Labelled Transition Systems

Luís S. Barbosa

DI-CCTC
Universidade do Minho
Braga, Portugal

3 March, 2011

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 $\Leftrightarrow$ interaction
- behaviour $\Leftrightarrow$ a structured record of interactions

Reactive systems

Concurrency vs interaction

$$x := 0;$$
$$x := x + 1 \mid x := x + 2$$

- both statements in parallel could read x before it is written
- which values can x take?
- which is the program outcome if exclusive access to memory and atomic execution of assignments is guaranteed?

Labelled Transition Space

Definition

A labelled transition space over a set N of names is a tuple $\langle S, N, \downarrow, \longrightarrow \rangle$ where

- $S = \{s_0, s_1, s_2, ...\}$ is a set of states
- $\downarrow \subseteq S$ is the set of terminating or final states

$$\downarrow s \;\Leftrightarrow\; s \in \downarrow$$

- $\longrightarrow \subseteq S \times N \times S$ is the transition relation, often given as an N-indexed family of binary relations

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

Labelled Transition Space

Morphism

A morphism relating two labelled transition spaces over N, $\langle S, N, \downarrow, \longrightarrow \rangle$ and $\langle S', N, \downarrow', \longrightarrow' \rangle$, is a function $h : S \longrightarrow S'$ st

$$\begin{array}{rcl} s \xrightarrow{a} s' & \Rightarrow & h\ s \stackrel{a}{\longrightarrow}{}'\ h\ s' \\ s \downarrow & \Rightarrow & h\ s \downarrow' \end{array}$$

morphisms preserve transitions and termination

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 $s \xrightarrow{\sigma}^* s''$ and $s'' \xrightarrow{a} s'$ then $s \xrightarrow{\sigma a}^* s'$, 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 $s \xrightarrow{\sigma}^* t$

Labelled Transition System

Labelled Transition System

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

LTS classification

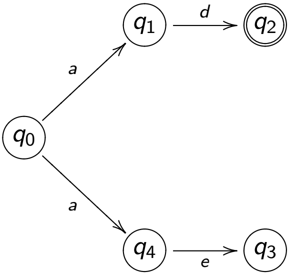
- deterministic
- non deterministic
- finite
- image finite
- ...

Labelled Transition System

Deadlock state

a reachable state that does not terminate and has no outgoing transitions.

Termination vs deadlock



Trace equivalence

Trace (from language theory)

A word $\sigma \in N^*$ is a trace of a state $s \in S$ iff there is another state $t \in S$ such that $s \xrightarrow{\sigma}^* t$

Trace (using ✓ to witness final states)

Tr(s), the set of traces of state s, is the minimal set including

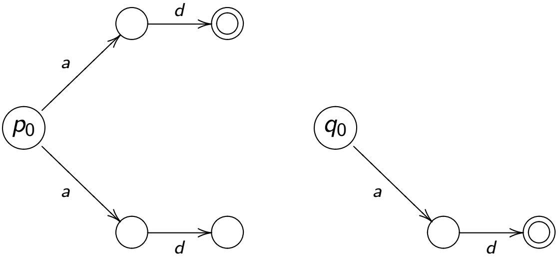
$$\begin{aligned} \epsilon &\in \mathrm{Tr}(s) \\ \checkmark &\in \mathrm{Tr}(s) \quad \text{if} \quad \downarrow s \\ a\sigma &\in \mathrm{Tr}(s) \quad \text{if} \ \exists_t \cdot s \xrightarrow{a} t \wedge \sigma \in \mathrm{Tr}(t) \end{aligned}$$

Trace equivalence

- Two states are trace equivalent if $\mathrm{Tr}(s) = \mathrm{Tr}(s')$
- Two systems are trace equivalent if their initial states are.

