Model Checking, Hybrid Automata, and Systems Biology

Carla Piazza

1Department 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
We have an hardware/software (reactive concurrent) system. We want to check whether the system satisfies some specifications or not.
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 $M$

Specification $F$ $\Rightarrow$ Temporal Logic Formula $\psi$
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 $\psi$

Now the problem is:

$\mathcal{M} \models \psi$

i.e., does the model $\mathcal{M}$ satisfies the formula $\psi$?
The problem $\mathcal{M} \models \psi$ looks very easy
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
- $\psi$ 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
- $\psi$ 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)
  $AG\neg (green \land open)$

- 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 $\psi_1$ and $\psi_2$ are formulæ, then also $\psi_1 \land \psi_2$, $\neg \psi_1$, $AX\psi_1$, $EX\psi_1$, $AF\psi_1$, $EF\psi_1$, $AG\psi_1$, $EG\psi_1$, $A(\psi_1 U \psi_2)$, $E(\psi_1 U \psi_2)$ are formulæ
Definition (CTL)

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

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

- path and state quantifiers are alternated
Definition (CTL)

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

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

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

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

- each $p \in \mathcal{P}$ is a formula
- if $\psi_1$ and $\psi_2$ are formulæ, then also $\psi_1 \land \psi_2$, $\neg \psi_1$, $A\!X\!\psi_1$, $E\!X\!\psi_1$, $A\!F\!\psi_1$, $E\!F\!\psi_1$, $A\!G\!\psi_1$, $E\!G\!\psi_1$, $A(\psi_1 U \psi_2)$, $E(\psi_1 U \psi_2)$ are formulæ

- path and state quantifiers are alternated
- the model checking problem can be solved in linear time, $O(|\psi| \ast |\mathcal{M}|)$ (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
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.
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

- 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
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?
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**
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 $Z$.
Definition (Continuous Transition)

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

\[
\langle v, r \rangle \xrightarrow{D} \langle v', s \rangle \iff \langle v, \lambda, v' \rangle \in \mathcal{E} \quad \text{and} \quad \begin{align*}
Inv(v)[r], & \quad Act(\langle v, \lambda, v' \rangle)[r], \\
Reset(\langle v, \lambda, v' \rangle)[r, s], & \quad \text{and} \\
Inv(v')[s] & \text{hold}
\end{align*}
\]
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)

\( \omega \) 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**
  
  \((\mathbb{R}, +, \times, <, 0, 1)\) 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.
$f_D$ 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*)

\[
\begin{align*}
\text{Old} & \leftarrow \emptyset \\
\text{New} & \leftarrow \text{Initial_set} \\
\textbf{while} & \text{ New} \neq \text{ Old} \textbf{ do} \\
& \quad \text{Old} \leftarrow \text{New} \\
& \quad \text{New} \leftarrow \text{Discrete_Reach}(H, \text{Continuous_Reach}(H, \text{Old})) \\
\textbf{return} & \text{ Old}
\end{align*}
\]
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 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
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 $\epsilon$”

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 $\psi$:

1. Either $\{|\psi|\}_\epsilon = \emptyset$ or $\{|\psi|\}_\epsilon$ contains an $\epsilon$-ball
2. $\{|\psi_1 \land \psi_2|\}_\epsilon \subseteq \{|\psi_1|\}_\epsilon \cap \{|\psi_2|\}_\epsilon$
3. $\{|\psi_1 \lor \psi_2|\}_\epsilon = \{|\psi_1|\}_\epsilon \cup \{|\psi_2|\}_\epsilon$
4. $\{|\psi|\}_\epsilon \cap \{|\neg \psi|\}_\epsilon = \emptyset$

It is a general framework: there exist many different $\epsilon$-semantics
Theorem (Reachability Problem)

Using $\epsilon$-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 $\epsilon$-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
\[ \exists T \geq 0 (Z' = T^2 - T + Z \land 1 \leq Z \leq 2) \]

Definition
An hybrid automaton \(H\) is semi-algebraic if \(Dyn, Inv, Reset,\) and \(Act\) are semi-algebraic
Semi-algebraic formulæ allow us to reduce reachability to satisfiability of first-order formulæ over $(\mathbb{R}, 0, 1, *, +, >)$.
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 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 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!
Reachability is reduced to:

$$\text{Reachable}[Z, Z'] \equiv \bigvee_{\forall \leq 0 (\text{Reach}_{\phi}(Z, Z', T))}$$

where $\phi$ is the set of all paths and $\text{Reach}_{\phi}(Z, Z', T)$ means that $Z$ reaches $Z'$ in time $T$ through $\phi$.
Reachability is reduced to:

\[
\text{Reachable}[Z, Z'] \equiv \bigvee_{\phi \in \Phi} \exists T \geq 0(\text{Reach}_{\phi}[Z, Z', T])
\]

where \( \Phi \) is the set of all paths and \( \text{Reach}_{\phi}[Z, Z', T] \) means that \( Z \) reaches \( Z' \) in time \( T \) through \( \phi \)

\( \Phi \) is infinite!
Reachability is reduced to:

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

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

\( Ph \) is infinite!

We need constraints on the resets and Selection theorems

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
I briefly presented:
- Model Checking
- Temporal Logics
- Hybrid Automata

Many interesting mathematical problems come 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