Model checking timed automata — region graphs

Mariëlle Stoelinga
Formal Methods & Tools
This screencast

System meets??

Safety Requirements:

Invariant:
at_crossing → gates closed

Algorithms?

Data structures?

Violated + counter example

Satisfied
Model checking for TA: principles

How to check if target state(s) are reachable?

Untimed case: Standard graph algorithms
- Breadth first search (BFS)
- Depth first search (DFS)
- …

Timed case
- Infinite state space
- Hence:
  - Group states together
  - Obtain finite state space: region graph / automaton
  - Apply standard BSF / DFS techniques

Untimed

Infinite state space

Finite Region Automaton
Untimed case: Model checking reachability

Standard BFS algorithm

- **Input:** Finite LTS $A = (S, s_0, A, \rightarrow)$, target $T$
- **Output:** is $T$ reachable from $s_0$
- **Method:**
  - $\text{passed} := \{} \%$ seen all successors
  - $\text{wait} := \{s_0\} \%$ to be processed
  - while $\text{wait} \neq \{}$ do
    - move $s$ from $\text{wait}$ to $\text{passed}$
    - if $s$ in $T$ then return YES
    - else
      - (compute successors)
      - If $s'$ is a successor of $s$ and $s'$ not in $\text{passed}$ then add $s'$ to $\text{passed}$
  - endwhile
**Regions**

**Idea:** group together equivalent states

- equivalent states satisfy same clock constraints
- **KEY INSIGHT:** if one state enables transition, then all equivalent enable the transition and move to equivalent states

→ Group equivalent states into one
→ There are finitely many groups / classes
= Finite Region Automaton

**When are states equivalent?**

1. single clock case
2. multiple clock case
Regions single-clock case

- Equivalent states satisfy exactly same clock constraints
- Which valuations are equivalent? (Write down!)

\[
\begin{align*}
\text{x } & \leftarrow 0 & \quad \text{v}_1(x) = 0 \\
\text{x } & \leftarrow 1 & \quad \text{v}_2(x) = 1 \\
\text{x } & \leftarrow 1.1 & \quad \text{v}_3(x) = 1.1 & \quad \text{v}_4(x) = 1.2 & \quad \text{v}_5(x) = 1.95 \\
\text{x } & \leftarrow 1.2 & \quad \text{v}_6(x) = 3 \\
\text{x } & \leftarrow 1.95 & \quad \text{v}_7(x) = 50 \quad \text{v}_8(x) = 200 \\
\text{x } & \leftarrow 3 & \\
\text{x } & \leftarrow 50 & \quad \text{x } & \leftarrow 200 
\end{align*}
\]

Finite nr of equiv classes = regions

Clock valuations \(u,v\) over \(C=\{x\}\) are equivalent iff:
- \(u(x), v(x)\) are either both strictly larger than \(K_x\), or
- all of the following hold
  - \(u(x)\) and \(v(x)\) have the same integer parts: \([u(x)] = [v(x)]\)
  - \(u(x)\) and \(v(x)\) both integers, or both non-integers: \(\{u(x)\} = 0\) iff \(\{v(x)\} = 0\)

Here \([r]\) is integer part, \(\{r\}\) is fractional part of real number \(r\)

\[k_x < 50\]

max clock constant for \(x\)
Recall: clock regions

Which valuations are equivalent? (Write down)

\[ x \leftarrow 1 \quad y \leftarrow 1.2 \\
1 \quad 1.2 \]

\[ x \leftarrow 1.1 \quad y \leftarrow 1.2 \\
1.1 \quad 1.2 \]

\[ x \leftarrow 1.2 \quad y \leftarrow 1.1 \\
1.2 \quad 1.1 \]

\[ x \leftarrow 1.2 \quad y \leftarrow 1.2 \\
1.2 \quad 1.2 \]

\[ x \leftarrow 1.95 \quad y \leftarrow 1.2 \\
1.95 \quad 1.2 \]

\[ x \leftarrow 20 \quad x \leftarrow 30 \\
20 \quad 30 \]

\[ y \leftarrow 1.2 \quad y \leftarrow 1.2 \\
1.2 \quad 1.2 \]

\[ y \leftarrow 1.2 \quad y \leftarrow 1.8 \\
1.2 \quad 1.8 \]

Clock valuations \( u, v \) over \( C \) are \textit{equivalent} iff for all \( x, y \in C \):

- \( u(x), v(x) \) are either both larger than \( K_x \), or
- all of the following hold
  - \( u(x) \) and \( v(x) \) have the same integer parts: \( [u(x)] = [v(x)] \)
  - \( u(x) \) and \( v(x) \) both integers, or both non-integers: \( \{u(x)\} = 0 \) iff \( \{v(x)\} = 0 \)
  - \( \{u(x)\} \leq \{u(y)\} \iff \{v(x)\} \leq \{v(y)\} \)

For \( K_x > 2 \)

In both cases: \( x \geq y \)
Valuations $u,v$ are *equivalent* iff for all $x, y \in C$:
- $u(x), v(x)$ are either both larger than $K_x$, or
- all of the following hold
  - same integer parts: $\lfloor u(x) \rfloor = \lfloor v(x) \rfloor$
  - both integers / non-integers: $\{u(x)\}=0$ iff $\{v(x)\}=0$
  - $\{u(x)\} \leq \{u(y)\} \iff \{v(x)\} \leq \{v(y)\}$
Region Automaton: Finite Partitioning

- **Action transitions:**
  - If $w \equiv v$ and $(l,w) -a-> (l',w')$ then $\exists v' \equiv w'$ and $(l,v) -a-> (l',v')$

- **Delay transitions:**
  - If $w \equiv v$ and $(l,w) -d-> (l',w')$ then $\exists d'v' \equiv w'$ s.t. $(l,v) -d-> (l',v')$

$\equiv$ is time-abstract bisimulation

### Region automaton of A:

- **States** are region-equivalence classes
- **Action transitions:** $C - a \rightarrow C'$ in RA iff $\exists s \in C, s' \in C'. s - a \rightarrow s'$ in TLTS
- **Delay transitions:** $C \rightarrow C'$ in RA iff $\exists s \in C, s' \in C'. s - a \rightarrow s'$ in TLTS
Example: Region Graph

- \( x \geq 2 \)
- \( a \)
- \( x := 0 \)

Maximal constant?
- \( K_x = 2 \)

Clock regions?
- \( x = 0 \)
- \( 0 < x < 1 \)
- \( x = 1 \)
- \( 1 < x < 2 \)
- \( x = 2 \)
- \( x > 2 \)
Thank you for your attention