Model Checking, Hybrid Automata, and Systems Biology

Carla Piazza[1]

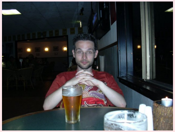
[1]Department of Mathematics and Computer Science,
University of Udine,
Udine, Italy

Part of our Group

Alberto Policriti

Bud Mishra

DIMI Udine, IGA Udine
NYU New York, DMI Trieste, DISA Udine



Alberto Casagrande

Giannina Vizzotto

Outline

- Model Checking and Temporal Logics
- Hybrid Automata
- Hybrid Automata in Systems Biology
- Semi-Algebraic Hybrid Automata
- Discrete vs Continuous
- Conclusions

Please, be patient with my English

Model Checking in Computer Science

We have an hardware/software (reactive concurrent) system

We want to check whether the system satisfies some specifications or not

Model Checking in Computer Science

We have an hardware/software (reactive concurrent) system
We want to check whether the system satisfies some specifications or not

H/S System S	$\Rightarrow$	Kripke Structure $\mathcal{M}$
Specification F	$\Rightarrow$	Temporal Logic Formula ψ

Model Checking in Computer Science

We have an hardware/software (reactive concurrent) system
We want to check whether the system satisfies some specifications or not

H/S System S	$\Rightarrow$	Kripke Structure $\mathcal{M}$
Specification F	$\Rightarrow$	Temporal Logic Formula ψ

Now the problem is:

$$\mathcal{M} \models \psi$$

i.e., does the model $\mathcal{M}$ satisfies the formula ψ?

Model Checking

The problem $\mathcal{M} \models \psi$ looks very easy

Model Checking

The problem $\mathcal{M} \models \psi$ looks very easy

We need to solve it efficiently

Model Checking

The problem $\mathcal{M} \models \psi$ looks very easy

We need to solve it efficiently

Let us look into the detail:

- $\mathcal{M}$ is a graph with labels on nodes and edges
- ψ is a formula talking about properties of paths

Model Checking

The problem $\mathcal{M} \models \psi$ looks very easy

We need to solve it efficiently

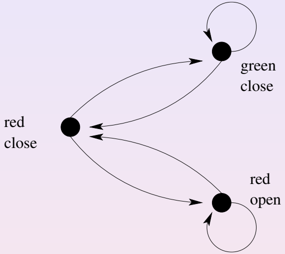
Let us look into the detail:

- $\mathcal{M}$ is a graph with labels on nodes and edges
- ψ is a formula talking about properties of paths

Can we solve it in polynomial time? And in linear time?

What about space complexity?

Example: Railroad Crossing



- We do not want green light for the train when the gate is open (safety)

$$AG\neg(green \wedge open)$$

- We do not want the train waiting forever (liveness)

$$red \rightarrow EF(green)$$

Temporal Logics

Definition (CTL)

Let $\mathcal{P}$ be a set of atomic propositions

- each $p \in \mathcal{P}$ is a formula
- if ψ_1 and ψ_2 are formulæ, then also $\psi_1 \wedge \psi_2$, $\neg\psi_1$, $AX\psi_1$, $EX\psi_1$, $AF\psi_1$, $EF\psi_1$, $AG\psi_1$, $EG\psi_1$, $A(\psi_1 U \psi_2)$, $E(\psi_1 U \psi_2)$ are formulæ

Temporal Logics

Definition (CTL)

Let $\mathcal{P}$ be a set of atomic propositions

- each $p \in \mathcal{P}$ is a formula
- if ψ_1 and ψ_2 are formulæ, then also $\psi_1 \wedge \psi_2$, $\neg\psi_1$, $AX\psi_1$, $EX\psi_1$, $AF\psi_1$, $EF\psi_1$, $AG\psi_1$, $EG\psi_1$, $A(\psi_1 U \psi_2)$, $E(\psi_1 U \psi_2)$ are formulæ

- path and state quantifiers are alternated

Temporal Logics

Definition (CTL)

Let $\mathcal{P}$ be a set of atomic propositions

- each $p \in \mathcal{P}$ is a formula
- if ψ_1 and ψ_2 are formulæ, then also $\psi_1 \wedge \psi_2$, $\neg\psi_1$, $AX\psi_1$, $EX\psi_1$, $AF\psi_1$, $EF\psi_1$, $AG\psi_1$, $EG\psi_1$, $A(\psi_1 U \psi_2)$, $E(\psi_1 U \psi_2)$ are formulæ

