

- labelled transition systems with time
- timed automata
- timed and untimed bisimilarity
- timed and untimed language equivalence
- networks of timed automata

- **Timeout in Alternating Bit protocol:**
  - In CCS timeouts were modelled using nondeterminism.
  - Enough to prove that the protocol is safe.
  - But too abstract for certain questions (like “What is the average time to deliver the message?”).
- **Many real-life systems depend on timing:**
  - Real-time controllers (production lines, computers in cars, railway crossings).
  - Embedded systems (mobile phones, remote controllers, digital watch).
  - ...

## Timed (labelled) transition system (TLTS)

TLTS is a triple  $(Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$  where

- $Proc$  is a set of states (or processes),
- $Act = N \cup \mathbb{R}^{\geq 0}$  is a set of **actions** (consisting of **labels** and **time-elapsing steps**), and
- for every  $a \in Act$ ,  $\xrightarrow{a} \subseteq Proc \times Proc$  is a binary relation on states called the transition relation.

We write

- $s \xrightarrow{a} s'$  if  $a \in N$  and  $(s, s') \in \xrightarrow{a}$ , and
- $s \xrightarrow{d} s'$  if  $d \in \mathbb{R}^{\geq 0}$  and  $(s, s') \in \xrightarrow{d}$ .

- **Time additivity:**

if  $s \xrightarrow{d} s'$  and  $0 \leq d' \leq d$  then  $s \xrightarrow{d'} s'' \xrightarrow{d-d'} s'$  for some state  $s''$ ;

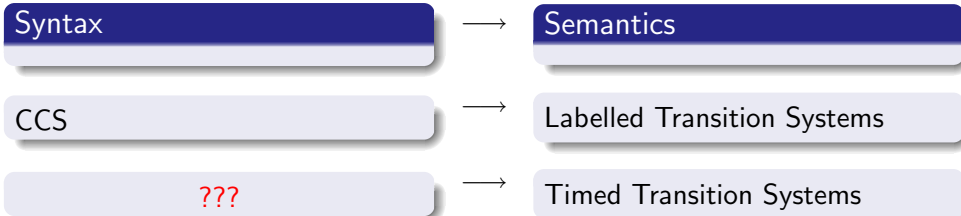
- **Zero delay:**

$s \xrightarrow{0} s$  for all states  $s$ ;

- **Time determinism:**

if  $s \xrightarrow{d} s'$  and  $s \xrightarrow{d} s''$  then  $s' = s''$ .

# How to Describe Timed Transition Systems?

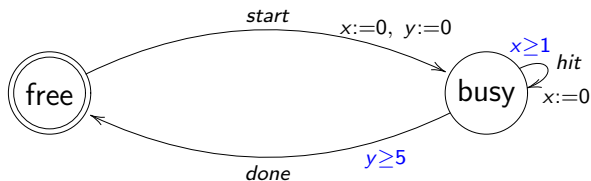


## A possible answer

**Timed Automata** [Alur, Dill'90]

i.e., finite-state automata equipped with real-valued clocks.

# Example: Hammer



Some important questions:

- How to model a real-time system?
- What **guards** should we allow?
- ....

# Definition of TA: Clock Constraints

Let  $C = \{x, y, \dots\}$  be a finite set of clocks.

$\mathcal{B}(C)$  ... the set of clock constraints over  $C$

$\mathcal{B}(C)$  is defined by the following abstract syntax

$$g ::= x \sim n \mid g_1 \wedge g_2$$

where  $x, y \in C$  are clocks,  $n \in \mathbb{N}$  and  $\sim \in \{\leq, <, =, >, \geq\}$ .

Example:  $x \leq 3 \wedge y > 0$

## Clock valuation

Clock valuation  $v$  is a function  $v : C \rightarrow \mathbb{R}^{\geq 0}$ .

