Syntax and semantics
Basic Imperative Statements and Control Structures

1 Learning Objectives

1. Evaluating variables in states.
2. Big-Step Semantics for STM; derivation trees.
3. Small-Step Semantics for STM; derivation trees.
4. Equivalence of Big-Step and Small-Step Semantics for BIMS.
5. Loop Constructs: Repeat-loops and For-loops.
6. Semantic Equivalence.
7. Abnormal termination.
8. Nondeterminism
9. Concurrency

2 Readings

Hüttel's book:
Part II – First Examples
Chapter 4. Basic Imperative Statements
Chapter 5. Control structures

3 Exercises

Exercise 1. Consider the following big-step semantics transition rule for the variable allocation.

$$(\textsc{ass})\ \langle x := a, s\rangle \to s[x \mapsto v] \qquad \text{where } s \vdash a \to_a v$$

Identify the part of the rule corresponding to the item below, underline it and label it with the number of the item it refers to.

1. the list of values for the variables before executing the command;
2. the evaluation of right hand side of the assignment;
3. the value of the right hand side of the transition;
4. the list of values for the variables after the command has been performed.

Exercise 2. Consider the following big-step semantics transition rule for the sequential composition.

$$(\textsc{comp-1})\ \frac{\langle S_1, s\rangle \Rightarrow \langle S_1', s'\rangle}{\langle S_1; S_2, s\rangle \Rightarrow \langle S_1'; S_2, s'\rangle} \qquad (\textsc{comp-2})\ \frac{\langle S_1, s\rangle \Rightarrow s'}{\langle S_1; S_2, s\rangle \Rightarrow \langle S_2, s'\rangle}$$

Identify the part of the rule corresponding to the item below, underline it and label it with the number of the item it refers to (the same item may refer to both of the rules).

1. The first command terminates after one step.
2. The first command is terminated, and the second command can be executed.
3. The first command does not terminate after one step.
4. The remaining portion of the first command.
5. The list of values for the variables after the sequential composition has been performed.

Exercise 3. Give a small-step semantics for the repeat-until command. *Hint:* It is a good idea to take inspiration from small-step rule for while-loops. But do not use while loops in the rules you propose. Think instead of what should be the first thing to happen when a repeat loop is executed. And what is the next thing to be done, then?

Exercise 4. Let $S = \texttt{i} := \texttt{1}; \texttt{while}\ (\neg \texttt{x} = \texttt{0})\ \texttt{do}\ (\texttt{i} := \texttt{i} * \texttt{x}; \texttt{x} := \texttt{x-1})$. Show the derivation for $\langle S, s\rangle \to s'$ where $s = [x \longrightarrow 3]$ and show how s' is defined. You do not have to specify the derivations trees for the **Aexp**- and **Bexp**-transitions.

Exercise 5. Find all transitions (if any) in the transition sequence starting with $\texttt{if}\ \ \texttt{x} > \texttt{3}\ \ \texttt{then}\ \ (\texttt{x} := \texttt{3+x}; \texttt{y} := \texttt{4})\ \ \texttt{else}\ \ \texttt{skip}$ from the state $s = [x \longrightarrow 4]$. Construct derivations trees for each transition. You do not need to specify derivations trees for **Aexp**- and **Bexp**-transitions.

Exercise 6. Give a small-step semantics for the repeat-until command. *Hint:* It is a good idea to take inspiration from small-step rule for while-loops. But do not use while loops in the rules you propose. Think instead of what should be the first thing to happen when a repeat loop is executed. And what is the next thing to be done, then?

Exercise 7. Some programming languages have a general loop construction.

$$\texttt{loop}\ S_1;\ \ \texttt{exit on}\ \ b;\ S_2\ \ \texttt{end}$$

The meaning of this command is that first S_1 is executed and then the condition b is evaluated. If the condition b is true, we exit the loop, otherwise, we perform S_2 and repeat the loop. Extend **Bims** with this command and give it a big-step semantics.

Exercise 8. Consider the big-step semantics for the while-loop in **Bims** described by the rules below.

$$\text{(WHILE-1)}\ \frac{\langle S, s\rangle \to s''\quad \langle \texttt{while}\ b\ \texttt{do}\ S, s''\rangle \to s'}{\langle \texttt{while}\ b\ \texttt{do}\ S, s\rangle \to s'} \qquad \text{if } s \vdash b \to_b \mathit{tt}$$

$$\text{(WHILE-2)}\ \langle \texttt{while}\ b\ \texttt{do}\ S, s\rangle \to s \quad \text{if } s \vdash b \to_b \mathit{ff}$$

Identify the part of the rule corresponding to each item below, underline it and label it with the number of the item it refers to.

1. The while-loop body is executed.
2. The evaluation of the condition is done *before* the loop body can be performed.
3. The list of the variables values after the execution of the loop body.
4. Variables values remains unchanged if the loop condition evaluates to false.
5. This rule is not compositional.
6. Here you can see that the rule is not compositional.

Exercise 9. Tick the answer you think is right.

	Yes	**No**
`skip` $\sim_{bss}$ `abort`	☐	☐
$\langle$`while 0 = 0 do skip`$, s\rangle \to s$	☐	☐
`while 0 = 0 do skip` $\sim_{bss}$ `abort`	☐	☐
Parallelism can be described with a big-step semantics	☐	☐
Nondeterminism can be described with a big-step semantics	☐	☐

Exercise 10. Is (`while 0=0 do skip`) $\sim_{bss}$ (`if 0=0 then skip else skip`)? Justify your answer by using the semantics as accurately as possible.