

Homework assignment

Use the template source file (`hw2.w1` in your repository), implement the semantics and the type system described below.

Grammar rules

$$\begin{aligned}
 \text{program} &::= CBlock[\{\text{statementSequence}\}] \\
 \text{statementSequence} &::= \epsilon | \\
 &\quad \text{statement} | \\
 &\quad \text{statement}, \text{statementSequence} \\
 \text{statement} &::= CDeclare[\text{type}, \text{varName}] | \\
 &\quad \text{expression} | \\
 &\quad CWhile[\text{expression}, \{\text{statementSequence}\}] \\
 \text{expression} &::= \text{number} | \\
 &\quad \text{varName} | \\
 &\quad CAssign[\text{varName}, \text{expression}] | \\
 &\quad COperator[\text{unaryOp}, \text{expression}] | \\
 &\quad COperator[\text{binaryOp}, \{\text{expression}, \text{expression}\}] \\
 \text{unaryOp} &::= \text{Minus} | \\
 &\quad \text{Not} \\
 \text{binaryOp} &::= \text{Plus} | \\
 &\quad \text{Subtract} | \\
 &\quad \text{Times} | \\
 &\quad \text{Divide} | \\
 &\quad \text{Greater} | \\
 &\quad \text{GreaterEqual} | \\
 &\quad \text{Less} | \\
 &\quad \text{LessEqual} | \\
 &\quad \text{Equal} | \\
 &\quad \text{Unequal} | \\
 &\quad \text{And} | \\
 &\quad \text{Or} \\
 \text{type} &::= \text{int} | \\
 &\quad \text{double}
 \end{aligned} \tag{1}$$

Semantics

Convention: $k, k_1, k_2 \in \{\text{number} \cup \{\text{Null}\}\}$, $e, e_1, e_2 \in \text{expression}$, $\text{stmSeq} \in \{\text{statementSequence}\}$, $\text{stm*} \in \text{statementSequence}$ and $\cdot, +$ are standard operators on numbers. Result of logical operations on numbers is 0 if false, 1 if true.

Block rule:

$$\frac{(s, \{\text{stm*}\}) \Rightarrow (s', o)}{(s, CBlock[\{\text{stm*}\}]) \Rightarrow (s', o)} \tag{2}$$

Statement sequence rules:

$$\overline{(s, \{\})} \Rightarrow (s, \text{Null}) \quad (3)$$

$$\frac{(s, stm) \Rightarrow (s', o_1) \quad (s', \{stm*\}) \Rightarrow (s'', o_2)}{(s, \{stm, stm*\}) \Rightarrow (s'', o_2)} \quad (4)$$

Statement rules:

$$\overline{(s, CDeclare[type, var])} \Rightarrow (s[var \mapsto \text{Undefined}], \text{Null}) \quad (5)$$

$$\frac{(s, e) \Rightarrow (s', k) \wedge k = 0}{(s, \text{While}[e, \{stm*\}]) \Rightarrow (s', \text{Null})} \quad (6)$$

$$\frac{(s, e) \Rightarrow (s', k) \wedge k \neq 0 \quad (s', \{stm*\}) \Rightarrow (s'', k_1) \quad (s'', \text{While}[e, \{stm*\}]) \Rightarrow (s''', k_2)}{(s, \text{While}[e, \{stm*\}]) \Rightarrow (s''', \text{Null})} \quad (7)$$

Expression rules:

$$\overline{(s, k)} \Rightarrow (s, k) \quad (8)$$

$$\overline{(s, var)} \Rightarrow (s, s[var]) \quad (9)$$

$$\frac{(s, e) \Rightarrow (s', k)}{(s, CAssign[var, e]) \Rightarrow (s'[var \mapsto k], k)} \quad (10)$$

Unary operator rules:

$$\frac{(s, e) \Rightarrow (s', k)}{(s, COperator[\text{Minus}, e]) \Rightarrow (s', -k)} \quad (11)$$

$$\frac{(s, e) \Rightarrow (s', k) \wedge k = 0}{(s, COperator[\text{Not}, e]) \Rightarrow (s', 1)} \quad (12)$$

$$\frac{(s, e) \Rightarrow (s', k) \wedge k \neq 0}{(s, COperator[\text{Not}, e]) \Rightarrow (s', 0)} \quad (13)$$

