Models of Computation

Written Examination on June 27, 2013

(First part: Exercises 1 and 2, 90 minutes   Second part: Exercises 3, 4 and 5, 90 minutes)

(Previous TSD students: Exercises 1, 2 and 3, 120 minutes)

Exercise 1 (8)

Define the *small step* semantics of IMP commands as an inference system with well formed formulas of the form $\langle C, \sigma\rangle \Rightarrow \langle C', \sigma'\rangle$, where C and C' are (possibly empty) sequences $c_0; c_1; \ldots; c_n$ of commands, and with the following inference rules:

$$\frac{}{\langle skip; C, \sigma\rangle \Rightarrow \langle C, \sigma\rangle} \qquad \frac{\langle a, \sigma\rangle \rightarrow n}{\langle (x := a); C, \sigma\rangle \Rightarrow \langle C, \sigma[n/x]\rangle} \qquad \frac{}{\langle (c_0; c_1); C, \sigma\rangle \Rightarrow \langle c_0; c_1; C, \sigma\rangle}$$

$$\frac{\langle b, \sigma\rangle \rightarrow \mathbf{true}}{\langle (\mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1); C, \sigma\rangle \Rightarrow \langle c_0; C, \sigma\rangle} \qquad \frac{\langle b, \sigma\rangle \rightarrow \mathbf{false}}{\langle (\mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1); C, \sigma\rangle \Rightarrow \langle c_1; C, \sigma\rangle}$$

$$\frac{\langle b, \sigma\rangle \rightarrow \mathbf{true}}{\langle (\mathbf{while}\ b\ \mathbf{do}\ c); C, \sigma\rangle \Rightarrow \langle c; (\mathbf{while}\ b\ \mathbf{do}\ c); C, \sigma\rangle} \qquad \frac{\langle b, \sigma\rangle \rightarrow \mathbf{false}}{\langle (\mathbf{while}\ b\ \mathbf{do}\ c); C, \sigma\rangle \Rightarrow \langle C, \sigma\rangle}$$

Prove by rule induction (but only for the assignment and the complex while rules) that the new semantics is deterministic, namely $\langle C, \sigma\rangle \Rightarrow \langle C', \sigma'\rangle$ and $\langle C, \sigma\rangle \Rightarrow \langle C'', \sigma''\rangle$ implies $C' = C''$ and $\sigma' = \sigma''$. Then prove (for all rules) that $\langle c, \sigma\rangle \rightarrow \sigma'$ implies, for all command sequences C, that $\langle c; C, \sigma\rangle \Rightarrow^+ \langle C, \sigma'\rangle$, where $\langle c, \sigma\rangle \rightarrow \sigma'$ is the ordinary semantics of IMP and where $\Rightarrow^+$ is the transitive closure of $\Rightarrow$.

Exercise 2 (7)

Consider the languages of finite and infinite strings of natural numbers $L \subseteq \omega^* \cup \omega^\infty$, ordered by inclusion, with the condition that $\alpha\beta \in L$ implies $\alpha \in L$, and $\alpha n \in L \wedge m < n$ implies $\alpha m \in L$ where as usual $\alpha \in \omega^\infty$ implies $\alpha\beta = \alpha$. Prove that such languages form a complete partial ordering with $\perp$. Finally, explain why such languages actually represent finite and infinite, finite- and infinite-branching, ordered trees.

Exercise 3 (5)

Determine the type, the canonical form and the Γ such that $[\![t]\!]\rho = fix\ \Gamma$ for the HOFL agent $t = rec\ F.\lambda f.\ \lambda x.\ \mathbf{if}\ fst(x) - snd(x)\ \mathbf{then}\ x\ \mathbf{else}\ ((F\ f)\ x)$.

Exercise 4 (5)

Consider the (infinite) transition system consisting of all the pairs (n, m) of natural numbers, where $(n+1, m) \rightarrow (n, m)$ and $(n, m+1) \rightarrow (n, m)$. Consider the μ-calculus formula $\mu x.p \vee \Box x$, give its denotational semantics and compute its approximations on the transition system above assuming that p holds on $(0, n)$ and on $(n, 0)$ for all n. Finally, give the denotational semantics and the fixpoint for $\mu x.p \wedge \Box x$, $\nu x.p \wedge \diamond x$, $\nu x.p \wedge \Box x$ on the same transition system.

