Real-Time Systems

Lecture 12: Networks of Timed Automata

2017-12-12

Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany

Content

- **Parallel Composition** of TA
  - **handshake** edges
  - **asynchronous** edges
- **Restriction** / **Channel Hiding**
- **Networks** of Timed Automata
  - **closed** networks
- **Operational Semantics** of Networks of Timed Automata
  - a **semantical** approach
- The **Uppaal** tool
  - **Demo I**:
    - **Model Editor**
    - **Simulator**

Parallel Composition of TA

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

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

Helper: Action Complementation

- The **complementation function**

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

is defined pointwise as follows:

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

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

Parallel Composition: Handshake and Asynchrony

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

- **Handshake Edges**:
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 $\{\alpha, \bar{\alpha}\} = \{a!, a?\}$, **then**

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

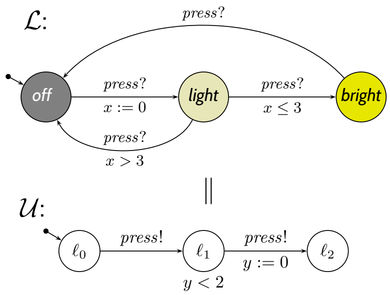
- **Asynchronous Edges**:
If $(\ell_1, \alpha, \varphi_1, Y_1, \ell_1') \in E_1$ **then** for all $\ell_2 \in L_2$,

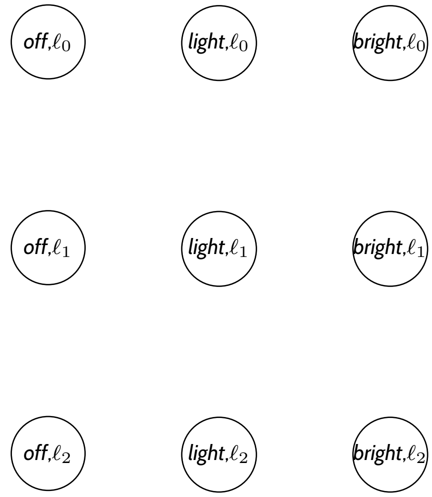
$$((\ell_1, \ell_2), \alpha, \varphi_1, Y_1, (\ell_1', \ell_2)) \in E.$$

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

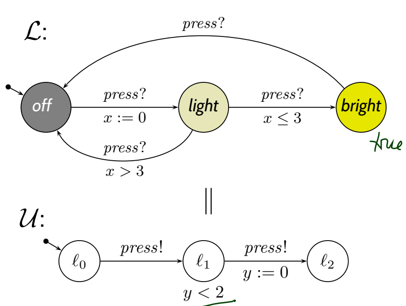


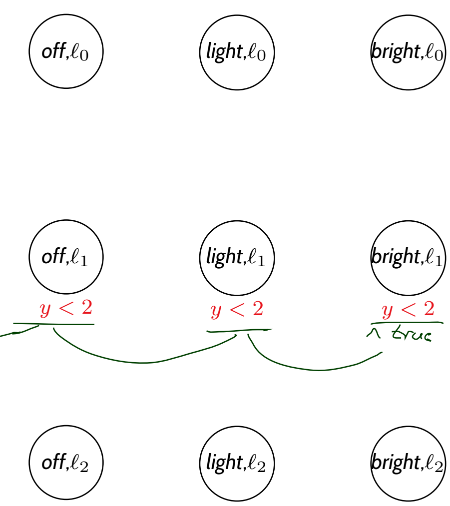


$\mathcal{L} \,\|\, \mathcal{U} =$
$(L_1 \times L_2, B_1 \cup B_2, X_1 \cup X_2, I, E, (\ell_{ini,1}, \ell_{ini,2}))$

