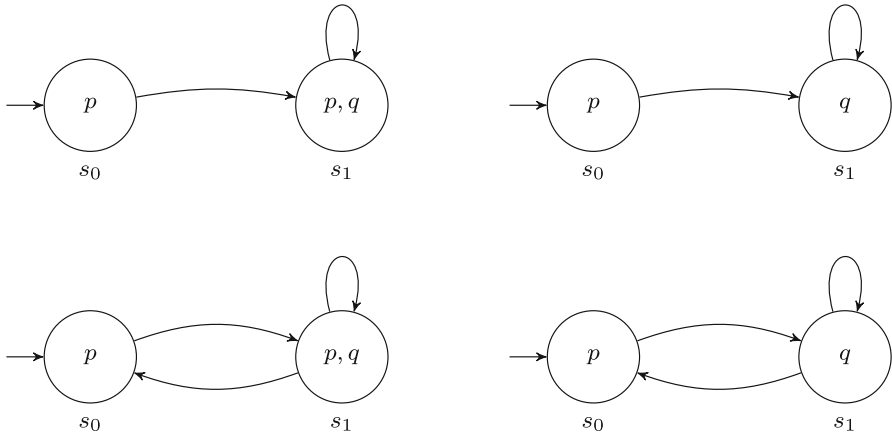


Fig. 10 Example of KMTS expansion

a set of models. In this section, we formalize an approach of how to perform model revision via KMTS. Our formalization is similar to that defined in Sect. 4 for the repair of a single Kripke model.

First, following Guerra et al. (2013), we define a new set of primitive operations, with focus on changes in KMTS:

- | | |
|----------------------------------|-------------------------------------|
| P1: Add a transition in R^- | P5: Change one state label in L^- |
| P2: Add a transition in R^+ | P6: Change one state label in L^+ |
| P3: Remove a transition of R^- | P7: Add a state in S |
| P4: Remove a transition of R^+ | P8: Remove a state of S |

Analogous to the case of Kripke models, we could define the concept of modification over KMTS:

Definition 14 An atomic modification over KMTS is a pair $\langle \mathcal{O}, \mathcal{D} \rangle$ such that $\mathcal{O} \in \{P1, \dots, P8\}$ denotes a primitive update and

1. if $\mathcal{O} \in \{P1, P2, P3, P4\}$, then $\mathcal{D} \in S \times S$, indicating the transition to be added or removed;
2. if $\mathcal{O} \in \{P5, P6\}$, then $\mathcal{D} \in S \times AP$, indicating a change of a state label; or
3. if $\mathcal{O} \in \{P7, P8\}$, then $\mathcal{D} \in S$, indicating the state to be added or removed.

Definition 15 Let $\mathcal{M} = \langle AP, S, s_0, R^+, R^-, L^+, L^- \rangle$ be a KMTS, the application of an atomic modification a over \mathcal{M} results in a model $\mathcal{M}[a]$ such that:

1. if $a = \langle P1, (s_i, s_j) \rangle$ and $s_i, s_j \in S$, then $\mathcal{M}[a] = \langle AP, S, s_0, R^+, R^- \cup \{(s_i, s_j)\}, L^+, L^- \rangle$
2. if $a = \langle P2, (s_i, s_j) \rangle$ and $(s_i, s_j) \in R^-$, then $\mathcal{M}[a] = \langle AP, S, s_0, R^+ \cup \{(s_i, s_j)\}, R^-, L^+, L^- \rangle$
3. if $a = \langle P3, (s_i, s_j) \rangle$ and $(s_i, s_j) \notin R^+$, then $\mathcal{M}[a] = \langle AP, S, s_0, R^+, R^- - \{(s_i, s_j)\}, L^+, L^- \rangle$

4. if $a = \langle \text{P4}, (s_i, s_j) \rangle$, then $\mathcal{M}[a] = \langle AP, S, s_0, R^+ - \{(s_i, s_j)\}, R^-, L^+, L^- \rangle$
 5. if $a = \langle \text{P5}, (s, p) \rangle$ and $s \notin L^+(p)$, then $\mathcal{M}[a] = \langle AP, S, s_0, R^+ - \{(s_i, s_j)\}, R^-, L^+, L'^- \rangle$ such that $L'^- = L^-$ except for $L'^-(p)$ where

$$L'^-(p) = \begin{cases} L^-(p) - \{s\}, & \text{if } s \in L^-(p) \\ L^-(p) \cup \{s\}, & \text{otherwise.} \end{cases}$$

6. if $a = \langle \text{P6}, (s, p) \rangle$ and $s \in L^-(p)$, then $\mathcal{M}[a] = \langle AP, S, s_0, R^+ - \{(s_i, s_j)\}, R^-, L'^+, L^- \rangle$ such that $L'^+ = L^+$ except for $L'^+(p)$ where

$$L'^+(p) = \begin{cases} L^+(p) - \{s\}, & \text{if } s \in L^+(p) \\ L^+(p) \cup \{s\}, & \text{otherwise.} \end{cases}$$

7. if $a = \langle \text{P7}, (s) \rangle$, then $\mathcal{M}[a] = \langle AP, S \cup \{s\}, s_0, R^+, R^-, L^+, L^- \rangle$
 8. if $a = \langle \text{P8}, (s) \rangle$ and for all $(s_i, s_j) \in R^-, s \neq s_i$ and $s \neq s_j$, then $\mathcal{M}[a] = \langle AP, S - \{s\}, s_0, R^+, R^-, L^+, L^- \rangle$
 9. In all other cases, $\mathcal{M}[a] = \mathcal{M}$.

