Modeling and Analysis of Hybrid Systems

Introduction

Prof. Dr. Erika Ábrahám

Informatik 2 - Theory of Hybrid Systems
RWTH Aachen University

SS 2013

Organizational

Lecture:

- Tuesday 13:15-14:15 in 5056
- Friday 13:15-14:30 in 5056

Exercise:

- Tuesday 14:15-15:00 in 5056

Exam dates will be chosen by Doodle vote:

- 1st: 26.07.2013 09:45-12:15
  01.08.2013 13:45-16:15
  02.08.2013 09:45-12:15
- 2nd: 18.09.2013 15:45-18:15
  19.09.2013 10:45-13:15

Learning materials and contact persons

Learning materials available in L2P:

- Slides
- Lecture notes
- Video recordings
- Some research publications
- Exercise sheets, solutions

Lecture:

Erika Ábrahám
room: 2U07 (Hauptbau, basement), phone: 0241/80-21242
email: abraham@informatik.rwth-aachen.de

Exercise:

Xin Chen
room: 2U08 (Hauptbau, basement), phone: 0241/80-21243
email: xin.chen@informatik.rwth-aachen.de

Further information (topic, evaluations etc.):`http://www-i2.informatik.rwth-aachen.de/i2/hybrid_lecture/`

Contents

1. Hybrid systems
2. Modeling
3. Specification
4. Analysis

Contents

1 Hybrid systems

2 Modeling

3 Specification

4 Analysis

“Hybrid”

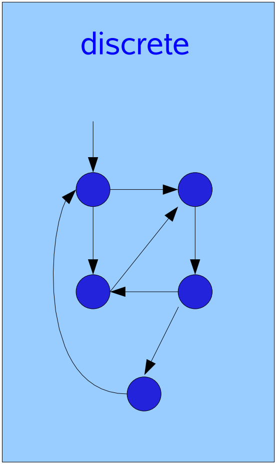
Wikipedia:

> “A hybrid is the combination of two or more different things, aimed at achieving a particular objective or goal.”

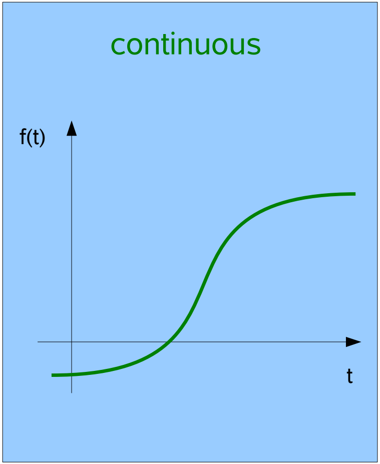
A hybrid rose

A hybrid car

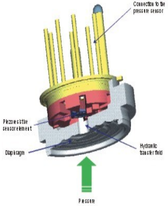
Hybrid in computer science

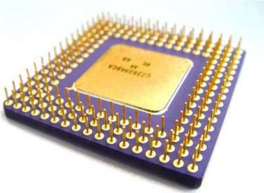


+



The discrete part


Connection to the pressure sensor
Piezoresistive sensorelement
Diaphragm
Hydraulic transfer fluid
Pressure



Combined with the continuous part

Example: Thermostat

Example: Thermostat

- Temperature x is controlled by switching a heater on and off
- x is regulated by a thermostat:
  - $17° \leq x \leq 18°$ $\rightsquigarrow$ "heater on"
  - $22° \leq x \leq 23°$ $\rightsquigarrow$ "heater off"

Example: Thermostat

- Temperature x is controlled by switching a heater on and off
- x is regulated by a thermostat:
  - $17^\circ \leq x \leq 18^\circ \rightsquigarrow$ "heater on"
  - $22^\circ \leq x \leq 23^\circ \rightsquigarrow$ "heater off"

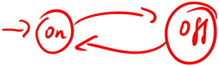
Continuous:	temperature
Discrete:	switching