- path and state quantifiers are alternated
- the model checking problem can be solved in linear time, $O(|\psi| * |\mathcal{M}|)$ (thanks to a fix-point computation and Tarjan algorithm for strongly connected components)

Temporal Logics

Definition (CTL)

Let $\mathcal{P}$ be a set of atomic propositions

- each $p \in \mathcal{P}$ is a formula
- if ψ_1 and ψ_2 are formulæ, then also $\psi_1 \wedge \psi_2$, $\neg\psi_1$, $AX\psi_1$, $EX\psi_1$, $AF\psi_1$, $EF\psi_1$, $AG\psi_1$, $EG\psi_1$, $A(\psi_1 U \psi_2)$, $E(\psi_1 U \psi_2)$ are formulæ

- path and state quantifiers are alternated
- the model checking problem can be solved in linear time, $O(|\psi| * |\mathcal{M}|)$ (thanks to a fix-point computation and Tarjan algorithm for strongly connected components)
- it is not so easy for other logics, e.g., LTL and CTL* are P-space complete

State Explosion Problem

We have to handle $\mathcal{M}$

State Explosion Problem

We have to handle $\mathcal{M}$

The number of states (nodes) of $\mathcal{M}$ grows exponentially w.r.t. the number of interacting components

State Explosion Problem

We have to handle $\mathcal{M}$

The number of states (nodes) of $\mathcal{M}$ grows exponentially w.r.t. the number of interacting components

Many solutions have been proposed:

- Symbolic Model Checking
- Abstract Model Checking
- On-the-fly Model Checking

allowing to successfully apply Model Checking to real cases

Some References

- Manna and Pnueli. Temporal Logics. 1981
- Clarke, Emerson, and Sistla. Quielle and Sifakis. Transition Systems. 1983
- Efficient Algorithms are studied for many logics.
- State Explosion Problem is an obstacle in the applications.
- Mc Millan, Clarke, et al.. Symbolic Model Checking. 1993
- Dams, Gerth, and Grumberg. Abstract Model Checking. 1996
- Henzinger. Model Checking on Hybrid Systems. 1997

Model Checking and Systems Biology

We can use Kripke Structures for representing Pathways, or Experimental Traces. . .

. . . and Temporal Logics for asking biological questions:

- is state *s* reachable?
- is the system always oscillating? (see Repressilator)

See, e.g., Fages, Mishra

- State Explosion Problem becomes dramatic
- How can we model continuous variables?
  Do they really exist?

Hybrid Systems

Many real systems have a double nature. They:

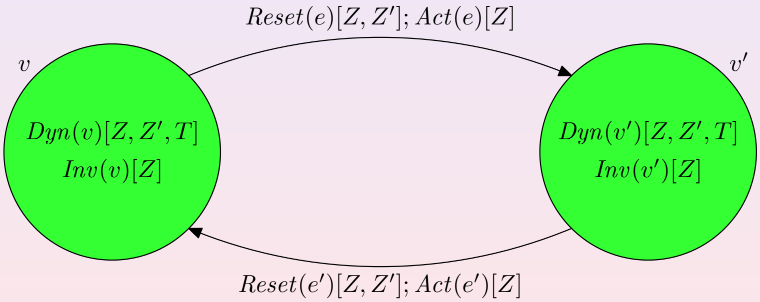
- evolve in a continuous way
- are ruled by a discrete system

We call such systems hybrid systems and we can formalize them using hybrid automata

Hybrid Automata - Intuitively

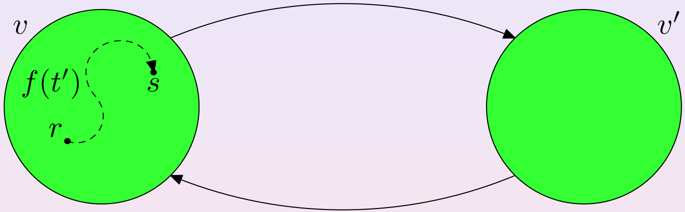
A hybrid automaton *H* is

a finite state automaton with continuous variables *Z*



A state is a pair $\langle v, r \rangle$ where r is an evaluation for Z

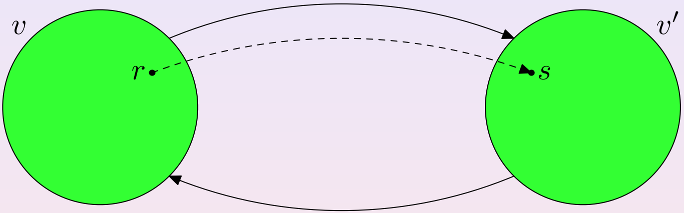
Hybrid Automata - Semantics



Definition (Continuous Transition)

