Principles of software composition 2018/19

Mid-term exam – June 5, 2019

[**Ex. 1**] Consider the HOFL term

$$t \stackrel{\text{def}}{=} \textbf{rec } f.\ \lambda x.\ \textbf{if } x \textbf{ then } 0 \textbf{ else } fx + fx \ :\ int \to int$$

1. Compute the canonical form of the term $t\ 1$, if any.
2. Compute the denotational semantics of t.

[**Ex. 2**] Consider the (recursive) CCS process definitions

$$\begin{array}{lcllcl}
A & \stackrel{\text{def}}{=} & \alpha.A + \gamma.B & \qquad B & \stackrel{\text{def}}{=} & \gamma.A + \beta.B \\
C & \stackrel{\text{def}}{=} & \overline{\gamma}.C + \overline{\alpha}.\textbf{nil} & & & \\
P & \stackrel{\text{def}}{=} & (A|C)\backslash\gamma\backslash\alpha & \qquad Q & \stackrel{\text{def}}{=} & \tau.\beta.Q + \tau.\textbf{nil}
\end{array}$$

1. Draw the LTSs of processes P and Q.
2. Show that P and Q are not strong bisimilar.
3. Show that Alice has a winning strategy for the weak bisimulation game on P and Q.

[**Ex. 3**] Write the following Google Go functions (and comment the code):

1. *filter* that takes an integer `n` and two channels `in` and `out` and forwards on `out` all the integers received from `in` that are different from `n`. It closes channel `out` when channel `in` is closed.
2. *nodup* that takes two channels `in` and `out` and forwards to `out` all integers received from `in` without sending duplicates. It closes channel `out` when all values have been forwarded.
   Hint: install a new *filter* before *nodup* for each value sent on `out`.
3. *main*, to test *nodup*.

[**Ex. 4**] Consider the atomic propositions p and q and the linear structure S such that

$$S(p) = \{n \in \mathbb{N} \mid n \text{ is odd}\} \qquad S(q) = \{n \in \mathbb{N} \mid n \text{ can be divided by 2 or by 3}\}$$

Which of the following hold? (Explain)

1. $S \models \mathsf{O}\ p$
2. $S \models \mathsf{G}\ \mathsf{F}\ p$
3. $S \models \mathsf{F}\ \mathsf{G}\ p$
4. $S \models \mathsf{F}\ (q\ \mathsf{U}\ p)$
5. $S \models \mathsf{G}\ (q\ \mathsf{U}\ \mathsf{O}p)$