Real-Time Systems

Lecture 15: The Universality Problem for TBA

2013-06-26

Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany

Contents & Goals

Last Lecture:

- Extended Timed Automata

This Lecture:

- **Educational Objectives:** Capabilities for following tasks/questions.
  - What's a TBA and what's the difference to (extended) TA?
  - What's undecidable for timed (Büchi) automata?
  - What's the idea of the proof?
- **Content:**
  - Uppaal Query Language
  - Timed Büchi Automata and timed regular languages [Alur and Dill, 1994].
  - The Universality Problem is undecidable for TBA [Alur and Dill, 1994]
  - Why this is unfortunate.
  - Timed regular languages are not everything.

The Logic of Uppaal

The Uppaal Fragment of Timed Computation Tree Logic

Consider $\mathcal{N} = \mathcal{C}(\mathcal{A}_1, \dots, \mathcal{A}_n)$ over data variables V.

- **basic formula**:
$$atom ::= \mathcal{A}_i.\ell \mid \varphi$$
  where $\ell \in L_i$ is a location and φ a constraint over X_i and V.
- **configuration formulae**:
$$term ::= atom \mid \neg term \mid term_1 \wedge term_2$$
- **existential path formulae**: ("exists finally", "exists globally")
$$e\text{-}formula ::= \exists\Diamond\, term \mid \exists\Box\, term$$
- **universal path formulae**: ("always finally", "always globally", "leads to")
$$a\text{-}formula ::= \forall\Diamond\, term \mid \forall\Box\, term \mid term_1 \longrightarrow term_2$$
- **formulae**:
$$F ::= e\text{-}formula \mid a\text{-}formula$$

Configurations at Time t

0 3.0 3.0 τ 3.0 10

- Recall: **computation path** (or path) **starting in** $\langle \vec{\ell}_0, \nu_0 \rangle, t_0$:
$$\xi = \langle \vec{\ell}_0, \nu_0 \rangle, t_0 \xrightarrow{\lambda_1} \langle \vec{\ell}_1, \nu_1 \rangle, t_1 \xrightarrow{\lambda_2} \langle \vec{\ell}_2, \nu_2 \rangle, t_2 \xrightarrow{\lambda_3} \dots$$
  which is **infinite or maximally finite**.
- Given ξ and $t \in \mathrm{Time}$, we use $\xi(t)$ to denote the set
$$\{\langle \vec{\ell}, \nu \rangle \mid \exists i \in \mathbb{N}_0 : t_i \le t \le t_{i+1} \wedge \vec{\ell} = \vec{\ell}_i \wedge \nu = \nu_i + t - t_i\}.$$
  of **configurations at time** t.
  - Why is it a set?
  - Can it be empty?

$\xi(0) = \{\langle \vec{\ell}_0, \nu_0 \rangle\}$

$\xi(0.2) = \{\langle \vec{\ell}_0, \nu_0 + 0.2 \rangle\}$

$\xi(3.0) = \{\langle \vec{\ell}_1, \nu_1 \rangle, \langle \vec{\ell}_2, \nu_2 \rangle\}$

Satisfaction of Uppaal-Logic by Configurations

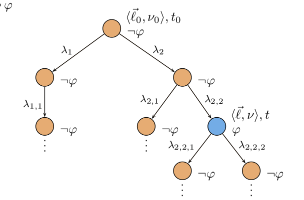
- We define a **satisfaction relation**
$$\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \models F$$
  between **time stamped configurations**
$$\langle \vec{\ell}_0, \nu_0 \rangle, t_0$$
  of a network $\mathcal{C}(\mathcal{A}_1, \dots, \mathcal{A}_n)$ and **formulae** F of the Uppaal logic.
- It is defined inductively as follows: i-th location in $\vec{\ell}_0$
- $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \models \mathcal{A}_i.\ell$ iff $\ell_{0,i} = \ell$
- $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \models \varphi$ iff $\nu_0 \models \varphi$
- $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \models \neg term$ iff $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \not\models term$
- $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \models term_1 \wedge term_2$ iff $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \models term_i$, $i = 1,2$

Satisfaction of Uppaal-Logic by Configurations

Exists finally:

- $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \models \exists\Diamond\, term$ iff $\exists$ path ξ of $\mathcal{N}$ starting in $\langle \vec{\ell}_0, \nu_0 \rangle, t_0$
  $\exists t \in$ Time, $\langle \vec{\ell}, \nu \rangle \in Conf$:
  $t_0 \leq t \wedge \langle \vec{\ell}, \nu \rangle \in \xi(t) \wedge \langle \vec{\ell}, \nu \rangle, t \models term$

Example: $\exists\Diamond\,\varphi$



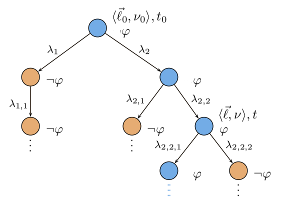
Satisfaction of Uppaal-Logic by Configurations

