Informal

A timed automaton is a finite automaton labeled with actions, extended with

- a global set of clocks
- clock invariants on states
- clocks guards for on edges
- clock assignments on the edges

A timed automaton has two types of transitions

- delay transitions:
  clocks may advance while invariant remain satisfied.
- action transitions:
  if the guard is satisfied we may follow an edge and execute the assigments.

Clock Constraints

If X is a set of clock variables then

$$\mathcal{C}(X) ::= x \prec c \mid c \prec x \mid \phi_1 \wedge \phi_2$$

where

$$x \in X \quad \prec \in \{<, \leq\} \quad c \in \mathbb{R}_0^+ = \{c \in \mathbb{R} \mid c \geq 0\} \quad \phi_1, \phi_2 \in \mathcal{C}(X)$$

Note that there there are many variations on the notion of timed automaton, which differ in their sets of atomic clock constraints. For example, Uppaal allows the difference of two clocks to be compared to a constant.

Timed Automaton

A timed automaton is a tuple $(\Sigma, S, S_0, X, I, T)$, where

- Σ is a finite alphabet;
- S is a set of locations;
- $S_0 \subseteq S$ is a set of inital states;
- X is a set of clock variables;
- $I : S \to \mathcal{C}(X)$ assigns location invariants;
- $T \subseteq S \times \Sigma \times \mathcal{C}(X) \times 2^X \times S$ is a set of transitions.
  A tuple $\langle s, a, \phi, C^0, t \rangle$ corresponds to an a transition from s to t, which is enabled if ϕ holds and which sets all clocks in C^0 to 0.

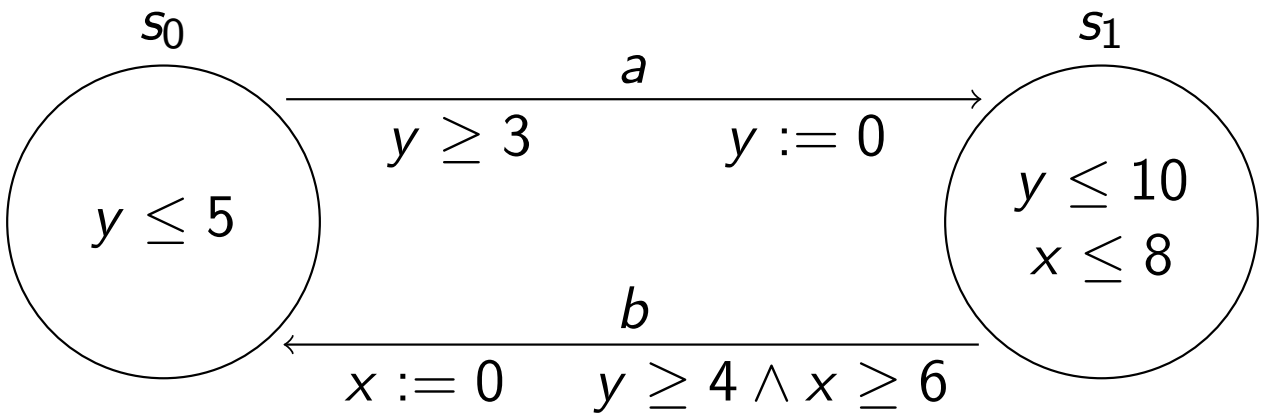
Semantics

The semantics of a timed automaton $(\Sigma, S, S_0, X, I, T)$ is an infinite LTS $(Q, \rightarrow, Q^0)$, where

- $Q = S \times \{v : X \rightarrow \mathbb{R}_0^+\}$
- $Q^0 = S_0 \times \{v : x \mapsto 0\}$
- $(s, v) \xrightarrow{d} (s, v + d)$ if $d > 0$ and $\forall 0 \leq e \leq d : \ v + e \models I(s)$, where $(v + d)(x) = v(x) + d$.
- $(s, v) \xrightarrow{a} (s', v')$ if $\langle s, a, \phi, C^0, s' \rangle \in T$, $v \models \phi(s)$ and

$$v'(x) = \begin{cases} 0 & \text{, if } x \in C^0 \\ v(x) & \text{, otherwise} \end{cases}$$

Example



Network of Timed Automata

A network of timed automata consists of

- A set of channels C.
- A set of Timed Automata $\mathcal{A}_i = (\Sigma, S_i, S_i^0, X, I_i, T_i)$ $(i = 1 \cdots N)$, where

$$\Sigma = \{\tau\} \cup \{c!, c? \mid c \in C\}$$

and X are shared.

