

# Fundamentos de lenguajes de programación

## Semántica de los Conceptos Fundamentales de Lenguajes de Programación

Facultad de Ingeniería. Universidad del Valle

Marzo 2018

# Contenido

## 1 Ejemplos

# Contenido

1 Ejemplos

2 Ejercicios

# Contenido

1 Ejemplos

2 Ejercicios

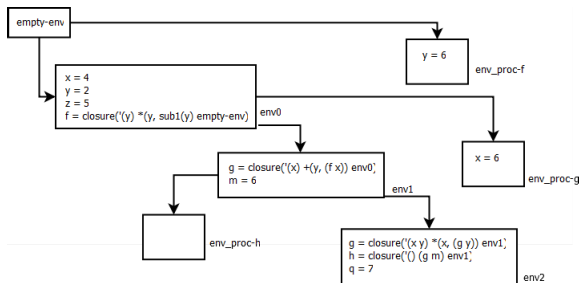
# Ejemplos procedimientos

## Ejemplo 1:

Sea el ambiente  $env_0$  con símbolos  $(x\ y\ z\ f)$  y valores  $(4\ 2\ 5\ (\text{closure}'(y)\ *(y, \text{sub1}(y))\ \text{empty-env}))$  el ambiente inicial de computación. Evaluar:

```
let
  g = proc(x) +(y, (f x))
  m = 6
in
  let
    g = proc(x y) *(x, (g y))
    h = proc() (g m)
    q = 7
  in
    -((h), q)
```

# Ejemplos procedimientos



# Ejemplos procedimientos

```
(eval-expression
  <<let
    g = proc(x) +(y, (f x))
    m = 6
  in
    let
      g = proc(x y) *(x, (g y))
      h = proc() (g m)
      q = 7
    in
      -((h), q)>>
  env0)
```

# Ejemplos procedimientos

```
= (eval-expression
   <<let
     g = proc(x y) *(x, (g y))
     h = proc() (g m)
     q = 7
   in
     -((h), q)>>
   env1)
```

```
= (eval-expression
   <<-((h), q)>>
   env2)
```



# Ejemplos procedimientos

```
= (-  
  (eval-expression  
    <<(h)>>  
    env2)  
  (eval-expression  
    <<q>>  
    env2))
```

# Ejemplos procedimientos

```
= (-  
  (apply-procedure  
    (closure '() << (g m) >> env1)  
    '())  
  7)
```

# Ejemplos procedimientos

```
= (-  
  (eval-expression  
    << (g m) >>  
    env-proc-h)  
  7)
```

```
= (-  
  (apply-procedure  
    (closure '(x) << +(y, (f x)) >> env0)  
    (eval-rands  
      '(<< m >>)  
      env-proc-h))  
  7)
```

# Ejemplos procedimientos

```
= (-  
  (apply-procedure  
    (closure '(x) << +(y, (f x)) >> env0)  
    '(6))  
  7)
```

```
= (-  
  (eval-expression  
    << +(y, (f x)) >>  
    env_proc-g)  
  7)
```

# Ejemplos procedimientos

```
= (-
    (+
      2
      (apply-procedure
        (closure '(y) *(y, sub1(y)) empty-env)
        (eval-rands '(<< x >>)
                     env_proc-g)
      )
    )
  )
```

# Ejemplos procedimientos

```
= (-  
  (+  
    2  
    (apply-procedure  
      (closure '(y) *(y, sub1(y)) empty-env)  
      '(6))  
  )  
  7)
```

```
= (-  
  (+  
    2  
    (eval-expression  
      << *(y, sub1(y)) >>  
      env-proc-f)  
  )  
  7)
```