Binary operator rules:

$$\frac{(s, e_1) \Rightarrow (s', k_1) \quad (s', e_2) \Rightarrow (s'', k_2)}{(s, COperator[\text{Plus}, \{e_1, e_2\}]) \Rightarrow (s'', k_1 + k_2)} \quad (14)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \quad (s', e_2) \Rightarrow (s'', k_2)}{(s, COperator[\text{Subtract}, \{e_1, e_2\}]) \Rightarrow (s'', k_1 - k_2)} \quad (15)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \quad (s', e_2) \Rightarrow (s'', k_2)}{(s, COperator[\text{Times}, \{e_1, e_2\}]) \Rightarrow (s'', k_1 \cdot k_2)} \quad (16)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \quad (s', e_2) \Rightarrow (s'', k_2) \wedge k_2 \neq 0}{(s, COperator[\text{Divide}, \{e_1, e_2\}]) \Rightarrow (s'', k_1 \div k_2)} \quad (17)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \quad (s', e_2) \Rightarrow (s'', k_2) \wedge k = 0}{(s, COperator[\text{Divide}, \{e_1, e_2\}]) \Rightarrow (s'', \$Failed)} \quad (18)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \quad (s', e_2) \Rightarrow (s'', k_2) \wedge k_1 > k_2}{(s, COperator[\text{Greater}, \{e_1, e_2\}]) \Rightarrow (s'', 1)} \quad (19)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \quad (s', e_2) \Rightarrow (s'', k_2) \wedge k_1 \leq k_2}{(s, COperator[Greater, \{e_1, e_2\}]) \Rightarrow (s'', 0)} \quad (20)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \quad (s', e_2) \Rightarrow (s'', k_2) \wedge k_1 \geq k_2}{(s, COperator[GreaterEqual, \{e_1, e_2\}]) \Rightarrow (s'', 1)} \quad (21)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \quad (s', e_2) \Rightarrow (s'', k_2) \wedge k_1 < k_2}{(s, COperator[GreaterEqual, \{e_1, e_2\}]) \Rightarrow (s'', 0)} \quad (22)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \quad (s', e_2) \Rightarrow (s'', k_2) \wedge k_1 < k_2}{(s, COperator[Less, \{e_1, e_2\}]) \Rightarrow (s'', 1)} \quad (23)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \quad (s', e_2) \Rightarrow (s'', k_2) \wedge k_1 \geq k_2}{(s, COperator[Less, \{e_1, e_2\}]) \Rightarrow (s'', 0)} \quad (24)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \quad (s', e_2) \Rightarrow (s'', k_2) \wedge k_1 \leq k_2}{(s, COperator[LessEqual, \{e_1, e_2\}]) \Rightarrow (s'', 1)} \quad (25)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \quad (s', e_2) \Rightarrow (s'', k_2) \wedge k_1 > k_2}{(s, COperator[LessEqual, \{e_1, e_2\}]) \Rightarrow (s'', 0)} \quad (26)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \quad (s', e_2) \Rightarrow (s'', k_2) \wedge k_1 = k_2}{(s, COperator[Equal, \{e_1, e_2\}]) \Rightarrow (s'', 1)} \quad (27)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \quad (s', e_2) \Rightarrow (s'', k_2) \wedge k_1 \neq k_2}{(s, COperator[Equal, \{e_1, e_2\}]) \Rightarrow (s'', 0)} \quad (28)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \quad (s', e_2) \Rightarrow (s'', k_2) \wedge k_1 \neq k_2}{(s, COperator[Unequal, \{e_1, e_2\}]) \Rightarrow (s'', 1)} \quad (29)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \quad (s', e_2) \Rightarrow (s'', k_2) \wedge k_1 = k_2}{(s, COperator[Unequal, \{e_1, e_2\}]) \Rightarrow (s'', 0)} \quad (30)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \wedge k_1 = 0}{(s, COperator[And, \{e_1, e_2\}]) \Rightarrow (s', 0)} \quad (31)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \wedge k_1 \neq 0 \quad (s', e_2) \Rightarrow (s'', k_2) \wedge k_2 = 0}{(s, COperator[And, \{e_1, e_2\}]) \Rightarrow (s'', 0)} \quad (32)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \wedge k_1 \neq 0 \quad (s', e_2) \Rightarrow (s'', k_2) \wedge k_2 \neq 0}{(s, COperator[And, \{e_1, e_2\}]) \Rightarrow (s'', 1)} \quad (33)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \wedge k_1 \neq 0}{(s, COperator[Or, \{e_1, e_2\}]) \Rightarrow (s', 1)} \quad (34)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \wedge k_1 = 0 \quad (s', e_2) \Rightarrow (s'', k_2) \wedge k_2 \neq 0}{(s, COperator[Or, \{e_1, e_2\}]) \Rightarrow (s'', 1)} \quad (35)$$

$$\frac{(s, e_1) \Rightarrow (s', k_1) \wedge k_1 = 0 \quad (s', e_2) \Rightarrow (s'', k_2) \wedge k_2 = 0}{(s, COperator[Or, \{e_1, e_2\}]) \Rightarrow (s'', 0)} \quad (36)$$