Trace equivalence

In any case, fails to preserve deadlock



although preserving sequencing

e.g. before every c an a action b must be done

Language equivalence

Language (from language theory)

A word $\sigma \in N^*$ is a run (or a complete trace) of a state $s \in S$ iff there is another state $t \in S$, such that $s \xrightarrow{\sigma}^* t$ and $\downarrow t$. The language recognized by a state $s \in S$is the set of runs of s

Language (using ✓ to witness final states)

$\mathsf{Lang}(s)$, the language recognized by a state s, is the minimal set including

$$\begin{aligned}
\epsilon &\in \mathsf{Lang}(s) \quad \text{if } s \text{ is a deadlock state} \\
\checkmark &\in \mathsf{Lang}(s) \quad \text{if } \downarrow s \\
a\sigma &\in \mathsf{Lang}(s) \quad \text{if } \exists_t \cdot s \xrightarrow{a} t \wedge \sigma \in \mathsf{Lang}(t)
\end{aligned}$$

Language equivalence

Language equivalence

- Two states are language equivalent if $\mathsf{Lang}(s) = \mathsf{Lang}(s')$, i.e., if both recognize the same language.
- Two systems are language equivalent if their initial states are.

Automata

Back to old friends?

automaton behaviour $\Leftrightarrow$ accepted language

Recall that finite automata recognize regular languages, i.e. generated by

- $L_1 + L_2 \triangleq L_1 \cup L_2$ (union)
- $L_1 \cdot L_2 \triangleq \{st \mid s \in L_1, t \in L_2\}$ (concatenation)
- $L^* \triangleq \{\epsilon\} \cup L \cup (L \cdot L) \cup (L \cdot L \cdot L) \cup \ldots$ (iteration)

Automata

There is a syntax to specify such languages:

$$E \quad ::= \quad \epsilon \mid a \mid E + E \mid E\,E \mid E^*$$

where $a \in \Sigma$.

- which regular expression specifies $\{a, bc\}$?
- and $\{ca, cb\}$?

and an algebra of regular expressions:

$$\begin{aligned}(E_1 + E_2) + E_3 &= E_1 + (E_2 + E_3) \\ (E_1 + E_2)\,E_3 &= E_1\,E_3 + E_2\,E_3 \\ E_1\,(E_2\,E_1)^* &= (E_1\,E_2)^*\,E_1\end{aligned}$$

After thoughts

... need more general models and theories:

- Several interaction points ($\neq$ functions)
- Need to distinguish normal from anomolous termination (eg deadlock)
- Non determinisim should be taken seriously: the notion of equivalence based on accepted language is blind wrt non determinism
- Moreover: the reactive character of systems entail that not only the generated language is important, but also the states traversed during an execution of the automata.

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.

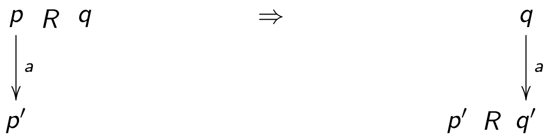
Simulation

Definition

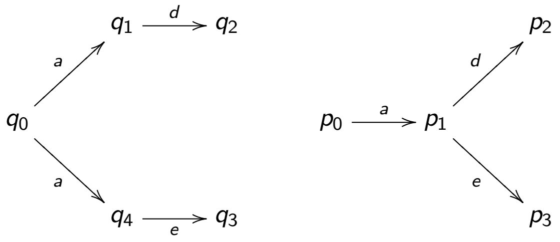
Given $\langle S_1, N, \downarrow_1, \longrightarrow_1 \rangle$ and $\langle S_2, N, \downarrow_2, \longrightarrow_2 \rangle$ over N, relation $R \subseteq S_1 \times S_2$ is a simulation iff, for all $\langle p, q \rangle \in R$ and $a \in N$,

(1) $p \downarrow_1 \;\Rightarrow q \downarrow_2$

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



Example



$q_0 \lesssim p_0$ 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

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

Lemma

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

Bisimulation

Definition

Given $\langle S_1, N, \downarrow_1, \longrightarrow_1 \rangle$ and $\langle S_2, N, \downarrow_2, \longrightarrow_2 \rangle$ 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 $a \in N$,

$$(1) \quad p \downarrow_1 \; \Leftrightarrow q \downarrow_2$$

$$(2.1) \quad p \xrightarrow{a}_1 p' \;\Rightarrow\; \langle \exists\, q' \;:\; q' \in S_2 :\; q \xrightarrow{a}_2 q' \wedge \langle p', q' \rangle \in R \rangle$$

$$(2.1) \quad q \xrightarrow{a}_2 q' \;\Rightarrow\; \langle \exists\, p' \;:\; p' \in S_1 :\; p \xrightarrow{a}_1 p' \wedge \langle p', q' \rangle \in R \rangle$$

