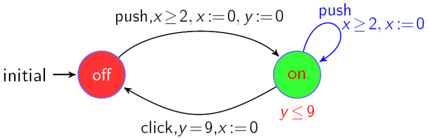
时间自动机

Timed automata

Timed Automata (TA) [Alur-Dill 1990]

1) A widely used formal model for verification of real-time systems.

2) Timed Automata: Finite automata /ω-automata + Clock variables.



A simple light controller

There are 3 modes: off, soft and bright.

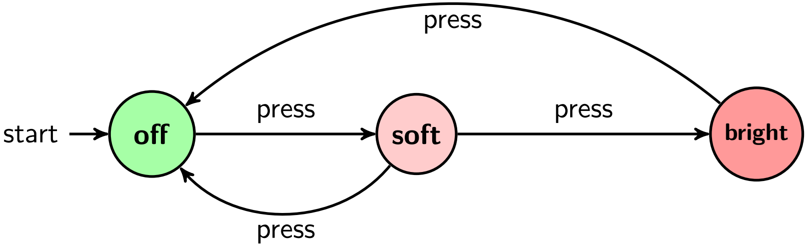
If press is issued twice quickly (say, in no more than 3 time units) then the light will get brighter. Otherwise the light is switched off.

时间自动机：小例子

A simple light controller

There are 3 modes: off, soft and bright.

If press is issued twice quickly (say, in no more than 3 time units) then the light will get brighter. Otherwise the light is switched off.



时间自动机：小例子

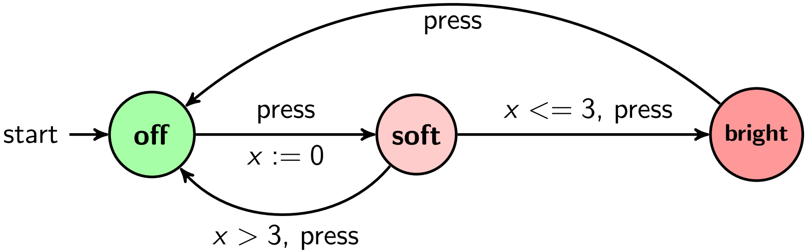
Introducing clock variables and clock constraints

A clock variable is a real-valued variable.

时间自动机：小例子

Introducing clock variables and clock constraints

A clock variable is a real-valued variable.



Delay at most 100 time units in bright mode

Introducing invariants

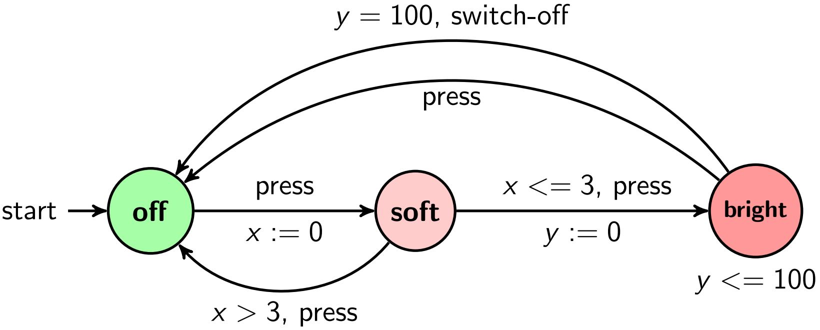
: Light Controller

Timed automata

时间自动机：小例子

Delay at most 100 time units in bright mode

Introducing invariants



: Light Controller

Locations(结点)： L

Clocks： X

Clock constraints $\Phi(X)$ is defined by the grammar

$$\phi ::= true \,|\, x \sim m \,|\, \phi_1 \wedge \phi_2$$

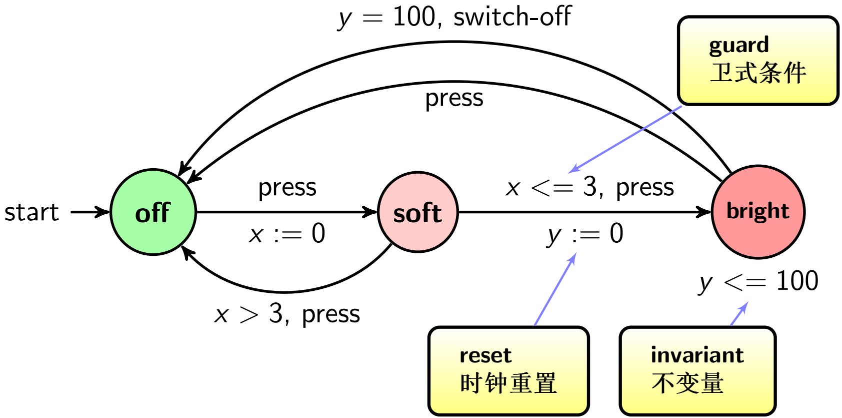
where $x \in X$, $\sim \in \{<, \leq, =, >, \geq\}$ and $m \in \mathbb{N}$. Invariants(不变量)

