Models and logics for reactive systems (Non deterministic systems)

Luís S. Barbosa

HASLab - INESC TEC
Universidade do Minho
Braga, Portugal

13 March 2014

Recalling

The syllabus:

- Models and logics for reactive systems
  - Classical (non deterministic) (mCRL2)
  - Timed (with real time constraints) (Uppaal)
  - Probabilistic (PRISM)
  - Cyber-physical (KeYmaera)
- Architecture for reactive systems
  - ...

Goal

To describe and analyse the behaviour of reactive systems

- supporting their design:
  - synchronization, scheduling, fairness, absence of deadlocks, ...
- analysing their performance:
  - queue throughput, response time in real-time systems, ...
- verifying their properties:
  - mutual exclusion, no deadlocks, liveness, ...

Reactive systems

Characteristics

- on-going interaction with environment leads to reactive rather than transformational behaviour
- concurrent, rather than sequential composition as a norm
- infinite behaviour, rather than terminating computation

Reactive systems

Concurrency vs interaction

$$\mathsf{x} := 0;$$
$$\mathsf{x} := \mathsf{x} + 1 \mid \mathsf{x} := \mathsf{x} + 2$$

- both statements in parallel could read x before it is written
- thus x be assigned 1, 2 or 3
- but 3 is the only possible outcome if exclusive access to memory and atomic execution of assignments is guaranteed

Reactive systems

This means that in the project of reactive systems the precise description of the mechanisms of both

- concurrency (interleaving, true concurrency, ...
- and interaction
  (shared memory, message passing, synchronous/asynchronous, ...)

is crucially important!

Reactive systems

… are often safety/mission critical

which means that correct and effective behaviour has to be ensured:

- Safety properties: *Nothing bad is going to happen*
  e.g. "at most one process in the critical section"
- Liveness properties: *Eventually something good will happen*
  e.g. "the server will finally answer'
- Fairness properties: *No component will starve to death*
  e.g. "any process requiring entry to the critical section will eventually be admitted"
- Performance properties: *The system will conform to certain QoS requirements*
  e.g. "an acknowledgement is sent in less than 10 ms"

Reactive systems

The formal analysis of reactive system and the verification of their properties requires suitable

$\boxed{\text{mathematical models}}$

Labelled transition systems

- basic model of a computational system with a natural representation of non determinism
- state vs event based descriptions rooted on the duality between states and transitions
- bare structure to be enriched in different directions: adding structure to states, transitions, or both
- able to be equipped with an algebra (compositionality)
- provides an interpretation structure for modal logics

Labelled Transition Space

Definition

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

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

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

Labelled Transition Space

Morphism

A morphism relating two labelled transition spaces 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

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 $s \xrightarrow{\sigma}^* t$

Labelled Transition System

Labelled Transition System

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

LTS classification

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

New LTS from old

Product

$$\frac{p \xrightarrow{a} p'}{p \mid_K q \xrightarrow{a} p' \mid_K q}\ a \notin K \qquad \frac{q \xrightarrow{a} q'}{p \mid_K q \xrightarrow{a} p \mid_K q'}\ a \notin K$$

$$\frac{p \xrightarrow{a} p' \quad q \xrightarrow{a} q'}{p \mid_K q \xrightarrow{a} p' \mid_K q'}\ a \in K$$

- synchronous, multiparty interaction
- ... other interaction disciplines are possible

New LTS from old

Abstraction

$$\frac{p \xrightarrow{a} p'}{\mathrm{hide}_K\ p \xrightarrow{a} \mathrm{hide}_K\ p'}\ a \notin K \qquad \frac{p \xrightarrow{a} p'}{\mathrm{hide}_K\ p \xrightarrow{\tau} \mathrm{hide}_K\ p'}\ a \in K$$

- τ represents the unobservable, internal action
- product + abstraction = composition

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 equivalence

- Two states are trace equivalent if they have the same set of traces
- Two systems are trace equivalent if their initial states are.

Automata

Back to old friends?

$$\boxed{\text{automaton behaviour} \quad \Leftrightarrow \quad \text{accepted language}}$$

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:

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

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
(but maybe along similar lines):

- Several interaction points ($\neq$ functions)
- 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 entails 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 $\langle S_1, N, \longrightarrow_1\rangle$ 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 $a \in N$,

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

$$\begin{tikzcd}
p \arrow[d, "a"] & R & q & & \Rightarrow & & & & q \arrow[d, "a"] \\
p' & & & & & & p' & R & q'
\end{tikzcd}$$

Example

$$
\begin{tikzcd}
& q_1 \arrow[r, "d"] & q_2 & & & p_2 \\
q_0 \arrow[ur, "a"] \arrow[dr, "a"'] & & & p_0 \arrow[r, "a"] & p_1 \arrow[ur, "d"] \arrow[dr, "e"'] & \\
& q_4 \arrow[r, "e"'] & q_3 & & & p_3
\end{tikzcd}
$$

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

Similarity

Definition

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

Lemma

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

Bisimulation

Definition

Given $\langle S_1, N, \longrightarrow_1 \rangle$ 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

$$
\begin{tikzcd}
& q_1 \arrow[dl, "a"'] \arrow[dr, "a"] & \\
q_2 \arrow[rr, "c"] & & q_3 \arrow[loop right, "c"]
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
m \arrow[d, "a"'] \\
n \arrow[loop right, "c"]
\end{tikzcd}
$$

$$
\begin{tikzcd}
q_1 \arrow[r, "a"] & q_2 \arrow[r, "a"] & q_3 \arrow[r, "a"] & \cdots
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
h \arrow[loop right, "a"]
\end{tikzcd}
$$

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 $\sim$.

