On Timed Simulation Relations for Hybrid Systems and Compositionality

Goran Frehse

VERIMAG
goran.frehse@verimag.fr
http://www-verimag.imag.fr/~frehse

Abstract. Timed and weak timed simulation relations are often used to show that operations on hybrid systems result in equivalent behavior or in conservative overapproximations. Given that systems are frequently designed and verified in a modular approach, it is desirable that this relationship is compositional, which is not the case for hybrid systems in general. We identify subclasses of linear hybrid automata that are compositional with respect to timed, respectively weak timed simulation.

1 Introduction

Hybrid automata are notoriously hard to analyze, so they are often overapproximated with hybrid automata of simpler dynamics, see [1,2,3] and references therein. The proofs used to show that the constructed automata are indeed conservative frequently involve timed simulation, or a weak variant that allows unobservable transitions as long as they don't change the variables. The analysis is usually challenging even for the abstracted system, and increases exponentially with the number of components and variables. Compositional reasoning is known as a valuable tool to counter this problem. However, timed and weak timed simulation are not compositional for hybrid automata with arbitrary dynamics. Consequently, a successful compositional analysis of the abstracted system does not imply safety of the original system when timed simulation was used in proving conservativeness.

In this paper we identify classes of hybrid automata that are compositional with respect to timed, respectively weak timed simulation. If such a class is used to overapproximate a system, conservativeness is consequently guaranteed and compositional reasoning valid. These results are directly applicable to strengthen the overapproximation operators in [1,2,3] with respect to compositionality.

Related Work. We use the hybrid automata in [4] with minor modifications. We define a subset of the controlled variables as output variables, specify the activities via their derivatives, include a set of initial states, and consider the same controlled variables in all locations. The hybrid automata in [4] are known to be compositional for trace inclusion [4], see [5] for applications. The controlled variables are needed to prove compositionality. We add output variables to hide internal (non-output) behavior, i.e., so we can compare automata whose output

E. Asarin and P. Bouyer (Eds.): FORMATS 2006, LNCS 4202, pp. 200–214, 2006.

variables behave identically while the internal workings may be different. More sophisticated hybrid input/output-automata (HIOA) are proposed and studied in detail in [6]. HIOA impose input-enabledness that we do not require, so the hybrid automata in this paper are equivalent to the pre-HIOA of [6]. The stricter I/O-distinction in [6] may be used to ensure some liveness properties; we only consider safety. We use a compositional type of simulation from [6], which we call *trace simulation* to set it apart from timed simulation.

Timed simulation is usually defined using labeled transition system (LTS) semantics [7]. Our definition is directly based on runs of hybrid automata, but is otherwise equivalent. In earlier work, we proposed semantic criteria for compositionality of timed simulation without giving an interpretation on the hybrid automaton level, and not for weak timed simulation [8]. The framework used in this paper presents a substantial improvement and simplification, and our previous results on compositionality and assume/guarantee-reasoning from [9,10] can be transferred to it. For the sake of brevity, we provide mostly proof sketches. Detailed proofs for most of the results (except those involving overlap-closure) can be found in [10].

In the following section, we present our hybrid automata and their semantics. In Sect. 3 we define trace and timed simulation, as well as their weak counterparts. In Sect. 4 we identify compositional subclasses for these types of simulation. Finally, we draw some conclusions in Sect. 5.

2 Hybrid Automata

We use a standard hybrid automaton model and parallel composition operator from [4], to which we add a subset of output variables. A variable is either an *uncontrolled variable* (also called input), and can therefore change arbitrarily at any time, or *controlled*. In parallel composition, controlled variables can not be changed independently by other automata in the composition. These elements are essential to compositionality [11]. A subset of the controlled variables are *output* variables, which, together with the uncontrolled variables, define the externally visible behavior of the automaton. Note that the uncontrolled variables may be restricted in their derivatives, and can only change arbitrarily inside the invariant. This allows us to model causal and noncausal coupling between variables, which is useful, e.g., to model conservation laws.

Preliminaries. Given a set $X = \{x_1, \ldots, x_n\}$ of variables, a *valuation* is a function $v : X \to \mathbb{R}$. We use $\dot{X}$ to denote the set $\{\dot{x}_1, \ldots, \dot{x}_n\}$ of dotted variables, and X' to denote the set $\{x'_1, \ldots, x'_n\}$ of primed variables. Let $V(X)$ denote the set of valuations over X. The *projection* of v to variables $\bar{X} \subseteq X$ is $v{\downarrow_{\bar{X}}} = \{x \to v(x) | x \in \bar{X}\}$. The *embedding* of a set $U \subseteq V(X)$ into variables $\bar{X} \supseteq X$ is the largest subset of $V(\bar{X})$ whose projection is in U, written as $U|^{\bar{X}}$. When a valuation u over X and a valuation v over $\bar{X}$ *agree*, i.e., $u{\downarrow_{X \cap \bar{X}}} = v{\downarrow_{X \cap \bar{X}}}$, we use $u \sqcup v$ to denote the valuation w defined by $w{\downarrow_X} = u$ and $w{\downarrow_{\bar{X}}} = v$. Arithmetic operations on valuations are defined in the straightforward way. An *activity* over X is a function $f : \mathbb{R}^{\geq 0} \to V(X)$. Let $Acts(X)$ denote the set of activities