

Exercise 1: Operational Semantics

—Solution—

Deadline for uploading solutions via Ilias:
November 2, 2023, 11:59pm Stuttgart time

Consider the following SIMP program P:

```
y := 2; while !x > 3 do if ¬ (!x = !y) then x := !x - 2 else x := !x - 1
```

and an initial store:

$$s = \{x \mapsto 4\}$$

Your task is to evaluate this program using the abstract machine, small-step operational semantics, and big-step operational semantics, as introduced in the lecture. As a reference, see the rules and axioms provided in the appendix (copied from Fernandez' book).

To ease the presentation of your solution, please use the following abbreviations when referring to parts of the program:

Abbreviation	Code
W	<code>while !x > 3 do if ¬ (!x = !y) then x := !x - 2 else x := !x - 1</code>
C_1	<code>!x > 3</code>
F	<code>if ¬ (!x = !y) then x := !x - 2 else x := !x - 1</code>
C_2	<code>¬ (!x = !y)</code>
T	<code>x := !x - 2</code>
E	<code>x := !x - 1</code>

For each task, we provide a template to fill in your solution. The correct solutions fit into the template and should align with those parts of the solution that we provide.

Hint: Sketch your solution on scratch paper before filling it into the template.

1 Abstract Machine [30 points]

Provide the semantics of the program as a sequence of transitions of the abstract machine for SIMP. Use the following template, where each line corresponds to a new configuration.

Solution:

Program	Stack	Store
$\rightarrow P \circ nil$	nil	$\{x \mapsto 4\}$
$\rightarrow y := 2 \circ W \circ nil$	nil	$\{x \mapsto 4\}$
$\rightarrow 2 \circ := \circ W \circ nil$	$y \circ nil$	$\{x \mapsto 4\}$
$\rightarrow := \circ W \circ nil$	$2 \circ y \circ nil$	$\{x \mapsto 4\}$
$\rightarrow W \circ nil$	nil	$\{x \mapsto 4, y \mapsto 2\}$
$\rightarrow C_1 \circ while \circ nil$	$C_1 \circ F \circ nil$	$\{x \mapsto 4, y \mapsto 2\}$
$\rightarrow !x > 3 \circ > \circ while \circ nil$	$C_1 \circ F \circ nil$	$\{x \mapsto 4, y \mapsto 2\}$
$\rightarrow 3 \circ > \circ while \circ nil$	$4 \circ C_1 \circ F \circ nil$	$\{x \mapsto 4, y \mapsto 2\}$
$\rightarrow > \circ while \circ nil$	$3 \circ 4 \circ C_1 \circ F \circ nil$	$\{x \mapsto 4, y \mapsto 2\}$
$\rightarrow while \circ nil$	$true \circ C_1 \circ F \circ nil$	$\{x \mapsto 4, y \mapsto 2\}$
$\rightarrow F \circ W \circ nil$	nil	$\{x \mapsto 4, y \mapsto 2\}$
$\rightarrow C_2 \circ if \circ W \circ nil$	$T \circ E \circ nil$	$\{x \mapsto 4, y \mapsto 2\}$
Recall C_2 : $\neg (!x = !y)$		
$\rightarrow !x = !y \circ \neg \circ if \circ W \circ nil$	$T \circ E \circ nil$	$\{x \mapsto 4, y \mapsto 2\}$
$\rightarrow !x \circ !y \circ = \circ \neg \circ if \circ W \circ nil$	$T \circ E \circ nil$	$\{x \mapsto 4, y \mapsto 2\}$
$\rightarrow !y \circ = \circ \neg \circ if \circ W \circ nil$	$4 \circ T \circ E \circ nil$	$\{x \mapsto 4, y \mapsto 2\}$
$\rightarrow = \circ \neg \circ if \circ W \circ nil$	$2 \circ 4 \circ T \circ E \circ nil$	$\{x \mapsto 4, y \mapsto 2\}$
$\rightarrow \neg \circ if \circ W \circ nil$	$false \circ T \circ E \circ nil$	$\{x \mapsto 4, y \mapsto 2\}$
$\rightarrow if \circ W \circ nil$	$true \circ T \circ E \circ nil$	$\{x \mapsto 4, y \mapsto 2\}$
$\rightarrow T \circ W \circ nil$	nil	$\{x \mapsto 4, y \mapsto 2\}$
Recall T : $x := !x - 2$		
$\rightarrow !x - 2 \circ := \circ W \circ nil$	$x \circ nil$	$\{x \mapsto 4, y \mapsto 2\}$
$\rightarrow !x \circ 2 \circ - \circ := \circ W \circ nil$	$x \circ nil$	$\{x \mapsto 4, y \mapsto 2\}$
$\rightarrow 2 \circ - \circ := \circ W \circ nil$	$4 \circ x \circ nil$	$\{x \mapsto 4, y \mapsto 2\}$
$\rightarrow - \circ := \circ W \circ nil$	$2 \circ 4 \circ x \circ nil$	$\{x \mapsto 4, y \mapsto 2\}$
$\rightarrow := \circ W \circ nil$	$2 \circ x \circ nil$	$\{x \mapsto 4, y \mapsto 2\}$
$\rightarrow W \circ nil$	nil	$\{x \mapsto 2, y \mapsto 2\}$
$\rightarrow C_1 \circ while \circ nil$	$C_1 \circ F \circ nil$	$\{x \mapsto 2, y \mapsto 2\}$
Recall C_1 : $!x > 3$		
$\rightarrow !x > 3 \circ > \circ while \circ nil$	$C_1 \circ F \circ nil$	$\{x \mapsto 2, y \mapsto 2\}$
$\rightarrow 3 \circ > \circ while \circ nil$	$2 \circ C_1 \circ F \circ nil$	$\{x \mapsto 2, y \mapsto 2\}$
$\rightarrow > \circ while \circ nil$	$3 \circ 2 \circ C_1 \circ F \circ nil$	$\{x \mapsto 2, y \mapsto 2\}$
$\rightarrow while \circ nil$	$false \circ C_1 \circ F \circ nil$	$\{x \mapsto 2, y \mapsto 2\}$
$\rightarrow nil$	nil	$\{x \mapsto 2, y \mapsto 2\}$

