Real-Time Systems

Lecture 11: Networks of Timed Automata

2013-06-11

Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany

Contents & Goals

Last Lecture:

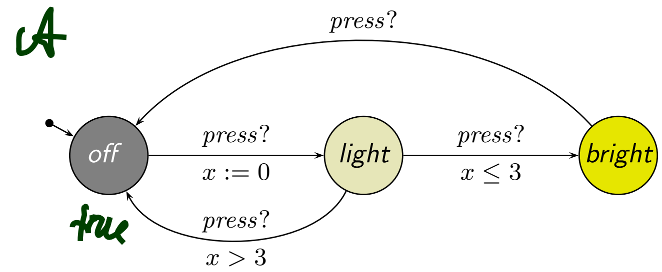
- Timed automata syntax
- TA operational semantics

This Lecture:

- **Educational Objectives:** Capabilities for following tasks/questions.
  - what's the (syntactical) parallel composition of TA?
- **Content:**
  - parallel composition of TA
  - Uppaal demo

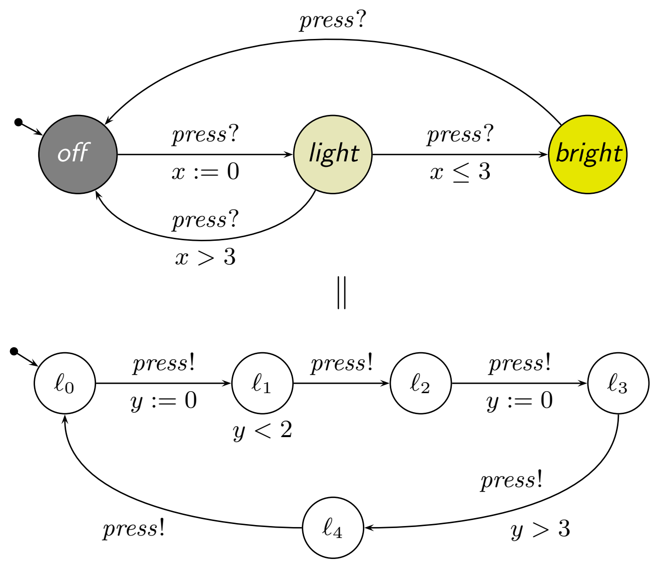
Recall: Plan

- Pure TA syntax
  - channels, actions
  - (simple) clock constraints
  - Def. TA
- Pure TA operational semantics
  - clock valuation, time shift, modification
  - operational semantics
  - discussion
- transition sequence, computation path, run
- network of TA
  - parallel composition (syntactical)
  - restriction
  - network of TA semantics
- Uppaal Demo, part 1
- Extended timed automata



$\mathcal{T}(\mathcal{A}) = (Conf(\mathcal{A}), B_{?!} \cup Time, \{\xrightarrow{\lambda} \mid \lambda \in B_{?!} \cup Time\}, C_{ini})$

