An brief introduction to timed automata

part B: modeling

Mariëlle Stoelinga

Formal Methods & Tools
Software systems correctness: It’s about time

**KEY:** Are these systems correct?
- Meet timing requirements, eg deadlines?
- TA modeling & analysis!

### Airbags
- Inflator
- Nitrogen Gas
- Crash Sensor

**Inflate in 40 ms**

### Medical
- **WCET:** very low

### Networks
- **Jitter, packet delay**

### Trains
- **Gates must close on time**

### Too late = lost game
Engineering correct systems: the model checking approach

- System
  - System model (TA)
  - Formalize

- Model checker
  - Meets??

- Requirements
  - Desired property
  - Formalize

- Violated + counter example
- Out of memory
- Satisfied

Automatic verification of system model against real-time property
Today

This screen cast

- Details of TA model
- Small, realistic example: rail road controller

Safety Requirements: `at_crossing → gates closed`

System model violates the invariant, so there is a counterexample.

Model checker checks the safety requirements and determines whether the invariant is satisfied or violated.
Timed Automata

Controller of Railroad Crossing

Gate

Diagram:

- **open** → **lowering**
- **lowering** → **closed**
- **closed** → **raising**
- **raising** → **open**
Timed Automata

Controller of Railroad Crossing

Gate

- **Time progress via clocks**
  - Time is spent in locations
  - Invariant tells how long you can stay
  - Transitions are instantaneous
  - Guard tells when you can take it
  - Clocks can be reset on transition

- **(Clock) resets**
  - Executed when taking a transition
  - You can assign 0, but no other values

- **Invariants**
  - How long can you be in a location?
  - As long as invariant is true!
  - Impose deadlines on leaving location

- **Lowering bars takes at most 15 seconds**

- **(Clock) guards**
  - When can you take a transition?
  - If the guard is true!
  - Impose enabling conditions

- **Actions**
  - Communication with other TAs
  - Internal / invisible transitions without label

- **Lowering bars takes at least 10 seconds**
  - Lowering bars takes 10 - 15 seconds
Timed Automata: small example

Controller of Railroad Crossing

- Gate is normally open
- If train approaches, gate closes in 10-15 seconds
- If train leaves, gate opens in 10-15 seconds

Clock constraints
- **guards:** transition *may* be taken (enabling conditions)
- **invariants:** transition *must* be taken (deadlines); ensure progress

- All clocks progress with same speed
- initially all clocks are 0

syntax
- \( x \leq c, x < c, x \geq c, x > c \)
- \( x - y \leq c, \ldots \)
- conjunctions: \( x < 3 \& y \geq 5 \)
- \( c \) integer

Important
- location ≠ state
- states
- location + clock values

Gate

- initial location
- (clock) guard
- location / node
- (clock) reset
- action / label
- (location) invariant
Timed Automata: executing transitions

- Start in location $L_1$
  - Invariant $\text{Inv}_1$ holds
- Check if guard $G$ holds
  - otherwise transition cannot be taken
- Take the action $a$
  - multiple outgoing $a$ actions: nondeterminism
- Reset all clocks in reset set
  - other clock retain their values
- Check if invariant in target state holds
  - otherwise transition cannot be taken
- Move to the target location

- All elements are optional

- $x > 10 \& y \leq 39 \& z < 3$
- GoAhead $x := 0$
Exercise

1. Model a traffic light as a TA

- The light can show three colors: *red*, *yellow* and *green*
- It cycles through these colors in the usual order
- Initially, the light is yellow
- The red light is shown for exactly 4 minutes
- The green light is shown for exactly 1 minute
- The yellow light is shown for a duration between 0.1 and 0.2 minutes
Thank you for your attention
& See you next time!