Lenguajes de Programación
Tarea 5

Karla Ramírez Pulido Alan Alexis Martínez López

Semestre 2023-2 **Fecha de inicio:** 10 de abril 2023
Facultad de Ciencias UNAM **Fecha de entrega:** 15 de abril 2023

Integrantes:
Dania Paula Gongora Ramírez
Salgado Tirado Diana Laura

Instrucciones

Resolver los siguientes ejercicios de forma clara y ordenada de acuerdo a los lineamientos de entrega de tareas disponibles en la página del curso. Esta tarea se puede hacer a lo más en equipos de 2 integrantes.

Ejercicios

1. **Utiliza el paso de parámetros que se indica para evaluar la siguiente expresión.**

```
{ 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.
Al llamar al seqn necesitaremos nuestro ambiente el cual es el siguiente :

Ya que se llama de la siguiente forma: {swap a b}

swap	*{*fun *{*x y*}* *{* with *{ {* tmp x *} }* *{*seqn *{*set x y *}* *{*set y tmp *} } }*	**0x12**
b	-8	**0x11**
a	8	**0x10**

Tenemos como parámetros formales x, y, los parámetros reales a, b. Si evaluamos la función swap copiando el valor en la función:

{{fun {x y}
{ with { { tmp x } }
{seqn {set x y }
{set y tmp } } } 8 -8}

con x = 8 , y = -8

1.-Comenzamos con la línea { with { { tmp 8} }

Por lo que tmp = 8.

2.-Continuamos en la línea {seqn {set x y }

Por lo que x = y , es decir x = - 8

3.-Seguimos en la línea *{*set y tmp *}*

Por lo que y = 8

Notemos que en ningún momento cambiamos las referencias de a y b , por lo tanto al continuar y aplicar la siguiente línea del seqn, la cual es *{* - a *{* + b a *} }*

