Introduction to labelled transition systems

Luís Soares Barbosa   José Proença

HASLab - INESC TEC
Universidade do Minho
Braga, Portugal

February 2018

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

- $S = \{s_0, s_1, s_2, ...\}$ 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

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 restriction of $\longrightarrow$.

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

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

where
- $\alpha \in N \cup \overline{N} \cup \{\tau\}$ is an action
- K s a collection of process names or process contants
- I is an indexing set
- $L \subseteq N \cup \overline{N}$ is a set of labels
- f is a function that renames actions s.t. $f(\tau) = \tau$ and $f(\overline{a}) = \overline{f(a)}$
- notation:

$\mathbf{0} = \sum_{i \in \emptyset} P_i$
$P_1 + P_2 = \sum_{i \in \{1,2\}} P_i$
$[f] = [b_1/a_1, \ldots, b_n/a_n]$

Process algebras

Syntax

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

Exercise: Which are syntactically correct?

$$a.b.A + B \quad (1)$$
$$(a.\mathbf{0} + \overline{a}.A) \setminus \{a, b\} \quad (2)$$
$$(a.\mathbf{0} + \overline{a}.A) \setminus \{a, \tau\} \quad (3)$$
$$a.B + [a/b] \quad (4)$$
$$\tau.\tau.B + \mathbf{0} \quad (5)$$
$$(a.B + b.B)[a/a, b/\tau] \quad (6)$$
$$(a.B + \tau.B)[a/b, a/a] \quad (7)$$
$$(a.b.A + \overline{a}.\mathbf{0})|B \quad (8)$$
$$(a.b.A + \overline{a}.\mathbf{0}).B \quad (9)$$
$$(a.b.A + \overline{a}.\mathbf{0}) + B \quad (10)$$
$$(\mathbf{0}|\mathbf{0}) + \mathbf{0} \quad (11)$$

CCS semantics - building an LTS

(act)
$$\frac{}{\alpha.P \xrightarrow{\alpha} P}$$

(sum-j)
$$\frac{P_j \xrightarrow{\alpha} P'_j}{\sum_{i\in I} P_i \xrightarrow{\alpha} P'_j} \quad j \in I$$

(com1)
$$\frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q}$$

(com2)
$$\frac{Q \xrightarrow{\alpha} Q'}{P|Q \xrightarrow{\alpha} P|Q'}$$

(com3)
$$\frac{P \xrightarrow{a} P' \quad Q \xrightarrow{\overline{a}} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$$

(res)
$$\frac{P \xrightarrow{\alpha} P'}{P\backslash L \xrightarrow{\alpha} P'\backslash L} \quad \alpha, \overline{\alpha} \notin L$$

(rel)
$$\frac{P \xrightarrow{\alpha} P'}{P[f] \xrightarrow{f(\alpha)} P'[f]}$$

Exercise: Draw the LTS's

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

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

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

CCS semantics - building an LTS

(act)
$$\frac{}{\alpha.P \xrightarrow{\alpha} P}$$

(sum-j)
$$\frac{P_j \xrightarrow{\alpha} P'_j}{\sum_{i \in I} P_i \xrightarrow{\alpha} P'_j} \quad j \in I$$

(com1)
$$\frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q}$$

(com2)
$$\frac{Q \xrightarrow{\alpha} Q'}{P|Q \xrightarrow{\alpha} P|Q'}$$

(com3)
$$\frac{P \xrightarrow{a} P' \quad Q \xrightarrow{\overline{a}} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$$

(res)
$$\frac{P \xrightarrow{\alpha} P'}{P \backslash L \xrightarrow{\alpha} P' \backslash L} \quad \alpha, \overline{\alpha} \notin L$$

