

Real-Time Systems

Lecture 14: Extended Timed Automata

2012-07-10

Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany

Contents & Goals

Last Lecture:

- Decidability of the location reachability problem: zones.
- Extended TA: data variables.

This Lecture:

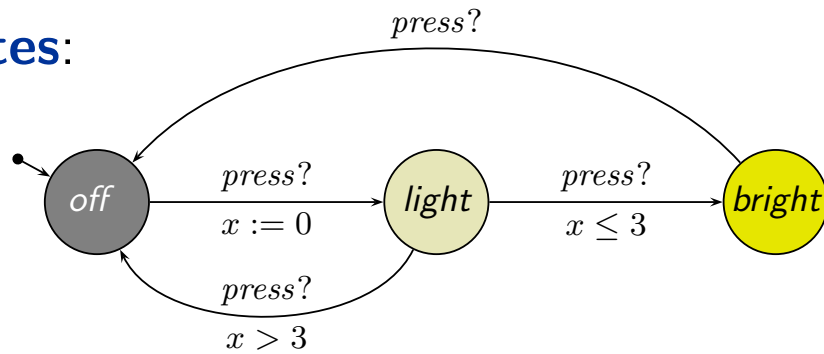
- **Educational Objectives:** Capabilities for following tasks/questions.
 - By what are TA extended? Why is that useful?
 - What's an urgent/committed location? What's the difference?
 - What's an urgent channel?
 - Where has the notion of “input action” and “output action” correspondences in the formal semantics?
- **Content:**
 - Extended TA:
 - Structuring Facilities
 - Restriction of Non-Determinism
 - The Logic of Uppaal

Extended Timed Automata

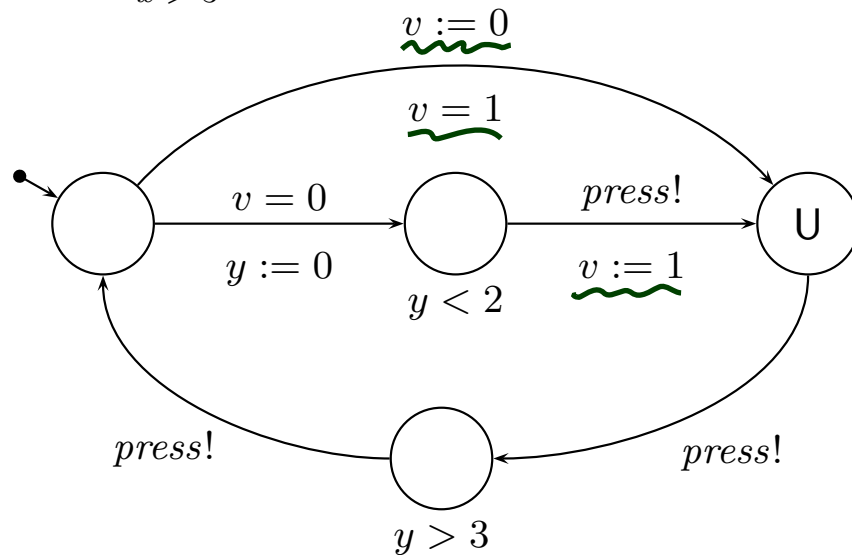
Recall: Example (Partly Already Seen in Uppaal Demo)

Templates:

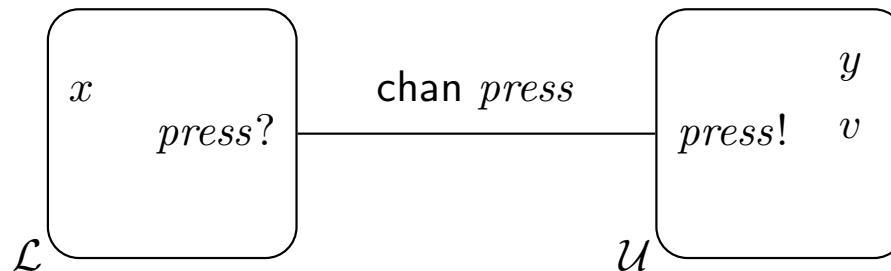
• \mathcal{L} :



• \mathcal{U} :



System:



Extensions:

- Data Variables (Expressions, Constraints, Updates)
- Structuring
- Urgent/Committed Location, Urgent Channel

Recall: Data Variables and Expressions

- Let $(v, w \in) V$ be a set of (integer) variables.
 - $(\psi_{int} \in) \Psi(V)$: **integer expressions** over V using func. symb. $+, -, \dots$
 - $(\varphi_{int} \in) \Phi(V)$: **integer (or data) constraints** over V using **integer expressions**, predicate symbols $=, <, \leq, \dots$, and boolean logical connectives. (incl. $\vee, \wedge, \Rightarrow, \Leftrightarrow, \neg, \dots$)
- Recall: $\Phi(X)$ for clocks X (simple clock constraint)
- Let $(x, y \in) X$ be a set of clocks.
 - $(\varphi \in) \Phi(X, V)$: (**extended**) **guards**, defined by

$$\varphi ::= \varphi_{clk} \mid \varphi_{int} \mid \varphi_1 \wedge \varphi_2$$

where $\varphi_{clk} \in \Phi(X)$ is a simple clock constraint (as defined before) and $\varphi_{int} \in \Phi(V)$ an **integer (or data) constraint**.

Examples: Extended guard or not extended guard? Why?

- $x - y < 0$
- (a) $\underbrace{x < y}_{\in \Phi(X)} \wedge \underbrace{v > 2}_{\in \Phi(V)}$ ✓
- (b) $\underbrace{x < y}_{\in \Phi(X)} \vee \underbrace{v > 2}_{\in \Phi(V)}$ ✗ (no)
- (c) $\underbrace{v < 1 \vee v > 2}_{\in \Phi(V)}$ ✓
- (d) $\underbrace{x < v}_{\in \Phi(X)} \wedge \underbrace{v > 2}_{\in \Phi(V)}$ ✗

Modification or Reset Operation

- **New:** a **modification** or **reset operation** is

$$x := 0, \quad x \in X,$$

or

$$v := \psi_{int}, \quad v \in V, \quad \psi_{int} \in \Psi(V).$$

- By $R(X, V)$ we denote the set of all resets.
- By \vec{r} we denote a finite list $\langle r_1, \dots, r_n \rangle$, $n \in \mathbb{N}_0$, of reset operations $r_i \in R(X, V)$; $\langle \rangle$ is the empty list.
- By $R(X, V)^*$ we denote the set of all such lists of reset operations.