Por lo que utilizando el ambiente al evaluar tenemos *{* - 8 *{* + (-8) 8 *} }={* - 8 *0 }* = - *8*

Por lo que la evaluación de la expresión con parámetros por valor es -8.

b. Paso de parámetros por referencia.

Primero veamos como al llamar al seqn necesitaremos nuestro ambiente el cual es el siguiente.

Ya que se llama de la siguiente forma: {swap a b}

swap	**0x12**
b -8	**0x11**
a 8	**0x10**

Tenemos como parámetros reales x, y, los parámetros reales a, b. Si evaluamos la función swap:

1.-Comenzamos con la línea { with { { tmp x } }
Por lo que tmp = a donde a pasa su dirección, la cual es 0x10, entonces tmp = 8.

2.-Continuamos en la línea {seqn {set x y }
Por lo que a = b, b pasa la dirección 0x11, entonces a = -8

3.-Seguimos en la línea *{*set y tmp *}*
Por lo que b = tmp donde sabemos que tmp es 8, entonces b = 8

Por lo que ahora nuestro ambiente es:

swap	**0x12**
b 8	**0x11**
a -8	**0x10**

4.- Seguimos con la línea del seqn, la cual es *{* - a *{* + b a *} }*
Por lo que utilizando el ambiente al evaluar tenemos *{* - -8 *{* + 8 -8 *} }={* - -8 *0 }* = *-8*

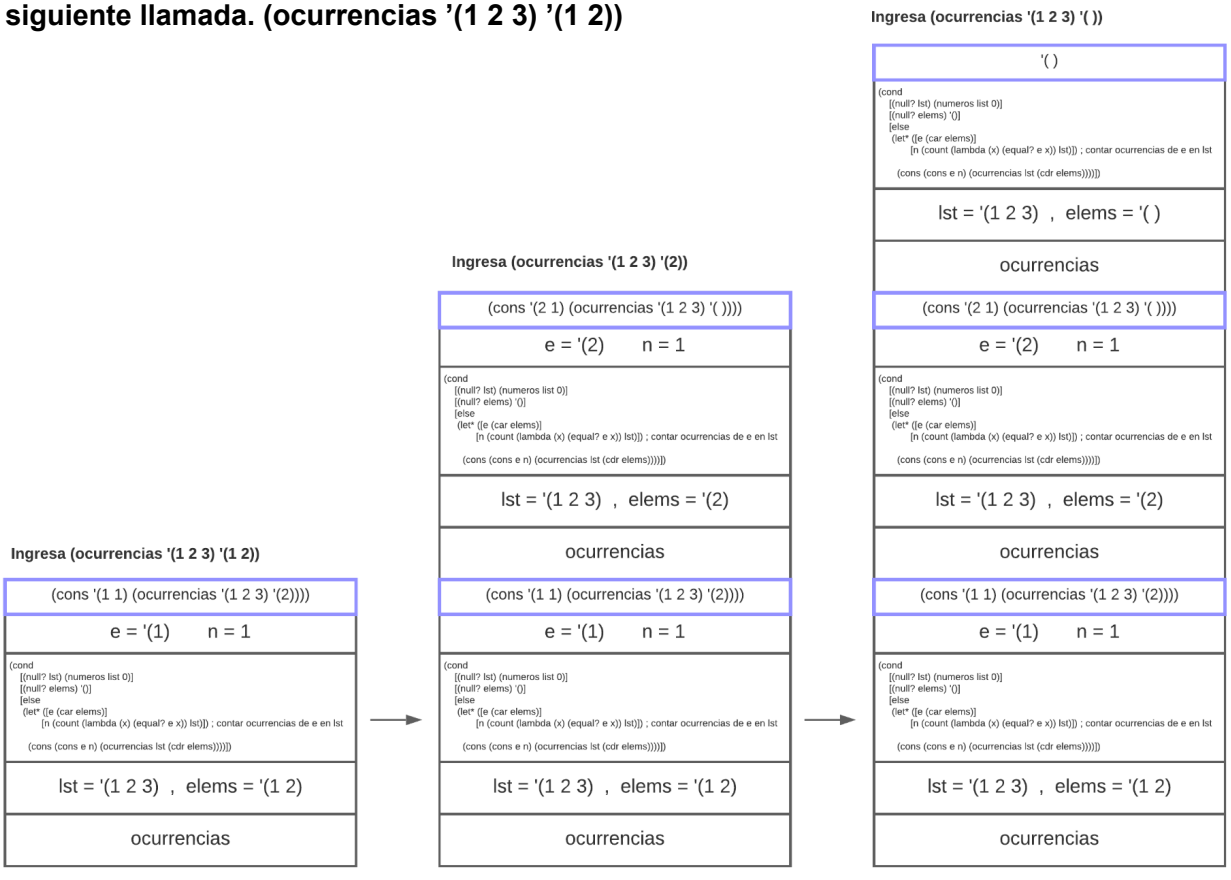
Por lo que la evaluación de la expresión con parámetros por referencia es -8.

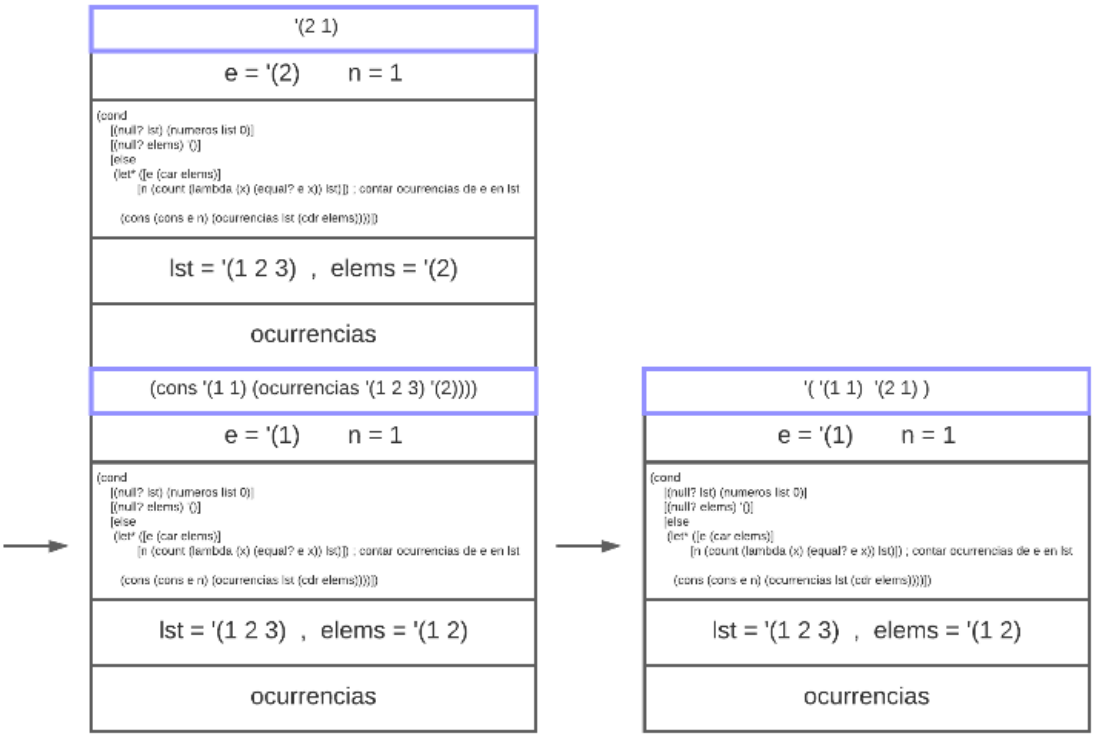
2. Define la función recursiva ocurrencias que recibe dos listas y devuelve una lista de parejas, en donde cada pareja contiene en su parte izquierda un elemento de la segunda lista y en su parte derecha el número de veces que aparece dicho elemento en la primera lista. Por ejemplo:

>(ocurrencias ’(3 5 8 5 2 1 2 2 0 3) ’(2 3 6))
’((2 . 3) (3 . 2) (6 . 0))