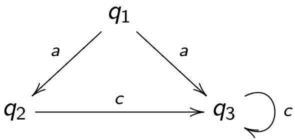
Bisimulation

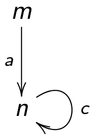
The Game characterization

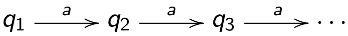
Two players R and I discuss whether the transition structures are mutually corresponding

- R starts by chosing a transition
- I replies trying to match it
- if I succeeds, R plays again
- R wins if I fails to find a corresponding match
- I wins if it replies to all moves from R and the game is in a configuration where all states have been visited or R can't move further. In this case is said that I has a wining strategy

Examples







Bisimilarity

Definition

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

Lemma

1. The identity relation id is a bisimulation
2. The empty relation $\perp$ 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

Bisimilarity

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 $\sim$.

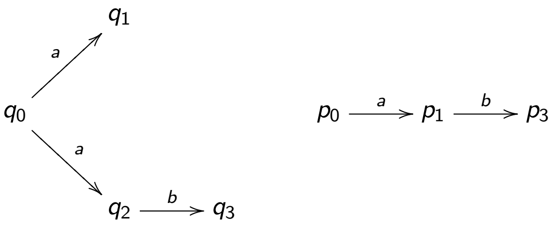
Bisimilarity

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

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

cf relational translation of definitions
$\lesssim$ and $\sim$ as greatest fix points (Tarski's theorem)

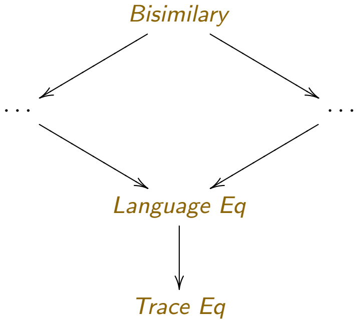
Notes

Complexity

- Virtually all forms of bisimulation can be determined in polynomial time on finite state transition systems
- ... whereas trace, or language equivalence are in general difficult (P-space hard)

Notes

The Van Glabbeek linear - branching time spectrum



... collapses for deterministic transition systems: why?

Abstraction

Main idea:
Take a set of actions as internal or non-observable

Approaches

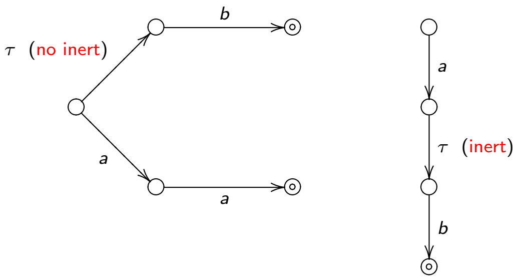
- R. Milner's weak bisimulation [Mil80]
- Van Glabbeek and Weijland's branching bisimulation [GW96]

Internal actions

τ abstracts internal activity

inert τ: internal activity is undetectable by observation

non inert τ: internal activity is indirectly visible



Branching bisimulation

- Intuition similar to that of strong bisimulation: But now, instead of letting a single action be simulated by a single action, an action can be simulated by a sequence of internal transitions, followed by that single action.
- An internal action τ can be simulated by any number of internal transitions (even by none).
- If a state can terminate, it does not need to be related to a terminating state: it suffices that a terminating state can be reached after a number of internal transitions.

Branching bisimulation

Definition

Given $\langle S_1, N, \downarrow_1, \longrightarrow_1\rangle$ and $\langle S_2, N, \downarrow_2, \longrightarrow_2\rangle$ over N, relation $R \subseteq S_1 \times S_2$ is a branching bisimulation iff for all $\langle p, q\rangle \in R$ and $a \in N$,

1. If $p \stackrel{a}{\longrightarrow}_1 p'$, then
   - either $a = \tau$ and $p' R q$
   - or, there is a sequence $q \stackrel{\tau}{\longrightarrow}_2 \cdots \stackrel{\tau}{\longrightarrow}_2 q'$ of (zero or more) τ-transitions such that pRq' and $q' \stackrel{a}{\longrightarrow}_2 q''$ with $p' R q''$.
2. If $p \downarrow_1$, then there is a sequence $q \stackrel{\tau}{\longrightarrow}_2 \cdots \stackrel{\tau}{\longrightarrow}_2 q'$ of (zero or more) τ-transitions such that pRq' and $q' \downarrow_2$.

