

## Lenguajes de Programación (CC41A) - Primavera 2009

Clase Auxiliar 5

Profesor: Tomás Barros

Auxiliar: Víctor Ramiro

1. Considere el siguiente programa:

```
{with {fact {fun {n} {if0 n
                        1
                        {* n {fact {- n 1}}}}}}
{fact 10}}
```

- a) De el equivalente de este programa *sin with*, usando la equivalencia entre *with* y otras expresiones del lenguaje.
- b) Describa su evaluación en un interprete con *substitución explícita* (es decir, usando *subst*).
- c) Describa su evaluación en un interprete con *substitución diferida* (es decir, usando un ambiente).

## 2. Recursión

a) ¿Cuál es el resultado (en Scheme or RCFAE) de esas expresiones?

- `(let ((x x)) x)`
- `(letrec ((x x)) x)`

Explique.

b) Defina, en Scheme, la función `map`.

- Es su definición recursiva por la cola?
- Si no, definala en forma recursiva por la cola.
- ¿Que ventaja tiene esa definición por sobre la definición recursiva normal? Explique.

3. Considere un lenguaje con la siguiente gramática:

```
<expr> ::= <id> | <num>
          | (+ <expr> <expr>)
          | (if <expr> <expr> <expr>)
          | (lambda (<id>) <body>)
          | (<expr> <expr>)
```

Defina en Scheme la función **free-vars** que retorna la lista de variables libres de una expresión dada. No se olvide de partir escribiendo al menos un test por cada tipo de expresión.

4. Describa la evaluación del siguiente programa en el lenguaje VBCFAE detallando en cada paso qué pasa con el Environment y con el Store. ¿Cuál es el resultado obtenido?

Sea claro al describir la evolución Environment/Store. Para ello **DEBE** usar diagramas.

```
{with {{x {box 100}}
      {y 20}}
 {with {{f {fun {z} {+ z {+ {unbox x} y}}}}}
      {seqn
        {set-box! x 200}
        {set! y 30}
        {with {{y 5}}
          {f 1}}}}}}
```