Proceedings of the 36th
Conference on Decision & Control
San Diego, California USA • December 1997



# Logic, Topological Semantics and Hybrid Systems

Sergei Artemov, Jennifer Davoren and Anil Nerode [1]

Mathematical Sciences Institute
Cornell University
Ithaca, NY 14853
{artemov,jennifer,anil}@math.cornell.edu

**Abstract**

This note is a preliminary discussion of logics and semantics for the specification, development, and verification of hybrid control systems, with special attention to the central issues of continuity and stability.

## 1 Introduction

This is a preliminary discussion of logics and semantics for the specification, development, and verification of hybrid control systems. In general, hybrid systems are interacting networks of continuous (usually nonlinear) plants and discrete automata. The discrete automata are control automata controlling actuators used to enforce desired plant behavior.

In the simplest case, a hybrid system consists of a continuous plant interacting with a digital control automaton. We assume that interactions between the plant and the control automaton occur at discrete times, say $t_n = n\Delta$. The plant is modelled by a system of autonomous differential equations $\dot{x} = f(x, c)$, where $x$ is trajectory on the plant state space $X$ and $c \in C$ is a control parameter. Hybrid systems are dynamical systems with mixed continuous and discrete states.

The fundamental problem of hybrid systems is to find algorithms which, given continuous plant differential equations and plant performance specifications which may include logical constraints, extract and verify digital control programs that force the state trajectories of the system to obey their performance specifications.

We have found the available logics and semantics inadequate to describe continuous dynamics, much less hybrid dynamics, because they do not deal with the continuity and stability issues central to continuous systems.

Kohn and Nerode ([10], [5], [6]) developed models and algorithms for formulating and solving this extraction and verification problem in a wide variety of cases. They reformulate the problem as a relaxed calculus of variations optimization problem on a suitably defined "carrier" manifold. The problem becomes one of finding a finite state control policy which, deployed, will force the system to come within a user defined $\epsilon$ of the mimimum of the total cost function for the relaxed calculus of variations problem. The $\epsilon$ represents a user compromise based on cost and feasibility.

## 2 Motivation for Topological Semantics I

In going from the approximate optimization problem to a finite state control automaton executing the control policy which ensures this approximate optimality, careful algorithms allow one to compute how close each parameter or constant of the simulation model (manifold description) has to be to its assumed value to assure $\epsilon$-optimality. These are the "margins of error" in the numerical descriptions of actuators, sensors, and physical process which can be allowed and still guarantee mathematically that if the model dynamics are correct, then the policy is $\epsilon$-optimal. This is a backward-chaining $\epsilon - -\delta$ argument for the relaxed dynamic programming problem on a manifold.

The *delta*'s specified then define open sets associated with the atomic propositions that assert values for parameters and data. As long as one is in the open set associated with each parameter and datum, the optimality is guaranteed. We think of these as open sets *denoted* by these atomic propositions which assert values of constants and parameters.

Suppose one has computed such open set denotations for these atomic propositions. By ensuring that the values of constants and parameters are within the appropriate denoted open sets, we guarantee that the proposition asserting that the total cost is the mimimum has as its denotation an open set *containing* an open ball of radius $\epsilon$ around the minimum value. Such an open ball is an open *subset* of the open set denoting the minimum assertion, and there can be other such $\epsilon$ balls.

[1]Research for all authors supported by the ARO under the MURI program "Integrated Approach to Intelligent Systems", grant number DAA H04-96-1-0341.

0-7803-3970-8/97 $10.00 © 1997 IEEE

We wish to introduce a notion of topological validity by extending the assignment of open sets from atomic statements to all statements, with an interpretation similar to that above. It is highly dependent on the particular extraction procedure, because the assignment to atomic propositions is determined by computations in the extraction procedure. The topologically valid statements are those whose denotation is the whole state space, so these will be "invariants" during the operation of the hybrid system. As long as model parameters and constants are in the open sets denoted by corresponding value assertions, topological validity persists. As the system runs, if one observes exceptions to a topologically valid assertion, the model has failed. If we then wish to restore $\epsilon$-optimality, we need either to enlarge $\epsilon$, to revise parameters and constants, or to change the structural model of the continuous plant, actuators, and sensors.

