Principles of Programming Languages - Homework 5

Abhi Agarwal

1 Problem 1

(a)

(i) $$\dfrac{\dfrac{}{\{x\to 3,y\to -2\}\vdash 3\Downarrow 3}\,EvalVal \quad \dfrac{x\in dom(\{x\to 3,y\to -2\})}{\{x\to 3,y\to -2\}\vdash x\Downarrow 3}\,EvalVar \quad 9=3*3}{\{x\to 3,y\to -2\}\vdash x*3\Downarrow 9}\,EvalTimes \quad \dfrac{}{(\{x\to 3,y\to -2\})\vdash 2\Downarrow 2}\,EvalVal \quad 11=9+2}{\{x\to 3,y\to -2\}\vdash 3*x+2\Downarrow 11}\,EvalPlus$$

(ii) $$\dfrac{\dfrac{\dfrac{}{\{x\to 3,y\to -2\}\vdash 2\Downarrow 2}\,EvalVal \quad \dfrac{y\in dom(\{x\to 3,y\to -2\})}{\{x\to 3,y\to -2\}\vdash y\Downarrow -2}\,EvalVar \quad 0=2+-2}{\{x\to 3,y\to -2\}\vdash 2+y\Downarrow 0}\,EvalPlus}{\{x\to 3,y\to -2\}\vdash const b=0\Downarrow 0}\,EvalConstDecl \quad false=toBool(0) \quad \dfrac{y\in dom(\{x\to 3,y\to -2\})}{\{x\to 3,y\to -2\}\vdash y\Downarrow -2}\,EvalVar}{\{x\to 3,y\to -2\}\vdash b?x:y\Downarrow -2}\,EvalIfThen$$

(b)

(i) 3 + $\underline{(1 \;\&\&\; 5)} \xrightarrow{a} \underline{3 + 5} \xrightarrow{b} 8 \xrightarrow{c} 8$

a: SearchBop1, DoAndTrue

b: SearchBop2, DoPlus

c: DoConstDecl

(ii) const x = $\underline{2 + 1}$; x * 0 ? x : x + x $\xrightarrow{a}$ const x = 3; $\underline{\text{x}}$ * 0 ? x : x + x $\xrightarrow{c}$ const x = 3; $\underline{0}$? x : x + x $\xrightarrow{d}$ const x = 3; $\underline{3 + 3} \xrightarrow{e} 6$

a: SearchConstDecl1, DoPlus

b: SearchConstDecl2, DoVar

c: SearchIf, DoTimes
d: SearchConstDecl1, DoPlus
e: DoConstDecl

(c)

SearchPlus1 $\frac{env \vdash e_2 \to e_2'}{env \vdash e_1 + e_2 \to e_1 + e_2'}$
SearchPlus2 $\frac{env \vdash e_1 \to e_1'}{env \vdash e_1 + v_2 \to e_1' + v_2}$
DoPlus $\frac{v = toNum(v_1) + toNum(v_2)}{env \vdash v_1 + v_2 = v}$
Order for right-to-left evaluation: SearchPlus1, SearchPlus2, DoPlus

(d)

Big-step SOS $\frac{env \vdash e_1 \to v_1 \;\; env \vdash e_2 \to v_2 \;\; v = v_2}{env \vdash e_1, e_2 \Downarrow v}$
Small-step SOS
SearchComma1 $\frac{env \vdash e_1 \to e_1'}{env \vdash e_1, e_2 \to e_1', e_2}$
SearchComma2 $\frac{env \vdash e_2' \to e_2'}{env \vdash v_1, e_2 \to v_1, e_2'}$
DoReturnValue $\frac{v = v_2}{env \vdash v_1, v_2 = v}$
Order for right-to-left evaluation: SearchComma1, SearchComma2, DoReturnValue

(e)

(i) This program evaluates to 5. During evaluation, the EvalVar is applied three times as follows:

The first application is the using occurrence of y in the definition of the function f on line 3. In this case, y was bound to the variable 3 in the call to f on line 4.

The second application is the occurrence of x in the definition of the function g on line 2. x is bound to the value 2 in the constant declaration of 2 on line 1.

The third application is for the occurrence of y in the definition of the function g on line 2. This occurrence of y was bound to the value 3 in the call to g on line 3.

(ii) This program evaluates to 6. During evaluation, the EvalVar rule is applied four times as follows:

The first application is the using occurrence of y in the definition of function f on line 3. In this case, y was bound to the value 3 in the call to f on line 4.

The second application is the using occurrence of x in the definition of the function g on line 2. This occurrence was bound to the value 3 in the call to g on line 3.

The third application is the using occurrence of y in the definition of the function g on line 2. This occurrence was bound to the value that will be passed into the function by the function call in line 3. The function call to g returns a function, and that function is called on line 3 using the value 3. So the occurrence was bound to the value 3 in the call to the anonymous function on line 3.

The fourth application is the using occurrence of y in the definition of function f on line 3. This occurrence was bound to the value 3 in the call to the anonymous function that is returned by calling g on line 3.

2 Problem 2

(a)

$e_1 = (3 * y) + 4$

(b)

$e_1 = (x * y) + 4$

(c)

$e_2 = const\, y = y;\ 3 + y$

(d)

$e_2 = const\, y = 3; x + y$

(e)

$e_3 = const\, x = (function(z)(x(z))); x(y(2))$

(f)

$e_3 = const\, x = (function(z)(y(x(z)))); x(y)$