(rel)
$$\frac{P \xrightarrow{\alpha} P'}{P[f] \xrightarrow{f(\alpha)} P'[f]}$$

Exercise: Draw the LTS's

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

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

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

mCRL2

`http://mcrl2.org`

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

mCRL2

Syntax (by example)

$$a.P \rightarrow$$ `a.P`

$$P_1 + P_2 \rightarrow$$ `P1 + P2`

$$P \backslash L \rightarrow$$ `block(L,P)`

$$P[f] \rightarrow$$ `rename(f,P)`

$$a.P | \bar{a}.Q \rightarrow$$ `hide({a},comm({a1|a2->a},a1.P||a2.P))`

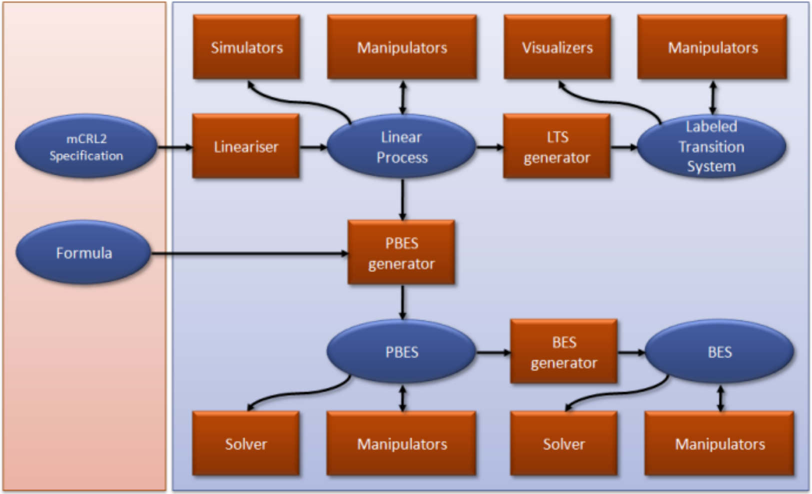
$$a.P | \bar{a}.Q \backslash \{a\} \rightarrow$$ `hide({a},block({a1,a2},comm({a1|a2->a},`
`a1.P||a2.Q)))`

mCRL2

```
act
  coin, coin', coinCom,
  coffee, coffee', coffeeCom, pub';

proc
  CM = coin.coffee'.CM;
  CS = pub'.coin'.coffee.CS;
  CMCS = CM || CS;
  SmUni = hide({coffeeCom,coinCom},
          block({coffee,coffee',coin,coin'},
          comm({coffee|coffee' → coffeeCom,
                coin|coin'     → coinCom},
          CMCS )));
init
  SmUni;
```

mCRL2 toolset overview



– mCRL2 tutorial: Modelling part –

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 $\mathsf{Tr}(s)$, for $s \in S$ is the minimal set satisfying

(1) $\epsilon \in \mathsf{Tr}(s)$

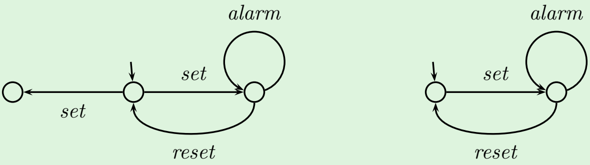
(2) $a\sigma \in \mathsf{Tr}(s) \Rightarrow \langle \exists\ s' \ :\ s' \in S :\ s \xrightarrow{a} s' \wedge \sigma \in \mathsf{Tr}(s') \rangle$

Trace equivalence

Definition

Two states s, r are trace equivalent iff $\mathsf{Tr}(s) = \mathsf{Tr}(r)$
(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

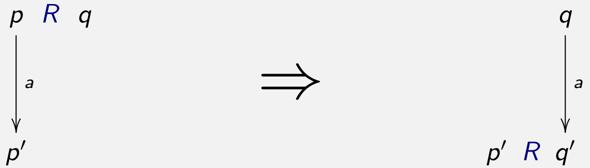
A state q simulates another state p if
every transition from q is corresponded by a transition from p and this capacity is kept along the whole life of the system to which state space q belongs to.

