Reachability in Constraint Databases and Model Checking for Hybrid Systems*

Michael Benedikt† Bell Laboratories
Martin Grohe‡ Univ. Freiburg
Leonid Libkin§ Bell Laboratories
Luc Segoufin¶ INRIA

Introduction. There is a fundamental connection between database theory and automated verification: the problem of evaluating a Boolean query against a finite relational database and the model-checking problem, (that is, deciding whether the state space of a finite state system, viewed as a Kripke structure, satisfies a given specification), are essentially the same. In both cases, the problem is to decide whether a finite structure satisfies a sentence of some logic.

We establish a similar connection on the level of infinite structures. We introduce a new query language for *constraint databases* called *path logic*. We then show how this language can be used to show the tractability of model-checking for temporal logic on *hybrid systems*.

The results mentioned in this abstract are all contained in [1], which studies various languages expressing reachability and connectivity queries in constraint databases. For background material we refer the reader to [5] on constraint databases and [3] on hybrid systems.

Constraint Databases. In constraint databases, possibly infinite relations on an infinite background structure $\mathcal{M}$ are represented by finite sets of constraints, that is, quantifier-free formulas over some structure. In this abstract we restrict our attention to the background structures $\mathbf{R}_{\text{lin}} = \langle \mathbb{R}, +, -, 0, 1, < \rangle$ and $\mathbf{R} = \langle \mathbb{R}, +, \cdot, 0, 1, < \rangle$. More generally, many of our results remain true if $\mathcal{M}$ is an *o-minimal structure* (cf. [8]).

A set $S \subseteq \mathbb{R}^n$ is *semi-algebraic* if it is the set of solutions to a finite set of polynomial equalities and inequalities in n variables. If the defining polynomials are all linear, then we call S *semi-linear*. Since both $\mathbf{R}_{\text{lin}}$ and $\mathbf{R}$ admit quantifier-elimination, semi-algebraic and semi-linear sets are precisely the *(first-order) definable* sets over $\mathbf{R}$ and $\mathbf{R}_{\text{lin}}$, respectively.

A *database schema* SC is a nonempty collection of relation names $\{S_1, \ldots, S_l\}$ with associated arities $p_1, \ldots, p_l > 0$. A *database instance* of SC over $\mathcal{M}$ is an expansion $D = \langle \mathcal{M}, X_1, \ldots, X_l \rangle$ of $\mathcal{M}$, where $X_i \subseteq \mathbb{R}^{p_i}$ is definable over $\mathcal{M}$. A *finite representation* of D is a collection of defining equalities and inequalities for the sets X_i.

The basic query language over constraint databases *first-order logic* FO over the language of $\mathcal{M}$ and the schema SC. For example, the sentence $\exists a \exists b \forall x \forall y \ (S(x,y) \to a \cdot x + b = y)$ over $\mathbf{R}$ and a binary relation S tests if the semi-algebraic set that interprets S is a subset of a line.

Path Logic. It is one of the fundamental results of the theory of constraint databases that (topological) connectivity is not expressible in FO [5]. *Path logic* L_{PATH} is an extension of FO in which reachability and connectivity queries are expressible. The language is based on the concept of a path and allows to express properties of paths with respect to given sets in the Euclidean space $\mathbb{R}^n$. For this abstract, let us think of a path as a continuous curve in $\mathbb{R}^n$. For example, to express that a set $S \subseteq \mathbb{R}^n$ is connected we would say "for all $\vec{x}, \vec{y} \in S$ there exists a path containing $\vec{x}, \vec{y}$ such that for all points p on this path which appear between $\vec{x}$ and $\vec{y}$ we have $p \in S$".

We omit the formal definition of the logic L_{PATH}, which is quite technical. In defining our path logic we had to be very careful in order to obtain a logic that is *closed* and *decidable*. Closure means that the result of applying a query to a constraint database yields a definable set. Decidability means that, given a finite representation of a database instance D and a sentence φ, it is decidable whether D satsifies φ. Closure and decidability are the most fundamental properties required from a reasonable query language for constraint databases.