Exercise 5 (5)

Alice wants to throw a snowball to Bob. The actions of A are: c and the corresponding rate λ_c for centering him; and (m, λ_m) for missing him. B, instead, can hide with (h, λ_h), and be hit with (c, ∞). Define a PEPA program $A \bowtie_c B$ and draw its (labelled) CTMC. Then build a (unlabelled) DTMC by normalizing to a sum of 1 all the outgoing rates of a process. Conclude observing the probability for Alice to hit Bob.

MOD - Written Examination - June 27 2013

(1)

Exercise 1

$P(\langle c_1,\sigma\rangle \to \langle c',\sigma'\rangle) \stackrel{def}{=} \langle c',\sigma'\rangle \to \langle c'',\sigma''\rangle \Rightarrow c'=c''$ and $\sigma'=\sigma''$

rule induction on the new semantics

- $$\frac{}{\langle skip; c,\sigma\rangle \Rightarrow \langle c,\sigma\rangle}$$ $P(\langle skip; c,\sigma\rangle \Rightarrow \langle c,\sigma\rangle) \stackrel{def}{=} \langle skip; c,\sigma\rangle \Rightarrow \langle c'',\sigma''\rangle \Rightarrow c=c'' \quad \sigma=\sigma''$

  obvious, the rule applies

- $$\frac{\langle a,\sigma\rangle \to n}{\langle (x:=a); c\rangle \Rightarrow \langle c,\sigma[n/x]\rangle}$$ $P(\langle (x:=a); c,\sigma\rangle \Rightarrow \langle c,\sigma[n/x]\rangle) \stackrel{def}{=} \langle (x:=a); c,\sigma\rangle \Rightarrow \langle c'',\sigma''\rangle) \Longrightarrow c''=c \quad \sigma''=\sigma[n/x]$

  let us assume the premise

  $\langle (x:=a); c,\sigma\rangle \Rightarrow \langle c'',\sigma''\rangle$ by goal reduction we get

  $$c''=c \quad \sigma''=\sigma[n'/x] \xleftarrow{\langle a,\sigma\rangle\, n'}$$ but we know $n'=n$ for arithmetic expressions

  thus $\sigma''=\sigma[n/x]$ Q.E.D.

- $$\frac{}{\langle (c_0;c_1); c,\sigma\rangle \Rightarrow \langle c_0;c_1;c,\sigma\rangle}$$

  $P(\langle (c_0;c_1); c,\sigma\rangle \Rightarrow \langle c_0;c_1;c,\sigma\rangle) \stackrel{def}{=} \langle (c_0;c_1); c,\sigma\rangle \Rightarrow \langle c',\sigma'\rangle \Longrightarrow c'=c_0;c_1;c,\ \sigma'=\sigma$

  $$\langle (c_0;c_1); c,\sigma\rangle \Rightarrow \langle c',\sigma'\rangle \xleftarrow[\sigma'=\sigma]{c'=c_0;c_1;c} \square$$ QED

- $$\frac{\langle b,\sigma\rangle \to true}{\langle (\text{if } b \text{ then } c_0 \text{ else } c_1); c,\sigma\rangle \to \langle c_0;c,\sigma\rangle}$$

  $P(\langle (\text{if } b \text{ then } c_0 \text{ else } c_1); c,\sigma\rangle \to \langle c_0;c,\sigma\rangle) \stackrel{def}{=}$

  $\langle (\text{if } b \text{ then } c_0 \text{ else } c_1); c,\sigma\rangle \to \langle c',\sigma'\rangle \Rightarrow c_0;c=c',\ \sigma=\sigma'$

  $$\langle (\text{if } b \text{ then } c_0 \text{ else } c_1); c,\sigma\rangle \to \langle c',\sigma'\rangle \xleftarrow[\sigma=\sigma']{c'=c_0;c} \langle b,\sigma\rangle \to true$$

  which is true by the premise

- $$\frac{\langle b,\sigma\rangle \to true}{\langle (\text{while } b \text{ do } c); C, \sigma\rangle \to \langle c; (\text{while } b \text{ do } c); C, \sigma\rangle}$$

$P(\langle (\text{while } b \text{ do } c); C, \sigma\rangle \to \langle c; (\text{while } b \text{ do } c); C, \sigma\rangle) \stackrel{def}{=}$

$\langle (\text{while } b \text{ do } c); C, \sigma\rangle \to \langle C', \sigma'\rangle \Rightarrow C' = c; (\text{while } b \text{ do } c); C,\ \sigma' = \sigma$

