Assume-Guarantee verification of Hybrid Systems in ARIADNE

Davide Bresolin and Tiziano Villa

University of Verona

Games 2009
Udine, Italy
Outline

1. Introduction to Hybrid Systems
2. The software package ARIADNE
3. Assume-guarantee reasoning in ARIADNE
4. Conclusions
Outline

1. Introduction to Hybrid Systems
2. The software package ARIADNE
3. Assume-guarantee reasoning in ARIADNE
4. Conclusions
Hybrid Systems

Many real systems have a double nature:

- they evolve in a continuous way;
- they are controlled by a discrete system.

How to model them?

Hybrid Systems/Automata
Hybrid Automata: Definition

Definition (Hybrid Automaton, Alur et al. 1992)

A hybrid automaton is a tuple $H = \langle \mathcal{V}, \mathcal{E}, \mathbb{R}^k, \text{Inv}, \text{Dyn}, \text{Act}, \text{Reset} \rangle$:

1. $\langle \mathcal{V}, \mathcal{E} \rangle$ is a finite directed graph; the vertexes, $\mathcal{V}$, are called locations or control modes, and the directed edges, $\mathcal{E}$, are called control switches;

2. Each location $v \in \mathcal{V}$ is labeled by the predicate $\text{Inv}(v)$ on the set $\mathbb{R}^k$ and the transitive relation $\text{Dyn}(v)$ on $\mathbb{R}^k \times \mathbb{R}^k \times \mathbb{R}^k$;

3. Each edge $e \in \mathcal{E}$ is labeled by the predicate $\text{Act}(e)$ on $\mathbb{R}^k$ and the relation $\text{Reset}(e)$ on $\mathbb{R}^k \times \mathbb{R}^k$. 
A state of an hybrid automaton is a pair \((v, r)\) where \(v\) is a discrete location and \(r\) is a point in \(\mathbb{R}^k\).

**Hybrid Automaton = Finite Automaton + Continuous Evolution**

Time flows when the automaton stays in a location:

- \(H\) evolves from \(r\) to \(s\) in time \(t\) when \(\text{Dyn}(v)[r, s, t]\);
- in location \(v\), \(r\) must satisfy \(\text{Inv}(v)[r]\);
- \(H\) can cross a transition \(e\) only if \(\text{Act}(e)[r]\);
- when \(H\) crosses \(e\), \(\text{Reset}(e)[r, s]\).
An example: the watertank

Let us consider a simple control problem consisting of controlling the water level in a cylindric tank equipped with an inlet pipe at the top and an outlet pipe at the bottom (see Figure 2.1). The outlet flow depends on the water level while the inlet flow is controlled by a valve whose position is regulated by a controller receiving the measurement of the water level by an appropriate sensor.

Outlet flow $F_{\text{out}}$ depends on the water level.

Inlet flow $F_{\text{in}}$ is controlled by the valve position.

The controller senses the water level and sends the appropriate commands to the valve.

**Control Problem**

Keep the water level between two given thresholds.
The watertank automaton

\[ \dot{x}(t) = -\lambda x(t) + \alpha(t)f(p(t)) \]
\[ \dot{\alpha}(t) = 1/T \]
\[ 0 < x(t) < h + \delta \]
\[ 0 \leq \alpha(t) \leq 1 \]

\[ \dot{x}(t) = 0 \]
\[ \dot{\alpha}(t) = 0 \]
\[ x = H \wedge u > \lambda H \]
\[ x \leq l + \delta \]
\[ x \geq h - \delta \]
\[ l - \delta < x(t) < H \]
\[ \alpha(t) = 0 \]

\[ \dot{x}(t) = -\lambda x(t) + \alpha(t)f(p(t)) \]
\[ \dot{\alpha}(t) = -1/T \]
\[ l - \delta < x(t) < H \]
\[ 0 \leq \alpha(t) \leq 1 \]

\[ \dot{x}(t) = 0 \]
\[ \dot{\alpha}(t) = -1/T \]
\[ x = H \wedge u > \lambda H \]
\[ x = H \wedge u \leq \lambda H \]
\[ 0 \leq \alpha(t) \leq 1 \]

\[ x = H \wedge u > \lambda H \]
\[ x \geq h - \delta \]
\[ x \leq l + \delta \]
\[ l - \delta < x(t) < H \]
\[ \alpha(t) = 1 \]

Davide Bresolin and Tiziano Villa (University of Verona)
Assume-Guarantee verification of Hybrid Systems in ARIADNE
Games09 8 / 25
Evolution of the watertank
Reachability Problem

Reachability

Given an hybrid automaton $H$ and two sets $S$ and $T$, is there any $s \in S$ and $t \in T$ such that there exists a trajectory of $H$ from $s$ to $t$?