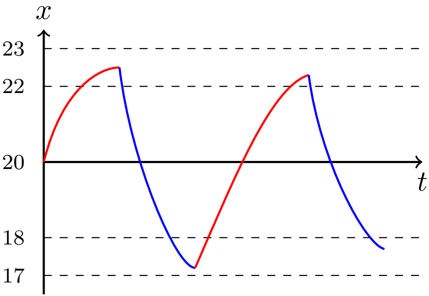
Example: Thermostat

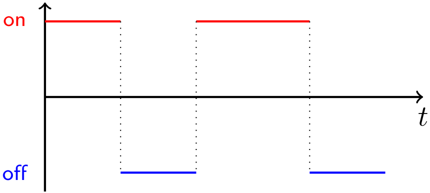
- Temperature x is controlled by switching a heater on and off
- x is regulated by a thermostat:
  - $17° \leq x \leq 18°$ ⇝ "heater on"
  - $22° \leq x \leq 23°$ ⇝ "heater off"

Continuous: temperature

Discrete: switching







Contents

1 Hybrid systems

2 Modeling

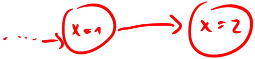
3 Specification

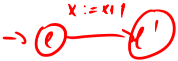
4 Analysis

Modeling

Modeling

ℓ: $x := x + 1$ ℓ':





- To be able to apply formal (mathematical) methods to a real system, we need a formal model of it.
- A model never exactly corresponds to the modeled real system.
- Abstract away unnecessary details.

Modeling

- To be able to apply formal (mathematical) methods to a real system, we need a formal model of it.
- A model never exactly corresponds to the modeled real system.
- Abstract away unnecessary details.
- What you probably already know: Kripke structures (state transition systems)

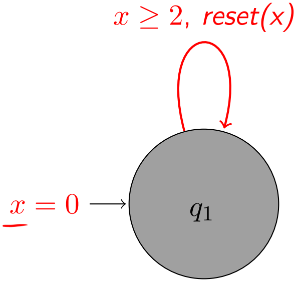
Modeling

- To be able to apply formal (mathematical) methods to a real system, we need a formal model of it.
- A model never exactly corresponds to the modeled real system.
- Abstract away unnecessary details.
- What you probably already know: Kripke structures (state transition systems)
- What you probably also know: Transition systems

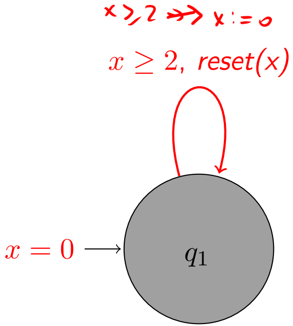
Modeling

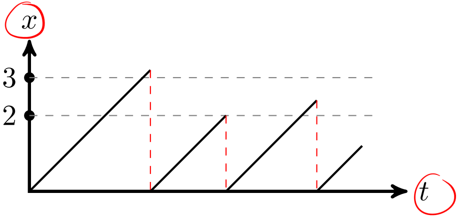
- To be able to apply formal (mathematical) methods to a real system, we need a formal model of it.
- A model never exactly corresponds to the modeled real system.
- Abstract away unnecessary details.
- What you probably already know: Kripke structures (state transition systems)
- What you probably also know: Transition systems
- What you perhaps know: Timed automata

Example: Timed automaton

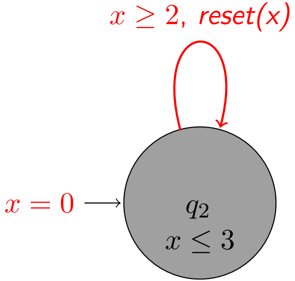


Example: Timed automaton

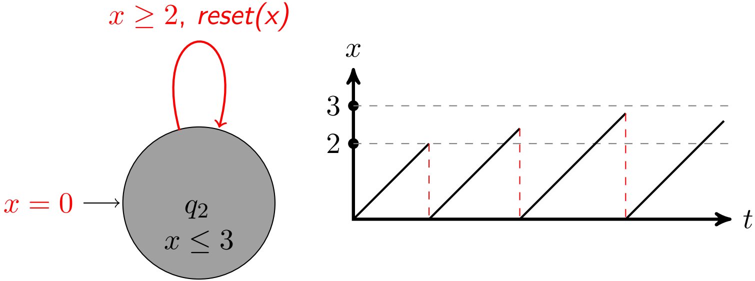




Example: Timed automaton



Example: Timed automaton



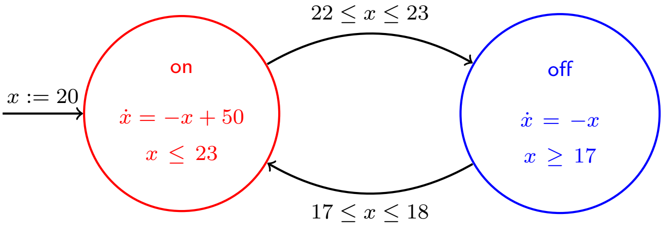
Modeling general hybrid systems: Hybrid automata

