Formal Semantics of Programming Languages

Exercise 1 (April 28, 2017)

Wolfgang Schreiner
Wolfgang.Schreiner@risc.jku.at

March 29, 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).

In any case, please register as a user in the Moodle course such that I can issue grades there.

Exercise 1: Expressions with Side Effects

Take the following language of evaluation sessions S, expressions E, numerals N, and identifiers I:

$$S ::= E$$
$$E ::= I \mid N \mid E_1 + E_2 \mid I := E \mid E_1; E_2$$

This language contains an expression $I := E$ which returns the value of E and updates as a side-effect the content of variable I with that value. The expression $E_1; E_2$ evaluates first E_1, then E_2 and returns the value of E_2.

An example expression in this language is

```
X:=5; X+(Y:=X); (Z:=Y;Z+1)+Z
```

whose value is 11.

1. Complete the abstract syntax of this language (introduce syntactic domains).
2. Give this language a denotational semantics (semantic algebras, for each syntactic domain the signature and definition of a valuation function).
3. Give this language a big-step operational semantics (for each syntactic domain a judgment and the rules for deriving the judgment).
4. Formulate for each syntactic domain the statement of the equivalence of denotational and operational semantics (you need not prove the statements).

Please note that the evaluation of every expression updates the store and returns the value. An evaluation session is started with a store that maps all identifiers to zero. The result of an evaluation session is a value. In the denotational semantics, we thus want a valuation function $\mathbf{S}$ such that $\mathbf{S}[\![S]\!]$ is a natural number.