Modification or Reset Operation

OLD:
 $(l, \alpha, \varphi, \gamma, l')$
 NEW:
 $(l, \alpha, \varphi, \vec{r}, l')$

- **New:** a **modification** or **reset (operation)** is

$$x := 0, \quad x \in X,$$

or

$$v := \psi_{int}, \quad v \in V, \quad \psi_{int} \in \Psi(V).$$

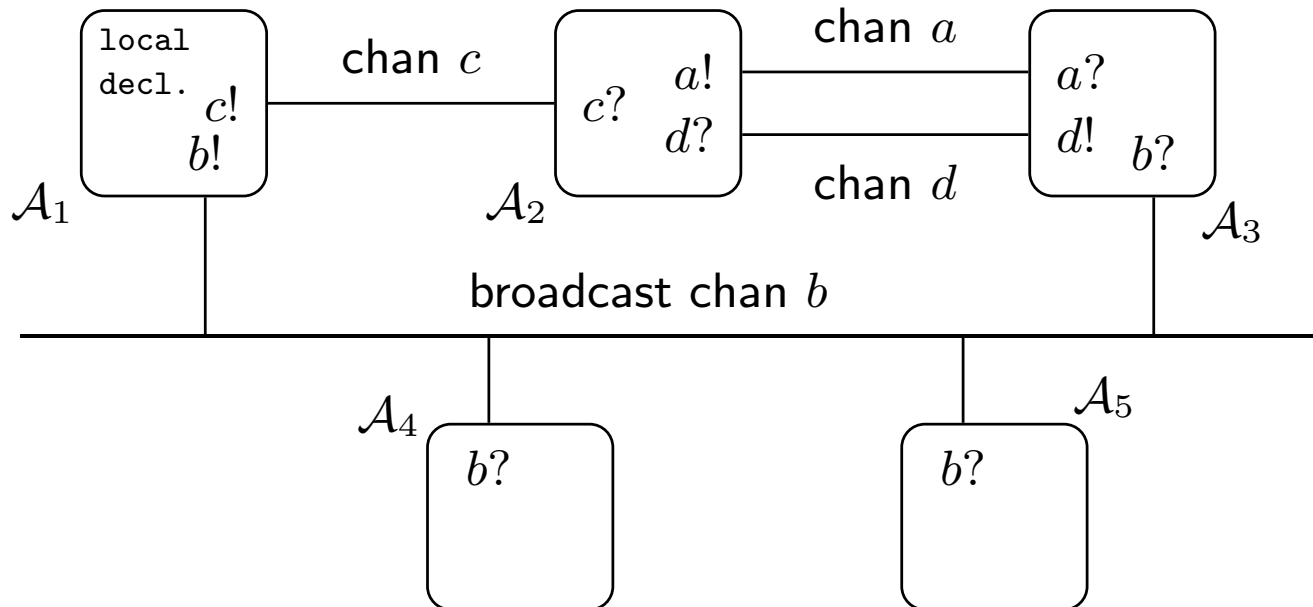
- By $R(X, V)$ we denote the set of all resets.
- By \vec{r} we denote a finite list $\langle r_1, \dots, r_n \rangle$, $n \in \mathbb{N}_0$, of reset operations $r_i \in R(X, V)$; $\langle \rangle$ is the empty list.
- By $R(X, V)^*$ we denote the set of all such lists of reset operations.

Examples: Modification or not? Why?

(a) $x := y,$	(b) $x := v,$	(c) $v := x,$	(d) $v := w,$	(e) $v := 0$
clock ↗ not 0 ↗	not 0 ↗	$\notin \Psi(V)$	$\in V$ $\in \Psi(V)$	$\in \Psi(V)$
X	X	X	✓	✓

Structuring Facilities

global decl.: clocks, variables, channels, constants



- Global declarations of of clocks, data variables, channels, and constants.
- Binary and broadcast channels: $\text{chan } c$ and $\text{broadcast chan } b$.
- Templates of timed automata.
- Instantiation of templates (instances are called **process**).
- System definition: list of processes.

Restricting Non-determinism

- **Urgent locations** — enforce local immediate progress.

U

- **Committed locations** — enforce **atomic** immediate progress.

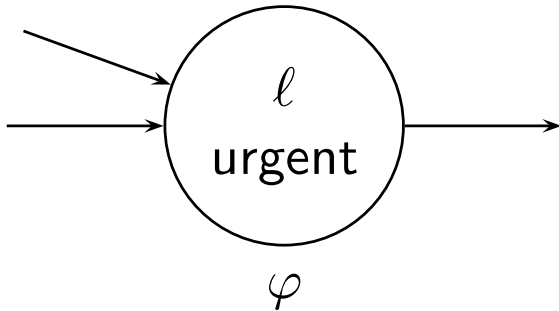
C

- **Urgent channels** — enforce cooperative immediate progress.

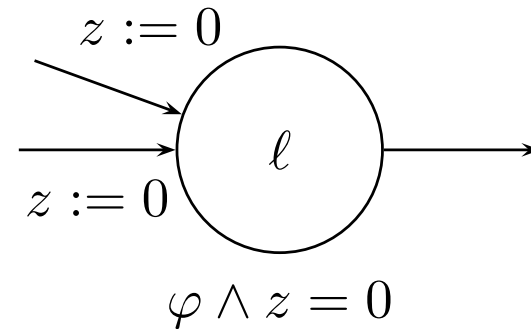
urgent chan press;

Urgent Locations: Only an Abbreviation...

Replace



with



where z is a fresh clock:

- reset z on all in-going edges,
- add $z = 0$ to invariant.

$N=3$ • 1
 $|U|=20$ • 3 ✓
each ant. •
at least one • 20

Question: How many fresh clocks do we need in the worst case for a network of N extended timed automata?

Definition 4.39. An **extended timed automaton** is a structure

$$\mathcal{A}_e = (L, C, B, U, X, V, I, E, \ell_{ini})$$

where L, B, X, I, ℓ_{ini} are as in Def. 4.3, except that location invariants in I are **downward closed**, and where

- $C \subseteq L$: **committed locations**,
- $U \subseteq B$: **urgent channels**,
- V : a set of data variables,
- $E \subseteq L \times B_{!?} \times \Phi(X, V) \times R(X, V)^* \times L$: a set of **directed edges** such that

$$(\ell, \alpha, \varphi, \vec{r}, \ell') \in E \wedge \text{chan}(\alpha) \in U \implies \varphi = \text{true}.$$

Edges $(\ell, \alpha, \varphi, \vec{r}, \ell')$ from location ℓ to ℓ' are labelled with an **action** α , a **guard** φ , and a list \vec{r} of **reset operations**.

Definition 4.40. Let $\mathcal{A}_{e,i} = (L_i, C_i, B_i, U_i, X_i, V_i, I_i, E_i, \ell_{ini,i})$, $1 \leq i \leq n$, be extended timed automata with pairwise disjoint sets of clocks X_i .

