Belief Revision on Computation Tree Logic*

Paulo T. Guerra and Renata Wassermann
University of São Paulo
São Paulo, Brazil
{paulotgo,renata}@ime.usp.br

Abstract

Model checking is one of the most effective techniques in automated system verification. Although this technique can handle complex verifications, model checking tools usually do not give any suggestions on how to repair inconsistent system models. In this paper, we show that approaches developed to update models of Computation Tree Logic (CTL) cannot deal with all kinds of changes. We introduce the concept of CTL model revision: an approach based on belief revision to handle system inconsistency in a static context.

1 Introduction

Model checking [Clarke *et al.*, 1986] is a formal verification technique that can be used to automatically check whether a system model satisfies or not its formal specification (the set of properties that describe the system behavior). System properties are described in some temporal logic formalism.

The Computation Tree Logic (CTL) [Clarke *et al.*, 1986] is a modal logic where we represent and reason about the future as a tree-like structure. Due to its branching characteristic, CTL allows quantification over possible execution paths that a system can follow. This makes the CTL formalism a good way for representing the formal specification of systems in model checking.

When a model is inconsistent with some desired property, most model checking implementations return a path indicating where the problem lies (the counterexample path). However, the counterexample path alone may not be enough. The more complex a system is, the harder it becomes to fix it. In general, model checking tools do not have any suggestion mechanism to assist the model repair.

Theory change [Alchourron *et al.*, 1985; Katsuno and Mendelzon, 1991] has been recently used for retrieve consistence between system models and its formal specification. Zhang and Ding [2008] introduce the concept of *model update*, a technique based on *belief update* [Katsuno and Mendelzon, 1991] that can be combined with model checking towards the development of a tool for automatic system modification.

However, the model update approach is not enough, or at least not adequate, to address every kind of system corrections. Just as belief update is not adequate to handle beliefs about a world that has not changed, model update may lose information when it is applied to some repair motivated by properties that do not refer to new system behaviors (see [Guerra and Wassermann, 2010] for an example of how model update can lose information). We argue that another aspect of theory change should be used: the *belief revision theory* [Alchourron *et al.*, 1985; Gärdenfors, 1988].

This work aims at the development of a belief revision approach to handle system modifications in a static context. Our approach will suggest corrections in the original system specification following belief revision principles.

Since system properties in model checking are commonly represented by temporal formulas of the Computation Tree Logic (CTL), our main task is to analyze the correct way of applying belief revision to this logic, because classical belief revision theory makes some assumptions (see [Hansson and Wassermann, 2002]) that are not satisfied by CTL.

2 Progress to Date

We developed a formal CTL model revision framework based on Zhang and Ding [2008] model update approach. First, we used their proposal of primitive model operations as metrics to define a closeness ordering from the belief revision point of view. Models are compared by their structural similarity: difference of states, transitions, labeling function.

Let ψ be a CTL formula representing the system specification, ϕ a CTL formula incompatible with ψ, I and J two models of ϕ and $\leq_\psi$ our closeness ordering, we say that $I \leq_\psi J$ if and only if there is a model K of ψ such that for every model L of ψ, I is more similar to K than J and L are, i. e., the difference of states, transitions and labeling between I and K is less than between J and L.

Our ordering criteria embeds a global view of sets of models, which is necessary to correctly apply belief revision. When we select only those models of ϕ that are minimal according to $\leq_\psi$, we keep just the most probable corrections of the original inconsistent modeling. Based on this ordering, we can formally define the model revision operator $\circ_c$.

[*] The first author is supported by the State of São Paulo Research Foundation (FAPESP), grant number 2010/15392-3. This work is part of FAPESP project LogProb, grant number 2008/03995-5.

Definition 1 *Given two CTL formulas ψ and ϕ, we define that the result of revising ψ by ϕ, denoted by $\psi \circ_c \phi$, is a CTL formula whose models are defined as*

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

where $Mod(\mu)$ denotes the set of possible models of a formula μ and $Min_{\psi}(Mod(\phi))$ denotes the set of all minimal models of ϕ with respect to the ordering $\leq_{\psi}$.

We proved that $\circ_c$ obeys the rationality postulates for classical belief revision, according to Katsuno and Mendelzon [1989] reformulation (see [Guerra, 2010]).