Exists globally:

- $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \models \exists\Box\, term$ iff $\exists$ path ξ of $\mathcal{N}$ starting in $\langle \vec{\ell}_0, \nu_0 \rangle, t_0$
  $\forall t \in$ Time, $\langle \vec{\ell}, \nu \rangle \in Conf$:
  $t_0 \leq t \wedge \langle \vec{\ell}, \nu \rangle \in \xi(t) \implies \langle \vec{\ell}, \nu \rangle, t \models term$

note: universally quantifying over elements in $\xi(t)$

Example: $\exists\Box\,\varphi$



Satisfaction of Uppaal-Logic by Configurations

- **Always finally:** "∀ path" "∃ t ∈ Time"
  - $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \models \forall\Diamond\, term$ iff $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \not\models \exists\Box\,\neg term$
- **Always globally:**
  - $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \models \forall\Box\, term$ iff $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \not\models \exists\Diamond\,\neg term$

"∀ path" "∀ t ∈ Time"

Satisfaction of Uppaal-Logic by Configurations

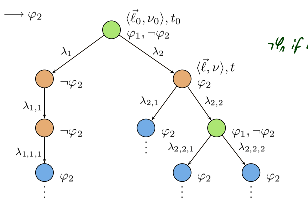
Leads to: CTL: AG(term₁ ⇒ AF term₂)

- $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \models term_1 \longrightarrow term_2$ iff $\forall$ path ξ of $\mathcal{N}$ starting in $\langle \vec{\ell}_0, \nu_0 \rangle, t_0$
  $\forall t \in$ Time, $\langle \vec{\ell}, \nu \rangle \in Conf$:
  $t_0 \leq t \wedge \langle \vec{\ell}, \nu \rangle \in \xi(t)$
  $\wedge \langle \vec{\ell}, \nu \rangle, t \models term_1$
  implies $\langle \vec{\ell}, \nu \rangle, t \models \forall\Diamond\, term_2$

not DC (of course)

Example: $\varphi_1 \longrightarrow \varphi_2$

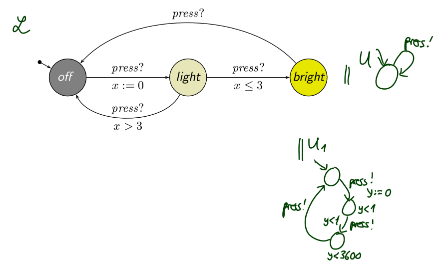
¬φ₁ if not given



Satisfaction of Uppaal-Logic by Networks

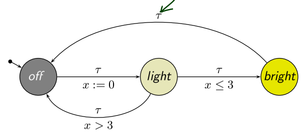
- We write
  $$\mathcal{N} \models e\text{-}formula$$
  if and only if
  $$\text{for some } \langle \vec{\ell}_0, \nu_0 \rangle \in C_{ini}, \langle \vec{\ell}_0, \nu_0 \rangle, 0 \models e\text{-}formula, \qquad (1)$$
  and
  $$\mathcal{N} \models a\text{-}formula$$
  if and only if
  $$\text{for all } \langle \vec{\ell}_0, \nu_0 \rangle \in C_{ini}, \langle \vec{\ell}_0, \nu_0 \rangle, 0 \models a\text{-}formula, \qquad (2)$$
  where C_{ini} are the initial configurations of $\mathcal{T}_e(\mathcal{N})$.
- If $C_{ini} = \emptyset$, (1) is a contradiction and (2) is a tautology.
- If $C_{ini} \neq \emptyset$, then
  $$\mathcal{N} \models F \text{ if and only if } \langle \vec{\ell}_{ini}, \nu_{ini} \rangle, 0 \models F.$$

Example



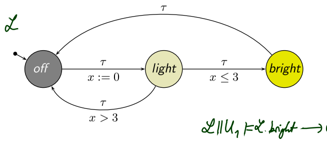
Example

now we have a closed network



Example

$\mathcal{L}$



- $\mathcal{N} \models \exists\Diamond\, \mathcal{L}.bright$? ✓
- $\mathcal{N} \models \exists\Box\, \mathcal{L}.bright$? ✗ (we must be in light before bright)
- $\mathcal{N} \models \exists\Box\, \mathcal{L}.off$? ✓ (*)
- $\mathcal{N} \models \forall\Diamond\, \mathcal{L}.light$? ✗ (because (*))
- $\mathcal{N} \models \forall\Box\, \underbrace{(\mathcal{L}.bright \implies x \geq 3)}_{term}$? ✗ (can have $\mathcal{L}$.bright and $x<3$)
- $\mathcal{N} \models \mathcal{L}.bright \longrightarrow \mathcal{L}.off$? ✗

$\mathcal{L} \| U_1 \models \mathcal{L}.bright \longrightarrow \mathcal{L}.off$? ✓ Uppaal (because the tool uses $\overset{\alpha}{\Longrightarrow}$ paths, not comp. paths actually)