The operational semantics of $\mathcal{C}(\mathcal{A}_{e,1}, \dots, \mathcal{A}_{e,n})$ (closed!) is the labelled transition system

$$\begin{aligned} & \mathcal{T}_e(\mathcal{C}(\mathcal{A}_{e,1}, \dots, \mathcal{A}_{e,n})) \\ &= (\text{Conf}, \text{Time} \cup \{\tau\}, \{\xrightarrow{\lambda} \mid \lambda \in \text{Time} \cup \{\tau\}\}, C_{ini}) \end{aligned}$$

where

- $X = \bigcup_{i=1}^n X_i$ and $V = \bigcup_{i=1}^n V_i$, $\mathcal{D}(V)$
- $\text{Conf} = \{ \langle \vec{\ell}, \nu \rangle \mid \ell_i \in L_i, \nu : X \cup V \rightarrow \text{Time}, \nu \models \bigwedge_{k=1}^n I_k(\ell_k) \}$,
- $C_{ini} = \{ \langle \vec{\ell}_{ini}, \nu_{ini} \rangle \} \cap \text{Conf}$,

and the transition relation consists of transitions of the following three types.

Helpers: Extended Valuations and Timeshift

- **Now:** $\nu : X \cup V \rightarrow \text{Time} \cup \mathcal{D}(V)$
- Canonically extends to $\nu : \Psi(V) \rightarrow \mathcal{D}$ (valuation of expression).
- “ \models ” extends canonically to expressions from $\Phi(X, V)$.

$$\begin{aligned} & \bullet \eta ::= \nu \mid f(v_1, \dots, v_n) \\ & \quad \nu \in V, f \in \mathcal{F} \\ & \bullet \mathcal{D}(V) \subseteq \mathcal{Z} \\ & \bullet \mathcal{I}: \mathcal{F} \rightarrow \bigcup_{n \in \mathbb{N}} (\mathbb{Z}^n \rightarrow \mathcal{Z}) \\ & \bullet \nu: V \rightarrow \mathcal{D}(V) \\ & \bullet \nu(f(v_1, \dots, v_n)) := \overline{\mathcal{I}(f)}(\nu(v_1), \dots, \nu(v_n)) \end{aligned}$$

Helpers: Extended Valuations and Timeshift

- **Now:** $\nu : X \cup V \rightarrow \text{Time} \cup \mathcal{D}(V)$
- Canonically extends to $\nu : \Psi(V) \rightarrow \mathcal{D}$ (valuation of expression).
- “ \models ” extends canonically to expressions from $\Phi(X, V)$.
- Extended **timeshift** $\nu + t$, $t \in \text{Time}$, applies to clocks only:
 - $(\nu + t)(x) := \nu(x) + t$, $x \in X$,
 - $(\nu + t)(v) := \nu(v)$, $v \in V$.
- **Effect of modification** $r \in R(X, V)$ on ν , denoted by $\nu[r]$:

$$\nu[x := 0](a) := \begin{cases} 0, & \text{if } a = x, \\ \nu(a), & \text{otherwise} \end{cases}$$



$$\nu[v := \psi_{int}](a) := \begin{cases} \nu(\psi_{int}), & \text{if } a = v, \\ \nu(a), & \text{otherwise} \end{cases}$$

- We set $\nu[\langle r_1, \dots, r_n \rangle] := \nu[r_1] \dots [r_n] = (((\underline{\nu[r_1]})[r_2]) \dots)[r_n]$.

Operational Semantics of Networks: Internal Transitions

- An **internal transition** $\langle \vec{l}, \nu \rangle \xrightarrow{\tau} \langle \vec{l}', \nu' \rangle$ occurs if there is $i \in \{1, \dots, n\}$ *no. of automata* such that
 - there is a τ -edge $(l_i, \tau, \varphi, \vec{r}, l'_i) \in E_i$,
 - $\nu \models \varphi$,
 - $\vec{l}' = \vec{l}[l_i := l'_i]$, *location of the i-th automaton in \vec{l}* *modification of i-th position*
 - $\nu' = \nu[\vec{r}]$,
 - $\nu' \models I_i(l'_i)$,
 - (**♣**) if $l_k \in C_k$ for some $k \in \{1, \dots, n\}$ then $l_i \in C_i$. *committed locations*

Operational Semantics of Networks: Synchronisation Transition

- A **synchronisation transition** $\langle \vec{l}, \nu \rangle \xrightarrow{\tau} \langle \vec{l}', \nu' \rangle$ occurs if there are $i, j \in \{1, \dots, n\}$ with $i \neq j$ such that
 - there are edges $(l_i, b!, \varphi_i, \vec{r}_i, l'_i) \in E_i$ and $(l_j, b?, \varphi_j, \vec{r}_j, l'_j) \in E_j$,
 - $\nu \models \varphi_i \wedge \varphi_j$,
 - $\vec{l}' = \vec{l}[l_i := l'_i][l_j := l'_j]$,
 - $\nu' = \nu[\vec{r}_i][\vec{r}_j]$,  "sender updates are applied first"
 - $\nu' \models I_i(l'_i) \wedge I_j(l'_j)$,
 - () if $l_k \in C_k$ for some $k \in \{1, \dots, n\}$ then $l_i \in C_i$ or $l_j \in C_j$.

Operational Semantics of Networks: Delay Transitions

- A **delay transition** $\langle \vec{l}, \nu \rangle \xrightarrow{t} \langle \vec{l}, \nu + t \rangle$ occurs if

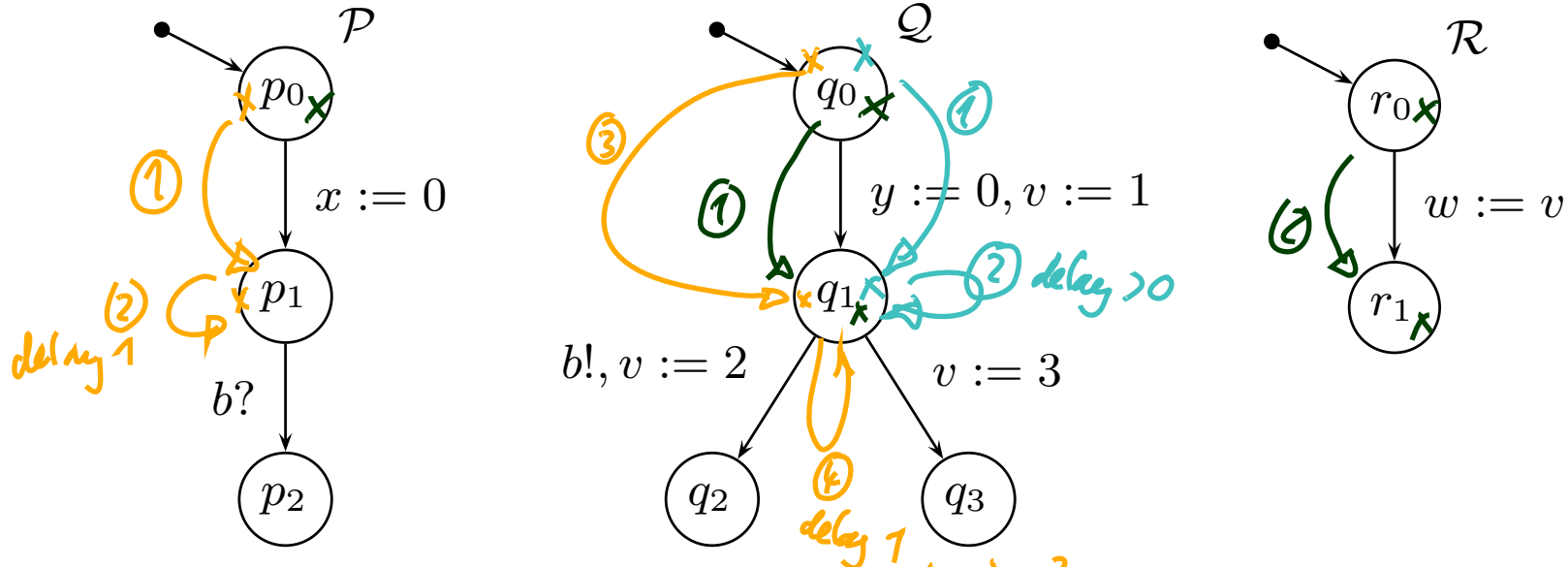
- $\nu + t \models \bigwedge_{k=1}^n I_k(l_k),$

