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 $N = C(A_1, \ldots, A_n)$ over data variables $V$.

- Basic formula: $atom ::= A_i.ℓ | ϕ$ where $ℓ \in L_i$ is a location and $ϕ$ a constraint over $X_i$ and $V$.

- Configuration formulae: $term ::= atom | ¬ term | term_1 ∧ term_2$.

- Existential path formulae: $('existsfinally', 'existsglobally') e-formula ::= ∃ ♦ term | ∃ □ term$.

- Universal path formulae: $('alwaysfinally', 'alwaysglobally', 'leadsto') a-formula ::= ∀ ♦ term | ∀ □ term | term_1 −→ term_2$.

- Formulae: $F ::= e-formula | a-formula$.

Satisfaction of Uppaal-Logic by Configurations

We define a satisfaction relation $⟨\vec{ℓ}_0, ν_0⟩, t_0 | = \ F$ between timestamped configurations $⟨\vec{ℓ}_0, ν_0⟩, t_0$ of a network $C(A_1, \ldots, A_n)$ and formulae $F$ of the Uppaal logic.

It is defined inductively as follows:

- $⟨\vec{ℓ}_0, ν_0⟩, t_0 | = = A_i.ℓ$ iff $ℓ_0,i = ℓ$.

- $⟨\vec{ℓ}_0, ν_0⟩, t_0 | = = ϕ$ iff $ν_0 | = = ϕ$.

- $⟨\vec{ℓ}_0, ν_0⟩, t_0 | = = ¬ term$ iff $⟨\vec{ℓ}_0, ν_0⟩, t_0 | \neq = term$.

- $⟨\vec{ℓ}_0, ν_0⟩, t_0 | = = term_1 ∧ term_2$ iff $⟨\vec{ℓ}_0, ν_0⟩, t_0 | = = term_1, i = 1, 2$.

- $⟨\vec{ℓ}_0, ν_0⟩, t_0 | = = \exists finally finally$.

- $⟨\vec{ℓ}_0, ν_0⟩, t_0 | = = \exists finally globally$.

- $⟨\vec{ℓ}_0, ν_0⟩, t_0 | = = \exists leadsto$.
Example

Satisfaction of Uppaal-Logic by Configurations

Example

Satisfaction of Uppaal-Logic by Configurations

Example

Satisfaction of Uppaal-Logic by Configurations
Example off light bright

\[ \tau_x := 0 \]

\[ \tau_x \leq 3 \]

\[ \tau_x > 3 \]

\[ -15 \cdot 2013 -06-26 – Sutl – 12/41 \]

Example off light bright

\[ \exists \cdot N| = \exists \cdot L. \text{bright} \]

\[ \exists \cdot N| = \exists \cdot \square L. \text{bright} \]

\[ \exists \cdot N| = \forall \cdot \bullet L. \text{light} \]

\[ \exists \cdot N| = \forall \cdot \square L. \text{bright} \]

\[ \Rightarrow x \geq 3 \]

\[ \cdot N| = L. \text{bright} \rightarrow L. \text{off} \]

New:

Given a timed word \( (a, 1), (b, 2), (a, 3), (b, 4), (a, 5), (b, 6), \ldots \), does \( A \) accept it?

New:

acceptance criterion is visiting accepting state infinitely often.
Definition. The set $\Phi(X)$ of clock constraints over $X$ is defined inductively by

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

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

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

- $\Sigma$ 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 $(s, s', a, \lambda, \delta) \in E$ 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 $\delta$ is a clock constraint over $X$.

- $F \subseteq S$ is a set of accepting states.

Example: TBA $A = (\Sigma, S, S_0, X, E, F)$

```
\begin{align*}
E & = \{(s_1, s_2, a, \{(x, x), (0, x)\}, x \leq 0), (s_2, s_3, a, \{(x, x), (0, x)\}, x \leq 0)\}
\end{align*}
```

References
