LAB EXERCISE #1

Hochschule Landshut
Concepts of Modern Programming Languages

SoSe 2024
DATE ISSUED: April 17, 2024
DUE: DURING LAB SESSION #1

Problem 1: Chapter 4 Mitchell, 4.12 Static Assignment Languages

If you don't have the book, the question is also in Moodle. Read section 4.4.1 for the distinction between "imperative" and "declarative" (also in Moodle).

Problem 2: Pass-by-name has some weird semantics.

Read about Jensen's device (Mitchell page 96, in Moodle) and implement it in C´s Macro processor or another language that offers call by name.
Hint: in Gnu C a compound statement can be used as an expression, c.f.
https://gcc.gnu.org/onlinedocs/gcc-3.1/gcc/Statement-Exprs.html

What gets printed?

Problem 3: Denotational Semantics

Read the explanation about denotational semantics in chapter 4.3 in Mitchell (pages 67-76, available in Moodle). Then do exercise 4.8, which is printed below for your convenience:

The Mitchell text describes a denotational semantics for the simple imperative language given by the grammar

$P ::= x := e \mid P_1; P_2 \mid$ if e then P_1 else P_2 | while e do P.
Each program denotes a function from *states* to *states*, in which a *state* is a function

from *variables* to *values.*

1. (a) Calculate the meaning $C[x:=1;x:=x+1;](s_0)$ in approximately the same detail as that of the examples given in the text, where $s_0 = \lambda v \in variables.\ 0$, giving every variable the value 0.
2. (b) Denotational semantics is sometimes used to justify ways of reasoning about programs. Write a few sentences, referring to your calculation in part (a), explaining why

   $C[x := 1; x := x + 1;](s) = C[x := 2;](s)$ for every state s.