Theorem 1. *L_{PATH} is closed and decidable. Furthermore, its data-complexity is in* PTIME*, i.e. for every sentence $\varphi \in L_{\text{PATH}}$ there is a polynomial time algorithm that, given a finite representation of a database instance D, decides whether D satsifies φ.*

Hybrid Systems. A *hybrid system* of dimension n is a tuple $H = (S, S_0, S_F, F, E, I, G, R)$, where

*Part of this work was done while M. Benedikt, M. Grohe and L. Libkin visited INRIA-Rocquencourt.

†Bell Laboratories, 263 Shuman Blvd, Naperville, IL 60566, USA. E-mail: benedikt@research.bell-labs.com.

‡Institut für Mathematische Logik, Eckerstr. 1, 79104 Freiburg, Germany. E-mail: grohe@logik.mathematik.uni-freiburg.de.

§Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974, USA. E-mail: libkin@research.bell-labs.com.

¶INRIA-Rocquencourt, B.P. 105, Le Chesnay Cedex 78153, France. E-mail: Luc.Segoufin@inria.fr.

- $S = Q \times \mathbb{R}^n$, where Q is a finite set, is the *state space*,
- $S_0 \subseteq S$ is the set of *initial states*,
- $S_F \subseteq S$ is the set of *final states*,
- $F : S \to \mathbb{R}^n$ assigns to each $q \in Q$ a vector field $F(q, \cdot)$,
- $E \subseteq Q \times Q$ is a set of *discrete transitions*,
- $I : Q \to 2^{\mathbb{R}^n}$ assigns to each $q \in Q$ a set $I(q)$ called the *invariant of* q,
- $G : E \to 2^{\mathbb{R}^n}$ assigns to each discrete transition $e = (q_1, q_2) \in E$ a set $G(e) \subseteq I(q_1)$ called the *guard of* e,
- $R : E \to 2^{\mathbb{R}^n}$ assigns to each discrete transition $e = (q_1, q_2) \in E$ a set $R(e) \subseteq I(q_2)$ called the *reset of* e.

Associated with the hybrid system H is a ternary *transition relation* $\to \subseteq S \times (E \cup \{c\}) \times S$, where c is a new symbol not contained in E. We write $s \xrightarrow{e} s'$ instead of $(s, e, s') \in \to$. We have two kinds of transitions:

- *Discrete Transitions:* $(q, \vec{x}) \xrightarrow{e} (q', \vec{x}')$ iff $e = (q, q') \in E$ and $\vec{x} \in G(e)$, $\vec{x}' \in R(e)$.
- *Continuous Transitions:* $(q, \vec{x}) \xrightarrow{c} (q', \vec{x}')$ iff $q = q'$ and there exists a $\delta \geq 0$ and a curve $x : [0, \delta] \to \mathbb{R}^n$ such that $x(0) = \vec{x}$, $x(\delta) = \vec{x}'$ and for every $t \in [0, \delta]$ it satisfies $\dot{x}(t) = F(q, x(t))$ and $x(t) \in I(q)$.

A *trajectory* of H is a sequence $s_1 e_1 s_2 e_2 \ldots$ such that for all $i \geq 1$ we have $s_i \xrightarrow{e_i} s_{i+1}$.

An *interpreted hybrid system* of signature $\Sigma = \{\pi_1, \ldots, \pi_m\}$ consists of a hybrid system H and a mapping Π that assigns to each state $s \in S$ a subset of Σ. Then Π associates with each trajectory $\tau = s_1 e_1 s_2 e_2 \ldots$ of H an ω-word $\Pi(\tau) := \Pi(s_1)\Pi(s_2)\ldots$ over the alphabet 2^Σ. We assume that the reader is familiar with the linear time temporal logic LTL (interpreted over ω-words), see [2]. The LTL-*model checking problem for hybrid systems* is defined as follows:

Input:	An interpreted hybrid system (H, Π) and an LTL-formula φ.
Problem:	Decide if for every trajectory τ of H the word $\Pi(\tau)$ satisfies φ.

