

Homework assignment

Use the template source file (`hw3.w1` in your repository), implement the denotational semantics described below.

Grammar rules

$$\begin{aligned}
 \text{program} &::= \text{compose}[\text{statement}, \text{program}] \\
 &\quad \epsilon \\
 \text{statement} &::= \text{assign}[\text{varName}, \text{expression}] \\
 &\quad \text{while}[b, c] \\
 \text{expression} &::= \text{number} \\
 &\quad \text{boolean} \\
 &\quad \text{plus}[\text{expr}, \text{expr}] \\
 &\quad \text{less}[\text{expr}, \text{expr}] \\
 &\quad \text{equal}[\text{expr}, \text{expr}] \\
 &\quad \text{or}[\text{expr}, \text{expr}] \\
 &\quad \text{and}[\text{expr}, \text{expr}] \\
 &\quad \text{value}[\text{varName}]
 \end{aligned} \tag{1}$$

Denotational Semantics

Value rules:

$$\llbracket n \rrbracket = \lambda env. n \tag{2}$$

$$\llbracket true \rrbracket = \lambda env. True \tag{3}$$

$$\llbracket false \rrbracket = \lambda env. False \tag{4}$$

$$\llbracket \text{varName} \rrbracket = \text{varName} \tag{5}$$

Statement rules:

$$\llbracket \text{assign}[\text{varName}, e] \rrbracket = \llbracket \text{assign} \rrbracket(\llbracket \text{varName} \rrbracket, \llbracket e \rrbracket) \tag{6}$$

$$\llbracket \text{assign} \rrbracket = \lambda v, e. \lambda env. env[v \mapsto e(env)] \tag{7}$$

$$\llbracket \text{compose}[stm, prg] \rrbracket = \llbracket \text{compose} \rrbracket(\llbracket stm \rrbracket, \llbracket prg \rrbracket) \tag{8}$$

$$\llbracket \text{compose} \rrbracket = \lambda s, p. \lambda env. p(s(env)) \tag{9}$$

$$\llbracket \text{epsilon} \rrbracket = \lambda env. env \tag{10}$$

$$\llbracket \text{while}[b, c] \rrbracket = \llbracket \text{while} \rrbracket(\llbracket b \rrbracket)(\llbracket c \rrbracket) \tag{11}$$

$$\llbracket \text{while} \rrbracket = Y \lambda r. \lambda b. \lambda c. \lambda env. (b(env) ? r(b)(c)(c(env)) : env) \tag{12}$$

Expression rules:

$$\llbracket \text{value}[\text{varName}] \rrbracket = \llbracket \text{value} \rrbracket(\llbracket \text{varName} \rrbracket) \tag{13}$$

$$\llbracket \text{value} \rrbracket = \lambda v. \lambda env. env[v] \tag{14}$$

$$\llbracket \text{binop}[e_1, e_2] \rrbracket = \llbracket \text{binop} \rrbracket(\llbracket e_1 \rrbracket, \llbracket e_2 \rrbracket) \tag{15}$$

$$\llbracket \text{plus} \rrbracket = \lambda e_1, e_2. \lambda env. e_1(env) + e_2(env) \tag{16}$$

$$\llbracket \text{less} \rrbracket = \lambda e_1, e_2. \lambda env. e_1(env) < e_2(env) \tag{17}$$

$$\llbracket \text{equal} \rrbracket = \lambda e_1, e_2. \lambda env. e_1(env) = e_2(env) \tag{18}$$

$$\llbracket \text{or} \rrbracket = \lambda e_1, e_2. \lambda env. e_1(env) \vee e_2(env) \tag{19}$$

$$\llbracket \text{and} \rrbracket = \lambda e_1, e_2. \lambda env. e_1(env) \wedge e_2(env) \tag{20}$$