Labelled Transition Systems

Luís Soares Barbosa

Interaction & Concurrency Course Unit (Lcc)

Universidade do Minho

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

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 System

Definition

A LTS 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

$$\downarrow s \;\equiv\; s \in \downarrow$$

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

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

Labelled Transition System

Morphism

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

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

$\boxed{\text{morphisms preserve transitions and termination}}$

Labelled Transition System

System

Given a LTS $\langle S, N, \downarrow, \longrightarrow \rangle$, 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 $s \xrightarrow{\sigma}^{*} t$

Labelled Transition System

Alternative characterization (coalgebraic)

A morphism $h : \langle S, \text{next} \rangle \longrightarrow \langle S', \text{next}' \rangle$ is a function $h : S \longrightarrow S'$ st the following diagram commutes

$$\begin{array}{ccc} S \times N & \xrightarrow{\text{next}} & \mathcal{P}S \\ {\scriptstyle h \times id}\downarrow & & \downarrow{\scriptstyle \mathcal{P}h} \\ S' \times N & \xrightarrow{\text{next}'} & \mathcal{P}S' \end{array}$$

i.e.,

$$\mathcal{P}h \cdot \text{next} \quad = \quad \text{next}' \cdot (h \times id)$$

or, going pointwise,

$$\{h\ x \mid x \in \text{next}\ \langle s, a \rangle\} \quad = \quad \text{next}'\ \langle h\ s, a \rangle$$

Labelled Transition System

Alternative characterization (coalgebraic)

A morphism $h : \langle S, \mathrm{next} \rangle \longrightarrow \langle S', \mathrm{next}' \rangle$

- preseves transitions:

$$s' \in \mathrm{next}\ \langle s, a \rangle \Rightarrow h\ s' \in \mathrm{next}'\ \langle h\ s, a \rangle$$

- reflects transitions:

$$r' \in \mathrm{next}'\ \langle h\ s, a \rangle \Rightarrow \langle \exists\ s' \in S\ :\ s' \in \mathrm{next}\ \langle s, a \rangle :\ r' = h\ s' \rangle$$

(why?)

Comparison

- Both definitions coincide at the object level:

$$\langle s, a, s' \rangle \in T \quad \equiv \quad s' \in \mathsf{next}\ \langle s, a \rangle$$

- Wrt morphisms, the relational definition is more general, corresponding, in coalgebraic terms to

$$\mathcal{P}h \cdot \mathsf{next} \quad \subseteq \quad \mathsf{next}' \cdot (h \times id)$$

Automata

Back to old friends?

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

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

- $L_1 + L_2 \mathrel{\widehat{=}} \; L_1 \cup L_2$ (union)
- $L_1 \cdot L_2 \mathrel{\widehat{=}} \; \{st \mid s \in L_1, t \in L_2\}$ (concatenation)
- $L^* \mathrel{\widehat{=}} \; \{\epsilon\} \cup L \cup (L \cdot L) \cup (L \cdot L \cdot L) \cup \ldots$ (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:

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

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 anomalous termination (eg deadlock)
- Nondeterminisim should be taken seriously: the notion of equivalence based on accepted language is blind wrt nondeterminism
- Moreover: the reactive characters of systems entail that not only the generated language is important, but also the states traversed during an execution of the automata.