$\langle v, r \rangle \xrightarrow{t}_C \langle v, s \rangle \iff$ there exists a continuous $f : \mathbb{R}^+ \mapsto \mathbb{R}^k$ such that $r = f(0)$, $s = f(t)$, and for each $t' \in [0, t]$ the formulæ $Inv(v)[f(t')]$ and $Dyn(v)[r, f(t'), t']$ hold

Hybrid Automata - Semantics



Definition (Discrete Transition)

$\langle v, r\rangle \xrightarrow{\langle v,\lambda,v'\rangle}_D \langle v', s\rangle \iff$ $\langle v, \lambda, v'\rangle \in \mathcal{E}$ and $Inv(v)[r]$, $Act(\langle v, \lambda, v'\rangle)[r]$, $Reset(\langle v, \lambda, v'\rangle)[r, s]$, and $Inv(v')[s]$ hold

Hybrid Automata – Escherichia

Escherichia coli is a bacterium detecting the food concentration through a set of receptors

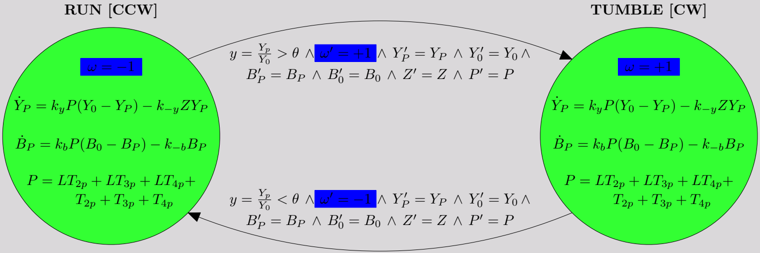
It responds in one of two ways:

- "RUNS" – moves in a straight line by moving its flagella counterclockwise (CCW)
- "TUMBLES" – randomly changes its heading by moving its flagella clockwise (CW)

In our example, we ignore any stochastic effect by modeling it deterministically

Hybrid Automata – Escherichia

Example (*E. Coli* Model)



ω is the angular velocity that takes discrete values $+1$ for CW and -1 for CCW

Hybrid Automata Issues

- Decidability. There are many undecidability results even on basic classes of hybrid automata. Why? What can we do?
- Complexity. Hybrid Automata involve notions coming from different areas Control Theory, Analysis, Computational Algebra, Logic, ... Are we exploiting all their powerful instruments?
- Compositionality. We would like to combine many hybrid automata representing different systems running in parallel. How can we do it?
- Precision. Hybrid automata have a semantics with infinite precision. Is this realistic in (biological) applications?

Which is Your Point of View?

- The world is dense
- The world is discrete

Which is Your Point of View?

- The world is dense

  $(\mathbb{R}, +, *, <, 0, 1)$ first-order theory is decidable

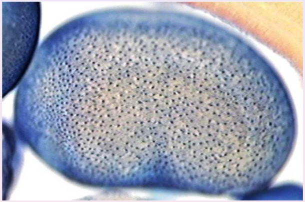
- The world is discrete

  Diophantine equations are undecidable

What about their interplay?

Delta-Notch

Delta and Notch are proteins involved in cell differentiation (see, e.g., Collier et al., Ghosh et al.)