Note that no primitive operation should violate the KMTS definition. For example, P2 can only be applied if the transition to be added in R^+ is already present in R^- . To completely remove a transition $(s, r) \in R^+ \cap R^-$ from a KMTS, we need to first apply P4 to remove (s, r) from R^+ , then use P3 to remove it from R^- , otherwise it would violate the condition that $R^+ \subseteq R^-$.

Similar to the modification in Kripke models, a modification in KMTS is a composition of primitive operations P1-P8 capable of generating a new KMTS.

A repair in a KMTS corresponds to a set of modifications that could change a model in order to satisfy a desired property. As before, we expect these modifications to be made rationally, following a principle of minimal structural change. We adopt here as a criterion of rationality the set of change postulates (R*1)–(R*5) described in Sect. 4.

Definition 16 Let \mathcal{M} be a model and α a temporal formula, a *repair* $\mathcal{R}(\mathcal{M}, \alpha)$ is said *admissible* if and only if it satisfies postulates (R*1)–(R*5).

Our goal however is to show that model revision (Guerra & Wassermann, 2010) can be performed through modifications in KMTS. In this sense, we need an additional constraint for KMTS repair, limiting it to modifications which result in KMTS that are equivalent to Kripke structures.⁴ We express this restriction by the following postulate:

(R*6) For all $\Delta \in \mathcal{R}(\mathcal{M}, \alpha)$, $\mathcal{M}[\Delta]$ is equivalent to a Kripke structure

A KMTS repair that satisfies (R*1)–(R*6) actually produces models that belongs to the set of models generated by the model update approach of Zhang and Ding (2008).

⁴ A KMTS is equivalent to a Kripke structure if its expansion generates a set with a single Kripke model. This occurs when in a KMTS there are no undeterminacies about labels or transitions.

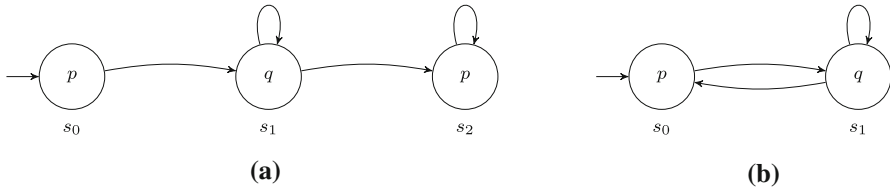


Fig. 11 Example of possible results of KMTS repair

Proposition 1 *If $\mathcal{R}\langle \mathcal{M}, \alpha \rangle$ satisfies (R*1)–(R*6), then for all $\Delta \in \mathcal{R}\langle \mathcal{M}, \alpha \rangle$, $\mathcal{M}[\Delta] \in \text{Poss}(\text{Update}(\mathcal{M}', \alpha))$ for some $\mathcal{M}' \in \mathbb{K}(\mathcal{M})$.*

Proof This result is a consequence of Theorem 2, since $\mathcal{M}[d]$ is equivalent to a Kripke structure (Postulate (R*6)) and each primitive modification in Kripke models PU1–PU5 could be achieved by equivalent modifications P1–P8 in KMTS (for example, $\langle \text{PUI}, (s, r) \rangle$ is equivalent to perform the following sequence $\langle \text{P1}, (s, r) \rangle, \langle \text{P2}, (s, r) \rangle$). □

However, only the addition of postulate (R*6) is not sufficient to ensure that a repair in KMTS produces the expected result of the model revision approach described in Guerra and Wassermann (2010).

Example 2 Let \mathcal{M} be the KMTS depicted in Fig. 9 and $\alpha = \text{EXEX}(p \wedge \neg q)$ a temporal property. The repair $\mathcal{R}\langle \mathcal{M}, \alpha \rangle = \{\Delta\}$ such that $\Delta = \{\langle \text{P3}, (s_1, s_0) \rangle, \langle \text{P5}, (s_1, p) \rangle, \langle \text{P7}, s_2 \rangle, \langle \text{P1}, (s_1, s_2) \rangle, \langle \text{P2}, (s_1, s_2) \rangle, \langle \text{P1}, (s_2, s_2) \rangle, \langle \text{P2}, (s_2, s_2) \rangle\}$ satisfies postulates (R*1)–(R*6), however does not result in a model that belongs to the revision of the set $\mathbb{K}(\mathcal{M})$.

Figure 11a depicts the model that produced by the repair described in Example 2. However, the expansion $\mathbb{K}(\mathcal{M})$ already contains a model for which the property $\text{EXEX}(p \wedge \neg q)$ holds (Fig. 11b). In this scenario, model revision criteria tend to keep the later model instead of producing the first one.

