Assertion-based Analysis of Hybrid Systems with PVS

Extended Abstract

31. December 2000

Erika Ábrahám-Mumm[1], Ulrich Hannemann[2], and Martin Steffen[1]

[1] Institut für Informatik und praktische Mathematik
Christian-Albrechts-Universität
Preußerstraße 1–9, D-24105 Kiel, Deutschland
[2] Computing Science Department, University of Nijmegen
P.O. Box 9010, NL - 6500 GL Nijmegen, The Netherlands

Abstract. Hybrid automata are a well-established mathematical model for discrete systems acting in a continuous environment. We present assertion-based proof methods for hybrid systems for inductive assertional proofs. The model and the proof-methods are rigorously formalized within the PVS theorem-prover. We validate the applicability of the approach on a number of examples.

Hybrid systems

Combining discrete state-machines with continuous behavior, hybrid systems [1] have been successfully used to model a large number of applications in areas such as real-time software, embedded systems, and others. Basically, its a state-based formalism augmented by real-valued variables that may continuously evolve over time. The discrete behavior is given as a label-transition system, typically in guarded-command notation, allowing shared-variable communication and synchronization over transition labels. The continuous behavior —the activities, as it is called— is typically specified per control-state by differential (in–)equations. Based on [1], the full paper presents a thorough cover of the model condensed in the following definition: A *hybrid system* is a tuple $(Loc, Var, Con, Ini, Lab, Edg, Act, Inv)$, where Loc is a finite non-empty set of *locations* and Var a finite, non-empty set of *variables.* The function $Con \in Loc \to 2^{Var}$ defines the *control variables* in each state, the set $Ini \subseteq \Sigma = Loc \times V$ the *initial states.* The *transitions* are given by $Edg \subseteq Loc \times Lab \times (2^V, V \to 2^V) \times Loc$, where Lab denotes a finite set of *labels* containing the stutter label τ. For all $l \in Loc$ there is a stutter transition $(l, \tau, (\phi, f), l) \in Edg$ such that $\phi = V$ and $f(\nu) = \{\nu' \mid \nu|_{Con(l)} = \nu'|_{Con(l)}\}$. The *activities* are given by $Act : Loc \to 2^{\mathcal{F}}$ such that $Act(l)$ is time-invariant for each location $l \in Loc$. The function $Inv : Loc \to 2^V$ specifies the *invariants.* The parallel composition of two hybrid systems H_1 and H_2 is given by a standard product construction and written as $H_1 \times H_2$.

A deductive approach

By its continuous part, hybrid automata are a priori *infinite* state systems. Moreover, their computational properties are undecidable in the general model (this is already true for timed-automata, an important subclass). Depending on various restrictions on the form of the invariants, the guards, the activities etc., a score of variants and simplifications of the general model have been investigated, especially to obtain decidable and automatically checkable subclasses of the general definition (cf. for instance [1] [2] [6] [5] [8]). The main line of research concentrated on model checking of finite abstractions of restricted subclasses of the general model. Besides the drawback of limited expressive power, fully-automatic approaches suffer from the usual state-space explosion problem, when dealing with the parallel composition of subsystems.

Hence in our work, we pursue an alternative route, using *deductive methods* and falling back upon a general-purpose theorem prover. To assure rigorous formal reasoning, we employ the interactive theorem prover PVS [7], based on higher-order logic, extensive libraries of data-structures and theories, powerful strategies to assist in routine verification tasks, and modularization facilities.

A classical approach for the verification state-based programs are *inductive assertions:* to prove the satisfaction of a property for all reachable states, it suffices to prove a weaker, inductive property, i.e., to prove the initial satisfaction and preservation under computational steps. To cope with the verification of parallel systems, it is advantageous to exploit the system's parallel structure, i.e., to use compositional proof techniques (cf. for instance [3] for an extensive monograph on the topic). In the paper we develop a compositional proof method to deal with the parallel composition of hybrid systems. The methods cover shared variable communication, label-synchronization, and especially the common continuous activities in the parallel composition of hybrid automata. The corresponding sound and complete proof rule is given below. Thus to prove a property φ to hold for all reachable states $Reach(H)$ of a hybrid system $H = H_1 \times H_2$ involves finding inductive assertion networks Q_i for the two components of H, where additional auxiliary variables may be used, and establishing a combined assertion network $Q_1' \times Q_2'$ that implies the desired property (the primed versions H_i' denote the systems augmented by auxiliary variables, and the relation $\geq$ captures sound augmentation).

