Approximating continuous systems with timed automata (why)

1. Formal verification
    - Safety (can a bad state be reached?)
    - Liveness (can you reach a desirable state?)

Source: Approximating continuous systems by timed automata – Maler, Batt

Approximating continuous systems with timed automata (what)

Continuous dynamical system

↓

Timed Automata

Source: Approximating continuous systems by timed automata – Maler, Batt

Approximating continuous systems with timed automata (what)

- Continuous dynamical system
  - $S = (X, f)$
  - $X = X_1 \ldots X_n = [0,m) \times \ldots \times [0,m)$ in R^n
- Dynamics:
-
- Solution:
-

Source: Approximating continuous systems by timed automata – Maler, Batt

Approximating continuous systems with timed automata (what)

- Timed Automata

  A = (Q, C, I, Δ) = (states,clocks,invariants,transition)

  Δ = (old, guard, transformation, new)

-
  – A time step: $(q, \mathbf{z}) \xrightarrow{t} (q, \mathbf{z} + t)$, $t \in \mathbb{R}_+$ such that $\mathbf{z} + t$ satisfies I_q, and $\mathbf{z} + t$ is the result of adding t to clocks active in $\mathbf{z}$.

-
  – A discrete step: $(q, \mathbf{z}) \xrightarrow{\delta} (q', \mathbf{z}')$, for some transition $\delta = (q, g, \rho, q') \in \Delta$, such that $\mathbf{z}$ satisfies g and $\mathbf{z}'$ is the result of applying ρ to $\mathbf{z}$

Source: Approximating continuous systems by timed automata – Maler, Batt

Approximating continuous systems with timed automata (how)

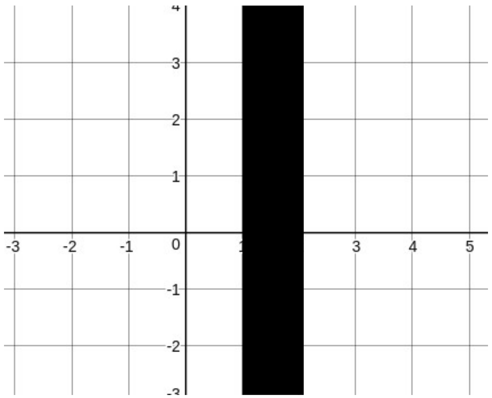
1. Indirect method
   - → transform original system into a model
   - → simpler class, easier verification
   - → decidable
2.

Source: Approximating continuous systems by timed automata – Maler, Batt

Approximating continuous systems with timed automata (how)

1.2. Partition state space into cells

-
- $V = V_1 \times \ldots \times V_n$ *where* $V_i = \{0, \ldots, m-1\}$
- *Cube:* $X_v = [v_1, v_1 + 1) \times \ldots \times [v_n, v_n + 1)$
- *Successor/predecessor:*
  - $\sigma^{+i}(\ldots v_i \ldots) = \sigma^{+i}(\ldots v_i + 1 \ldots)$
  - $\sigma^{-i}(\ldots v_i \ldots) = \sigma^{-i}(\ldots v_i - 1 \ldots)$
- *Common facet: (n-1) dimensional intersection of 2 cubes*
- *I-slice with r: set of cubes* $X_{i,r}$: $r <= x_i <= r+1$
-



2.

Approximating continuous systems with timed automata (how)

1.
2. 3. Define a transition between neighboring cells

Definition 4 (Abstraction by Automata). *The automaton* $\bar{\mathcal{A}} = (V, \bar{\delta})$ *is an abstraction of* $\mathcal{S}$ *if* $\bar{\delta}$ *consists of all pairs* $(v, \sigma^{+i}(v))$ *of cubes such that* f_i *admits a positive value on their common facet and all pairs* $(v, \sigma^{-}(v))$ *such that* f_i *admits a negative value on their common facet.*

Source: Approximating continuous systems by timed automata – Maler, Batt

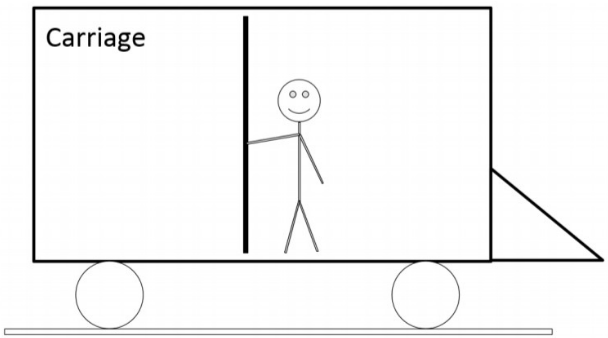
Approximating continuous systems with timed automata (how)