Let  $v$  be a clock valuation. Then

- $v + d$  is a clock valuation for any  $d \in \mathbb{R}^{\geq 0}$  and it is defined by

$$(v + d)(x) = v(x) + d \text{ for all } x \in C$$

- $v[r]$  is a clock valuation for any  $r \subseteq C$  and it is defined by

$$v[r](x) \begin{cases} 0 & \text{if } x \in r \\ v(x) & \text{otherwise.} \end{cases}$$



## Evaluation of clock constraints ( $v \models g$ )

$$v \models x < n \quad \text{iff } v(x) < n$$

$$v \models x \leq n \quad \text{iff } v(x) \leq n$$

$$v \models x = n \quad \text{iff } v(x) = n$$

⋮

$$v \models g_1 \wedge g_2 \quad \text{iff } v \models g_1 \text{ and } v \models g_2$$

## Definition

A **timed automaton** over a set of clocks  $C$  and a set of labels  $N$  is a tuple

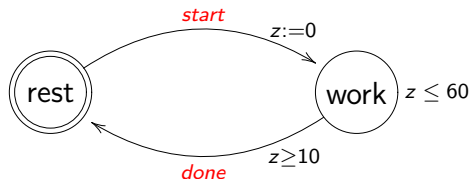
$$(L, \ell_0, E, I)$$

where

- $L$  is a finite set of **locations**
- $\ell_0 \in L$  is the **initial location**
- $E \subseteq L \times \mathcal{B}(C) \times N \times 2^C \times L$  is the set of **edges**
- $I : L \rightarrow \mathcal{B}(C)$  assigns **invariants** to locations.

We usually write  $\ell \xrightarrow{g, a, r} \ell'$  whenever  $(\ell, g, a, r, \ell') \in E$ .

# Example: A Worker



What does this mean?

# Semantics of Timed Automata

Let  $A = (L, \ell_0, E, I)$  be a timed automaton.

## Timed transition system generated by $A$

$T(A) = (Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$  where

- $Proc = L \times (C \rightarrow \mathbb{R}^{\geq 0})$ , i.e. states are of the form  $(\ell, \nu)$  where  $\ell$  is a location and  $\nu$  a valuation such that  $\nu \models I(\ell)$
- $Act = N \cup \mathbb{R}^{\geq 0}$
- $\longrightarrow$  is defined as follows:

$(\ell, \nu) \xrightarrow{a} (\ell', \nu')$  if there is  $(\ell \xrightarrow{g, a, r} \ell') \in E$  s.t.  $\nu \models g$ ,  $\nu' = \nu[r]$ ,  $\nu' \models I(\ell')$

$(\ell, \nu) \xrightarrow{d} (\ell, \nu + d)$  if  $\nu + d' \models I(\ell)$  for each  $d' \in [0, d]$ .

Let  $A_1$  and  $A_2$  be timed automata.

## Timed Bisimilarity

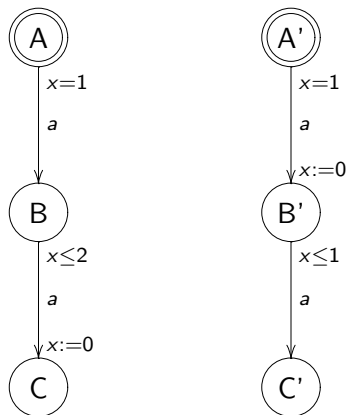
We say that  $A_1$  and  $A_2$  are **timed bisimilar** iff the transition systems  $T(A_1)$  and  $T(A_2)$  generated by  $A_1$  and  $A_2$  are strongly bisimilar.

Remark: both

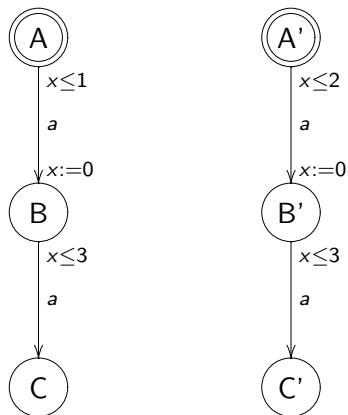
- $\xrightarrow{a}$  for  $a \in N$  and
- $\xrightarrow{d}$  for  $d \in \mathbb{R}^{\geq 0}$

are considered as normal (**visible**) transitions.