Type rules

Convention: $e, e_1, e_2 \in \text{expression}$, $d \in \text{Double}$, $n \in \text{Integer}$, $\text{var} \in \text{varName}$, $\text{stmSeq} \in \{\text{statementSequence}\}$, $\text{stm*} \in \text{statementSequence}$ and $t \in \{\text{int}, \text{double}\}$.

$$\overline{\Gamma \vdash n : \text{int}} \quad (37)$$

$$\overline{\Gamma \vdash d : \text{double}} \quad (38)$$

$$\frac{\Gamma \vdash \{\text{stm*}\} : \diamond}{\Gamma \vdash \text{CBlock}[\{\text{stm*}\}] : \diamond} \quad (39)$$

$$\overline{\Gamma \vdash \{\} : \diamond} \quad (40)$$

$$\frac{\Gamma \vdash e : t \quad \Gamma \vdash \{\text{stm*}\} : \diamond}{\Gamma \vdash \{e, \text{stm*}\} : \diamond} \quad (41)$$

$$\frac{\Gamma \vdash e : \text{int} \quad \Gamma \vdash \text{var} : \text{double}}{\Gamma \vdash \text{CAssign}[\text{var}, e] : \text{double}} \quad (42)$$

$$\frac{\Gamma \vdash e : t \quad \Gamma \vdash \text{var} : t}{\Gamma \vdash \text{CAssign}[\text{var}, e] : t} \quad (43)$$

$$\frac{\Gamma \vdash \text{CAssign}[\text{var}, e] : t \quad \Gamma \vdash \{\text{stm*}\} : \diamond}{\Gamma \vdash \{\text{CAssign}[\text{var}, e], \text{stm*}\} : \diamond} \quad (44)$$

$$\frac{\text{var} \notin \text{dom}(\Gamma)}{\Gamma \vdash \text{CDeclare}[\text{type}, \text{var}] : \diamond} \quad (45)$$

$$\frac{\Gamma \vdash \text{CDeclare}[\text{type}, \text{var}] : \diamond \quad \Gamma \cup (\text{var}, \text{type}) \vdash \{\text{stm*}\} : \diamond}{\Gamma \vdash \{\text{CDeclare}[\text{type}, \text{var}], \text{stm*}\} : \diamond} \quad (46)$$

$$\frac{\Gamma \vdash e : t \quad \text{stmSeq} : \diamond}{\Gamma \vdash \text{CWhile}[e, \text{stmSeq}] : \diamond} \quad (47)$$

$$\frac{\Gamma \vdash \text{CWhile}[e, \text{stmSeq}] : \diamond \quad \Gamma \vdash \{\text{stm*}\} : \diamond}{\Gamma \vdash \{\text{CWhile}[e, \text{stmSeq}], \text{stm*}\} : \diamond} \quad (48)$$

$$\overline{\Gamma \vdash \text{var} : \Gamma(\text{var})} \quad (49)$$

$$\frac{\Gamma \vdash \text{var} : \Gamma(\text{var}) \quad \Gamma \vdash \{\text{stm*}\} : \diamond}{\Gamma \vdash \{\text{var}, \text{stm*}\} : \diamond} \quad (50)$$

$$\frac{\Gamma \vdash e : t}{\Gamma \vdash \text{COperator}[\text{unaryOp}, e] : t} \quad (51)$$

$$\frac{\Gamma \vdash \text{COperator}[\text{unaryOp}, e] : t \quad \Gamma \vdash \{\text{stm*}\} : \diamond}{\Gamma \vdash \{\text{COperator}[\text{unaryOp}, e], \text{stm*}\} : \diamond} \quad (52)$$

$$\frac{\Gamma \vdash e_1 : \text{int} \quad \Gamma \vdash e_2 : \text{double}}{\Gamma \vdash \text{COperator}[\text{binaryOp}, \{e_1, e_2\}] : \text{double}} \quad (53)$$

$$\frac{\Gamma \vdash e_1 : \text{double} \quad \Gamma \vdash e_2 : \text{int}}{\Gamma \vdash \text{COperator}[\text{binaryOp}, \{e_1, e_2\}] : \text{double}} \quad (54)$$

$$\frac{\Gamma \vdash e_1 : t \quad \Gamma \vdash e_2 : t}{\Gamma \vdash \text{COperator}[\text{binaryOp}, \{e_1, e_2\}] : t} \quad (55)$$

$$\frac{\Gamma \vdash \text{COperator}[\text{binaryOp}, \{e_1, e_2\}] : t \quad \Gamma \vdash \{\text{stm*}\} : \diamond}{\Gamma \vdash \{\text{COperator}[\text{binaryOp}, \{e_1, e_2\}], \text{stm*}\} : \diamond} \quad (56)$$