$\langle \ell, \nu \rangle$

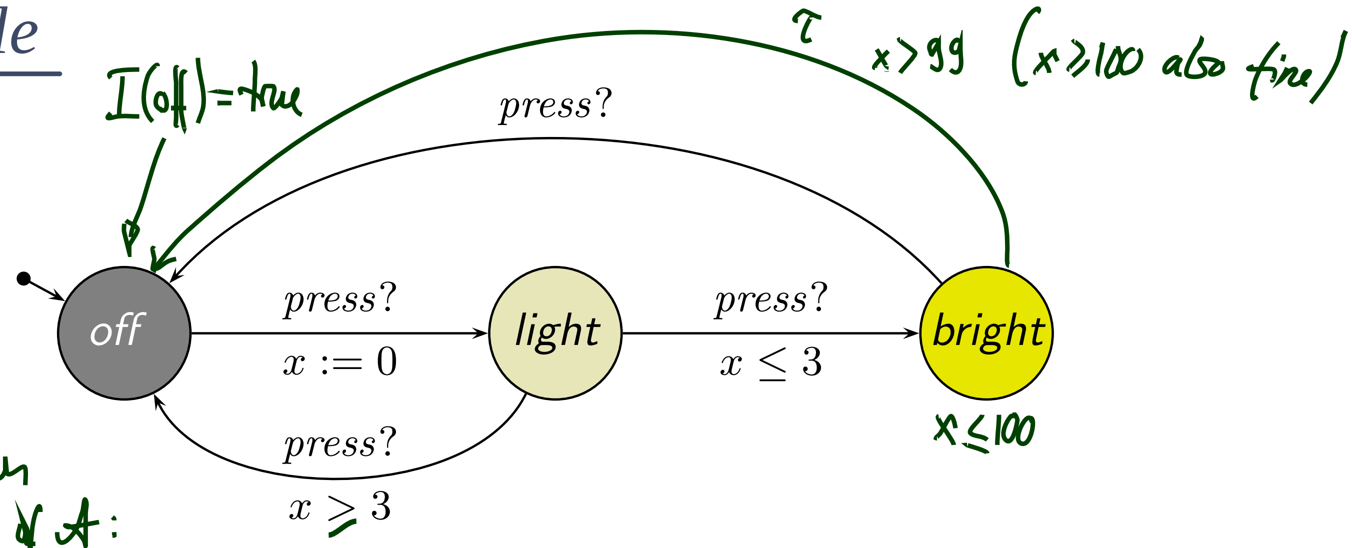


Network of TA

Recall: Pure Timed Automaton

Example

$\mathcal{A}$:



a transition sequence of $\mathcal{A}$:

$$\langle \underbrace{\mathsf{off}}_{\ell_{ini}}, \underbrace{x=0}_{\nu_0}\rangle \xrightarrow{2.5} \langle \mathsf{off}, x=2.5\rangle \xrightarrow{1.7} \langle \mathsf{off}, x=4.2\rangle \xrightarrow{10} \langle \mathsf{off}, x=14.2\rangle \xrightarrow{0} \langle \mathsf{off}, x=14.2\rangle$$

$$\xrightarrow{press?} \langle \mathsf{light}, x=0\rangle \xrightarrow{2.1} \langle \mathsf{light}, x=2.1\rangle$$

$$\xrightarrow{press?} \langle \mathsf{bright}, x=2.1\rangle \xrightarrow{10} \langle \mathsf{bright}, x=12.1\rangle$$