Simulation

Definition

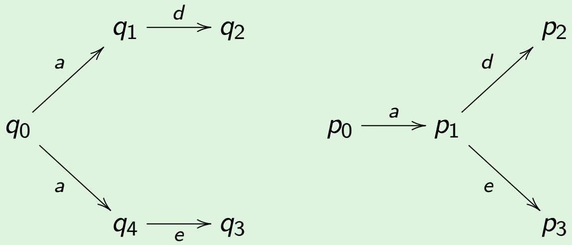
Given $\langle S_1, N, \longrightarrow_1 \rangle$ and $\langle S_2, N, \longrightarrow_2 \rangle$ over N, relation $R \subseteq S_1 \times S_2$ is a simulation iff, for all $\langle p, q \rangle \in R$ and $a \in N$,

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



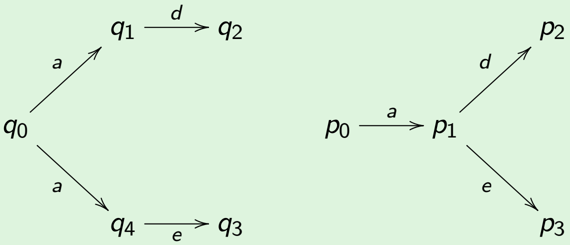
Example

Find simulations



Example

Find simulations



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

Similarity

Definition

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

We say q simulates p.

Lemma

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

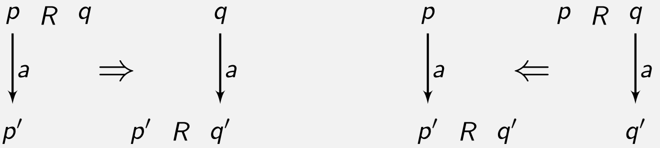
Bisimulation

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

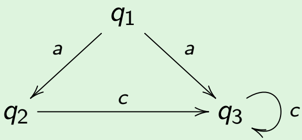
$$(1) \quad p \xrightarrow{a}_1 p' \Rightarrow \langle \exists\, q' : q' \in S_2 : q \xrightarrow{a}_2 q' \wedge \langle p', q' \rangle \in R \rangle$$

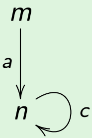
$$(2) \quad q \xrightarrow{a}_2 q' \Rightarrow \langle \exists\, p' : p' \in S_1 : p \xrightarrow{a}_1 p' \wedge \langle p', q' \rangle \in R \rangle$$



Examples

Find bisimulations

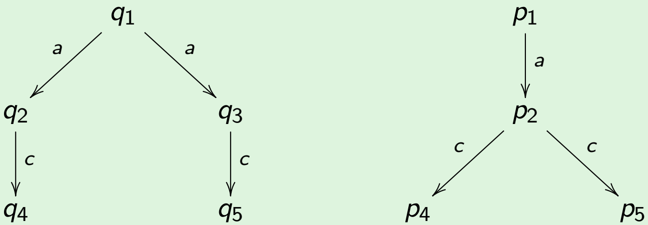


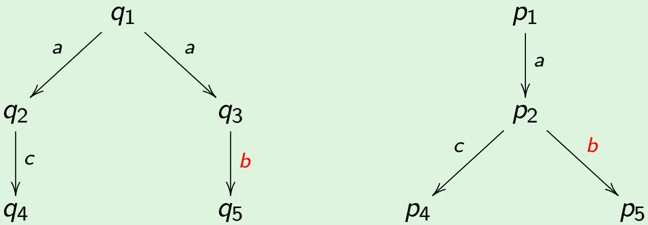


$$q_1 \xrightarrow{a} q_2 \xrightarrow{a} q_3 \xrightarrow{a} \cdots$$

Examples

Find bisimulations





After thoughts

- Follows a $\forall, \exists$ pattern: p in all its transitions challenge q which is called to find a match 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 $a \in N$

$$(1)\;\; p \xrightarrow{a}_1 p' \;\Rightarrow\; \langle \exists\, q' \;:\; q' \in S_2 :\; q \xrightarrow{a}_2 q' \wedge p' == q' \rangle$$
$$(2)\;\; q \xrightarrow{a}_2 q' \;\Rightarrow\; \langle \exists\, p' \;:\; p' \in S_1 :\; p \xrightarrow{a}_1 p' \wedge p' == q' \rangle$$

- 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 id is a bisimulation
2. The empty relation $\bot$ is a bisimulation
3. The converse R° 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.,

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

Hint: define a relation R as

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

and show R is a bisimulation.