## 3 Motivation for Topological Semantics II

We take it as an engineering requirement, as in [9], that we do not build a system unless arbitrarily close continuous states fire the same digital input symbols. This entails that the inverse image of each digital input symbol under an analog to digital conversion is an open set in the topology on the continuous state space, and that only the finite subtopology generated by those finitely many open sets which fire input letters is used in the decision making of the digital control program. This means that two states of the continuous physical system are indistinguishable by the control program if they have all the same neighborhoods in this finite topology. The Kohn-Nerode extraction procedure does produce programs with this property. They come equipped with a finite topology.

We want hybrid systems languages and language semantics in which proving a digital control program correct automatically proves that the control program obeys this requirement. When we prove that a control program makes the hybrid system satisfy its program specification, we want this to entail that if any state observed is replaced by any other state indistinguishable from it according to its associated finite topology, the specification is still satisfied by the hybrid system. We also want hybrid system languages expressive enough to deal with extracting such control programs. We would like to develop languages and semantics useful to a wide variety of researchers, developers and users. But we begin more modestly, by discussing our own home territory, the Kohn-Nerode hybrid systems architecture, where we have a quite concrete idea of what we need.

## 4 Current Language for Kohn-Nerode Hybrid Systems Architecture

What language naturally describes the Kohn-Nerode architecture for extracting digital control programs programs from mathematical simulation models of the sensors, physical system, and actuators?

### First Layer

In the Kohn-Nerode architecture, the digital control programs themselves are expressed as declarative input-output Horn clause programs based on a finite number of relation and constant symbols. The inputs are letters representing digitized states, the outputs are letters representing digitized control orders for the actuators of the physical system. The program is an exectutable explicit description of a finite automaton, but its state diagram, which may contain millions of states, is never computed. Instead, the answer substitution mechanism of Prolog is used to answer the question "What control law do I use *NOW* ?" based on the Horn clause program plus the input atomic statments which have been fired by sensing the state of the controlled physical system.

### Meta Layer

The Kohn-Nerode architecture has an *agent program* for each digital control program which observes the behavior of that program and of the physical system it controls. If the agent program observes a deviation of system performance from the system performance specification, usually expressed as a violation of a quantitative or logical conservation law, the agent program concludes that the model of the physical system, on the basis of which the control program was extracted, needs to be revised. It revises the model, extracts a new digital control program, and then substitutes the new program for the one in use. The Kohn-Nerode agent solves a relaxed variational problem on a manifold describing the revised model and approximates to get a new digital control program. The agent runs on a much slower time scale than the digital conrol program.

A Kohn-Nerode agent is a Horn clause program with assert and retract predicates. A simulation model of the physical system is a Horn clause subprogram of the agent. To change this model when violations of conservation laws are observed, one has to retract and add clauses, which is why the agent program has to be a meta-logic program. So a complete account of the logic and semantics of hybrid systems has to include metaprogramming semantics in order to permit this kind of adaptation for unmodelled dynamics. We do not discuss metaprogram semantics here, but the topological semantics described below can be extended to

metaprograms of agents by generalizing several known ways ways of giving semantics to assert and retract.

