

# Modeling and Analysis of Hybrid Systems

## Introduction

Prof. Dr. Erika Ábrahám

Informatik 2 - Theory of Hybrid Systems  
RWTH Aachen University

SS 2015

## Lecture:

- Monday 15:15-16:00 in AH 3
- Tuesday 12:15-13:45 in 5056

## Exercise:

- Monday 16:00-16:45 in AH 3

## Exam dates:

- 1st: 27.07.2015 14:15-16:45
- 2nd: 16.09.2015 15:45-18:15

# Learning materials and contact persons

Learning materials available in L2P:

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

Lecture:

Erika Ábrahám

room: 4229 (E1, 2nd floor), phone: 0241/80-21242

email: abraham@informatik.rwth-aachen.de

Exercise:

Stefan Schupp

room: 4228 (E1, 2nd floor), phone: 0241/80-21243

email: stefan.schupp@informatik.rwth-aachen.de

Further information (topic, evaluations etc.):

<http://ths.rwth-aachen.de/teaching/ss15/>

[lecture-modelling-and-analysis-of-hybrid-systems/](#)

access: lanzarote.informatik.rwth-aachen.de:8080

username: tim-username

password: tim-password

1 Hybrid systems

2 Modeling

3 Specification

4 Analysis

**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

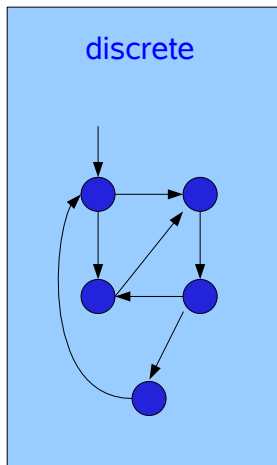




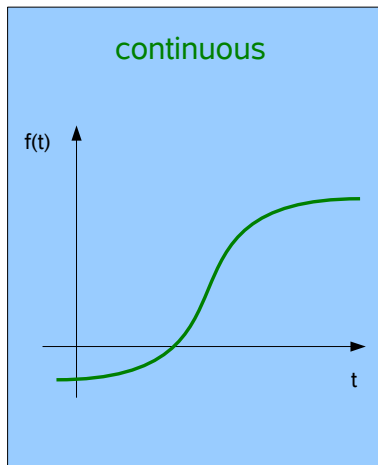
# A hybrid car



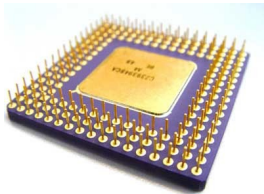
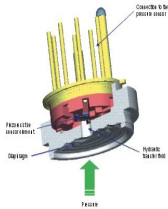
# Hybrid in computer science



+



# The discrete part



## 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:
  - $17^\circ \leq x \leq 18^\circ \rightsquigarrow$  "heater on"
  - $22^\circ \leq x \leq 23^\circ \rightsquigarrow$  "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

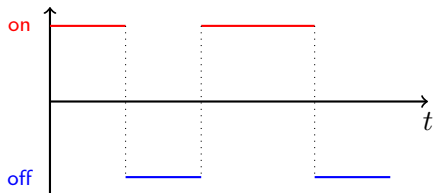
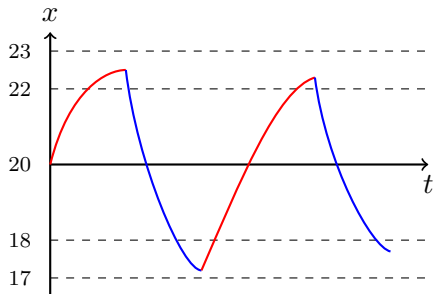
Discrete: switching

# Example: Thermostat

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

Continuous: temperature

Discrete: switching





1 Hybrid systems

**2 Modeling**

3 Specification

4 Analysis



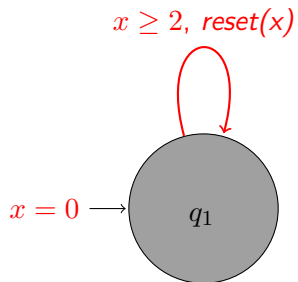
- To be able to apply formal (mathematical) methods to a real system, we need a formal **model** of it.
- A model never exactly corresponds to the modeled real system.
- **Abstract** away unnecessary details.

