CLIPS (C Language Integrated Production System)

Hemos visto agentes, diferentes algoritmos y sistemas para tratar de crear sistemas inteligentes.

A continuación, analizaremos otro tipo de sistemas, llamados sistemas expertos. En la introducción de la asignatura, vimos que tras "el invierno de la IA", entre 1970 y 1990 trataron de crear este tipo de sistemas, los cuales siguen siendo usados hoy en día.

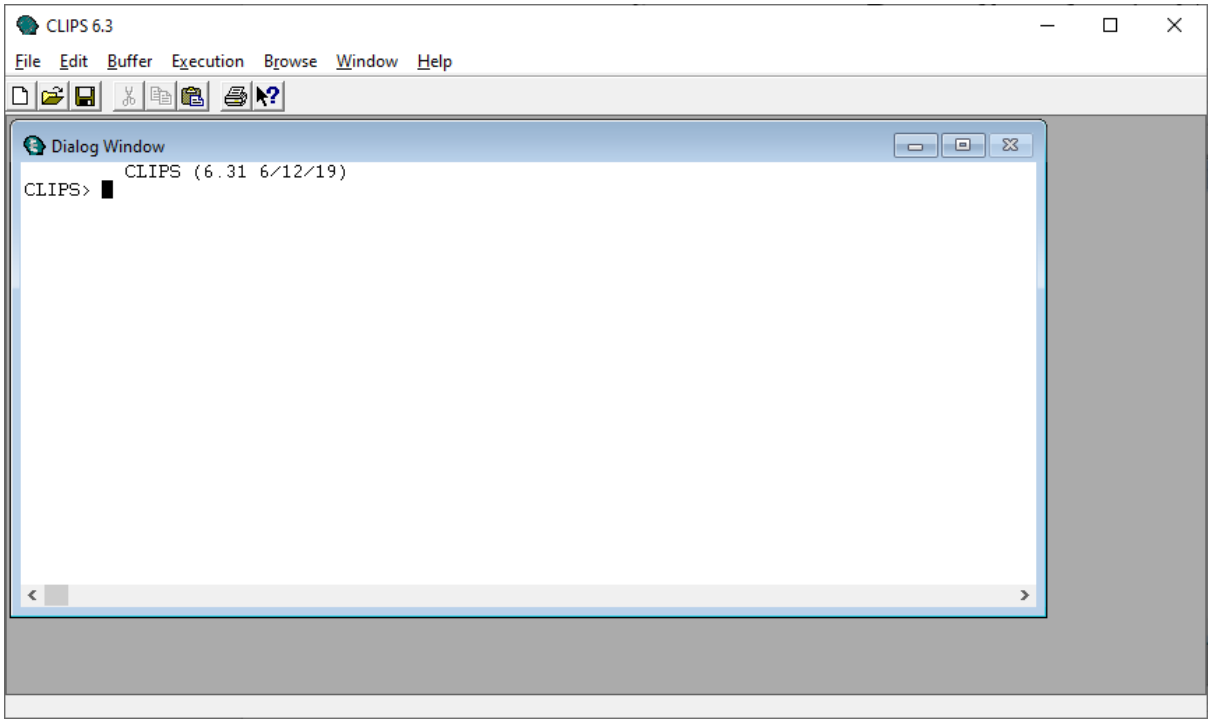
CLIPS (C Language Integrated Production System), es una herramienta creado por la NASA en 1984, con el objetivo de crear estos tipos de sistemas expertos. La base de estos sistemas, son las reglas. Básicamente, las reglas son un tipo de programas pequeños, como si fueran un IF.

Para trabajar con CLIPS, recomiendo usar Windows. En este sistema operativo nos ofrecen una interfaz, no así en Linux. Es posible también instalar en MAC.
Para Windows, descargar desde este link:

clips_windows_executables_630.zip

Para las demás plataformas, podéis encontrar CLIPS en la siguiente dirección:
https://sourceforge.net/projects/clipsrules/files/CLIPS/

Una vez descargado, en Windows, ejecutaremos el fichero CLIPSIDE64.exe, y se nos abrirá la interfaz (interprete) de CLIPS



Como todos los programas, en la parte de arriba tenemos la opción de abrir un documento, guardar, editar, etc.