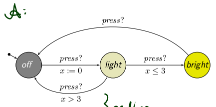
✗ prev. Def (because not satisfied in some Zeno paths)

Timed Büchi Automata

[Alur and Dill, 1994]

… vs. Timed Automata

$\mathcal{A}$:

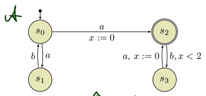


produces

$\xi = \langle off, 0\rangle, 0 \xrightarrow{1} \langle off, 1\rangle, 1$
$\xrightarrow{press?} \langle light, 0\rangle, 1 \xrightarrow{3} \langle light, 3\rangle, 4$
$\xrightarrow{press?} \langle bright, 3\rangle, 4 \xrightarrow{\cdots} \ldots$

ξ is a computation path and run of $\mathcal{A}$.

$\mathcal{A}$



New: Given a **timed word** (accept)

$(a, 1), (b, 2), (a, 3), (b, 4), (a, 5), (b, 6), \ldots,$

does $\mathcal{A}$ **accept** it?

New: acceptance criterion is
visiting accepting state infinitely often.

Timed Languages

Definition. A **time sequence** $\tau = \tau_1, \tau_2, \ldots$ is an infinite sequence of time values $\tau_i \in \mathbb{R}_0^+$, satisfying the following constraints:

(i) **Monotonicity:**
τ increases **strictly** monotonically, i.e. $\tau_i < \tau_{i+1}$ for all $i \geq 1$.

(ii) **Progress:** For every $t \in \mathbb{R}_0^+$, there is some $i \geq 1$ such that $\tau_i > t$.

Definition. A **timed word** over an alphabet Σ is a pair (σ, τ) where

- $\sigma = \sigma_1, \sigma_2, \cdots \in \Sigma^\omega$ (set of infinite words over Σ) is an infinite word over Σ, and
- τ is a time sequence.

Definition. A **timed language** over an alphabet Σ is a set of timed words over Σ.

Example: Timed Language

Timed word over alphabet Σ: a pair (σ, τ) where
- $\sigma = \sigma_1, \sigma_2, \ldots$ is an infinite word over Σ, and
- τ is a time sequence (strictly (!) monotonic, non-Zeno).

a could be 'system beeps'
b could be 'system flashes light'

$$L_{crt} = \{((ab)^\omega, \tau) \mid \exists\, i\ \forall\, j \geq i : (\tau_{2j} < \tau_{2j-1} + 2)\}$$

timestamp of a 'b' — timestamp of the 'a' before

1 — 10,0

a b a b a b … a b a b a b …
τ_1 τ_2 τ_3 τ_4 τ_5 τ_6 … τ_{27} τ_{28} τ_{29} τ_{30} τ_{31} τ_{32}

$\tau_{30} < \tau_{29} + 2$

finite prefix where timestamps don't matter

from here on…

..b at most (not including) 2 time units after the a before

Timed Büchi Automata

not simple! (negation, but no differences)

Definition. The set $\Phi(X)$ of **clock constraints** over X is defined inductively by

$$\delta ::= x \leq c \mid c \leq x \mid \neg\delta \mid \delta_1 \wedge \delta_2$$

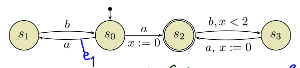
where $x \in X$ and $c \in \mathbb{Q}$ is a rational constant.

Definition. A **timed Büchi automaton** (TBA) $\mathcal{A}$ is a tuple $(\Sigma, S, S_0, X, E, F)$, where

- Σ is an alphabet,
- S is a finite set of states, $S_0 \subseteq S$ is a set of start states,
- X is a finite set of clocks, and
- $E \subseteq S \times S \times \Sigma \times 2^X \times \Phi(X)$ gives the set of transitions.
  An edge $(s, s', a, \lambda, \delta)$ represents a transition from state s to state s' on input symbol a. The set $\lambda \subseteq X$ gives the clocks to be reset with this transition, and δ is a clock constraint over X.
- $F \subseteq S$ is a set of **accepting states**.

Example: TBA

$$\mathcal{A} = (\Sigma, S, S_0, X, E, F)$$
$$(s, s', a, \lambda, \delta) \in E$$



- $\Sigma = \{a, b\}$
- $S = \{s_1, s_0, s_2, s_3\}$
- $S_0 = \{s_0\}$
- $X = \{x\}$
- $F = \{s_2\}$
- $E = \{ \underbrace{(s_0, s_1, a, \emptyset, true)}_{= e_1}, (s_1, s_0, b, \emptyset, true), (s_0, s_2, a, \{x\}, true), (s_2, s_3, b, \emptyset, x<2), (s_3, s_2, a, \{x\}, true) \}$

References

References

[Alur and Dill, 1994] Alur, R. and Dill, D. L. (1994). A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235.

[Olderog and Dierks, 2008] Olderog, E.-R. and Dierks, H. (2008). *Real-Time Systems - Formal Specification and Automatic Verification*. Cambridge University Press.