Tarea 2 CC41A Lenguajes de Programación
4 de mayo 2004
Plazo: 2 semanas (inmodificable) - Individual
No se aceptan tareas fuera de plazo

1. Objetivos

Esta tarea persigue construir un intérprete de MiScheme en Scheme, de modo de extender el intérprete visto en clases. Básicamente muestra el dominio de la implementación de variables léxicas y dinámicas así como de la interpretación de las invocaciones funcionales.

2. Extensiones propuestas

2.1. Variables Léxicas y Dinámicas

Se parte de un intérprete de alcance léxico normal y se pide agregar la posibilidad de tener variables dinámicas creadas con una forma especial:

```
(dyn-let ((var val) (var val) ...)
       aqui dentro var existe y se busca en el environment dinámico
```

Si existe una variable dinámica en tiempo de ejecución, debe primar sobre la versión léxica.

Ejemplo de uso:

```
(define f (evaluate '(lambda(b) (list a b)) '((a . 1))))
(evaluate '(f 2) '())
-> (1 2)
(evaluate '(dyn-let ((a 3)) (f 2)) ())
-> (3 2)
```

2.2. Números como funciones

La idea es aceptar expresiones con un número en posición funcional, e interpretarlos como un sub-índice de lista. Los negativos retornan la sublista final y los positivos el elemento en la posición dada:

```
(2 '(a b c d e))
-> c
(-2 '(a b c d e))
-> (d e)
```

2.3. Maps automáticos

La idea es aceptar listas de valores funcionales en posición funcional, y eso aplica cada función a los argumentos y retorna la lista de resultados. Ejemplo:

```
((+ *) (1 2 3 4))
-> (10 24)
```

3. Implementación

Ustedes deben entregar el intérprete completo en una función evaluate como la vista en clases.

4. Datos

Se les dejó en la página web del curso los evaluadores vistos en clases.

5. Entrega

No se aceptan tareas fuera de plazo. Una tarea que tiene las variables léxicas y dinámicas tiene un 4.0. Las otras funcionalidades suben 1.5 puntos cada una.