# Real-Time Systems

# Lecture 16: Automatic Verification of DC Properties for Timed Automata

2018-01-11

Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany



## Content

- **Uppaal Query Language**
  - **Syntax**
  - Excursion: **Transition Graph**
  - **Satisfaction Relation**
- A **satisfaction relation** between timed automata and DC formulae
  - **observables** of timed automata
  - **evolution** induced by **computation path**
- A **simple and wrong** solution.
  - **ad-hoc** fix for invariants
- **Testable DC Properties**
  - **observer construction**
  - **untestable DC properties**

## *The Logic of Uppaal*





## *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 $\varphi$ 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 $t$*

- 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 $\xi$ and $t \in \mathrm{Time}$, we use $\xi(t)$ to denote the set

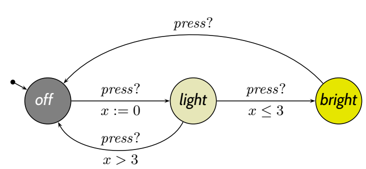
$$\{\langle \vec{\ell}, \nu \rangle \mid \exists\, i \in \mathbb{N}_0 : t_i \leq t \leq t_{i+1} \wedge \vec{\ell} = \vec{\ell}_i \wedge \nu = \nu_i + t - t_i\}.$$

  of **configurations at time** $t$.

  - Why is it a set?
  - Can it be empty?



## *Example*




$$\begin{aligned}\xi = \langle \textit{off}, x = 0 \rangle, 0 &\xrightarrow{4.2} \langle \textit{off}, x = 4.2 \rangle, 4.2 \xrightarrow{press?} \langle \textit{light}, x = 0 \rangle, 4.2 \\ &\xrightarrow{2.1} \langle \textit{light}, x = 2.1 \rangle, 6.3 \xrightarrow{press?} \langle \textit{bright}, x = 2.1 \rangle, 6.3 \\ &\xrightarrow{10} \langle \textit{bright}, x = 12.1 \rangle, 16.3 \xrightarrow{press?} \langle \textit{off}, x = 12.1 \rangle, 16.3 \\ &\xrightarrow{press?} \langle \textit{light}, x = 0 \rangle, 16.3 \xrightarrow{0} \langle \textit{light}, x = 0 \rangle, 16.3\end{aligned}$$

$$\boxed{\xi(t) = \{\langle \vec{\ell}, \nu \rangle \mid \exists\, i \in \mathbb{N}_0 : t_i \leq t \leq t_{i+1} \wedge \vec{\ell} = \vec{\ell}_i \wedge \nu = \nu_i + t - t_i\}}$$

- $\xi(0) =$ { <off, x=0> }
- $\xi(0.1) =$ { <off, x=0.1> }
- $\xi(4.1999) =$ { <off, x=4.1999> }
- $\xi(4.2) =$ { <off, x=4.2>, <light, 0> }
- $\xi(4.2001) =$ { <light, x=0.0001> }
- $\xi(16.3) =$ { <bright, x=12.1>, ... }
- $\xi(27) =$ { }

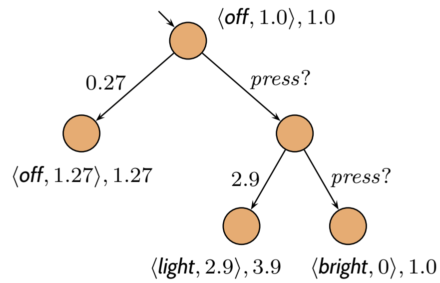
## *Excursion: Computation / Transition Graph*

- **Recall**: **operational semantics** of network $\mathcal{N}$ of timed automata is a **labelled transition system**

$$\mathcal{T}(\mathcal{N}) = (Conf, \mathsf{Time} \cup \{\tau\}, \{\xrightarrow{\lambda} \mid \lambda \in \mathsf{Time} \cup \{\tau\}\}, C_{ini}).$$

- (Parts of) $\mathcal{T}(\mathcal{N})$ can be represented as a directed, edge-labelled **graph** $(V, E, \mathsf{Time} \cup \{\tau\})$ where
  - **vertices** $V \subseteq Conf$ are (possibly time-stamped) **configurations**,
  - **graph-edges** $(c, \lambda, c')$ correspond to **transitions** $c \xrightarrow{\lambda} c'$.
- There may be at most one designated **start vertex** $c$,
  - paths in the graph **originating** at $c$
  - represent transition sequences (or computation paths) of $\mathcal{T}(\mathcal{N})$ **starting in** $c$.

