Listas y Recursividad en LISP

LISP: Lists and recursion

Sebastián Molina Loaiza
Ingeniería de Sistemas y Computación
Universidad Tecnológica de Pereira, Pereira – Colombia
semolina@utp.edu.co

Resumen **- En el presente artículo se analizará la creación de funciones en LISP para resolver problemas que contengan listas y recursividad.**

Palabras clave **– LISP, recursividad, listas, funciones.**

Abstract- **This article has been written with the purpose of illustrating the implementation of different lists functions on LISP language.**

Key Word- **LISP, functions, lists, recursion.**

I. INTRODUCCIÓN

Se implementarán y analizarán la construcción de diferentes funciones en lenguaje LISP a fin de ilustrar el funcionamiento de las listas y la estructura del lenguaje.

II. FUNCIÓN MIEMBRO DE LISTA

Se realizará el desarrollo de una función que detecta si un elemento hace parte de una lista y sí existe, devuelve la lista a partir del elemento indicado.

```
(defun miembro (objeto lista)
    (if (null lista)
        nil
        (if (eql (car lista) objeto)
            lista
            (miembro objeto (cdr lista))
        )
    )
)
```

En primer lugar, se define la función miembro la cual recibe un elemento y una lista como parámetros. Si la lista está vacía, se retorna null. De lo contrario se compara el elemento con la cabeza de la lista, si el elemento existe, se retorna la lista a partir de dicha posición, de modo contrario se llama a la función miembro con el mismo elemento y la cola de la lista.

III. FUNCIÓN LONGITUD DE LISTA

LISP nos provee de la función **length** la cual recibe una lista como parámetro y nos retorna la cantidad de elementos que contiene, es decir, su longitud.

```
(print (length '(a b c d e)))
; RESULTADO
; 5
```

A modo de práctica, se implemento la función **longitud**, que simula y tiene la misma funcionalidad de **length.**

```
(defun longitud (lista)
    (cond ((null lista) 0)
          (t (+ (longitud (cdr lista)) 1))))

 (print (longitud '(a b c d e f g h)))
; RESULTADOS
; 8
```

La función **longitud** toma como caso base la lista vacía, en cuyo caso la longitud vale cero, luego, empieza a llamarse recursivamente con la cola de la lista, sumando 1 por cada iteración, al final se obtendrá la longitud de la lista.

Fecha de Recepción: 26 de Enero de 2012
Fecha de Aceptación: 18 de Abril de 2012

IV. FUNCIÓN TÉRMINO N DE UNA LISTA

LISP provee la función **nth** la cual nos permite extraer el enésimo elemento de una lista. Se define la función **elemento-n** para simular el funcionamiento de nth.

```
(defun termino-n (n lista)
    (cond ((zerop n) (car lista))
          ; La función zerop verifica si n es cero
          (t (termino-n (- n 1) (cdr lista)))))

 (print (termino-n 5 '(a b c d e f g h i j k l)))
; RESULTADOS
; F
```

La función **termino-n** toma como caso base cuando de desea obtener el término 0, en cuyo caso la respuesta sería la cabeza de la lista, de lo contrario empezará a llamarse con el término decrementado en 1 y la cola de la lista.

V. CONCLUSIONES

LISP es un lenguaje con un campo de aplicación muy amplio que nos permite analizar las funciones orientadas a la inteligencia artificial a fin de entender a fondo su comportamiento.

VI. AGRADECIMIENTOS

Agradezco al material subido por el ingeniero Gilberto Vargas Cano quien mediante la plataforma **redesep** brindó la documentación necesaria para la realización de este documento.

VII. REFERENCIAS

[1] http://www.redesep.com/materias/blanda/index.php