Here we deal only with the semantics of the Horn clause logic program used to control the physical system. In accord with the discussion above we take it that each input program clause is fired by an open set of points in the state space of the continuous system. An input program clause is an atomic statement. There may be several fired by a single state. We regard each atomic statement as denoting the open set of the state space that fires it. We can give an open set denotation to every statement in propositional logic, in particular to all Horn clause propositional programs. Assume the logic is based on "and", "or", "implies", "not". The denotation of "$A$ or $B$" is the union of the denotations of $A$ and $B$. The denotation of "$A$ and $B$" is the intersection of the denotations of $A$ and of $B$. The denotation of "$A$ implies $B$" is the interior of the union of the complement of the denotation of $A$ and the denotation of $B$. The denotation of the negation of $A$ is the interior of the complement of the denotation of $A$. This is the Tarski topological semantics ([11]) for the Intuitionistic propositional logic based on these connectives and based on assigning open subsets of the state space to atomic statements. Since there are only a finite number of input symbols for the automaton, there are only finitely many atomic statements that can be fired, and all denotations of all propositional logic statements are contained in a finite subtopology of the state space of the continuous system generated by the denotations of atomic statements. A propositional statement semantically valid in this model is one denoting the open set which is the whole space, while a statement is a semantic contradiction in this model if it denotes the empty set.

Over such a finite model, how would one interpret a several-input/single-output Horn clause program $P$ as a non-deterministic finite automaton? There is a transition from input atomic statements $A_1, ..., A_n$ to output atomic statement $B$ iff the Horn program $P$ together with $A_1, ..., A_n$ semantically entails $B$ in the finite topological model above. This is a decidable question. Validity of statements in such a finite topological model is decidable. To see this, write the inductive definition of validity out in the form of forcing over the finite partially ordered set of open sets of the topology. Observe that due to the finiteness of the space, this can be checked.

In Kohn-Nerode, $A_1, ...., A_n$ are the input atomic statements fired by the current state being in the denoted open sets, and $B$ is the statement as to what control order should be sent to an actuator. In fact, the Kohn-Nerode control programs are expressed in predicate Horn logic and the agent programs are in meta predicate Horn logic.

As explained above, the Prolog input-output control program can not deal with points in the state space of the physical system directly. Rather it deals with equivalence classes of points under indistinguishability with respect to the finite topology generated by denotations of input statements for the input-output Prolog program. We may regard the Prolog control program as a logic program with its intended domain a set of names for the indistinguishability classes. These classes alternately may be characterized as sets of the form: an open set (of the finite topology) minus all smaller open sets.

All Prolog predicates in the control program may as well be taken as predicates about this finite domain of indistinguishability classes. If we have a name for each indistinuishability class, the extension of a topological valuation of atomic statements of predicate logic to arbitrary statements is easy. Over a finite domain in which everything has a name, existential quantifiers are finite disjunctions of all instances and universal quantifiers are finite conjunctions of all instances. So every first order statement over the domain denotes an open set, in particular Horn clause programs, which are simply conjunctions of Horn implications. So a topological semantics for first order logic is then associated with a finite state Prolog control program which has associated open set denotations for atomic statements. In practice, these represent tolerances necessary so that the extracted program is $\epsilon$-optimal.

When the domain is infinite, one has to use Tarski's general definition to extend a valuation of atomic statements to all statements. The change is that universal quantification denotes the interior of the intersection of denotations of all instances. (This is obviously unnecessary for finite spaces since the finite intersection of open sets is open.) Existential quantification denotes the union of denotations of all instances.

Agent programs require this twist for their topological semantics. Since they contain a simlation model of the system, they have to have the actual state space of the physical system as a domain, and the domain here is many-sorted. The predicates on this domain are open relations between points of that state space. The models used above, consisting of finite subtopologies on the state space, are crude approximations to the "full" model of the state space equipped with its standard topology. When the topology on the state space is compact Hausdorff, it can be represented as the inverse limit of the finite subtopology models. The whole subject of hybrid systems can be seen as the interplay involved in this limit, between constinuous state spaces and their finite approximations. Symbolic dynamics works in much the same way, but has not in the past emphasized input-output behavior or the automata inherent in approximation.