- (\clubsuit) there are no $i, j \in \{1, \dots, n\}$ and $b \in U$ with $(\underline{l_i}, \underline{b!}, \varphi_i, \vec{r}_i, l'_i) \in E_i$ and $(\underline{l_j}, \underline{b?}, \varphi_j, \vec{r}_j, l'_j) \in E_j,$

- (\clubsuit) there is no $i \in \{1, \dots, n\}$ such that $l_i \in C_i.$

urgent channels
↓

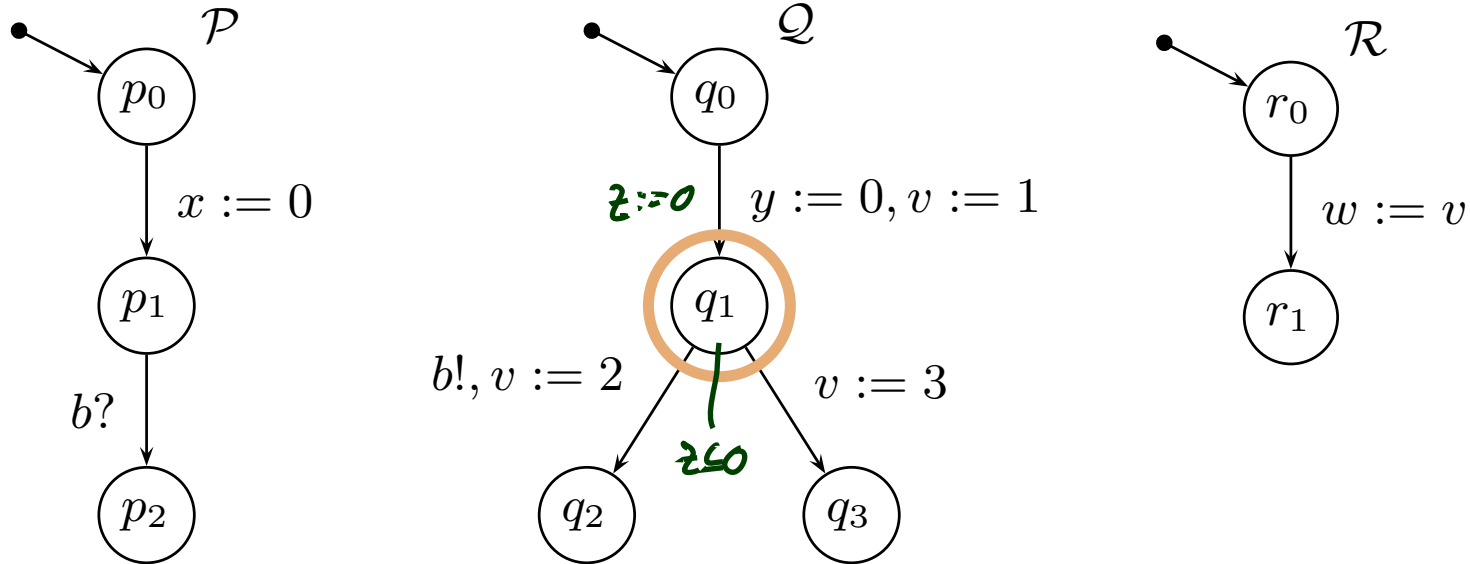
Restricting Non-determinism: Example



config. readability
in all reachable config

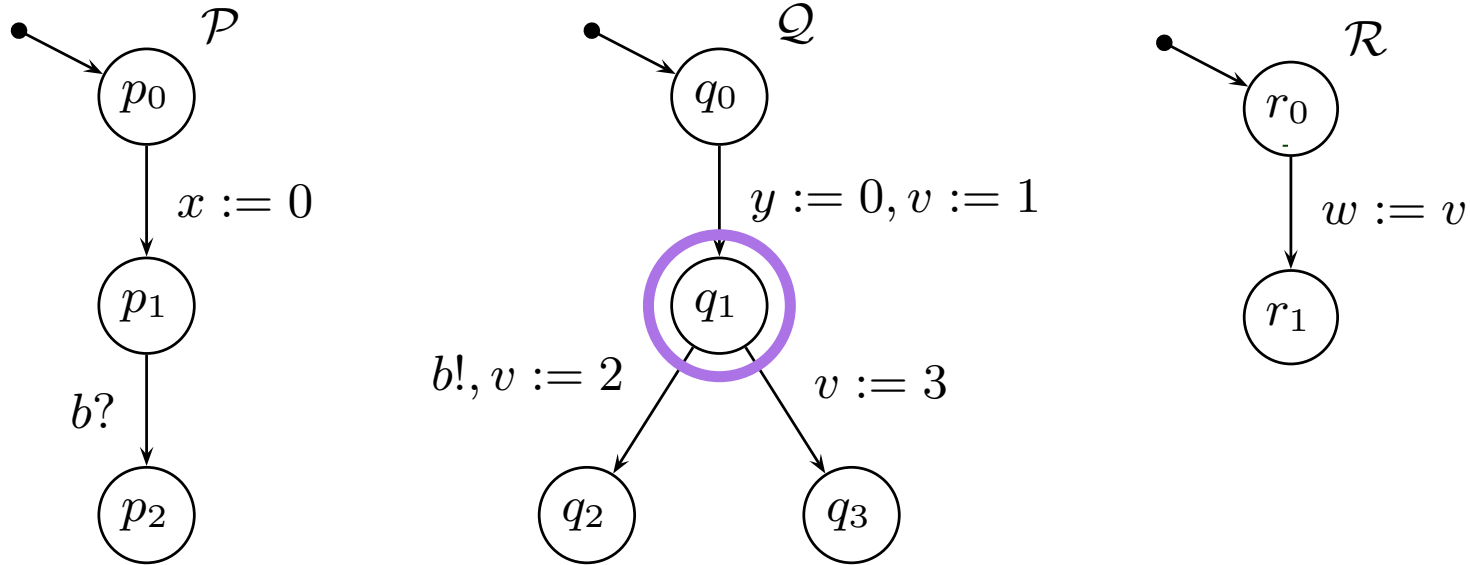
	Property 1 $\exists \diamond w = 1$	Property 2 $\forall \square (Q.q_1 \implies y \leq 0)$	Property 3 $\forall \square (\mathcal{P}.p_1 \wedge Q.q_1 \implies (x \geq y \implies y \leq 0))$
$\mathcal{N} := \mathcal{P} \parallel \mathcal{Q} \parallel \mathcal{R}$	✓	✗	✗
\mathcal{N}, q_1 urgent			
\mathcal{N}, q_1 comm.			
\mathcal{N}, b urgent			