# Ejemplos procedimientos

$$= (-$$
$$(+$$
$$^2$$
$$(* 6 5))$$
$$7)$$

$$= (-$$
$$(+ 2 30)$$
$$7)$$

$$= (- 32 7)$$

$$= 25$$

## Ejemplo 2:

Sea el ambiente  $env_0$  con símbolos  $(x\ y\ z\ f)$  y valores  $(4\ 2\ 5\ (\text{closure}'(y)\ *(y, 6)\ \text{empty-env}))$  el ambiente inicial de computación. Evaluar:

```
let
  g = proc(f h x) (f (h x))
  k = 8
in
  +((g f f k), z)
```



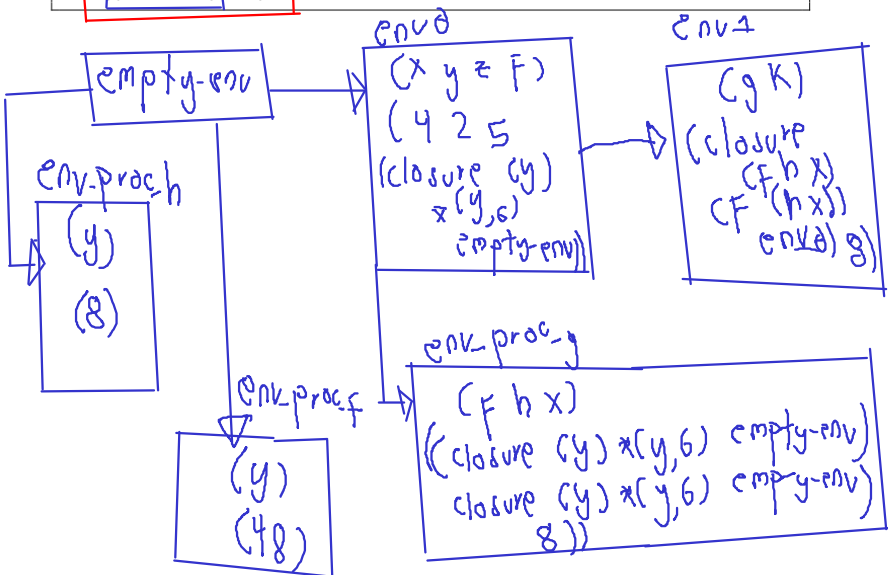
Sea el ambiente  $env_0$  con símbolos  $(x\ y\ z\ f)$  y valores  $(4\ 2\ 5\ (closure'\ (y)\ *(y,\ 6)\ empty-env))$  el ambiente inicial de computación. Evaluar:

```

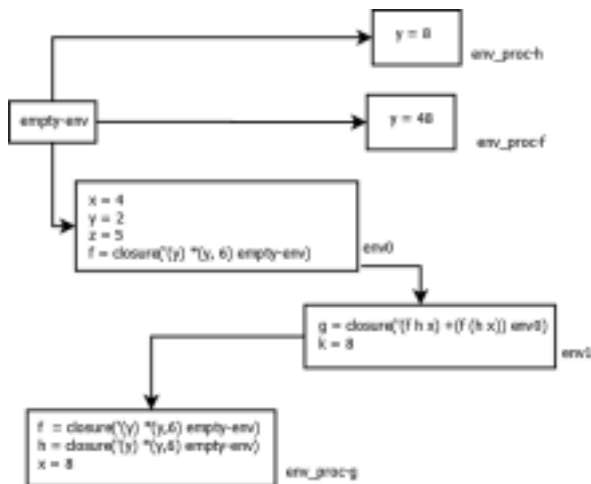
let
  g = proc(f h x) (f (h x))
  k = 8
in
  +((g f f k), z)

```

Handwritten annotations:  $288$  above the `g` definition,  $48$  next to `(f (h x))`,  $288$  next to `g` in the body, and  $293$  next to the final expression.