**Example**: Desktop Lamp.






## *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 $\mathcal{C}(\mathcal{A}_1, \dots \mathcal{A}_i \dots, \mathcal{A}_n)$ and **formulae** $F$ of the Uppaal logic.

- It is defined inductively as follows (starting with **atoms** and **terms**):
  - $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \models \mathcal{A}_i.\ell$ 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*

- 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 (starting with **atoms** and **terms**):
  - $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \models \mathcal{A}_i.\ell$ 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*

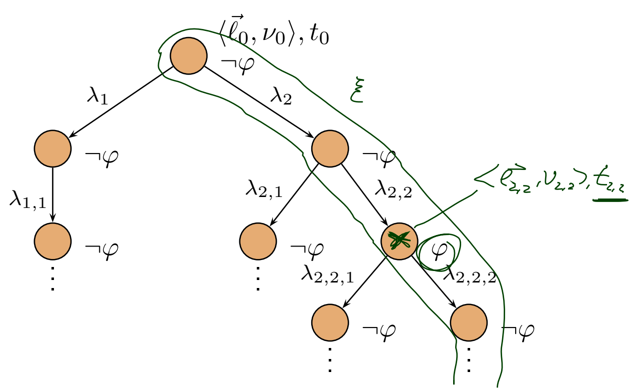
**Exists finally**:

- $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \models \exists\Diamond\, term$ iff $\exists$ path $\xi$ of $\mathcal{N}$ starting in $\langle \vec{\ell}_0, \nu_0 \rangle, t_0$

  $\exists t \in \mathrm{Time},\ \langle \vec{\ell}, \nu \rangle \in \mathit{Conf}$ :

  $t_0 \leq t \ \wedge\ \langle \vec{\ell}, \nu \rangle \in \xi(t) \ \wedge\ \langle \vec{\ell}, \nu \rangle, t \models term$

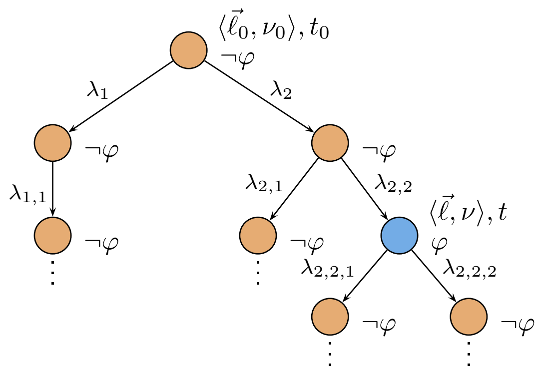
**Example**: $\exists\Diamond\, \varphi$



