Static Analysis in the Continuously Changing World

Sriram Sankaranarayanan*

University of Colorado, Boulder, CO.
firstname.lastname@colorado.edu

Abstract. In this talk, we examine static analysis techniques for continuous-time dynamical systems. Continuous time systems arise in many domains including engineered control systems, physical and biological systems. They are increasingly of interest to the static analysis community, due to the focus on *hybrid (cyber-physical) systems* that capture discrete programs interacting with a continuous external environment. We examine two types of properties that are typically verified: reachability and stability, and explore parallels between commonly used static analysis approaches and a variety of approaches to prove/disprove reachability and stability properties.

1 Introduction

Static analysis refers to a broad class of techniques that reason about the correctness of systems in the presence of uncertainties [10]. The key defining characteristics of static analysis techniques include (a) reasoning collectively about a large, often infinite set of system behaviors using *abstract domains* to represent sets of states, and (b) soundness guarantees on the results of the analysis. Static analysis has witnessed a creative explosion of techniques that focus on reasoning about programs. Abstract interpretation has been successful in providing a convenient common framework for designing, implementing and comparing various static analysis techniques [4].

In this talk, we examine parallels between the world of discrete-time computer programs and continuous-time systems defined by Ordinary Differential Equations (ODEs). The mathematical theory of differential equations provides us a framework for reasoning about these systems [9]. Continuous-time systems arise in a wide variety of engineering disciplines (control systems), physics and biology. The study of continuous systems in the formal verification community has a long history due to the intense interest in hybrid dynamical systems that model discrete programs interacting with a continuous external environment [17,8]. We explore two classes of techniques for the static analysis of continuous time and hybrid systems: (a) *flowpipe construction* approaches that use repeated forward propagation over time, and (b) automatic synthesis of *positive invariants* and *Lyapunov functions*.

Flowpipe construction techniques characterize the behavior of continuous-time and hybrid systems in the presence of uncertainties due to the initial state and input signals. Flowpipe construction techniques compute conservative approximations of the time trajectories of ODEs using numerical domains such as intervals, octagons, convex

* The research presented was performed in collaboration with Ashish Tiwari, Aditya Zutshi, Erika Ábraham and Xin Chen. We gratefully acknowledge the support of the US National Science Foundation (NSF) under award numbers CNS-0953941 and CPS-1035845.

F. Logozzo and M. Fähndrich (Eds.): SAS 2013, LNCS 7935, pp. 4–5, 2013.

polyhedra and Taylor models. We examine the capabilities of flowpipe construction tools such as HyTech [7], Checkmate [3], D/Dt [1], Phaver [5], SpaceEx [6] and Flow* [2].

Another class of deductive techniques derive proofs of unreachability in the form positive invariants and stability proofs using Lyapunov functions. We examine proof rules for for invariance and stability of ODEs, and the use of these rules to synthesize invariants and Lyapunov functions [16,14,12,11]. Tools such as KeYmaera support automatic invariant synthesis [13], while the SOSTools package supports the automatic synthesis of Lyapunov functions [15]. We examine some of the successes and existing shortcomings of these approaches in our talk.

References

1. Asarin, E., Dang, T., Maler, O.: The $\mathbf{d/dt}$ tool for verification of hybrid systems. In: Brinksma, E., Larsen, K.G. (eds.) CAV 2002. LNCS, vol. 2404, pp. 365–370. Springer, Heidelberg (2002)
2. Chen, X., Ábrahám, E., Sankaranarayanan, S.: Taylor model flowpipe construction for non-linear hybrid systems. In: Proc. RTSS 2012, pp. 183–192. IEEE (2012)
3. Chutinan, A., Krogh, B.: Computing polyhedral approximations to flow pipes for dynamic systems. In: Proceedings of IEEE CDC. IEEE Press (1998)
4. Cousot, P., Cousot, R.: Abstract Interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. ACM Principles of Programming Languages, 238–252 (1977)
5. Frehse, G.: PHAVer: Algorithmic verification of hybrid systems past hyTech. In: Morari, M., Thiele, L. (eds.) HSCC 2005. LNCS, vol. 3414, pp. 258–273. Springer, Heidelberg (2005)
6. Frehse, G., Le Guernic, C., Donzé, A., Cotton, S., Ray, R., Lebeltel, O., Ripado, R., Girard, A., Dang, T., Maler, O.: SpaceEx: Scalable verification of hybrid systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 379–395. Springer, Heidelberg (2011)
7. Henzinger, T.A., Ho, P.: HYTECH: The Cornell hybrid technology tool. In: Antsaklis, P.J., Kohn, W., Nerode, A., Sastry, S.S. (eds.) HS 1994. LNCS, vol. 999, pp. 265–293. Springer, Heidelberg (1995)
8. Lunze, J., Lamnabhi-Lagarrigue, F. (eds.): Handbook of Hybrid Systems Control: Theory, Tools and Applications. Cambridge University Press (2009)
9. Meiss, J.D.: Differential Dynamical Systems. SIAM Publishers (2007)
10. Nielson, F., Nielson, H.R., Hankin, C.: Principles of Program Analysis. Springer (1999)
11. Papachristodoulou, A., Prajna, S.: On the construction of lyapunov functions using the sum of squares decomposition. In: IEEE CDC, pp. 3482–3487. IEEE Press (2002)
12. Platzer, A.: Differential dynamic logic for hybrid systems. J. Autom. Reasoning 41(2), 143–189 (2008)
13. Platzer, A., Quesel, J.-D.: KeYmaera: A hybrid theorem prover for hybrid systems (System description). In: Armando, A., Baumgartner, P., Dowek, G. (eds.) IJCAR 2008. LNCS (LNAI), vol. 5195, pp. 171–178. Springer, Heidelberg (2008)
14. Prajna, S., Jadbabaie, A.: Safety verification of hybrid systems using barrier certificates. In: Alur, R., Pappas, G.J. (eds.) HSCC 2004. LNCS, vol. 2993, pp. 477–492. Springer, Heidelberg (2004)
15. Prajna, S., Papachristodoulou, A., Seiler, P., Parrilo, P.A.: SOSTOOLS: Sum of squares optimization toolbox for MATLAB (2004)
16. Sankaranarayanan, S., Sipma, H., Manna, Z.: Constructing invariants for hybrid systems. Formal Methods in System Design 32(1), 25–55 (2008)
17. Tabuada, P.: Verification and Control of Hybrid Systems: A Symbolic Approach. Springer (2009)