We also started the implementation of our approach. We used as a basis the algorithm proposed by Zhang and Ding [2008]. Their algorithm performs a set of minimal modifications on a given CTL model to satisfies an arbitrary given CTL formula and produce a new CTL model that satisfies this formula and minimally differs from the original model.

Their algorithm makes recursive calls to parse the input formula, checks which portion is not satisfied in the model and then choose an appropriate correction. For example, the CTL formula EXp means that in a model there should exist a transition from the initial state to some state where p holds. If a model is inconsistent, the algorithm should choose between: (a) add a new state where p holds and then connect the initial state to it; or (b) add a transition to some existing state where p holds.

However, Zhang and Ding's functions cannot be directly applied to model revision. If a model satisfies the desired property and also differs minimally from some specification model, we say that this model belongs to the model update result, but we cannot say the same for model revision. Model revision needs a global view of modifications to ensure that this modification is one of the best choices. So, we developed a new set of modification functions to generate several satisfiable models for each input model[1].

Given a set of models that satisfies a system specification and a desired property, our CTL model revision algorithm generates a set of possibles modification model of each input model and than refine it, eliminating all non-minimal models according to our ordering criteria. The result is a set of models representing possibles minimal modifications, according to our model revision principles. Despite of some natural differences, our approach is strongly based on the functions of [Zhang and Ding, 2008].

After some effort following this line, we believe that this adaptation is not a promising way of implementing model revision. Some problems that do not occur in model update arise on model revision. Thus, our algorithm has currently two main issues: how to select a correct (and finite) subset of models of the system specification to compose the input; and how to generate the complete set of possible modifications for each model.

3 Plans for Remaining Work

There are several issues to be explored during the PhD. First we need to continue the semantic analysis of CTL model revision. In special, we need to complete the proof of the representation theorem for our model revision operator $\circ_c$. We have shown that $\circ_c$ obeys the rationality postulates, we need verify whether that every possible operator that obeys the postulates can be represented by our operator.

Another issue is to investigate CTL revision beyond the semantic level. We need to analyze how to do CTL belief revision when the beliefs are represented by a set of CTL formulas and not by a set of CTL models, since this formalism is widely used in the related literature.

We plan to formalize how the various classical approaches fail to be applicable to CTL. Our goal is to highlight the points similar (or different) between the classical belief revision approach and CTL belief revision .

We also intend to explore other ways of implementing CTL revision, leaving out the Zhang and Ding characteristic of performing minimal model change in each recursive step of the algorithm. We believe that investigating some CTL characteristics, as its fixed-point property, is a promising way to develop an effective CTL revision algorithm.

Finally, we plan to extend our results to other temporal logics with common characteristics, like Linear-time Temporal Logic or CTL*.

References

[Alchourron *et al.*, 1985] Carlos E. Alchourron, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *J. Symb. Logic*, 50(2):510–530, 1985.

[Clarke *et al.*, 1986] Edmund M. Clarke, E. Allen Emerson, and A. Prasad Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 8(2):244–263, 1986.

[Gärdenfors, 1988] Peter Gärdenfors. *Knowledge in flux*. MIT press, 1988.

[Guerra and Wassermann, 2010] Paulo T. Guerra and Renata Wassermann. Revision of CTL models. In *Advances in Artificial Intelligence – IBERAMIA 2010*, volume 6433 of *LNCS*, pages 153–162. Springer Berlin / Heidelberg, 2010.

[Guerra, 2010] Paulo T. Guerra. Revisão de modelos CTL. Master's thesis, Universidade de São Paulo, 2010.

[Hansson and Wassermann, 2002] Sven O. Hansson and Renata Wassermann. Local change. *Studia Logica*, 70(1):49–76, 2002.

[Katsuno and Mendelzon, 1989] Hirofumi Katsuno and Alberto O. Mendelzon. A unified view of propositional knowledge base updates. In *Proc. of IJCAI'89*, pages 1413–1419. Morgan Kaufmann, 1989.

[Katsuno and Mendelzon, 1991] Hirofumi Katsuno and Alberto O. Mendelzon. On the difference between updating a knowledge base and revising it. In *Proc. of KR*, pages 387–395. Morgan Kaufmann, 1991.

[Zhang and Ding, 2008] Yan Zhang and Yulin Ding. CTL model update for system modifications. *Journal of Artificial Intelligence Research*, 31(1):113–155, 2008.

[1] Our first choice was to compose Zhang and Ding's functions, but this may lead to some problems, like infinite recursive calls.