Problems

These problems will be discussed at seminar 6. Programs in Haskell are not required to compile. The exercises in Nielson have been renumbered. My references are to the 1999 edition.

---

1 Consider a statement with denotational semantics

$$\mathcal{S}_{ds}[\![\texttt{try S establish b}]\!]\sigma = \begin{cases} \mathcal{S}_{ds}[\![S]\!]\sigma, & \texttt{if } \mathcal{B}[\![\texttt{b}]\!](\mathcal{S}_{ds}[\![S]\!]\sigma) = \texttt{tt} \\ s, & \texttt{if } \mathcal{B}[\![\texttt{b}]\!](\mathcal{S}_{ds}[\![S]\!]\sigma) = \texttt{ff} \end{cases} \tag{1}$$

Explain informally what the statement does.

2 Add a "swap" statement, $x_0 :=: x_1$, to While. The execution of the statement should exchange the values of the variables. Define the direct style denotational semantics for this statement.

3 In most programming languages the evaluation of an expression may change the state. As an example `n++` in Java is an expression that returns the initial value of `n` and then increments the value of `n`. Define a denotational semantics for Aexp extended with x`++`. The semantic function must now return a pair with the value and the new state.

4 (4.50) Show that $\mathcal{S}_{ds}[\![\texttt{while true do skip}]\!]$ is the totally undefined function by computing FIX F, where F is taken from table 4.1 or $F\,g \stackrel{\Delta}{=} \lambda\sigma\,.\,(\mathcal{B}[\![\texttt{b}]\!]\sigma\ ?\ (g \circ \mathcal{S}_{ds}[\![S]\!])\sigma\ :\ \sigma)$. It suffices to compute the first two values of $F^n\ \bot$.

5 Extend **While** with a `halt` statement that terminates the execution of the program in the current state. Define $\mathcal{S}_{cs}[\![\texttt{halt}]\!]$ using continuation style semantics.

6 (4.74) Assume that $\mathcal{S}'_{cs}[\![S_i]\!]c\ \sigma = c(\mathcal{S}_{ds}[\![S_i]\!]\sigma)$ for $i = 1, 2$. Show that $\mathcal{S}'_{cs}[\![S]\!]c\ \sigma = c(\mathcal{S}_{ds}[\![S]\!]\sigma)$ when $S =$ `if b then` S_1 `else` S_2.