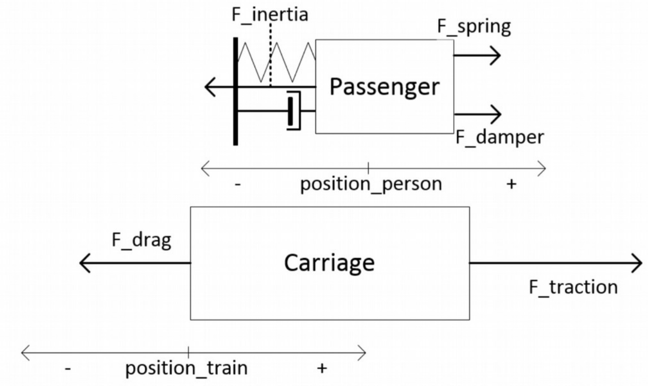
1.4. Add clocks (temporal logic)

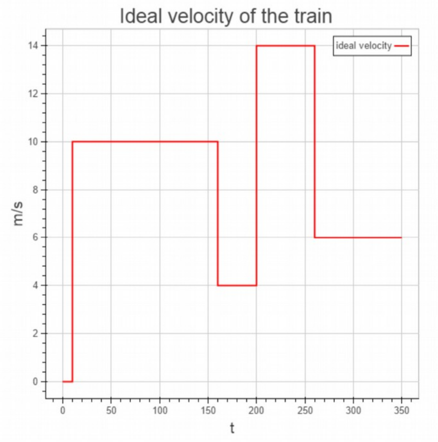
- 2 clocks per dimension
- One general clock

Source: Approximating continuous systems by timed automata – Maler, Batt

Safety example

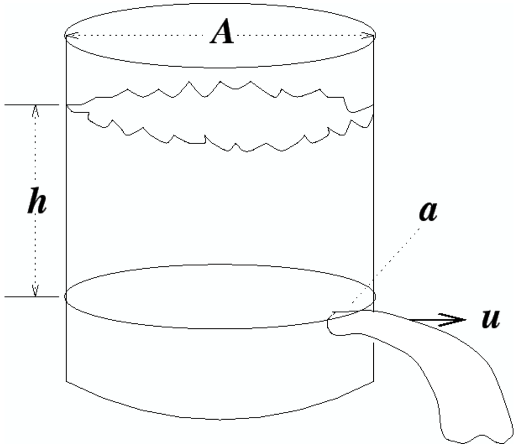






Example of a transformation

- Leaky bucket
  I.
  II.
  III.

