

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS



Lenguajes de Programación

Tarea 6: Paso de Parámetros

Johann Ramón Gordillo Guzmán

418046090

José Jhovan Gallardo Valdéz

310192815

Tarea presentada como parte del curso de **Lenguajes de Programación** impartido por la profesora **M.I. Karla Ramírez Pulido**.

27 de Abril del 2020

Link al código fuente: <https://github.com/JohannGordillo/>

1. Preguntas

1. Evalúa la siguiente expresión usando el intérprete para cajas visto en clase. Debes usar alcance estático y evaluación glotona. Mostrar el ambiente (*stack*) y memoria (*heap*) finales.

```
{with {{a {newbox 17}}
      {with {b {newbox 29}}
            {with {foo {fun {} {setbox a {openbox a}}}}
                  {with {foo {fun {} {setbox b {openbox a}}}}
                        {seqn {foo}
                            {+ {openbox a} {openbox b}}}}}}}}}
```

Respuesta.

Primero, construyamos el ambiente y la memoria:

		5	{fun {} {setbox b {openbox a}}}
		4	{fun {} {setbox a {openbox a}}}
foo	5	3	29
foo	4	2	3
b	2	1	17
a	0	0	1

Ahora, evaluamos la expresión del *with* más anidado, ésta es:

```
{seqn {foo}
      {+ {openbox a} {openbox b}}}}}
```

Por lo que, al ser una expresión *seqn*, primero vamos a evaluar {foo}. Al buscar *foo* en el Stack, obtenemos:

```
{fun {} {setbox b {openbox a}}}
```

Como *foo* no recibe ningún parámetro, no agregamos nada al Stack y únicamente evaluamos {setbox b {openbox a}}, lo que nos lleva a tener el Stack y Heap finales:

		5	{fun {} {setbox b {openbox a}}}
		4	{fun {} {setbox a {openbox a}}}
foo	5	3	17
foo	4	2	3
b	2	1	17
a	0	0	1

Finalmente, la expresión *seqn* regresa el resultado de evaluar $\{+ \{\text{openbox } a\} \{\text{openbox } b\}\}$.

Al evaluar $\{\text{openbox } a\}$, obtenemos 17.

Al evaluar $\{\text{openbox } b\}$, también obtenemos 17.

Así, tenemos lo siguiente:

```
{+ {openbox a} {openbox b}}
{+ 17 {openbox b}}
{+ 17 17}
34
```

∴ El resultado de evaluar la expresión es 34.

2. Dada la definición de la función **next-location** que genera nuevas direcciones de memoria, vista en clase, modificala para que no tenga efectos secundarios, es decir, que no dependa de ninguna variable externa. *Hint*: Modificala usando la técnica de *Store Passing Style*, puedes añadir los parámetros que necesites.

(define last-location -1) ;; al inicio del intérprete.

```
;; next-location: number
(define (next-location)
  (begin
    (set! last-location (add1 last-location))
    last-location))
```

Respuesta.

```
;; next-location: Store → number
(define (next-location sto)
  (type-case Store sto
    (mtSto () 0)
    (aSto (location value sto)
          (+ 1 location))))
```

Nota: Habrá que modificar un poco el intérprete visto en clase para que use esta implementación de la función **next-location**.

3. Evalúa la siguiente expresión usando el intérprete para cajas visto en clase. Debes usar alcance estático y evaluación glotona. Mostrar el ambiente (*stack*) y memoria (*heap*) finales. Con base en los resultados obtenidos, responde, ¿por qué no fue necesario añadir ambientes recursivos para evaluar esta expresión?

```
{with {suma {fun {n} {if0 n 0 {+ n {suma {- n 1}}}}}
      {suma 2}}
```

Respuesta.

Primero, construyamos el ambiente:

suma	0	0	{fun {n} {if0 n 0 {+ n {suma {- n 1}}}}}
------	---	---	--

Ahora, tenemos que evaluar la expresión del with:

```
{suma 2}
```

Es la aplicación de la función *suma* con el argumento 2. Usamos alcance estático, por lo que nuestro ambiente queda como:

suma	0	1	2
n	1	0	{fun {n} {if0 n 0 {+ n {suma {- n 1}}}}}

Luego, evaluaremos:

```
{if0 2 0 {+ 2 {suma {- 2 1}}}}
```

Esto es:

```
{+ 2 {suma 1}}
```

Por lo que ahora tenemos que evaluar {suma 1}.

Notese que *suma* no está libre, pues al usar Store-Passing style el heap es global y cada expresión tiene acceso a la memoria, así como a los cambios en ella producto de expresiones previas.

Así, obtenemos lo siguiente:

```
{+ 2 {+ 1 {suma 0}}}
```

Y análogamente, volvemos a hacer la llamada recursiva para finalmente tener la expresión:

```
{+ 2 {+ 1 0}}
```

Lo anterior se evalúa como sigue:

```
{+ 2 {+ 1 0}}
{+ 2 1}
3
```

∴ El resultado de evaluar la expresión es 3.