Labels/Actions： Σ

Edges(边)： E

$$\ell_1 \xrightarrow{g,\, a,\, \lambda} \ell_2$$

其中$g \in \Phi(X)$ 是卫式条件(guard), $a \in \Sigma$是标号, $\lambda \subseteq X$是时钟重置(clock reset)



: Light Controller

Timed automata

A timed automaton is a tuple $\mathcal{M} = < L, \ell, \Sigma, X, Inv, E >$, where

L: locations, $\ell_0 \in L$: the initial location;

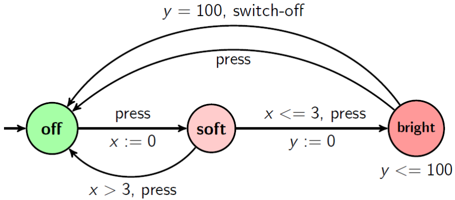
Σ: labels;

X: clocks;

$\Phi(X)$: clock constraints;

$Inv : L \mapsto \Phi(X)$ associates to each location an invariant;

$E \subseteq L \times \Phi(X) \times \Sigma \times 2^X \times L$ is a finite set of edges: $e = (\ell, g, a, \lambda, \ell')$ represents a transition from ℓ to ℓ'.



Time domain $\mathcal{T}$: 非负实数集$\mathbb{R}^{\geq 0}$, 非负有理数集，非负整数集, ...

Clock valuation μ: $X \mapsto \mathcal{T}$,
时钟X上所有时钟赋值的集合记为：$\mathsf{Val}(X, \mathcal{T})$

States: $(\ell, \mu) \in L \times \mathsf{Val}(X, \mathcal{T})$

Clock increment $\mu + \delta$:
$(\mu + \delta)(x) = \mu(x) + \delta$ for all $x \in X$.

Clock reset $\mu[\lambda := 0]$:
$\mu[\lambda := 0](x) = 0$ if $x \in \lambda$, else $\mu(x)$.

记号$\mu \models \phi$: 如果时钟赋值μ满足时钟约束ϕ

The semantics of a timed automaton $\mathcal{M}$ is defined to be an labelled transition system $(S_M, s_0, \Sigma \cup \mathcal{T}, \rightarrow)$, where

- $S_M = L \times \mathrm{Val}(X, \mathcal{T})$ is the set of all states;
- $s_0 = (\ell_0, \mu_0)$ with $\mu_0(x) = 0$ for all $x \in X$, is the initial state;
- $\rightarrow \subseteq S_M \times (\Sigma \cup \mathcal{T}) \times S_M$ is a set of transitions, and $\rightarrow$ consists of two kinds of transitions:
  - delay transition: $(\ell, \mu) \xrightarrow{\delta} (\ell, \mu + \delta)$, if $\delta \in \mathcal{T}$, $\mu \models \mathit{Inv}(\ell)$, and $\mu + \delta \models \mathit{Inv}(\ell)$;
  - discrete transition: $(\ell, \mu) \xrightarrow{a} (\ell', \mu[\lambda := 0])$, if there is an edge $(\ell, g, a, \lambda, \ell') \in E$ such that $\mu \models g$, and $\mu[\lambda := 0] \models \mathit{Inv}(\ell')$.

Semantics
例

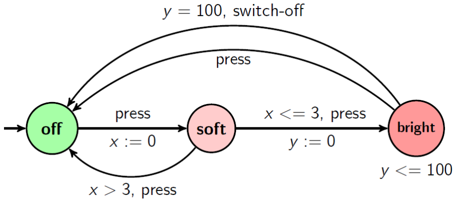
$$X = \{x,\ y\}$$

$$(\mathit{off},\ (x = 0, y = 0)) \xrightarrow{3.2} (\mathit{off},\ (x = 3.2, y = 3.2))$$

$$(\mathit{off},\ (x = 3.2, y = 3.2)) \xrightarrow{\text{press}} (\mathit{soft},\ (x = 0, y = 3.2))$$

$$(\mathit{soft},\ (x = 0, y = 3.2)) \xrightarrow{2.1} (\mathit{soft},\ (x = 2.1, y = 5.2))$$

$$(\mathit{soft},\ (x = 2.1, y = 5.2)) \xrightarrow{\text{press}} (\mathit{bright},\ (x = 2.1, y = 0))$$