To obtain a model according to the result of model revision, we need to restrict the modifications in KMTS to those which make the best use of the model uncertainties. We define this principle of relevance of the modifications by means of a new postulate of rationality. First, however, we need to define what we call *alternative choices* of a modification:

Definition 17 Let Δ be a KMTs modification, the set of *alternative choices* of Δ , denoted by $ea(\Delta)$, is defined by

$$\begin{aligned}
 ea(\Delta) = & \{ \langle \text{P2}, (s, r) \rangle \mid \langle \text{P3}, (s, r) \rangle \in \Delta \text{ and } (s, r) \in R^- \setminus R^+ \} \cup \\
 & \{ \langle \text{P3}, (s, r) \rangle \mid \langle \text{P2}, (s, r) \rangle \in \Delta \text{ and } (s, r) \in R^- \setminus R^+ \} \cup \\
 & \{ \langle \text{P5}, (s, p) \rangle \mid \langle \text{P6}, (s, p) \rangle \in \Delta \text{ and } s \in L^-(p) \setminus L^+(p) \} \cup \\
 & \{ \langle \text{P6}, (s, p) \rangle \mid \langle \text{P5}, (s, p) \rangle \in \Delta \text{ and } s \in L^-(p) \setminus L^+(p) \}.
 \end{aligned}$$

In order to produce a KMTS equivalent to a Kripke structure, a modification Δ must contain changes that refine all undeterminations of the KMTS. In the Example 2,

when we performed the operation $\langle P3, (s_1, s_0) \rangle$ we chose to remove a *may* transition from the KMTS. An alternative choice would be to maintain this transition and add it to the set of *must* transitions ($\langle P2, (s_1, s_0) \rangle$). Similarly, when we make $\langle P5, (s_1, p) \rangle$ we choose to transform p from indeterminate to false, alternatively we could make p true in s_1 ($\langle P6, (s_1, p) \rangle$).

The set $ea(\Delta)$ contains thus the dual modifications to those present in Δ that perform refinement of uncertainties. We then use this notion to define a new postulate of rationality related to the choices made by a repair operator:

(R*7) For all $\Delta \in \mathcal{R}(\mathcal{M}, \alpha)$, there is no $\Delta' \subset \Delta$ such that for some $\Delta'' \subseteq ea(\Delta)$, $\mathcal{M}[\Delta' \cup \Delta''] \models \alpha$.

Postulate (R*7) states that in order to satisfy a property, different choices about uncertainties should not make irrelevant parts of a modification, in the sense that if we adopted a different set of choices for the solution, it would be possible satisfy the property using a proper subset of this modification. To satisfy this postulate, the modification Δ of Example 2 could not belong to $\mathcal{R}(\mathcal{M}, \alpha)$ since that for $\Delta' = \{\langle P5, (s_1, p) \rangle\}$ and $\Delta_{ea} = \{\langle P2, (s_1, s_0) \rangle\}$ we have $\Delta' \subset \Delta$, $\Delta_{ea} \subseteq ea(\Delta)$ and $\mathcal{M}[\Delta' \cup \Delta_{ea}] \models \alpha$.

In Theorem 3, we show that if the KMTS repair operator satisfies postulates (R*1)–(R*7), it generates only models belonging to the result of model revision as defined in Guerra and Wassermann (2010).

Theorem 3 *If $\mathcal{R}(\mathcal{M}, \alpha)$ satisfies (R*1)–(R*7), then for all $\Delta \in \mathcal{R}(\mathcal{M}, \alpha)$, $\mathcal{M}[\Delta] \in \text{Min}_{\mathbb{K}(\mathcal{M})}(\text{Mod}(\alpha))$.*

Proof Let $\mathcal{M} = \langle AP, S, s_0, R^+, R^-, L^+, L^- \rangle$ be a KMTS and α a temporal formula. Suppose, for the purpose of contradiction, that $\mathcal{R}(\mathcal{M}, \alpha)$ satisfies (R*1)–(R*7) and that there is a $d \in \mathcal{R}(\mathcal{M}, \alpha)$ such that $\mathcal{M}[d] \notin \text{Min}_{\mathbb{K}(\mathcal{M})}(\text{Mod}(\alpha))$.

Since $\mathcal{R}(\mathcal{M}, \alpha)$ satisfies (R*4), we have that $\mathcal{M}[d] \in \text{Mod}(\alpha)$, thus there must be $\mathcal{M}' \in \text{Mod}(\alpha)$ such that $\mathcal{M}' <_{\mathbb{K}(\mathcal{M})} \mathcal{M}[d]$. Therefore, there are $\mathcal{M}_1, \mathcal{M}_2 \in \mathbb{K}(\mathcal{M})$ such that $\mathcal{M}[d] \in \text{Poss}(\text{Update}(\mathcal{M}_1, \alpha))$, $\mathcal{M}' \in \text{Poss}(\text{Update}(\mathcal{M}_2, \alpha))$ and