What about formal reasoning about topological models? The axioms and rules of inference of Intuitionistic logic are exactly those that preserve validity in all Tarski topological models. So in reasoning about finite topology models or about the topological models of the physical system state space, all valid Intuitionistic axioms and rules of inference may be used. But one has to be careful not to use the rest of classical logic if topological validity is to be preserved.

## 5 Other Work in Progress

The modal logics of programs are attractive, but are not able to deal with continuous or mixed dynamical systems. We have been investigating the modalities associated with hybrid and continuous systems.

In the second author's dissertation [3], and in [1], several new modal logics are developed and investigated. The logics are polymodal extensions of the classical modal propositional logic **S4**, obtained by adjoining new modal operators $[c]$ whose topological semantics are given by the inverse image of a total function on the state space. Intuitionistic propositional logic can be faithfully embedded in these new logics in a standard way. In **S4**, the modalities $\Box$ and $\Diamond$ correspond topologically to the interior and closure operators, respectively, so in the new logics, the scheme $[c]\Box A \rightarrow \Box[c]A$, for all sentences $A$, expresses the *continuity* of the function interpretting the $[c]$ modality. Being able to express continuity, and hence *stability*, in the language is a distinct advantage over other logics and fomalisms for hybrid systems.

## References


[1] Sergei Artemov, Jennifer Davoren and Anil Nerode. Modal Logics and Topological Semantics for Hybrid Systems. Technical Report 97-05, Mathematical Sciences Institute, Cornell University, June 1997.

[2] Dirk van Dalen. Intuitionistic Logic. In D. Gabbbay and F. Guenthner (eds.), *Handbook of Philosophical Logic, Volume III: Alternatives to Classical Logic.* D. Reidel, Dordrecht, 1986; 225-339.

[3] Jennifer Davoren. Modal Logics for Continuous Dynamics. Ph.D. Dissertation, Department of Mathematics, Cornell University, August 1997.

[4] Wolf Kohn, Anil Nerode, Jeffrey B. Remmel and Alexander Yakhnis. Viability in Hybrid Systems. *Theoretical Computer Science* **138** (1995) 141-168.

[5] Wolf Kohn, Anil Nerode and Jeffrey B. Remmel. Hybrid Systems as Finsler Manifolds: Finite State Control as Approximation to Connections. In P. Antsaklis, W. Kohn, A. Nerode and S. Sastry (eds.), *Hybrid Systems II*, Lecture Notes in Computer Science **999**, Springer-Verlag, Berlin, 1995; 294-321.

[6] Wolf Kohn, Anil Nerode and Jeffrey B. Remmel. Continualization: A Hybrid Systems Control Technique for Computing. *Proceedings of 1996 IMACS Conference on Computation Engineering in Systems Applications (CEAS '96).*

[7] John W. Lloyd. *Foundations of Logic Programming.* 2nd edition. Springer-Verlag, Berlin 1987.

[8] Anil Nerode. Some Stone Spaces and Recursion Theory. *Duke Mathematical Journal* **26** (1959) 397-406.

[9] Anil Nerode and Wolf Kohn. Models for Hybrid Systems: Automata, Topologies, Controllability, Observability. In R. Grossman, A. Nerode, A. Ravn and H. Rischel (eds.), *Hybrid Systems*, Springer Lecture Notes in Computer Science **736**, Springer-Verlag, Berlin, 1993; 297-316.

[10] Anil Nerode and Wolf Kohn. Multiple Agent Hybrid Control Architecture. In R. Grossman, A. Nerode, A. Ravn and H. Rischel (eds.), *Hybrid Systems*, Springer Lecture Notes in Computer Science **736**, Springer-Verlag, Berlin, 1993; 317-356.

[11] Alfred Tarski. Der Aussagenkalkül und die Topologie. *Fundamenta Mathematicae* **31** (1938) 103-134. Reprinted (and translated by J. H. Woodger) as: Sentential Calculus and Topology. In A. Tarski, *Logic, Semantics, Metamathematics*, Oxford University Press, 1956; 421-454.