

# Introduction to labelled transition systems

José Proença

HASLab - INESC TEC  
Universidade do Minho  
Braga, Portugal

February, 2016

# Reactive systems

## Reactive system

system that computes by reacting to stimuli from its environment along its overall computation

- in contrast to sequential systems whose meaning is defined by the results of finite computations, the behaviour of reactive systems is mainly determined by **interaction** and **mobility** of **non-terminating** processes, evolving **concurrently**.
- **observation**  $\equiv$  interaction
- **behaviour**  $\equiv$  a structured record of interactions

# Labelled Transition System

## Definition

A LTS over a set  $N$  of names is a tuple  $\langle S, N, \longrightarrow \rangle$  where

- $S = \{s_0, s_1, s_2, \dots\}$  is a set of states
- $\longrightarrow \subseteq S \times N \times S$  is the transition relation, often given as an  $N$ -indexed family of binary relations

$$s \xrightarrow{a} s' \equiv \langle s', a, s \rangle \in \longrightarrow$$

# Labelled Transition System

## Morphism

A **morphism** relating two LTS over  $N$ ,  $\langle S, N, \longrightarrow \rangle$  and  $\langle S', N, \longrightarrow' \rangle$ , is a function  $h : S \longrightarrow S'$  st

$$s \xrightarrow{a} s' \quad \Rightarrow \quad h s \xrightarrow{a}' h s'$$

morphisms **preserve** transitions

# Labelled Transition System

## System

Given a LTS  $\langle S, N, \longrightarrow \rangle$ , each state  $s \in S$  determines a **system** over all states reachable from  $s$  and the corresponding restrictions of  $\longrightarrow$  and  $\downarrow$ .

## LTS classification

- deterministic
- non deterministic
- finite
- finitely branching
- image finite
- ...

# Reachability

## Definition

The reachability relation,  $\longrightarrow^* \subseteq S \times N^* \times S$ , is defined inductively

- $s \xrightarrow{\epsilon}^* s$  for each  $s \in S$ , where  $\epsilon \in N^*$  denotes the empty word;
- if  $s \xrightarrow{a} s''$  and  $s'' \xrightarrow{\sigma}^* s'$  then  $s \xrightarrow{a\sigma}^* s'$ , for  $a \in N, \sigma \in N^*$

## Reachable state

$t \in S$  is **reachable** from  $s \in S$  iff there is a word  $\sigma \in N^*$  st  $s \xrightarrow{\sigma}^* t$

# Process algebras

## CCS - Syntax

$$\mathcal{P} \ni P, Q ::= K \mid \alpha.P \mid \sum_{i \in I} P_i \mid \{f\}P \mid P|Q \mid P \setminus L$$

where

- $\alpha \in N \cup \bar{N} \cup \{\tau\}$ ,
- $K$  is a collection of process names or process constants,
- $I$  is an indexing set, and
- $L \subseteq N \cup \bar{N}$
- $f$  is a function that renames actions
- notation:  $\mathbf{0} = \sum_{i \in \emptyset} P_i$ ;  $P_1 + P_2 = \sum_{i \in \{1,2\}} P_i$ ;  $f = [b_1/a_1, \dots, b_n/a_n]$

# Process algebras

$$\mathcal{P} \ni P, Q ::= K \mid \alpha.P \mid \sum_{i \in I} P_i \mid \{f\}P \mid P|Q \mid P \setminus L$$

Which are syntactically correct?

$$a.b.A + B \quad (1)$$

$$(a.\mathbf{0} + \bar{a}.A)\{\backslash\}a, b \quad (2)$$

$$(a.\mathbf{0} + \bar{a}.A)\{\backslash\}a, \tau \quad (3)$$

$$a.V + [a/b] \quad (4)$$

$$\tau.\tau.B + \mathbf{0} \quad (5)$$

$$(a.B + b.B)[a/b, b/a] \quad (6)$$

$$(a.B + \tau.B)[a/b, b/a] \quad (7)$$

$$(a.b.A + \bar{a}.\mathbf{0})|B \quad (8)$$

$$(a.b.A + \bar{a}.\mathbf{0}).B \quad (9)$$

$$(a.b.A + \bar{a}.\mathbf{0}) + B \quad (10)$$

$$(\mathbf{0}|\mathbf{0}) + \mathbf{0} \quad (11)$$



# CCS semantics - building an LTS

$$\begin{array}{c}
 \text{(act)} \\
 \hline
 \alpha.P \xrightarrow{\alpha} P
 \end{array}
 \qquad
 \begin{array}{c}
 \text{(sum-j)} \\
 \hline
 \frac{P_j \xrightarrow{\alpha} P'_j}{\sum_{i \in I} P_i \xrightarrow{\alpha} P'_j} \quad j \in I
 \end{array}$$
  