Usando Store-Passing style no es necesario que usemos ambientes recursivos, ya que el heap y sus cambios son pasados a través de cada expresión, dándoles acceso cuando les sea necesario. En efecto, el alcance del heap es dinámico en el sentido de que refleja el historial de cómputo.

4. Evalúa la siguiente expresión usando el paso de parámetros que se indica:

```
{with {{a 8}
      {b -8}
      {swap {fun {x y}
              {with {{tmp x}
                    {seqn {set x y}
                          {set y tmp}}}}}}
      {seqn {swap a b}
            {- a {+ b a}}}}
```

a) Paso de parámetros por valor.

Respuesta.

Vemos que en el cuerpo del with tenemos un seqn, que primero ejecuta {swap a b}.

swap es una función que recibe dos parámetros, y en la aplicación le estamos pasando a a, b , por lo que:

Parámetros formales: x, y

Parámetros reales: a, b

Construyamos el ambiente y la memoria:

swap	2
b	1
a	0

2	(fun (x y) fun-body)
1	-8
0	8

Entonces, como estamos usando paso de parámetros por valor, $temp = x$ es una copia del valor de $a = 8$.

Análogamente, al hacer $x = y$, x recibe una copia del valor del parámetro real $b = -8$.

Posteriormente, hacemos $y = temp$, por lo que $y = 8$.

Sin embargo, el intercambio de valores se llevó a cabo dentro de la función.

Finalmente, la segunda expresión de *seqn* es $\{- a \{+ b a\}\}$, y al buscar los valores en el ambiente obtenemos que a está en la dirección de memoria 0, y b está en la dirección de memoria 1. Al buscar en el heap estos valores, obtenemos:

$$\begin{cases} a = 8 \\ b = -8 \end{cases}$$

Y al evaluar la expresión:

$$\{- a \{+ b a\}\} = \{- 8 \{+ -8 8\}\} = \{- 8 0\} = 8$$

\therefore El resultado de evaluar la expresión usando **paso de parámetros por valor** es 8.

b) Paso de parámetros por referencia.

Respuesta.

Vemos que en el cuerpo del with tenemos un *seqn*, que primero ejecuta $\{\text{swap } a \ b\}$.

swap es una función que recibe dos parámetros, y en la aplicaciones le estamos pasando a a, b , por lo que:

Parámetros formales: x, y

Parámetros reales: a, b

Construyamos el ambiente y la memoria:

swap	2
b	1
a	0

2	(fun (x y) fun-body)
1	-8
0	8

Entonces, como estamos usando paso de parámetros por referencia, $temp = x$ pasa la dirección en memoria de a , que es 0. En esta dirección encontramos el valor de 8, por lo que $temp = 8$.

Análogamente, al hacer *set* con $x = y$, pasa la dirección en memoria de b , que es 1. En esta dirección encontramos el valor de -8. Por lo tanto $x = -8$.

Posteriormente, hacemos $y = temp$, por lo que $y = 8$.

Sin embargo, el intercambio de valores se llevó en el ambiente global, de manera que tenemos:

swap	2	2	(fun (x y) fun-body)
b	1	1	8
a	0	0	-8

Finalmente, la segunda expresión de *seqn* es $\{- a \{+ b a\}\}$, y al buscar los valores en el ambiente obtenemos que a está en la dirección de memoria 0, y b está en la dirección de memoria 1. Al buscar en el heap estos valores, obtenemos:

$$\begin{cases} a = -8 \\ b = 8 \end{cases}$$

Y al evaluar la expresión:

$$\{- a \{+ b a\}\} = \{- -8 \{+ 8 -8\}\} = \{- -8 0\} = -8$$

∴ El resultado de evaluar la expresión usando **paso de parámetros por referencia** es -8.

5. Evalúa la siguiente expresión usando el paso de parámetros que se indica.

$$\begin{aligned} &\{\text{with } \{\{a \ 1\} \\ &\quad \{\text{foo } \{\text{fun } \{x\} \\ &\quad \quad \{\text{seqn } \{\text{set } a \ \{+ \ \{* \ x \ -2\} \ a\}\} \\ &\quad \quad \quad a\}\}\}\} \\ &\quad \{\{\text{fun } \{y\} \ \{+ \ y \ \{- \ y \ y\}\} \ \{\text{foo } 3\}\}\}\} \end{aligned}$$

Respuesta.

a) Paso de parámetros por nombre.

Respuesta.

Evaluamos la expresión del with más anidado: $\{\{\text{fun } \{y\} \ \{+ \ y \ \{- \ y \ y\}\} \ \{\text{foo } 3\}\}\}$

Recordemos que el lado izquierdo de la aplicación de una función es un punto estricto, por lo que lo evaluamos y obtenemos: $\{+ \ \{\text{foo } 3\} \ \{- \ \{\text{foo } 3\} \ \{\text{foo } 3\}\}\}$

Los operandos son también puntos estrictos, por lo que primero evaluaremos el lado izquierdo: $\{\text{foo } 3\}$.