- (i) For all $i = 1..5$, $\text{Diff}_{\text{PU}}(\mathcal{M}_2, \mathcal{M}') \subseteq \text{Diff}_{\text{PU}}(\mathcal{M}_1, \mathcal{M}[d])$ and, for some $j = 1..5$, we have $\text{Diff}_{\text{PU}}(\mathcal{M}_1, \mathcal{M}[d]) \not\subseteq \text{Diff}_{\text{PU}}(\mathcal{M}_2, \mathcal{M}')$; or
- (ii) For all $i = 1..5$, $\text{Diff}_{\text{PU}}(\mathcal{M}_2, \mathcal{M}') = \text{Diff}_{\text{PU}}(\mathcal{M}_1, \mathcal{M}[d])$ and for all $p \in AP$ and $s \in \text{Diff}_{\text{PU3}}(\mathcal{M}_2, \mathcal{M}')$, we have that $s \in \text{diff}(L'(p), L_2(p))$ implies $s \in \text{diff}(L(p), L_1(p))$, but for some $q \in AP$ and $r \in \text{diff}(L(p), L_1(p))$, we have $r \in \text{diff}(L(q), L_1(q))$ and $r \notin \text{diff}(L'(q), L_2(q))$.

Let d' be a modification in \mathcal{M} such that $\mathcal{M}' = \mathcal{M}[d']$ and that does not contain irrelevant primitive operations (i.e., there is no subset of d' that applied to \mathcal{M} also results in \mathcal{M}'). The difference between d' and d lies in the following cases:

1. If $\langle P1, (s, r) \rangle \in d' \setminus d$, then $\langle P2, (s, r) \rangle \in d'$, otherwise d' would not be minimal or $\mathcal{M}[d']$ would not be a Kripke structure. Thus, we have that $(s, r) \in \text{Diff}_{\text{PU}}(\mathcal{M}_2, \mathcal{M}[d'])$ and $(s, r) \in \text{Diff}_{\text{PU}}(\mathcal{M}_1, \mathcal{M}[d])$, therefore $\langle P2, (s, r) \rangle \in d$ and $\langle P1, (s, r) \rangle \in d$, a contradiction.
2. If $\langle P2, (s, r) \rangle \in d' \setminus d$, then

- (a) If $(s, r) \notin R^-$, then $(s, r) \in \text{Diff}_{\text{PUI}}(\mathcal{M}_2, \mathcal{M}[d'])$ and $(s, r) \in \text{Diff}_{\text{PUI}}(\mathcal{M}_1, \mathcal{M}[d])$. Therefore $\langle P2, (s, r) \rangle \in d$, a contraction.
- (b) If $(s, r) \in R^+$, then d' would not be minimal, a contraction.
- (c) If $(s, r) \in R^- \setminus R^+$, then $\langle P3, (s, r) \rangle \in d$, since $\langle P2, (s, r) \rangle \in d$ or $\langle P3, (s, r) \rangle \in d$, otherwise $\mathcal{M}[d]$ would not be a Kripke structure and $\mathcal{R}\langle \mathcal{M}, \alpha \rangle$ would violate postulate (R*6).
3. If $\langle P3, (s, r) \rangle \in d' \setminus d$, then
- (a) If $(s, r) \notin R^-$, then d' would not be minimal, a contraction.
- (b) If $(s, r) \in R^+$, then $(s, r) \in \text{Diff}_{\text{PL2}}(\mathcal{M}_2, \mathcal{M}[d'])$ and $(s, r) \in \text{Diff}_{\text{PL2}}(\mathcal{M}_1, \mathcal{M}[d])$. Thus $\langle P3, (s, r) \rangle \in d$, a contradiction.
- (c) If $(s, r) \in R^- \setminus R^+$, then $\langle P2, (s, r) \rangle \in d$, since $\langle P2, (s, r) \rangle \in d$ or $\langle P3, (s, r) \rangle \in d$, otherwise $\mathcal{M}[d]$ would not be a Kripke structure and $\mathcal{R}\langle \mathcal{M}, \alpha \rangle$ would violate postulate (R*6).
4. If $\langle P4, (s, r) \rangle \in d' \setminus d$, then $\langle P3, (s, r) \rangle \in d'$, then d' would not be minimal or $\mathcal{M}[d']$ would not be a Kripke structure. Thus, we have that $(s, r) \in \text{Diff}_{\text{PL2}}(\mathcal{M}_2, \mathcal{M}[d'])$ and $(s, r) \in \text{Diff}_{\text{PL2}}(\mathcal{M}_1, \mathcal{M}[d])$, therefore $\langle P3, (s, r) \rangle \in d$ and $\langle P4, (s, r) \rangle \in d$, a contradiction.
5. If $\langle P5, (s, p) \rangle \in d' \setminus d$, then
- (a) If $s \notin L^-(p)$, then also $\langle P6, (s, p) \rangle \in d'$, otherwise $\mathcal{M}[d']$ would not be a Kripke structure. Thus, we have that $s \in \text{Diff}_{\text{PUB}}(\mathcal{M}_2, \mathcal{M}[d'])$ and $s \in \text{Diff}_{\text{PUB}}(\mathcal{M}_1, \mathcal{M}[d])$, therefore $\langle P6, (s, p) \rangle \in d$ and $\langle P5, (s, p) \rangle \in d$, a contradiction.
- (b) If $s \in L^-(p)$ and $s \in L^+(p)$, then also $\langle P6, (s, p) \rangle \in d'$, due to the restriction over primitive update $P5$. Thus, we have that $s \in \text{Diff}_{\text{PUB}}(\mathcal{M}_2, \mathcal{M}[d'])$ and $s \in \text{Diff}_{\text{PUB}}(\mathcal{M}_1, \mathcal{M}[d])$, and also $\langle P6, (s, p) \rangle \in d$ and $\langle P5, (s, p) \rangle \in d$, a contradiction.
- (c) If $s \in L^-(p)$ and $s \notin L^+(p)$, then $\langle P6, (s, p) \rangle \in d$, since $\langle P5, (s, r) \rangle \in d$ or $\langle P6, (s, r) \rangle \in d$, otherwise $\mathcal{M}[d]$ would not be a Kripke structure and $\mathcal{R}\langle \mathcal{M}, \alpha \rangle$ would violate postulate (R*6).
6. If $\langle P6, (s, p) \rangle \in d' \setminus d$, then
- (a) If $s \in L^+(p)$, then also $\langle P5, (s, p) \rangle \in d'$, otherwise $\mathcal{M}[d']$ would not be a Kripke structure. Thus, we have that $s \in \text{Diff}_{\text{PUB}}(\mathcal{M}_2, \mathcal{M}[d'])$ and $s \in \text{Diff}_{\text{PUB}}(\mathcal{M}_1, \mathcal{M}[d])$, therefore $\langle P5, (s, p) \rangle \in d$ and $\langle P6, (s, p) \rangle \in d$, a contradiction.
- (b) If $s \notin L^+(p)$ and $s \notin L^-(p)$, then also $\langle P5, (s, p) \rangle \in d'$, due to the restriction over primitive update $P6$. Thus, we have that $s \in \text{Diff}_{\text{PUB}}(\mathcal{M}_2, \mathcal{M}[d'])$ and $s \in \text{Diff}_{\text{PUB}}(\mathcal{M}_1, \mathcal{M}[d])$, and also $\langle P5, (s, p) \rangle \in d$ and $\langle P6, (s, p) \rangle \in d$, a contradiction.
- (c) If $s \notin L^+(p)$ and $s \in L^-(p)$, then $\langle P5, (s, p) \rangle \in d$, since $\langle P5, (s, r) \rangle \in d$ or $\langle P6, (s, r) \rangle \in d$, otherwise $\mathcal{M}[d]$ would not be a Kripke structure and $\mathcal{R}\langle \mathcal{M}, \alpha \rangle$ would violate postulate (R*6).