**Exists finally**:

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

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



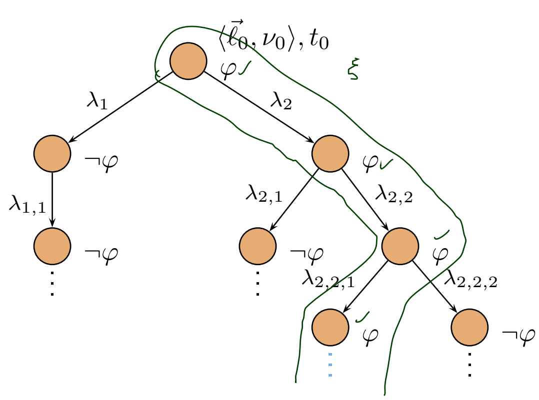



## Satisfaction of Uppaal-Logic by Configurations

**Exists globally**:

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

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

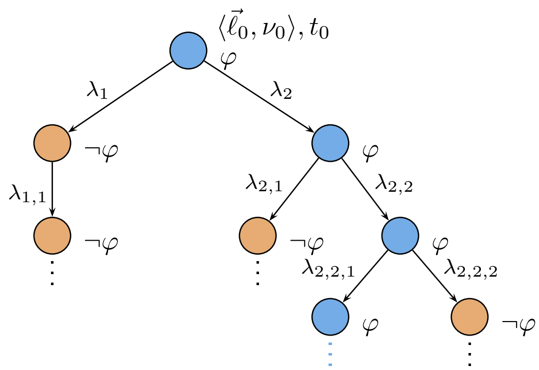


## *Satisfaction of Uppaal-Logic by Configurations*

**Exists globally**:

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

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





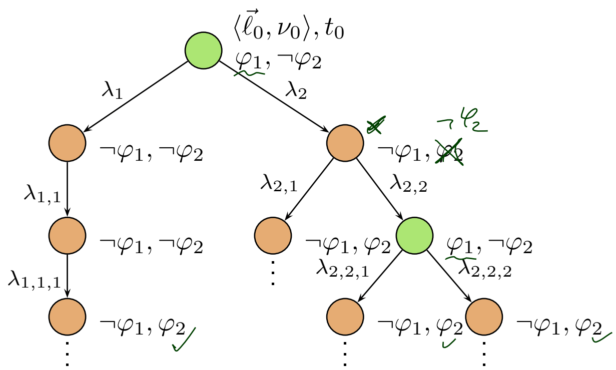

## *Satisfaction of Uppaal-Logic by Configurations*

- **Always finally**:
  - $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \models \forall\Diamond\, term$ 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 $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \not\models \exists\Diamond\, \neg term$

**Leads to:**

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

**Example**: $\varphi_1 \longrightarrow \varphi_2$







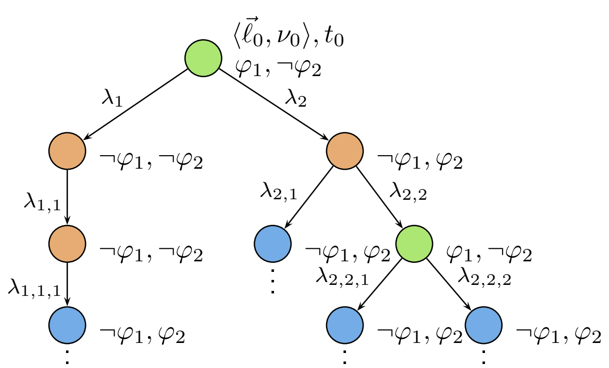

## *Satisfaction of Uppaal-Logic by Configurations*

**Leads to:**

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

  ($\sim \forall\Box(term_1 \to A\Diamond\, term_2)$)
  "response pattern"

**Example**: $\varphi_1 \longrightarrow \varphi_2$



# *Satisfaction of Uppaal-Logic by Networks*

- We write $\mathcal{N} \models e\text{-}formula$ if and only if

$$\text{for some } \langle \vec{\ell}_0, \nu_0 \rangle \in C_{ini}, \quad \langle \vec{\ell}_0, \nu_0 \rangle, 0 \models e\text{-}formula, \tag{1}$$

  and $\mathcal{N} \models a\text{-}formula$ if and only if

$$\text{for all } \langle \vec{\ell}_0, \nu_0 \rangle \in C_{ini}, \quad \langle \vec{\ell}_0, \nu_0 \rangle, 0 \models a\text{-}formula, \tag{2}$$

  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

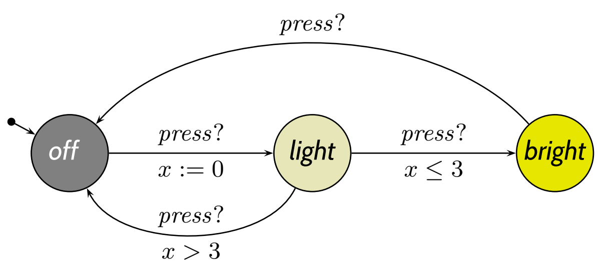
$$\mathcal{N} \models F \text{ if and only if } \langle \vec{\ell}_{ini}, \nu_{ini} \rangle, 0 \models F.$$

