```
The semantics for statements:
c    : constant
str  : string
r    : range
car  : char array
v, w : vector
mat  : matrix
x, y : variables
H    : heap for storage
ssep : statement separator
ctx  : stack, function_table

             c if H = H', x → c
           str if H = H', x → str
           car if H = H', x → car
           v   if H = H', x → v
           mat if H = H', x → mat
H(x)= {  H'(x) if H = H', y → c'  and y <> x
         H'(x) if H = H', y → str'and y <> x
         H'(x) if H = H', y → car'and y <> x
         H'(x) if H = H', y → vec'and y <> x
         H'(x) if H = H', y → mat'and y <> x
             0 if H = ·
```

Search Stack 1

$$\frac{name \in frames}{Stack : name \rightarrow Stack : value}$$

Search Stack 2

$$\frac{name \notin frames}{Stack : name \rightarrow Stack : skip}$$

Update Stack

$$\frac{}{Stack : name, value \rightarrow Stack ; name \Rightarrow value ; skip}$$

clear_var_in_frame

$$\frac{}{Stack \rightarrow Stack - Stack[name] ; skip}$$

push_frame

$$\frac{}{Stack, frame \rightarrow Stack + frame ; skip}$$

pop_frame

$$\frac{Statck \rightarrow head, tail}{Stack \rightarrow head ; skip}$$

get_fn 1

$$\frac{name \in function\,table}{function\,table, name \rightarrow function ; skip}$$

get_fn 2

$$\frac{name \notin function\ table}{function\ table, name \rightarrow skip}$$

H; e ⇓ c

const

H; c ⇓ c

var1

H; x ⇓ H(x)

L; H1; s1 → L; H2; s2

seq1

L; H; ssep s → L; H; s

seq2

L; H; s1 → L; H'; s1'
L; H; s1 ssep s2 → L; H'; s1' ssep s2

assign1

L; H; e ⇓ c
L; H; x=e → L; H; x → c; ssep

if1

L; H; e ⇓ b b==1
L; H; if e ssep s1 end → L; H; s1

if2

L; H; e ⇓ b b==0
L; H; if e ssep s1 end → L; H; ssep

if3

L; H; e ⇓ b b==1
L; H; if e ssep s1 else s2 end → L; H; s1

if4

L; H; e ⇓ b b==0
L; H; if e ssep s1 else s2 end → L; H; s2

if5

L; H; e1 ⇓ b1 b1==0 L; H; e2 ⇓ b2 b2==1
L; H; if e1 ssep s1 elseif e2 ssep s2 end → L; H; s2

if6

L; H; e1 ⇓ b1 b1==0 L; H; e2 ⇓ b2 b2==0
L; H; if e1 ssep s1 elseif e2 ssep s2 end → L; H; ssep

if7

L; H; e1 ⇓ b1 b1==0 L; H; e2 ⇓ b2 b2==0
L; H; if e1 ssep s1 elseif e2 ssep s2 else s3 end → L; H; s3

for

L; H; e ⇓ r
L; H; for e ssep s end → L; H; if e <= END ssep (s; for e’ ssep s end) end

Binary Operations

addition

$$\frac{e1 \rightarrow c1, e2 \rightarrow c2}{ctx, e1+e2 \rightarrow ctx, c1+c2}$$

subtraction

$$\frac{e1 \rightarrow c1, e2 \rightarrow c2}{ctx, e1 + e2 \rightarrow ctx, c1 + c2}$$

multiplication

$$\frac{e1 \rightarrow c1, e2 \rightarrow c2}{ctx, e1 * e2 \rightarrow ctx, c1 * c2}$$

division

$$\frac{e1 \rightarrow c1, e2 \rightarrow c2}{ctx, e1 + e2 \rightarrow ctx, c1 / c2}$$

less than

$$\frac{e1 \rightarrow c1, e2 \rightarrow c2}{ctx, e1 < e2 \rightarrow ctx, c1 < c2}$$

greater than

$$\frac{e1 \rightarrow c1, e2 \rightarrow c2}{ctx, e1 > e2 \rightarrow ctx, c1 > c2}$$

less than or equal to

$$\frac{e1 \rightarrow c1, e2 \rightarrow c2}{ctx, e1 \leq e2 \rightarrow ctx, c1 \leq c2}$$

greater than or equal to

$$\frac{e1 \rightarrow c1, e2 \rightarrow c2}{ctx, e1 \geq e2 \rightarrow ctx, c1 \geq c2}$$

equals

$$\frac{e1 \rightarrow c1, e2 \rightarrow c2}{ctx, e1 = e2 \rightarrow ctx, c1 = c2}$$

not equal

$$\frac{\texttt{e1} \rightarrow \texttt{c1},\ \texttt{e2} \rightarrow \texttt{c2}}{\texttt{ctx; e1 ~= e2} \rightarrow \texttt{c1 ~= c2}}$$

logical OR

$$\frac{\texttt{e1} \rightarrow \texttt{c1},\ \texttt{e2} \rightarrow \texttt{c2}}{\texttt{ctx; e1 || e2} \rightarrow \texttt{c1 || c2}}$$

logical AND

$$\frac{\texttt{e1} \rightarrow \texttt{c1},\ \texttt{e2} \rightarrow \texttt{c2}}{\texttt{ctx; e1 \&\& e2} \rightarrow \texttt{c1 \&\& c2}}$$

$v^e, w^e,$ `: vectors of expressions`

(n, v^e) : vector of n expression $[v_1, v_2, \ldots, v_n]$

v^c, w^c : vectors of constants

(m, n, A) : matrix of size $m \times n$ $\begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \\ a_{m,1} & & a_{m,n} \end{bmatrix}$

Array access of an integer index

$$\overline{v(c) \rightarrow v_c}$$

Array access of an expression that evaluates to an integer

$$\frac{e \rightarrow c}{V(e) \rightarrow V_c}$$

Vector expression evaluation

$$\frac{v^e = [e_1, e_2, \ldots e_n]}{v^e \rightarrow [v_1^c, v_2^c, \ldots, v_n^c]}$$

Array acess of a range

$$\frac{RANGE \rightarrow (BEGIN, END)}{V(RANGE) \rightarrow [v_{begin}, v_{begin+1}, v_{begin+2}, \ldots V_{end}]}$$

scalar-vector multiplication

$$\frac{e \rightarrow c}{e * v \rightarrow [c * v_1, c * v_2, \ldots c * v_n]}$$

vector-vector addition

$$\frac{v^e \rightarrow v^c, w^e \rightarrow w^c}{v^e + w^e \rightarrow [v_1^c + w_1^c, v_2^c + w_2^c, \ldots, v_n^c + w_n^c]}$$