7. If $\langle P7, s \rangle \in d' \setminus d$, then $s \in \text{Diff}_{\text{PU4}}(\mathcal{M}_2, \mathcal{M}[d'])$ and $s \in \text{Diff}_{\text{PU4}}(\mathcal{M}_1, \mathcal{M}[d])$, since $\langle P7, s \rangle \in d$, a contradiction.
8. If $\langle P8, s \rangle \in d' \setminus d$, then $s \in \text{Diff}_{\text{PU5}}(\mathcal{M}_2, \mathcal{M}[d'])$ and $s \in \text{Diff}_{\text{PU5}}(\mathcal{M}_1, \mathcal{M}[d])$, therefore $\langle P8, s \rangle \in d$, a contradiction.

In these cases, the only situations that do not lead to contradictions are those where the modification in d' are those that belong to $ea(d)$. However, this violates postulate (R*7), which contradicts the assumption that $\mathcal{R}\langle \mathcal{M}, \alpha \rangle$ satisfies all postulates. Therefore there is no such model \mathcal{M}' where $\mathcal{M}' <_{\mathbb{K}(\mathcal{M})} \mathcal{M}[d]$ and thus $\mathcal{M}[d]$ is indeed a repair solution according to model revision. \square

Theorem 3 shows that we can find repair solutions according to the minimal principle of the model revision using KMTS and modifications over them.

5.1 Computational Properties

Zhang and Ding (2008), show that CTL Model Update is co-NP-complete due to the complexity of checking whether a given model is an admissible update.

Theorem 4 (Zhang & Ding, 2008) *Given two CTL models, \mathcal{M} and \mathcal{M}' , and a CTL formula α , it is co-NP-complete to decide whether \mathcal{M}' is an admissible model update of \mathcal{M} to satisfy α .*

Although Guerra and Wassermann (2010) do not discuss computational properties of CTL Model Revision, its approach is at least as hard as CTL Model Update since Definition 7 is a special case of Definition 6 where \mathcal{W} has a single model, $\mathcal{W} = \{\mathcal{M}\}$. However, model revision is heavily based on a full set comparison in order to determine whether a repair candidate is minimal according to $\leq_{\mathcal{W}}$. One main concern is that the size of the set $\leq_{\mathcal{W}}$ tends to grow according to the number of uncertainties. The problem get worse when we need use compact representations like KMTS to define \mathcal{W} since any new uncertainty might doubles the size of \mathcal{W} .

