

Reachability Analysis of Dynamical Systems having Piecewise-Constant Derivatives

Eugene Asarin Oded Maler Amir Pnueli

CNRS - VERIMAG
Grenoble, France

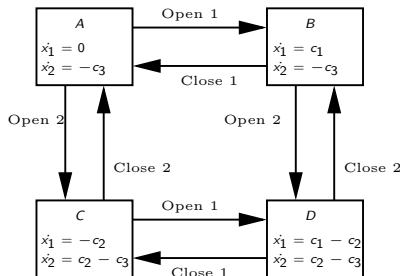
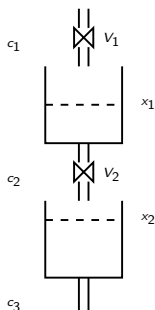
1993-1995

Outline of Talk

- ▶ Some generalities on “linear” hybrid automata and PCD systems
- ▶ Decidability of reachability problems in the plane
- ▶ Undecidability in dimension 3 and above by simulating pushdown stacks
- ▶ Going higher in the arithmetical hierarchy
- ▶ So what?

A Motivating Example: Buffer Networks

- ▶ Consider a network of containers/buffers for water/data
- ▶ Channels can be switched on and off
- ▶ When a channel is on, its flow rate is a constant
- ▶ Each combination of open/close valves leads to a different derivatives for the buffer levels, based on the difference between their in- and outflows

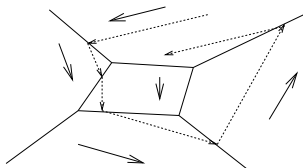


“Linear” Hybrid Automata and PCD Systems

- ▶ A sub-class of hybrid automata
- ▶ Can be viewed as piecewise-trivial dynamical systems: derivatives are constant in every control state (location) and the evolution is along a straight line
- ▶ Transition guards (switching surface) and invariants (staying conditions) are linear (hyperplanes, polytopes)
- ▶ Local continuous evolution needs no numerical analysis; Computing the effect of time passage amounts to quantifier elimination in linear algebra
- ▶ Investigated a lot by Henzinger et al. (HYTECH), currently supported by the tool PHAVER (G. Frehse)
- ▶ PCD (piecewise-constant derivative): a sub-class of linear hybrid automata closer in spirit to continuous dynamical systems

PCD (Piecewise-Constant Derivatives) Systems

- ▶ Dynamical System: $\mathcal{H} = (X, f)$, $X = \mathbb{R}^d$
- ▶ $f : X \rightarrow X$ defines differential equation $\frac{d^+ \mathbf{x}}{dt} = f(\mathbf{x})$
- ▶ A trajectory of \mathcal{H} starting at $\mathbf{x}_0 \in X$ is $\xi : \mathbb{R}_+ \rightarrow X$ s.t.
 - ▶ $\xi(0) = \mathbf{x}_0$
 - ▶ $f(\xi(t))$ is defined for every t and is equal to the right derivative of $\xi(t)$
- ▶ PCD: X is partitioned into a finite number of polyhedra (regions) and f is constant in each region
- ▶ Trajectories are thus broken lines



PCDs are Effective

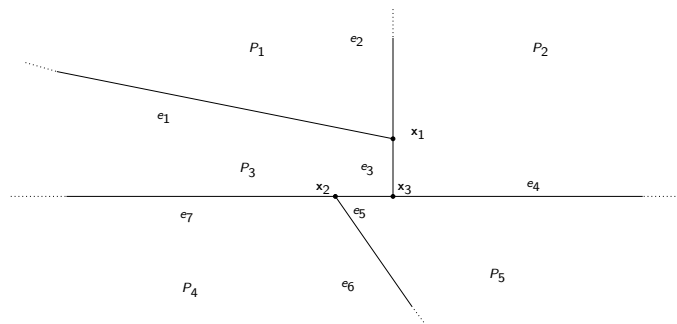
- ▶ A description of a PCD system: $\{(P_1, \mathbf{c}_1), \dots, (P_n, \mathbf{c}_n)\}$
- ▶ each P_i is a convex polyhedron (intersection of linear inequalities) and \mathbf{c}_i is its corresponding derivative (slope)
- ▶ Effectiveness: given a PCD description and a rational point $\mathbf{x} = \xi(0)$
- ▶ There exists $\epsilon > 0$ s.t. we can compute precisely $\mathbf{x}' = \xi(\Delta)$ for every Δ , $0 < \Delta t < \epsilon$; $\mathbf{x}' = \mathbf{x} + \mathbf{c} \cdot \Delta$
- ▶ Unlike arbitrary dynamical systems where you can only approximate