# Ejemplos procedimientos



# Ejemplos procedimientos

```
(eval-expression
  <<let
    g = proc(f h x) (f (h x))
    k = 8
  in
    +((g f f k), z)>>
  env0)
```

# Ejemplos procedimientos

```
=(eval-expression  
  <<+((g f f k), z)>>  
  env1)
```

```
= +((eval-expression  
      <<(g f f k)>>  
      env1),  
    (eval-expression  
      <<z>>  
      env1))
```

# Ejemplos procedimientos

```
= +(((eval-expression
      <<g>>
      env1)
  (eval-rands
    '(<<f>> <<f>> <<k>>)
    env1)),
5)
```

# Ejemplos procedimientos

```
= +((apply-procedure
      (closure '(f h x) <<(f (h x))>> env0)
      '( (closure '(y) *(y, 6) empty-env)
        (closure '(y) *(y, 6) empty-env)
        8)),
      5)
```

```
= +((eval-expression
      <<(f (h x))>>
      env_proc-g),
      5)
```

# Ejemplos procedimientos

```
= +(((eval-expression
      <<f>>
      env_proc-g)
  (eval-rands
    '(<<(h x)>>)
    env_proc-g)),
5)
```

```
= +((apply-procedure
  (closure '(y) *(y, 6) empty-env)
  '(((eval-expression
      <<h>>
      env_proc-g)
  (eval-rands
    '(<<x>>)
    env_proc-g))))),
5)
```

# Ejemplos procedimientos

```
= +((apply-procedure
      (closure '(y) *(y, 6) empty-env)
      '((apply-procedure
          (closure '(y) *(y, 6) empty-env)
          '(8))))) ,
5)
```

```
= +((apply-procedure
      (closure '(y) *(y, 6) empty-env)
      '((eval-expression
          <<*(y, 6)>>
          env_proc-h)))) ,
5)
```



# Ejemplos procedimientos

```
= +((apply-procedure  
      (closure '(y) *(y, 6) empty-env)  
      '(48)),  
    5)
```

```
= +((eval-expression  
      *(y, 6)  
      env-proc-f),  
    5)
```

# Ejemplos procedimientos

= +(288, 5)

= 293

## Ejemplo 3:

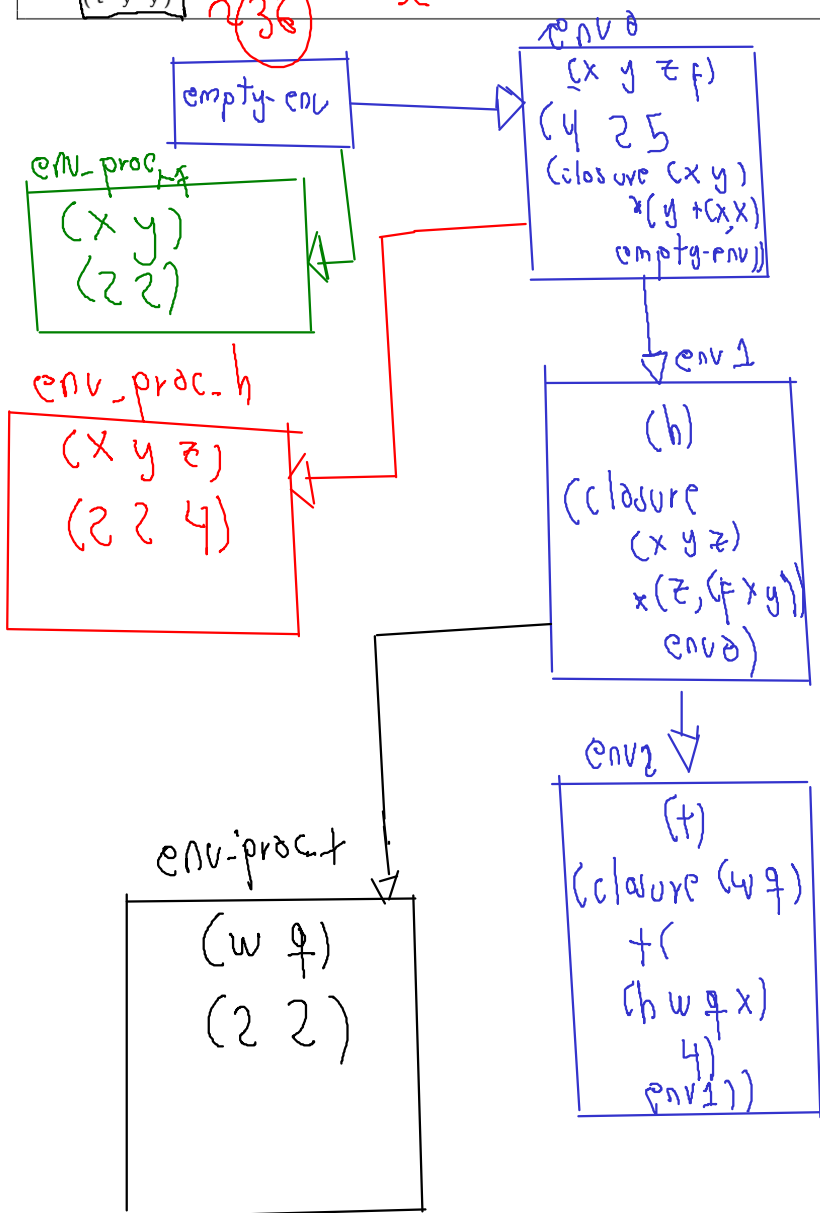
Sea el ambiente  $env_0$  con símbolos  $(x\ y\ z\ f)$  y valores  $(4\ 2\ 5\ (\text{closure}'(x\ y)\ *(y,\ +(x,x))\ \text{empty-env}))$  el ambiente inicial de computación. Evaluar:

```
let
  h = proc(x y z) *(z, (f x y))
in
  let
    t = proc(w q) +((h w q x), 4)
  in
    (t y y)
```