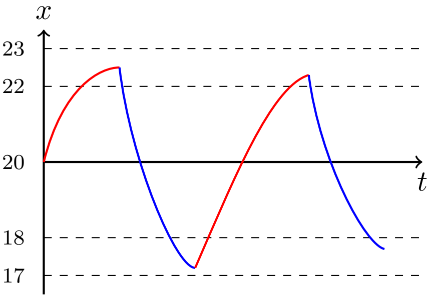
Modeling general hybrid systems: Hybrid automata

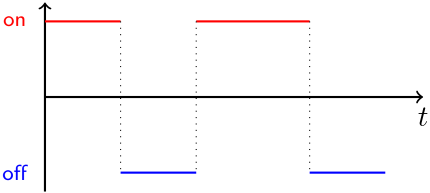
Let's take again the thermostat as an example.

Modeling general hybrid systems: Hybrid automata

Let's take again the thermostat as an example.







Contents

1 Hybrid systems

2 Modeling

3 Specification

4 Analysis

Logic

Logic

- We want to specify how a hybrid system is expected to behave.

Logic

- We want to specify how a hybrid system is expected to behave.
- We are especially interested in safety and liveness.

Logic

- We want to specify how a hybrid system is expected to behave.
- We are especially interested in safety and liveness.
- E.g., we can use the logic TCTL for specification.

Logic

- We want to specify how a hybrid system is expected to behave.
- We are especially interested in safety and liveness.
- E.g., we can use the logic TCTL for specification.
- In TCTL we can express properties like:

  "The temperature is always below $20°C$."

Logic

- We want to specify how a hybrid system is expected to behave.
- We are especially interested in safety and liveness.
- E.g., we can use the logic TCTL for specification.
- In TCTL we can express properties like:

"The temperature is always below $20°C$."

- Or

"If the temperature is above $20°C$ it will get below $20°C$ within 5 seconds."

Logic

- We want to specify how a hybrid system is expected to behave.
- We are especially interested in safety and liveness.
- E.g., we can use the logic TCTL for specification.
- In TCTL we can express properties like:

  "The temperature is always below $20°C$."
- Or

  "If the temperature is above $20°C$ it will get below $20°C$ within 5 seconds."
- Or

  "It is always the case that the temperature will somewhen in the future get above $20°C$."

Contents

1 Hybrid systems

2 Modeling

3 Specification

4 Analysis

The analysis of hybrid systems

The analysis of hybrid systems

- Assume we modeled a hybrid system as a hybrid automaton.
- Assume we specified a property of the system.
- Can we prove that the system satisfies the property?

The analysis of hybrid systems

- Assume we modeled a hybrid system as a hybrid automaton.
- Assume we specified a property of the system.
- Can we prove that the system satisfies the property?
- Well, it depends...

The analysis of hybrid systems

- Assume we modeled a hybrid system as a hybrid automaton.
- Assume we specified a property of the system.
- Can we prove that the system satisfies the property?
- Well, it depends...
- ...on the fact if the logic is decidable for the underlying modeling language.

The analysis of hybrid systems

- Assume we modeled a hybrid system as a hybrid automaton.
- Assume we specified a property of the system.
- Can we prove that the system satisfies the property?
- Well, it depends...
- ...on the fact if the logic is decidable for the underlying modeling language.
- We will see for which classes of hybrid automata the reachability question is decidable.

The analysis of hybrid systems

- Assume we modeled a hybrid system as a hybrid automaton.
- Assume we specified a property of the system.
- Can we prove that the system satisfies the property?
- Well, it depends...
- ...on the fact if the logic is decidable for the underlying modeling language.
- We will see for which classes of hybrid automata the reachability question is decidable.
- We will deal with
    - (unbounded) reachability for timed automata.
    - (unbounded) reachability for initialized rectangular automata.
    - bounded reachability for linear hybrid automata.
    - reachability approximation for general hybrid automata.