# Example of Timed Bisimilar Automata



# Example of Timed Non-Bisimilar Automata



# Untimed Bisimilarity

Let  $A_1$  and  $A_2$  be timed automata. Let  $\epsilon$  be a new (fresh) action.

## Untimed Bisimilarity

We say that  $A_1$  and  $A_2$  are **untimed bisimilar** iff the transition systems  $T(A_1)$  and  $T(A_2)$  generated by  $A_1$  and  $A_2$  where **every transition of the form  $\xrightarrow{d}$  for  $d \in \mathbb{R}^{\geq 0}$  is replaced with  $\xrightarrow{\epsilon}$**  are strongly bisimilar.

Remark:

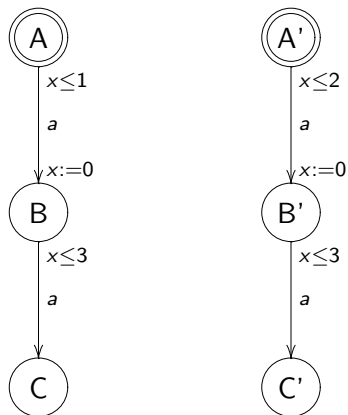
- $\xrightarrow{a}$  for  $a \in N$  is treated as a visible transition, while
- $\xrightarrow{d}$  for  $d \in \mathbb{R}^{\geq 0}$  are all labelled by a single visible action  $\xrightarrow{\epsilon}$ .

## Corollary

Any two timed bisimilar automata are also untimed bisimilar.



# Timed Non-Bisimilar but Untimed Bisimilar Automata



## Theorem [Cerans'92]

Timed bisimilarity for timed automata is in EXPTIME (decidable in deterministic exponential time).

## Theorem [Larsen, Wang'93]

Untimed bisimilarity for timed automata is in EXPTIME.

# Timed Traces

Let  $A = (L, \ell_0, E, I)$  be a timed automaton over a set of clocks  $C$  and a set of labels  $N$ .

## Timed Traces

A sequence  $(t_1, a_1)(t_2, a_2)(t_3, a_3) \dots$  where  $t_i \in \mathbb{R}^{\geq 0}$  and  $a_i \in N$  is called a **timed trace of  $A$**  iff there is a transition sequence

$$(\ell_0, v_0) \xrightarrow{d_1} \cdot \xrightarrow{a_1} \cdot \xrightarrow{d_2} \cdot \xrightarrow{a_2} \cdot \xrightarrow{d_3} \cdot \xrightarrow{a_3} \dots$$

in  $A$  such that  $v_0(x) = 0$  for all  $x \in C$  and

$$t_i = t_{i-1} + d_i \quad \text{where } t_0 = 0.$$

Intuition:  $t_i$  is the absolute time (**time-stamp**) when  $a_i$  happened since the start of the automaton  $A$ .

# Timed and Untimed Language Equivalence

The set of all timed traces of an automaton  $A$  is denoted by  $L(A)$  and called the **timed language of  $A$** .

**Theorem [Alur, Courcoubetis, Dill, Henzinger'94]**

Timed language equivalence (the problem whether  $L(A_1) = L(A_2)$  for given timed automata  $A_1$  and  $A_2$ ) is undecidable.

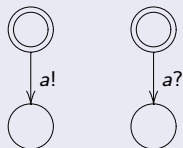
We say that  $a_1 a_2 a_3 \dots$  is an **untimed trace of  $A$**  iff there exist  $t_1, t_2, t_3, \dots \in \mathbb{R}^{\geq 0}$  such that  $(t_1, a_1)(t_2, a_2)(t_3, a_3) \dots$  is a timed trace of  $A$ .