$$\begin{array}{c}
 \text{(com1)} \\
 \hline
 \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q}
 \end{array}
 \qquad
 \begin{array}{c}
 \text{(com2)} \\
 \hline
 \frac{Q \xrightarrow{\alpha} Q'}{P|Q \xrightarrow{\alpha} P|Q'}
 \end{array}
 \qquad
 \begin{array}{c}
 \text{(com3)} \\
 \hline
 \frac{P \xrightarrow{a} P' \quad Q \xrightarrow{\bar{a}} Q'}{P|Q \xrightarrow{\tau} P'|Q'}
 \end{array}$$
  

$$\begin{array}{c}
 \text{(res)} \\
 \hline
 \frac{P \xrightarrow{\alpha} P'}{P \setminus L \xrightarrow{\alpha} P' \setminus L} \quad \alpha, \bar{\alpha} \notin L
 \end{array}
 \qquad
 \begin{array}{c}
 \text{(rel)} \\
 \hline
 \frac{P \xrightarrow{\alpha} P'}{\{f\} P \xrightarrow{f(\alpha)} \{d\} P'}
 \end{array}$$

Draw the LTS's

$CM = \text{coin}.\overline{\text{coffee}}.CM$

$CS = \overline{\text{pub}}.\overline{\text{coin}}.\text{coffee}.CS$

$SmUni = (CM|CS) \setminus \{\text{coin}, \text{coffee}\}$

# CCS semantics - building an LTS

$$\begin{array{c}
 \text{(act)} \\
 \frac{}{\alpha.P \xrightarrow{\alpha} P} \\
 \\
 \text{(sum-j)} \\
 \frac{P_j \xrightarrow{\alpha} P'_j}{\sum_{i \in I} P_i \xrightarrow{\alpha} P'_j} \quad j \in I \\
 \\
 \text{(com1)} \qquad \text{(com2)} \qquad \text{(com3)} \\
 \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q} \qquad \frac{Q \xrightarrow{\alpha} Q'}{P|Q \xrightarrow{\alpha} P|Q'} \qquad \frac{P \xrightarrow{a} P' \quad Q \xrightarrow{\bar{a}} Q'}{P|Q \xrightarrow{\tau} P'|Q'} \\
 \\
 \text{(res)} \qquad \text{(rel)} \\
 \frac{P \xrightarrow{\alpha} P'}{P \setminus L \xrightarrow{\alpha} P' \setminus L} \quad \alpha, \bar{\alpha} \notin L \qquad \frac{P \xrightarrow{\alpha} P'}{\{f\} P \xrightarrow{f(\alpha)} \{d\} P'}
 \end{array}$$

Draw the LTS's

$$CM = \text{coin}.\overline{\text{coffee}}.CM$$

$$CS = \overline{\text{pub}}.\overline{\text{coin}}.\text{coffee}.CS$$

$$SmUni = (CM|CS) \setminus \{\text{coin}, \text{coffee}\}$$

# Behavioural Equivalences – Intuition

Two LTS should be **equivalent** if they cannot be distinguished by interacting with them.

## Equality of functional behaviour

is not preserved by **parallel** composition: non **compositional** semantics, cf,

$x:=4; x:=x+1$  and  $x:=5$

## Graph isomorphism

is too strong (why?)

# Trace

## Definition

Let  $T = \langle S, N, \longrightarrow \rangle$  be a labelled transition system. The set of **traces**  $\text{Tr}(s)$ , for  $s \in S$  is the minimal set satisfying

$$(1) \quad \epsilon \in \text{Tr}(s)$$

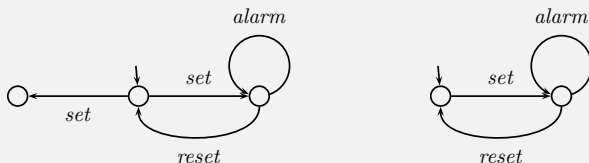
$$(2) \quad a\sigma \in \text{Tr}(s) \Rightarrow \langle \exists s' : s' \in S : s \xrightarrow{a} s' \wedge \sigma \in \text{Tr}(s') \rangle$$

# Trace equivalence

## Definition

Two states  $s, r$  are **trace equivalent** iff  $\text{Tr}(s) = \text{Tr}(r)$   
(i.e. if they can perform the same finite sequences of transitions)

## Example



**Trace equivalence** applies when one can neither interact with a system, nor distinguish a slow system from one that has come to a stand still.

# Simulation

the quest for a behavioural equality:  
able to identify states that cannot be distinguished by any realistic  
form of observation

## Simulation

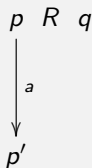
A state  $q$  **simulates** another state  $p$  **if**  
every transition from  $q$  is corresponded by a transition from  $p$  **and**  
this capacity is kept along the whole life of the system to which  
state space  $q$  belongs to.