The reachability problem for Hybrid Automata is undecidable (Alur et al. 1995).

Can I solve the problem, at least in some cases?

- Restrict to special classes of Hybrid Automata (Timed Automata, Rectangular Automata, …)
- Use approximation techniques to obtain an approximation of the reachable set.
Reachability Problem

Reachability

Given an hybrid automaton $H$ and two sets $S$ and $T$, is there any $s \in S$ and $t \in T$ such that there exists a trajectory of $H$ from $s$ to $t$?

The reachability problem for Hybrid Automata is undecidable (Alur et al. 1995).

Can I solve the problem, at least in some cases?

- Restrict to special classes of Hybrid Automata (Timed Automata, Rectangular Automata, …)
- Use approximation techniques to obtain an approximation of the reachable set.
Reachability Problem

Reachability

Given an hybrid automaton $H$ and two sets $S$ and $T$, is there any $s \in S$ and $t \in T$ such that there exists a trajectory of $H$ from $s$ to $t$?

The reachability problem for Hybrid Automata is undecidable (Alur et al. 1995).

Can I solve the problem, at least in some cases?

- Restrict to special classes of Hybrid Automata (Timed Automata, Rectangular Automata, . . . )
- Use approximation techniques to obtain an approximation of the reachable set.
Outline

1. Introduction to Hybrid Systems
2. The software package ARIADNE
3. Assume-guarantee reasoning in ARIADNE
4. Conclusions
Developed by a joint team including CWI, the University of Verona, the University of Udine and the company PARADES (Rome).

Based on a rigorous mathematical semantics for the numerical analysis of continuous and hybrid systems.

The computational kernel is written using a mix of generic and polymorphic programming strategies resulting in a highly efficient, modular and extensible framework.

Released as an open source distribution.
Subsets of $\mathbb{R}^n$ are approximated by finite unions of basic sets:
- intervals, simplices, cuboids, parallelotopes, zonotopes, polytopes, spheres and ellipsoids

Finite unions of basic sets of a given type are called *denotable sets*. 
Approximating regions

Approximating $S$ with $A$

1. **Inner approximation:** $S$ strictly contains $A$.
2. **Outer approximation:** $S$ is strictly contained in $A$.
3. **$\epsilon$-lower approximation:** every point of $A$ is at distance less than $\epsilon$ from a point of $S$.

- Inner approximation is used for specification of systems properties.
- Outer and $\epsilon$-lower approximation are used for computing evolution.
Given an hybrid automaton $H$, an initial set $I$ and a time $t$, ARIADNE can compute:

- an outer approximation of the states reached by $H$ starting from $I$ up to time $t$.

- for a given $\varepsilon > 0$, an $\varepsilon$-lower approximation of the states reached by $H$ starting from $I$ up to time $t$. 

Davide Bresolin and Tiziano Villa (University of Verona) 
Assume-Guarantee verification of Hybrid Systems in ARIADNE 
Games09 15 / 25
Outline

1. Introduction to Hybrid Systems
2. The software package ARIADNE
3. Assume-guarantee reasoning in ARIADNE
4. Conclusions
Assume-guarantee system specification

- The system is specified as a set of components.
- Every component is annotated with a pair $(A, G)$ of assumptions and guarantees.
- The requirements of the whole system are decomposed into a set of simpler requirements that, if satisfied, guarantees that the overall requirements are satisfied.
Let $C$ be a component of the system, annotated with assumptions $A$ and guarantees $G$. With ARIA\textsc{DNE} we can verify whether the component $C$ respects the guarantees or not (with some limitations).