$\langle (\text{while } b \text{ do } c); C, \sigma\rangle \to \langle C', \sigma'\rangle \xleftarrow[\sigma' = \sigma]{C' = c; (\text{while } b \text{ do } c); C} \langle b,\sigma\rangle \to true$

which is true by the premise

QED

- $$\frac{\langle b,\sigma\rangle \to false}{\langle (\text{while } b \text{ do } c); C, \sigma\rangle \to \langle C, \sigma\rangle}$$

$P(\langle (\text{while } be \text{ do } c); C, \sigma\rangle \to \langle C, \sigma\rangle) \stackrel{def}{=}$

$\langle (\text{while } b \text{ do } c); C, \sigma\rangle \to \langle C', \sigma'\rangle \Rightarrow C = C'\ \sigma = \sigma'$

$\langle (\text{while } b \text{ do } c); C, \sigma\rangle \to \langle C', \sigma'\rangle \xleftarrow{C' = C,\ \sigma' = \sigma} \langle b,\sigma\rangle \to false$

which is true by the premise QED

<u>Equivalence of the two semantics</u> (one direction)

$P(\langle c,\sigma\rangle \to \sigma') \stackrel{def}{=} \langle c; C, \sigma\rangle \Rightarrow \langle C, \sigma'\rangle$

rule induction on the old semantics

- $$\frac{}{\langle skip, \sigma\rangle \to \sigma}$$ $P(\langle skip, \sigma\rangle \to \sigma) \stackrel{def}{=} \langle skip; C, \sigma\rangle \Rightarrow \langle C, \sigma\rangle$

correct from the $\Rightarrow$ rule.

- $$\frac{\langle a,\sigma\rangle \to n}{\langle x := a, \sigma\rangle \to \sigma[n/x]}$$

$P(\langle x := a, \sigma\rangle \to \sigma[n/x]) \stackrel{def}{=} \langle (x := a); C, \sigma\rangle \Rightarrow \langle c, \sigma[n/x]\rangle$

correct from the $\Rightarrow$ rule with the same premise

- $$\frac{\langle c_0,\sigma\rangle \to \sigma'' \quad \langle c_1,\sigma''\rangle \to \sigma'}{\langle c_0; c_1,\sigma\rangle \to \sigma'}$$

Induction properties

$P(\langle c_0,\sigma\rangle \to \sigma'') \stackrel{def}{=} \langle c_0; C',\sigma\rangle \Rightarrow^{+} \langle C',\sigma''\rangle$

$P(\langle c_1,\sigma''\rangle \to \sigma') \stackrel{def}{=} \langle c_1; C'',\sigma''\rangle \Rightarrow^{+} \langle C'',\sigma'\rangle$

To prove

$P(\langle c_0; c_1,\sigma\rangle \to \sigma') \stackrel{def}{=} \langle c_0; c_1; C,\sigma\rangle \Rightarrow^{+} \langle C,\sigma'\rangle$

For every C, we assume $C' = c_1; C$ and $C'' = C$. We get

$\langle c_0; c_1; C,\sigma\rangle \Rightarrow^{*} \langle c_1; C,\sigma''\rangle \Rightarrow^{+} \langle C,\sigma'\rangle$ QED.

- $$\frac{\langle b,\sigma\rangle \to true \quad \langle c_0,\sigma\rangle \to \sigma'}{\langle \text{if } b \text{ then } c_0 \text{ else } c_1,\sigma\rangle \to \sigma'}$$

Induction property

$P(\langle c_0,\sigma\rangle \to \sigma') \stackrel{def}{=} \langle c_0; C',\sigma\rangle \Rightarrow^{+} \langle C',\sigma'\rangle$

To prove

$P(\langle \text{if } b \text{ then } c_0 \text{ else } c_1,\sigma\rangle \to \sigma') \stackrel{def}{=} \langle (\text{if } b \text{ then } c_0 \text{ else } c_1); C,\sigma\rangle \to \langle C,\sigma'\rangle$

$\langle (\text{if } b \text{ then } c_0 \text{ else } c_1); C,\sigma\rangle \to \langle c_0; C,\sigma\rangle \Rightarrow^{+} \langle C,\sigma'\rangle$ QED.

same premise, with $C' = C$

- Similarly for the other conditional rule

$$\circ\ \frac{\langle b,\sigma\rangle \to \mathrm{true} \quad \langle c,\sigma\rangle \to \sigma'' \quad \langle \mathrm{while}\ b\ \mathrm{do}\ c, \sigma''\rangle \to \sigma'}{\langle \mathrm{while}\ b\ \mathrm{do}\ c, \sigma\rangle \to \sigma'}$$