Decision Problems for PCD

- ▶ Point-to-point reachability **Reach**($\mathcal{H}, \mathbf{x}, \mathbf{x}'$):
- ▶ Given: a PCD \mathcal{H} and $\mathbf{x}, \mathbf{x}' \in X$,
- ▶ Are there a trajectory ξ and $t \geq 0$ such that $\xi(0) = \mathbf{x}$ and $\xi(t) = \mathbf{x}'$?
- ▶ Region-to-region reachability **R-Reach**(\mathcal{H}, P, P'):
- ▶ Given: a PCD \mathcal{H} and two polyhedral sets $P, P' \subseteq X$
- ▶ Are there two points $\mathbf{x} \in P$ and $\mathbf{x}' \in P'$ such that **Reach**($\mathcal{H}, \mathbf{x}, \mathbf{x}'$) ?

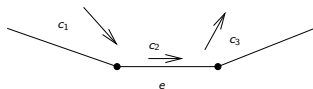
PCDs on the Plane

- ▶ Polyhedral partition of the plane into polygons/regions (P)
- ▶ Induced boundary elements: edges (e) and vertices (x)
- ▶ A kind of abstract finite alphabet to describe qualitative behaviors as sequences of regions or edges

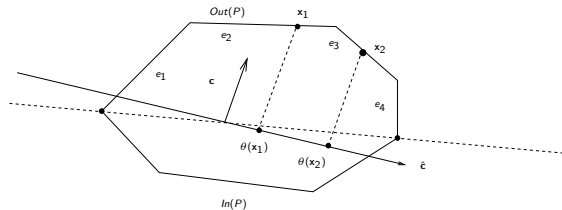


Orientation and Ordering of Boundaries

- ▶ Edges (and vertices) can be classified as entry and exit according to the relation between the slope c and the the vector e which defines the inequality
- ▶ Edge e below is exit for c_1 and entry for c_3

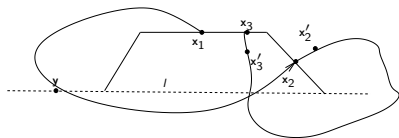
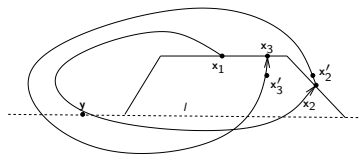


- ▶ The whole boundary of a region can be decomposed into two connected sets, entry $In(P)$ and exit $Out(p)$
- ▶ A linear order can be imposed on each of them:



A Fundamental Property of Planar Systems

- ▶ Let ξ be any trajectory that intersects $Out(P)$ in three consecutive points, \mathbf{x}_1 , \mathbf{x}_2 and \mathbf{x}_3 . Then: $\mathbf{x}_1 \preceq \mathbf{x}_2$ implies $\mathbf{x}_2 \preceq \mathbf{x}_3$

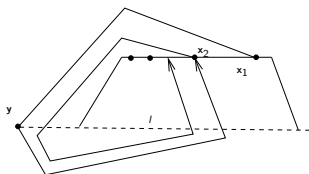


- ▶ The figure shows why it cannot be otherwise as the trajectory must intersect itself
- ▶ Jordan's theorem, not true in 3 dimensions

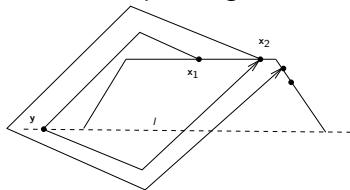
Spirals

- ▶ Consequently all repetitive behaviors are spirals

Contracting:



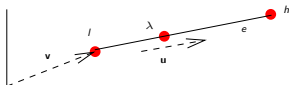
Expanding:



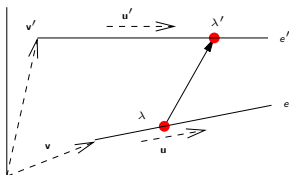
- ▶ The sequences of intersections with an edge is monotonic and you cannot return to an edge you have “abandoned”
- ▶ Since there are finitely many edges we can conclude:
- ▶ For every trajectory, the sequence of edges it crosses is ultimately-periodic: $e_1, \dots, e_i, (e_{i+1}, \dots, e_{i+j})^\omega$

Representation (Parametrization)

- ▶ A representation scheme for an edge e is a pair of vectors \mathbf{v} , \mathbf{u} and an interval $[l, h]$ such that $e = \{\mathbf{v} + \lambda\mathbf{u} : \lambda \in [l, h]\}$



- ▶ Consider an entry edge e with (\mathbf{u}, \mathbf{v}) representation and exit edge e' with $(\mathbf{u}', \mathbf{v}')$ representation
- ▶ The corresponding successor function is defined as $f_{e,e'}(\lambda) = \lambda'$ iff by entering P at $\mathbf{x} = (e, \lambda)$, you exit as $\mathbf{x}' = (e', \lambda')$



Successor Function is Linear

- ▶ Successor function is well-defined, computable and linear:
 $\lambda' = A_{e,e'}\lambda + B_{e,e'}$ where

$$A_{e,e'} = \frac{\mathbf{c} \cdot \mathbf{a}}{\mathbf{c} \cdot \mathbf{a}'} \quad \text{and} \quad B_{e,e'} = \frac{\hat{\mathbf{c}} \cdot (\mathbf{v} - \mathbf{v}')}{\mathbf{c} \cdot \mathbf{a}'}$$

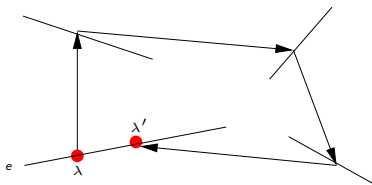
- ▶ Here c is the slope and a and a' are the normals to e and e'
- ▶ (Some basic linear algebra, quantifier elimination...)
- ▶ Predecessor:

$$\lambda = \frac{\lambda' - B_{e,e'}}{A_{e,e'}}$$

- ▶ Moreover: if $e \in \text{In}(P)$ and $e' \in \text{Out}(P)$ then $A_{e,e'} > 0$

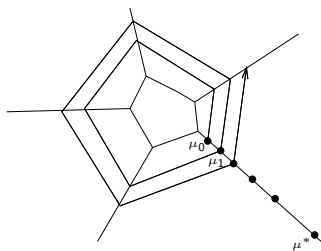
Signature Successor Function

- ▶ A cyclic signature: a sequence $\sigma = e_1, \dots, e_k$ of edges s.t. $e_1 = e_k$



- ▶ The function f_σ from e_1 to itself represents the effect on a point going through a cycle (Poincaré map)
- ▶ In our case it is linear $f_\sigma(\lambda) = A_\sigma \lambda + B_\sigma$ (composition of linear partial functions)
- ▶ $A_\sigma = A_{e_1, e_2} \cdot A_{e_2, e_3} \cdots A_{e_{k-1}, e_k}$
- ▶ $B_\sigma = (\cdots ((B_{e_1, e_2} \cdot A_{e_2, e_3} + B_{e_2, e_3}) \cdot A_{e_3, e_4} + B_{e_3, e_4}) \cdots) \cdot A_{e_{k-1}, e_k} + B_{e_{k-1}, e_k}$

Intersections of a Spiral and an Edge



- ▶ $\mu_{i+1} = A_\sigma \cdot \mu_i + B_\sigma$
- ▶ $\mu_n = \begin{cases} \mu_0 + B_\sigma \cdot n & \text{if } A_\sigma = 1 \\ \mu_0 \cdot A_\sigma^n + B_\sigma \cdot \frac{A_\sigma^n - 1}{A_\sigma - 1} & \text{otherwise} \end{cases}$
- ▶ We can compute $\mu^* = \lim_{n \rightarrow \infty} \mu_n$