# Simulation

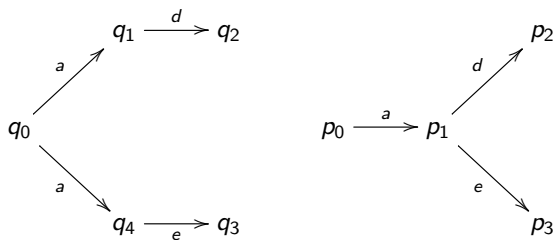
## Definition

Given  $\langle S_1, N, \longrightarrow_1 \rangle$  and  $\langle S_2, N, \longrightarrow_2 \rangle$  over  $N$ , relation  $R \subseteq S_1 \times S_2$  is a **simulation** iff, for all  $\langle p, q \rangle \in R$  and  $a \in N$ ,

$$(1) \quad p \xrightarrow{a}_1 p' \Rightarrow \langle \exists q' : q' \in S_2 : q \xrightarrow{a}_2 q' \wedge \langle p', q' \rangle \in R \rangle$$



# Example



$$q_0 \lesssim p_0 \quad \text{cf.} \quad \{ \langle q_0, p_0 \rangle, \langle q_1, p_1 \rangle, \langle q_4, p_1 \rangle, \langle q_2, p_2 \rangle, \langle q_3, p_3 \rangle \}$$



# Similarity

## Definition

$$p \lesssim q \equiv \langle \exists R :: R \text{ is a simulation and } \langle p, q \rangle \in R \rangle$$

## Lemma

The similarity relation is a preorder  
(ie, reflexive and transitive)

# Bisimulation

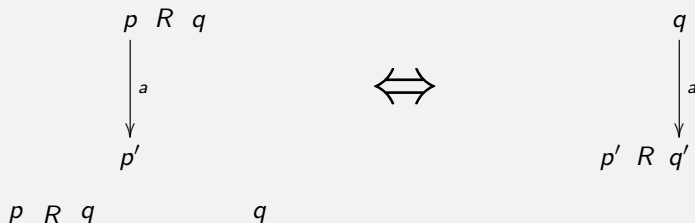
## Definition

Given  $\langle S_1, N, \longrightarrow_1 \rangle$  and  $\langle S_2, N, \longrightarrow_2 \rangle$  over  $N$ , relation  $R \subseteq S_1 \times S_2$  is a **bisimulation** iff both  $R$  and its converse  $R^\circ$  are simulations.

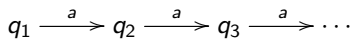
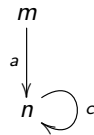
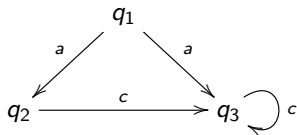
I.e., whenever  $\langle p, q \rangle \in R$  and  $a \in N$ ,

$$(1) \quad p \xrightarrow{a}_1 p' \Rightarrow \langle \exists q' : q' \in S_2 : q \xrightarrow{a}_2 q' \wedge \langle p', q' \rangle \in R \rangle$$

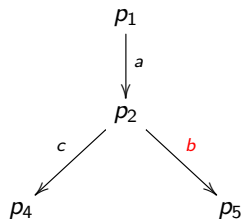
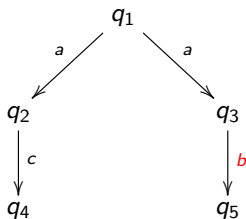
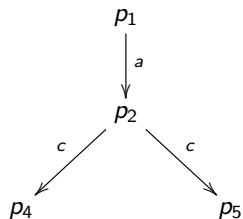
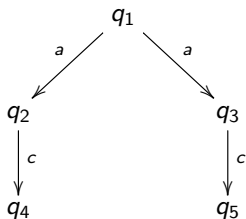
$$(2) \quad q \xrightarrow{a}_2 q' \Rightarrow \langle \exists p' : p' \in S_1 : p \xrightarrow{a}_1 p' \wedge \langle p', q' \rangle \in R \rangle$$



# Examples



# Examples



## After thoughts

- Follows a  $\forall, \exists$  pattern:  $p$  in all its transitions challenge  $q$  which is called to find a matchh to each of those (and conversely)
- Tighter correspondence with transitions
- Based on the information that the transitions convey, rather than on the shape of the LTS
- Local checks on states
- Lack of hierarchy on the pairs of the bisimulation (no temporal order on the checks is required)

which means bisimilarity can be used to reason about infinite or circular behaviours.

