

## **Ejercicio 1.**

### **1. Hechos**

Introduzca los siguientes hechos en la ventana Dialog y abra la ventana Facts (seleccionándola en la opción Window del menú):

(nombre Juanjo)  
(nombre Juan)  
(apellido-1 Cruz)  
(apellido-1 Perez)  
(apellido-2 Lopez)  
(nombre Federico)  
(apellido-1 Perez)  
(apellido-2 Jimenez)

### **Cuestiones 1:**

*¿Qué devuelve Clips al añadir un hecho a la Base de Hechos (BH)?*  
*¿Por qué en un caso devuelve False?*

### **2. Reglas**

Programe 2 reglas (Abra la ventana Agenda, para ver como se activan las reglas)

La primera debe tener como antecedente el hecho (*nombre Juan*) y como consecuente una instrucción que imprima en pantalla "Tu nombre de pila es Juan":

La segunda tiene como antecedente los hechos (*nombre Juan*), (*apellido-1 Perez*) y (*apellido-2 Lopez*), y el consecuente imprimirá en pantalla "Te llamas Juan Perez Lopez".

### **Cuestiones 2:**

*¿Se ha activado alguna regla? ¿Qué hechos activan cada regla?*

### **3. Ejecución de Reglas**

Ejecute las reglas paso a paso con, por ejemplo, (*run 1*) y (*run 1*)

### **Cuestiones 3:**

*¿Qué regla se ejecuta primero? ¿Por qué?*

*¿Qué pasa si reiniciamos con (*clear*)?*

#### 4. Definición de hechos con deffacts

Defina los hechos con deffacts y almacénelos junto con las reglas en un fichero con extensión .clp (o .txt). Use el WordPad o cualquier editor para generar un fichero de texto plano. Cargue el fichero con la opción (load) del menú principal.

##### Cuestiones 4:

*Si se introducen los hechos con (deffacts), y se carga el fichero ¿qué ocurre en la BH y en la Agenda?*

*Ejecute (reset). ¿Qué ocurre en la BH y en la Agenda?*

*¿Cuál es el primer hecho que se ha activado?*

*Ejecute el programa para chequear que obtiene el mismo resultado que antes.*

*Limpie la base de conocimiento.*

#### 5. Uso de variables en las reglas

Implemente un programa que controle un robot (chofer o peatón) para pasar un cruce con semáforo. El programa será interactivo, preguntando en primer lugar al usuario el tipo de robot que va a manejar (chófer o peatón), y en segundo lugar, el estado del semáforo, para así decidir la acción del robot (seguir, parar o pasar con cuidado).

La primera regla se activará con initial-fact (hecho que se activa por defecto una vez cargado el fichero de reglas e insertado (reset)). La regla hace uso de printout y (read), para solicitar el dato.

```
; regla preguntando el tipo de robot
;

(defrule regla-tipo-robot
  (initial-fact)
=>
  (printout t "Que tipo de robot es (chofer/peaton)? ")
  (assert (tipo-robot (read)))
)
```

La segunda regla preguntará cómo está el semáforo para los coches, y se activará sólo en el caso de que el robot sea un chófer. Almacenará el color del semáforo en un hecho (semáforo <color>), donde <color> puede ser verde, rojo, ámbar o intermitente.

```
;
; regla regla-semaforo-chofer
```

```
;
```

```
(defrule regla-semaforo-chofer .....
```

La tercera regla preguntará cómo está el semáforo para los peatones, y se activará sólo en el caso de que el robot sea un peatón. Almacenará el color del semáforo en un hecho (semáforo <color>), donde <color> puede ser verde, rojo, ámbar o intermitente.

```
;
```

```
; regla regla-semaforo-peaton
```

```
;
```

```
(defrule regla-semaforo-peaton .....
```

La cuarta regla permitirá pasar al robot si el semáforo está en verde. Usará una variable (las variables comienzan siempre por ?) que permite equiparar el hecho (tipo-robot ?x) con cualquier tipo de robot (peatón o coche). Dicha variable se queda ligada al valor con el que se equipara durante la ejecución de la regla, pudiéndose usar su valor en el consecuente de la regla (p.ej., en el printout).

```
;
```

```
; regla robot-puede-pasar
```

```
;
```

```
(defrule robot-puede-pasar
```

```
(tipo-robot ?tipo)
```

```
(semaforo verde)
```

```
=>
```

```
(printout t "El robot " ?tipo " puede pasar" crlf)
```

```
)
```

La quinta regla se activa si el estado del semáforo es rojo y, al igual que la anterior, muestra el mensaje correspondiente.

```
;
```

```
; regla debe-esperar
```

```
;
```

La sexta regla se activa cuando el semáforo está en ámbar o intermitente, y muestra el mensaje oportuno. Utilice el símbolo de la disyuntiva para considerar las dos opciones en el mismo antecedente de la regla: (semaforo ambar | intermitente)

```
;
```

```
; regla con-cuidado
```

```
;
```

Cargue el fichero .clp en el entorno,(reset) y (run) para ejecutar el programa.

**Cuestiones 5:**

*¿Qué devuelve el programa cuando el robot es un coche y el semáforo para los coches está en rojo? Copie y pegue el resultado mostrado en la dialog window tras ejecutar (run).*