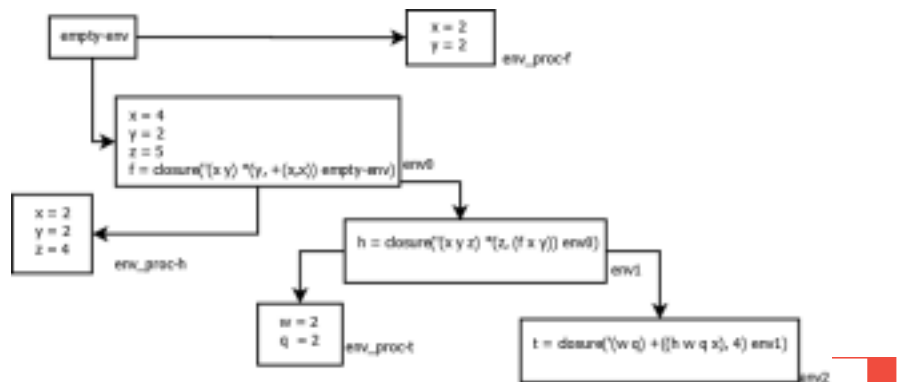
Sea el ambiente  $env_0$  con símbolos  $(x\ y\ z\ f)$  y valores  $(4\ 2\ 5)$  ( $\text{closure}'(x\ y)\ * (y, +(x,x))\ \text{empty-env}$ ) el ambiente inicial de computación. Evaluar:

```
let
  h = proc(x y z) * (z, (f x y))
in
  let
    t = proc(w q) + ((h w q x), 4)
  in
    (t y y)
```

Handwritten annotations:  $4$  above  $z$ ,  $8$  next to  $*(z, (f x y))$ ,  $32$  next to  $4$ ,  $36$  next to  $(t y y)$ .