```
#lang plai

(define (ocurrencias lst elems)
  (cond
    [(null? lst) (numeros list 0)]
    [(null? elems) '()]

    [else
     (let* ([e (car elems)]
            [n (count (lambda (x) (equal? e x)) lst)]) ; contar ocurrencias de e en lst
       (cons (cons e n) (ocurrencias lst (cdr elems))))]))


;;Función auxiliar para poner elementos con su pareja n
(define (numeros lst n)
  (cond
    [(null? lst) '()]
    [else (cons (cons (cons (car lst) n) '()) (numeros (cdr lst) n))]

   ))
```

3. A partir del Ejercicio 2, muestra los registros de activación generados por la función con la siguiente llamada. (ocurrencias ’(1 2 3) ’(1 2))



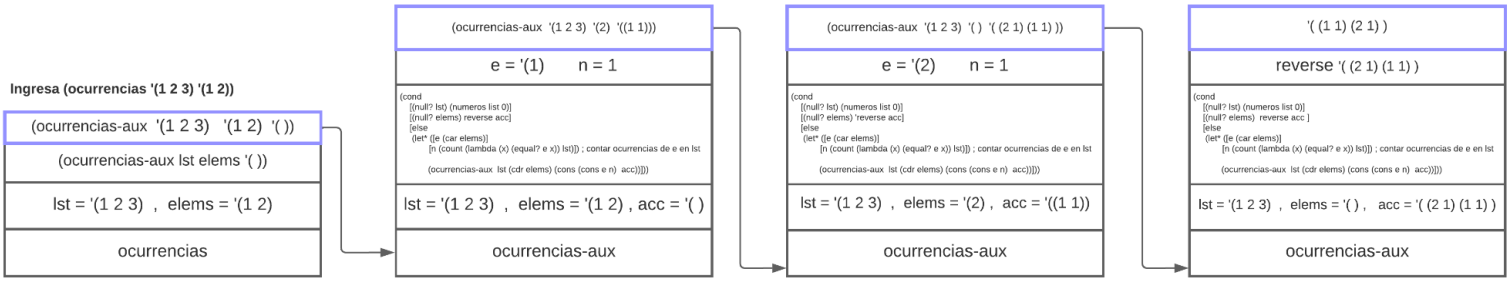


Por lo tanto el resultado de **(ocurrencias '(1 2 3) '(1 2))** es '((1 1) (2 1))

4. Usando recursión de cola optimiza la función del Ejercicio 2. Toda función auxiliar ocupada debe ser optimizada.

```
#lang plai
(define (ocurrencias lst elems)
  (define (ocurrencias-aux lst elems acc)
    (cond
      [(null? elems) (reverse acc)]
      [(null? lst) (numeros list 0)]

      [else       (let* ([e (car elems)]
                  [n (count (lambda (x) (equal? e x)) lst)])
          (ocurrencias-aux lst (cdr elems) (cons (cons e n) acc)))]))
  (ocurrencias-aux lst elems '()))

;;Función auxiliar para poner elementos con su pareja n
(define (numeros lst n)
  (define (numeros-aux lst n acc)
    (cond
      [(null? lst) (reverse acc)]
      [else       (numeros-aux (cdr lst) n (cons (cons (car lst) n) acc))]))
  (numeros-aux lst n '()))
```

5. A partir del Ejercicio 4, muestra los registros de activación generados por la función con la siguiente llamada. (ocurrencias ’(1 2 3) ’(1 2))



Por lo tanto el resultado de **(ocurrencias ’(1 2 3) ’(1 2))** es ‘((1 1) (2 1))