

Formal Semantics of Programming Languages Exercise 2 (June 9)

Wolfgang Schreiner
Wolfgang.Schreiner@risc.jku.at

May 18, 2017

The exercise is to be submitted by the deadline stated above as a report with a decent cover page (title of the course, your name, Matrikelnummer, email address) in one of the following forms:

1. either as a single PDF file uploaded in Moodle (no emails, please), or
2. as a stapled paper report handed out to me (in class or in my mailbox).

Exercise 2: Expressions with Side effects

Take the following language of programs P , declarations d , commands C , expressions E , numerals N , and identifiers I :

$$\begin{aligned} P &::= D; C \\ D &::= _ \mid \mathbf{var} \ I; D \\ C &::= I := E \mid C_1; C_2 \mid \mathbf{if} \ E_1 = E_2 \ \mathbf{then} \ C \mid \mathbf{if} \ E_1 = E_2 \ \mathbf{then} \ C_1 \ \mathbf{else} \ C_2 \\ E &::= I \mid N \mid E_1 + E_2 \mid \mathbf{exec} \ C \ \mathbf{result} \ E \end{aligned}$$

The expression “**exec** C **result** E ” executes C and then returns the result of the evaluation of E . Correspondingly, the evaluation of an expression may alter the store.

1. Define a denotational semantics for this language. In this semantics, a declaration D introduces a set of identifiers as an *environment*; the result is an error declaration, if the same variable is declared twice.

A command valuation $\mathbf{C}[[C]](e)(s)$ describes the result of executing command C in environment e and store s ; the result is an error store, if a variable not declared in e was accessed (read or written).

2. Define a corresponding big-step operational semantics for this language. Use the same notions of environment and store that you also used in the denotational semantics. In particular, it shall be possible to derive a judgement for an erroneous program with a result configuration that indicates that an error occurred.
3. Formulate for each domain P, D, C, E the statement “the operational semantics of the domain is equivalent to its denotational semantics”.
4. Prove the equivalence statement for the expression **exec** C **result** E (in this proof you can assume that the equivalence statement holds for E and C).