

Theory of Programming: Homework #1

Due date: 2024/01/03

Written part

This portion requires written proofs (in Chinese or English, not on Isabelle, even if they refer to problems in the book *Concrete Semantics*).

1. Carefully write down (as in the lecture slides) the other direction in the equivalence between big-step and small-step semantics.
2. (see Exercise 7.1) Define (by induction on the structure of the program) a function *assigned* from programs to a set of variables that may be assigned to in the program. Prove (by induction on big-step semantics) that if $(c, s) \Rightarrow t$ and x is not in the assigned set, then the value of x in s and t are the same.
3. (see Exercise 7.4) Define a small-step semantics for the evaluation of arithmetic expressions, specifying a left-to-right evaluation order. The syntax for arithmetic expressions is given by:

$$expr = N \ int \mid V \ var \mid Plus \ expr \ expr$$

where N denotes constants, V denotes variables, and $Plus$ denotes addition. For example, it should be possible to derive from the semantics the following:

$$(Plus (Plus (N \ 3) (V \ x)) (V \ y), \langle x := 5, y := 2 \rangle) \rightarrow^* (N \ 10, \langle x := 5, y := 2 \rangle).$$