- Represent the component by an hybrid automata $H$ with inputs and outputs;
- Assumptions $A$ are represented by hybrid automata $H_A$ that specify the possible inputs for $H$;
- Guarantees $G$ specify the possible outputs $Y$ of the automata;

This is a reachability analysis problem:

\[
\text{Reach}(H\|A) \subseteq \text{Sat}(G)
\]
Safety checking by grid refinement

1. Compute an outer-approximation $O$ of $\text{Reach}(H\|H_A)$ using a grid of a given size.
2. If $O \subseteq \text{Sat}(G)$, the system is verified to be safe. Exit with success.
3. Otherwise, compute an $\varepsilon$-lower approximation $L_\varepsilon$ of $\text{Reach}(H\|H_A)$. The value of $\varepsilon$ depends on the size of the grid.
4. If there exists at least a point in $L_\varepsilon$ that is outside $\text{Sat}(G)$ by more than $\varepsilon$, the system is verified to be unsafe. Exit with failure.
5. Otherwise, set the grid to a finer size and restart from point 1.
Verifying the water tank

Safety property: the water level between 5.25 and 8.25 meters.

First iteration: grid $1/8 \times 1/80$.

Outer reach is not safe, try lower reach.
Verifying the water tank

Safety property: the water level between 5.25 and 8.25 meters.

First iteration: grid $1/8 \times 1/80$.

Lower reach is safe, refine grid.
Verifying the water tank

Safety property: the water level between 5.25 and 8.25 meters.

Second iteration: grid $1/16 \times 1/160$.

Outer reach is not safe, try lower reach.
Verifying the water tank

Safety property: the water level between 5.25 and 8.25 meters.

Second iteration: grid $1/16 \times 1/160$.

Lower reach is safe, refine grid.
Verifying the water tank

Safety property: the water level between 5.25 and 8.25 meters.

Third iteration: grid $1/32 \times 1/320$.

Outer reach is safe, system is proved safe.
### Definition

Given two components $C_1$ and $C_2$, with assumptions and guarantees $(A_1, G_1)$ and $(A_2, G_2)$, we say that $C_1$ dominates $C_2$ if and only if under weaker assumptions ($A_2 \subseteq A_1$), stronger promises are guaranteed ($G_1 \subseteq G_2$).

If this is the case, the component $C_2$ can be replaced with $C_1$ in the system without affecting the whole system behaviour.
Dominance checking by reachability analysis

1. Represent the two components by two hybrid automata $H_1$ and $H_2$ with inputs and outputs;

2. Assumptions $A_1$ and $A_2$ are represented by hybrid automata $H_{A_1}$ and $H_{A_2}$ that specify the possible inputs $U_1, U_2$ for the components;

3. Guarantees $G_1$ and $G_2$ specify the possible outputs $Y_1, Y_2$ of the automata;

4. $H_1$ dominates $H_2$ if and only if $Y_1 \subseteq Y_2$;

This is a reachability analysis problem:

$$\text{Reach}(H_{A_1} \parallel H_1)|_{Y_1} \subseteq \text{Reach}(H_{A_2} \parallel H_2)|_{Y_2}$$
Dominance checking in ARIADNE

The approximate reachability routines of ARIADNE can be used to test dominance of components:

1. Compute an $\varepsilon$-lower approximation $L_2^\varepsilon$ of $\text{Reach}(H_{A_2} \parallel H_2)|_{Y_2}$
2. Remove a border of size $\varepsilon$ from $L_2^\varepsilon$
3. Compute an outer approximation $O_1$ of $\text{Reach}(H_{A_1} \parallel H_1)|_{Y_1}$
4. If $O_1 \subseteq L_2^\varepsilon - \varepsilon$ then $\text{Reach}(H_{A_1} \parallel H_1)|_{Y_1} \subseteq \text{Reach}(H_{A_2} \parallel H_2)|_{Y_2}$ and thus $H_1$ dominates $H_2$
5. If not, we cannot say anything about $H_1$ and $H_2$, we retry with a finer approximation.
Outline

1. Introduction to Hybrid Systems
2. The software package ARIADNE
3. Assume-guarantee reasoning in ARIADNE
4. Conclusions
Conclusions

- **ARIADNE** can compute approximation of the reachable set of hybrid automata.

- It is currently used to verify complex systems using advanced verification strategies.

- **Future improvements:**
  - Add support for the analysis of networks of hybrid automata.
  - Provide input support for hybrid automata description languages.
  - Improve the verification and model checking capabilities.