Model checking
timed automata

---
part 1: semantics

Mariëlle Stoelinga
Formal Methods & Tools
Model checking TAs: how?

Is R reachable?

P

3 \leq y - x \leq 5
x := 0
y := 0

P, x \leftarrow 0, y \leftarrow 0

3
P, x \leftarrow 3, y \leftarrow 3

b
P, x \leftarrow 0, y \leftarrow 3

4
P, x \leftarrow 4, y \leftarrow 7

c
Q, x \leftarrow 0, y \leftarrow 7

d
R, x \leftarrow 0, y \leftarrow 7

Yes:

R reachable = some run (path) leads to R

Runs work on states = locations + clock values

Transitions can be delays or actions

Transitions must respect invariants (delays) and guards (actions)

Mathematically: TLTSs

1. What does reachability mean?
2. How to compute in TA if given location is reachable?

Ideas?

Exercise (pairs)

- Write down some other runs of H
  - Some reach R
  - Some do not reach R
Model checking TAs: runs

Is R reachable?

Yes:
- R reachable = some trace leads to R
- Traces work on states = locations + clock values
- Transitions can be delays or actions
- Transitions must respect invariants (delays) and guards (actions)
- Mathematically: TLTSs

TLTSs
- Infinitely many states / transitions: not suitable for model checking
- Pin down meaning
  - No ambiguity
  - Equivalence of TAs = same semantics

State in TLTS

Delay transition in TLTS (must respect invariant in P)

Immediate transition in TLTS (must respect guard of transition taken in TA)
Timed Automata: equivalence

L
x < 30

x > 10 & y ≤ 39

GoAhead
x:=0

L'
z < 3

x > 10 & y ≤ 39
& z < 3

GoAhead
x:=0

L'
z < 3

Equivalence

- Exactly same semantics
Model checking TAs: overview

1. What does reachability mean?
2. How to compute in TA if given location is reachable?

finite LTS
- states = lo
- wha
- reuse existing theory:
  - paths, traces, reachability
  - bisimulation, (timed) trace equivalence
- ...

infinite LTS
- states = lo
- wha
- reuse existing theory:
  - paths, traces, reachability
  - bisimulation, (timed) trace equivalence
- ...

finite LTS
- timing info in states
- basic model checking
  - complexity
  - decidability
- efficient model checking
- data structure: DBMs

Zone-construction
- Uppaal

Region construction
- Region Automaton

group states

group more states

Today

Semantics

TA

TLTS

ZA

Zone Automaton
A **TLTS** is a 4-tuple \((S, s_0, A, \rightarrow)\) given by:

- \(S\) is the set of *states*
- \(s_0 \in S\) is the *initial state*
- \(A\) is the *action alphabet*
- \(\rightarrow \subseteq S \times (A \cup R_{\geq 0}) \times S\) is the *transition relation* s.t.
  
  for all \(s, s_1, s_3 \in S\) and \(d_1, d_2 \in R_{\geq 0}\)
  
  - \(s \xrightarrow{d_1} s\)
  - \(\exists s_2 \in S . \ s_1 \xrightarrow{d_1} s_2 \text{ and } s_2 \xrightarrow{d_2} s_3 \iff s_1 \xrightarrow{d_1 + d_2} s_3\)
Model checking TAs: runs

A **TLTS** is a 4-tuple \((S, s_0, A, \rightarrow)\) given by

- \(S\) is the set of *states*
- \(s_0 \in S\) is the *initial state*
- \(A\) is the *action alphabet*
- \(\rightarrow \subseteq S \times (A \cup R_{\geq 0}) \times S\) *transition relation*

for all \(s, s_1, s_3 \in S\) and \(d_1, d_2 \in R_{\geq 0}\)

- \(s \rightarrow s\)
- \(\exists s_2 \in S. s_1 \rightarrow s_2\) and \(s_2 \rightarrow s_3 \iff s_1 \rightarrow s_2\)

---

**Go through the formal definition**

- All 4 ingredients
- Some notation needed

---

**Immediate transition** in TLTS

(must respect guard of transition taken in TA)

**Delay transition** in TLTS

(must respect invariant in P)
1b: How to obtain underlying TLTS from TA?

Examples
- (P, <x<0, y<7>)
- (P, <x<1.2, y<3.4>)
- (P, <x<v2, y<log 37>)
- (Q, <x<3, y<3>)

- Invariant violated
- Still a state
- Not all are reachable
  - E.g. if invariant violated
  - Infinitely many (reachable) states

Initial state in TLTS
- Initial location
- All clocks equal 0

Immediate actions in TLTS
- All actions in TA

States in TLTS
Tuple with
- Location
- Clock valuation in v: C → R≥0

Initial state in TLTS
- Initial location
- All clocks equal 0

Immediate actions in TLTS
- All actions in TA

Initial state in H
- (P, <x<0, y<0>) i.e. (P,0)

Immediate actions in H?
{a,b,c,d}
A **TLTS** is a 4-tuple \((S, s_0, A, \rightarrow)\) given by

- **S** is the set of *states*
- **s_0 \in S** is the *initial state*
- **A** is the *action alphabet*

\[
\subseteq S \times (A \cup R_{\geq 0}) \times S \text{ transition relation}
\]

for all \(s, s_1, s_3 \in S\) and \(d_1, d_2 \in R_{\geq 0}\)

- \(s \rightarrow s\)
- \(\exists s_2 \in S. s_1 \rightarrow s_2 \text{ and } s_2 \rightarrow s_3 \iff s_1 \rightarrow s_2\)

**Go through the formal definition**

- All 4 ingredients
- Some notation needed

**Delay transition** in TLTS
(must respect invariant in P)

**Immediate transition** in TLTS
(must respect guard of transition taken in TA)
1b: How to obtain underlying TLTS from TA?