  - $c!$ stands for send signal on c.
  - $c?$ stands for receive signal on c.
  - τ stands for invisible. (sometimes denoted i)

Semantics (1/2)

The semantics of this network is an infinite LTS $(Q, \rightarrow, Q^0)$, where

- $Q = S_1 \times \cdots \times S_N \times \{v : X \rightarrow \mathbb{R}_0^+\}$
- $Q^0 = S_1^0 \times \cdots \times S_N^0 \times \{v : x \mapsto 0\}$
- $(\vec{s}, v) \xrightarrow{d} (\vec{s}, v + d)$ if $d > 0$ and
  $\forall i : \ \forall 0 \leq e \leq d : \ v + e \models I_i(s_i)$
- A τ transition is possible if a single automaton can so such a step.
- A τ transition is possible if one automaton can send on channel c ($c!$) and another can receive ($c?$).

Semantics (2/2)

- $(s_1 \cdots s_i \cdots s_N, v) \xrightarrow{\tau} (s_1 \cdots s_i' \cdots s_N, v')$ if
  1. $\langle s_i, \tau, \phi, C^0, s_i' \rangle \in T_i,\ v \models \phi(s)$
  2. $v'(x) = \begin{cases} 0 & \text{, if } x \in C^0 \\ v(x) & \text{, otherwise} \end{cases}$
- $(s_1 \cdots s_i \cdots s_j \cdots s_N, v) \xrightarrow{\tau} (s_1 \cdots s_i' \cdots s_j' \cdots s_N, v')$ if
  1. $i \neq j$
  2. $\langle s_i, c!, \phi_i, C_i^0, s_i' \rangle \in T_i,\ v \models \phi_i(s)$
  3. $\langle s_j, c?, \phi_j, C_j^0, s_j' \rangle \in T_j,\ v \models \phi_j(s)$
  4. $v'(x) = \begin{cases} 0 & \text{, if } x \in C_i^0 \cup C_j^0 \\ v(x) & \text{, otherwise} \end{cases}$

CTL*

Formulas built from

atomic propositions e.g. p, q, r

boolean operators $\neg, \vee, \wedge$

path quantifiers

- A All paths
- E Exists some path

temporal operators

- X neXt state
- F Future state
- G Globally
- U Until
- R Release

CTL*

State formulas (ϕ_{SF})

$$\phi_{SF} ::= p \mid \neg\phi_{SF} \mid \phi_{SF} \vee \phi_{SF} \mid \phi_{SF} \wedge \phi_{SF} \mid \mathsf{A}\,\phi_{PF} \mid \mathsf{E}\,\phi_{PF}$$

Path formulas (ϕ_{PF})

$$\phi_{PF} ::= \phi_{SF} \mid \neg\phi_{PF} \mid \phi_{PF} \vee \phi_{PF} \mid \phi_{PF} \wedge \phi_{PF} \mid \mathsf{X}\,\phi_{PF} \mid \mathsf{F}\,\phi_{PF} \mid \mathsf{G}\,\phi_{PF} \mid \phi_{PF}\ \mathsf{U}\ \phi_{PF} \mid \phi_{PF}\ \mathsf{R}\ \phi_{PF}$$

Semantics of CTL*

A finite state program M over a set of atomic proposition Prop is a structure

$$(W, w_0, R, V)$$

where

- W is a finite set of states
- $w_0 \in W$ is the initial state
- $R \subseteq W \times W$ is an accessibility relation
- $V : W \longrightarrow 2^{\text{Prop}}$ assigns truth values to atomic propositions
- A path π is a possibly infinite list of states $(s_i)_{i=0}^{?}$, such that $s_i \; R \; s_{i+1}$.
- π^k denotes removing the first k elements.