$$\xrightarrow{press?} \langle \mathsf{off}, x=12.1\rangle$$

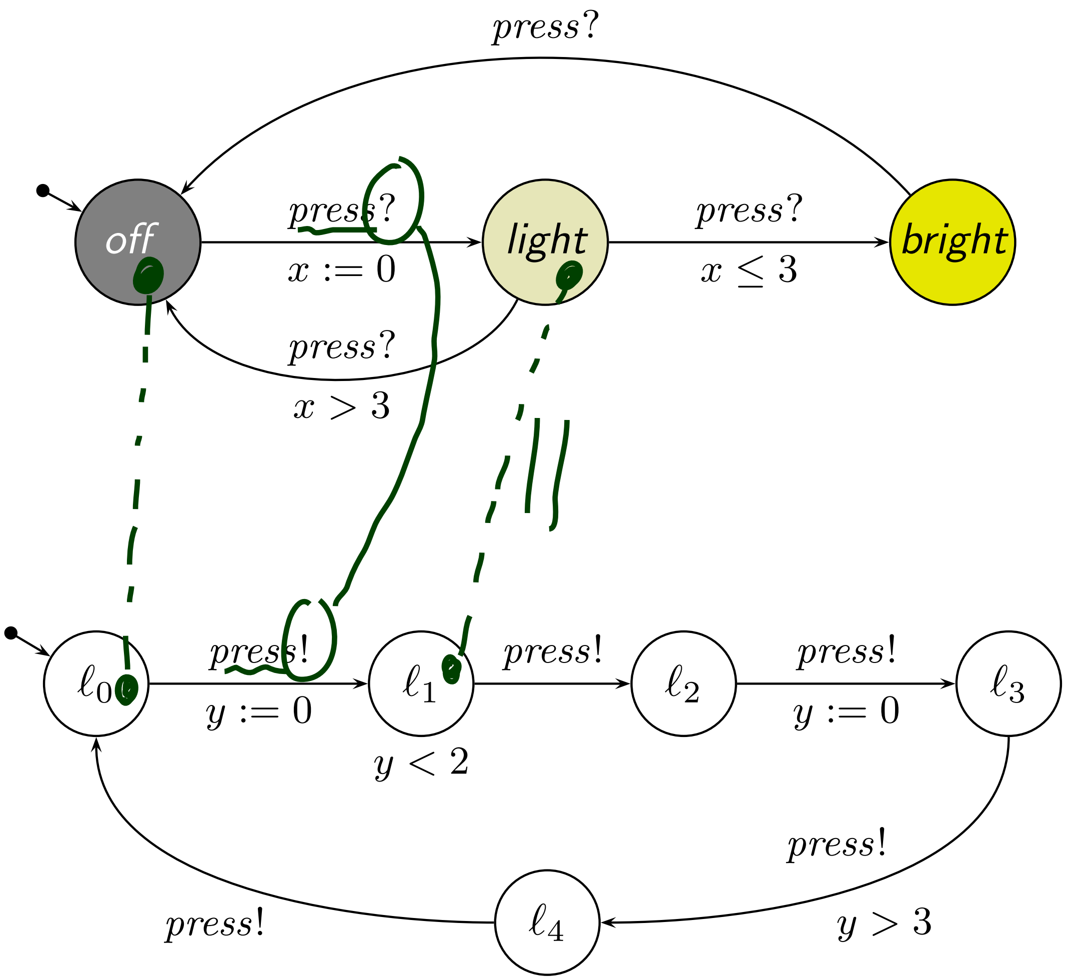
NO NOT A CONFIGURATION

$$\xrightarrow{press?} \langle \mathsf{light}, x=0\rangle \xrightarrow{0} \langle \mathsf{light}, x=0\rangle$$

$$\xrightarrow{press?} \langle bright, x=0\rangle \xrightarrow{127} \langle bright, x=127\rangle$$

$$\xrightarrow{100} \langle b, x=100\rangle \xrightarrow{\tau} \langle off, x=100\rangle$$

Recall: Light Controller and User



Parallel Composition

Definition 4.12.
The **parallel composition** $\mathcal{A}_1 \parallel \mathcal{A}_2$ of two timed automata

$$\mathcal{A}_i = (L_i, B_i, X_i, I_i, E_i, \ell_{ini,i}), \quad i = 1, 2,$$

with **disjoint** sets of clocks X_1 and X_2 yields the timed automaton

$$\mathcal{A} = (L_1 \times L_2, B_1 \cup B_2, X_1 \cup X_2, I, E, (\ell_{ini,1}, \ell_{ini,2}))$$

where

- $I(\ell_1, \ell_2) := I_1(\ell_1) \wedge I_2(\ell_2)$, and
- E consists of **handshake** and **asynchronous communication**.
  (→ **next slide**)

Helper: Action Complementation

- The **complementation function**

$$\overline{\cdot} : Act \to Act$$

  is defined pointwise as

  - $\overline{a!} = a?$
  - $\overline{a?} = a!$
  - $\overline{\tau} = \tau$

- **Note**: $\overline{\overline{\alpha}} = \alpha$ for all $\alpha \in Act$.

Parallel Composition: Handshake and Asynchrony

$\mathcal{A}_1 \parallel \mathcal{A}_2 = (L_1 \times L_2, B_1 \cup B_2, X_1 \cup X_2, I, E, (\ell_{ini,1}, \ell_{ini,2}))$ with

- **Handshake**:

  If there is $a \in B_1 \cup B_2$ such that

  $$(\ell_1, \alpha, \varphi_1, Y_1, \ell_1') \in E_1, \text{ and } (\ell_2, \bar{\alpha}, \varphi_2, Y_2, \ell_2') \in E_2,$$

  and $\{a!, a?\} = \{\alpha, \bar{\alpha}\}$, then

  $$((\ell_1, \ell_2), \tau, \varphi_1 \wedge \varphi_2, Y_1 \cup Y_2, (\ell_1', \ell_2')) \in E.$$

- **Asynchrony**:

  If $(\ell_1, \alpha, \varphi_1, Y_1, \ell_1') \in E_1$ then for all $\ell_2 \in L_2$,

