Series 1

Exercise 1

Let consider the following statement \( (z := x; x := y); y := z \), and the environment \( \sigma_0 \) which maps every variables but \( x \) and \( y \) to 0, maps \( x \) to 5, and \( y \) to 7. Give a derivation tree of this statement.

Exercise 2

We consider the arithmetical expressions defined in the lecture course. Let \( a, a' \in \text{Aexp} \), and \( \sigma, \sigma' \) two states. Let \( X \) be the set of variables appearing in \( a \).

1. Prove that if \( \forall x \in X \cdot \sigma(x) = \sigma'(x) \), then \( A[a] \sigma = A[a] \sigma' \).
2. Prove that \( A[a[a'/x]] \sigma = A[a] \sigma[x \mapsto A[a'] \sigma] \).

Exercise 3

We consider the following statements:

- while \( \neg(x = 1) \) do \( (y := y \times x; x := x - 1) \) od
- while \( 1 \leq x \) do \( (y := y \times x; x := x - 1) \) od
- while true do skip od

where \( x \) designates a variable of type \( \mathbb{Z} \).

For each of the preceding statement, determine whether:

1. its execution loops in every state
2. its execution stops in every state

Prove your answers.

Exercise 4

We wish to add the following statement to the While language:

\[
\text{repeat } S \text{ until } b
\]

1. Provide the semantics rules in order to define \( \text{repeat } S \text{ until } b \) without using the \( \text{while } b \text{ do } \cdots \text{od} \) construction.
2. Prove that
   (a) \( \text{repeat } S \text{ until } b \)
   (b) \( S \text{ if } b \text{ then skip else } (\text{repeat } S \text{ until } b) \)
   are semantically equivalent.
3. We want to prove that the statement \texttt{repeat} \( S \) \texttt{until} \( b \) does not add any expressive power to the \textbf{While} language. To do so, give a function which transforms every program with the statement \texttt{repeat} \( S \) \texttt{until} \( b \) into a program in the \textbf{While} language. Is the given transformation computable? Compare the size of a program and the size of its image by this transformation.

\textbf{Exercise 5}

Prove that, for all statements \( S_1, S_2, S_3 \), the following statements are semantically equivalent:

1. \( S_1; (S_2; S_3) \)
2. \((S_1; S_2); S_3\)

Prove that, in general, \( S_1; S_2 \) is not semantically equivalent to \( S_2; S_1 \).

\textbf{Exercise 6}

Prove that the natural operational semantics of the \textbf{While} language is deterministic.

\textbf{Exercise 7}

Give a natural operational semantics for arithmetical and boolean expressions which is equivalent to the inductive semantics given in the lecture course.