 

INSTITUTO POLITECNICO NACIONAL

ESCUELA SUPERIOR DE COMPUTO

JESUS ALBERTO DORIAN LUNA MARTINEZ

Tarea 1. Investigación Lips

ROBERTO TECLA PARRA

3CM3

LIPS

Es un lenguaje que muestra la información estructurada en listas en las que se pueden gestionar la información que estas contienen. De ahí el nombre que se le dió a este lenguaje, Lisp (List-Processing), porque fue creado principalmente para el procesamiento de listas. Es un lenguaje funcional que se apoya en la utilización de funciones matemáticas para el control de los datos. Pero el elemento fundamental en el Lisp es la lista. Y desde el punto de vista más amplio del término. Cada función del lisp y cada programa que se genera con él vienen dado en forma de lista. Por esta razón los datos no se pueden diferenciar sintácticamente de los programas.

A este tipo de lenguaje se les denomina aplicativos o funcionales porque se basan en la aplicación de funciones a los datos. El lisp diferencia dos tipos de elementos básicos: El átomo, datos elementales de varios tipos como números, símbolos, caracteres y cadenas de caracteres. Y las Listas, entre las que podemos nombrar a una en especial, la lista “nil”, que es una lista nula que no tiene ningún elemento.

Ejemplo de átomos

* Hola
* Casa
* Mientras
* Uno34
* F4fg5

Ejemplo de No átomos

* 5456dgfv: Comienza con dígito.
* Ab cd : Incluye un espacio entre medio.
* %bc  : No comienza con una letra.
* A5.)  : Incluye caracteres que no son ni letras ni dígitos.

Ejemplos de Listas

* (ESTA ES UNA LISTA)
* (ESTALISTAESDISTINTAALAANTERIOR)
* (ESTA LISTA (TAMBIEN) ES DISTINTA)
* ((ESTA ES OTRA) (POSIBILIDAD DE LISTA))

|  |  |  |
| --- | --- | --- |
| **Lista** | **Número de términos** | **Términos** |
| (HOLA) | 1 | HOLA |
| (ESTA ES UNA LISTA) | 4 | ESTA, ES, UNA, LISTA |
| ((AB T56) HOLA ()) | 3 | (AB T56), HOLA, () |

LISP sigue una filosofía de tratamiento no-destructivo de los parámetros, de modo que la mayoría de las funciones devuelven un lista resultado de efectuar alguna transformación sobre la que recibieron, pero sin alterar esta última.

Una de las razones por las que el Lisp esta especialmente dotado para la programación en inteligencia artificial (IA), es precisamente, porque su código y todos los datos tienen la misma estructura, en forma de lista. El lisp fue uno de los primeros lenguajes en manejar el bloque de excepciones con los comandos catch y throw.

Características de LISP

A continuación se detallan algunas de las características de las principales variantes del lenguaje LISP:

* LISP posee un manejo de memoria automático que libera el espacio utilizado por los objetos que dejan de ser necesitados.
* Incluye un mecanismo bastante simple para utilizar evaluación perezosa de expresiones.
* LISP no posee un sistema de tipos estáticos como puede ocurrir en c/c++. LISP asocia los tipos a los valores en vez que a las variables, por ello los errores de mal uso de tipos solo puedan ser detectados en tiempo de ejecución y no de compilación.
* Las implementaciones en LISP suelen programarse de manera que cálculos iterativos puedan realizarse en un espacio constante (en memoria) aunque hayan sido descritos mediante el uso de la recursividad.

Funciones en Lisp

Funciones normales, son las que se suelen incluir en las implementaciones de LISP.

Funciones de lista, para manipulación de listas:

* car L, devuelve el primer elemento de L.
* cdr L, devuelve la cola (lista - primero).
* cons x y, devuelve uan lista formada por x e y.
* List x y z, devuelve la lista (x y z).
* quote x, no se evalúa x.

Predicados

* atom x, devuelve True si x es un átomo.
* numberp x, devuelve True si x es un número.
* greaterp x y, devuelve True si x>y.
* lessp x y, devuelve True si x<y.
* null x, devuelve True si x es nulo.
* and x y, devuelve x and y.
* or x y, devuelve x or y.
* not x, devuelve not x.
* eq x y, devuelve True si x=y.

Funciones aritmética

* +, -, \*, y /.
* rem x y, devuelve el módulo x/y (remainder).

Funciones de entrada y salida

* Load nombrearchivo, lee el archivo a memoria.
* print x, imprime el elemento x.
* open nombrearchivo, abre un archivo y devuelve una puntero al mismo.
* read, lee del terminal un átomo.
* help, proporciona ayuda.
* Trace, traza la función.
* bye, termina LISP.

Abstracción y encapsulamiento

LISP, en origen, no incluye características de abstracción de datos. CLOS fue una ampliación de LISP con orientación de objetos. Características:

* Herencia múltiple.
* Funciones genérica.
* Metaclases y metaobjetos.
* Técnica de creación e inicialización de objetos que permite control del proceso por parte del usuario.

Evaluación del lenguaje

* LISP ha evolucionado durante más de 30 años y desarrollado para inteligencia artificial, pero no es adecuado para aplicaciones convencionales.
* Las versiones compiladas son algo más eficientes.

Ejemplo de código en LISP

(defun sum (x) ;calcula la suma de una lista x

(cond ((null x) O)

((atom x) x)

(t (+ (car x) (sum (cdr x)]

(defun cont (x) ; cuenta el número de valores de x

(cond ((null x) O)

((atom x) 1)

(t (addl (cont (cdr x)]

(defun media () ; el programa principal comienza aquí

(print 'introducir la lista a promediar')

(setq x (read))

(setq n (count x))

(setq med (/ (sum x) n))

(princ "la media es = ` `)

(print med)]

El programa está compuesto de tres funciones del LISP, cada una indicada por la cabecera "defun" (abreviación de "define function). La primera función "sum" calcula la suma aritmética de los elementos de la lista x. La segunda, "cont", calcula el número de valores de la lista. La tercera función, "media", es el programa principal y controla la entrada (usando "read), el cálculo de la media "med" y la salida (usando "print").

Los comentarios en LISP comienzan con el delimitador especial punto y coma (;) y continúan hasta el final de la línea. Las variables están normalmente sin declarar y tienen un ámbito global. También pueden especificarse variables acotadas, las cuales son locales a una función.