The Limit of the Sequence

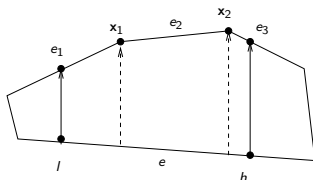
<i>Case</i>	<i>Limit</i>
$A_\sigma = 1, B_\sigma = 0$	μ_0
$A_\sigma = 1, B_\sigma > 0$	∞
$A_\sigma = 1, B_\sigma < 0$	$-\infty$
$A_\sigma < 1$	$\frac{B_\sigma}{1 - A_\sigma}$
$A_\sigma > 1, \mu_0 = \frac{B_\sigma}{1 - A_\sigma}$	μ_0
$A_\sigma > 1, \mu_0 > \frac{B_\sigma}{1 - A_\sigma}$	∞
$A_\sigma > 1, \mu_0 < \frac{B_\sigma}{1 - A_\sigma}$	$-\infty$

Main Positive Result

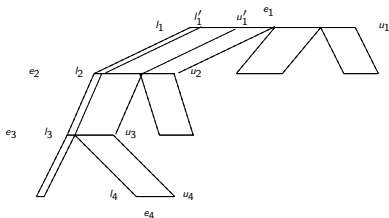
- ▶ An algorithm for deciding **Reach**($\mathcal{H}, \mathbf{x}, \mathbf{x}'$):
- ▶ Start “simulating” forward from \mathbf{x}
- ▶ When you encounter a cycle, compute its limit points on all edges and determine whether it is the ultimate cycle (limits on each edge stays inside edge range)
- ▶ If not, continue simulating until you leave it (in a finite number of iterations)
- ▶ If it is the ultimate cycle, and \mathbf{x}' is beyond the limit, the answer is “no”
- ▶ If \mathbf{x}' is before the limit then continue simulation until you reach \mathbf{x}' (“yes”) or bypass it (“no”)

Region-to-Region Reachability (Sketch)

- ▶ Can be reduced to edge-to-edge reachability
- ▶ An entry edge interval splits into finitely many exits edges



- ▶ Can build a successor tree and compute a limit along each branch

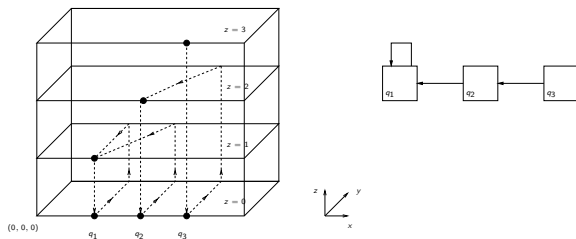


Can we go to Higher Dimensions?

- ▶ On one hand: calculating successors can be generalized to higher dimensions (more book-keeping though)
- ▶ On the other: no Jordan theorem so trajectories are not necessarily ultimately-periodic (Chaos et co.)
- ▶ We show undecidability for 3 dimensions by showing that PCDs can simulate any TM (2PDA) and hence deciding reachability for PCDs solves the halting problem
- ▶ Interesting “model of computation”

Simulation of Finite-State Automata

- ▶ Every finite deterministic automaton can be simulated by a 3-dimensional PCD system



Region	Defining conditions	$\mathbf{c} = (\dot{x}, \dot{y}, \dot{z})$
F	$(z = 0) \wedge (y < 1)$	$(0, 1, 0)$
U_{ij}	$(x = i) \wedge (y = 1) \wedge (z < j)$	$(0, 0, 1)$
B_{ij}	$(z = j) \wedge (x + (j - i)y = j) \wedge (y > 0)$	$(j - i, -1, 0)$
D	$(z > 0) \wedge (y = 0)$	$(0, 0, -1)$

- ▶ Regions U_{ij} and B_{ij} are defined for every i, j such that $\delta(q_i) = q_j$

Push-down Automata (PDA)

- ▶ Pushdown stack: an element of Σ^*0^ω .
- ▶ Two operations:

$$\text{PUSH: } \Sigma \times \Sigma^\omega \rightarrow \Sigma^\omega \quad \text{POP: } \Sigma^\omega \rightarrow \Sigma \times \Sigma^\omega$$

$$\text{PUSH}(v, S) = v \cdot S \quad \text{POP}(v \cdot S) = (v, S)$$

- ▶ PDA: an infinite transition system $\mathcal{A} = (Q \times \Sigma^*0^\omega, \delta)$
- ▶ Q is finite and δ is defined using a finite collection of statements of one of the following forms:

q_i : $S := \text{PUSH}(v, S)$;
GOTO q_j

q_i : $(v, S) := \text{POP}(S)$;
IF $v = 0$ GOTO q_{i_0} ;
...
IF $v = k - 1$ GOTO $q_{i_{k-1}}$;