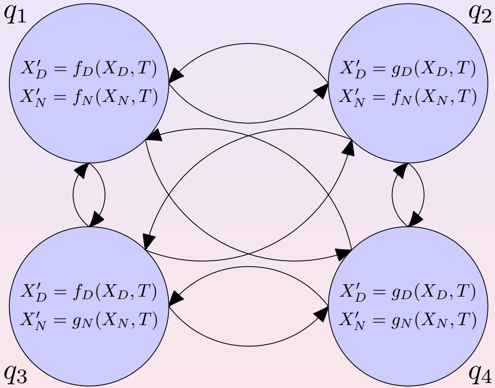


Notch production is triggered by high Delta levels in neighboring cells

Delta production is triggered by low Notch concentrations in the same cell

High Delta levels lead to differentiation

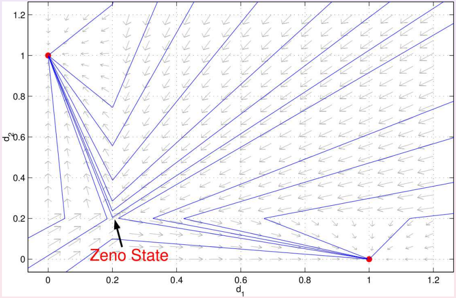
Delta-Notch: Single Cell Automaton



f_D and f_N increase Delta and Notch, g_D and g_N decrease Delta and Notch, respectively

Delta-Notch: Two Cells Automaton

It is the Cartesian product of two “single cell” automata



The Zeno state can occur only in the case of
two cells with identical initial concentrations

Verification

Question

Can we automatically verify hybrid automata?

Let us start from the basic case of Reachability
Assume that Continuous/Discrete transitions are computable

Verification

Question

Can we automatically verify hybrid automata?

Let us start from the basic case of Reachability
Assume that Continuous/Discrete transitions are computable

Naive_Reachability$(H, Initial\_set)$

$Old \leftarrow \emptyset$

$New \leftarrow Initial\_set$

while $New \neq Old$ **do**

$\quad Old \leftarrow New$

$\quad New \leftarrow Discrete\_Reach(H, Continuous\_Reach(H, Old))$

return Old

Bounded Sets and Undecidability

Even if the invariants are bounded, reachability is undecidable

Proof sketch

Encode two-counter machine by exploiting density:

- each counter value, n, is represented in a continuous variable by the value 2^{-n}
- each control function is mimed by a particular location

Where is the Problem?

Keeping in mind our examples:

Question "Meaning"

What is the meaning of these undecidability results?

Question "Decidability"

Can we avoid undecidability by adding some *natural* hypothesis to the semantics?

Undecidability in Real Systems

Undecidability in our models comes from ...

- infinite domains: unbounded invariants
- dense domains: the "trick" n as 2^{-n}

Undecidability in Real Systems

Undecidability in our models comes from ...

- infinite domains: unbounded invariants
- dense domains: the "trick" n as 2^{-n}

But which real system does involve ...

- unbounded quantities?
- infinite precision?

Unboundedness and density abstract discrete large quantities

Dense vs Discrete - Intuition

What if we do not really want to completely abandon dense domains?

We need to introduce a finite level of precision in bounded dense domains, we can distinguish two sets only if they differ of “at least ϵ”

Intuitively, we can see that something new has been reached only if a reasonable large set of new points has been discovered, i.e., we are myope

Finite Precision Semantics

Definition (ϵ-Semantics)

Let $\epsilon > 0$. For each formula ψ:

(ϵ) either $\{\!|\psi|\!\}_\epsilon = \emptyset$ or $\{\!|\psi|\!\}_\epsilon$ contains an ϵ-ball

($\cap$) $\{\!|\psi_1 \wedge \psi_2|\!\}_\epsilon \subseteq \{\!|\psi_1|\!\}_\epsilon \cap \{\!|\psi_2|\!\}_\epsilon$

($\cup$) $\{\!|\psi_1 \vee \psi_2|\!\}_\epsilon = \{\!|\psi_1|\!\}_\epsilon \cup \{\!|\psi_2|\!\}_\epsilon$

($\neg$) $\{\!|\psi|\!\}_\epsilon \cap \{\!|\neg\psi|\!\}_\epsilon = \emptyset$

It is a general framework: there exist many different ϵ-semantics

A Decidability Result

Theorem (Reachability Problem)

Using ϵ-semantics and assuming both bounded invariants and decidability for specification language, we have decidability of reachability problem for hybrid automata

See A. Casagrande, C. Piazza, and A. Policriti. Discreteness, Hybrid Automata, and Biology. WODES'08

A Decidability Result

Theorem (Reachability Problem)

Using ϵ-semantics and assuming both bounded invariants and decidability for specification language, we have decidability of reachability problem for hybrid automata

See A. Casagrande, C. Piazza, and A. Policriti. Discreteness, Hybrid Automata, and Biology. WODES'08

How can we ensure the decidability for specification language?

