

52 Logical Methods in Computer Science Vol. 8 (4:17) 2012, pp. 1-44 www.lmcs-online.org Submitted Published Jan. 7, 2011 Nov. 23, 2012

A COMPLETE AXIOMATIZATION OF QUANTIFIED DIFFERENTIAL DYNAMICAL LOGIC FOR DISTRIBUTED HYBRID SYSTEMS AND REPLAZER

Carnegie Mellon University, Computer Science Department, Pittsburgh, PA, USA
e-mail address: aplatzer@cs.cmu.edu

Abstract. We address a fundamental mismatch between the combinations of dynamics that occur in cyber-physical systems and the limited kinds of dynamics supported in analysis. Modern applications combine communication, computation, and control. They may even form dynamic distributed networks, where neither structure nor dimension stay the same while the system follows hybrid dynamics, i.e., mixed discrete and continuous dynamics. We provide the logical foundations for closing this analytic gap. We develop a formal model for distributed hybrid systems. It combines quantified differential equations with quantified assignments and dynamic dimensionality-changes. We introduce a dynamic logic for verifying distributed hybrid systems and present a proof calculus for this logic. This is the first formal verification approach for distributed hybrid systems. We prove that our calculus is a sound and complete axiomatization of the behavior of distributed hybrid systems relative to quantified differential equations. In our calculus we have proven collision freedom in distributed car control even when an unbounded number of new cars may appear dynamically on the road.

1. Introduction

Many safety-critical computers are embedded in cyber-physical systems like cars [HESV91, RS+06] and aircraft [DMC05]. How do we know that their designs will work as intended? Most initial designs do not. And some deployed systems still do not. Ensuring the correct functioning of cyber-physical systems is a central challenge in computer science, mathematics, and engineering, because it is the key to designing smart and reliable control. Scientists and engineers need analytic tools to understand and predict the behavior of their systems. As systems become ever more complex, it becomes prohibitively expensive or impossible to 1998 ACM Subject Classification: F.3.1, F.4.1, D.2.4, C.1.m, C.2.4, D.4.7.

Key words and phrases: Differential dynamic logic, Distributed hybrid systems, Axiomatization, Theorem proving, Quantified differential equations, Proof theory.

An extended abstract has appeared at CSL'10 [Pla10c]. 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 NASA grant NNG-05GF84H, and by the ONR award N00014-10-1-0188.