Antes de comenzar con las reglas, analizaremos los elementos básicos de CLIPS, variables, condiciones, bucles, cómo crear nuestras funciones, etc.

Los tipos de datos admitidos por este lenguaje, son enteros, reales, STRINGS y SIMBOLOS (cadenas sin comillas).
Como ocurre en Python, no hace falta definir de qué tipo van a ser las variables.
Podéis probar las secuencias que aparecen en la diapositiva 6/22 en el intérprete de CLIPS.

Tranquilos, veremos que hay una forma más cómoda de crear nuestros programas. Para ello, crearemos un fichero con extensión CLP, por ejemplo, ***prueba.clp.*** Este fichero, os recomiendo que lo editéis con notepad++, Visual Studio Code (con el plugin ***Clips Language support*** creado por nerg).

Vamos a escribir dentro del fichero ***prueba.clp*** el siguiente código:

```
(deffunction suma (?a ?b)
    (return (+ ?a ?b))
)
```

¡Cuidado con los paréntesis, son necesarios!
Como podemos observar, todas las funciones se indican al principio (+ ?a ?b) y NO (?a + ?b). **Por otra parte, antes de cada función, siempre necesitaremos paréntesis!**

Guardamos el fichero, volvemos a la interfaz de CLIPS, y dentro de la pestaña ***File,*** seleccionamos ***load***, y cargamos nuestro programa ***prueba.clp***. Si no hay ningún tipo de error en el código, en el intérprete de CLIPS, tras cargar, nos tiene que aparecer TRUE. Todos los comandos que vayan apareciendo en las diapositivas, podéis ir añadiendo a esta función suma que hemos creado.

Para ejecutar nuestra función **suma**, escribiremos lo siguiente en el intérprete:
CLIPS> **(suma 3 6)**
Nos tendría que devolver 9.

Bien, como os habréis dado cuenta, para definir las funciones hay que usar la palabra reservada **deffunction.** Después le daremos un nombre a nuestra función **"suma",** indicaremos los parámetros, y después iría el cuerpo del programa.

Los parámetros o variables simples, se indican con el signo de interrogación y un **?nombre**.
Para poder usar listas, se definen con dólar y signo de interrogación y **$?nombre**.
Para crear las listas, se usa el comando **create$.**

Para asignar un valor a las variables, se usa el comando **bind:**
(bind ?a 5) → a la variable **?a** le hemos asignado el valor 5

Para las listas (conocidos como variables multicampo), haremos lo siguiente:
(bind $?a (create$ 1 2 3 4)) → a la variable **$?a** le hemos asignado el valor (1 2 3 4)

Defglobal *se usa para definir variables globales:*
(defglobal ?*a* = 3)

para diferenciar las diferentes variables, en este caso se usa signo de interrogación con asterisco y otro asterisco después del nombre

Para cambiar de valor a una variable global:
(bind ?*var_global* 5)

En la diapositiva 10/22, se muestran los comandos para poder leer/escribir en ficheros. Además, podremos usar el comando ***printout*** para escribir en pantalla. Este comando necesita el parámetro ***t*** para indicar que se va a escribir por pantalla, si no, deberemos indicar el nombre del fichero donde queramos escribir algo.

En la diapositiva 11, 12 y 13, tenemos diferentes estructuras de control, bucles, condiciones, etc.

En la diapositiva 14 y 15, tenemos diferentes funciones propias de CLIPS, así como operadores lógicos y matemáticos.

En la diapositiva 16 y 17, se muestran diferentes funciones para tratar variables de tipo STRING o SIMBOLO.

En la diapositiva 18 y 19, se muestran diferentes funciones para tratar variables multicampo (listas). Si se necesita pasar como parámetro una lista a una función, tal y como se indica en la diapositiva 21, este parámetro siempre ha de ir el último.

¿Y qué ocurre si se necesita pasar dos listas a una función? Para ello, engañaremos a CLIPS, donde, en vez de poner los nombres de los parámetros con $?, para una de ellas solamente indicaremos con ?, tal y como se muestra en la diapositiva 21.