Method for timed automata: Finite abstraction

Method for timed automata: Finite abstraction

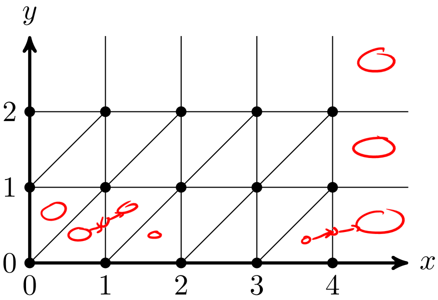
Constructive proof of decidability via finite abstraction:

Method for timed automata: Finite abstraction

Constructive proof of decidability via finite abstraction:

Method for timed automata: Finite abstraction

Constructive proof of decidability via finite abstraction:



Method for initialized rectangular automata: Transformation

Method for initialized rectangular automata: Transformation

Leading back the proof of decidability to a known problem:

Method for initialized rectangular automata: Transformation

$\dot{x} = 1$ $\dot{x} \in [2,3]$

Leading back the proof of decidability to a known problem:

Timed automaton

↑

Initialized stopwatch automaton

↑

Initialized singular automaton

↑

→ Initialized rectangular automaton

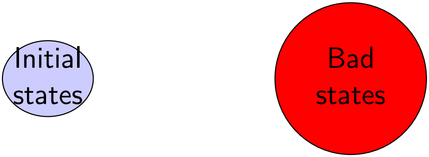
Method for linear hybrid automata: Fixedpoint computation

Method for linear hybrid automata: Fixedpoint computation

Forward reachability computation:

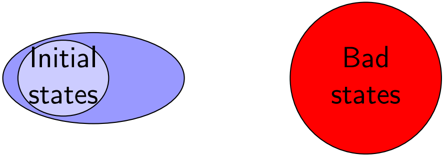
Method for linear hybrid automata: Fixedpoint computation

Forward reachability computation:



Method for linear hybrid automata: Fixedpoint computation

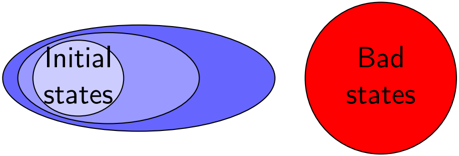
Forward reachability computation:



Method for linear hybrid automata: Fixedpoint computation

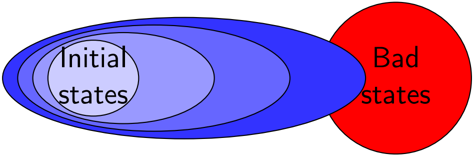
$$X \supseteq Reach(X)$$

Forward reachability computation:



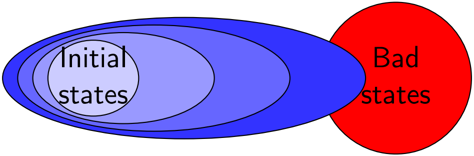
Method for linear hybrid automata: Fixedpoint computation

Forward reachability computation:



Method for linear hybrid automata: Fixedpoint computation

Forward reachability computation:



$x = 0$

$\downarrow$

$x \in \{0,1\}$

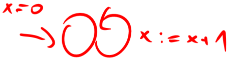
$\downarrow$

$x \in \{0,1,2\}$

$\vdots$

Note: the method is incomplete

$x=0 \rightarrow$ while (true) { $x := x+1$ }



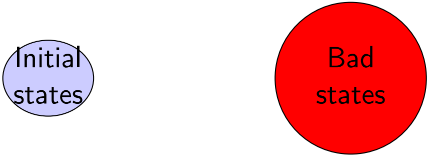
Method for linear hybrid automata: Fixedpoint computation

Method for linear hybrid automata: Fixedpoint computation

Backward reachability computation:

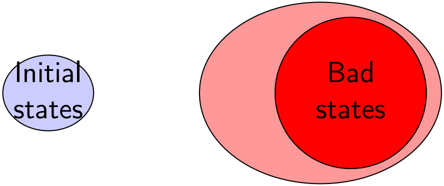
Method for linear hybrid automata: Fixedpoint computation

