

# INFORMATICA APLICADA

## Práctica numero 2: Fundamentos de CLIPS (II)

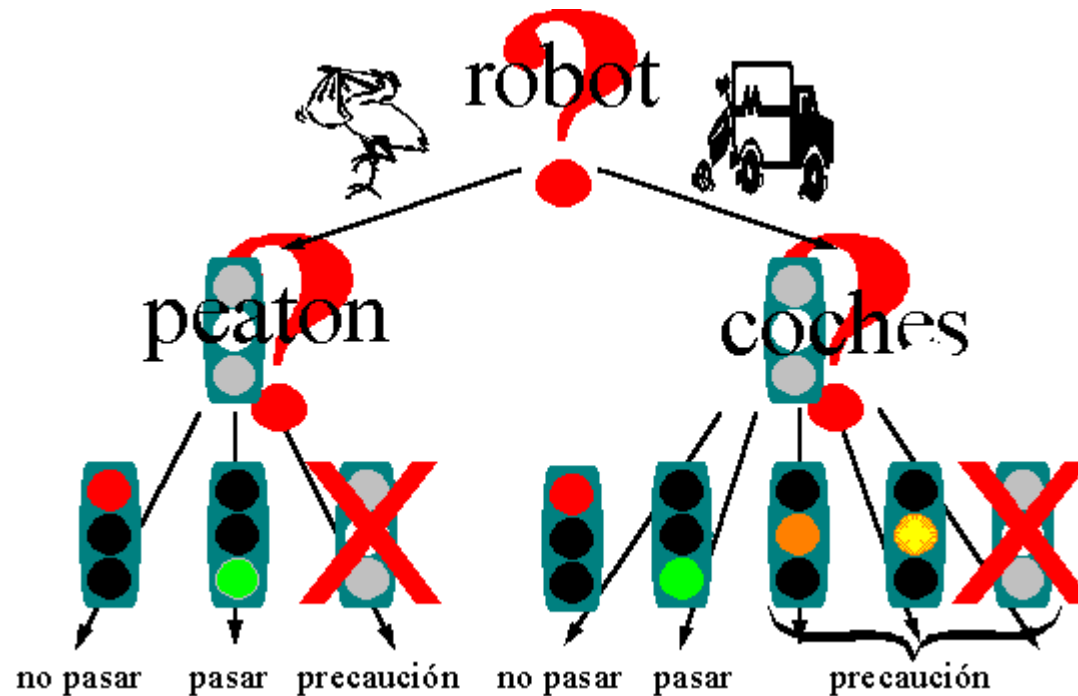
### Ejercicio 3. Paso con Semáforos (I).

#### 1. Planteamiento del problema

Se trata de desarrollar un sencillo SE, que controle un robot (chofer o peatón) para pasar un cruce con semáforo.

Se ha optado por un programa interactivo, que nos pregunta el tipo de robot que vamos a manejar (chófer o peatón), y según el caso, el estado del semáforo, para así decidir la acción del robot.

El árbol de decisión es el que se muestra en la figura.



#### 2. Definición de reglas.

## 2.1 Regla que-robot

Nos pregunta por el tipo de robot que estamos estudiando, siendo las respuestas posibles coche y peaton. Dependiendo de nuestra respuesta, activará el hecho correspondiente que activará la siguiente regla.

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

## 2.2 Regla como-semaforo-chofer

Nos pregunta por el estado del semáforo de los coches, y se activa si se cumple el hecho (robot coches).

```
;
; regla como-semaforo-chofer
;
(defrule como-semaforo-chofer
(tipo-robot chofer)
=>
(printout t "Como esta el semaforo de los coches?" crlf)
(printout t "(rojo/verde/ambar/intermitente/averiado) ")
(assert (semaforo (read)))
)
```

## 2.3 Regla como-semaforo-peaton

Nos pregunta por el estado del semáforo de los peatones, y se activa si en que-robot se cumple el hecho (robot peaton).

```
;
; regla como-semaforo-peaton
;
(defrule como-semaforo-peaton
(tipo-robot peaton)
=>
(printout t "Como esta el semaforo de los peatones?" crlf)
(printout t "(rojo/verde/averiado) ")
(assert (semaforo (read)))
)
```

## 2.4 Regla puede-pasar

Se activa si el estado del semáforo (cualquiera de ellos), es verde. Muestra un mensaje indicando que el robot puede pasar.

```
;
; regla puede-pasar
;
(defrule puede-pasar
(tipo-robot ?tipo)
(semáforo verde)
=>
(printout t "El robot " ?tipo " puede pasar" crlf)
)
```

## 2.5. Regla debe-esperar

Se activa si el estado del semáforo es rojo y, al igual que la anterior, muestra el mensaje correspondiente.

```
;
; regla debe-esperar
;
(defrule debe-esperar
(tipo-robot ?tipo)
(semáforo rojo)
=>
(printout t "El robot " ?tipo " debe esperar" crlf)
)
```

## 2.6 Regla con-cuidado

Se activa cuando el semáforo está en ámbar, averiado o intermitente, y muestra el mensaje oportuno.

```
;
; regla con-cuidado
;
(defrule con-cuidado
(tipo-robot ?tipo)
(semáforo ambar | intermitente | averiado)
=>
(printout t "El robot " ?tipo " debe ir con cuidado" crlf)
)
```

[\[Código del programa\]](#) [\[Siguiete Ejercicio\]](#) [\[Indice General\]](#)

---

Práctica realizada por **Juan José Cruz Jiménez**

e-mail: [i42crjij@uco.es](mailto:i42crjij@uco.es)