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

Example

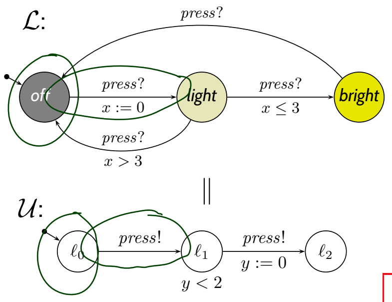


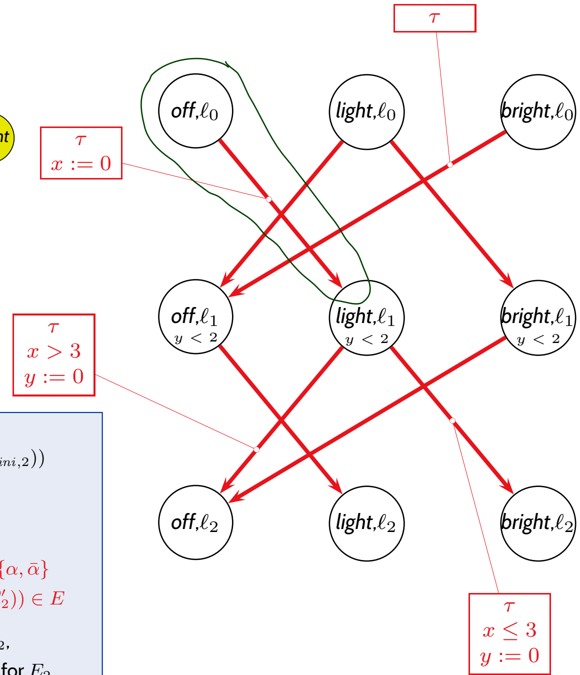


$\mathcal{L} \,\|\, \mathcal{U} =$
$(L_1 \times L_2, B_1 \cup B_2, X_1 \cup X_2, I, E, (\ell_{ini,1}, \ell_{ini,2}))$

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

Example

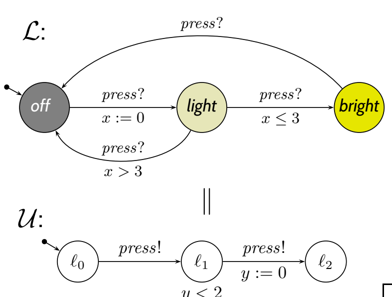


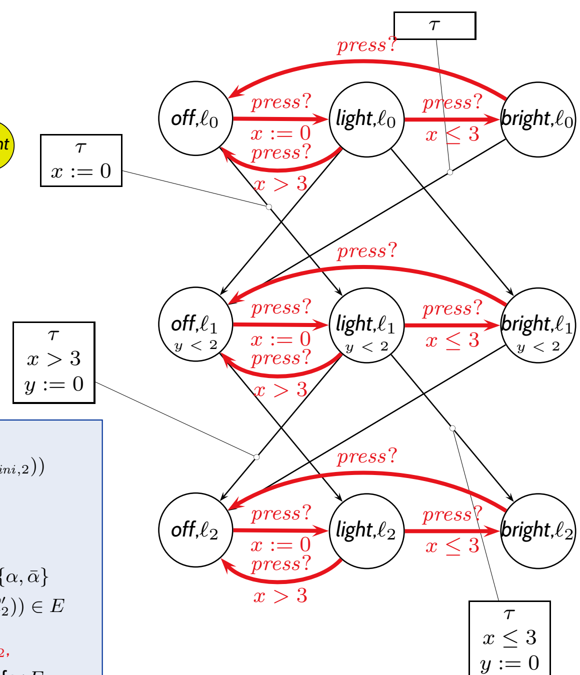


$\mathcal{L} \,\|\, \mathcal{U} =$
$(L_1 \times L_2, B_1 \cup B_2, X_1 \cup X_2, I, E, (\ell_{ini,1}, \ell_{ini,2}))$

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

Example

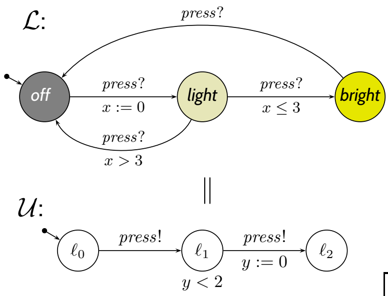


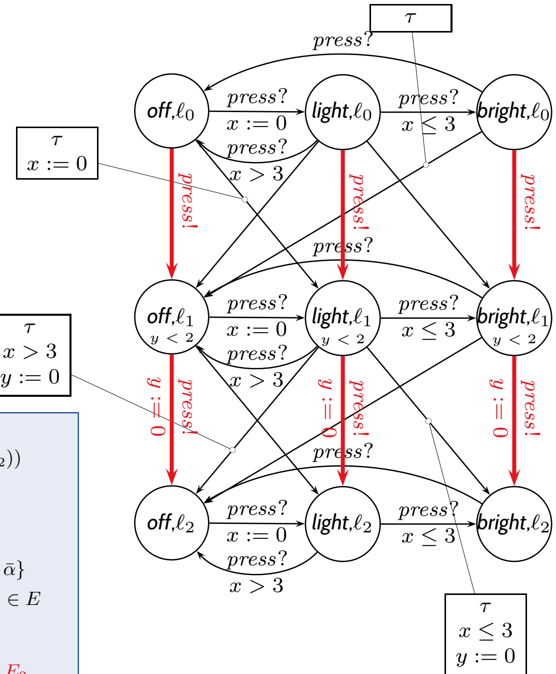


$\mathcal{L} \,\|\, \mathcal{U} =$
$(L_1 \times L_2, B_1 \cup B_2, X_1 \cup X_2, I, E, (\ell_{ini,1}, \ell_{ini,2}))$