Theorem 3 shows however that postulates (R*1)–(R*7) can ensure model revision results and thus they could be used to perform model revision. Similar to Zhang and Ding (2008), we show in Theorem 5 that to verify whether a repair satisfies (R*1)–(R*7) is also co-NP-complete.

Theorem 5 *Let \mathcal{M} be a KMTS and α a satisfiable CTL formula, it is co-NP-complete to check whether a given repair $\mathcal{R}\langle \mathcal{M}, \alpha \rangle$ satisfies (R*1)–(R*7).*

Proof *Membership proof* First, we need to show that the problem is in co-NP. For this purpose, we consider the complement problem: checking whether a modification $\mathcal{R}\langle \mathcal{M}, \alpha \rangle$ do not satisfy (R*1)–(R*7). For postulate (R*1), since α is satisfiable, we make a straightforward verification if $\mathcal{R}\langle \mathcal{M}, \alpha \rangle = \emptyset$. For postulate (R*2), we need to compute $\mathcal{M}[\Delta]$ for $\Delta \in \mathcal{R}\langle \mathcal{M}, \alpha \rangle$ and then verify if $\mathcal{M}[\Delta] \not\models \alpha$. The first step consists in rewriting the codification of \mathcal{M} according to Δ what can be done in polynomial time. The second step consists in performing a model checking, that takes time $O(|\mathcal{M}| \times |\alpha|)$ (Clarke et al., 1999). Both steps can be performed in polynomial time. For postulate (R*3), we need to verify if $\mathcal{M} \models \alpha$, that takes time $O(|\mathcal{M}|^2 \times |\alpha|)$

(Huth, 2002), and $\mathcal{R}\langle \mathcal{M}, \alpha \rangle \neq \{\emptyset\}$. Both steps can be performed in polynomial time. For postulate (R*4), we first make a non-deterministic guess of a $\Delta' \subset \Delta$ and then check if $\mathcal{M}[\Delta'] \models \alpha$, what again can be done in polynomial time. Postulate (R*5) is always satisfied since primitive operations P1–P8 are reversible. For postulate (R*6) we check if the expansion of $\mathbb{K}(\mathcal{M}[\Delta])$ has more than one element. And finally, for postulate (R*7) we make two non-deterministic guesses, $\Delta' \subset \Delta$ and $\Delta'' \subseteq ea(\Delta)$, and then we check if $\mathcal{M}[\Delta'] \models \alpha$. Therefore, all postulates verification can be achieved in polynomial time with a non-deterministic Turing machine.

Hardness proof Second, we show a polynomial reduction of a known co-NP-complete problem. Here, we show a polynomial time reduction from the problem of deciding whether a propositional formula φ is valid to the problem of deciding whether a set do modifications satisfies the postulates (R*1)–(R*7). Let A_φ be the set of all propositional atoms occurring in φ and a, b two propositional atoms that do not occur in φ . We then specify a KMTS $\mathcal{M} = (A_\varphi \cup \{a, b\}, \{s_0\}, s_0, \{(s_0, s_0)\}, \{(s_0, s_0)\}, L, L)$ such that $L(a) = L(b) = \emptyset$ and $L(p) = \{s_0\}$ for $p \in A_\varphi$. Note that \mathcal{M} is equivalent to a Kripke model with a single state where all atoms in A_φ are assigned to true and a, b are assigned to false. Let $\psi = \bigwedge_{p \in A_\varphi} \neg p$, we define a formula $\alpha = ((\varphi \rightarrow a) \wedge (b \wedge \psi)) \vee (\neg\varphi \wedge a)$ and a set of modifications $\mathcal{R}\langle \mathcal{M}, \alpha \rangle = \{\Delta\}$ where $\Delta = \{\langle P5, (s_0, p) \mid p \in A_\varphi \cup \{a, b\} \rangle \cup \{\langle P6, (s_0, p) \mid p \in A_\varphi \cup \{a, b\} \rangle\}$. We show that φ is valid if and only if $\mathcal{R}\langle \mathcal{M}, \alpha \rangle$ satisfies (R*1)–(R*7).

Postulates (R*1), (R*2), (R*3), (R*5) and (R*6) are trivially satisfied since α is a satisfiable, $\mathcal{R}\langle \mathcal{M}, \alpha \rangle \neq \emptyset$, $\mathcal{M}[\Delta] \models \alpha$, $\mathcal{M} \not\models \alpha$, $\mathcal{M}[\Delta][\Delta] = \mathcal{M}$ and $\mathcal{M}[\Delta]$ is equivalent to a Kripke structure. Postulate (R*7) is equivalent to (R*4) since $ea(\Delta) = \emptyset$. And finally, for postulates (R*4) we have two cases:

Case 1 If φ is valid, then must be the case that $\mathcal{M}[\Delta]$ satisfy $(\varphi \rightarrow a) \wedge (b \wedge \psi)$. Thus postulate (R*4) is satisfied since there is no subset $\Delta' \subset \Delta$ that makes a, b, ψ hold in s_0 , simultaneously, while preserving postulate (R*6).