# Ejemplos procedimientos



# Ejemplos procedimientos

```
(eval-expression
  <<let
    h = proc(x y z) *(z, (f x y))
  in
    let
      t = proc(w q) +((h w q x), 4)
    in
      (t y y)>>
  env0)
```

# Ejemplos procedimientos

```
= (eval-expression  
  <<let  
    t = proc(w q) +((h w q x), 4)  
    in  
      (t y y)>>  
  env1)
```

```
= (eval-expression  
  <<(t y y)>>  
  env2)
```

# Ejemplos procedimientos

```
= ((eval-expression  
    <<t>>  
    env2)  
   (eval-rands  
     '(<<y>> <<y>>  
       env2)))
```

```
= (apply-procedure  
   (closure '(w q) <<+((h w q x),4)>> env1)  
   '(2 2))
```



# Ejemplos procedimientos

```
= (eval-expression  
  <<+((h w q x),4)>>  
  env-proc-t)
```

```
= +((eval-expression  
      <<(h w q x)>>  
      env_proc-t),  
  4)
```

# Ejemplos procedimientos

```
= +(((eval-expression
      <<h>>
      env_proc-t)
      (eval-rands
        '(<<w>> <<q>> <<x>>))
      env_proc-t)),
4)
```

```
= +((apply-procedure
      (closure '(x y z) <<*(z, (f x y))>> env0)
      '(2 2 4)),
4)
```

# Ejemplos procedimientos

```
= +((eval-expression
      <<*(z, (f x y))>>
      env_proc-h),
  4)
```

```
= +(*((eval-expression
        <<z>>
        env_proc-h),
      (eval-expression
        <<(f x y)>>
        env_proc-h)),
  4)
```

# Ejemplos procedimientos

```
= +(* (4,
      (eval-expression
        <<(f x y)>>
        env_proc-h)),
  4)
```

```
= +(* (4,
      ((eval-expression
        <<f>>
        env_proc-h)
      (eval-rands
        '(<<x>> <<y>>)
        env_proc-h))),
  4)
```

# Ejemplos procedimientos

```
= +(* (4,
      (apply-procedure
       (closure '(x y) <<*(y, +(x,x))>> empty-env)
       '(2 2))),
  4)
```

```
= +(* (4,
      (eval-expression
       <<*(y, +(x,x))>>
       env-proc-f)),
  4)
```

# Ejemplos procedimientos

```
= +(* (4,
      *((eval-expression
          <<y>>
          env-proc-f),
      (eval-expression
          <<+(x,x)>>
          env-proc-f))),
4)
```

```
= +(* (4,
      *(2,
        +((eval-expression
            <<x>>
            env-proc-f),
          (eval-expression
            <<x>>
            env-proc-f))))),
4)
```



# Ejemplos procedimientos

$$= +(* (4, \\ \quad * (2, \\ \quad \quad + (2, 2) ) ) , \\ \quad 4)$$

$$= +(* (4, \\ \quad * (2, 4) ) , \\ \quad 4)$$

$$= +(* (4, 8) , \\ \quad 4)$$

$$= + (32, 4)$$

$$= 36$$

# Contenido

1 Ejemplos

2 Ejercicios



## Ejercicio 1:

Sea el ambiente  $env_0$  con símbolos  $(x\ y\ z)$  y valores  $(4\ 2\ 5)$  el ambiente inicial de computación. Evaluar:

```
let
  m = 5
  h = proc(x y) *(x, +(y,3))
in
  let
    g = proc(f t) -(m, (f t m))
  in
    (g h z)
```

## Ejercicio 2:

Sea el ambiente  $env_0$  con símbolos  $(x\ y\ z\ f)$  y valores  $(4\ 2\ 5\ (\text{closure}'(x\ y\ z)\ \text{if}\ -(x,y)\ \text{then}\ +(z,x)\ \text{else}\ 0\ \text{empty-env}))$  el ambiente inicial de computación. Evaluar:

```
let
  j = 12
  k = 11
  l = proc(a b) *(2, (f a b x))
in
  (l (l j k) y)
```