Comp105-HW2

aktsuetaki

September 2020

1 Question 10

1.1 a

The variable x is defined either locally in a function, or globally outside the current function.
(note I am using local variables as a comp11 student may not recognize it if we say formal parameter)

1.2 d

Some expression e resolves to v and does not change the global or local variables. must be the same.

2 Question 11

2.1 a

$x \notin \xi < e, \xi, \phi, \rho > \Downarrow < v, \xi', \phi, \rho' >$

2.2 c

$< e, \xi, \phi, \rho > \Downarrow < v, \xi, \phi, \rho' >$

2.3 d

$< e, \xi, \phi, \rho > \Downarrow < v, \xi', \phi, \rho' >$

3 Exercises R

3.1 WhileEnd

$$\frac{< e1,\xi,\phi,\rho > \Downarrow < v1,\xi',\phi,\rho' > \quad v1 = 0}{< While(e1,e2),\xi,\phi,\rho > \Downarrow < 0,\xi',\phi,\rho' >}\text{WhileEnd'}$$

$$\frac{< e1,\xi,\phi,\rho > \Downarrow < v1,\xi',\phi,\rho' > \quad v1 = 0}{< While(e1,e2),\xi,\phi,\rho > \Downarrow < 0,\xi',\phi,\rho' >}\text{WhileEnd}$$

This function WhileEnd' resolves the to the same as WhileEnd as in both cases the value resoved to is 0 and in both cases the ξ and ρ can be changed one time.

3.2 FormalAssign

$$\frac{x \in dom\rho \quad < e,\xi,\phi,\rho > \Downarrow < v,\xi',\phi,\rho' >}{< set(x,e2),\xi,\phi,\rho > \Downarrow < 0,\xi',\phi,\rho' >}\text{-FormalAssign'}$$

$$\frac{x \in dom\rho \quad < e,\xi,\phi,\rho > \Downarrow < v,\xi',\phi,\rho' >}{< set(x,e2),\xi,\phi,\rho > \Downarrow < 0,\xi',\phi,\rho'\{x- > v\} >}\text{-FormalAssign}$$

The FormalAssign' is different than FormalAssign as x is never updates in ρ. Because of this the expression will resolve to v (output v) , but not change x in memory. As such whenever you call x later in the function the value will not have changed.
(i.e)
(Val x 0) (Set x 1) (+ x 1)
x

This will return 1 in FormalAssign' and 2 in FormalAssign. Thus there is a difference.

4 Question 16

4.1 a

Changing the impcore GlobalVar to AwkGlobalVar

$$\frac{x \notin dom\rho \quad x \notin dom\xi \qquad < e, \xi, \phi, \rho > \Downarrow < 0, \xi', \phi, \rho' >}{< var(x), \xi, \phi, \rho > \Downarrow < 0, \xi'\{x- > 0\}, \phi, \rho' >}\text{AwkGlobalVar}$$

4.2 b

Changing the impcore GlobalVar to IconGlobalVar

$$\frac{x \notin dom\rho \quad x \notin dom\xi \qquad < e, \xi, \phi, \rho > \Downarrow < 0, \xi', \phi, \rho' >}{< var(x), \xi, \phi, \rho > \Downarrow < 0, \xi', \phi, \rho'\{x- > 0\} >}\text{IconGlobalVar}$$

4.3 c

I prefer to use Icon as adding a local variable rather than a global variable makes object oriented programming easier as it makes adding new variables more modular. If we use Awks the global variables created must be carefully worked around in case other functions. i.e. if two functions want to define a variable x using AwkGlobalVar, the first will define the x correctly and the second will overwrite the first functions x value.

4.4 d

```
(define Reset-Val
(begin
(set x 1)
x
)
)

(define def-var
(begin
(set x 0)
Reset-Val
x
)
)
```

Calling def-var we can test if a function is in awks or icon or standard impcore. If in standard the function will return an error as (set x 0) calls x without a global or local variable named x.