Semi-Algebraic Hybrid Automata

Definition (Semi-Algebraic Theory)

First-order polynomial formulæ over the reals $(\mathbb{R}, 0, 1, *, +, >)$

Example

$$\exists T \geq 0 (Z' = T^2 - T + Z \wedge 1 \leq Z \leq 2)$$

Definition

An hybrid automaton *H* is semi-algebraic if *Dyn*, *Inv*, *Reset*, and *Act* are semi-algebraic

Semi-Algebraic Automata and Decidability

Semi-algebraic formulæ allow us to reduce reachability to satisfiability of first-order formulæ over $(\mathbb{R}, 0, 1, *, +, >)$

Semi-Algebraic Automata and Decidability

Semi-algebraic formulæ allow us to
reduce reachability to satisfiability
of first-order formulæ over $(\mathbb{R}, 0, 1, *, +, >)$

First-order formulæ over $(\mathbb{R}, 0, 1, *, +, >)$ are decidable [Tarski]

Semi-Algebraic Automata and Decidability

Semi-algebraic formulæ allow us to
reduce reachability to satisfiability
of first-order formulæ over $(\mathbb{R}, 0, 1, *, +, >)$

First-order formulæ over $(\mathbb{R}, 0, 1, *, +, >)$ are decidable [Tarski]

May be reachability is decidable over Semi-algebraic automata
even with the standard infinite precision semantics?

Semi-Algebraic Automata and Decidability

Semi-algebraic formulæ allow us to
reduce reachability to satisfiability
of first-order formulæ over $(\mathbb{R}, 0, 1, *, +, >)$

First-order formulæ over $(\mathbb{R}, 0, 1, *, +, >)$ are decidable [Tarski]

May be reachability is decidable over Semi-algebraic automata
even with the standard infinite precision semantics?

No!

Semi-Algebraic Automata and (Un)Decidability

Reachability is reduced to:

$$\mathit{Reachable}[Z, Z'] \equiv \bigvee_{ph \in Ph} \exists T \geq 0 (\mathit{Reach}_{ph}[Z, Z', T])$$

where Ph is the set of all paths and $\mathit{Reach}_{ph}[Z, Z', T]$ means that Z reaches Z' in time T through ph

Semi-Algebraic Automata and (Un)Decidability

Reachability is reduced to:

$$Reachable[Z, Z'] \equiv \bigvee_{ph \in Ph} \exists T \geq 0 (Reach_{ph}[Z, Z', T])$$

where Ph is the set of all paths and $Reach_{ph}[Z, Z', T]$ means that Z reaches Z' in time T through ph

Ph is infinite!

Semi-Algebraic Automata and (Un)Decidability

Reachability is reduced to:

$$Reachable[Z, Z'] \equiv \bigvee_{ph \in Ph} \exists T \geq 0 (Reach_{ph}[Z, Z', T])$$

where Ph is the set of all paths and $Reach_{ph}[Z, Z', T]$ means that Z reaches Z' in time T through ph

Ph is infinite!

We need constraints on the resets and Selection theorems

See A. Casagrande, B. Mishra, C. Piazza, and A. Policriti. Inclusion Dynamics Hybrid Automata. Information and Computation, 2008

Composition of Hybrid Automata

We can define the Parallel Composition (cartesian product) of hybrid automata

Is reachability still decidable?

Yes!. . . Sometimes . . . To prove it we had to prove the decidability of linear systems of “Diophantine” equations with semi-algebraic coefficients:

- loops in the discrete structure of the automata give rise to integer variables
- the continuous dynamics produce the semi-algebraic coefficients

A. Casagrande, P. Corvaja, C. Piazza, and B. Mishra. Decidable Compositions of O-minimal Automata. ATVA’08

Conclusions

- I briefly presented:
  - Model Checking
  - Temporal Logics
  - Hybrid Automata
- Many interesting mathematical problems comes from the interplay between discrete and continuous components in hybrid automata
- I sketched two biological examples
- How do we construct hybrid automata from biological data?

Some Names

- Thomas A. Henzinger
- Rajeev Alur
- Claire Tomlin
- Ashish Tiwari
- François Fages