Journal of Symbolic Computation ••• (••••) •••–•••

Contents lists available at ScienceDirect

Journal of Symbolic Computation

www.elsevier.com/locate/jsc

Barrier certificates revisited ☆

Liyun Dai [a], Ting Gan [a], Bican Xia [a], Naijun Zhan [b]

[a] *LMAM & School of Mathematical Sciences, Peking University, China*
[b] *State Key Lab. of Computer Science, Institute of Software, Chinese Academy of Sciences, China*

ARTICLE INFO

Article history:
Received 23 May 2015
Accepted 28 July 2015
Available online xxxx

Keywords:
Hybrid system
Barrier certificate
Formal verification
Invariant
Nonlinear system
Semi-definite programming
Sum of squares

ABSTRACT

A barrier certificate can separate the state space of a considered hybrid system (HS) into safe and unsafe parts according to the safety property to be verified. Therefore this notion has been widely used in the verification of HSs. A stronger condition on barrier certificates (BCs) means that fewer BCs can be synthesized, as the expressiveness of synthesized BCs is weaker. On the other hand, synthesizing more expressive BCs normally means higher complexity. Kong et al. (2013a) investigated how to relax the condition of BCs while still keeping their convexity so that one can synthesize more expressive BCs efficiently using semi-definite programming (SDP). In this paper, we first discuss how to relax the condition of BCs in a general way, while still keeping their convexity. Thus, one can utilize different weaker conditions flexibly to synthesize different kinds of BCs with more expressiveness efficiently using SDP, which gives more opportunities to verify the considered system. We also show how to combine two functions together to form a combined BC in order to prove a safety property under consideration, whereas neither of them may be a BC separately. In fact, the notion of combined BCs is strictly more expressive than that of BCs, so it further brings more chances to verify a considered system. Another contribution of this paper is to

☆ This work has been supported partly by "973 Program" under grant No. 2014CB340701, by Natural Science Foundation of China under grants 91118007, 91418204, 11290141 and 11271034, by the CAS-SAFEA International Partnership Program for Creative Research Teams, and by CDZ project CAP (GZ 1023).

E-mail addresses: dailiyun@pku.edu.cn (L. Dai), gant@pku.edu.cn (T. Gan), xbc@math.pku.edu.cn (B. Xia), znj@ios.ac.cn (N. Zhan).

URL: http://lcs.ios.ac.cn/~znj/ (N. Zhan).

http://dx.doi.org/10.1016/j.jsc.2016.07.010
0747-7171/© 2016 Elsevier Ltd. All rights reserved.

Please cite this article in press as: Dai, L., et al. Barrier certificates revisited. J. Symb. Comput. (2016), http://dx.doi.org/10.1016/j.jsc.2016.07.010

investigate how to avoid the unsoundness of SDP based approaches caused by numerical error through symbolic checking.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Embedded systems (ESs) make use of computer units to control physical devices so that the behavior of the controlled devices meets expected requirements. They have become ubiquitous in our daily life, e.g. automotive, aerospace, consumer electronics, communications, medical, manufacturing and so on. ESs are used to carry out highly complex and often critical functions, e.g. ESs are used to monitor and control industrial plants, complex transportation equipment, communication infrastructure, etc. The development process of ESs is widely recognized as a highly complex and challenging task. A thorough validation and verification activity is necessary to enhance the quality of the ESs and, in particular, to fulfill the quality criteria mandated by the relevant standards. Hybrid systems (HSs) are mathematical models with precise mathematical semantics for ESs, wherein continuous physical dynamics are combined with discrete transitions. Based on HSs, rigorous analysis and verification of ESs become feasible, so that errors can be detected and corrected in the very early stage of the design of ESs.

In the past, analysis and verification of HSs were mainly done through directly computing reachable sets, either by model-checking (e.g., Alur et al., 1995; Puri and Varaiya, 1994; Henzinger and Ho, 1995) or by decision procedures (e.g., Lafferriere et al., 2001). The basic idea is to partition the state space of a considered system into finitely many equivalent classes, or represent it by finitely many computable sets according to the solutions of the ODE of the system. Since there is only a very small class of ODEs with closed form solutions, the scalability of these approaches is very restricted, only applicable to very specific linear HSs. Recently, there are lots of work based on abstraction and numeric approximation to scale up these approaches, e.g., Eggers et al. (2012), Chen et al. (2013), Gao et al. (2013), Ratschan and She (2007). In principle, these approaches are quite successful in the falsification of a considered HS by debugging bugs using bounded model-checking, but have difficulty in proving the error-avoidance of the system. As an alternative, deductive methods have been recently proposed and successfully applied in practice (Platzer and Clarke 2008, 2009; Liu et al., 2010). The most challenging part of a deductive method is how to discover invariants, which hold at all reachable states of the system. For technical reasons, people only consider how to synthesize inductive invariants, which are preserved by all discrete and continuous transitions. In general, a safety property itself is an invariant, but may not be an inductive invariant. Obviously, an inductive invariant is an approximation of the reachable set, which may be discovered according to the ODE, rather than its solutions. The basic idea is as follows: first, predefine a property template (linear or non-linear, depending on the property to be verified); then, encode the conditions of a property to be inductive (discretely and/or continuously) into some constraints on state variables and parameters; finally, find out solutions to the constraints. So, how to define inductive conditions and the power of constraint solving are essential to these approaches.

Many approaches have been proposed following the line discussed above. E.g., in Jirstrand (1998), Rodríguez-Carbonell and Tiwari (2005), the authors independently proposed different approaches for constructing inductive invariants for linear HSs; Sankaranarayanan et al. presented a computational method to automatically generate algebraic invariants for algebraic HSs in Sankaranarayanan (2010), Sankaranarayanan et al. (2004), based on the theory of pseudo-ideals over polynomial rings and quantifier elimination; Prajna and Jadbabaie (2004) and Prajna et al. (2007) provided a new notion of inductive invariants called *barrier certificates* (BC) for verifying the safety of semi-algebraic HSs in the stochastic setting using the technique of sum-of-squares (**SOS**); Platzer and Clarke (2008) extended the idea of *BCs* by considering Boolean combinations of multiple polynomial inequalities; Gulwani and Tiwari (2008) and Taly and Tiwari (2009) investigated how to generate inductive invariants with more expressiveness for semi-algebraic HSs through relaxing the inductive conditions by considering

Please cite this article in press as: Dai, L., et al. Barrier certificates revisited. J. Symb. Comput. (2016), http://dx.doi.org/10.1016/j.jsc.2016.07.010

Download English Version:

https://daneshyari.com/en/article/4945990

Download Persian Version:

https://daneshyari.com/article/4945990

Daneshyari.com