2 Small-Step Semantics [40 points]

Provide the semantics of the program as an evaluation sequence using the small-step semantics of SIMP. Use the following template to provide your solution. For each transition, indicate above the arrow why you can take the transition. Specifically:

- On top of each transition, indicate the rule or axiom that you use to take the transition. Note that you have to indicate the name of the reduction axiom or rule at the bottom of the proof tree. For example, if you resolve a variable in a statement, which is one of two statements in a sequence, you have to indicate the sequence rule, not the variable rule.
- We do not ask you to provide the proof trees for all transitions, but please provide the proof tree of the first transition enabled the *if* reduction rule. Use the template at the end of the question to provide this proof tree. We provide the proof tree for the first transition enabled by the *seq* reduction rule as an example. It is strongly recommended to at least sketch the proof trees of all transitions for yourself, because it helps understand what transition steps you can(not) take.

Hint 1: The transitions used in the correct solution should contain 6 uses of axioms and 13 uses of rules.

Hint 2: Each transition is enabled either by a rule or an axiom. In case of an axiom, the proof tree is flat, i.e., without anything on top of the horizontal line. In case of a rule, the proof tree has at least one horizontal line with a hypothesis on top (see the example of the first usage of *seq* below).

Solution:

$$\begin{aligned} & \langle P, \{x \mapsto 4\} \rangle \\ & \xrightarrow{(seq)} \langle skip ; W, \{x \mapsto 4, y \mapsto 2\} \rangle \\ & \xrightarrow{(skip)} \langle W, \{x \mapsto 4, y \mapsto 2\} \rangle \\ & \xrightarrow{(while)} \langle \text{if } C_1 \text{ then } (F; \text{while } C_1 \text{ do } F) \text{ else } skip, \{x \mapsto 4, y \mapsto 2\} \rangle \\ & \xrightarrow{(if)} \langle \text{if } 4 > 3 \text{ then } (F; \text{while } C_1 \text{ do } F) \text{ else } skip, \{x \mapsto 4, y \mapsto 2\} \rangle \\ & \xrightarrow{(if)} \langle \text{if } true \text{ then } (F; \text{while } C_1 \text{ do } F) \text{ else } skip, \{x \mapsto 4, y \mapsto 2\} \rangle \\ & \xrightarrow{(ifT)} \langle F; \text{while } C_1 \text{ do } F, \{x \mapsto 4, y \mapsto 2\} \rangle \\ & \text{Recall } F \text{ represents } \text{if } \neg (!x = !y) \text{ then } T \text{ else } E \\ & \xrightarrow{(seq)} \langle \text{if } \neg (4 = !y) \text{ then } T \text{ else } E; \text{while } C_1 \text{ do } F, \{x \mapsto 4, y \mapsto 2\} \rangle \\ & \xrightarrow{(seq)} \langle \text{if } \neg (4 = 2) \text{ then } T \text{ else } E; \text{while } C_1 \text{ do } F, \{x \mapsto 4, y \mapsto 2\} \rangle \\ & \xrightarrow{(seq)} \langle \text{if } \neg false \text{ then } T \text{ else } E; \text{while } C_1 \text{ do } F, \{x \mapsto 4, y \mapsto 2\} \rangle \\ & \xrightarrow{(seq)} \langle \text{if } true \text{ then } T \text{ else } E; \text{while } C_1 \text{ do } F, \{x \mapsto 4, y \mapsto 2\} \rangle \\ & \xrightarrow{(seq)} \langle T; \text{while } C_1 \text{ do } F, \{x \mapsto 4, y \mapsto 2\} \rangle \\ & \text{Recall } T \text{ represents } x := !x - 2 \\ & \xrightarrow{(seq)} \langle x := 4 - 2; \text{while } C_1 \text{ do } F, \{x \mapsto 4, y \mapsto 2\} \rangle \\ & \xrightarrow{(seq)} \langle x := 2; \text{while } C_1 \text{ do } F, \{x \mapsto 4, y \mapsto 2\} \rangle \\ & \xrightarrow{(seq)} \langle skip ; \text{while } C_1 \text{ do } F, \{x \mapsto 2, y \mapsto 2\} \rangle \\ & \xrightarrow{(skip)} \langle \text{while } C_1 \text{ do } F, \{x \mapsto 2, y \mapsto 2\} \rangle \\ & \xrightarrow{(while)} \langle \text{if } C_1 \text{ then } (F; \text{while } C_1 \text{ do } F) \text{ else } skip, \{x \mapsto 2, y \mapsto 2\} \rangle \\ & \text{Recall } C_1 \text{ represents } !x > 3 \\ & \xrightarrow{(if)} \langle \text{if } 2 > 3 \text{ then } (F; \text{while } C_1 \text{ do } F) \text{ else } skip, \{x \mapsto 2, y \mapsto 2\} \rangle \end{aligned}$$

$$\frac{(if)}{\rightarrow} \langle \text{if } false \text{ then } (F; \text{while } C_1 \text{ do } F) \text{ else } skip, \{x \mapsto 2, y \mapsto 2\} \rangle$$

$$\frac{(if_F)}{\rightarrow} \langle skip, \{x \mapsto 2, y \mapsto 2\} \rangle$$

Proof trees (s stands for $\{x \mapsto 4\}$ and s' stands for $\{x \mapsto 4, y \mapsto 2\}$):

- Proof tree for first use of *seq* rule:

$$\frac{\frac{}{\langle y := 2, s \rangle \rightarrow \langle skip, s' \rangle} (:=)}{\langle y := 2; W, s \rangle \rightarrow \langle skip; W, s' \rangle} (seq)$$

- Proof tree for first use of *if* rule:

$$\frac{\frac{\frac{}{\langle !x, s' \rangle \rightarrow \langle 4, s' \rangle} (var)}{\langle !x > 3, s' \rangle \rightarrow \langle 4 > 3, s' \rangle} (bop_L)}{\langle \text{if } !x > 3 \text{ then } (F; W) \text{ else } skip, s' \rangle \rightarrow \langle \text{if } 4 > 3 \text{ then } (F; W) \text{ else } skip, s' \rangle} (if)$$

Subtree t_2 :

$$\begin{array}{c}
 \frac{}{\langle !x, s' \rangle \Downarrow \langle 4, s' \rangle} \text{ (var)} \quad \frac{}{\langle !y, s' \rangle \Downarrow \langle 2, s' \rangle} \text{ (var)} \quad \frac{}{\langle !x, s' \rangle \Downarrow \langle 4, s' \rangle} \text{ (var)} \quad \frac{}{\langle 2, s' \rangle \Downarrow \langle 2, s' \rangle} \text{ (const)} \\
 \hline
 \frac{}{\langle !x = !y, s' \rangle \Downarrow \langle \text{false}, s' \rangle} \text{ (not)} \quad \frac{}{\langle !x - 2, s' \rangle \Downarrow \langle 2, s' \rangle} \text{ (:=)} \\
 \frac{}{\langle C_2, s' \rangle \Downarrow \langle \text{true}, s' \rangle} \quad \frac{}{\langle T, s' \rangle \Downarrow \langle \text{skip}, s'' \rangle} \text{ (if}_T\text{)} \\
 \hline
 \langle F, s' \rangle \Downarrow \langle \text{skip}, s'' \rangle
 \end{array}$$