Encoding Stacks into $[0, 1]$

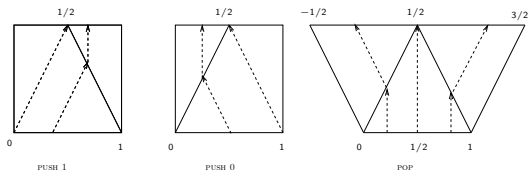
- ▶ Contents of a stack $S = s_1 s_2 \dots$ where s_1 is the top of the stack
- ▶ Encoding using k -ary representation $r : \Sigma^\omega \rightarrow [0, 1]$

$$r(S) = \sum_{i=1}^{\infty} s_i k^{-i}$$

- ▶ Stack operations have arithmetic counterparts:

$$\begin{aligned} S' = \text{PUSH}(v, S) & \quad \text{iff} \quad r(S') = (r(S) + v)/k \\ (S', v) = \text{POP}(S) & \quad \text{iff} \quad r(S') = kr(S) - v \end{aligned}$$

Building Blocks for the Simulation, $k = 2$ and $\Sigma = \{0, 1\}$



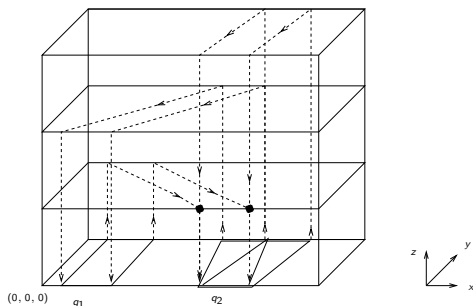
- ▶ A trajectory starting at $\mathbf{x} = (x, 0)$, $x \in [0, 1]$ and ending at $\mathbf{x}' = (x', 1)$ satisfies:
- ▶ $x' = (x + 1)/2$ (PUSH 1), $x' = x/2$ (PUSH 0) and $x' = 2x - 1/2$ (POP)
- ▶ In other words, $x = r(S)$ at the “input port” ($y = 0$) of an element, then $x' = r(S')$ at the “output port” ($y = 1$) where S' is the operation outcome.
- ▶ The POP element has two output ports which are selected according to the value of the top element popped

Simulation of PDAs by PCDs

- ▶ Put the appropriate element for each state and connect via “bands” that “carry” the stack value
- ▶ A PCD for the PDA defined by:

$q_1 : S := \text{PUSH}(1, S); \text{GOTO } q_2;$

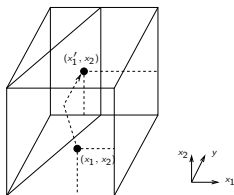
$q_2 : (v, S) := \text{POP}(S); \text{If } v = 1 \text{ THEN GOTO } q_2 \text{ ELSE GOTO } q_1$



- ▶ Every PDA can be simulated by a 3-dimensional PCD system

Simulating 2PDAs

- ▶ Automata with 2 push-down stacks can simulate Turing machines
- ▶ We can represent the configuration of two stacks by a point in $[0, 1]^2$ and build the corresponding gadgets, e.g. $\text{PUSH}(S_1, 0)$



- ▶ Hence a straightforward realization of 2PDA in 4 dimensions
- ▶ With some considerable effort we can squeeze everything into 3 dimensions and conclude:
- ▶ The reachability problem for PCD systems in 3 dimensions is undecidable

Theoreticians go Wild

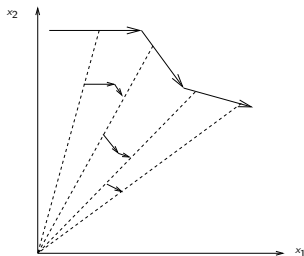
- ▶ Arithmetical hierarchy: the classes $\Sigma_1, \Sigma_2, \dots$ and Π_1, Π_2, \dots of sets of integers defined inductively:
- ▶ Σ_1 consists of sets $P \subseteq \mathbf{N}$ such that there is a Turing machine that halts on an input n iff $n \in P$
- ▶ The class Π_i consists of all the sets P such that $\bar{P} \in \Sigma_i$
- ▶ Σ_{i+1} is the class of all sets P defined as $P = \{n : \exists m \langle m, n \rangle \in P'\}$ for some $P' \in \Pi_i$, where $\langle \rangle$ is some computable pairing function
- ▶ The arithmetical hierarchy is infinite, satisfying the strict inclusions $\Pi_i \subset \Sigma_{i+1}$ and $\Sigma_i \subset \Pi_{i+1}$
- ▶ We show (with the help of Zeno paradox) how all the arithmetical hierarchy can be realized by PCDs

