The semantics for statements:
c: constant
str: string
car: char array
v : vector
mat: matrix
x, y: variables
H: heap for storage

```
            c if H = H', x->c
          str if H = H', x->str
          car if H = H', x->car
          vec if H = H', x->vec
          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 = ·
```

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