1'., 2'. symmetrically ...

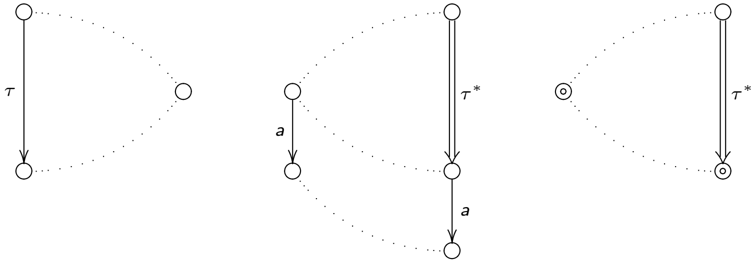
Branching bisimilarity

Definition

$$p \approx_b q \;\Leftrightarrow\; \langle \exists\, R \;::\; R \text{ is a branching bisimulation and } \langle p, q\rangle \in R \rangle$$

Branching bisimulation

... preserves the branching structure



Branching bisimilarity

… does not preserve τ-loops

satisfying a notion of fairness: if a τ-loop exists, then no infinite execution sequence will remain in it forever if there is a possibility to leave

Branching bisimilarity

Problem

If an alternative is added to the initial state then transition systems that were branching bisimilar may cease to be so.

Example: add a *b*-labelled branch to the initial states of

Rooted branching bisimilarity

Startegy

Impose a rootedness condition [R. Milner, 80]:

Initial τ-transitions can never be inert, *i.e.*, two states are equivalent if they can simulate each other?s initial transitions, such that the resulting states are branching bisimilar.

Rooted branching bisimulation

Definition

Given $\langle S_1, N, \downarrow_1, \longrightarrow_1\rangle$ and $\langle S_2, N, \downarrow_2, \longrightarrow_2\rangle$ over N, relation $R \subseteq S_1 \times S_2$ is a rooted branching bisimulation iff

1. it is a branching bisimulation
2. for all $\langle p, q\rangle \in R$ and $a \in N$,
   - If $p \xrightarrow{a}_1 p'$, then there is a $q' \in S_2$ such that $q \xrightarrow{a}_2 q'$ and $p' \approx q'$
   - If $q \xrightarrow{a}_2 q'$, then there is a $p' \in S_1$ such that $p \xrightarrow{a}_1 p'$ and $p' \approx q'$

Rooted branching bisimilarity

Definition

$p \approx_{rb} q \;\Leftrightarrow\; \langle \exists\, R \;::\; R$ is a rooted branching bisimulation and $\langle p, q\rangle \in R\rangle$

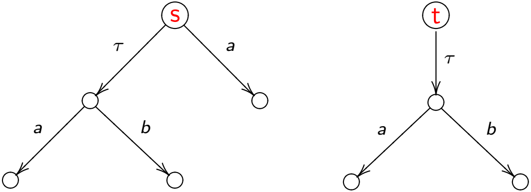
Lemma

$$\sim \;\subseteq\; \approx_{rb} \;\subseteq\; \approx_b$$

Of course, in the absence of τ actions, $\sim$ and $\approx_b$ coincide.

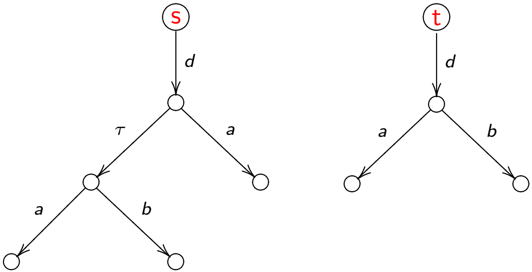
Example

branching but not rooted



Example

rooted branching bisimilar



Weak bisimulation

Definition [Milner,80]