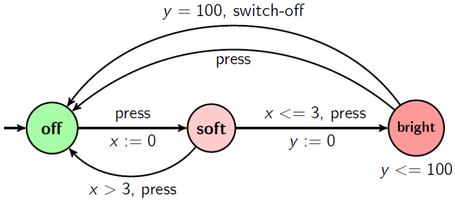


Semantics: Runs

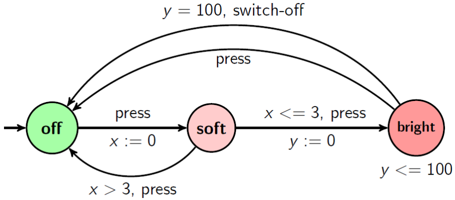
$$\rho = (\ell_0, \mu_0) \xrightarrow{\delta_0, a_0} (\ell_1, \mu_1) \xrightarrow{\delta_1, a_1} (\ell_2, \mu_2) \xrightarrow{\delta_2, a_2} \ldots$$

is called a run (or an execution) of $\mathcal{M}$ iff

- (ℓ_0, μ_0) is the initial state;
- for all $i \in \mathbb{N}$, $(\ell_i,\ \mu_i) \xrightarrow{\delta_i} (\ell_i,\ \mu_i + \delta_i) \xrightarrow{a_i} (\ell_{i+1},\ \mu_{i+1})$.



例子：Run



$$(\mathit{off}, (x=0, y=0)) \xrightarrow{3.2,\,\text{press}} (\mathit{soft}, (x=0, y=3.2)) \xrightarrow{2.5,\,\text{press}}$$
$$(\mathit{bright}, (x=2.5, y=0)) \xrightarrow{51.2,\,\text{press}} (\mathit{off}, (x=53.7,\ y=51.2)) \xrightarrow{31.3,\,\text{press}}$$
$$(\mathit{soft}, (x=0,\ y=82.5)) \xrightarrow{12.3,\,\text{press}} (\mathit{off}, (x=12.3,\ y=94.8)) \xrightarrow{10.5,\,\text{press}}$$
$$(\mathit{soft}, (x=0,\ y=105.3)) \xrightarrow{1.5,\,\text{press}} (\mathit{bright}, (x=1.5,\ y=0))$$
$$\xrightarrow{100,\,\text{switch-off}} (\mathit{off}, (x=101.5,\ y=100))\ldots$$

Timed languages

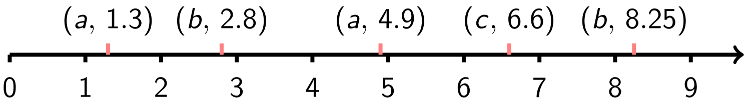
Timed words

Σ: finite set of labels;

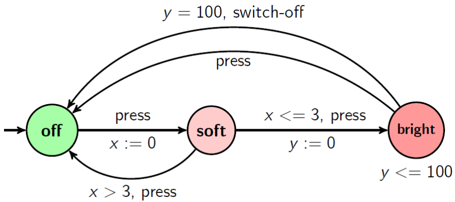
A timed word over Σ is a sequence of the form $w = (a_0, t_0)(a_1, t_1)(a_2, t_2)\dots$ with $t_i \in \mathcal{T}$ for all $i \in \mathbb{N}$, $t_i \leq t_{i+1}$ and $a_i \in \Sigma$ for all $i \in \mathbb{N}$;

A timed word $w = (a_0, t_0)(a_1, t_1)(a_2, t_2)\dots$ is called accepting by a timed automaton $\mathcal{M}$ (over Σ), if there exists a sequence $s_0 s_1 s_2 \dots$ of states such that s_0 is the initial state of $\mathcal{M}$ and $s_0 \xrightarrow{t_0,\, a_0} s_1 \xrightarrow{t_1 - t_0,\, a_1} s_2 \xrightarrow{t_2 - t_1,\, a_2} s_3 \xrightarrow{t_3 - t_2,\, a_3} s_4 \dots$ is a run of $\mathcal{M}$.

The timed language of $\mathcal{M}$, denoted $\mathcal{L}(\mathcal{M})$, is defined to be the set of all accepting timed words of $\mathcal{M}$.

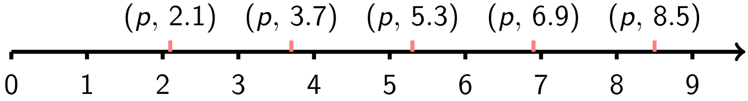


Timed languages
Timed words



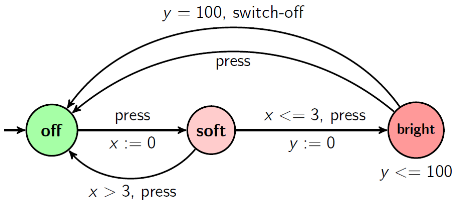
Assume $\Sigma = \{\text{press}, \text{switch-off}\}$