Case 2 If φ is not valid, then there is a subset $A'_\varphi \subseteq A_\varphi$ such that A'_φ entails $\neg\varphi$. Let $\Delta' = \{\langle P5, (s_0, p) \mid p \in A'_\varphi \cup \{a\} \rangle \cup \{\langle P6, (s_0, p) \mid p \in A'_\varphi \cup \{a\} \rangle\}$, we have that $\mathcal{M}[\Delta'] \models \neg\varphi \wedge a$. Thus postulate (R*4) is not satisfied since $\Delta' \subset \Delta$ and $\mathcal{M}[\Delta'] \models \alpha$.

Therefore we have that φ is valid if and only if $\mathcal{R}\langle \mathcal{M}, \alpha \rangle$ satisfies (R*1)–(R*7). \square

6 Conclusions

In this work, we investigate an approach to repair inconsistencies in formal system specifications based on belief revision theory (Alchourron et al., 1985). We show that this problem can be divided into two subproblems: the repair of specifications described by means of temporal formulas and the repair of systems models by means of minimal structural changes on these. In this work, we investigate an approach to repair inconsistencies in formal system specifications based on belief revision theory (Alchourron et al., 1985). We show that this problem can be divided into two subproblems: the repair of specifications described by means of temporal formulas and the repair of systems models by means of minimal structural changes on these models. We

analyzed several issues related to the problem and present a formal characterization of revision over fully and partially specified models.

We have investigated the problem of model revision based on the approaches presented on Guerra and Wassermann (2010) and Zhang and Ding (2008). In these works, the authors show that their repair operators satisfy a set of rationality postulates, without however demonstrating that the set of postulates they use completely characterizes all possible model repair operators.

We use the same principles of model update described in Zhang and Ding (2008) to formally define a structural change operation over models and then propose a set of postulates over structural modifications. We show in Guerra and Wassermann (2018) that the proposed postulates indeed capture the expected rationality for the repair problem and that they have a direct relation with the postulates for sets of formulas.

Finally we developed an approach focused on partially specified models, based on a new set of primitive operations. We define rationality criteria for these operators based on those postulates defined in Guerra and Wassermann (2018). We show that by interpreting a Kripke Modal Transition System as a compact representation of sets of Kripke models we can perform the model revision of Guerra (2010) through repair on KMTS. The key point of this definition of rationality is the postulate that establishes the relevance of modifications with respect to the refinements of uncertainties.

In future work, we plan to propose implementations of model repair operations as in Guerra and Wassermann (2010); Zhang and Ding (2008) for both complete and partial representation of system models. We also intend to investigate the model change approach beyond structural metrics, evaluating the application of the proposed theory in more general applications.

Funding Paulo T. Guerra was funded by Grant #2010/15392-3, São Paulo Research Foundation (FAPESP). This work was developed under the Brazilian Research Council (CNPq) Grant #447178/2014-8.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical Approval This work does not contain any studies with human participants or animals performed by any of the authors.

References

- Alchourron, C. E., Gärdenfors, P., & Makinson, D. (1985). On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2), 510–530.
- Alchourron, C. E., & Makinson, D. (1982). On the logic of theory change: Contraction functions and their associated revision functions. *Theoria*, 48(1), 14–37. <https://doi.org/10.1111/j.1755-2567.1982.tb00480.x>
- Baier, C., & Katoen, J. P. (2008). *Principles of model checking*. MIT Press. <https://doi.org/10.1093/comjnl/bxp025>
- Buccafurri, F., Eiter, T., Gottlob, G., & Leone, N. (1999). Enhancing model checking in verification by AI techniques. *Artificial Intelligence*, 112(1–2), 57–104. [https://doi.org/10.1016/S0004-3702\(99\)00039-9](https://doi.org/10.1016/S0004-3702(99)00039-9)