Induction properties:

$$P(\langle c,\sigma\rangle \to \sigma'') \stackrel{\mathrm{def}}{=} \langle c; c', \sigma\rangle \to^{+} \langle c', \sigma''\rangle$$

$$P(\langle \mathrm{while}\ b\ \mathrm{do}\ c, \sigma''\rangle \to \sigma') \stackrel{\mathrm{def}}{=} \langle (\mathrm{while}\ b\ \mathrm{do}\ c); c'', \sigma''\rangle \to^{+} \langle c'', \sigma'\rangle$$

To prove

$$P(\langle \mathrm{while}\ b\ \mathrm{do}\ c, \sigma\rangle \to \sigma') \stackrel{\mathrm{def}}{=} \langle (\mathrm{while}\ b\ \mathrm{do}\ c); c, \sigma\rangle \to^{+} \langle c, \sigma'\rangle$$

$$\langle (\mathrm{while}\ b\ \mathrm{do}\ c); c, \sigma\rangle \to \langle c; (\mathrm{while}\ b\ \mathrm{do}\ c); c, \sigma\rangle$$

$$\to^{+} \langle (\mathrm{while}\ b\ \mathrm{do}\ c); c, \sigma''\rangle \to^{+} \langle c, \sigma\rangle$$

same premise, $c' = (\mathrm{while}\ b\ \mathrm{do}\ c); c \qquad c'' = c$

QED

$$\circ\ \frac{\langle b,\sigma \to \mathrm{false}\rangle}{\langle \mathrm{while}\ b\ \mathrm{do}\ c, \sigma\rangle \to \sigma}$$

$$P(\langle \mathrm{while}\ b\ \mathrm{do}\ c, \sigma\rangle \to \sigma) \stackrel{\mathrm{def}}{=} \langle (\mathrm{while}\ b\ \mathrm{do}\ c); c, \sigma\rangle \to \langle c, \sigma\rangle$$

immediate from the rule, same premise.

QED.

Exercise 2

Being the structure the class of all the subsets of a given universe, $\omega^* \cup \omega^\infty$, with restrictions represented by universally quantified axioms $\forall \alpha, \beta . \alpha\beta \in L \Rightarrow \alpha \in L$, $\forall \alpha, n . \alpha n \in L \Rightarrow \alpha m \in L$, $m < n$, ordered by inclusion, the structure is a partial ordering.
The empty set satisfies the axioms, and thus it is the $\perp$.
For the completeness, given a chain

$L_1 \subseteq L_2 \subseteq \ldots$ the limit $\bigcup_{i \in \omega} L_i = L$ exists as a set,

but we must prove that it satisfies the axioms.

$\alpha\beta \in \bigcup L_i \overset{?}{\Rightarrow} \alpha \in \bigcup L_i$

$\alpha\beta \in \bigcup L_i$ implies $\exists K . \alpha\beta \in L_K$, thus $\alpha \in L_K$, $\alpha \in \bigcup L_i$.
Similarly, $\alpha n \in \bigcup L_i$, $m < n \Rightarrow \alpha m \in \bigcup L_i$.
In fact, $\alpha n \in \bigcup L_i$ implies $\exists K . \alpha n \in L_K$, $\alpha m \in L_K$ $\alpha m \in \bigcup L_i$

The strings of a language L are the paths of a possibly infinite, infinite branching tree.
A finite path α represents a node of the tree.
Its immediate descendents are all the strings of the form $\alpha 0, \alpha 1, \ldots, \alpha n, \ldots$ ordered according to n.
This set can be empty, but then no $\alpha\beta$ exists in L.
The empty string ε is the root.

Exercise 3

$$t = \mathrm{rec}\ F.\ \lambda f.\ \lambda x\ \mathrm{if}\ \underbrace{\mathrm{fst}(\underbrace{x}_{int*int})}_{int} - \underbrace{\mathrm{snd}(\underbrace{x}_{int*int})}_{int}\ \mathrm{then}\ \underbrace{x}_{int*int}\ \mathrm{else}\ \underbrace{((\underbrace{F}_{\tau}\underbrace{f}_{int*int})\underbrace{x}_{int*int})}_{int*int}$$