Recognition by PCDs

- ▶ PCD recognizer: $\hat{\mathcal{H}} = (\mathbb{R}^d, f, l, r, \mathbf{x}^A, \mathbf{x}^R)$, $\mathcal{H} = (\mathbb{R}^d, f)$ is a PCD
- ▶ $l = [0, 1] \times \{0\}^{d-1}$ is a one-dimensional subset of X (the “input port”)
- ▶ $r : \mathbf{N} \rightarrow [0, 1] \cap \mathcal{Q}$ is a recursive injective coding function
- ▶ $\mathbf{x}^A, \mathbf{x}^R \in \mathbb{R}^d - l$ are two distinct points (accepting and rejecting states)
- ▶ We assume that $f(\mathbf{x}^A) = f(\mathbf{x}^R) = 0$
- ▶ $\hat{\mathcal{H}}$ semi-recognizes $P \subseteq \mathbf{N}$ iff for every n , the trajectory starting at $(r(n), 0, \dots, 0)$ can continue forever and it eventually reaches \mathbf{x}^A iff $n \in P$
- ▶ We say that $\hat{\mathcal{H}}$ (fully) recognizes P when, in addition, this trajectory reaches \mathbf{x}^R iff $n \notin P$
- ▶ Previous result: every Σ_1 set P is semi-recognized by some 3-dimensional bounded PCD

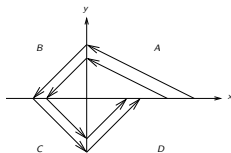
Principal Lammata

- ▶ From a PCD that semi-recognizes P one can construct a (higher-dimensional) PCD that recognizes P
- ▶ From a PCD that recognizes P one can construct:
 1. a PCD that semi-recognizes $\{x : \exists y \langle x, y \rangle \in P\}$
 2. a PCD that recognizes \overline{P} .
- ▶ The last two are relatively-easy and trivial (respectively)
- ▶ The main idea of the first:

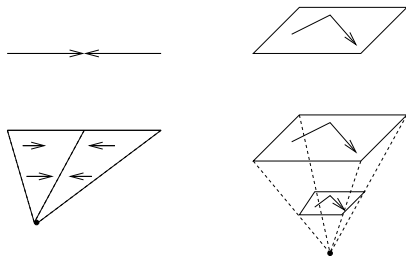


Gadgets used in the Construction

- ▶ Division by 2:



- ▶ Projectivisation:



- ▶ Corollary: PCDs can realize the whole arithmetical hierarchy

Credits and Follow-ups

- ▶ Decidability : OM and A. Pnueli, Reachability Analysis of Planar Multi-Linear Systems, 1993
- ▶ Generalized by Asarin, Pace, Schneider and Yovine to planar differential inclusions (and implemented)
- ▶ Undecidability: E. Asarin and OM, On some Relations between Dynamical Systems and Transition Systems, 1994
- ▶ Numerous papers on decidability boundaries for linear hybrid automata (Henzinger et al)
- ▶ Some small open problems remain, e.g. M. Mahfoudh, B. Krogh and OM, On Control with Bounded Computational Resources, 2002
- ▶ Higher undecidability: E. Asarin and OM, Achilles and the Tortoise Climbing Up the Arithmetical Hierarchy, 1995
- ▶ Studied extensively by O. Bournez

So What?

- ▶ Beyond the nice intellectual exercise (and a warm-up for those whose geometry and linear algebra are, at best, rusty) the results are rather disappointing
- ▶ Even for these systems, whose continuous dynamics is trivial we cannot answer anything
- ▶ How will we cope with “real” dynamics?
- ▶ We are asking the wrong questions, inspired by our discrete verification background
- ▶ In the continuous world having precise/exact answers is an oxymoron
- ▶ We should ask weaker, approximate questions on stronger systems with real differential equations