Path-oriented bounded reachability analysis of composed linear hybrid systems

Lei Bu · Xuandong Li

Abstract The existing techniques for reachability analysis of linear hybrid systems do not scale well to the problem size of practical interest. The performance of existing techniques is even worse for reachability analysis of a composition of several linear hybrid automata. In this paper, we present an efficient path-oriented approach to bounded reachability analysis of composed systems modeled by linear hybrid automata with synchronization events. It is suitable for analyzing systems with many components by selecting critical paths, while this task was quite insurmountable before because of the state explosion problem. This group of paths will be transformed to a group of linear constraints, which can be solved by a linear programming solver efficiently. This approach of symbolic execution of paths allows design engineers to check important paths, and accordingly increase the faith in the correctness of the system. This approach is implemented into a prototype tool Bounded reAchability CHecker (BACH). The experimental data show that both the path length and the number of participant automata in a system checked using BACH can scale up greatly to satisfy practical requirements.

Keywords Hybrid systems · Bounded reachability analysis · Linear hybrid automata · Linear programming

1 Introduction

The model checking problem for hybrid systems is very difficult. Even for a relatively simple class of hybrid systems, linear hybrid automata (denoted as LHA), the reachability analysis problem is undecidable [1–4]. Several model checking tools have been developed for reachability analysis of LHA, but they do not scale well to the size of practical problems. The state-of-the-art tool HYTECH [5] and its improvement PHAVer [6] need expensive polyhedra computation, which greatly restricts the solvable problem size.

In recent years, Bounded Model Checking (denoted as BMC) [7] has been presented as a technique alternative to BDD-based symbolic model checking, whose basic idea is to encode the next-state relation of a system as a propositional formula, unroll this formula to some integer k, and search for a counterexample in the model executions whose length is bounded by k. The BMC problems can be solved by Boolean Satisfiability (denoted as SAT) methods, which have achieved tremendous progress in recent years, as summarized in [8].

As extensions to BMC, there are several related works [9–11] to check linear hybrid systems. In these techniques, the model checking problems are reduced to the satisfiability problem of a boolean combination of propositional variables and linear mathematical constraints. Based on these techniques, several tools were developed, such as MathSAT [11] and HySAT [9], and all of them are based on a SAT-solver that calls on demand solver for conjunctions of the domin-specific constraints [12]. But the experiment results show that it is difficult to apply those tools to analysis problems of practical size. The performance of existing techniques is even worse for reachability analysis of a composition of several linear hybrid automata.
As the existing techniques do not perform well concerning analysis problems of practical size, in this paper, we propose a complementary approach to develop an efficient path-oriented technique for bounded reachability analysis of LHA compositions. This technique checks a group of paths at a time, one path for each LHA, where both the path length and the number of participant automata checked can scale up greatly to satisfy practical requirements. This approach of symbolic execution of paths can be used by design engineers to check critical paths, and thereby increase the faith in the system correctness.

For a linear hybrid system consisting of several components (LHA), with our approach users can assign a specific path to each LHA, respectively, and all of the paths are transformed into a group of linear constraints automatically. Then, a few of constraints about system integration according to the synchronization events in each path will be added to ensure that the components cooperate correctly. It follows that the reachability problem along those specific paths can be reduced to a linear program. We shall use a simple example to illustrate this idea below.

Most traditional verification methods of hybrid systems consisting of several components require the composition of the set of automata to a unique global automaton, which leads to the critical problem of state explosion. For example, Fig. 1a gives a simple system consisting of three subsystems: S, T, and K which synchronize with each other by events b, e, and f. Even if these three subsystems are all very simple, the state space of the resulting automata is still quite large as we can see from Fig. 1b. While using our path-oriented approach, as each of those three subsystems only has one path, we simply select all of them for analysis, as shown in Fig. 1c. First, for each of these three paths we generate a group of linear constraints that represents all the timed runs corresponding to the path. For example, for the path \langle t_1, \delta_1 \rangle \rightarrow \langle t_2, \delta_2 \rangle \rightarrow \langle t_3, \delta_3 \rangle \rightarrow \langle t_4, \delta_4 \rangle \rightarrow \langle t_5, \delta_5 \rangle of the system T, we use \langle t_i, \delta_i \rangle to indicate that the system has stayed in location t_i for time delay \delta_i (nonnegative variable). Any timed run corresponding to this path can be represented by

\[
\langle t_1, \delta_1 \rangle \rightarrow \langle t_2, \delta_2 \rangle \rightarrow \langle t_3, \delta_3 \rangle \rightarrow \langle t_4, \delta_4 \rangle \rightarrow \langle t_5, \delta_5 \rangle
\]

where \delta_1, \delta_2, \delta_3, \delta_4, \delta_5 must satisfy all the time constraints enforced by the system, which forms a group of linear constraints. Second, several constraints will be added to ensure that these three components cooperate accurately according to the synchronization events, which are illustrated by the dashed lines and SYN(event) in Fig. 1c. Because such a state space representation by linear constraints is equivalent to the Cartesian product representation shown in Fig. 1b in terms of reachability analysis, the reachability analysis problem along these three paths can be transformed into a linear programming problem, which can be solved efficiently.

Therefore, the path-oriented approach presented in this paper is to check if an LHA composition satisfies a reachability specification along a given group of its component paths. This approach has been implemented into a prototype tool Bounded reAchability CHecker (BACH). The experimental data show that both the path length and the number of participant components in a system checked using BACH can scale up greatly to satisfy practical requirements.

The rest of the paper is organized as follows. In the next section, we define the class of linear hybrid automata and the compositions of linear hybrid automata considered in this paper. Section 3 presents the linear programming based solution for the path-oriented reachability analysis of LHA compositions. Section 4 describes several case studies to show the ability of BACH, and also gives a comparison with other tools. Finally the conclusion is made in Sect. 5.