$(\text{press}, 2.1)(\text{press}, 3.7)(\text{press}, 5.3)(\text{press}, 6.9)(\text{press}, 8.5)\ldots$ is an acceptig timed word;



$(\text{press}, 2.1)(\text{press}, 3.7)(\text{switch-off}, 5.3)(\text{press}, 6.9)(\text{press}, 8.5)\ldots$ is not an acceptig timed word;

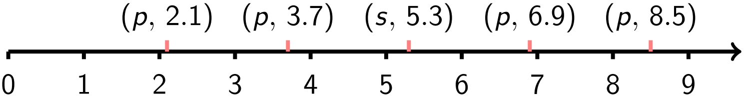
$(\text{press}, 2.1)(\text{press}, 3.7)(\text{switch-off}, 103.7)(\text{press}, 108)(\text{press}, 120)\ldots$ is



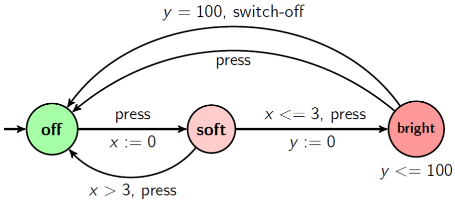
Assume $\Sigma = \{\text{press}, \text{switch-off}\}$

$(\text{press}, 2.1)(\text{press}, 3.7)(\text{press}, 5.3)(\text{press}, 6.9)(\text{press}, 8.5)\ldots$ is an acceptig timed word;

$(\text{press}, 2.1)(\text{press}, 3.7)(\text{switch-off}, 5.3)(\text{press}, 6.9)(\text{press}, 8.5)\ldots$ is not an acceptig timed word;



$(\text{press}, 2.1)(\text{press}, 3.7)(\text{switch-off}, 103.7)(\text{press}, 108)(\text{press}, 120)\ldots$ is



Assume $\Sigma = \{\text{press}, \text{switch-off}\}$

$(\text{press}, 2.1)(\text{press}, 3.7)(\text{press}, 5.3)(\text{press}, 6.9)(\text{press}, 8.5)\ldots$ is an acceptig timed word;

$(\text{press}, 2.1)(\text{press}, 3.7)(\text{switch-off}, 5.3)(\text{press}, 6.9)(\text{press}, 8.5)\ldots$ is not an acceptig timed word;

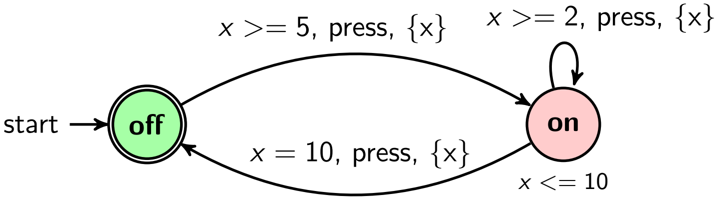
$(\text{press}, 2.1)(\text{press}, 3.7)(\text{switch-off}, 103.7)(\text{press}, 108)(\text{press}, 120)\ldots$ is an acceptig timed word;

Timed Büchi Automata (TBA)

Timed Automata + Büchi accepting conditions

Some locations are assigned as accepting

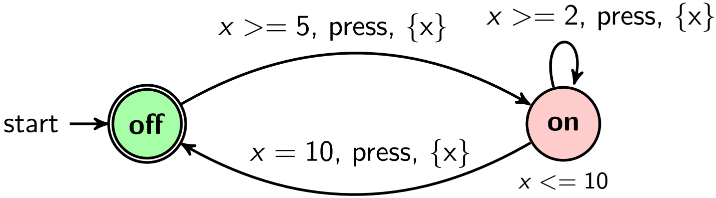
Used to express liveness.



不再停留在结点on中不出来，不再接受时间字$(press, 6.1)(press, 9.1)(press, 12.1)(press, 15.1)\dots$

Timed Büchi Automaton: $\mathfrak{M} = < L, l_0, \Sigma, X, Inv, E, F >$

- L: locations, $\ell_0 \in L$: the initial location;
- Σ: labels;
- X: clocks;
- $\Phi(X)$: clock constraints;
- $Inv : L \mapsto \Phi(X)$ associates to each location an invariant;
- $E \subseteq L \times \Phi(X) \times \Sigma \times 2^X \times L$ is a finite set of edges: $e = (\ell, g, a, \lambda, \ell')$ represents a transition from ℓ to ℓ';
- $F \subseteq L$: accepting locations.



Timed languages

类似于时间自动机，我们可给出时间Büchi自动机接受的无穷时间字的定义和接受的时间语言的定义

TBA和TA所接受的时间语言关于交、并、补运算的封闭性

- 关于交、并运算是封闭的
- 对于补运算（一般）是不封闭的，但确定性TBA(TA)关于补运算是封闭的
- 对补运算不封闭的例子