$\models$

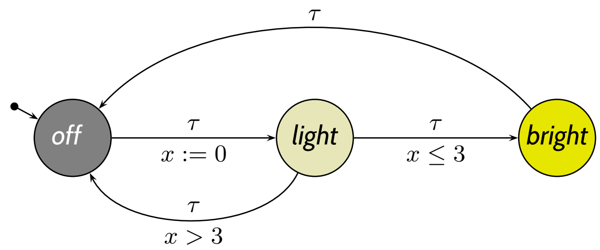
$A\Box$ coffee ✗ 1 ✓ ||||



# *Example*



## Example



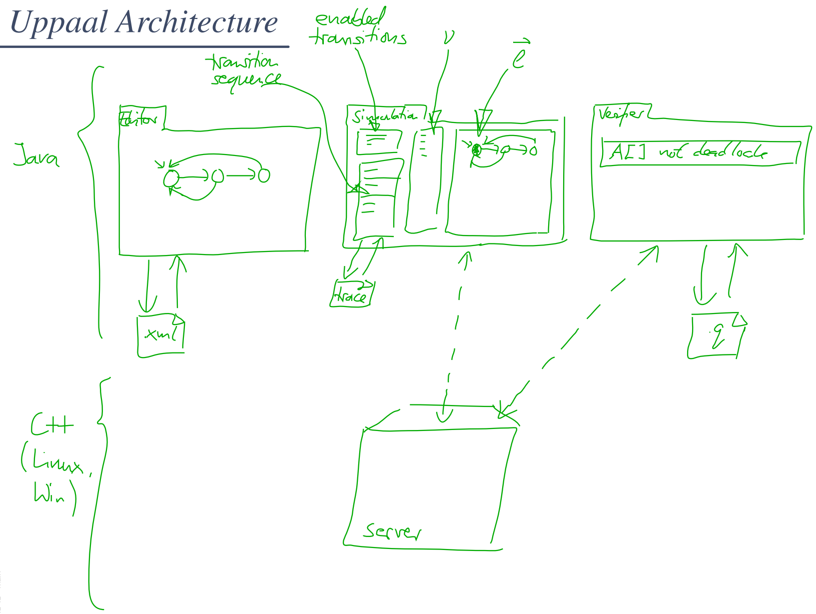

- $\mathcal{N} \models \exists\Diamond\, \mathcal{L}.bright$? ✓
- $\mathcal{N} \models \exists\Box\, \mathcal{L}.bright$? ✗ (we always start in off...)
- $\mathcal{N} \models \exists\Box\, \mathcal{L}.off$? ✓ (stay in off forever)
- $\mathcal{N} \models \forall\Diamond\, \mathcal{L}.light$? ✗ —"—
- $\mathcal{N} \models \forall\Box(\mathcal{L}.bright \implies x \geq 3)$? ✗
- $\mathcal{N} \models (\mathcal{L}.bright \longrightarrow \mathcal{L}.off)$? ...



*Uppaal* Larsen et al. (1997); Behrmann et al. (2004)

*Demo, Vol. 2*

# *Uppaal Architecture*






# *Content*

**Introduction**

- **Observables and Evolutions**
- **Duration Calculus** (DC)
- Semantical Correctness Proofs
- DC Decidability
- DC Implementables
- **PLC-Automata**

- **Timed Automata** (TA), Uppaal
- Networks of Timed Automata
- Region/Zone-Abstraction
- TA model-checking
- Extended Timed Automata ✓
- 2 Undecidability Results

