



Taller 3 - Tipos y objetos

Fundamentos de Lenguajes Programación

Carlos Andres Delgado S, Ing *

Octubre de 2015

1. (**2pts.**) Descargue los analizadores de chequeo de tipos (chequeo e inferencia) con dos ejemplos para cada uno, uno con errores de tipo y el otro sin errores.
2. (**2pts.**) Descargue los interpretadores de objetos (simples y planos) y muestre cuatro ejemplos de declaración de objetos simples y planos (cuatro para cada uno).
3. (**4pts.**) Extienda el chequeo de tipos para las expresiones tipo *if-exp*, si la expresión *test-exp* no es un booleano, los tipos de las expresiones **true-exp** y **false-exp** no son chequeados.
4. (**6pts.**) Agregue el tipo de dato *par*. Este tipo de dato consiste en dos valores x_1 y x_2 . Agregue las siguientes producciones:

Tipo := pair of Type * Type
 pair-type (ty1 ty2)

Expression := newpair (Expression , Expression)
 pair-exp (exp1 exp2)

Expression := unpair (Identifier Identifier) = Expression
 unpair-exp (var1 var2 pair)

Ejemplos:

```
bool t1
int t2
s1 = newpair (t1 , t2)
```

```
bool t3
int t4
unpair(t3 t4)= s1
```

En el primer caso los contenidos de t1 y t2 son agrupados en s1. En el segundo ejemplo en t3 se almacena el primer elemento de s1 y en t4 el segundo elemento.

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

5. (4pts.) ¿Que está mal en la siguiente expresión?

```
letrec
? even(? odd) =
  proc (? x)
    if zero?(x) then 1 else (odd -(x,1))
  in
    letrec
      ? odd(bool x) =
        if zero?(x) then 0 else ((even odd) -(x,1))
      in
        (odd 13)
```

Realice un análisis con la ayuda del interpretador de inferencia de tipos.

6. (6pts.) Implemente en el interpretador de objetos un objeto tipo cola con los siguientes requerimientos
- a) Un método **empty?**, retorna `#t` si la cola está vacía en caso contrario retorna `#f`
 - b) Un método **encolar**, recibe un elemento, el cual es encolado.
 - c) Un método **descolar**, retorna y elimina el primer elemento de la cola, si la cola está vacía debe retornar un error
 - d) Un método **size** donde retorna el tamaño de la cola.
7. (6pts.) Implemente en el interpretador de objetos un objeto tipo pila, con funciones pop, push y size. **pop** elimina el primer elemento de la pila y lo retorna. **push** ingresa un elemento a la primera posición de la pila y **size** retorna el tamaño de la pila.

Aclaraciones

1. El taller es en grupos de mínimo (2) estudiantes y máximo tres (3) estudiantes.
2. La solución del taller debe ser subida al campus virtual en la fecha especificada por el docente
3. Las primeras líneas de cada archivo deben contener los nombres y códigos de los estudiantes.
4. En ese mismo archivo, vendrán comentados los procedimientos que llevan al código de la declaración, las operaciones, la expresión BNF de las estructuras que se están utilizando, y algunos ejemplos de prueba. Por ejemplo, si se pidiera construir el procedimiento remove-first, deberá existir un código como:

```
;; <lista-de-simbolos> := ({<exp-simbolo>}*)
;; <exp-simbolo> := <simbolo> | <lista-de-simbolos>
;;
;; remove-first : simbolo * lista-de-simbolos -> lista-de-simbolos
;;
;; Proposito:
;; Procedimiento que remueve la primera ocurrencia de un simbolo
;; en una lista de simbolos.
;;

(define remove-first
  (lambda (s los)
    (if (null? los)
        '()
        (if (eqv? (car los) s)
            (cdr los)
            (cons (car los)
                  (remove-first s (cdr los)))))))
```

```
(remove-first s (cdr los))))))
```

```
;; Pruebas
```

```
(remove-first 'a '(a b c))
```

```
(remove-first 'b '(e f g))
```

```
(remove-first 'a4 '(c1 a4 c1 a4))
```

```
(remove-first 'x '())
```