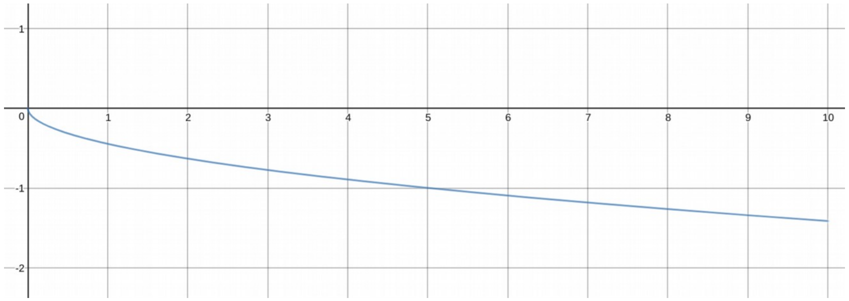


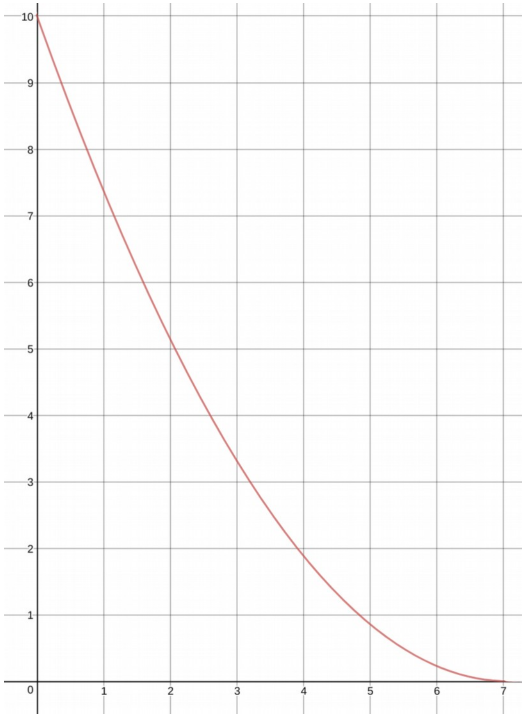
- Dynamical System
  I. X = R
  II.
- Solution
  –
  –
  –

Source: http://math.usu.edu/~powell/biomath/lb-02/lb-02.html

Example of a transformation

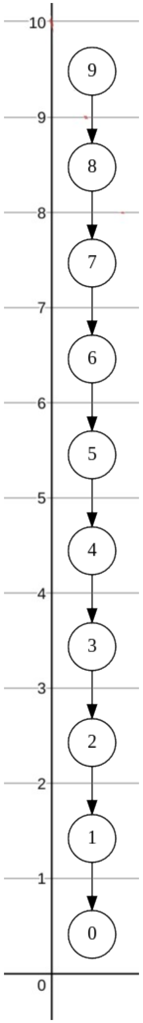
- Solution





Source: Approximating continuous systems by timed automata – Maler, Batt

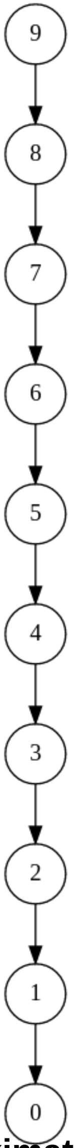
Example of a transformation



- One dimensional
- Cubes of the form:
- Cubes are lines
- Facets are points
- Slices are the cubes itself

