Model Checking for Hybrid Logic

Martin Lange

Abstract We consider the model checking problem for Hybrid Logic. Known algorithms so far are global in the sense that they compute, inductively, in every step the set of all worlds of a Kripke structure that satisfy a subformula of the input. Hence, they always exploit the entire structure. Local model checking tries to avoid this by only traversing necessary parts of the input in order to establish or refute the satisfaction relation between a given world and a formula. We present a framework for local model checking of Hybrid Logic based on games. We show that these games are simple reachability games for ordinary Hybrid Logic and weak Büchi games for Hybrid Logic with operators interpreted over the transitive closure of the accessibility relation of the underlying Kripke frame, and show how to solve these games thus solving the local model checking problem. Since the first-order part of Hybrid Logic is inherently hard to localise in model checking, we give examples, in the end, of how global model checkers can be optimised in certain special cases using well-established techniques like fixpoint approximations and divide-and-conquer algorithms.

Keywords Local model checking · Büchi games

1 Introduction

1.1 Hybrid Logic

There is a well-known opposition between Modal Logic (ML) on one hand and First-Order Logic (FOL) on the other. Modal Logic appeals because of its computational properties. Satisfiability in the modal logic K for example is just PSPACE-complete,
for some other modal logics it is even NP-complete only, etc. The model checking problem for ML is solvable in polynomial time. All this is connected to the fact that modal logics have a built-in aspect of locality. The ML-properties that a world of a Kripke structure has are composed from the properties of its immediate neighbourhood.

FOL is not restricted in this way since, in particular, quantification ranges over all elements of the underlying relational structure. Hence, a world of a Kripke structure can have an FOL-property because some other world that is not in its neighbourhood has another property. Not surprisingly, such operators lead to higher expressive power. For instance, while ML-properties are bisimulation-invariant, i.e. no formula of ML can distinguish bisimilar models, FOL-formulas are well capable of that. However, this increase in expressive power naturally leads to an increase in computational complexity: the model checking problem is PSPACE-complete and the satisfiability problem is non-elementary.

Hence, there is no general preference of one of these logics over the other. It is even easy to imagine that there are cases in which neither ML nor FOL is sufficient because of either lack of expressive power or lack of efficient decision procedures. For such cases one would naturally try to combine ML and FOL in a way that retains the good properties of both. One such approach leads to Hybrid Logic (HL) which incorporates into Modal Logic (or better: Temporal Logic) certain first-order features (Bull 1970; Passy and Tinchev 1991).

This logic has attracted a lot of interest in itself (Goranko 1996; Areces et al. 2000; ten Cate 2005; Areces and ten Cate 2006), studying in particular its proof theory and its model theory. On the other hand, Hybrid Logic also has found applications in many diverse areas, for example in artificial intelligence and knowledge representation because it is closely related to description logics (Areces 2000; Blackburn and Tzakova 1998); in computational linguistics because of its relation to feature logics (Blackburn 1993); as a logic for semi-structured data (Bidoit et al. 2004; Franceschet and de Rijke 2006); etc.

While the main focus in the research on decision procedures traditionally lay on the satisfiability and validity problem, the model checking problem for Hybrid Logic has lately also been found worthy of studying, in particular because of its close relationship to the querying and the constraints evaluation problem for XML data (Franceschet and de Rijke 2006). Franceschet and de Rijke classify the model checking problem for various fragments of Hybrid Logic w.r.t. their computational complexity and achieve various result between polynomial and exponential time. For the upper bounds they present explicit model checking algorithms and analyse their space and time consumption. These algorithms are straight-forward extensions of known algorithms for temporal or propositional dynamic logics. This is certainly sufficient for the complexity classification, but it is fair to ask whether there are “better” algorithms. In program verification for example, where model checking techniques have successfully been used for 25 years by now, these naive approaches would not be considered state-of-the-art. Franceschet and de Rijke also identify the verification of mobile reactive systems as an application for Hybrid Logic model checking (Franceschet and de Rijke 2003). It is fair to say that such applications require optimised algorithms to stand a chance of being of practical use. The aim of this paper is to provide a framework