Restricting Non-determinism: Urgent Location



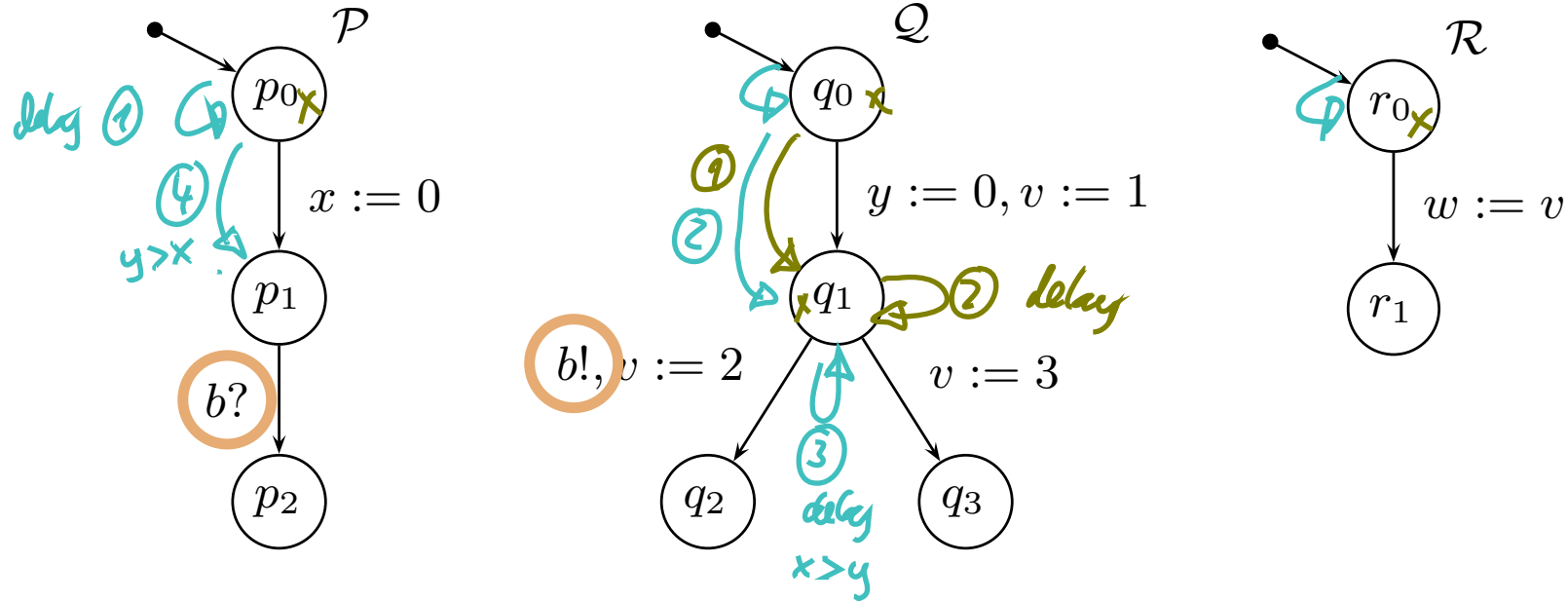
	Property 1	Property 2	Property 3
	$\exists \diamond w = 1$	$\forall \square \mathcal{Q}.q_1 \implies y \leq 0$	$\forall \square (\mathcal{P}.p_1 \wedge \mathcal{Q}.q_1 \implies (x \geq y \implies y \leq 0))$
\mathcal{N}	✓	✗	✗
\mathcal{N}, q_1 urgent	✓	✓	✓
\mathcal{N}, q_1 comm.			
\mathcal{N}, b urgent			

Restricting Non-determinism: Committed Location



	Property 1	Property 2	Property 3
	$\exists \diamond w = 1$	$\forall \square \mathcal{Q}.q_1 \implies y \leq 0$	$\forall \square (\mathcal{P}.p_1 \wedge \mathcal{Q}.q_1 \implies (x \geq y \implies y \leq 0))$
\mathcal{N}	✓	✗	✗
\mathcal{N}, q_1 urgent	✓	✓	✓
\mathcal{N}, q_1 comm.	✗	✓	✓
\mathcal{N}, b urgent			

Restricting Non-determinism: Urgent Channel



	Property 1	Property 2	Property 3
	$\exists \diamond w = 1$	$\forall \square Q.q_1 \implies y \leq 0$	$\forall \square (\mathcal{P}.p_1 \wedge Q.q_1 \implies (x \geq y \implies y \leq 0))$
\mathcal{N}	✓	✗	✗
\mathcal{N}, q_1 urgent	✓	✓	✓
\mathcal{N}, q_1 comm.	✗	✓	✓
\mathcal{N}, b urgent	✓	✗	✓

Extended vs. Pure Timed Automata

Extended vs. Pure Timed Automata

$$\mathcal{A}_e = (L, C, B, U, X, V, I, E, \ell_{ini})$$

$$(\ell, \alpha, \varphi, \vec{r}, \ell') \in L \times B_{!?} \times \Phi(X, V) \times R(X, V)^* \times L$$

vs.

$$\mathcal{A} = (L, B, X, I, E, \ell_{ini})$$

$$(\ell, \alpha, \varphi, Y, \ell') \in E \subseteq L \times B_{?!} \times \Phi(X) \times 2^X \times L$$

- \mathcal{A}_e is in fact (or specialises to) a **pure** timed automaton if
 - $C = \emptyset$,
 - $U = \emptyset$,
 - $V = \emptyset$,
 - for each $\vec{r} = \langle r_1, \dots, r_n \rangle$, every r_i is of the form $x := 0$ with $x \in X$.
- $I(\ell), \varphi \in \Phi(X)$ is then a consequence of $V = \emptyset$.

Operational Semantics of Extended TA

Theorem 4.41. If $\mathcal{A}_1, \dots, \mathcal{A}_n$ specialise to pure timed automata, then the operational semantics of

$$\mathcal{C}(\mathcal{A}_1, \dots, \mathcal{A}_n)$$

and

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

where $\{b_1, \dots, b_m\} = \bigcup_{i=1}^n B_i$, **coincide**, i.e.

$$\mathcal{T}_e(\mathcal{C}(\mathcal{A}_1, \dots, \mathcal{A}_n)) = \mathcal{T}(\text{chan } b_1, \dots, b_m \bullet (\mathcal{A}_1 \parallel \dots \parallel \mathcal{A}_n)).$$

today

per TA

Reachability Problems for Extended Timed Automata

Theorem 4.33. [*Location Reachability*] The location reachability problem for **pure** timed automata is **decidable**.

