

8 Semantics of Programming Languages (nk480)

Languages like FORTH and POSTSCRIPT are *stack-based languages*; they store intermediate values on a stack rather than binding to variable names. In this question we will look at how to give a type system and operational semantics for a simple stack-based language. The syntax and informal meaning of our language is given by:

$e ::=$	$\underline{n}$	Push the numeral $n$ on the stack
	$\underline{b}$	Push the Boolean $b$ on the stack
	<b>Add</b>	Replace the top two stack elements with their sum
	<b>Eq</b>	Replace the top two stack elements with the result of comparing them for equality
	<b>Cond</b> ( $e_1, e_2$ )	Delete the top stack element and execute $e_1$ or $e_2$ , depending on if the top of the stack was <b>True</b> or <b>False</b>
	<b>Skip</b>	No-op
	$e_1; e_2$	Run $e_1$ and then $e_2$
$v ::=$	$\underline{b} \mid \underline{n}$	Values
$s ::=$	$\cdot \mid s, v$	Stacks
$\tau ::=$	<b>bool</b> $\mid$ <b>num</b>	Types
$\Gamma ::=$	$\cdot \mid \Gamma, \tau$	Stack Types

We take a value  $v$  to be a Boolean or numeral, and define a stack  $s$  to be a stack of values (growing at the right). Correspondingly, there are types **bool** and **num** for values, and stack types  $\Gamma$  for stacks  $s$ .

The small-step operational semantics is then defined by a transition relation  $\langle e_1 \mid s_1 \rangle \mapsto \langle e_2 \mid s_2 \rangle$ . One rule for this relation is:

$$\overline{\langle \text{Add} \mid s, \underline{n}, \underline{m} \rangle \mapsto \langle \text{Skip} \mid s, \underline{n + m} \rangle}$$

The typing relation is given as a relation  $\Gamma \vdash e \dashv \Gamma'$ , which means that  $e$ , when run with a stack of shape  $\Gamma$ , yields a stack of shape  $\Gamma'$ . One rule for this relation is:

$$\overline{\Gamma, \text{num}, \text{num} \vdash \text{Add} \dashv \Gamma, \text{num}}$$

- (a) Give the remaining rules for the operational semantics. [7 marks]
- (b) Give the remaining rules for the typing judgement. [7 marks]
- (c) Formulate and state the progress and preservation lemmas for this language. [6 marks]