Backward reachability computation:



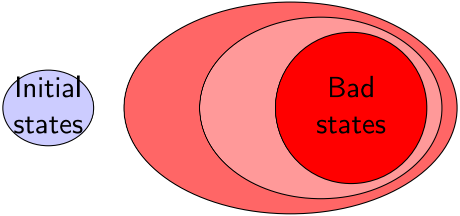
Method for linear hybrid automata: Fixedpoint computation

Backward reachability computation:



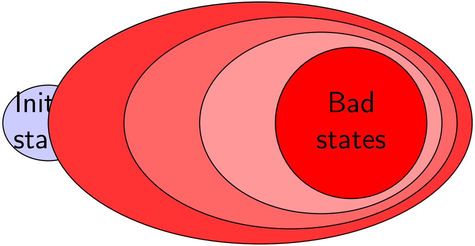
Method for linear hybrid automata: Fixedpoint computation

Backward reachability computation:



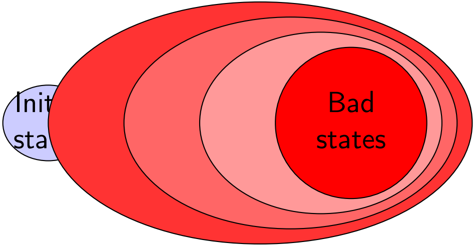
Method for linear hybrid automata: Fixedpoint computation

Backward reachability computation:



Method for linear hybrid automata: Fixedpoint computation

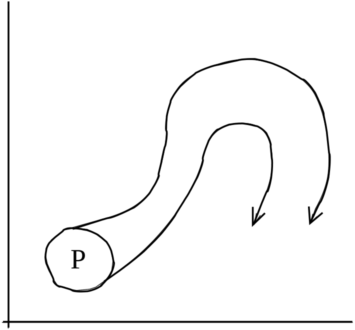
Backward reachability computation:

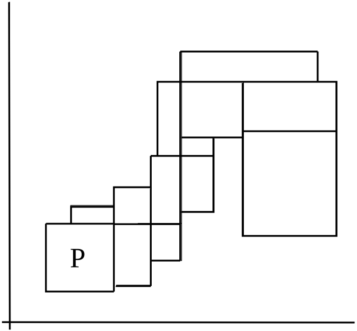


Note: also the backward method is incomplete

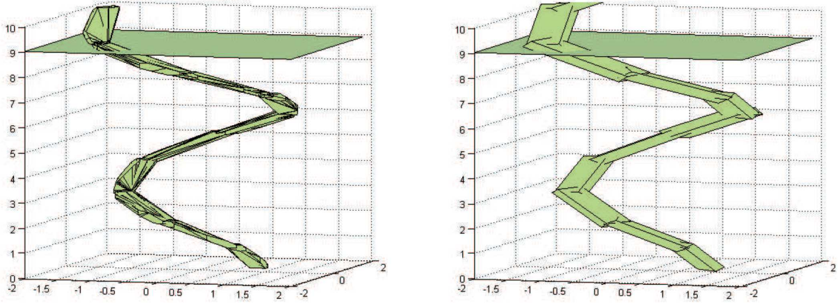
Method for hybrid automata: Approximation

Method for hybrid automata: Approximation

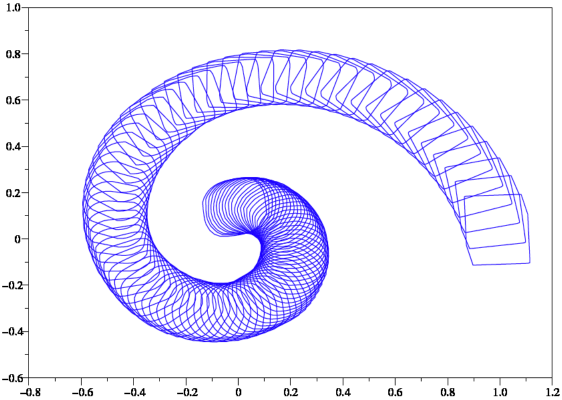




Polyhedra (left) and oriented rectangular hulls (right) in reachability computation



Zonotopes in reachability computation



1.0
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1.0
1.2