LOGICAL METHODS IN COMPUTER SCIENCE
DOI: 10.2168/LMCS-8(4:17)2012 CC BY-PLATZER

test all possible interactions and rule out unsafe behavior by simulation. Formal verification techniques are used routinely to overcome this for finite systems. But for cyber-physical systems, there is not even a foundation for verification that would cover all required behavior. There is a fundamental mismatch between the actual dynamics of cyber-physical system applications and the limits imposed by current modeling and analysis. Cyber-physical systems in automotive, aviation, railway, and power grids combine communication, computation, and control. Combining computation and control leads to hybrid systems [ACHH92, Bra95, Hen96, BBM98, Pla10b], whose behavior involves both discrete and continuous dynamics originating, e.g., from discrete control decisions and differential equations of motion. Combining communication and computation leads to distributed systems [Lyn96, AL01, AdBO10], whose dynamics are discrete transitions of system parts that communicate with each other. They may form dynamic distributed systems, where the structure of the system is not fixed but evolves over time and agents may appear or disappear during the system evolution. Combinations of all three aspects (communication, computation, and control) are used in sophisticated applications, e.g., cooperative (2)(2)(3)(3)(4)(4)(1)(1) operative distributed car control [HESV91] and decentralized aircraft control [PSFB07]. Neither the structure nor dimension of the system stay the same, because new Figure 1: Distributed car control. cars can appear on the street or leave it; see Fig. 1. These systems are (dynamic) distributed hybrid systems, i.e., systems that combine the dynamics of distributed systems with the discrete and continuous dynamics of hybrid systems. More generally, distributed hybrid systems are multi-agent hybrid systems that interact through remote communication or physical interaction. They cannot be considered just as a distributed system (because, e.g., the continuous evolution of positions and velocities matters crucially for collision freedom in car control) nor just as a hybrid system (because the evolving system structure and appearance of new agents can make an otherwise collision-free system unsafe). It is generally impossible to split the analysis of distributed hybrid systems soundly into an analysis of a distributed system (without continuous movement) and an analysis of a hybrid system (without structural changes or appearance), because all kinds of dynamics interact. Just like hybrid systems are difficult to analyze from a purely discrete or a purely continuous perspective [Hen96, Pla12]. Distributed hybrid systems have been considered to varying degrees in modeling languages [DGV96, Rou04, KSP06, MS06]. In order to build these systems, however, scientists and engineers also need analytic tools to understand and predict their behavior. But formal verification and proof techniques do not yet support the required combination of dynamical effects which is not surprising given the numerous sources of undecidability for distributed hybrid systems verification. In this article, we provide the logical foundations to close this fundamental analytic gap. We develop quantified hybrid programs (QHPs) as a formal model for distributed hybrid systems, which combine dynamical effects from multiple sources: discrete transitions, continuous evolution, dimension changes, and structural dynamics. In order to account A COMPLETE AXIOMATIZATION OF QDL FOR DISTRIBUTED HYBRID SYSTEMS 3 for changes in the dimension and for co-evolution of an unbounded and evolving number of participants, we generalize the notion of states from assignments for primitive system variables like x to full first-order structures. In a QHP, function term $x(i)$ may denote the position of car i of type C , the term $f(i)$ could be the car registered by communication as the car following car i , and the term $d(i, f(i))$ could denote the minimum safety distance negotiated between car i and its follower $f(i)$. The values of all these terms may evolve for all i as time progresses according to interacting laws of discrete and continuous dynamics, because all cars evolve simultaneously. They are also affected by changing the system dimension as new cars appear, disappear, or by reconfiguring the system structure dynamically, e.g., by remote communication or physical interaction. The defining characteristic of QHPs is that they allow quantified hybrid dynamics in which variables like i that occur in function arguments of the system dynamics are quantified over, such that the system co-evolves, e.g., for all cars i of type C . This quantification is necessary to characterize the distributed hybrid systems dynamics with an unbounded and possibly evolving number of participants. Quantification is also necessary to represent structural dynamics when the number of participants is not fixed. There is a crucial difference between a primitive system variable x and a first-order function term $x(i)$, where i is quantified over. Hybrid dynamics of primitive system variables can model a concrete number of, say, four cars (putting scalability issues aside), but neither a parametric number of n cars nor systems with a variable number of cars (a number n that may change over time). With first-order function symbols $x(i)$ and hybrid dynamics quantifying over all cars i , a single QHP can represent any number of cars at once. QHPs can even represent (dis)appearance of cars by changing the domain that quantifiers range over dynamically at runtime. QHPs are thus a formal model for general (dynamic) distributed hybrid systems. Verification of distributed hybrid systems is challenging. We show that they have three independent sources of undecidability: discrete dynamics, continuous dynamics, and structural / dimensional dynamics. As an analysis tool for distributed hybrid systems, we introduce a specification and verification logic for QHPs that we call quantified differential dynamic logic (QdL). QdL provides dynamic logic [Pra76, HKT00] modal operators $[\pm]$ and $h \pm i$ that refer to the states reachable by QHP \pm and can be placed in front of any formula. Formula $[\pm]$ expresses that all states reachable by system \pm satisfy formula, while $h \pm i$ expresses that there is at least one reachable state satisfying. These modalities can express necessary or possible properties of the transition behavior of QHP \pm . With its ability to specify and verify properties of (dynamic) distributed hybrid systems and quantified dynamics, QdL is a major extension of prior work for static hybrid systems [Pla08].