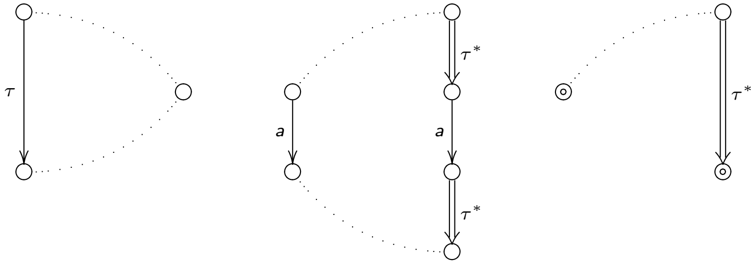
Given $\langle S_1, N, \downarrow_1, \longrightarrow_1\rangle$ and $\langle S_2, N, \downarrow_2, \longrightarrow_2\rangle$ over N, relation $R \subseteq S_1 \times S_2$ is a weak bisimulation iff for all $\langle p, q\rangle \in R$ and $a \in N$,

1. If $p \xrightarrow{a}_1 p'$, then
   - either $a = \tau$ and $p' R q$
   - or, there is a sequence $q \xrightarrow{\tau}_2 \cdots \xrightarrow{\tau}_2 t \xrightarrow{a}_2 t' \xrightarrow{\tau}_2 \cdots \xrightarrow{\tau}_2 q'$ involving zero or more τ-transitions, such that $p' R q'$.
2. If $p \downarrow_1$, then there is a sequence $q \xrightarrow{\tau}_2 \cdots \xrightarrow{\tau}_2 q'$ of (zero or more) τ-transitions such that $q' \downarrow_2$.

1'., 2'. symmetrically ...

Weak bisimulation

... does not preserve the branching structure



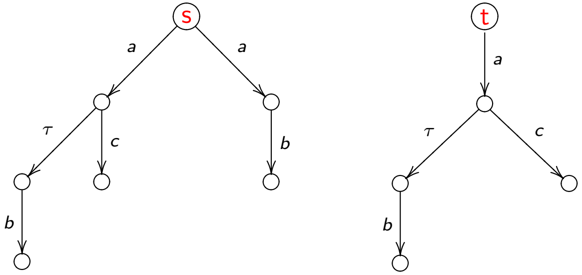
Weak bisimilarity

Definition

$$p \approx_w q \;\Leftrightarrow\; \langle \exists\, R \;::\; R \text{ is a branching bisimulation and } \langle p, q\rangle \in R \rangle$$

Example

weak but not branching



Rooted weak bisimulation

Definition

Given $\langle S_1, N, \downarrow_1, \longrightarrow_1 \rangle$ and $\langle S_2, N, \downarrow_2, \longrightarrow_2 \rangle$ over N, relation $R \subseteq S_1 \times S_2$ is a rooted branching bisimulation iff for all $\langle p, q \rangle \in R$ and $a \in N$,

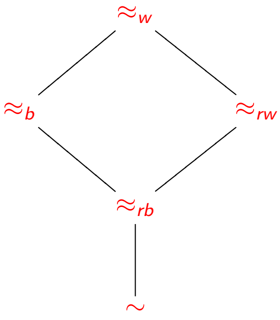
- If $p \xrightarrow{\tau}_1 p'$, then there is a non empty sequence of τ such that $q \xrightarrow{\tau}_2 \xrightarrow{\tau}_2 \ldots \xrightarrow{\tau}_2 \xrightarrow{\tau}_2 q'$ and $p' \approx_w q'$
- Symmetrically ...

Rooted weak bisimilarity

Definition

$$p \approx_{rw} q \;\Leftrightarrow\; \langle \exists\, R \;::\; R \text{ is a rooted weak bisimulation and } \langle p, q \rangle \in R \rangle$$

Lemma



(ordered by $\subseteq$)

The questions to follow ...

- We already have a semantic model for reactive systems. With which language shall we describe them?
- How to compare and transform such systems?
- How to express and prove their proprieties?

$\rightsquigarrow$ process languages and calculi
cf. CCS (Milner, 80), CSP (Hoare, 85),
ACP (Bergstra & Klop, 82),
π-*calculus* (Milner, 89), among many others

$\rightsquigarrow$ modal (temporal, hybrid) logics