- 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)

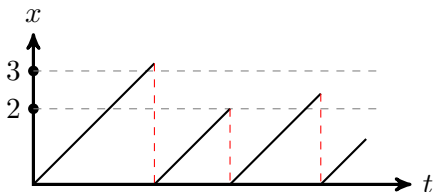
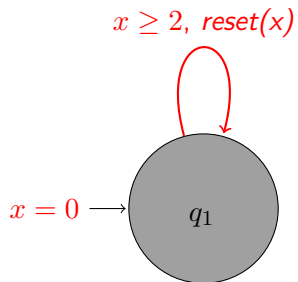
- To be able to apply formal (mathematical) methods to a real system, we need a formal **model** of it.
- A model never exactly corresponds to the modeled real system.
- **Abstract** away unnecessary details.
- What you probably already know: **Kripke structures** (state transition systems)
- What you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

- 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

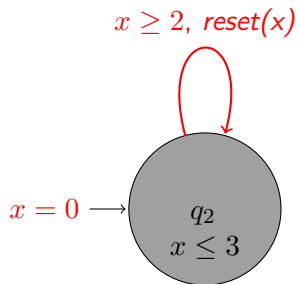


# Example: Timed automaton

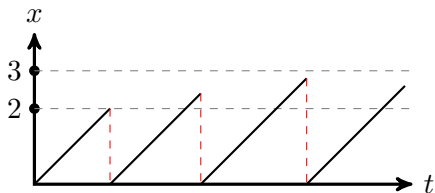
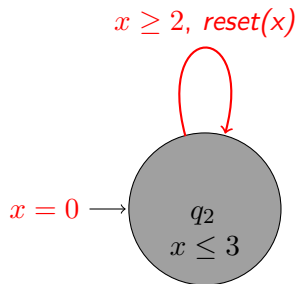




# Example: Timed automaton



# Example: Timed automaton



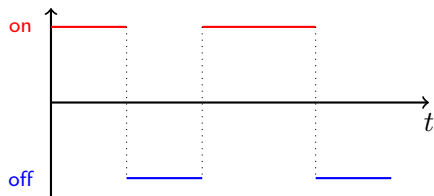
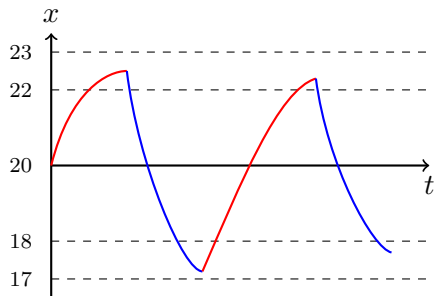
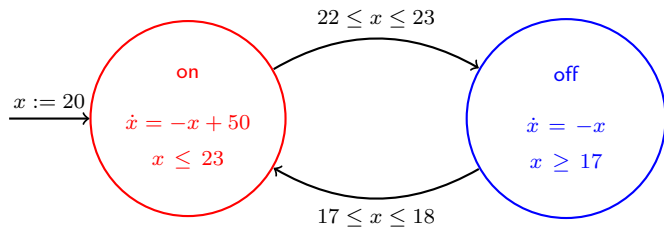
# Modeling general hybrid systems: Hybrid automata

# 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.



1 Hybrid systems

2 Modeling

**3 Specification**

4 Analysis



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



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

- 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.

- 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  $20^{\circ}\text{C}$ .”

- 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  $20^{\circ}C$ .”

- Or

“If the temperature is above  $20^{\circ}C$  it will get below  $20^{\circ}C$  within 5 seconds.”

- 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  $20^{\circ}C$ .”

- Or

“If the temperature is above  $20^{\circ}C$  it will get below  $20^{\circ}C$  within 5 seconds.”

- Or

“It is always the case that the temperature will somewhen in the future get above  $20^{\circ}C$ .”

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

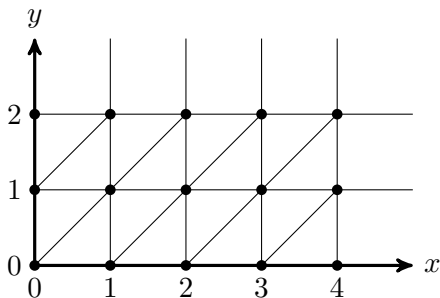
Constructive proof of decidability via finite abstraction:

Constructive proof of decidability via finite abstraction:



# Method for timed automata: Finite abstraction

Constructive proof of decidability via finite abstraction:

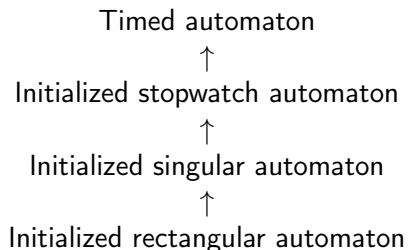




# Method for initialized rectangular automata: Transformation

Leading back the proof of decidability to a known problem:

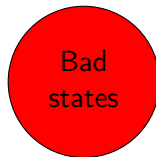
Leading back the proof of decidability to a known problem:



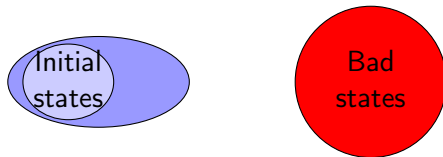
# Method for linear hybrid automata: Fixedpoint computation

Forward reachability computation:

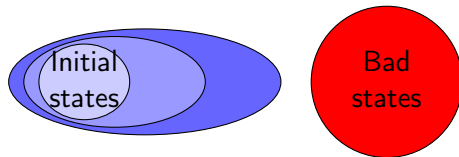
Forward reachability computation:



Forward reachability computation:

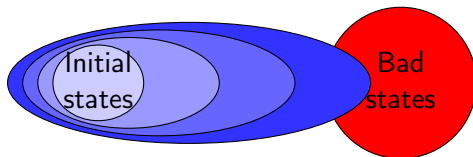


Forward reachability computation:

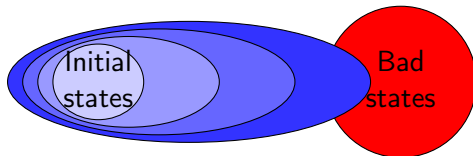




Forward reachability computation:



Forward reachability computation:

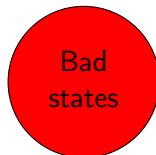


Note: the method is incomplete

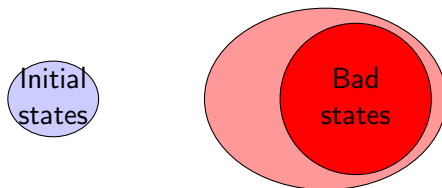
# Method for linear hybrid automata: Fixedpoint computation

Backward reachability computation:

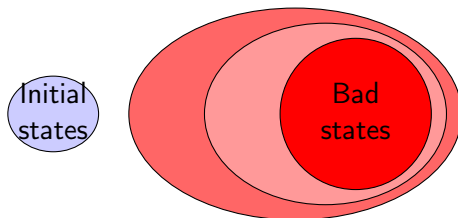
Backward reachability computation:



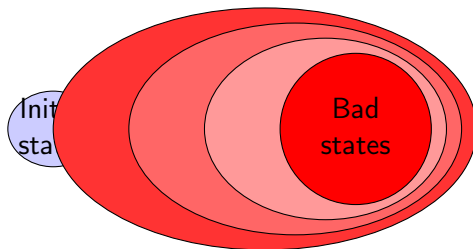
Backward reachability computation:



Backward reachability computation:

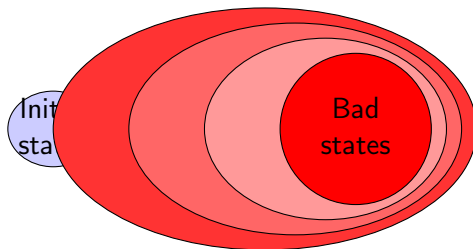


Backward reachability computation:





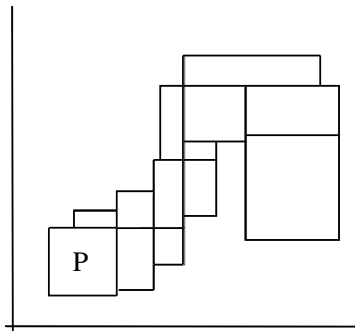
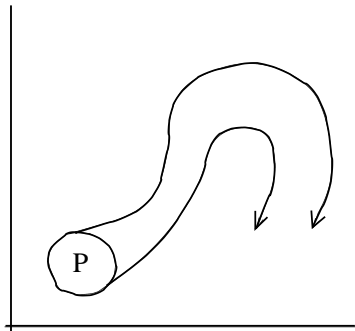
Backward reachability computation:



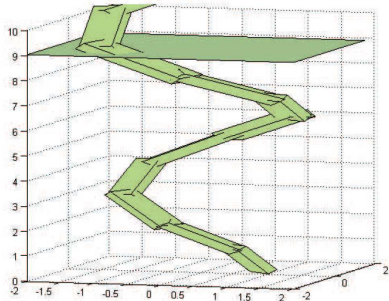
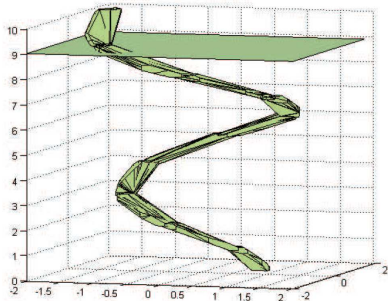
Note: also the backward method is incomplete

# Method for hybrid automata: Approximation

# Method for hybrid automata: Approximation



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



# Zonotopes in reachability computation