**Theorem [Alur, Dill'94]**

Untimed language equivalence for timed automata is decidable.

# Networks of Timed Automata

## Timed Automata in Parallel



## Intuition in CCS

$$(\bar{a}.Nil \mid a.Nil) \setminus \{a\}$$

Let  $C$  be a set of clocks and  $Chan$  a set of channels.

We let  $Act = N \cup \mathbb{R}^{\geq 0}$  where

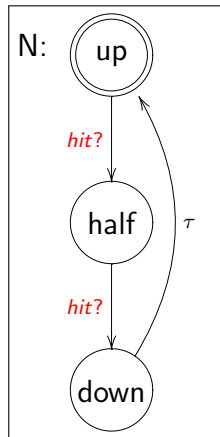
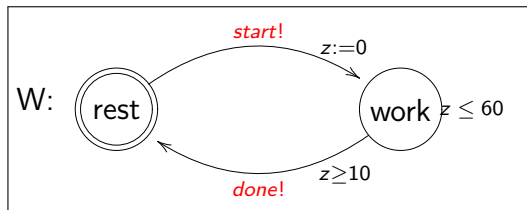
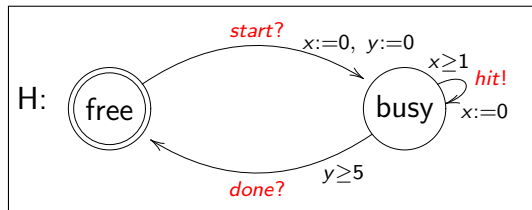
- $N = \{c! \mid c \in Chan\} \cup \{c? \mid c \in Chan\} \cup \{\tau\}$ .

Let  $A_i = (L_i, \ell_0^i, E_i, I_i)$  be timed automata for  $1 \leq i \leq n$ .

## Networks of Timed Automata

We call  $A = A_1 \mid A_2 \mid \dots \mid A_n$  a **network of timed automata**.

# Example: Hammer, Worker, Nail



# Timed Transition System Generated by $A = A_1 | \dots | A_n$

$T(A) = (Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$  where

- $Proc = (L_1 \times L_2 \times \dots \times L_n) \times (C \rightarrow \mathbb{R}^{\geq 0})$ , i.e. states are of the form  $((l_1, l_2, \dots, l_n), v)$  where  $l_i$  is a location in  $A_i$ ,  $v \models I_i(l_i)$
- $Act = \{\tau\} \cup \mathbb{R}^{\geq 0}$
- $\longrightarrow$  is defined as follows:

$((l_1, \dots, l_i, \dots, l_n), v) \xrightarrow{\tau} ((l_1, \dots, l'_i, \dots, l_n), v')$  if there is  $(l_i \xrightarrow{g, \tau, r} l'_i) \in E_i$  s.t.  $v \models g$  and  $v' = v[r]$  and  $v' \models I_i(l'_i) \wedge \bigwedge_{k \neq i} I_k(l_k)$

$((l_1, \dots, l_n), v) \xrightarrow{d} ((l_1, \dots, l_n), v + d)$  for all  $d \in \mathbb{R}^{\geq 0}$  s.t.  $v \models \bigwedge_k I_k(l_k)$   
and  $v + d \models \bigwedge_k I_k(l_k)$

$((l_1, \dots, l_i, \dots, l_j, \dots, l_n), v) \xrightarrow{\tau} ((l_1, \dots, l'_i, \dots, l'_j, \dots, l_n), v')$  if  $i \neq j$   
and there are  $(l_i \xrightarrow{g_i, a!, r_i} l'_i) \in E_i$  and  $(l_j \xrightarrow{g_j, a?, r_j} l'_j) \in E_j$  s.t.  $v \models g_i \wedge g_j$  and  
 $v' = v[r_i \cup r_j]$  and  $v' \models I_i(l'_i) \wedge I_j(l'_j) \wedge \bigwedge_{k \neq i, j} I_k(l_k)$