



Actividad - Fundamentos de lenguajes de programación

Carlos Andres Delgado S, Msc *

Marzo de 2019

Nombre: _____

Código: _____

- Debe mostrar el procedimiento realizado en cada uno de los puntos, no es válido únicamente mostrar la respuesta.
- Debe entregar un archivo en PDF de lo realizado, en este deben estar los procedimientos para cada punto.
- No se aceptan fotos de mala calidad en los archivos, se recomienda use los editores de ecuación que ofrecen los editores de texto. El archivo de entrega no puede pesar más de 25MB ya que será rechazado por el Campus Virtual. Si su punto no se puede ver claramente no será aceptado
- Debe entregar los interpretadores modificados, así mismo en comentarios explicar los cambios que se realizaron
- La fecha de entrega es el Domingo 31 de Marzo de 2019 a las 11:59am, hora de Colombia.
- Este trabajo ser realizado en parejas
- Este trabajo debe ser sustentado el 02 de Abril de 2019, la nota de la sustentación es individual y va entre 0 y 1. Esta nota se multiplica por la nota obtenida en esta actividad.

* carlos.andres.delgado@correounivalle.edu.co

En todos los casos suponga que el ambiente inicial es el ambiente vacío

- (15 puntos) Dibuje los ambientes de la expresión considerando paso por referencia e indique el valor de la expresión. Para la siguiente expresión.

```
let
  z = 4
  u = 5
  f = proc(x) begin set x = 3; +(x,2) end
  g = proc(x y) +(y, x)
in
  let
    j = proc(p) begin set p = +(p,2); *(2,p) end
    i = proc(p) begin set p = +(4,p); +(p,3) end
  in
    let
      t = (j z)
      p = (f z)
      k = (i z)
      l = (g z u)
      o = (j u)
      r = (f u)
    in
      +(t,p,k,l,o,r)
```

- (20 puntos) Utilizando inferencia de tipos indique el tipo de la expresión. Considere el ambiente inicial como vacío.

```
let
  x = 3
  y = 4

  f1 = proc(? a, ? b, ? c, ? d)
    if b then (a c d)
    else proc(? t, int w) *(t,w)

  f2 = proc(int n, int m)
    proc(int g, ? h) +(*(g,m), *(h,n))

  f3 = proc(int i, ? j) >(i,j)

in
  let
    g1 = (f1 f2 (f3 x y) +(y,3) *(x,2))
    g2 = (f2 x y)
  in
    (g1 (g2 x y) +(x,y))
```

Indique el valor para cada variable de tipo. Para cada expresión relacionada a continuación indique sus ecuaciones de tipo. Muestre el proceso que realiza para determinar el tipo de toda la expresión y de todas las variables de tipo.

| Variable de tipo | Expresión | Ecuaciones de de tipo |
|------------------|---|-----------------------|
| t_x | x | |
| t_y | y | |
| t_a | a | |
| t_b | b | |
| t_c | c | |
| t_t | t | |
| t_w | w | |
| t_d | d | |
| t_n | n | |
| t_m | m | |
| t_g | g | |
| t_h | h | |
| t_i | i | |
| t_j | j | |
| t_{f1} | proc(? a, ? b, ? c, ? d) if b then (a c d) else proc(? t, int w) *(t,w) | |
| t_1 | if b then (a c d) else proc(? t, int w) *(t,w) | |

| | | |
|----------|---|--|
| t_2 | (a c d) | |
| t_3 | proc(? t, int w) *(t,w) | |
| t_4 | *(t,w) | |
| t_{f2} | proc(int n, int m) proc(int g, ? h) +(*(g,m), *(h,n)) | |
| t_5 | proc(int g, ? h) +(*(g,m), *(h,n)) | |
| t_6 | +(*(g,m), *(h,n)) | |
| t_7 | *(g,m) | |
| t_8 | *(h,n) | |
| t_{f3} | proc(int i, ? j) >(i,j) | |
| t_9 | >(i,j) | |
| t_{g1} | (f1 f2 (f3 x y) +(y,3) *(x,2)) | |
| t_{10} | (f3 x y | |
| t_{11} | +(y,3) | |
| t_{12} | *(x,2) | |
| t_{g2} | (f2 x y) | |
| t | (g1 (g2 x y) +(x,y)) | |
| t_{13} | (g2 x y) | |
| t_{14} | +(x,y) | |

3. (30 puntos) Para la siguiente expresión

```

class c1 extends object
  field a
  field b
  method initialize (x,y)
    begin
      set a=+(10,x);
      set b=+(19,y);
      +(a,b)
    end
  method m1 () send self m2 (1)
  method m2 (n) *(n,+(a,b))

class c2 extends c1
  field b
  field c
  method initialize ()
    begin
      set b=15;
      set c=18;
      super initialize(b,c);
      send self m3(1)
    end
  method m2 (n) *(n,*(a,+(b,c)))
  method m3 (n) super m2(n)

class c3 extends c2
  method initialize() super initialize()
  method m2 (n) super m3(n)
  method m3 (n) -(c,b)
  method m4 (n) +(n,b)

class c4 extends c2
  method initialize() super initialize()
  method m2 (n) super m3(n)
  method m3 (n) -(c,b)
  method m4 (n) +(n,b)
  method m5 (o) +( send o m1(), send self m3(10))

let
  a = new c4()
  b = new c3()
  c = 4
  in
    let
      x = send a m5(b)
      y = send b m4(1)
      z = send b m2(4)
    in
      +(x,y,z)

```

- (15 puntos) Dibuje los ambientes generados para los 3 llamados de métodos para objetos simples
 - (15 puntos) Dibuje los ambientes generados para los 3 llamados de método para objetos planos
4. (15 puntos) Modifique el interpretador de chequeo de tipos para aceptar listas, cuya regla gramatical es:

```
<expresion> ::= "[" <expresion>*(,)" "
```

Ejemplo

```
[1, 2, 3, 4, 5]
```

Aplique la siguiente regla de tipos:

```
[T1 T2 T3 ... TN]
T1 = T2 = T3 = .. = TN
;;Es decir que todos los elementos de lista debe ser del mismo tipo
```

Explique con comentarios cómo realizó el proceso de crear las listas, de representarlas en Racket y de validar los tipos. Indique 3 expresiones de lista sin error de tipos y 3 expresiones de listas con error de tipo.

5. (10 puntos) Seleccione dos lenguajes de programación fuertemente tipados y dos débilmente tipados. De cinco ejemplos de expresiones en las cuales se muestre la diferencia entre estos dos lenguajes. No puede usar los ejemplos que se hicieron en el clase.
6. (15 puntos) Escriba expresiones de tipo que cumplan:
- $(int * (int \rightarrow bool) * (int \rightarrow (bool \rightarrow int))) \rightarrow (int \rightarrow bool)$
 - $(int * bool * (int * (int \rightarrow int) * int \rightarrow (bool \rightarrow int))) \rightarrow (int \rightarrow (bool \rightarrow int))$
 - $int * (bool \rightarrow (bool \rightarrow bool)) * (bool \rightarrow bool) \rightarrow (int * (bool \rightarrow (int \rightarrow bool))) \rightarrow (bool \rightarrow int)$