  $$((\ell_1, \ell_2), \alpha, \varphi_1, Y_1, (\ell_1', \ell_2)) \in E.$$

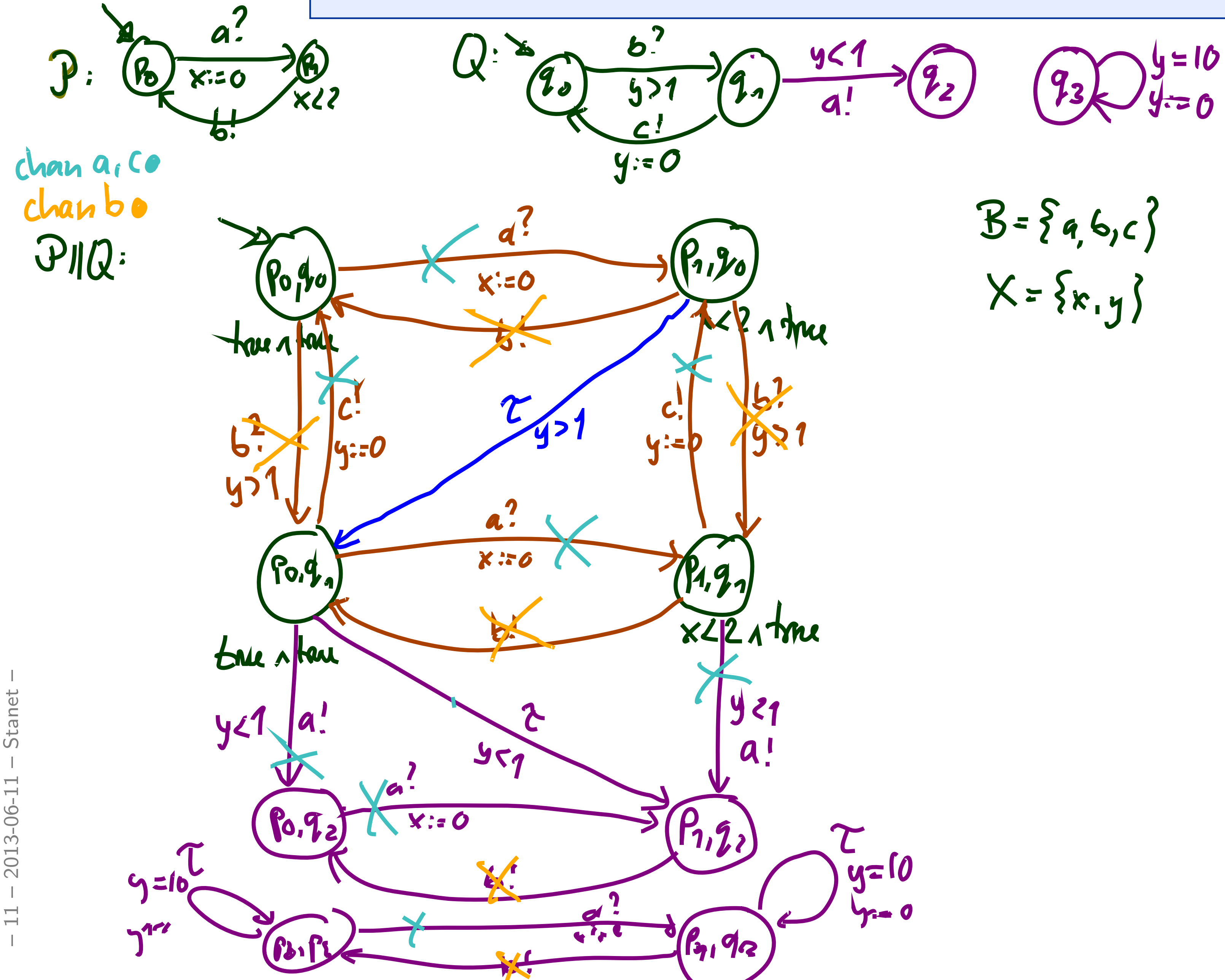
  If $(\ell_2, \alpha, \varphi_2, Y_2, \ell_2') \in E_2$ then for all $\ell_1 \in L_1$,

  $$((\ell_1, \ell_2), \alpha, \varphi_2, Y_2, (\ell_1, \ell_2')) \in E.$$

Example

$\mathcal{L} \parallel \mathcal{U} = (L_1 \times L_2, B_1 \cup B_2, X_1 \cup X_2, I, E, (\ell_{ini,1}, \ell_{ini,2}))$

- If $a \in B_1 \cup B_2$ s.t. $(\ell_1, \alpha, \varphi_1, Y_1, \ell_1') \in E_1$, $(\ell_2, \bar{\alpha}, \varphi_2, Y_2, \ell_2') \in E_2$, $\{a!, a?\} = \{\alpha, \bar{\alpha}\}$, then $((\ell_1, \ell_2), \tau, \varphi_1 \wedge \varphi_2, Y_1 \cup Y_2, (\ell_1', \ell_2')) \in E$
- If $(\ell_1, \alpha, \varphi_1, Y_1, \ell_1') \in E_1$ then f.a. $\ell_2 \in L_2$, $((\ell_1, \ell_2), \alpha, \varphi_1, Y_1, (\ell_1', \ell_2)) \in E$, and conv.



Restriction

Definition 4.13.
A **local channel** b is introduced by the **restriction operator** which, for a timed automaton $\mathcal{A} = (L, B, X, I, E, \ell_{ini})$ yields

$$\mathsf{chan}\, b \bullet \mathcal{A} := (L, B \setminus \{b\}, X, I, E', \ell_{ini})$$

where

- $(\ell, \alpha, \varphi, Y, \ell') \in E'$
  if and only if $(\ell, \alpha, \varphi, Y, \ell') \in E$ and $\alpha \notin \{b!, b?\}$.

