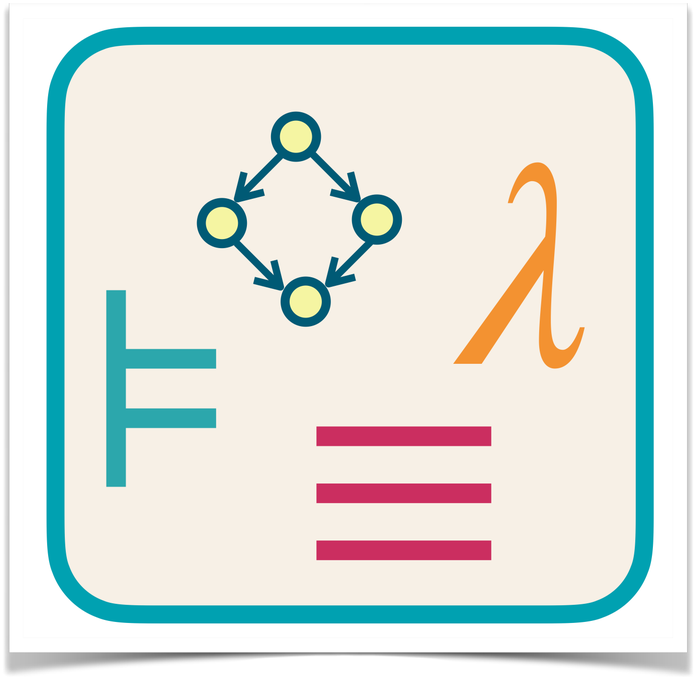


LPL 2026/27 (063AA, 9CFU)

Programming Languages

Fabio Gadducci

and Laboratory

Vincenzo Ciancia

02e - Exercises on semantics

Let w be the command:

$$w \stackrel{\text{def}}{=} \textbf{while } x \times x = y \textbf{ do } (\, x := x \times x \,;\; y := x \times y \,)$$

Find the set of memories S such that $\forall \sigma \in S.\ \langle w, \sigma \rangle \not\to$.

Suppose one wants to insert some measure of efficiency in the operational semantics of IMP.

1. Redefine the operational semantics of IMP commands in such a way that the transition predicate takes the form

$$\langle c, \sigma \rangle \xrightarrow{n} \sigma'$$

   with the meaning that "the command c, when executed in the state σ converges to the state σ' by evaluating exactly n boolean guards."

2. Prove by rule induction that for all c, σ, σ':

$$\langle c, \sigma \rangle \to \sigma' \;\Rightarrow\; \exists n \in \mathbb{N}.\, \langle c, \sigma \rangle \xrightarrow{n} \sigma'.$$

Let us consider expressions of the form $e \;:=\; n \mid e^*$ with $n \in \mathbb{N}$, whose operational semantics is defined by the rules

$$\frac{}{n \to n}(num) \qquad \frac{e \to n}{e^* \to n}(once) \qquad \frac{e^* \to n_1 \quad e^* \to n_2}{e^* \to n_1 \times n_2}(more)$$

It is evident that $e^* \to n$ implies $(e^*)^* \to n$ (by the second rule). Prove the converse by rule induction, i.e., that $\forall e, n.\ (\ (e^*)^* \to n \ \Rightarrow\ e^* \to n\ )$.

Let us extend IMP with the command

$$x := c$$

whose denotational semantics is

$$\mathscr{C}[\![x := c]\!]\sigma \stackrel{\mathsf{def}}{=} \left(\lambda\sigma_1.\ \sigma[{}^{\sigma_1(x)}/_{x}]\right)^* (\mathscr{C}[\![c]\!]\sigma)$$

1. Define the operational semantics for the new construct.

2. Extend the proofs of correctness and completeness between the operational and the denotational semantics to account for the new construct.

Suppose we replace the **while-do** command of IMP with the command **for** a **do** c whose operational semantics is defined by the rules:

$$\frac{\langle a, \sigma\rangle \to 0}{\langle \textbf{for } a \textbf{ do } c, \sigma\rangle \to \sigma} \qquad \frac{\langle a, \sigma\rangle \to n \neq 0 \quad \langle c, \sigma\rangle \to \sigma'' \quad \langle \textbf{for } a - 1 \textbf{ do } c, \sigma''\rangle \to \sigma'}{\langle \textbf{for } a \textbf{ do } c, \sigma\rangle \to \sigma'}$$

1. Show a **for-do** program that diverges.

2. Extend the proof of determinacy taking into account the new construct.