Source: Approximating continuous systems by timed automata – Maler, Batt

Example of a transformation

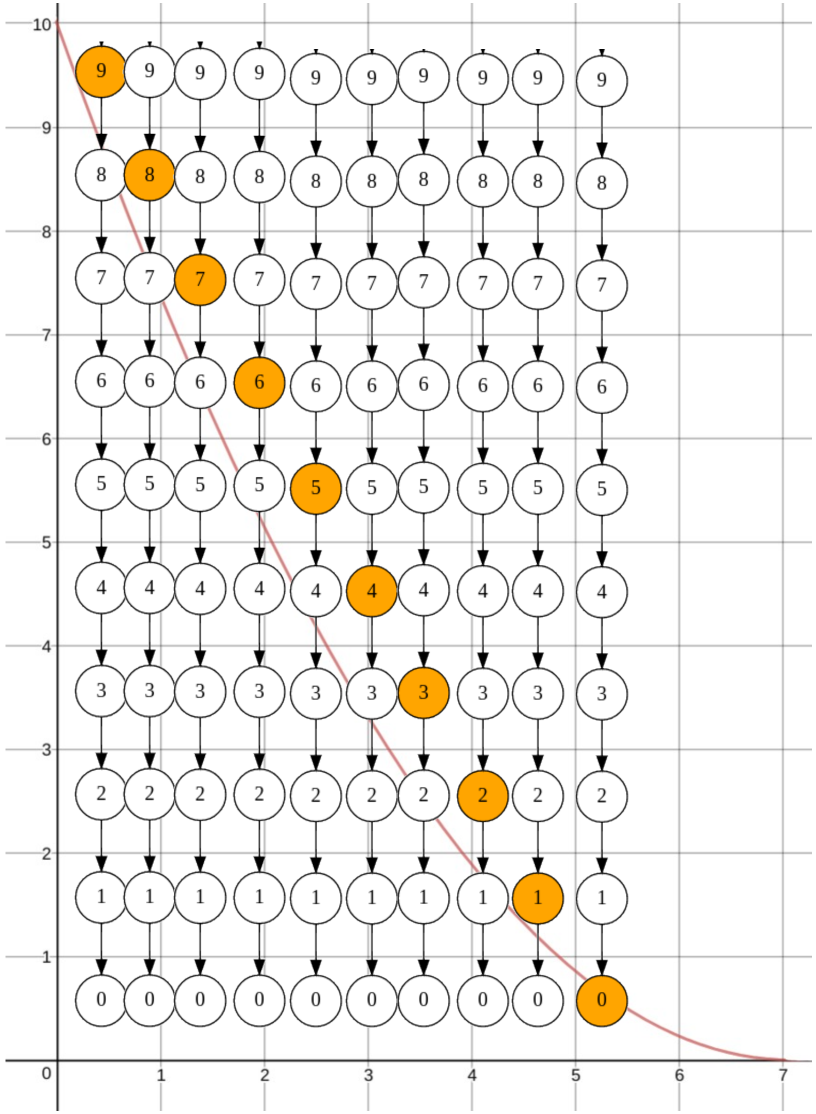


Transitions:

I. From A to B if the value of f < 0 on their common facet
II. From C to D if the value of f > 0 on their common facet

Source: Approximating continuous systems by timed automata – Maler, Batt

Example of a transformation



Stop here?

Source: Approximating continuous systems by timed automata – Maler, Batt

Example of a transformation

Stop here?

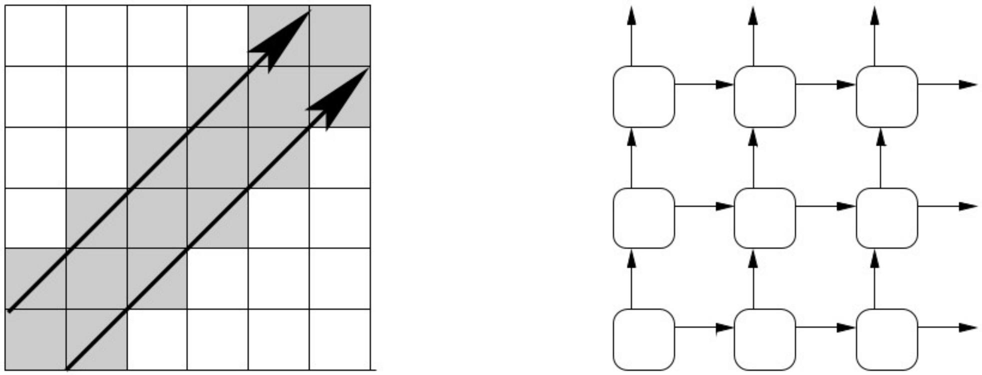
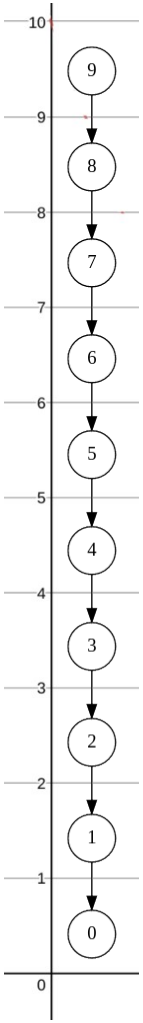


Fig. 1. (a): A simple continuous system with constant derivatives. The states reachable from the initial cube lie between the two arrows and their cube abstraction is shaded; (b) The automaton derived according to Definition 4 in which the whole state space is reachable.

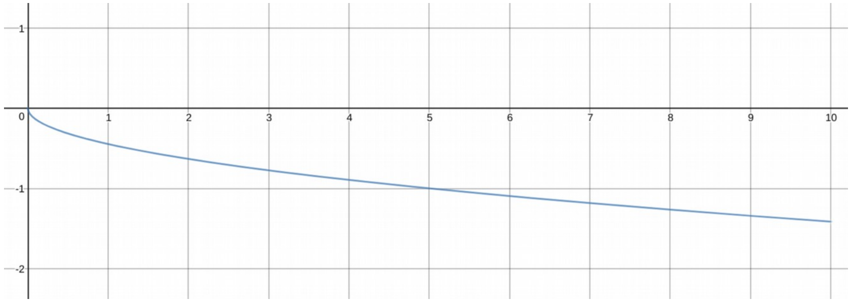
Example of a transformation



- Cannot stay in cube for more than: 1 / f_v

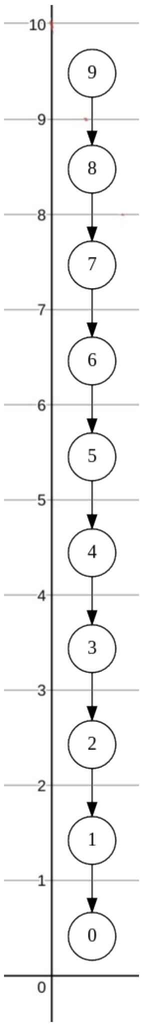
Assume f_{min} and f^{max} are the min and max derivates for a certain interval