- **Abbreviation**:

$$\mathsf{chan}\, b_1 \ldots b_m \bullet \mathcal{A} := \mathsf{chan}\, b_1 \bullet \ldots \mathsf{chan}\, b_m \bullet \mathcal{A}$$

Networks of Timed Automata

- A timed automaton $\mathcal{N}$ is called **network of timed automata** if and only if it is obtained as

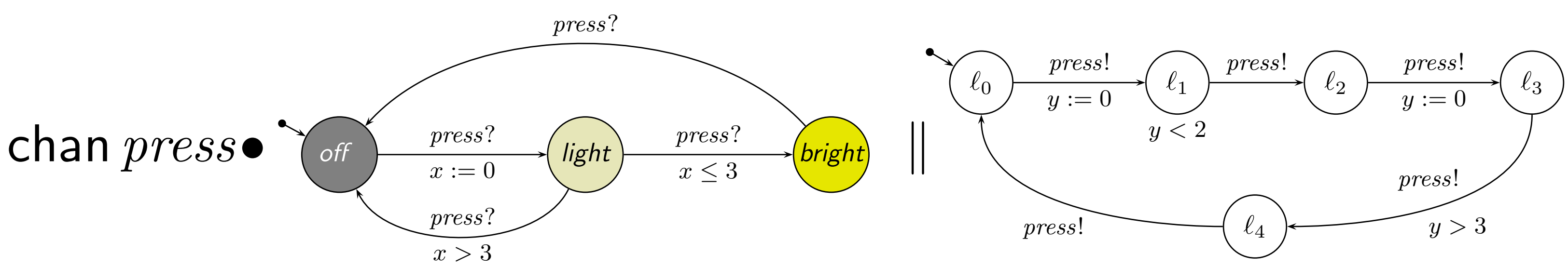
$$\mathsf{chan}\, b_1 \ldots b_m \bullet (\mathcal{A}_1 \parallel \ldots \parallel \mathcal{A}_n)$$

Closed Networks

- A network
$$\mathcal{N} = \mathsf{chan}\, b_1 \dots b_m \bullet (\mathcal{A}_1 \parallel \dots \parallel \mathcal{A}_n)$$
is called **closed** if and only if
$$\{b_1, \dots, b_m\} = \bigcup_{i=1}^{n} B_i.$$

- Then, by Lemma 4.16 (later), **local transitions** don't occur (since $B = \emptyset$). Transitions are thus either internal actions τ or delay transitions.

Example:



is closed.

$$
\begin{tikzcd}[column sep=small]
 & \mathrm{chan}\ b_1,\dots,b_k \bullet A_1 \| \dots \| A_n \arrow[ddd, squiggly, "\text{Lemma 4.16}"'] \arrow[dr, squiggly] & & \\
 & & \mathrm{chan}\ b \bullet B \arrow[d, squiggly] & \\
 & & A_1 \| A_2 \arrow[r, squiggly, "\text{Def. 4.12}"] & A \arrow[d, squiggly, "\text{Def. 4.4}"] \\
 & \mathcal{J}(N) = (\mathrm{Conf}(N), \dots, \{\xrightarrow{?} \mid \dots\}, C_{ini}) & = & \mathcal{J}(A) = (\mathrm{Conf}(A), \dots, \{\xrightarrow{?} \mid \dots\}, C_{ini})
\end{tikzcd}
$$

Lemma 4.16. Let $\mathcal{A}_i = (L_i, B_i, X_i, I_i, E_i, \ell_{ini,i})$ with $i = 1, \dots, n$ be a set of timed automata with disjoint clocks. Then the operational semantics of the network

$$\mathsf{chan}\, b_1 \dots b_m \bullet (\mathcal{A}_1 \parallel \dots \parallel \mathcal{A}_n)$$

yields the labelled transition system

$$(\mathit{Conf}(\mathcal{N}), \mathsf{Time} \cup B_{?!}, \{\xrightarrow{\lambda} \mid \lambda \in \mathsf{Time} \cup B_{?!}\}, C_{ini})$$

with

- $X = \bigcup_{i=1}^{n} X_i$,
- $B = \bigcup_{i=1}^{n} B_i \setminus \{b_1, \dots, b_m\}$,
- $\mathit{Conf}(\mathcal{N}) = \{\langle \vec{\ell}, \nu \rangle \mid$
  $\vec{\ell} \in L_1 \times \dots \times L_n \wedge \nu : X \to \mathsf{Time} \wedge \nu \models \bigwedge_{k=1}^{n} I_k(\ell_k)\}$,
- $C_{ini} = \{\langle (\ell_{ini,1}, \dots, \ell_{ini,n}), \nu_{ini} \rangle\} \cap \mathit{Conf}(\mathcal{N})$
  where $\nu_{ini}(x) = 0$ for all $x \in X$,
- and three types of transition relations ($\rightarrow$ **next slides**).