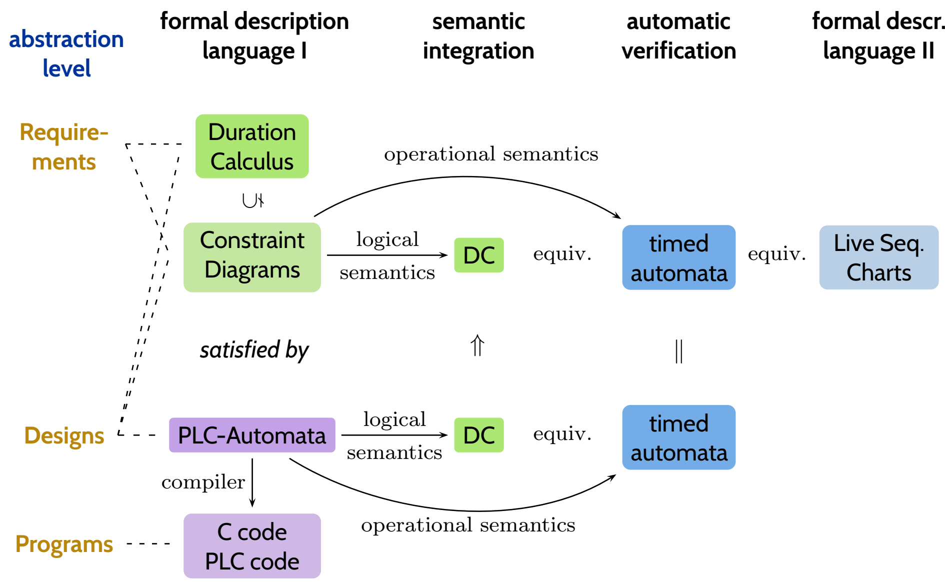
$obs : \text{Time} \to \mathscr{D}(obs)$

$\langle obs_0, \nu_0 \rangle, t_0 \xrightarrow{\lambda_0} \langle obs_1, \nu_1 \rangle, t_1 \ldots$

1

- **Automatic Verification...**
  ...whether a TA satisfies a DC formula, observer-based
- **Recent Results**:
  - **Timed Sequence Diagrams**, or **Quasi-equal Clocks**, or **Automatic Code Generation**, or ...

## *Tying It All Together*







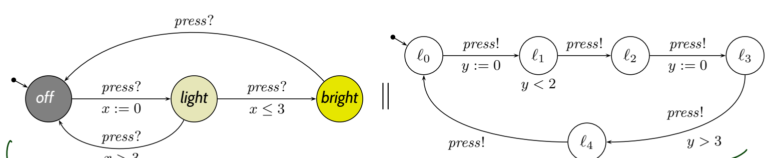

## *Content*

- **Uppaal Query Language**
  - **Syntax**
  - Excursion: **Transition Graph**
  - **Satisfaction Relation**
- A **satisfaction relation** between timed automata and DC formulae
  - **observables** of timed automata
  - **evolution** induced by **computation path**
- A **simple and wrong** solution.
  - **ad-hoc** fix for invariants
- **Testable DC Properties**
  - **observer construction**
  - **untestable DC properties**

# *Observer-based Automatic Verification of DC Properties for Timed Automata*



# *Model-Checking DC Properties with Uppaal*




$\langle \ell, \nu \rangle \longrightarrow \langle \ell, \nu \rangle \longrightarrow \langle \ell, \nu \rangle \cdots$

DC formula

$\models^? \underbrace{F}_{\mathcal{I} : Obs \to (\mathsf{Time} \to \mathcal{D})}$

- **Question 1**: what is the "$\models$"-relation here?
- **Question 2**: what kinds of DC formulae can we check with Uppaal?
  - **Clear**: Not every DC formula.
    (Otherwise contradicting undecidability results.)
  - **Quite clear**: $F = \Box\lceil \mathsf{off} \rceil$ or $F = \neg\Diamond\lceil \mathsf{light} \rceil$
    (Use Uppaal's fragment of TCTL, something like (!) $\forall\Box$ off.)
  - **Maybe**: $F = \ell > 5 \implies \Diamond\lceil \mathsf{off} \rceil^5$
  - **Not so clear**: $F = \neg\Diamond(\lceil \mathsf{bright} \rceil \, ; \lceil \mathsf{light} \rceil)$

# *Observing Timed Automata*



## *Network of TA Satisfies DC Formula*

**Question 1**: what is the "$\models$"-relation here?

What should it mean if we say "network $\mathcal{N}$ satisfies DC formula $F$" (written $\mathcal{N} \models F$)?

**Two main options**:

- Characterise the behaviour of $\mathcal{N}$ by a DC formula $F_\mathcal{N}$ and set

$$\mathcal{N} \models F \quad : \text{iff} \quad \left( \models \underset{DC}{\underset{\Uparrow}{F_\mathcal{N}}} \implies \underset{DC}{\underset{\Uparrow}{F}} \right)$$

  (as we have done for PLC automata).