Nuestro ambiente es:

foo	1
y	2
a	0

2	{foo 3}
1	{fun {x} {seqn ...}}
0	1

Luego, hacemos que a sea $-6 + 1 = -5$. Por lo que el ambiente será:

foo	1
y	2
a	0

2	{foo 3}
1	{fun {x} {seqn ...}}
0	-5

Sin embargo, al usar paso de parámetros por nombre, estamos usando paso por referencia, por lo que al volver a evaluar {foo 3}, obtenemos el siguiente ambiente:

foo	1
y	2
a	0

2	{foo 3}
1	{fun {x} {seqn ...}}
0	-11

Lo anterior debido a que el valor de a cambió en el ambiente a -5 , y al sumarle -6 obtenemos -11 que también cambiará en el ambiente.

Finalmente, evaluaremos {foo 3} por última ocasión, de manera que el ambiente final es:

foo	1
y	2
a	0

2	{foo 3}
1	{fun {x} {seqn ...}}
0	-5

Sin embargo, al usar paso de parámetros por nombre, estamos usando paso por referencia, por lo que al volver a evaluar {foo 3}, obtenemos el siguiente ambiente:

foo	1
y	2
a	0

2	{foo 3}
1	{fun {x} {seqn ...}}
0	-17

Ahora procedemos a evaluar la expresión completa:

```
{+ {foo 3} {- {foo 3} {foo 3}}}  
{+ -5 {- {foo 3} {foo 3}}}  
{+ -5 {- -11 {foo 3}}}  
{+ -5 {- -11 -17}}  
{+ -5 6}  
1
```

b) Paso de parámetros por necesidad.

Respuesta.

Evaluamos la expresión del with más anidado: $\{\{\text{fun } \{y\} \{+ y \{- y y\}\} \{\text{foo } 3\}\}\}$

Recordemos que el lado izquierdo de la aplicación de una función es un punto estricto, por lo que lo evaluamos y obtenemos: $\{+ \{\text{foo } 3\} \{- \{\text{foo } 3\} \{\text{foo } 3\}\}\}$

Los operandos son también puntos estrictos, por lo que primero evaluaremos el lado izquierdo: $\{\text{foo } 3\}$. Nuestro ambiente es:

foo	1
y	2
a	0

2	$\{\text{foo } 3\}$
1	$\{\text{fun } \{x\} \{\text{seqn } \dots\}\}$
0	1

Luego, hacemos que a sea $-6 + 1 = -5$. Por lo que el ambiente será:

foo	1
y	2
a	0

2	$\{\text{foo } 3\}$
1	$\{\text{fun } \{x\} \{\text{seqn } \dots\}\}$
0	-5

$\therefore \{\text{foo } 3\} = -5$, y como estamos usando el paso de parámetros por necesidad, el resultado se mantiene, por lo que nuestra expresión a evaluar queda como:

```
{+ -5 {- -5 -5}}  
{+ -5 0}  
-5
```

6. Evalúa la siguiente expresión usando el paso de parámetros por referencia-regreso.

$$\{\text{with } \{\{b \ 2\} \\ \{f \ \{\text{fun } \{x\} \ \{\text{seqn } \{\text{set } x \ 4\} \\ \{+ \ x \ b\}\}\}\}\} \\ \{+ \ \{f \ b\} \ b\}\}$$

Respuesta.

Primero evaluaremos $\{f \ b\}$ y luego buscaremos en el ambiente el valor de b .

Construyamos el ambiente y el heap:

f	1	1	(fun (x) fun-body)
b	0	0	2

Ahora, evaluemos $\{f \ b\}$. El parámetro formal es x y el parámetro real es b . Recordemos que en el paso de parámetros por referencia-regreso se recibe una copia del valor de b .

En el cuerpo de la función tenemos un *seqn*, que primero ejecuta $\{\text{set } x \ 4\}$, por lo que $x = 4$, y al salir el registro de activación generado por la función, se cambiará en memoria el valor de b por 4, pero no mientras el registro de activación de f este activo.

Luego, regresaremos $\{+ \ x \ b\}$. Como el registro de activación de la función sigue activo, el valor de b no ha cambiado, por lo que $b = 2$.

Por lo tanto $\{+ \ x \ b\} = \{+ \ 4 \ 2\} = 6$. Este valor será regresado por la aplicación de la función.

Ahora, al ser paso de parámetros por referencia-regreso, ya que ha salido el registro de activación de la función de la pila de llamadas, el valor en la localidad de memoria asociada a b cambiará a 4. Así, el ambiente y el heap se ven como:

f	1	1	(fun (x) fun-body)
b	0	0	4

Ahora, buscamos b en el ambiente, lo que nos da la localidad en memoria 0. Al buscar esta dirección en el heap (operación de complejidad $O(1)$), obtenemos que $b = 4$.

Finalmente, hacemos la suma:

$\{+ \ \{f \ b\} \ b\} = \{+ \ 6 \ 4\} = 10$.

∴ El resultado de evaluar la expresión usando **paso de parámetros por referencia-regreso** es 10.

2. Bibliografía

- Ramírez, K. (2020).
Notas del curso de Lenguajes de Programación.
Facultad de Ciencias - UNAM
Ciudad de México, México.
- Krishnamurthi, S. (2017).
Programming Languages: Application and Interpretation.
Estados Unidos de América.