

PRÁCTICA NÚMERO 3

INTRODUCCIÓN A CLIPS

(Nota: Todos los archivos de texto que se ven a continuación pueden ser creados mediante el editor que incluye el programa CLIPS y al que se accede a través del menú File -> Editor...)

Introducción de comandos CLIPS desde archivos

Si se quieren introducir desde un archivo una serie de comandos CLIPS se debe utilizar el comando **batch**.

Por ejemplo, si se tienen los siguientes archivos:

```
;*****  
;  
;                archivo hechos.clp  
;*****
```

(hay naranjas)
(hay patatas)
(hay huevos)
(hay cebolla)
(hay pimientos)

```
;*****  
;  
;                archivo reglas.clp  
;*****
```

(defrule posible-tortilla-sin-cebolla

```
(hay patatas)
(hay huevos)
=>
(printout t "Se puede hacer una tortilla de patatas sin cebolla." crlf)
)
```

```
(defrule posible-tortilla-con-cebolla
  (hay patatas)
  (hay huevos)
  (hay cebolla)
=>
  (printout t "Se puede hacer una tortilla de patatas con cebolla." crlf)
)
```

```
;*****
;
;               archivo cargar.bat
;*****
```

```
(clear)
(load "reglas.clp")
(load-facts "hechos.clp")
```

Se pueden cargar los hechos y las reglas mediante el comando:

```
(batch "cargar.bat")
```

también se puede cargar el archivo con la instrucción File -> Load Batch.

También se puede guardar todo en un archivo (hechos y reglas) teniendo en cuenta que los hechos deberán aparecer dentro de constructores (deffacts) y se deberá introducir un comando reset en el archivo batch:

```
;*****  
;  
;          archivo programa.clp  
;*****
```

```
(deffacts condiciones-iniciales
```

```
  (hay naranjas)
```

```
  (hay patatas)
```

```
  (hay huevos)
```

```
  (hay cebolla)
```

```
  (hay pimientos)
```

```
)
```

```
(defrule posible-tortilla-sin-cebolla
```

```
  (hay patatas)
```

```
  (hay huevos)
```

```
=>
```

```
  (printout t "Se puede hacer una tortilla de patatas sin cebolla." crlf)
```

```
)
```

```
(defrule posible-tortilla-con-cebolla
```

```
  (hay patatas)
```

```
  (hay huevos)
```

```
  (hay cebolla)
```

```
=>
```

```
  (printout t "Se puede hacer una tortilla de patatas con cebolla." crlf)
```

```
)
```

```
;*****  
;  
;                                archivo cargar.bat  
;*****
```

```
(clear)  
(load "programa.clp")  
(reset)
```

Dentro del propio archivo batch se puede insertar cualquier comando como se introduciría en la propia ventana de comandos de CLIPS, por lo que se puede tener esta otra configuración a la hora de introducir hechos y reglas:

```
;*****  
;  
;                                archivo reglas.clp  
;*****
```

```
(defrule posible-tortilla-sin-cebolla  
  (hay patatas)  
  (hay huevos)  
=>  
  (printout t "Se puede hacer una tortilla de patatas sin cebolla." crlf)  
)
```

```
(defrule posible-tortilla-con-cebolla  
  (hay patatas)  
  (hay huevos)  
  (hay cebolla)  
=>
```

```
(printout t "Se puede hacer una tortilla de patatas con cebolla." crlf)
)
```

```
;*****
;
;               archivo cargar.bat
;*****
```

```
(clear)
(assert (hay naranjas))
(assert (hay patatas))
(assert (hay huevos))
(assert (hay cebolla))
(assert (hay pimientos))
(load "reglas.clp")
```

Otra forma de introducir los hechos y reglas es, una vez que se tiene el archivo programa.clp (visto anteriormente), cargarlo en memoria ejecutando el comando del menú File -> Load Constructs y eligiendo dicho archivo. Antes de cargar el archivo es conveniente efectuar un comando (clear). Después de cargar el archivo, se inicializan los hechos mediante el comando (reset).

Estructuras de la programación procedural

CLIPS permite el uso de estructuras de control típicas de la programación procedural, como los bucles **while** e **if then else**.

El bucle while tiene la forma:

```
(while <condicion> [do] <acciones>)
```

la partícula "do" es opcional.

La instrucción **break** termina el bucle while cuando se ejecuta.

La sentencia if then else tiene la forma:

(if <condicion> then <acciones1> [else <acciones2>])

Depuración de programas

Para controlar la ejecución de los programas, se pueden utilizar puntos de interrupción, que se establecen mediante la sentencia **break**, indicando el nombre de la regla donde se parará la ejecución del programa:

(set-break <nombre_regla>)

para quitar el punto de interrupción:

(remove-break <nombre_regla>)

para mostrar los puntos de interrupción:

(show-breaks)

EJERCICIOS

1.- Entrada de datos desde archivos

Se desea obtener una base de hechos del tipo:

(persona <nombre> <apellido1> <apellido2>)

Para construir esta base de hechos se obtendrán los datos de un archivo externo que tenga en cada línea un registro con los datos de un hecho:

<nombre> <apellido1> <apellido2>

Se pide construir la base de hechos mediante un programa en CLIPS que lea los datos del archivo línea a línea hasta el final del fichero y los introduzca en la base de hechos.

El ejercicio se realizará usando las sentencias read o readline para el manejo de archivos.

2.- Operaciones con conjuntos

Se tienen dos conjuntos que pueden contener los siguientes elementos (sin repeticiones):

A B C D E F G H I J K L M N O P

Se pide calcular como conjuntos resultado:

1. El conjunto unión de los dos iniciales.
2. El conjunto intersección de los iniciales.

Los conjuntos resultado serán del mismo tipo que los de origen, es decir, no podrá haber repetición de elementos.

El ejercicio se realizará de dos formas: sin usar plantillas y usándolas.

Los resultados se deben presentar por pantalla.

3.- Sistema Experto

Implementar un sistema en CLIPS que realice una serie de preguntas al usuario sobre las características de un animal para intentar asignarlo a una de las categorías consideradas: mamíferos, aves, reptiles y anfibios. El sistema debe devolver la categoría a la que pertenece el animal, si no encaja en ninguna categoría, debe indicar que no sabe a que categoría pertenece.

Características de las categorías:

- *Mamífero*: sangre caliente, piel con pelo y se mueve andando.
- *Ave*: sangre caliente, piel con plumas y se mueve volando.
- *Reptil*: sangre fría y piel con escamas.
- *Anfibio*: sangre fría y piel porosa.

Hay que tener en cuenta que algunos animales no cumplen estas características, por ejemplo el avestruz y el murcielago. En el caso de que las características del animal coincidan con las de estos animales, el sistema debe devolver el nombre del animal en lugar de su categoría.