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

Example

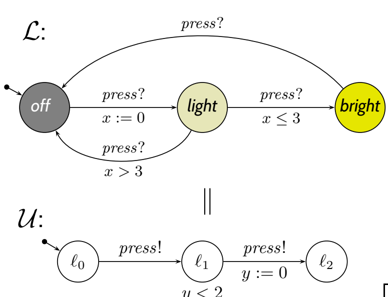


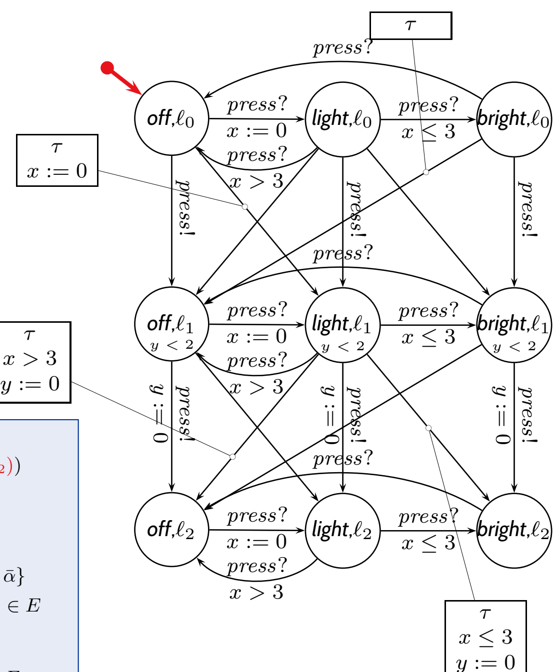


$\mathcal{L} \parallel \mathcal{U} =$
$(L_1 \times L_2, B_1 \cup B_2, X_1 \cup X_2, I, E, (\ell_{ini,1}, \ell_{ini,2}))$

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

Example

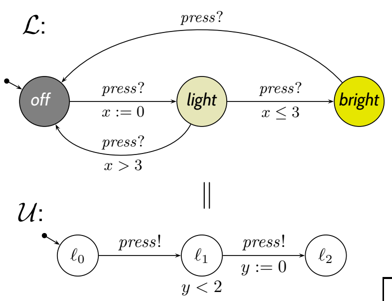


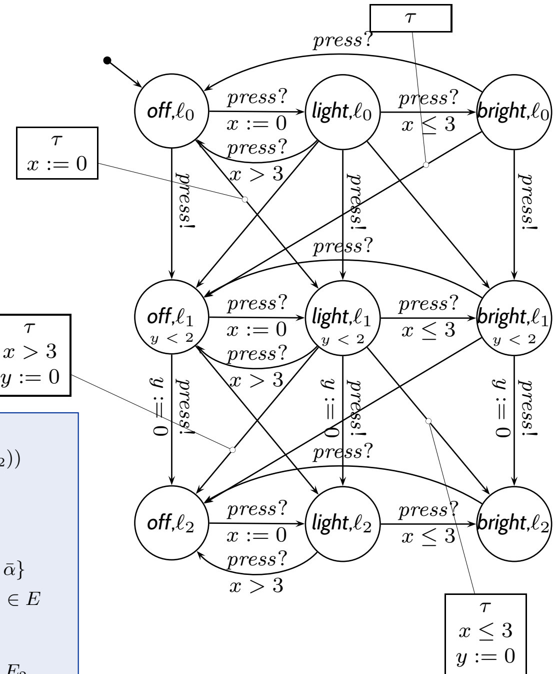


$\mathcal{L} \parallel \mathcal{U} =$
$(L_1 \times L_2, B_1 \cup B_2, X_1 \cup X_2, I, E, (\ell_{ini,1}, \ell_{ini,2}))$

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

Example





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

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

Content

- **Parallel Composition** of TA
  - **handshake** edges
  - **asynchronous** edges
- **Restriction** / **Channel Hiding**
- **Networks** of Timed Automata
  - **closed** networks
- **Operational Semantics** of Networks of Timed Automata
  - a **semantical** approach
- The **Uppaal** tool
  - **Demo I**:
    - **Model Editor**
    - **Simulator**