$int*int$ $\quad$ $int*int \to int*int$

τ $\quad$ $int*int \to int*int$

$\tau \to int*int \to int*int$ $\quad$ $\tau \to int*int \to int*int$ $\quad$ $int*int \to int*int$

$$t \to c \Leftarrow \lambda f.\lambda x\ \mathrm{if}\ \mathrm{fst}(x) - \mathrm{snd}(x)\ \mathrm{then}\ x\ \mathrm{else}\ ((t f) x) \to c$$

$$c = \lambda f.\lambda x\ \mathrm{if}\ \mathrm{fst}(x) - \mathrm{snd}(x)\ \mathrm{then}\ x\ \mathrm{else}\ ((t f) x) \quad \square$$

$$[\![t]\!]\rho = \mathrm{fix}\ \Gamma$$

$$\Gamma = \lambda d_F. \lfloor \lambda d_f. \lfloor \lambda d_x\ \mathrm{Cond}(y, d_x, \mathrm{let}\ \varphi \Leftarrow z.\ \varphi d_x) \rfloor \rfloor$$

$$y = \mathrm{let}\ v \Leftarrow d_x.\ (\pi_1 v - \pi_2 v)$$

$$z = \mathrm{let}\ \psi \Leftarrow d_F.\ \psi d_f$$

Exercice 4

$[\![\mu x. p \vee \Box x]\!] \rho = fix\ \lambda S.\ \rho P \cup \{\sigma \mid \forall \sigma' : \sigma \to \sigma' : \sigma' \in S\}$

$P = \{(0,n), (n,0) \mid n \in \omega\}$

$S_0 = \phi$

$S_1 = P$

$S_2 = S_1 \cup \{(1,1)\}$

$S_3 = S_2 \cup \{(2,1), (1,2)\}$

1 1 1 1 1 1
1 2 3 4
1 3 4
1 4
1
1

$S_K = \{(n,m) \mid n+m = k\}$

$\bigcup_{k \in \omega} S_K = \{(n,m) \mid n,m \in \omega\} = V$

$[\![\nu x. p \wedge \Box x]\!] \rho = fix\ \lambda S.\ \rho P \cap \{\sigma \mid \forall \sigma' : \sigma \to \sigma' : \sigma' \in S\}$

$S_0 = \phi$

$S_1 = \{(0,0)\}$

$S_2 = S_1 \cup \{(0,1), (1,0)\}$

$\bigcup S_K = P$

$[\![\nu x.\, p \wedge \Diamond x]\!]\rho = \mathrm{Fix}\, \lambda S.\, \rho P \cap \{\sigma \mid \exists \sigma' : \sigma \to \sigma'.\ \sigma' \in S\}$

$S_0 = V$

$S_1 = P - \{(0,0)\}$

$S_2 = P - \{(0,1),(1,0)\}$

$\bigcup_{i \in \omega} S_i = \phi$

1	2	3	4	5
2	1	1	1	1
3	1	1	1	1
4	1	1	1	1
5	1	1	1	1

$[\![\nu x.\, p \wedge \Box x]\!]\rho = \mathrm{Fix}\, \lambda S.\, \rho P \cap \{\sigma \mid \forall \sigma' : \sigma \to \sigma'.\ \sigma' \in S\}$

$S_0 = V$

$S_1 = P$

$S_2 = P = fix$

This is the unique fixpoint of $\lambda S.\, \rho P \cap \{\sigma \mid \forall \sigma' : \sigma \to \sigma'.\ \sigma' \in S\}$

Exercise 5

$$A = (c, \lambda_c).STOP + (m, \lambda_m).STOP$$
$$B = (c, \omega).STOP + (h, \lambda_h).STOP$$

$$S = A \bowtie_c B$$

$$\begin{tikzcd}
S \arrow[r, "{(c,\lambda_c)}"] \arrow[d, "{(h,\lambda_h)}"'] \arrow[dr, "{(m,\lambda_m)}"] & STOP \bowtie_c STOP & \\
A \bowtie_c STOP \arrow[d, "{(m,\lambda_m)}"'] & STOP \bowtie_c B \arrow[r, "{(h,\lambda_h)}"'] & STOP \bowtie_c STOP \\
STOP \bowtie_c STOP & &
\end{tikzcd}$$

$$\begin{tikzcd}
S \arrow[r, "\frac{\lambda_c}{\lambda_c+\lambda_m+\lambda_h}"] \arrow[d, "\frac{\lambda_h}{\lambda_c+\lambda_m+\lambda_h}"'] \arrow[dr, "\frac{\lambda_m}{\lambda_c+\lambda_m+\lambda_h}"] & STOP \bowtie_c STOP & \\
\bullet \arrow[d, "1"'] & STOP \bowtie B \arrow[r, "1"'] & STOP \bowtie_c STOP \\
STOP \bowtie_c STOP & &
\end{tikzcd}$$

probability : $\dfrac{\lambda_c}{\lambda_c+\lambda_m+\lambda_h}$