Operational Semantics of Networks: Local Transitions

For each $\lambda \in \mathsf{Time} \cup B_{!?}$ the transition relation $\xrightarrow{\lambda} \subseteq \mathit{Conf}(\mathcal{N}) \times \mathit{Conf}(\mathcal{N})$ has one of the following three types:

(i) **Local transition**:

$$\langle \vec{\ell}, \nu \rangle \xrightarrow{\alpha} \langle \vec{\ell}', \nu' \rangle$$

if there is $i \in \{1, \dots, n\}$ such that

- $(\ell_i, \alpha, \varphi, Y, \ell_i') \in E_i$, $\alpha \in B_{!?}$, (i-th automaton has corresp. edge)
- $\nu \models \varphi$, (guard is satisfied)
- $\vec{\ell}' = \vec{\ell}[\ell_i := \ell_i']$, (only i-th location changes)
- $\nu' = \nu[Y := 0]$, and ($\mathcal{A}_i$'s clocks are reset)
- $\nu' \models I_i(\ell_i')$. (destination invariant holds)

tuple modification

(ii) **Synchronisation transition**:

$$\langle \vec{\ell}, \nu \rangle \xrightarrow{\tau} \langle \vec{\ell}', \nu' \rangle$$

if there are $i, j \in \{1, \dots, n\}$, $i \neq j$, and $b \in B_i \cap B_j$, such that

- $(\ell_i, b!, \varphi_i, Y_i, \ell_i') \in E_i$ and $(\ell_j, b?, \varphi_j, Y_j, \ell_j') \in E_j$,
- $\nu \models \varphi_i \wedge \varphi_j$,
- $\vec{\ell}' = \vec{\ell}[\ell_i := \ell_i'][\ell_j := \ell_j']$,
- $\nu' = \nu[Y_i \cup Y_j := 0]$, and
- $\nu' \models I_i(\ell_i') \wedge I_j(\ell_j')$.

Operational Semantics of Networks: Delay

(iii) **Delay transition**:

$$\langle \vec{\ell}, \nu \rangle \xrightarrow{t} \langle \vec{\ell}, \nu + t \rangle$$