Restriction / Channel Hiding

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

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

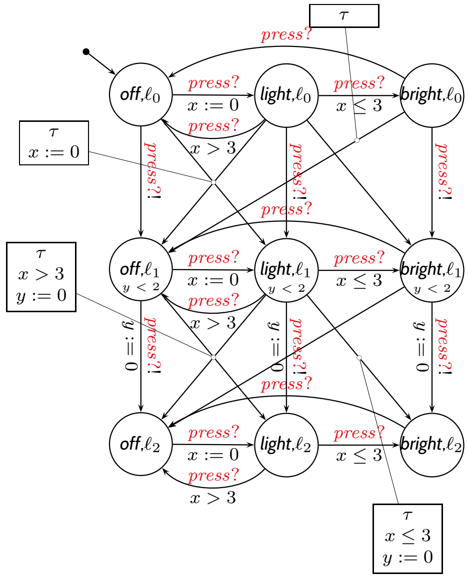
where

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

- **Abbreviation**:

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

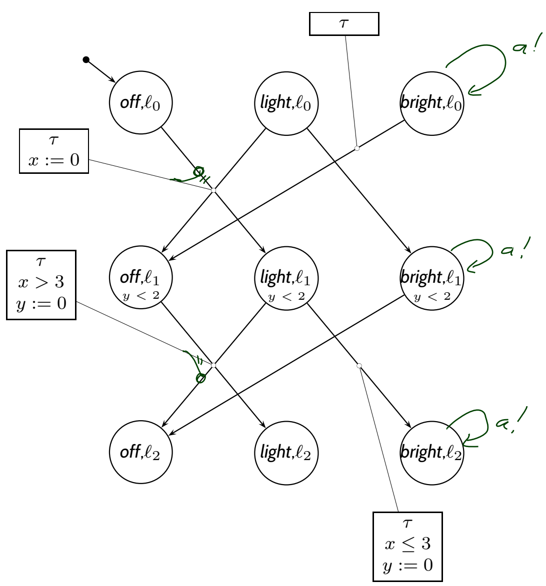
Example



chan $\textcolor{red}{press}$ • $\mathcal{L} \parallel \mathcal{U}$

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

Example



chan $press$ • $\mathcal{L} \parallel \mathcal{U}$

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

Content

* **Parallel Composition** of TA
  * **handshake** edges
  * **asynchronous** edges
* **Restriction** / **Channel Hiding**
* **Networks** of Timed Automata
  * **closed** networks
* **Operational Semantics** of Networks of Timed Automata
  * a **semantical** approach
* The **Uppaal** tool
  * **Demo I**:
    * **Model Editor**
    * **Simulator**

Network of TA

Networks of Timed Automata

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

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

Closed Networks

- A network

$$\mathcal{N} = \text{chan } b_1 \dots b_m \bullet (\mathcal{A}_1 \parallel \dots \parallel \mathcal{A}_n)$$

  is called **closed** if and only if

$$\{b_1, \dots, b_m\} = \bigcup_{i=1}^{n} B_i$$

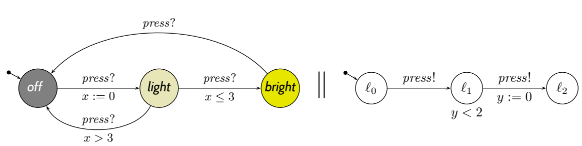
  where B_i is the alphabet of $\mathcal{A}_i$.

- 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:

chan $press$ $\bullet$

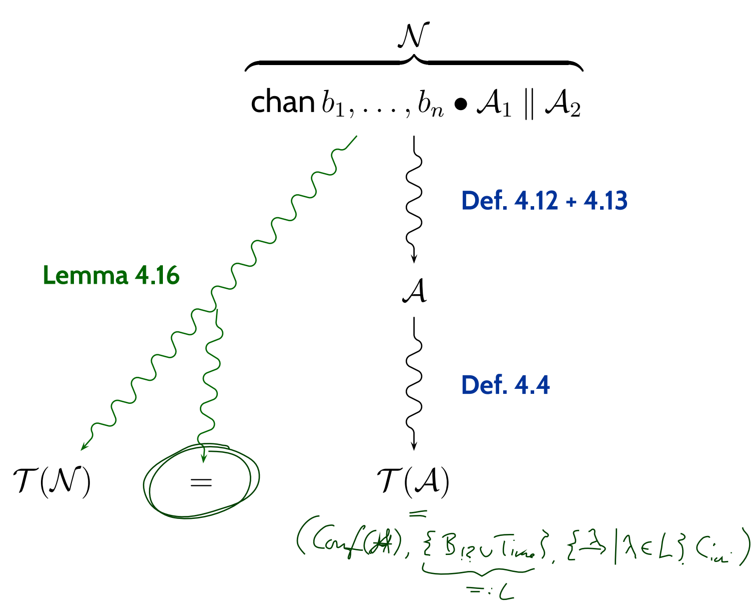


is closed.