-
- Cannot stay in slice for more than:
- $t_i^{max} = 1 / f^{max}$
  -
- Cannot leave slice in less time than:
  - $t_i^{max} = 1 / f_{min}$



Source: Approximating continuous systems by timed automata – Maler, Batt

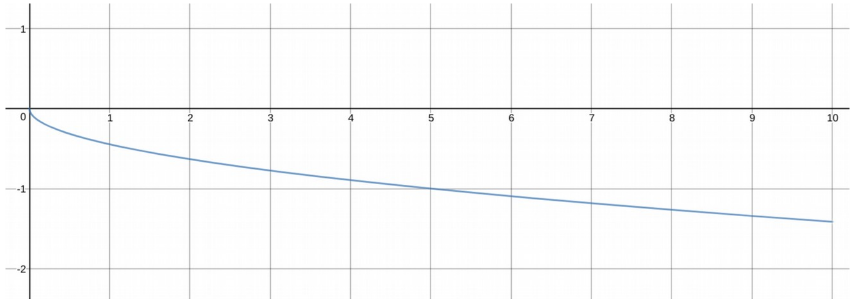
Example of a transformation



Extremal values in a cube:

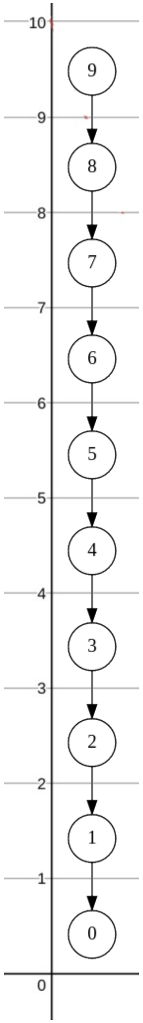
- Monotonic decreasing function
- Max value at top of cube
- Min value at bottom of cube
- Minimal absolute = min value

Extremal values in a slice = cube



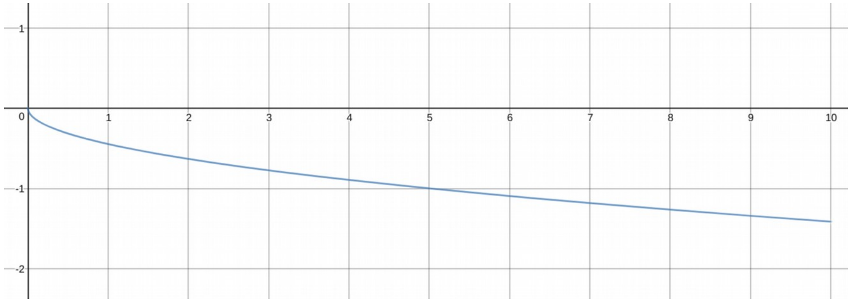
Source: Approximating continuous systems by timed automata – Maler, Batt

Example of a transformation



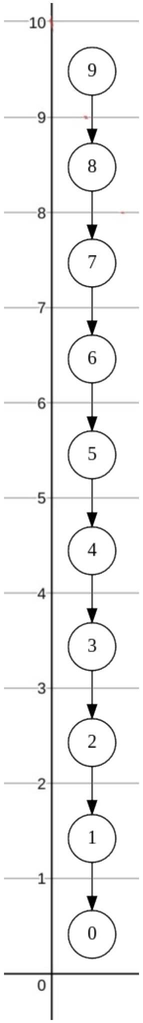
- Cannot stay in cube for more than:

- Assume f_{min} and f^{max} are the min and max derivates for a certain interval
-
- Cannot stay in slice for more than:
  - $t_i^{max} = 1 / f^{max}$
- Cannot leave slice in less time than:
  - $t_i^{max} = 1 / f_{min}$



Source: Approximating continuous systems by timed automata – Maler, Batt

Example of a transformation



Clocks:

One general clock:

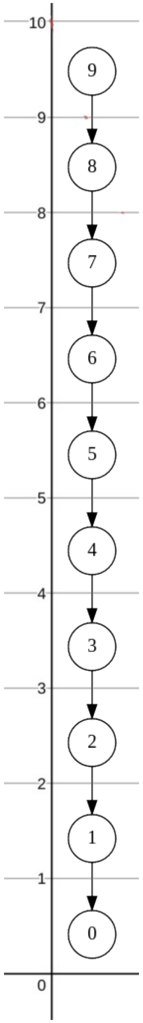
- z – reset at every transition

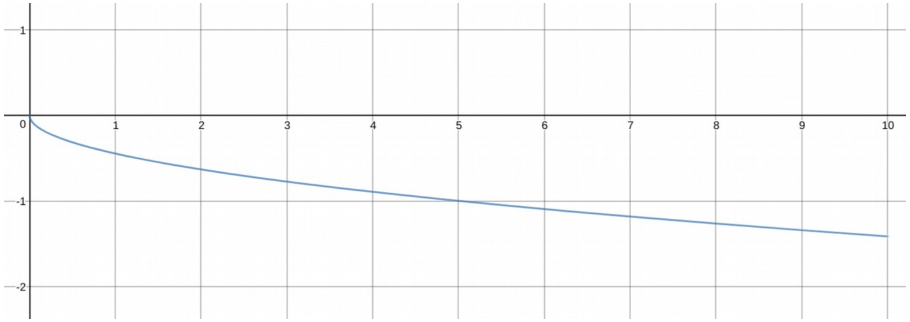
Two clocks per dimension:

- z_1^+ - reset when entering $slice_i$ from the left
- z_1^- - reset when entering $slice_i$ from the right
- We will only use first one for simplicity

Source: Approximating continuous systems by timed automata – Maler, Batt

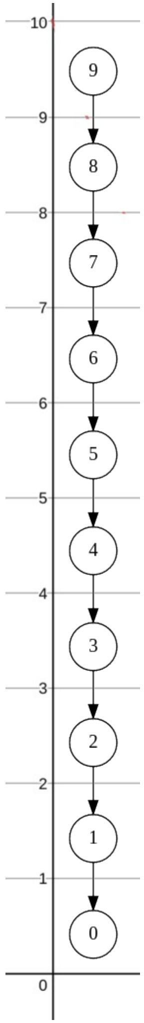
Example of a transformation





Source: Approximating continuous systems by timed automata – Maler, Batt

Example of a transformation



Clocks:

Invariant:

-

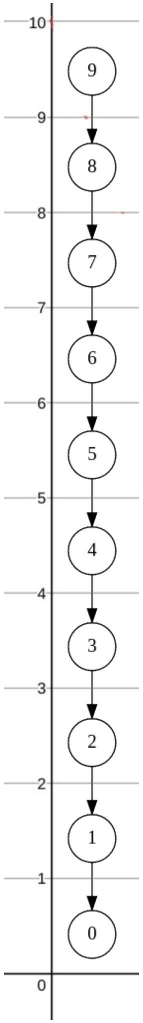
Transition:

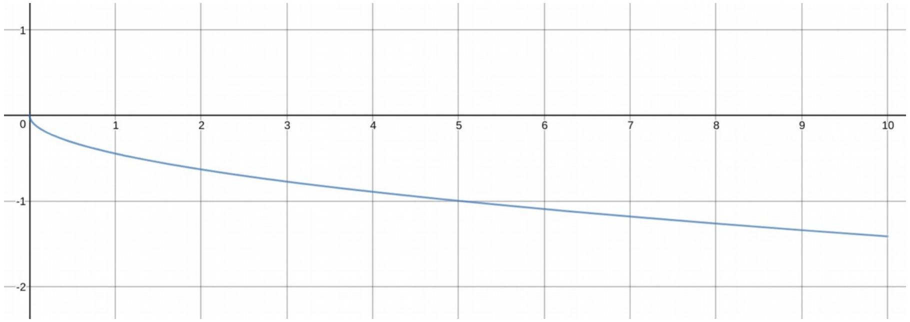
- successor
- predecessor

Predecessor:

Source: Approximating continuous systems by timed automata – Maler, Batt

Example of a transformation





Source: Approximating continuous systems by timed automata – Maler, Batt

What's next?

- Transformation CT-CBD to Timed Automata (Uppaal)
- Worked out non-trivial use case
- Extension/Modification?