## After thoughts

Compare the definition of bisimilarity with

$p == q$  if, for all  $a \in N$

$$(1) \quad p \xrightarrow{a}_1 p' \Rightarrow \langle \exists q' : q' \in S_2 : q \xrightarrow{a}_2 q' \wedge p' == q' \rangle$$

$$(2) \quad q \xrightarrow{a}_2 q' \Rightarrow \langle \exists p' : p' \in S_1 : p \xrightarrow{a}_1 p' \wedge p' == q' \rangle$$

## After thoughts

$p == q$  if, for all  $a \in N$

$$(1) \quad p \xrightarrow{a}_1 p' \Rightarrow \langle \exists q' : q' \in S_2 : q \xrightarrow{a}_2 q' \wedge p' == q' \rangle$$

$$(2) \quad q \xrightarrow{a}_2 q' \Rightarrow \langle \exists p' : p' \in S_1 : p \xrightarrow{a}_1 p' \wedge p' == q' \rangle$$

- The meaning of  $==$  on the pair  $\langle p, q \rangle$  requires having already established the meaning of  $==$  on the derivatives
- ... therefore the definition is **ill-founded** if the state space reachable from  $\langle p, q \rangle$  is infinite or contain loops
- ... this is a **local** but **inherently inductive** definition (to revisit later)

# After thoughts

## Proof method

To prove that two behaviours are bisimilar, find a bisimulation containing them ...

- ... **impredicative** character
- **coinductive** vs **inductive** definition



# Properties

## Definition

$$p \sim q \equiv \langle \exists R :: R \text{ is a bisimulation and } \langle p, q \rangle \in R \rangle$$

## Lemma

- 1 The identity relation  $\text{id}$  is a bisimulation
- 2 The empty relation  $\perp$  is a bisimulation
- 3 The converse  $R^\circ$  of a bisimulation is a bisimulation
- 4 The composition  $S \cdot R$  of two bisimulations  $S$  and  $R$  is a bisimulation
- 5 The  $\bigcup_{i \in I} R_i$  of a family of bisimulations  $\{R_i \mid i \in I\}$  is a bisimulation

# Properties

## Lemma

The bisimilarity relation is an equivalence relation  
(ie, reflexive, symmetric and transitive)

## Lemma

The class of all bisimulations between two LTS has the structure of a **complete lattice**, ordered by set inclusion, whose top is the **bisimilarity** relation  $\sim$ .

# Properties

## Lemma

In a **deterministic** labelled transition system, two states are bisimilar iff they are trace equivalent, i.e.,

$$s \sim s' \Leftrightarrow \text{Tr}(s) = \text{Tr}(s')$$

Hint: define a relation  $R$  as

$$\langle x, y \rangle \in R \Leftrightarrow \text{Tr}(x) = \text{Tr}(y)$$

and show  $R$  is a bisimulation.

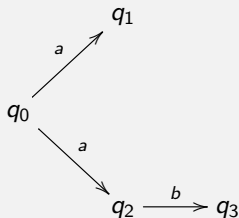
# Properties

## Warning

The bisimilarity relation  $\sim$  is not the symmetric closure of  $\lesssim$

## Example

$$q_0 \lesssim p_0, p_0 \lesssim q_0 \quad \text{but} \quad p_0 \not\sim q_0$$



$$p_0 \xrightarrow{a} p_1 \xrightarrow{b} p_3$$

# Notes

Similarity as the greatest simulation

$$\lesssim \triangleq \bigcup \{S \mid S \text{ is a simulation}\}$$

Bisimilarity as the greatest bisimulation

$$\sim \triangleq \bigcup \{S \mid S \text{ is a bisimulation}\}$$

# Exercises

## P,Q Bisimilar?

$$P = a.P_1$$

$$P_1 = b.P + c.P$$

$$Q = a.Q_1$$

$$Q_1 = b.Q_2 + c.Q$$

$$Q_2 = a.Q_3$$

$$Q_3 = b.Q + c.Q_2$$

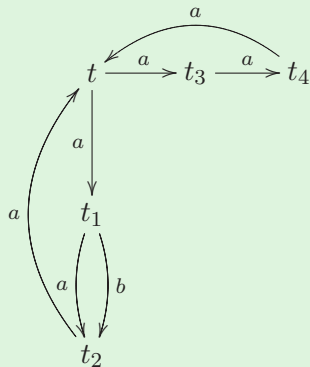
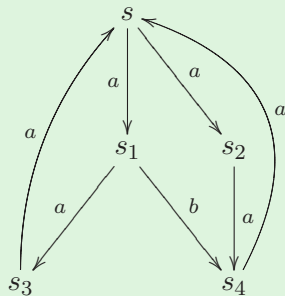
## P,Q Bisimilar?

$$P = a.(b.\mathbf{0} + c.\mathbf{0})$$

$$Q = a.b.\mathbf{0} + a.c.\mathbf{0}$$

# Exercises

## Find a bisimulation



# mCRL2

- Formal **specification language** with an associated toolset
- Used for **modelling**, **validating** and **verifying** concurrent systems and protocols



# mCRL2 - toolset overview