Content

- **Parallel Composition** of TA
  - **handshake** edges
  - **asynchronous** edges
- **Restriction** / **Channel Hiding**
- **Networks** of Timed Automata
  - **closed** networks
- **Operational Semantics** of Networks of Timed Automata
  - a **semantical** approach
- The **Uppaal** tool
  - **Demo I**:
    - **Model Editor**
    - **Simulator**

Operational Semantics of Networks of TA: The Plan



Operational Semantics of Networks

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

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

yields the labelled transition system

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

with

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

Op. Semantics of Networks: Local Transitions

For each $\lambda \in \mathsf{Time} \cup B_{!?}$ the transition relation $\xrightarrow{\lambda} \subseteq Conf(\mathcal{N}) \times Conf(\mathcal{N})$ 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)

Op. Semantics of Networks: Synchronisation

(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
- $\nu' \models I_i(\ell_i') \wedge I_j(\ell_j')$.

Op. Semantics of Networks: Delay

(iii) **Delay transition**:

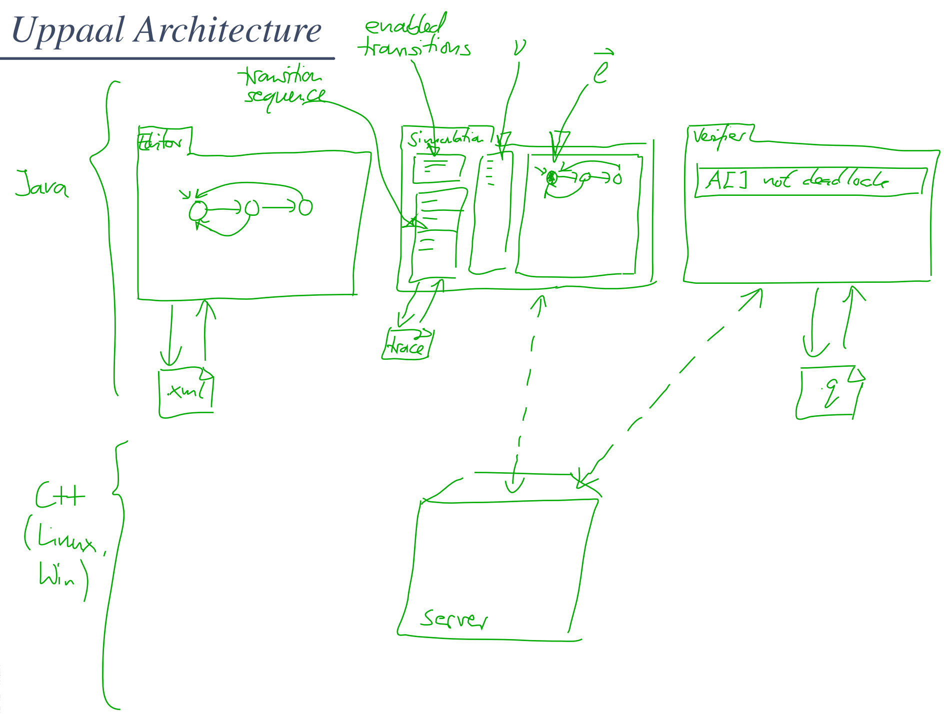
$$\langle \vec{\ell}, \nu \rangle \xrightarrow{t} \langle \vec{\ell}, \nu + t \rangle$$

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

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

Uppaal Larsen et al. (1997); Behrmann et al. (2004)
Demo, Vol. 1

Uppaal Architecture



Tell Them What You've Told Them...

- The **parallel composition**
  - of two **timed automata**
  - is again a **timed automaton**.

  IOW: the set of timed automata
  is **closed under parallel composition**.

- **Channel restriction** introduces **local channels**.
  - Hiding **all channels** yields a **closed network**.
  - Uppaal always interprets a network as **closed**.

- Behaviour of a **network** can alternatively
  be characterised **semantically**.

- The **Uppaal** tool is one way to
  model and **simulate** (networks of) timed automata.

  (And to **verify** $\rightarrow$ next lecture(s).)

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.