Bisimulation

Definition (alternative)

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

$$\langle p, q\rangle \in R \;\Leftrightarrow\; \langle \forall\, a, C \;:\; a \in N, C \in (S_1 \cup S_2)/R :\; p \xrightarrow{a}_1 C \Leftrightarrow q \xrightarrow{a}_2 C\rangle$$

where, for an equivalence class C,

$$p \xrightarrow{a} C \;\Leftrightarrow\; \langle \exists\; p' \;:\; p' \in C :\; p \xrightarrow{a} p'\rangle$$

Bisimilarity

Warning

$\boxed{\text{The bisimilarity relation } \sim \text{ is not the symmetric closure of } \lesssim}$

Example

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

$$
\begin{tikzcd}
 & q_1 & \\
q_0 \arrow[ur, "a"] \arrow[dr, "a"'] & & \\
 & q_2 \arrow[r, "b"] & q_3
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
p_0 \arrow[r, "a"] & p_1 \arrow[r, "b"] & p_3
\end{tikzcd}
$$

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

The Van Glabbeek linear - branching time spectrum

$$
\begin{tikzcd}
 & \textit{Bisimilary} \arrow[dl] \arrow[dr] & \\
\cdots \arrow[dr] & & \cdots \arrow[dl] \\
 & \textit{Trace Eq} &
\end{tikzcd}
$$

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)

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]

Abstraction

- Intuition similar to that of strong bisimulation: But now, instead of letting a single action be simulated by a single action, within an envelope of internal transitions
- An internal action τ can be simulated by any number of internal transitions (even by none).

Weak bisimulation

Definition [Milner,80]

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 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. symmetrically ...

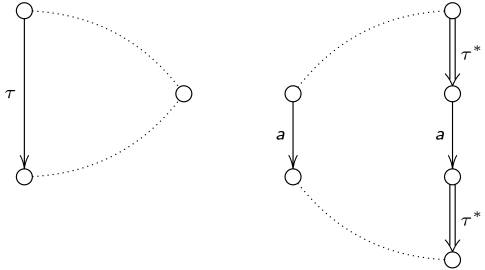
Note

it corresponds to a strong bisimulation over $\stackrel{s}{\Longrightarrow}$ for $s \in N^*$

Weak bisimilarity

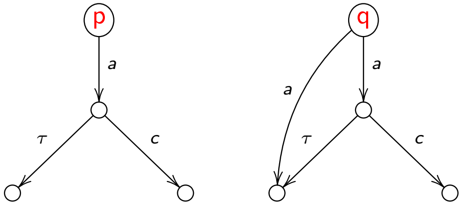
Definition

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



Example

abstracts over internal actions but branching is not preserved



Branching 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 branching 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 $q \xrightarrow{\tau}_2 \cdots \xrightarrow{\tau}_2 q'$ of (zero or more) τ-transitions such that pRq' and $q' \xrightarrow{a}_2 q''$ with $p' R q''$.
2. symmetrically ...

Exercise

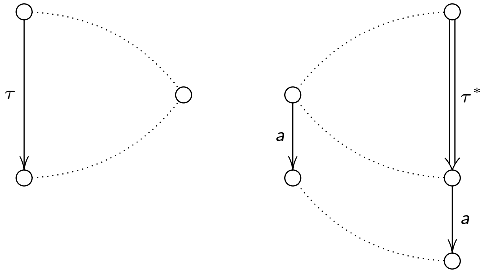
Give an alternative definition in terms of equivalence classes

Branching bisimilarity

Definition

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

… preserves the branching structure



Divergence

Branching and weak bisimilarity do 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

Exercise

Modify the corresponding definitions to enforce preserving divergence

The rootedness condition

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 $\langle S_1, N, \longrightarrow_1 \rangle$ and $\langle S_2, N, \longrightarrow_2 \rangle$ 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 $p' \approx_b q'$
   - If $q \xrightarrow{a}_2 q'$, then there is a $p' \in S_1$ such that $p \xrightarrow{a}_1 p'$ and $p' \approx_b q'$

Rooted branching bisimilarity

Definition

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

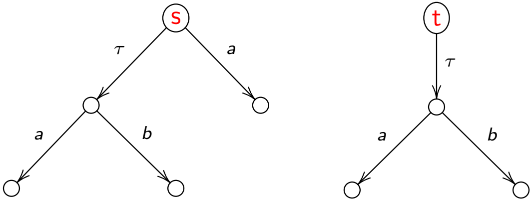
Lemma

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

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

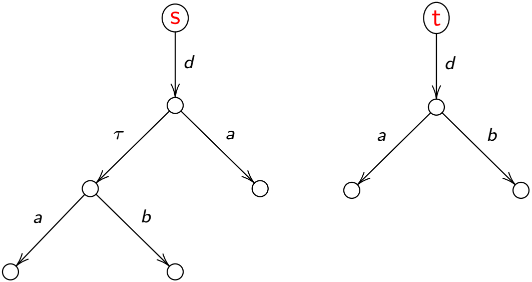
Example

branching but not rooted



Example

rooted branching bisimilar



Rooted weak bisimilarity

The same recipe applies to 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

$$\begin{tikzcd}
 & \approx_w \arrow[dl, dash] \arrow[dr, dash] & \\
\approx_b \arrow[dr, dash] & & \approx_{rw} \arrow[dl, dash] \\
 & \approx_{rb} \arrow[d, dash] & \\
 & \sim &
\end{tikzcd}$$

(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