Concurrency Theory (WS 2016) Out: Wed, 21 Dec Due: Wed, 11 Jan

Exercise Sheet 9

D'Osualdo, Lederer, Schneider Technische Universität Kaiserslautern

Please submit your solution before Wed, 11 Jan at 10h. You can send them via mail.

Problem 1: Bisimulations

Let $(S, \to)$ be a labelled transition system over Act. For a $s_0 \in S$ we call $T = (S, \to, s_0)$ an *initialised* LTS. In what follows we omit "initialised" and call it simply LTS. We say that $T_1 = (S_1, \to_1, s_0^1)$ strongly (bi)simulates $T_2 = (S_2, \to_2, s_0^2)$ when s_0 is (bi)similar to s_1.

Merging two states $s_1, s_2 \in S$ yields a LTS $T_{s_1 \leftarrow s_2} := (S \setminus \{s_2\}, \to', s_0)$ where $\to'$ is $\to$ except that every transition leading to s_2 now leads to s_1 and every transition stemming from s_2 now stems from s_1.

Let $R \subseteq S \times S$ be an equivalence relation, we write $[s]_R = \{s' \in S \mid s \; R \; s'\}$ for the equivalence class of s. We define the *quotient* of T under R as $T/_R := (S/_R, \to, [s_0]_R)$ where $S/_R = \{[s]_R \mid s \in S\}$ and $[s]_R \xrightarrow{\alpha} [s']_R$ if $s \xrightarrow{\alpha} s'$.

For $q \in S$, $\mathcal{L}_q(T) := \{w \in Act^* \mid \exists p \in S : q \xrightarrow{w} p\}$ denotes the set of traces starting at q.

a) Show that $T_{s_1 \leftarrow s_2}$ simulates T for every $s_1, s_2 \in S$.

b) Show that $T/_R$ simulates T for every equivalence relation $R \subseteq S \times S$.

c) Show that if $R \subseteq S \times S$ is a bisimulation then $T/_R$ is bisimilar to T.

d) Let T_1, T_2 be LTS over Act. Show that if R is a simulation from T_1 to T_2, then $\mathcal{L}_{q_1}(T_1) \subseteq \mathcal{L}_{q_2}(T_2)$ for any $q_1 R q_2$.

e) Give two LTS that simulate each other but are not bisimilar.

Problem 2: Standard Form

Show that every CCS process is structurally congruent to a process of the form

$$\boldsymbol{\nu} a_1. \cdots \boldsymbol{\nu} a_m.(M_1 \parallel \cdots \parallel M_n)$$

for some $n, m \in \mathbb{N}$, where each M_i is a sequential CCS process and not $\mathbf{0}$. When $n = 0$ there are no restrictions, when $m = 0$ the process overall is $\mathbf{0}$. This form is called *standard form*.

Recall that sequential CCS processes are the ones of the form $\sum_{i \in I} \alpha_i.P_i$ or $\mathsf{A}[\,\vec{a}\,]$.

Addendum to CCS Reaction Rules Process definitions lead to transitions by using the relation $\equiv_\Delta$ instead of $\equiv$ in the STRUCT rule, where $\equiv_\Delta$ is structural congruence extended with the law

$$\mathsf{A}[\,\vec{a}\,] \equiv_\Delta Q[\,\vec{a}/\vec{x}\,] \quad \text{if } \mathsf{A}[\,\vec{x}\,] := Q \in \Delta.$$

Problem 3: The Diabolic Claw Machine

To use the Diabolic Claw Machine, one has to feed it a coin first. After this is done, the user can repeatedly move the claw left or right into overall 3 positions. Initially, the claw is in middle position. At any point, the user may press a button which will make the machine try to grab a toy with the claw (this may not succeed). If successful, the machine puts the toy into its output tray. In any case, the machine then moves the claw back to the initial position and is ready for a new coin. Note that the user is not required to remove the toy from the output tray to start a new round, i.e. toys can stack in the tray.

a) Model the Diabolic Claw Machine in CCS.
   Use the names $coin, left, right, grab, toy$ for the corresponding actions.
b) Draw the corresponding LTS.
c) Assume there is no button to be pressed; instead, the machine decides when to grab. Change your process definitions to model this.

Problem 4: Santa's Coming to CCS Town

Remembering last year's chaos, it became clear to Santa that he should formalize the schedule for Christmas in CCS:

$$\begin{aligned}\mathsf{Santa} &:= \mathit{letter}.\big(\boldsymbol{\nu} g.\boldsymbol{\nu} b.\boldsymbol{\nu} p.\boldsymbol{\nu} c.(\mathsf{Kid}[\,g,b,p,c\,] \parallel \mathsf{Elf}_0[\,g,b,p,c\,]) \parallel \mathsf{Santa}\big) + \mathit{xmas}.\mathsf{Frenzy}\\ \mathsf{Frenzy} &:= \overline{\mathit{ohohoh}}.\mathsf{Frenzy} + \mathit{1jan}.\mathsf{Santa}\\ \mathsf{World} &:= \overline{\mathit{letter}}.\mathsf{World} + \overline{\mathit{xmas}}.\boldsymbol{\tau}.\overline{\mathit{1jan}}.\mathsf{World}\end{aligned}$$

Santa assigns a supervisor Elf to a Kid every time he receives a letter of wishes. When Christmas starts ($\overline{\mathit{xmas}}$) Santa drives into a frenzy of "ohohoh"s, which signal to the Elfs that they can stop supervising and they can deliver presents to good kids (the ones that did at most one bad action) or coal to bad ones. The delivery period officially ends on Jan 1st ($\overline{\mathit{1jan}}$).

$$\begin{aligned}\mathsf{Elf_i}[\,\mathit{good},\mathit{bad},\mathit{pres},\mathit{coal}\,] &:= \mathit{good}.\mathsf{Elf_i}[\,\mathit{good},\mathit{bad},\mathit{pres},\mathit{coal}\,]\\ &\quad + \mathit{bad}.\mathsf{Elf_{i+1}}[\,\mathit{good},\mathit{bad},\mathit{pres},\mathit{coal}\,]\\ &\quad + \mathit{ohohoh}.\overline{\mathit{pres}} \qquad \text{for } i = 0,1\\ \mathsf{Elf_2}[\,\mathit{good},\mathit{bad},\mathit{pres},\mathit{coal}\,] &:= \mathit{ohohoh}.\overline{\mathit{coal}}\end{aligned}$$

However, Santa was unable to model the ever unpredictable kids.

a) Help Santa by providing some model for kids $\mathsf{Kid}[\,\mathit{good},\mathit{bad},\mathit{pres},\mathit{coal}\,] := ?$.

b) Give a reaction sequence from $\mathsf{Santa} \parallel \mathsf{World}$ where one kid gets a present and one coal.

c) Can it happen that a kid who is always good receives coal? Justify.

d) Is is possible that some presents or coal remain undelivered after 1st of January?

We wish you a merry Christmas and a happy new year!