Notiztitel 09.03.2005

I.8. Process Algebra

We add datavariables etc.

To do

I.9 Action labelled vs. state labelled models

A Kripke structures vs LTS

B CTL vs. aCTL

C Bisimulation and branching bisimulation on Kripke structures

Todo

I.10. Semantic equivalence vs. logical equivalence

To do

Part II Markov Algebra and Model Checking

II.1 CTMC algebra

A] Syntax: Consider the language nMA (nano Markov Algebra) generated by the following grammar:

$$E ::= 0 \mid E + E \mid (\lambda).E \mid X \mid \mathrm{rec}\ X : E \mid E \parallel E \qquad \lambda \in \mathbb{R}^+$$

delay prefix

Examples: $(5).0 + (0,3).0$

B) Semantics

The semantics of a term $E \in nMA$ is the $\langle$ multiTS $(nMA, \mathbb{R}^+, \to, E)$

where $\to : (nMA \times \mathbb{R}^+ \times nMA) \to \mathbb{N}_0$

is the smallest multi relation satisfying

$$\overline{(\lambda).E \xrightarrow{\lambda} E}$$

$$\frac{E \xrightarrow{\lambda} E'}{rec\,X{:}E \xrightarrow{\lambda} E'\{^{rec\,X:E}/_X\}}$$

$$\frac{E \xrightarrow{\lambda} E'}{E+F \xrightarrow{\lambda} E'}$$

$$\frac{E \xrightarrow{\lambda} E'}{F+E \xrightarrow{\lambda} E'}$$

$$\frac{E \xrightarrow{\lambda} E'}{E \parallel F \xrightarrow{\lambda} E' \parallel F}$$

$$\frac{E \xrightarrow{\lambda} E'}{F \parallel E \xrightarrow{\lambda} F \parallel E'}$$

Examples:

$$
\begin{tikzcd}
 & (\lambda).0 \mid\mid\mid (\mu).0 \arrow[dl, "\lambda"'] \arrow[dr, "\mu"] & \\
{} \arrow[dr, "\mu"'] & & {} \arrow[dl, "\lambda"] \\
 & 0 \mid\mid\mid 0 &
\end{tikzcd}
$$

Why is this sound?

Memory-less property!

More examples:

$$\begin{tikzcd}
(\mu).0 + (\mu).0 \arrow[d, bend right, "\mu"'] \arrow[d, bend left, "\mu"] \\
0
\end{tikzcd}$$

multi-transition relation

$$\begin{tikzcd}
(\mu).0 + (\nu).0 \arrow[d, bend right, "\mu"'] \arrow[d, bend left, "\nu"] \\
0
\end{tikzcd}$$

$$\begin{tikzcd}
 & (\mu).0 + (\mu).(0+0) \arrow[dl, "\mu"'] \arrow[dr, "\mu"] & \\
0 & & 0+0
\end{tikzcd}$$

$$\begin{tikzcd}
 & (\mu).0 + (\nu).(0+0) \arrow[dl, "\mu"'] \arrow[dr, "\nu"] & \\
0 & & 0+0
\end{tikzcd}$$

C) Bisimulation semantics

Let $P \in nMA$ and let $S \subseteq nMA$.

We let $$r(P,S) = \sum_{P' \in S} \sum_{\substack{\lambda \in \mathbb{R}^+ \\ \to(P,\lambda,P') > 0}} \lambda \cdot \text{"}\!\to\!\text{"}(P,\lambda,P')$$

Example:

$$r\left((\lambda).0 + (\lambda.0) + (\mu).0 + (\mu).(0+0), \{0, 0+0\}\right) =$$

$$\underbrace{\mu}_{(0+0)} + \underbrace{(\mu \cdot 1 + \lambda \cdot 2)}_{(0)} = 2\mu + 2\lambda$$

Def: An equivalence relation $\mathcal{E}$ on nMA is a "lumping bisimulation" if $(P,Q) \in \mathcal{E}$ implies for all equivalence classes $C \in {}^{nMA}/_{\mathcal{E}}$ that

$$\boxed{r(P,C) = r(Q,C)}.$$

Two processes P and Q are called "lumping bisimilar" written "$P \simeq Q$" if the pair is contained in some lumping bisimulation.

Lemma 1-4 hold *mutatis mutandis*.

D Towards μMA: adding synchronisation.

There are two approaches:

- Take pairs "(a, λ)" meaning: a is delayed according to "λ".

$$\frac{E \xrightarrow{a,\lambda} E' \quad \wedge \quad F \xrightarrow{a,\mu} F'}{E \parallel_A F \xrightarrow{a,\, f(\lambda,\mu)} E' \parallel_A F'} \quad a \in A$$

$f : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$, but what?

PEPA [Hillston 94]

TIPP [Herzog 93]

- Take two prefixes "$a.E$" and "$(\lambda).E$".

Done here

$\boxed{E}$ milli Markov Process Algebra

Syntax:

$$E ::= 0 \mid E+E \mid a.E \mid (\lambda).E \mid X \mid \mathrm{rec}\ X : E \mid E \parallel_A E \mid \boxed{E}_A \mid f(E)$$

Semantics: SOS as before, plus

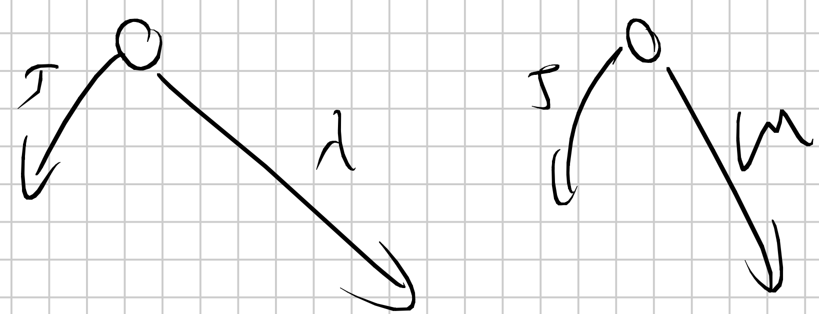
$$\frac{E \xrightarrow{\lambda} E'}{E \parallel_A F \xrightarrow{\lambda} E' \parallel_A F} \qquad \frac{E \xrightarrow{\lambda} E'}{F \parallel_A E \xrightarrow{\lambda} F \parallel_A E'}$$

$\boxed{F}$ Bisimulation on mMPA.

An equivalence $\mathcal{E}$ on mMPA is a "lumping bisimulation" if $(P,Q) \in \mathcal{E}$ implies for all actions $a \in Act$ and all equivalence classes $C \in {}^{mMPA}/_{\mathcal{E}}$ that

i) $P \xrightarrow{a} P'$ implies $\exists Q' : Q \xrightarrow{a} Q'$ such that $(P', Q') \in \mathcal{E}$

ii) if $P \not\xrightarrow{\tau}$ then $r(P, C) = r(Q, C)$



II.2. DTMC Model checking

see (or better: listen) the DivX-file

II. 3. CTMC Model checking

also see (or better: listen)
the DivX-file