Assignments on Program Semantics

Theory of Programming Languages
Master's Degree in Formal Methods in Computer Science
Year 2021–2022

Submission deadlines:

- October 20th: Exercise 1.
- October 27th: Exercises 2 and 3.
- November 4th: Exercise 4.

Submission instructions: For each submission, students are required to upload a PDF file with the solution(s) to the exercise(s). You can use LaTeX (recommended) or any other word processor/typesetter. Handwritten submissions are also accepted but, in the latter case, the student has to scan their submission in order to obtain a PDF file. Submissions have to be written in English.

1. Let us assume $e, e' \in \mathbf{AExp}$ and $x \in \mathbf{Var}$. The notation $e[x/e']$ denotes the result of replacing all occurrences of x in e by e'. For example: $(x+y)[x/(3*z)] = (3*z)+y$.

   (a) Define $e[x/e']$ in a compositional way.

   (b) Prove the following *substitution lemma*: for all $e, e' \in \mathbf{AExp}$, $x \in \mathbf{Var}$, $\sigma \in \mathbf{State}$:

   $$\mathcal{A}[\![e[x/e']]\!]\,\sigma = \mathcal{A}[\![e]\!]\,\sigma[x \mapsto \mathcal{A}[\![e']\!]\,\sigma]$$

2. Assume we extend the syntax of *While* statements with a new construct: `repeat` S `until` b. This statement is executed as follows:

   (1) Execute S.

   (2) Check whether b is false. In this case, step back to (1). Otherwise, finish.

   Define the big-step and small-step semantic rules for this new construct. You cannot rely on the rules of `while` to define the rules of `repeat`. Finally, prove that `repeat` S `until` b is equivalent to (S; `while` $\neg b$ `do` S)

3. Add the following iterative construct to *While*: `for` $x := e_1$ `to` e_2 `do` S. Define its big-step and small-step semantic rules. You cannot rely on the `while` or `repeat` construct to do this exercise.

4. Given the function $F : (\mathbf{State} \to \mathbf{State}_\bot) \to (\mathbf{State} \to \mathbf{State}_\bot)$ defined as follows:

   $$F(f) = \mathit{cond}(\mathcal{B}[\![n > 0]\!], f \circ \mathcal{S}[\![x := 2*x;\ n := n-1]\!], \mathit{id})$$

(a) Give an explicit definition for $F(\lambda\sigma.\bot)$, $F^2(\lambda\sigma.\bot)$ and $F^3(\lambda\sigma.\bot)$.

(b) From the results above, conjecture a general definition for $F^i(\lambda\sigma.\bot)$ where $i \geq 1$. [Optional] Prove by induction on i that your conjecture is correct.

(c) Give an explicit definition for $\bigsqcup_i F^i(\lambda\sigma.\bot)$.

(d) Which is the least fixed point of F? Justify your answer.

(e) Given the above, compute the state resulting from the execution of the following program

$$x := 1; \texttt{while}\ n > 0\ \texttt{do}\ (x := 2 * x;\ n := n - 1)$$

under the initial state $\sigma = [n \mapsto 4]$.