Semantics of SA-Decl

We will use a stack machine as our abstract machine for operational semantics.

This stack machine consists of:

- a single-assignment store (a memory unit)
- a stack of semantic statements
  - a semantic statement is a pair of
    - an environment (a map from identifiers to references)
    - a statement.

Computation proceeds by

- push the pair of the program and an empty environment onto the stack
- at each step:
  - pop off the top of the stack
  - perform a computation based on what kind of statement was on top
  - repeat.

We write a "snapshot" of this machine as $\langle [Stack], \sigma \rangle$ where Stack is the stack and σ is a store.

Stack has the form $(S, \Sigma) \mid Stack$ or ε.

So, the initial snapshot of the computation of a program S is:

$$([(S, \emptyset) \mid \varepsilon], \emptyset)$$

where $\emptyset$ is the empty environment / state.

Rules for statements

Here, we will define a relation

$$\_ \longrightarrow \_ : \text{Snapshot} \to \text{Snapshot}$$

$$\frac{}{([(\text{skip}, \Sigma) \mid \text{Stack}], \sigma) \to ([\text{Stack}], \sigma)} \text{ skip}$$

$$\frac{}{([(S_1\ S_2, \Sigma) \mid \text{Stack}], \sigma) \longrightarrow ([(S_1, \Sigma) \mid (S_2, \Sigma) \mid \text{Stack}], \sigma)} \text{ composition}$$

$$\frac{}{([(\text{local var in } S, \Sigma) \mid \text{Stack}], \sigma) \to ([(S, \Sigma[\text{var} := \text{freshReference}()]) \mid \text{Stack}], \sigma,$$ local