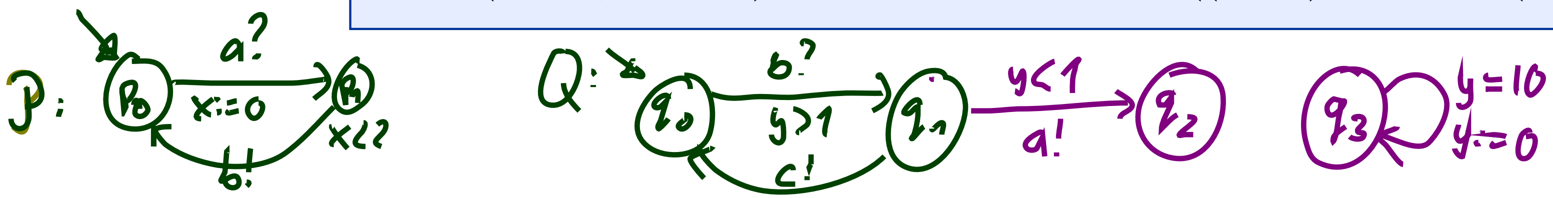
if for all $t' \in [0, t]$,

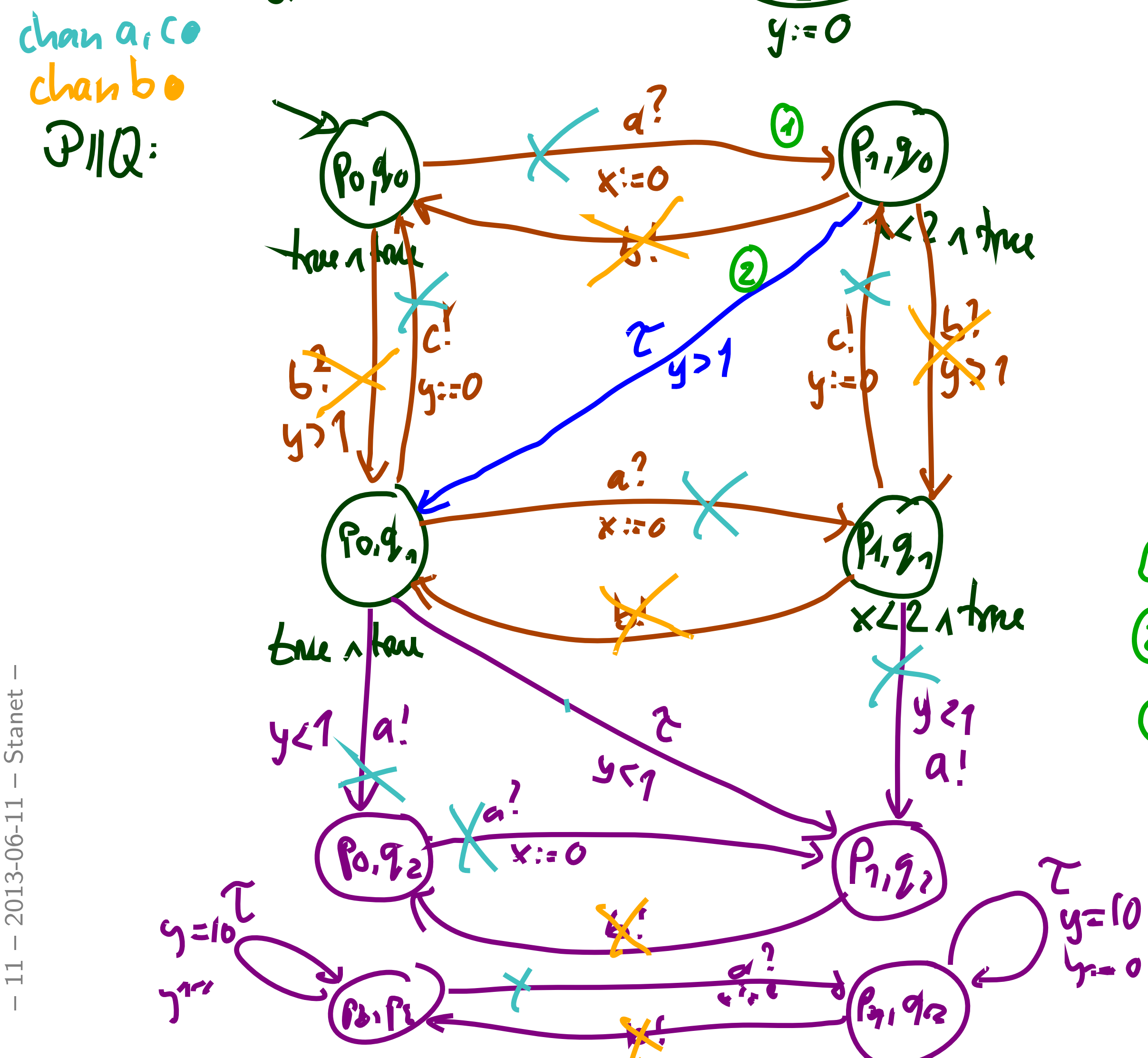
- $\nu + t' \models \bigwedge_{k=1}^{n} I_k(\ell_k)$.

Example

$\mathcal{L} \parallel \mathcal{U} = (L_1 \times L_2, B_1 \cup B_2, X_1 \cup X_2, I, E, (\ell_{ini,1}, \ell_{ini,2}))$

- If $a \in B_1 \cup B_2$ s.t. $(\ell_1, \alpha, \varphi_1, Y_1, \ell_1') \in E_1$, $(\ell_2, \bar{\alpha}, \varphi_2, Y_2, \ell_2') \in E_2$, $\{a!, a?\} = \{\alpha, \bar{\alpha}\}$, then $((\ell_1, \ell_2), \tau, \varphi_1 \wedge \varphi_2, Y_1 \cup Y_2, (\ell_1', \ell_2')) \in E$
- If $(\ell_1, \alpha, \varphi_1, Y_1, \ell_1') \in E_1$ then f.a. $\ell_2 \in L_2$, $((\ell_1, \ell_2), \alpha, \varphi_1, Y_1, (\ell_1', \ell_2)) \in E$, and conv.





$B = \{a, b, c\}$

$X = \{x, y\}$

Example transition sequence of chan b ▸ P∥Q:

$\langle (p_0, q_0), \begin{smallmatrix} x=0 \\ y=0 \end{smallmatrix} \rangle \xrightarrow{27} \langle (p_0, q_0), \begin{smallmatrix} x=27 \\ y=27 \end{smallmatrix} \rangle$

