

Control 3 – Lenguajes de Programación

Departamento de Ciencias de la Computación

Universidad de Chile

Profesor: Éric Tanter

5 de Julio del 2010

2 horas / 2 puntos por pregunta

1. Considere el siguiente intérprete de un lenguaje con estructuras de datos mutables (box), escrito usando *store-passing style* :

```
(define (interp expr env store)
  (type-case Expr expr
    ...
    [app (fun-expr arg-expr)
      (type-case Value*Store (interp fun-expr env store)
        [v*s (fun-value fun-store)
          (let ((arg-value (interp arg-expr env fun-store))
                (new-loc (next-location fun-store)))
            (interp (closureV-body fun-value)
                    (aEnv (closureV-param fun-value)
                          new-loc
                          (closureV-env fun-value))
                    (aSto new-loc
                          arg-value
                          fun-store))) ...)
```

- a) (0.3) ¿Qué problema tiene este intérprete?
- b) (0.5) Escriba un programa que ilustre que la semántica del `app` es errónea.
- c) (0.5) Modifique el intérprete para que funcione adecuadamente.
- d) (0.7) ¿Que semántica de paso de parametros implementa ese interprete? Explique, y detalle lo que tendría que cambiar para obtener otra semántica.

2. a) (0.5) ¿Cual es el resultado del siguiente programa? Explique paso a paso.
- ```
(let ((x 1)
 (y 2))
 (set! (if (< x y) x y) 10)
 (cons x y))
```
- b) (0.75) Provee un mecanismo de asignación condicional, y uselo en el programa anterior.
- c) (0.75) Describa los dos tipos de problemas que pueden ocurrir con un sistema de macro no higienico. Explique cual de los dos podría ocurrir con su definición anterior.
3. a) (0.5) En un sistema de objetos simples, sin delegación ni clases, cual es el alcance (scope) del identificador `self`? ¿porqué cambia al agregar delegación o clases? Ilustre.
- b) (0.8) Durante el semestre, hemos construidos interpretes en el lenguaje PLAI Scheme, usando `define-type/typecase`. ¿Son esos interpretes orientado a objetos? Defina un interprete para el AE usando objetos simples<sup>1</sup>, y discute (con ejemplos) las ventajas y desventajas de esa nueva implementación, en comparación con la otra. (*no necesitan definir el parser*)
- c) (0.7) En un lenguaje con prototipos, ¿como se puede simular clases? (de un ejemplo en el sistema visto en el curso<sup>2</sup>). Recíprocamente, ¿cómo se simulan prototipos con clases? (de un ejemplo en cualquier lenguaje con clases)

---

<sup>1</sup>(OBJECT ((field f v) ...) ((method m (x ...) body) ...))

<sup>2</sup>(OBJECT parent ((field f v) ...) ((method m (x ...) body) ...))