

Formal Semantics of Programming Languages Exercise 4 (June 28)

Wolfgang Schreiner
Wolfgang.Schreiner@risc.jku.at

June 3, 2013

The exercise is to be submitted by the denoted deadlines 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 4: Procedures

Augment the language of Figure 7.2 to include procedures with a single parameter:

$$D ::= \dots \mid \mathbf{proc} I_1(\mathbf{var} I_2) = C.$$
$$C ::= \dots \mid \mathbf{call} I_1(E).$$

On invocation of procedure I_1 , the value of argument E is assigned to the formal parameter I_2 (but note that after return from the procedure call the original value of I_2 is restored).

Give the semantic equations for these constructs under each of the following assumptions:

1. The domain *Denotable-value* is augmented with the summand

$$Proc1 = Store \rightarrow Denotable-value \rightarrow Poststore_{\perp}$$

to accommodate procedures.

2. The domain *Denotable-value* is augmented with the summand

$$Proc2 = Environment \rightarrow Store \rightarrow Denotable-value \rightarrow Poststore_{\perp}$$

to accommodate procedures.

What kind of scoping is used in each case?

Sketch for each of case the evaluation of the semantics function corresponding to the program

```
begin
  var Y; // first Y
  proc P(var X) = (X := X+1; Y := X+Y);
  begin
    var A; var X; var Y; // second Y
    A := 2; X := 3; Y := 5; P(A)
  end
end
```

What are in each case immediately after the procedure call the values of all variables that are currently “alive” (i.e., A , X , both instances of Y), assuming that the program is executed in a store that is initialized to 0 in all places?