- Burch, J., Clarke, E., McMillan, K., Dill, D., & Hwang, L. (1992). Symbolic model checking: 10²⁰ States and Beyond. *Information and Computation*, 98(2), 142–170. [https://doi.org/10.1016/0890-5401\(92\)90017-A](https://doi.org/10.1016/0890-5401(92)90017-A)
- Chatzieftheriou, G., Bonakdarpour, B., Smolka, S. A., & Katsaros, P. (2012). Abstract model repair. In A. Goodloe & S. Person (Eds.), *NASA formal methods. Lecture notes in computer science* (Vol. 7226, pp. 341–355). Springer. https://doi.org/10.1007/978-3-642-28891-3_32
- Clarke, E. M., & Emerson, E. A. (1982). Design and synthesis of synchronization skeletons using branching time temporal logic. In D. Kozen (Ed.), *Logics of programs. Lecture notes in computer science* (Vol. 131, pp. 52–71). Springer. <https://doi.org/10.1007/BFb0025774>
- Clarke, E. M., Emerson, E. A., & Sistla, A. P. (1986). Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 8(2), 244–263. <https://doi.org/10.1145/5397.5399>
- Clarke, E. M., Grumberg, O., & Peled, D. A. (1999). *Model checking*. Springer.
- Cresswell, M. J., & Hughes, G. E. (2012). *A new introduction to modal logic*. Routledge.
- Finger, M., & Wassermann, R. (2008). Revising specifications with CTL properties using bounded model checking. In *Proceedings of the 19th Brazilian symposium on artificial intelligence (SBIA'08), Lecture notes in artificial intelligence* (Vol. 5249, pp. 157–166). Springer.
- Grove, A. (1988). Two modellings for theory change. *Journal of Philosophical Logic*, 17(2), 157–170. <https://doi.org/10.1007/BF00247909>
- Guerra, P. T. (2010). Revisão de Modelos CTL. Master's thesis, Universidade de São Paulo.
- Guerra, P. T., Andrade, A., & Wassermann, R. (2013). Toward the revision of CTL models through Kripke modal transition systems. In J. Iyoda & L. M. de Moura (Eds.), *Formal methods: Foundations and applications. 16th Brazilian symposium on formal methods (SBMF 2013), Lecture notes in computer science* (Vol. 8195, pp. 115–130). Springer.
- Guerra, P. T., & Wassermann, R. (2010). Revision of CTL models. In A. Kuri-Morales & G. Simari (Eds.), *Advances in artificial intelligence—IBERAMIA 2010, LNCS* (Vol. 6433, pp. 153–162). Springer.
- Guerra, P. T., & Wassermann, R. (2017). On the uncomputability of partial meet contraction for linear-time temporal logic. In *Brazilian logic meeting—Book of abstracts*. Extended version accepted for publication in the South American Journal of Logic, 2019.
- Guerra, P. T., Wassermann, R. (2018). Two AGM-style characterizations of model repair. In *Proceedings of the 16th international conference on principles of knowledge representation and reasoning (KR 2018)* (pp. 645–646).
- Hansson, S. O. (1999). *A textbook of belief dynamics. Theory change and database updating*. Kluwer Academic.
- Hansson, S. O., & Wassermann, R. (2002). Local change. *Studia Logica*, 70(1), 49–76.
- Herzig, A., & Rifi, O. (1999). Propositional belief base update and minimal change. *Artificial Intelligence*, 115(1), 107–138. [https://doi.org/10.1016/S0004-3702\(99\)00072-7](https://doi.org/10.1016/S0004-3702(99)00072-7)
- Huth, M. (2002). Model checking modal transition systems using Kripke structures. In A. Cortesi (Ed.), *Verification, model checking, and abstract interpretation* (pp. 302–316). Springer.
- Huth, M., Jagadeesan, R., & Schmidt, D. (2001). Modal transition systems: A foundation for three-valued program analysis. In D. Sands (Ed.), *Programming languages and systems. Lecture notes in computer science* (Vol. 2028, pp. 155–169). Springer. https://doi.org/10.1007/3-540-45309-1_11
- Katsuno, H., & Mendelson, A. O. (1991). On the difference between updating a knowledge base and revising it. In *Proceedings of the second international conference on principles of knowledge representation and reasoning (KR'91)* (pp. 387–395). Morgan Kaufmann.
- Kozen, D. (1983). Results on the propositional μ -calculus. *Theoretical Computer Science*, 27(3), 333–354. [https://doi.org/10.1016/0304-3975\(82\)90125-6](https://doi.org/10.1016/0304-3975(82)90125-6)
- Larsen, K. G. (1990). Modal specifications. In J. Sifakis (Ed.), *Automatic verification methods for finite state systems. Lecture notes in computer science* (Vol. 407, pp. 232–246). Springer.
- Larsen, K. G., & Thomsen, B. (1988). A modal process logic. In *Proceedings of the third annual symposium on logic in computer science, 1988. LICS '88* (pp. 203–210). <https://doi.org/10.1109/LICS.1988.5119>
- Pnueli, A. (1977). The temporal logic of programs. In *Proceedings of the 18th annual symposium on foundations of computer science, SFCS'77* (pp. 46–57). IEEE Computer Society. <https://doi.org/10.1109/SFCS.1977.32>
- Queille, J. P., & Sifakis, J. (1982). Specification and verification of concurrent systems in CESAR. In *Proceedings of the 5th colloquium on international symposium on programming* (pp. 337–351). Springer. <http://dl.acm.org/citation.cfm?id=647325.721668>

- Ribeiro, J., Nayak, A., & Wassermann, R. (2018). Towards belief contraction without compactness. In *Knowledge representation and reasoning conference (KR)* (pp. 287–296).
- Ribeiro, J. S., & Andrade, A. (2015). A 3-valued contraction model checking game: Deciding on the world of partial information. In *Formal methods and software engineering—Proceedings of the 17th international conference on formal engineering methods, ICFEM 2015* (pp. 84–99).
- Sousa, T. C., & Wassermann, R. (2007). Handling inconsistencies in CTL model-checking using belief revision. In *Proc. of the Brazilian symposium on formal methods*.
- Van Zee, M., Doder, D., Dastani, M., & Van Der Torre, L. (2015). AGM revision of beliefs about action and time. In *Proc. international conference on artificial intelligence* (pp. 3250–3256). AAAI Press. <http://dl.acm.org/citation.cfm?id=2832581.2832702>
- Zhang, Y., & Ding, Y. (2008). CTL model update for system modifications. *Journal of Artificial Intelligence Research*, 31(1), 113–155.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.