Theorem 4.34. [*Constraint Reachability*] The constraint reachability problem for **pure** timed automata is **decidable**.

- And what about \hat{W} **extended** timed automata?

What About Extended Timed Automata?

Extended Timed Automata add the following features:

- **Data-Variables**

- As long as the domains of all variables in V are finite, adding data variables doesn't hurt.
- If they're infinite, we've got a problem (encode two-counter machine).

- **Structuring Facilities**

- Don't hurt — they're merely abbreviations.

- **Restricting Non-determinism**

- Restricting non-determinism doesn't affect the configuration space.
- Restricting non-determinism only **removes** certain transitions, so makes region automaton even smaller.

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:**

$$atom ::= \mathcal{A}_i.l \mid \varphi$$

where $l \in L_i$ 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”)

$$e\text{-formula} ::= \exists \diamond term \mid \exists \square term$$

- **universal path formulae:** (“always finally”, “always globally”, “leads to”)

$$a\text{-formula} ::= \forall \diamond term \mid \forall \square term \mid term_1 \longrightarrow term_2$$

- **formulae:**

$$F ::= e\text{-formula} \mid a\text{-formula}$$

Configurations at Time t

- Recall: **computation path** (or path) **starting in** $\langle \vec{\ell}_0, \nu_0 \rangle, t_0$:

$$\xi = \langle \vec{\ell}_0, \nu_0 \rangle, t_0 \xrightarrow{\lambda_1} \langle \vec{\ell}_1, \nu_1 \rangle, t_1 \xrightarrow{\lambda_2} \langle \vec{\ell}_2, \nu_2 \rangle, t_2 \xrightarrow{\lambda_3} \dots$$

$\in \text{Time} \setminus \{0\}$

which is **infinite or maximally finite**.

0

3.0

3.0

- Given ξ and $t \in \text{Time}$, 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?

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

$$\xi(0.27) = \{ \langle \vec{\ell}_0, \nu_0 + 0.27 \rangle \}$$

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

Satisfaction of Uppaal-Logic by Configurations

- We define a **satisfaction relation**

$$\langle \vec{l}_0, \nu_0 \rangle, t_0 \models F$$

between **time stamped configurations**

$$\langle \vec{l}_0, \nu_0 \rangle, t_0$$

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

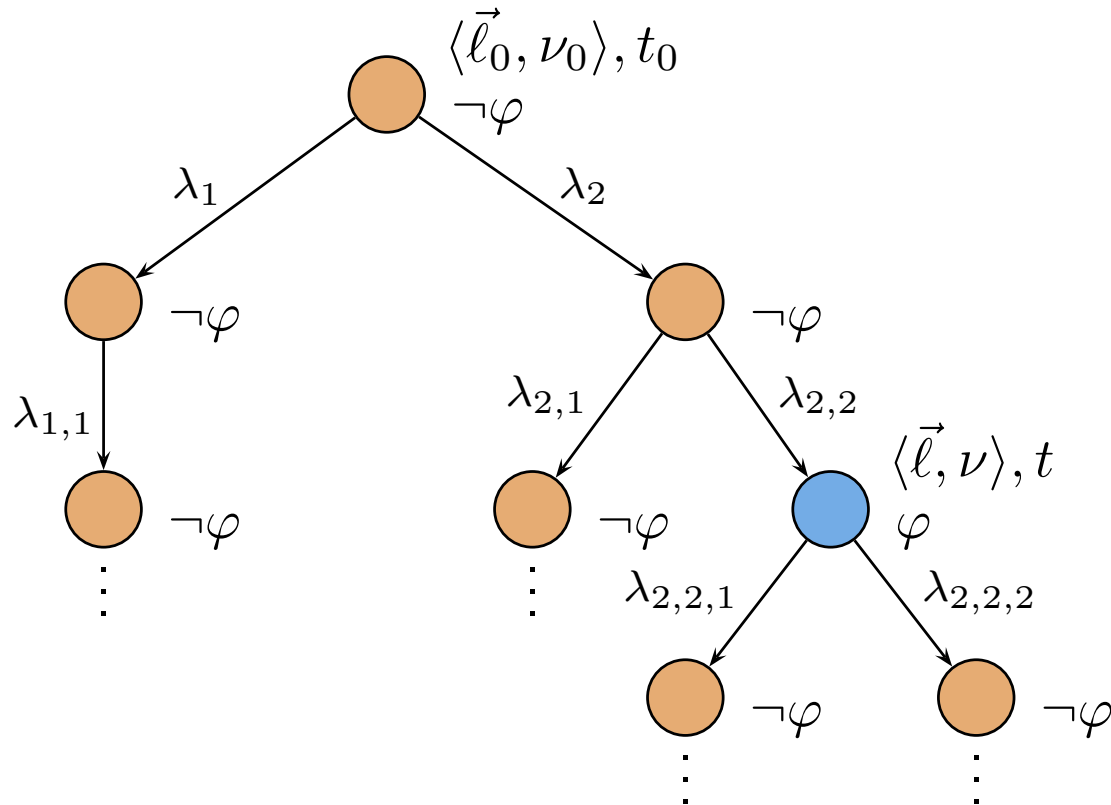
- It is defined inductively as follows:
- $\langle \vec{l}_0, \nu_0 \rangle, t_0 \models \mathcal{A}_i.l$ iff $l_{0,i} = l$ *i -th location in \vec{l}_0*
- $\langle \vec{l}_0, \nu_0 \rangle, t_0 \models \varphi$ iff $\nu_0 \models \varphi$
- $\langle \vec{l}_0, \nu_0 \rangle, t_0 \models \neg term$ iff $\langle \vec{l}_0, \nu_0 \rangle, t_0 \not\models term$
- $\langle \vec{l}_0, \nu_0 \rangle, t_0 \models term_1 \wedge term_2$ iff $\langle \vec{l}_0, \nu_0 \rangle, t_0 \models term_i, i = 1, 2$

Satisfaction of Uppaal-Logic by Configurations

Exists finally:

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

Example: $\exists \diamond \varphi$

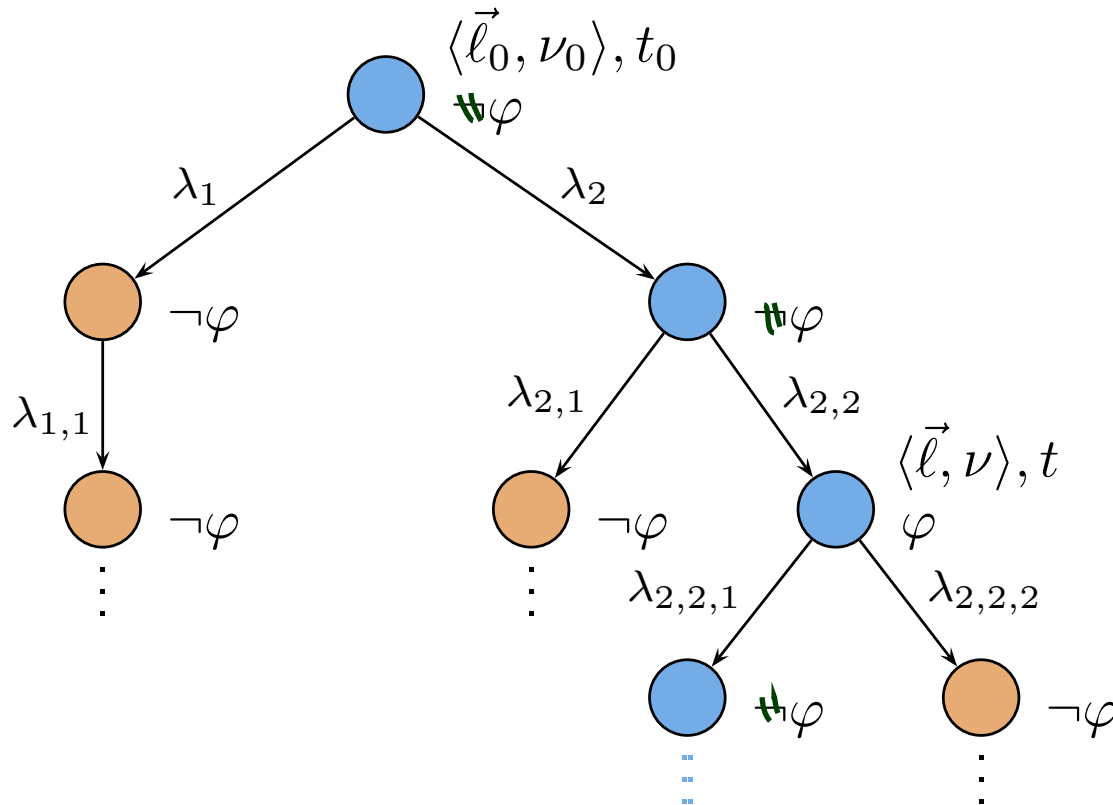


Satisfaction of Uppaal-Logic by Configurations

Exists globally:

- $\langle \vec{l}_0, \nu_0 \rangle, t_0 \models \exists \square \text{ term}$
iff
 $\exists \text{ path } \xi \text{ of } \mathcal{N} \text{ starting in } \langle \vec{l}_0, \nu_0 \rangle, t_0$
note: universally quantifying over all elements in $\xi(t)$
- $\forall t \in \text{Time}, \langle \vec{l}, \nu \rangle \in \text{Conf} :$
- $t_0 \leq t \wedge \langle \vec{l}, \nu \rangle \in \xi(t) \implies \langle \vec{l}, \nu \rangle, t \models \text{term}$

Example: $\exists \square \varphi$



Satisfaction of Uppaal-Logic by Configurations

- **Always finally:**

“ \forall path
 $\vdash \exists$ time \circ term”

- $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \models \forall \diamond term$ iff $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \not\models \exists \square \neg term$

- **Always globally:**

- $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \models \forall \square term$ iff $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \not\models \exists \diamond \neg term$

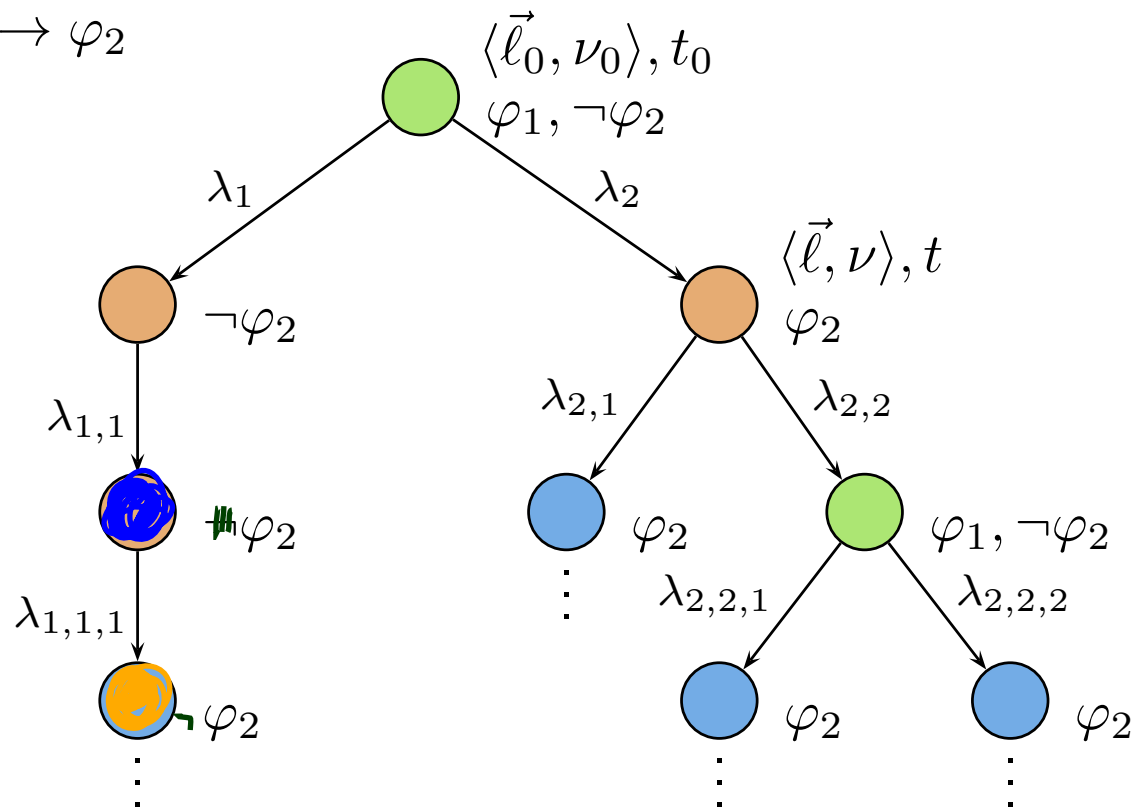
“ \forall path
 \forall time \circ term”

Satisfaction of Uppaal-Logic by Configurations

Leads to: (TL: "AG(term₁) ⇒ AF term₂")

- $\langle \vec{\ell}_0, \nu_0 \rangle, t_0 \models \text{term}_1 \longrightarrow \text{term}_2$ iff \forall path ξ of \mathcal{N} starting in $\langle \vec{\ell}_0, \nu_0 \rangle, t_0$
 $\forall t \in \text{Time}, \langle \vec{\ell}, \nu \rangle \in \text{Conf} :$
 $t_0 \leq t \wedge \langle \vec{\ell}, \nu \rangle \in \xi(t)$
 $\wedge \langle \vec{\ell}, \nu \rangle, t \models \text{term}_1$
 implies $\langle \vec{\ell}, \nu \rangle, t \models \forall \Diamond \text{term}_2$