Semantics of CTL*

Given a program $M \equiv (W, w_0, R, V)$ and a state $s \in W$:

$$
\begin{aligned}
M, s &\models p && \text{, if } p \in V(s) \\
M, s &\models \neg\phi && \text{, if } M, s \not\models \phi \\
M, s &\models \phi_1 \vee \phi_2 && \text{, if } M, s \models \phi_1 \text{ or } M, s \models \phi_2 \\
M, s &\models \phi_1 \wedge \phi_2 && \text{, if } M, s \models \phi_1 \text{ and } M, s \models \phi_2 \\
M, s &\models \mathsf{E}\,\phi && \text{, if } \exists \text{path } \pi \text{ starting in } s \text{ such that } M, \pi \models \phi \\
M, s &\models \mathsf{A}\,\phi && \text{, if } \forall \text{path } \pi \text{ starting in } s \; M, \pi \models \phi
\end{aligned}
$$

Semantics of CTL*

$$
\begin{aligned}
&M,\pi \models \phi && \text{, if } \phi \text{ is a state formula and } M,\pi(0) \models \phi \\
&M,\pi \models \neg\phi && \text{, if } M,\pi \not\models \phi \\
&M,\pi \models \phi_1 \vee \phi_2 && \text{, if } M,\pi \models \phi_1 \text{ or } M,\pi \models \phi_2 \\
&M,\pi \models \phi_1 \wedge \phi_2 && \text{, if } M,\pi \models \phi_1 \text{ and } M,\pi \models \phi_2 \\
&M,\pi \models \mathsf{X}\,\phi && \text{, if } M,\pi^1 \models \phi \\
&M,\pi \models \mathsf{F}\,\phi && \text{, if } \exists k \geq 0 : M,\pi^k \models \phi \\
&M,\pi \models \mathsf{G}\,\phi && \text{, if } \forall k \geq 0 : M,\pi^k \models \phi \\
&M,\pi \models \phi_1\ \mathsf{U}\ \phi_2 && \text{, if } \exists k \geq 0 : \begin{cases} M,\pi^k \models \phi_2 \text{ and} \\ \forall 0 \leq i < k : M,\pi^i \models \phi_1 \end{cases} \\
&M,\pi \models \phi_1\ \mathsf{R}\ \phi_2 && \text{, if } \forall k \geq 0 : \begin{cases} \forall 0 \leq i < k : M,\pi^i \not\models \phi_1 \\ \Rightarrow M,\pi^k \models \phi_2 \end{cases}
\end{aligned}
$$

Equivalences

$$
\begin{aligned}
\neg \mathsf{A}\,\phi &= \mathsf{E}\,\neg\phi \\
\mathsf{F}\,\phi &= \mathit{True}\,\mathsf{U}\,\phi \\
\neg \mathsf{F}\,\phi &= \mathsf{G}\,\neg\phi \\
\neg \mathsf{X}\,\phi &= \mathsf{X}\,\neg\phi \\
\neg(\phi_1\ \mathsf{U}\ \phi_2) &= \neg\phi_1\ \mathsf{R}\ \neg\phi_2
\end{aligned}
$$

Subsets of CTL*
Linear Temporal Logic (LTL)

All CTL* formulas of the form Aϕ, where ϕ is path-quantifier free:

$$\begin{aligned} \phi \quad ::= \quad & p \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \\ & X\,\phi \mid F\,\phi \mid G\,\phi \mid \phi\ U\ \phi \mid \phi\ R\ \phi \end{aligned}$$

The leading A is omitted.

Subsets of CTL*
Computation Tree Logic (CTL)

All CTL* formulas in which a temporal operator is immediately preceded by a path quantifier:

$$\begin{aligned}\phi \quad ::= \quad & p \mid \neg\phi \mid \phi \vee \phi \mid \phi \wedge \phi \mid \\ & \mathsf{AX}\,\phi \mid \mathsf{EX}\,\phi \mid \mathsf{AF}\,\phi \mid \mathsf{EF}\,\phi \mid \mathsf{AG}\,\phi \mid \mathsf{EG}\,\phi \mid \\ & \mathsf{A}[\phi\ \mathsf{U}\ \phi] \mid \mathsf{E}[\phi\ \mathsf{U}\ \phi] \mid \mathsf{A}[\phi\ \mathsf{R}\ \phi] \mid \mathsf{E}[\phi\ \mathsf{R}\ \phi]\end{aligned}$$

Alternative Notation

Uppaal	LaTeX	CTL
`A[]` ϕ	`A`$\Box\,\phi$	AGϕ
`A<>` ϕ	`A`$\Diamond\,\phi$	AFϕ
`E[]` ϕ	`E`$\Box\,\phi$	EGϕ
`E<>` ϕ	`E`$\Diamond\,\phi$	EFϕ
ψ `-->` ϕ	$\psi \leadsto \phi$	AG$(\psi \rightarrow$ AF$\phi)$