In general, this problem is undecidable [4]. We now consider a special class of hybrid systems introduced in [6, 7]. Let $\mathcal{M} = \mathbf{R}$ or $\mathcal{M} = \mathbf{R}_{\text{lin}}$ as before. (More generally, if we let $\mathcal{M}$ be an o-minimal structure over the reals, then we obtain the class of *o-minimal hybrid systems.*) A hybrid system $H = (S, S_0, S_F, F, E, I, G, R)$ is *$\mathcal{M}$-definable* if $Q \subseteq \mathbb{R}$ is a definable set and the sets S_0, S_F, the mappings I, G, R, and the relation $T := \{(q, \vec{x}, \vec{y}) \mid (q, \vec{x}) \xrightarrow{c} (q, \vec{y})\}$ are definable. An interpreted hybrid system (H, Π) is *$\mathcal{M}$-definable* if H is $\mathcal{M}$-definable, and for every π in the signature, the set $\Pi^{-1}(\pi)$ is definable.

Theorem 2. *For every fixed LTL-formula φ and $n \geq 1$, the restriction of the LTL-model-checking problem to $\mathcal{M}$-definable interpreted hybrid systems of dimension n can be solved in* PTIME.

Proof sketch. We code hybrid systems as constraint databases and show that for every LTL-formula φ, there exists and can be effectively found an L_{PATH} formula φ^* such that, for every interpreted hybrid system (H, Π), we have $(H, \Pi) \models \varphi^*$ iff for every trajectory τ of H, $\Pi(\tau)$ satisfies φ. Then the result follows from Theorem 1. □

Remarks. (1) The analogous result holds for the branching time temporal logic CTL*.

(2) If $\mathcal{M}$ is a structure over the reals such that its expansion with $+, \cdot, 0, 1$ is a decidable o-minimal structure, then the restriction of the LTL-model-checking problem for hybrid systems to $\mathcal{M}$-definable interpreted hybrid systems is still decidable.

(3) The proof can be easily modified to show that for an $\mathcal{M}$-definable interpreted hybrid system $(H(\vec{x}), \Pi)$ that is (definably) parameterized by real parameters $\vec{x}$, and an LTL formula φ, the set $\{\vec{x} : (H(\vec{x}), \Pi) \models \varphi\}$ is definable. Furthermore, for fixed dimension n as above, the representation of the solution set can be obtained in PTIME from a representation of the system.

References

[1] M. Benedikt, M. Grohe, L. Libkin and L. Segoufin. Reachbility and connectivity queries in constraint databases. In *PODS 2000.*

[2] E. A. Emerson. Temporal and modal logic. Chapter 16 of Volume B of *Handbook of Theoretical Computer Science*, Elsevier, 1990.

[3] T. A. Henzinger. The theory of hybrid automata. In *LICS'96*, pages 278–292.

[4] T. A. Henzinger, P. Kopke, A. Puri, P. Varaiya. What's decidable about hybrid automata? In *STOC 1995.*

[5] G. Kuper, L. Libkin and J. Paredaens, eds. *Constraint Databases.* Springer Verlag, 2000.

[6] G. Lafferriere, G. Pappas and S. Sastry. A new class of decidable hybrid systems. In *Hybrid Systems 1999*, LNCS 1569, Springer, 1999, pages 137–151.

[7] G. Lafferriere, G. Pappas and S. Sastry. O-minimal hybrid systems. *Mathematics of Control, Signals, and Systems*, to appear.

[8] L. van den Dries. *Tame Topology and O-minimal Structures.* Cambridge, 1998.