If we have an awks representation (set x 0) will create an global variable x which means when (set x 1) is called in Reset-Val, the value of x will be set to 1 and when x is called in def-var, it will return 1.

we have an Icon representation, (set x 0) will create an local variable x and when (set x 1) is called in Reset-Val another local variable will be created. The value of x will thus still be set to 0 when x is called in def-var as the local variable x in Reset-Val will not interact with local variable x in def-var. It will thus return 0;

5 Question 12

$$\begin{array}{ll}
begin() \quad x \in \rho & \\
\hline
set < (SetX3), \xi, \phi, \rho > \Downarrow < 3, \xi', \phi, \rho'\{x- > 3\} > & \text{Begin} \\
\hline
< x, \xi', \phi, \rho' > \Downarrow < \rho(x), \xi'', \phi, \rho'' > \Rightarrow \rho(x) = 3 & \text{Set} \\
\hline
< begin(setx3)x, \xi, \phi, \rho > \Downarrow < v, \xi'', \phi, \rho'' > \Rightarrow v = 3 \quad \rho'' = \rho' \quad \xi'' = \xi' & \text{x}
\end{array}$$

Begin (set x 3) x is a command that starts with begin then moving to set and then x. As such we start with begin where $\rho(x) = 99$ then we set x to 3 then call x. as (set x 3) changes $\rho(x- > 3)$ the value the expression is resolved to will be 3.

6 Question 13

If statement
$x \in \rho$
————————x
$< x, \xi, \phi, \rho > \Downarrow < v_1, \xi', \phi, \rho' > \quad v_1 \neq 0 \quad < x, \xi, \phi, \rho > \Downarrow < \rho(x), \xi', \phi, \rho' >$
——If-not-equals
$< If(var(x), var(x), 0), \xi, \phi, \rho > \Downarrow < V, \xi'', \phi, \rho'' > V = \rho(x)$

and

$x \in \rho$
————————x
$< x, \xi, \phi, \rho > \Downarrow < v_1, \xi', \phi, \rho' > \quad v_1 = 0 \quad < x, \xi, \phi, \rho > \Downarrow < 0, \xi', \phi, \rho' >$
——If-equals
$< If(var(x), var(x), 0), \xi, \phi, \rho > \Downarrow < V, \xi'', \phi, \rho'' > V = \rho(x) = 0$

These if staements have $\xi'' = \xi' = \xi$ and $\rho'' = \rho' = \rho$ as after the if statement calls doesn't change any variable and x = x as shown above.

x
$x \in \rho \quad \rho(x) = x < x, \xi, \phi, \rho > \Downarrow < \rho(x), \xi', \phi, \rho' >$
——————————————————————————-x
$< Var(x), \xi, \phi, \rho > \Downarrow < \rho(x), \xi', \phi, \rho' >$

The X expression has $\xi' = \xi$ and $\rho' = \rho$ as after Var(x) only retrieves $\rho(x)$ and $\rho(x) = v_1 = v_2$.

As both the if statement and Var(x) both resolve to $\rho(x)$ and the ξ and ρ environments remain the same throughout for both, the two expressions are functionally identical.

7 Exercises F

Conjecture: If global variable y is defined, and if y does not appear in expression e, and if e evaluates to some value, then the evaluation of e does not change the value of y.

7.1 a

For some function F that changes global variable y and some expression e where y is not a free variable in it, we can have the following counter examples for why the conjecture above does not hold.

7.1.1

set x F

7.1.2

if F 17 99

7.1.3

begin F (set x 1) x

7.1.4

* x F

7.1.5

check-expect F 2

7.2 b

For some set expression "S", the expression is represented as follows: (set "variable-name" exp). in the case when the variable-name is in the ξ environment, the $\xi(x)$ is updated to whatever exp resolves to, and when variable-name is in the ρ environment, the $\rho(x)$ is upated to whatever exp resolves to. In the case when exp is the function call F following the definition of F in subsection a above, the set expression will be properly updated, while the F function changes y. As such any function that uses set as a sub-expression also could change a global variable, for example the if expression.

7.3 c

The begin expression fails if one of the sub-expressions is a function call to F as discribed in subsection a. as the Apply function will concatinates all the free variables from the sub expressions, save for the function due to the function resolving before the free variables are counted, the free variables will not include y if all other sub-expressions of begin do not include y as a free variable.

7.4 d

The conjecture fails as if an expression calls a function within it, the function will resolve and return some value, but the variables within the function will not be seen by the expression. Because of this the free variables of e will not include those of the function. Thus if the function contains code to change some global variable y, the expression e will not include y as a free variable although it is changed. We thus can have an expression that changes a global variable y without having y as a free variable, meaning y does not appear in the expression.