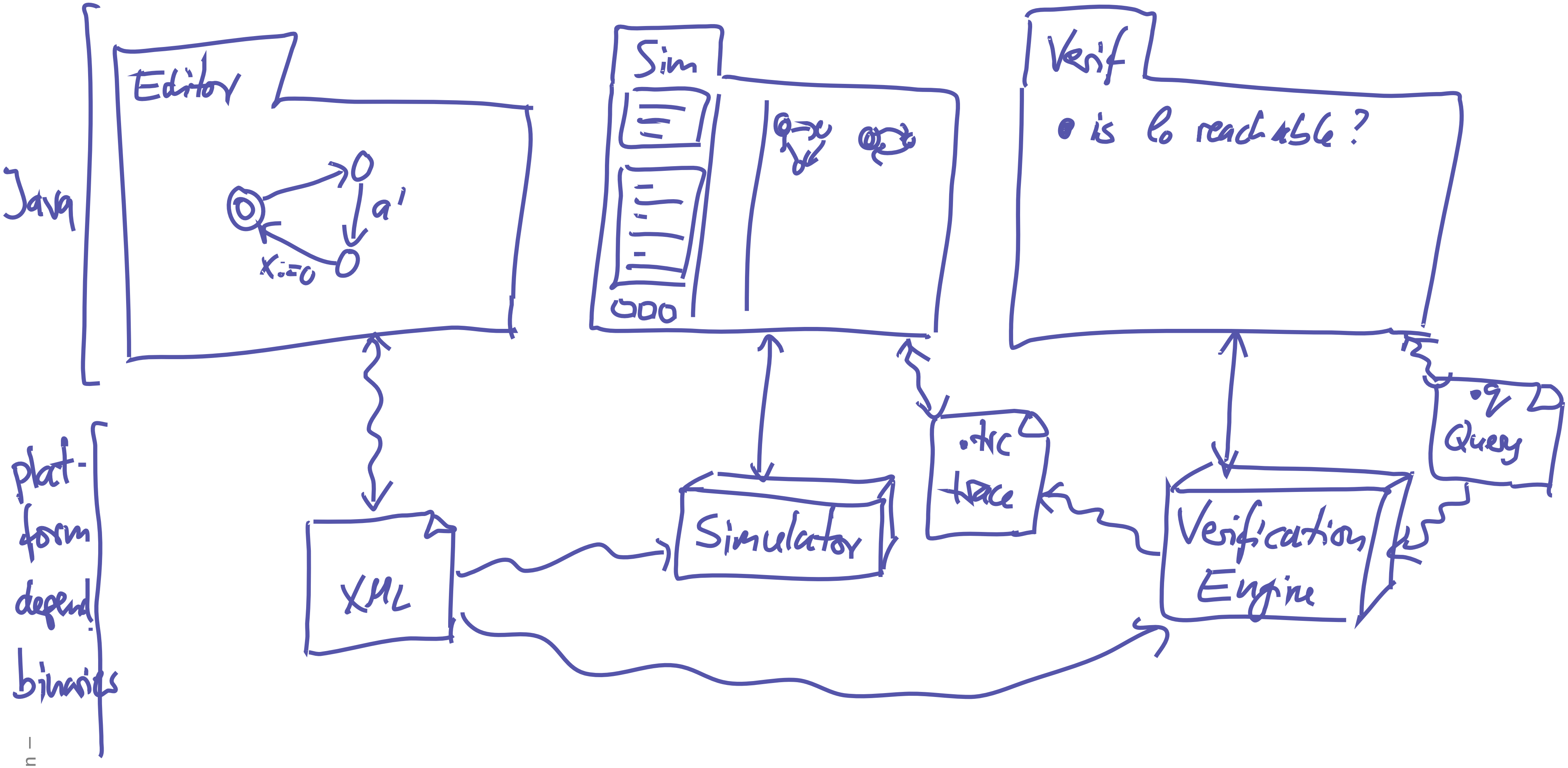
① $\xrightarrow{a?} \langle (p_1, q_0), \begin{smallmatrix} x=0 \\ y=27 \end{smallmatrix} \rangle$

② $\xrightarrow{\tau} \langle (p_0, q_1), \begin{smallmatrix} x=0 \\ y=27 \end{smallmatrix} \rangle \xrightarrow{10} \langle (p_0, q_1), \begin{smallmatrix} x=10 \\ y=37 \end{smallmatrix} \rangle$

③ $\xrightarrow{c!} \langle (p_0, q_0), \begin{smallmatrix} x=10 \\ y=0 \end{smallmatrix} \rangle$ ...

Uppaal [Larsen et al., 1997, Behrmann et al., 2004]
Demo, Vol. 1

Uppaal Architecture



References

References

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

[Larsen et al., 1997] 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 and Dierks, 2008] Olderog, E.-R. and Dierks, H. (2008). *Real-Time Systems - Formal Specification and Automatic Verification*. Cambridge University Press.