

*Apuntes de **COMMON LISP***

para la asignatura Inteligencia Artificial

INDICE

MODULO 1. SIMBOLOS, ATOMOS Y CONSES

1.1. SIMBOLOS LISP Y OBJETOS SIMBOLICOS 1

1.2. TIPOS **ATOM**: NUMEROS, CARACTERES Y STRINGS 2

1.3. TIPOS **CONS** 4

1.4. EVALUACIÓN EN LISP 8

EJERCICIOS 10

MODULO 2. FUNCIONES PREDEFINIDAS SOBRE LISTAS

2.1. EXPRESIONES SIMBOLICAS Y FORMAS 13

2.2. FUNCION QUOTE 14

2.3. FUNCIONES DE SELECCION 14

2.4. FUNCIONES DE CONSTRUCCION 17

2.5. OTRAS FUNCIONES 23

EJERCICIOS 24

MODULO 3. FUNCIONES PARA OPERAR CON NUMEROS

3.1. OPERACIONES ARITMETICAS 27

3.2. FUNCIONES MATEMATICAS 28

3.3. FUNCIONES PARA COMPARAR NUMEROS 29

3.4. PREDICADOS NUMERICOS 29

EJERCICIOS 31

MODULO 4. PREDICADOS Y OPERADORES LOGICOS

4.1. PREDICADOS DEFINIDOS EN LISP 33

4.2. OPERADORES LOGICOS 35

EJERCICIOS 37

MODULO 5. FUNCIONES DEFINIDAS POR EL USUARIO Y ESTRUCTURAS CONDICIONALES

5.1. DEFINICION DE FUNCIONES 40

5.2. ESTRUCTURAS CONDICIONALES 44

5.3. DOCUMENTACION 47

EJERCICIOS 49

MODULO 6. ENTRADA/SALIDA SIMPLE Y STREAMS

6.1. ENTRADA/SALIDA SIMPLE 51

6.2. STREAMS 53

EJERCICIOS 55

MODULO 7. LIGADURAS, ASIGNACION Y AMBITO

7.1. LIGADURA DE LOS PARAMETROS EN LA FUNCION 57

7.2. FUNCION DE ASIGNACION 59

7.3. SENTENCIA LET 60

7.4. AMBITO LEXICO Y DINAMICO 63

EJERCICIOS 65

MODULO 8. ESTRUCTURAS ITERATIVAS

8.1. ESTRUCTURAS ITERATIVAS 67

8.2. FUNCIONES ITERATIVAS PREDEFINIDAS 69

EJERCICIOS 72

MODULO 9. MACROS Y OTROS TIPOS DE DATOS

9.1. DEFINICION, EXPANSION Y EVALUACION DE MACROS 75

9.2. LISTAS ASOCIATIVAS 77

9.3. LISTAS DE PROPIEDADES 79

9.4. ESTRUCTURAS DE REGISTROS 80

INTRODUCCIÓN

Lisp fué propuesto por John McCarthy a finales de los 50. Se diseñó como alternativa al modelo de computación tradicional, es un lenguaje dirigido a procesamiento simbólico en lugar de numérico, implementando el modelo de las funciones recursivas que proporcionan una definición sintáctica y semánticamente clara. Las **listas** son la base tanto de los programas como de los datos en LISP: es un acrónimo de LISt Processing. Lisp proporciona un conjunto potente de funciones que manipulan listas, implementadas internamente como punteros. Lisp ofrece a los programadores la potencia y generalidad de las estructuras de punteros, librándole de la manipulación explícita de los mismos y de sus operaciones.

LISP fué en sus orígenes un lenguaje simple y pequeño, consistente en funciones de construcción y acceso a listas, definición de nuevas funciones, unos pocos predicados para detectar igualdades y funciones para evaluar llamadas a funciones. El único fundamento de control era la recursión y las condicionales simples. Si se necesitaban funciones más complejas se definían en función de las primitivas. A través de los años se le han ido incorporando funciones especializadas como estructuras de datos, control del programa, aritmética de reales y enteros, entrada/salida, edición de funciones LISP y traceado de la ejecución de un programa. Estas extensiones han provocado la aparición de diferentes dialectos del LISP: INTERLISP, MACLISP, FRANZLISP, ZETALISP, etc. Debido a esta proliferación de dialectos, en 1983, la Agencia de Proyectos Avanzados de Investigación propuso un dialecto standard del lenguaje conocido bajo el nombre de Common Lisp.

En un principio, muchos de los programas desarrollados dentro del ámbito de la Inteligencia Artificial se implementaron en LISP. Posteriormente, aparecieron formalismos de más alto nivel como los sistemas de producción, sistemas basados en objetos, deducción de teoremas lógicos, sistemas expertos basados en reglas, etc. utilizando al Lisp como lenguaje para su implementación. Muchos entornos modernos para la creación de sistemas expertos están implementados en Lisp. Pero incluso utilizando estas herramientas, es necesario frecuentemente acceder al Lisp para aumentar la funcionalidad del lenguaje de alto nivel. En algún sentido se podría considerar al Lisp como el "ensamblador" de la Inteligencia Artificial.

MODULO 1

SIMBOLOS, ATOMOS Y CONSES

El objetivo fundamental de este capítulo será:

- conocer los diferentes tipos de datos que puede apuntar un símbolo en memoria,
- dar ejemplos de diferentes tipos de datos atómicos,
- escribir gráficamente un símbolo con apuntadores del nombre en memoria,
- conocer si una lista es una lista de verdad o una lista punteada,
- conocer si un elemento es un átomo, una lista o ninguno de los dos.

- determinar el número de elementos del nivel superior de la lista,
- dibujar celdas y apuntadores que representen las estructuras CONS y las relaciones de sus elementos en memoria.
- convertir las celdas y apuntadores, que representan listas de verdad y listas punteadas, en listas con niveles únicos o múltiples.

1.1. SIMBOLOS LISP Y OBJETOS SIMBOLICOS

En Lisp todo son objetos que se representan mediante símbolos. La regla de oro de Lisp (lenguaje funcional) es la evaluación de sus objetos. Existen diferentes reglas de evaluación dependiendo del tipo de objeto.

Cada símbolo tiene asociados cinco componentes que lo caracterizan: nombre imprimible, el valor, una definición de función, una lista de propiedades y un paquete en el que se define dicho símbolo. Gráficamente podemos representarlo como: