Logical Analysis of Hybrid Systems*

A Complete Answer to a Complexity Challenge

André Platzer

Computer Science Department, Carnegie Mellon University, Pittsburgh, USA
aplatzer@cs.cmu.edu

Abstract. Hybrid systems have a complete axiomatization in differential dynamic logic relative to continuous systems. They also have a complete axiomatization relative to discrete systems. Moreover, there is a constructive reduction of properties of hybrid systems to corresponding properties of continuous systems or to corresponding properties of discrete systems. We briefly summarize and discuss some of the implications of these results.

1 Overview

Hybrid systems [2,6,11] are dynamical systems that combine discrete and continuous dynamics. They are important for modeling embedded systems and cyber-physical systems. Hybrid systems are very natural models for many application scenarios, especially because each part of the system can be modeled in the most natural way. Discrete aspects of the system, e.g., discrete switching, computing, and control decisions can be modeled by discrete dynamics. Continuous aspects of the system, e.g., motion or continuous physical processes can be modeled by continuous dynamics. And hybrid systems simply combine either kind of dynamics with each other as one hybrid system in very flexible ways.

This flexibility makes hybrid systems very natural for system modeling. Even very complicated systems can be modeled as hybrid systems. Yet,reachability in hybrid systems is undecidable [11]. Even purely discrete systems are already undecidable, as witnessed by the halting problem. And even purely continuous systems are already undecidable [23, Theorem 2]. Are hybrid systems fundamentally more difficult than purely discrete or purely continuous systems? Or do they only add natural ways of expressing system models without causing additional complexities that are fundamentally more difficult to solve? Are hybrid systems more complex than discrete systems? Are they more complex than

* This material is based upon work supported by the National Science Foundation under NSF CAREER Award CNS-1054246, NSF EXPEDITION CNS-0926181, and under Grant Nos. CNS-1035800 and CNS-0931985, by the ONR award N00014-10-1-0188, by the Army Research Office under Award No. W911NF-09-1-0273, and by the German Research Council (DFG) as part of the Transregional Collaborative Research Center "Automatic Verification and Analysis of Complex Systems" (SFB/TR 14 AVACS).

M. Kutrib, N. Moreira, and R. Reis (Eds.): DCFS 2012, LNCS 7386, pp. 43–49, 2012.

continuous systems? And: are continuous systems more complex or are discrete systems more complex?

Since hybrid systems combine two independent sources of undecidability, discrete and continuous dynamics, the first intuition may be that hybrid systems should be fundamentally more difficult than either of the fragments. That turns out not to be the case, because there are complete proof-theoretical alignments of the discrete dynamics, continuous dynamics, and hybrid dynamics [23, 30]. In this paper, we explain a few of the consequences of these results.

For background on logic for hybrid systems, we refer to the literature [23, 26, 31]. Dynamic logic [39] has been developed and used very successfully for conventional discrete programs, both for theoretical [7–10, 12, 14, 15, 18, 20, 21, 42] and practical purposes [4, 9, 40]. We refer to other sources for more detail on dynamic logic for hybrid systems [22–26, 30]. Logic of hybrid systems has been used to obtain interesting theoretical results [22–30, 32], while, at the same time, enabling the practical verification of complex applications across different fields [3, 16, 17, 19, 24, 26, 35, 37, 41] and inspiring algorithmic logic-based verification approaches [24, 26, 33, 34, 36, 38, 41]. Extensions to logic for distributed hybrid systems [27, 29] and logic for stochastic hybrid systems [28] can be found elsewhere.

2 Differential Dynamic Logic

Differential dynamic logic $\mathsf{d}\mathcal{L}$ [22, 23, 30, 31] is a dynamic logic [39] for hybrid systems [6, 11]. To set the stage, we give a brief introduction to $\mathsf{d}\mathcal{L}$. We refer to previous work [23, 26, 30, 31] for more details.

Regular Hybrid Programs. We use (regular) *hybrid programs* (HP) [23] as hybrid system models. HPs form a Kleene algebra with tests [13]. The *atomic HPs* are instantaneous discrete jump *assignments* $x := \theta$, *tests* $?H$ of a first-order formula[1] H of real arithmetic, and *differential equation (systems)* $x' = \theta \,\&\, H$ for a continuous evolution restricted to the domain of evolution described by a first-order formula H. Compound HPs are generated from these atomic HPs by nondeterministic choice ($\cup$), sequential composition ($;$), and Kleene's nondeterministic repetition (*). We use polynomials with rational coefficients as terms. HPs are defined by the following grammar (α, β are HPs, x a variable, θ a term possibly containing x, and H a formula of first-order logic of real arithmetic):

$$\alpha, \beta \ ::= \ x := \theta \mid ?H \mid x' = \theta \,\&\, H \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^*$$

The first three cases are called atomic HPs, the last three compound HPs. These operations can define all hybrid systems [26]. We, e.g., write $x' = \theta$ for the unrestricted differential equation $x' = \theta \,\&\, \mathit{true}$. We allow differential equation systems and use vectorial notation. Vectorial assignments are definable from scalar assignments (and $;$).

[1] The test $?H$ means "if H then *skip* else *abort*".

A *state* ν is a mapping from variables to $\mathbb{R}$. Hence $\nu(x) \in \mathbb{R}$ is the value of variable x in state ν. The set of states is denoted $\mathcal{S}$. We denote the value of term θ in ν by $\nu[\![\theta]\!]$. Each HP α is interpreted semantically as a binary reachability relation $\rho(\alpha)$ over states, defined inductively by:

- $\rho(x := \theta) = \{(\nu, \omega) : \omega = \nu$ except that $\omega[\![x]\!] = \nu[\![\theta]\!]\}$
- $\rho(?H) = \{(\nu, \nu) : \nu \models H\}$
- $\rho(x' = \theta \,\&\, H) = \{(\varphi(0), \varphi(r)) : \varphi(t) \models x' = \theta$ and $\varphi(t) \models H$ for all $0 \leq t \leq r$ for a solution $\varphi : [0, r] \to \mathcal{S}$ of any duration $r\}$; i.e., with $\varphi(t)(x') \stackrel{\text{def}}{=} \frac{\mathsf{d}\varphi(\zeta)(x)}{\mathsf{d}\zeta}(t)$, φ solves the differential equation and satisfies H at all times [23]
- $\rho(\alpha \cup \beta) = \rho(\alpha) \cup \rho(\beta)$
- $\rho(\alpha; \beta) = \rho(\beta) \circ \rho(\alpha)$
- $\rho(\alpha^*) = \bigcup_{n \in \mathbb{N}} \rho(\alpha^n)$ with $\alpha^{n+1} \equiv \alpha^n; \alpha$ and $\alpha^0 \equiv ?\mathit{true}$.

We refer to our book [26] for a comprehensive background. We also refer to [23,26] for an elaboration how the case $r = 0$ (in which the only condition is $\varphi(0) \models H$) is captured by the above definition.

$\mathsf{d}\mathcal{L}$ Formulas. The *formulas of differential dynamic logic* ($\mathsf{d}\mathcal{L}$) are defined by the grammar (where ϕ, ψ are $\mathsf{d}\mathcal{L}$ formulas, θ_1, θ_2 terms, x a variable, α a HP):

$$\phi, \psi \ ::= \ \theta_1 \geq \theta_2 \mid \neg\phi \mid \phi \wedge \psi \mid \forall x\, \phi \mid [\alpha]\phi$$

The *satisfaction relation* $\nu \models \phi$ is as usual in first-order logic (of real arithmetic) with the addition that $\nu \models [\alpha]\phi$ iff $\omega \models \phi$ for all ω with $(\nu, \omega) \in \rho(\alpha)$. The operator $\langle\alpha\rangle$ dual to $[\alpha]$ is defined by $\langle\alpha\rangle\phi \equiv \neg[\alpha]\neg\phi$. Consequently, $\nu \models \langle\alpha\rangle\phi$ iff $\omega \models \phi$ for some ω with $(\nu, \omega) \in \rho(\alpha)$. Operators $=, >, \leq, <, \vee, \rightarrow, \leftrightarrow, \exists x$ can be defined as usual in first-order logic. A $\mathsf{d}\mathcal{L}$ formula ϕ is *valid*, written $\vDash \phi$, iff $\nu \models \phi$ for all states ν.

3 Complete Relations

Even though hybrid systems are very expressive, they nevertheless have a complete axiomatization in differential dynamic logic $\mathsf{d}\mathcal{L}$ [23, 30] relative to elementary properties of differential equations. The completeness notions are inspired by those of Cook [5] and Harel et al. [10], yet different, because the data logic of hybrid systems is perfectly decidable (first-order real arithmetic). Using the proof calculus of $\mathsf{d}\mathcal{L}$, the problem of proving properties of hybrid systems reduces to proving properties of continuous systems [23]. Furthermore, the proof calculus of $\mathsf{d}\mathcal{L}$ reduces the problem of proving properties of hybrid systems to proving properties of discrete systems [30].

FOD is the *first-order logic of differential equations*, i.e., first-order real arithmetic augmented with formulas expressing properties of differential equations, that is, $\mathsf{d}\mathcal{L}$ formulas of the form $[x' = \theta]F$ with a first-order formula F. We have shown that the $\mathsf{d}\mathcal{L}$ calculus is a sound and complete axiomatization relative to FOD.

Theorem 1 (Continuous relative completeness of dℒ [23, 30]). *The* dℒ *calculus is a* sound and complete axiomatization *of hybrid systems relative to its continuous fragment FOD, i.e., every valid* dℒ *formula can be derived from FOD tautologies:*

$$\vDash \phi \quad \textit{iff} \quad \mathrm{Taut}_{FOD} \vdash \phi$$

In particular, if we want to prove properties of hybrid systems, all we need to do is to, instead, prove properties of continuous systems, because the dℒ calculus completely handles all other steps in the proofs that deal with discrete or hybrid systems. Since the proof of Theorem 1 is constructive, there even is a complete constructive reduction of properties of hybrid systems to corresponding properties of continuous systems. The dℒ calculus can prove hybrid systems properties exactly as good as properties of the corresponding continuous systems can be verified. One important step in the proof of Theorem 1 shows that all required invariants and variants for repetitions can be expressed in the logic dℒ. Furthermore, the dℒ calculus defines a decision procedure for dℒ sentences (closed formulas) relative to an oracle for FOD.

This result implies that the continuous dynamics dominates the discrete dynamics for once the continuous dynamics is handled, all discrete and hybrid dynamics can be handled as well. This is reassuring, because we get the challenges of discrete dynamics solved for free (i.e., by the dℒ calculus) once we address continuous dynamics.

However, in a certain sense, continuous dynamics may appear to be more complicated to handle by discrete proof systems than continuous dynamics. After all, computers are discrete, so mechanized proofs on computers will ultimately need to understand continuous effects from a purely discrete perspective. If the continuous dynamics are not just subsuming discrete dynamics but were inherently more, then that could be understood as an indicator that hybrid systems verification is fundamentally impossible with discrete means. Of course, if this were the case, the argument would not even be quite so simple, because meta-proofs may still enable discrete finitary proof objects to entail infinite continuous object-properties. In fact, they do, because finite dℒ proof objects entail properties in uncountable continuous spaces.

Fortunately, we can settle worries about the insufficiency of discrete ways of understanding continuous phenomena once and for all by studying the proof-theoretical relationship between discrete and continuous dynamics. We have shown not only that the axiomatization of dℒ is complete relative to the continuous fragment, but that it is also complete relative to the discrete fragment [30]. The *discrete fragment* of dℒ is denoted by DL, i.e., the fragment without differential equations. It is, in fact, sufficient to restrict DL to the operators $:=, ^*$ and allow either ; or vector assignments.

Theorem 2 (Discrete relative completeness of dℒ [30]). *The* dℒ *calculus is a* sound and complete axiomatization *of hybrid systems relative to its discrete fragment DL, i.e., every valid* dℒ *formula can be derived from DL tautologies.*

$$\vDash \phi \quad \textit{iff} \quad \mathrm{Taut}_{DL} \vdash \phi$$

Thus, the $\mathsf{d}\mathcal{L}$ calculus can prove properties of hybrid systems exactly as good as properties of discrete systems can be proved. Again, the proof of Theorem 2 is constructive, entailing that there is a constructive way of reducing properties of hybrid systems to properties of discrete systems using the $\mathsf{d}\mathcal{L}$ calculus. Furthermore, the $\mathsf{d}\mathcal{L}$ calculus defines a decision procedure for $\mathsf{d}\mathcal{L}$ sentences relative to an oracle for DL.

As a corollary to Theorems 1 and 2, we can proof-theoretically and constructively equate

$$\text{hybrid} = \text{continuous} = \text{discrete}$$

Even though each kind of dynamics comes from fundamentally different principles, they all meet in terms of their proof problems being interreducible, even constructively. The complexity of the proof problem of hybrid systems, the complexity of the proof problem of continuous systems, and the complexity of the proof problem of discrete systems are, thus, equivalent.

Since the proof problems interreduce constructively, every technique that is successful for one kind of dynamics perfectly lifts to the other kind of dynamics through the $\mathsf{d}\mathcal{L}$ calculus. Induction is the primary technique for proving properties of discrete systems. Hence, by Theorem 2, there is a corresponding induction technique for continuous systems and for hybrid systems. And, indeed, *differential invariants* [25] are such an induction technique for differential equations that has been used very successfully for hybrid systems with more advanced differential equations [26, 33–35, 37]. Differential invariants had already been introduced in 2008 [25] before Theorem 2 was proved [30], but Theorem 2 implies that a differential invariant induction technique has to exist.

4 Conclusions and Future Work

We have summarized recent results about complete axiomatizations of hybrid systems relative to continuous systems and relative to discrete systems. These axiomatizations equate the proof problems for all three classes of systems and align the complexity of the their proof problems. Practical consequences of this result include differential invariants and the utility of discretization schemes, but many other consequences are just waiting to be discovered.

References

1. Proceedings of the 27th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2012, June 25–28, 2012, Dubrovnik, Croatia. IEEE Computer Society (2012)
2. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The algorithmic analysis of hybrid systems. Theor. Comput. Sci. 138(1), 3–34 (1995)
3. Aréchiga, N., Loos, S.M., Platzer, A., Krogh, B.H.: Using theorem provers to guarantee closed-loop system properties. In: Tilbury, D. (ed.) ACC (2012)

4. Beckert, B., Hähnle, R., Schmitt, P.H. (eds.): Verification of Object-Oriented Software: The KeY Approach, LNCS, vol. 4334. Springer (2007)
5. Cook, S.A.: Soundness and completeness of an axiom system for program verification. SIAM J. Comput. 7(1), 70–90 (1978)
6. Davoren, J.M., Nerode, A.: Logics for hybrid systems. IEEE 88(7), 985–1010 (July 2000)
7. Fischer, M.J., Ladner, R.E.: Propositional dynamic logic of regular programs. J. Comput. Syst. Sci. 18(2), 194–211 (1979)
8. Harel, D.: First-Order Dynamic Logic. Springer, New York (1979)
9. Harel, D., Kozen, D., Tiuryn, J.: Dynamic logic. MIT Press, Cambridge (2000)
10. Harel, D., Meyer, A.R., Pratt, V.R.: Computability and completeness in logics of programs (preliminary report). In: STOC. pp. 261–268. ACM (1977)
11. Henzinger, T.A.: The theory of hybrid automata. In: LICS. pp. 278–292. IEEE Computer Society, Los Alamitos (1996)
12. Istrail, S.: An arithmetical hierarchy in propositional dynamic logic. Inf. Comput. 81(3), 280–289 (1989)
13. Kozen, D.: Kleene algebra with tests. ACM Trans. Program. Lang. Syst. 19(3), 427–443 (1997)
14. Kozen, D., Parikh, R.: An elementary proof of the completeness of PDL. Theor. Comp. Sci. 14, 113–118 (1981)
15. Leivant, D.: Matching explicit and modal reasoning about programs: A proof theoretic delineation of dynamic logic. In: LICS. pp. 157–168. IEEE Computer Society (2006)
16. Loos, S.M., Platzer, A.: Safe intersections: At the crossing of hybrid systems and verification. In: Yi, K. (ed.) ITSC. pp. 1181–1186. Springer (2011)
17. Loos, S.M., Platzer, A., Nistor, L.: Adaptive cruise control: Hybrid, distributed, and now formally verified. In: Butler, M., Schulte, W. (eds.) FM. LNCS, vol. 6664, pp. 42–56. Springer (2011)
18. Meyer, A.R., Parikh, R.: Definability in dynamic logic. J. Comput. Syst. Sci. 23(2), 279–298 (1981)
19. Mitsch, S., Loos, S.M., Platzer, A.: Towards formal verification of freeway traffic control. In: Lu, C. (ed.) ICCPS. pp. 171–180. IEEE (2012)
20. Parikh, R.: The completeness of propositional dynamic logic. In: Winkowski, J. (ed.) MFCS. LNCS, vol. 64, pp. 403–415. Springer (1978)
21. Peleg, D.: Concurrent dynamic logic. J. ACM 34(2), 450–479 (1987)
22. Platzer, A.: Differential dynamic logic for verifying parametric hybrid systems. In: Olivetti, N. (ed.) TABLEAUX. LNCS, vol. 4548, pp. 216–232. Springer (2007)
23. Platzer, A.: Differential dynamic logic for hybrid systems. J. Autom. Reas. 41(2), 143–189 (2008)
24. Platzer, A.: Differential Dynamic Logics: Automated Theorem Proving for Hybrid Systems. Ph.D. thesis, Department of Computing Science, University of Oldenburg (Dec 2008), appeared with Springer
25. Platzer, A.: Differential-algebraic dynamic logic for differential-algebraic programs. J. Log. Comput. 20(1), 309–352 (2010)
26. Platzer, A.: Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics. Springer, Heidelberg (2010)
27. Platzer, A.: Quantified differential dynamic logic for distributed hybrid systems. In: Dawar, A., Veith, H. (eds.) CSL. LNCS, vol. 6247, pp. 469–483. Springer (2010)
28. Platzer, A.: Stochastic differential dynamic logic for stochastic hybrid programs. In: Bjørner, N., Sofronie-Stokkermans, V. (eds.) CADE. LNCS, vol. 6803, pp. 431–445. Springer (2011)

29. Platzer, A.: A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems. Logical Methods in Computer Science (2012), special issue for selected papers from CSL'10
30. Platzer, A.: The complete proof theory of hybrid systems. In: LICS [1]
31. Platzer, A.: Logics of dynamical systems (invited tutorial). In: LICS [1]
32. Platzer, A.: The structure of differential invariants and differential cut elimination. Logical Methods in Computer Science (2012), to appear
33. Platzer, A., Clarke, E.M.: Computing differential invariants of hybrid systems as fixedpoints. In: Gupta, A., Malik, S. (eds.) CAV. LNCS, vol. 5123, pp. 176–189. Springer (2008)
34. Platzer, A., Clarke, E.M.: Computing differential invariants of hybrid systems as fixedpoints. Form. Methods Syst. Des. 35(1), 98–120 (2009), special issue for selected papers from CAV'08
35. Platzer, A., Clarke, E.M.: Formal verification of curved flight collision avoidance maneuvers: A case study. In: Cavalcanti, A., Dams, D. (eds.) FM. LNCS, vol. 5850, pp. 547–562. Springer (2009)
36. Platzer, A., Quesel, J.D.: KeYmaera: A hybrid theorem prover for hybrid systems. In: Armando, A., Baumgartner, P., Dowek, G. (eds.) IJCAR. LNCS, vol. 5195, pp. 171–178. Springer (2008)
37. Platzer, A., Quesel, J.D.: European Train Control System: A case study in formal verification. In: Breitman, K., Cavalcanti, A. (eds.) ICFEM. LNCS, vol. 5885, pp. 246–265. Springer (2009)
38. Platzer, A., Quesel, J.D., Rümmer, P.: Real world verification. In: Schmidt, R.A. (ed.) CADE. LNCS, vol. 5663, pp. 485–501. Springer (2009)
39. Pratt, V.R.: Semantical considerations on Floyd-Hoare logic. In: FOCS. pp. 109–121. IEEE (1976)
40. Reif, W., Schellhorn, G., Stenzel, K.: Proving system correctness with KIV 3.0. In: McCune, W. (ed.) CADE. LNCS, vol. 1249, pp. 69–72. Springer (1997)
41. Renshaw, D.W., Loos, S.M., Platzer, A.: Distributed theorem proving for distributed hybrid systems. In: Qin, S., Qiu, Z. (eds.) ICFEM. LNCS, vol. 6991, pp. 356–371. Springer (2011)
42. Segerberg, K.: A completeness theorem in the modal logic of programs. Notices AMS 24, 522 (1977)