Example: $\varphi_1 \longrightarrow \varphi_2$



Satisfaction of Uppaal-Logic by Networks

- We write

$$\mathcal{N} \models e\text{-formula}$$

if and only if

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

and

$$\mathcal{N} \models a\text{-formula}$$

if and only if

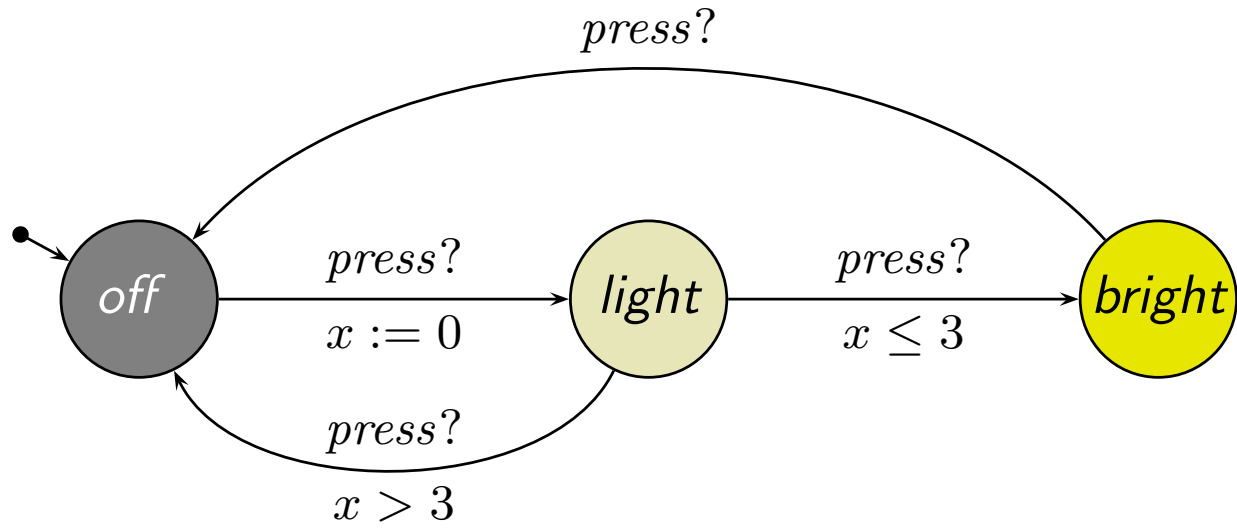
$$\text{for all } \langle \vec{\ell}_0, \nu_0 \rangle \in C_{ini}, \langle \vec{\ell}_0, \nu_0 \rangle, 0 \models a\text{-formula}, \quad (2)$$

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

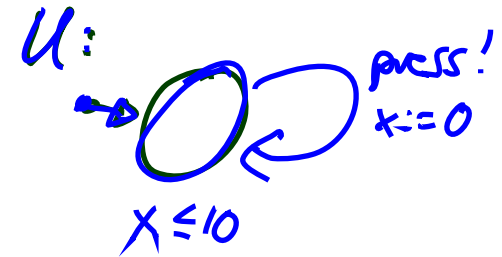
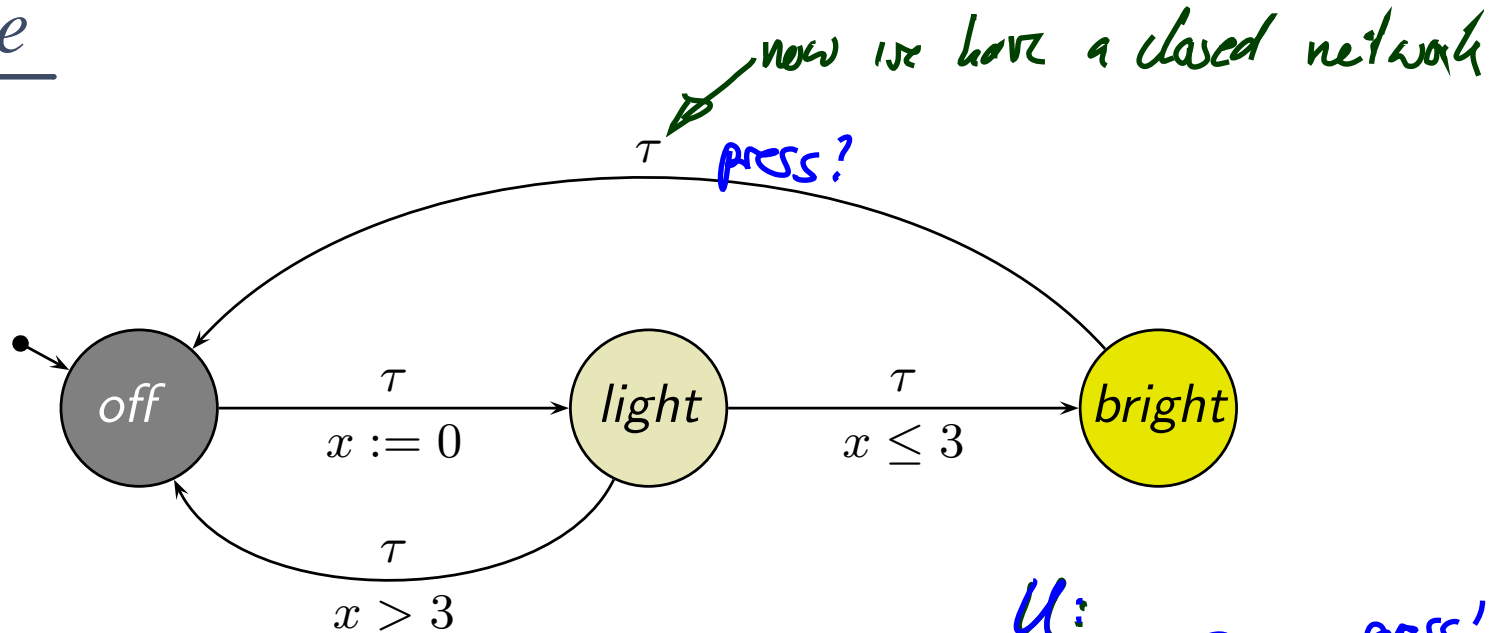
$$\mathcal{N} \models F \text{ if and only if } \langle \vec{\ell}_{ini}, \nu_{ini} \rangle, 0 \models F.$$

Example



Example

\mathcal{L} :



- $\mathcal{N} \models \exists \diamond \mathcal{L}. \text{bright?}$ ✓
- $\mathcal{N} \models \exists \square \mathcal{L}. \text{bright?}$ ✓
- $\mathcal{N} \models \exists \square \mathcal{L}. \text{off?}$ ✓
- $\mathcal{N} \models \forall \diamond \mathcal{L}. \text{light?}$ ✗
- $\mathcal{N} \models \forall \square (\mathcal{L}. \text{bright} \implies x \geq 3)?$ ✗
- $\mathcal{N} \models \mathcal{L}. \text{bright} \longrightarrow \mathcal{L}. \text{off?}$ ✗

References

References

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