Properties

Warning

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

i.e., $\left[p \lesssim q \text{ and } q \lesssim p\right]$ does not imply $\left[p \sim q\right]$

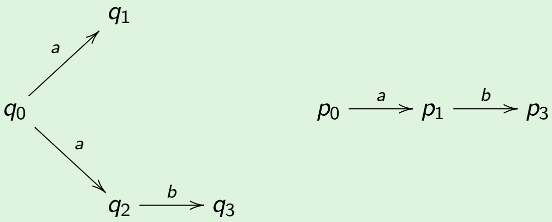
Properties

Warning

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

Example

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



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?

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

$$Q = a.Q_1$$
$$Q_1 = b.Q_2 + c.Q$$
$$Q_2 = a.Q_3$$
$$Q_3 = b.Q + c.Q_2$$

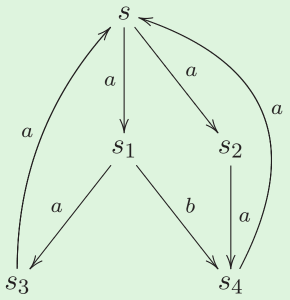
P,Q Bisimilar?

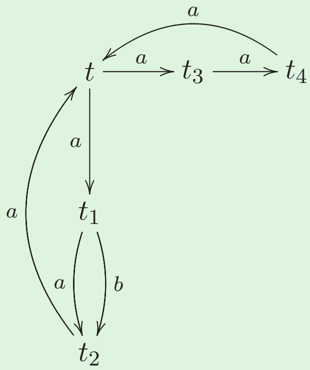
$$P = a.(b.\mathbf{0} + c.\mathbf{0})$$

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

Exercises

Find a bisimulation





More bisimulations

Considering τ-transitions

Weak transition

$$p \stackrel{\alpha}{\Longrightarrow} q \quad \text{iff} \quad p\,(\stackrel{\tau}{\longrightarrow})^*\, q_1 \stackrel{a}{\longrightarrow} q_2\,(\stackrel{\tau}{\longrightarrow})^*\, q$$

$$p \stackrel{\tau}{\Longrightarrow} q \quad \text{iff} \quad p\,(\stackrel{\tau}{\longrightarrow})^*\, q$$

where $\alpha \neq \tau$ and $(\stackrel{\tau}{\longrightarrow})^*$ is the reflexive and transitive closure of $\stackrel{\tau}{\longrightarrow}$.

Weak bisimulation (vs. strong)

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 for all $\langle p, q \rangle \in R$ and $a \in N \cup \{\tau\}$,

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

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

More bisimulations

Considering τ-transitions

Branching bisimulation

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 for all $\langle p, q\rangle \in R$ and $a \in N \cup \{\tau\}$,

(1) if $p \xrightarrow{a}_1 p'$ then either

(1.1) $a = \tau$ and $\langle p', q\rangle \in R$ or

(1.2) $\langle \exists\ q', q'' \in S_2\ ::\ q\ (\xrightarrow{\tau}_2)^*\ q' \xrightarrow{a}_2 q'' \wedge \langle p, q'\rangle \in R \wedge \langle p', q''\rangle \in R\rangle$

(2) if $q \xrightarrow{a}_2 q'$ then either

(2.1) $a = \tau$ and $\langle p', q'\rangle \in R$ or

(2.2) $\langle \exists\ p', p'' \in S_1\ ::\ p\ (\xrightarrow{\tau}_1)^*\ p' \xrightarrow{a}_1 p'' \wedge \langle p', q\rangle \in R \wedge \langle p'', q'\rangle \in R\rangle$