$$\frac{\begin{array}{c} H_1' \times H_2' \geq H_1 \times H_2 \qquad Q_1' \; inductive \; for \; H_1' \quad Q_2' \; inductive \; for \; H_2' \\ \forall (l, \nu') : \Sigma_{H'}. Q_1' \times Q_2'(l, \nu') \rightarrow \varphi(l, \nu'|_{Var}) \end{array}}{Reach(H) \rightarrow \varphi} \; \textsc{Comp}$$

Machine-assisted verification

Besides hybrid systems and their parallel composition, we formalized the operational step semantics and a number of proof-rules similar to rule COMP of above within PVS, and applied the theory to the verification of a number of examples. Figure 1 sketches one of the examples we dealt with. It's an extension of the well-tried thermostat example, consisting of n separate heating systems H_i running in parallel and emptying a

common, refillable fuel tank. For the complete examples we have again to refer to the full paper.

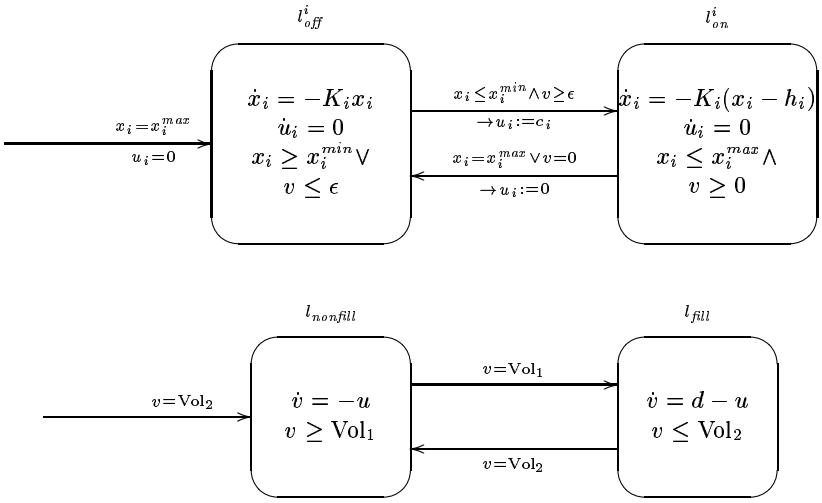

Fig. 1. Thermostats with common fuel tank: $H_1 \times \ldots H_n \times H_{fuel}$

The library of PVS-theories and the examples is available at ftp://ftp.informatik.uni-kiel.de/pub/kiel/softtech/eab/.

Acknowledgements The work was supported by the technology foundation STW, project EIF 3959, "Formal Design of Industrial Safety-Critical Systems" and further by the German Research Council (DFG) within the special program KONDISK (Analysis and Synthesis of Technical Systems with Continuous-discrete Dynamics) under grant LA 1012/5-1.

References

1. R. Alur, C. Courcoubetis, T. Henzinger, P. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138:3–34, 1995. A preliminary version appeared in the proceedings of 11th. International Conference on Analysis and Optimization of Systems: Discrete Event Systems (LNCI 199).
2. R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:252–235, 1994.
3. W.-P. de Roever, F. de Boer, U. Hannemann, J. Hooman, Y. Lakhnech, M. Poel, and J. Zwiers. *Concurrency Verification: Introduction to Compositional and Noncompositional Proof Methods*. Cambridge University Press, 2001. to appear.

4. R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors. *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*. Springer-Verlag, 1993.
5. T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata. In *27th Annual ACM Symposium on Theory of Computing*. ACM Press, 1995.
6. Y. Kesten, A. Pnueli, J. Sifakis, and S. Yovine. Integration graphs: a class of decidable hybrid systems. In Grossman et al. [4], pages 179–208.
7. S. Owre, J. M. Rushby, and N. Shankar. PVS: A prototype verification system. In D. Kapur, editor, *Automated Deduction (CADE-11)*, volume 607 of *Lecture Notes in Computer Science*, pages 748–752. Springer-Verlag, 1992.
8. O. Roux and V. Rusu. Uniformity for the decidability of hybrid automata. In Radha Cousot and D. A. Schmidt, editors, *Proceedings of SAS '96*, volume 1145 of *Lecture Notes in Computer Science*, pages 301–316. Springer-Verlag, 1996.