Delay transitions
- Time can progress as long as invariant holds.
- In location $l$, you can delay for $d$ iff increasing all clock by $d$ still satisfies the invariant of $l$.
- All delay transitions of TA given by
  \[
  \{ (<l,u>, d, <l,u+d>) \mid l \in L, \ u: C \rightarrow R_{\geq 0}, \ u+d \models \text{inv}(l) \}
  \]

Notation
- $u: C \rightarrow R_{\geq 0}$ a clock valuation

Example
- $<x \leftarrow 4, y \leftarrow 7>$ means
- $u: \{x,y\} \rightarrow R_{\geq 0}$ with $u(x) = 4$, $u(y) = 7$

Satisfaction: Given
- $u: C \rightarrow R_{\geq 0}$ a clock valuation,
- $g \in B(C)$ a clock guard
  \[
  u \models g \iff g \text{ holds for values from } u \]

Exercise (pairs)
- Think of some examples with / without $u \models x - z < 6 \& z > 0$
1b: How to obtain underlying LTS from TA?

Delay transitions
- Time can progress as long as invariant holds.
- In location $l$, you can delay for $d$ if increasing all clocks by $d$ still satisfies the invariant of $l$.
- All delay transitions of TA given by
  \[ \{ (<l,u>, d, <l,u+d>) \mid l \in L, u : C \rightarrow R_{\geq 0}, u+d \models inv(l) \} \]

Exercise (pairs)
- Think of some examples with / without $u \models x - z < 6 \& z > 0$

More notation
- $R \subseteq C$ subset of clocks, $d \in R_{\geq 0}$
- $u[R:=0]$ valuation with all clocks in $R$ equal to 0 and other clocks as in $u$.
- $u + d$ valuation with all clocks increased by $d$.

Satisfaction: Given
- $u : C \rightarrow R_{\geq 0}$ a clock valuation,
- $g \in B(C)$ a clock guard
- $u$ satisfies $g$
- $u \models g$ iff $g$ holds for values from $u$. 

Exercise (pairs)
- Think of some examples with / without $u \models x - z < 6 \& z > 0$
Timed Automata: executing transitions

1. Start in location $L_1$
   - Invariant $Inv_1$ holds
2. Check if guard $G$ holds
   - otherwise transition cannot be taken
3. Take the action $a$
   - multiple outgoing $a$ actions: nondeterminism
4. Reset all clocks in reset set
   - other clock retain their values
5. Check if invariant in target state holds
   - otherwise transition cannot be taken
6. Move to the target location $L_2$
1b: How to obtain underlying TLTS from TA?

What are the transitions in $H$?
- $(P, <x=4, y=7>) \xrightarrow{c} (Q, <x=0, y=7>)$
- $(Q, <x=0, y=7>) \xrightarrow{d} (R, <x=3, y=7>)$

Action transitions in $(L, l_0, \Sigma, E, \text{Inv})$

\[
\{ (<l,u>,a,<l',u[r:=0]> ) \mid (l,g,a,r,l') \in E, \ u \models g, \ u[r:=0] \models \text{inv}(l') \} \]
tuple \( (L, l_0, \Sigma, C, E, \text{Inv}) \) \rightarrow \text{Semantics} \rightarrow \text{tuple} \ (S, s_0, A, \rightarrow)
Let $H = (L, l_0, \Sigma, C, E, \text{inv})$ be a TA. The **underlying TLTs of $H$** is given by $(S, s_0, A, \rightarrow)$ where

- $S = L \times (C \rightarrow R_{\geq 0})$
- $s_0 = (l_0, 0)$ is the intial state
- $A = \Sigma$
- $\rightarrow =$

$\{ (l,u),d,(l,u+d) \mid l \in L, u \in B(C), u+d \models \text{inv}(l) \}$

$U$

$\{ (l,u),a,(l',u[r:=0]) \mid (l,g,a,r,l') \in E, u \models 9, u[r:=0] \models \text{inv}(l') \}$
Model checking TAs: overview

1. What does reachability mean?
2. How to compute in TA if given location is reachable?

finite LTS
- timing info in states
- basic model checking
  - complexity
  - decidability

infinite LTS
- states = lo
- wha
- reuse existing theory:
  - paths, traces, reachability
  - bisimulation, (timed) trace
equivalence
  - ...

TA

TLTS

Semantics

Region construction

group states

Zone-construction

Uppaal

ZA

group more states

Region Automaton

Zone Automaton

finite LTS
- timing info in states
- efficient model checking
- data structure: DBMs

Today

Region Automaton

Zone Automaton
Thank you for your attention & See you next time!