- "Transform" each **computation paths** $\xi$ of $\mathcal{N}$ into an **evolution** $\mathcal{I}_\xi$ and set

$$\mathcal{N} \models F \quad : \text{iff} \quad \forall\, \xi \bullet \mathcal{I}_\xi \models_0 F$$

  that is, the **evolution** of each **computation path** of $\mathcal{N}$ **realises** $F$ from $0$.

In the following, we shall discuss the **second one**.

# Observables of a Network of Timed Automata

Let $\mathcal{N}$ be a network of $n$ extended timed automata

$$\mathcal{A}_{e,i} = (L_i, C_i, B_i, U_i, X_i, V_i, I_i, E_i, \ell_{ini,i}), \quad 1 \leq i \leq n$$

**For simplicity:** assume that all $L_i$ and $V_i$ are pairwise disjoint (otherwise rename).

**Definition.** The observables Obs($\mathcal{N}$) of $\mathcal{N}$ are

$$\{\ell_1, \dots, \ell_n\} \,\dot{\cup} \bigcup_{1 \leq i \leq n} V_i$$

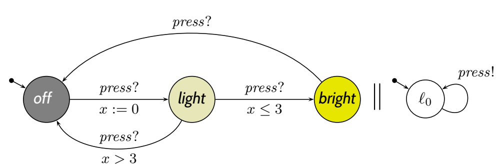
$\{ ☺_1, \ldots, ☺_n \}$

with

- $\mathcal{D}(\ell_i) = L_i$,
- $\mathcal{D}(v)$ is the domain of data-variable $v$ in $\mathcal{A}_{e,i}$.



# Example

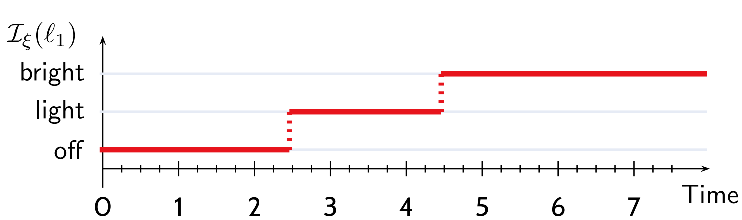



- **Observables**: Obs($\mathcal{N}$) $= \{\ell_1, \ell_2\}$ with
  - $\mathcal{D}(\ell_1) = \{$*off*, *light*, *bright*$\}$, $\quad \mathcal{D}(\ell_2) = \{\ell_0\}$. (No data variables in $\mathcal{N}$!)

Consider **computation path**

$$\xi = \left\langle {\text{off} \atop 0} \right\rangle, 0 \xrightarrow{2.5} \left\langle {\text{off} \atop 2.5} \right\rangle, 2.5 \xrightarrow{\tau} \left\langle {\text{light} \atop 0} \right\rangle, 2.5 \xrightarrow{2.0} \left\langle {\text{light} \atop 2.0} \right\rangle, 4.5 \xrightarrow{\tau} \left\langle {\text{bright} \atop 2.0} \right\rangle, 4.5 \dots$$

and **construct interpretation** $\mathcal{I}_\xi$ : Obs($\mathcal{N}$) $\to$ (Time $\to \mathcal{D}$):



## *Tell Them What You've Told Them. . .*

- Properties **to be checked** for a timed automata model can be specified using the **Uppaal Query Language**,
  - which is a **tiny little fragment** of Timed CTL (TCTL),
  - and as such **by far** not as expressive as Duration Calculus.
- **TCTL** is another **means** to **formalise requirements**.

---

- For **testable** DC formulae $F$, we can automatically verify whether a network $\mathcal{N}$ satisfies $F$.
  - by constructing an **observer automaton**
  - and **transforming** $\mathcal{N}$ appropriately.
- There are **untestable** DC formulae.
  (Everything else would be surprising.)



## *References*

# *References*

Behrmann, G., David, A., and Larsen, K. G. (2004). A tutorial on uppaal 2004-11-17. Technical report, Aalborg University, Denmark.

Larsen, K. G., Pettersson, P., and Yi, W. (1997). UPPAAL in a nutshell. *International Journal on Software Tools for Technology Transfer*, 1(1):134–152.

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