Programmation 1
TD n°6

20 octobre 2020

\[ \begin{align*}
\rho \vdash x := e & \Rightarrow \rho[x \mapsto [e]_{\rho}] & (:=) \\
\rho \vdash \text{skip} & \Rightarrow \rho & (\text{Skip}) \\
\rho \vdash c_1 \Rightarrow \rho' & \rho' \vdash c_2 \Rightarrow \rho'' & (\text{Seq}) \\
\rho \vdash \text{if } e \text{ then } c_1 \text{ else } c_2 \Rightarrow \rho' & \text{ si } [e]_{\rho} \neq 0 & (\text{if}_1) \\
\rho \vdash \text{while } e \text{ do } c \Rightarrow \rho'' & \text{ si } [e]_{\rho} = 0 & (\text{while}_\text{fin}) \\
\rho \vdash c \Rightarrow \rho' & \rho' \vdash \text{while } e \text{ do } c \Rightarrow \rho'' & (\text{while}) \\
\text{si } [e]_{\rho} \neq 0 \\
\end{align*} \]

Figure 1 – La sémantique opérationnelle à grands pas de IMP.

Exercise 1:
Let \( \text{while } 1 \text{ do skip } \). By considering the form of derivations, explain why, for any environment \( \rho \), there is no environment \( \rho' \) such that \( \rho \vdash \text{while } 1 \text{ do skip } \Rightarrow \rho' \).

Exercise 2:
Given two programs \( c_1, c_2 \), we say that \( c_1 \) and \( c_2 \) are equivalent, denoted as \( c_1 \sim c_2 \) iff for all environments \( \rho, \rho' \), \((\rho \vdash c_1 \Rightarrow \rho') \Leftrightarrow (\rho \vdash c_2 \Rightarrow \rho') \).

1. Show that \( \text{while } e \text{ do } c \sim \text{if } e \text{ then } (c; \text{while } e \text{ do } c) \text{ else skip } \).

2. What are the programs equivalent to \( \text{while } 1 \text{ do skip } \)?

Exercise 3:
In the course, you saw a version of the IMP language without Boolean expressions. In your compiler, you had defined these expressions as :

\[ b ::= \text{True} | \text{False} | e \equiv e | e < e | \neg b | b \lor b | b \land b \]

1. Give an operational semantics for Boolean expressions.

2. Modify your rules so that they implement the “or” such that : when evaluating \( b_1 \lor b_2 \), \( b_2 \) is not evaluated if \( b_1 \) evaluates to \text{True}. Same thing for “and”.

3. Modify your rules so that they implement the parallel evaluation of the “or”. Same thing for the “and”.

4. In the three deduction systems obtained, what can be said about the number of derivations of a triple \((b, \rho) \rightarrow _?\)?
Exercise 4:

One of the objectives of the following courses will be to introduce the mathematical tools necessary to be able to define a denotational semantics of IMP: \([c]_\rho = \rho'\), which means that the execution of the program \(c\) in the environment \(\rho\) leads to the environment \(\rho'\). This «functional» writing suggests that the environment \(\rho'\) is fully determined by \(c\) and \(\rho\).

1. Show this property for the operational semantics: for all \(c, \rho, \rho_1, \rho_2\), if \(\rho \vdash c \Rightarrow \rho_1\) and \(\rho \vdash c \Rightarrow \rho_2\) then \(\rho_1 = \rho_2\).

2. We consider the non-deterministic language given by the following syntax:

\[ c ::= \text{skip} \mid x := e \mid c_1 ; c_2 \mid \text{if } e \text{ then } c \text{ else } c \mid \text{while } e \text{ do } c \mid c_1 \lor c_2 \]

where the instruction \(c_1 \lor c_2\) signifies «execute \(c_1\) or execute \(c_2\), non-deterministically».

Propose operational semantics for this language.