

## Ejercicio 4: Modificación de hechos e introducción de datos por teclado

### 1 Modificando hechos con plantillas

Partiremos del programa hecho en el ejercicio anterior, con el objetivo de almacenar el diagnóstico y el tratamiento en la plantilla del paciente.

1. Añada un nuevo atributo llamado *diagnostico* a la plantilla paciente.
2. Modifique las reglas correspondientes, almacenando el hecho paciente en una variable para poder modificarlo  
(defrule Sarampion  
?p <- (Paciente (fiebre alta) (exantema si) (vacunado no) (id ?x) (diagnostico nil))  
=>  
(modify ?p (diagnostico sarampion))  
(printout t "Diagnostico del paciente " ?x ": Sarampion" crlf))
3. Pruebe el ejemplo con varios pacientes en la BH (definidos en el mismo deffacts).

**Cuestión 1:** ¿Qué infiere el sistema para los siguientes pacientes?

```
(deffacts BHinicial
(Paciente (fiebre baja) (exantema si) (id P1))
(Paciente (fiebre alta) (exantema si) (id P2))
(Paciente (fiebre moderada) (garganta-irritada si) (id P3))
(Paciente (fiebre baja) (sarpullido si) (id P4)))
```

### 2 Infiriendo el grado de severidad de la fiebre a partir de la temperatura del paciente

Modifique el programa anterior para que infiera el grado de severidad de la fiebre (alta, moderada o baja), a partir del valor de la temperatura del paciente.

1. Cambie el atributo fiebre:  
(slot fiebre (type SYMBOL)(allowed-symbols alta moderada baja no nil)(default nil))
2. Añada una propiedad llamada temperatura a la plantilla paciente, e infiera la fiebre del paciente teniendo en cuenta el siguiente conocimiento:
  - a. Para valores menores de 36,7 grados, no hay fiebre.
  - b. Entre 36,7 y 37,5 grados, el grado de fiebre se puede considerar bajo.
  - c. Entre 37,6 y 39 grados, el grado de fiebre se puede considerar moderado.
  - d. Fiebres a partir de 39 grados se consideran elevadas.
3. Siga las mismas pautas de trabajo que en el apartado anterior para programar las 4 reglas necesarias. Tenga además en cuenta que para chequear el valor de la temperatura puede usar test en la parte antecedente de la regla. Por ejemplo:

```
(test (> ?t 39))
```

*(test (and (<= ?t 39) (> ?t 37.6)))*

**Cuestión 2:** ¿Qué infiere el sistema para los siguientes pacientes?

*(deffacts BHinicial*

*(Paciente (temperatura 36.5) (exantema si) (id P1))*

*(Paciente (temperatura 39.5) (exantema si) (id P2))*

*(Paciente (temperatura 38) (garganta-irritada si) (id P3))*

*(Paciente (temperatura 37.3) (sarpullido si) (id P4)))*

### 3 Introduciendo datos por teclado

Modifique el programa anterior para introducir los datos de un paciente por teclado. Utilice las funciones *bind* y *read* para ello.

- a. Se necesitará un hecho de paciente inicial para modificarlo con los datos provenientes de teclado.

*(deffacts BHinicial (Paciente (id P1)(exantema si))*

- b. Añada una regla llamada *Rinicial* para pedir el dato de la temperatura por teclado. Usaremos el hecho inicial para disparar inicialmente esta regla. Además dicha regla sólo se debe disparar si no hay un valor de temperatura para el paciente. Haga uso de *bind* para ligar el valor leído por teclado a una variable *?t* que se usará para modificar la temperatura del paciente.

*(defrule Rinicial*

*(initial-fact)*

*?p <- (Paciente (temperatura nil))*

*=>*

*(printout t "Entra la temperatura del paciente: ")*

*(bind ?t (read))*

*(modify ?p (temperatura ?t)))*

**Cuestión 3:** ¿Qué sucede cuando ejecutamos (*reset*)? ¿De qué regla(s) habla CLIPS? ¿Qué valor tiene la temperatura del paciente inicialmente? ¿Qué condición puede estar fallando?

Pruebe a añadir la condición: *(test (numberp ?t))*

**Cuestión 4:** ¿En qué reglas debe añadirse? ¿En qué lugar del antecedente, como primera, segunda o tercera condición? ¿Qué sucede cuando ejecutamos (reset)?

#### 4 Pidiendo sólo datos necesarios

Defina ahora valores por defecto nil, siguiendo la siguiente plantilla.

*(deftemplate Paciente*

*(slot fiebre (type SYMBOL)(allowed-symbols alta moderada baja no nil)(default nil))*

*(slot exantema (type SYMBOL)(allowed-symbols si no nil) (default nil))*

*(slot sarpullido (type SYMBOL)(allowed-symbols si no nil)(default nil))*

*(slot garganta-irritada (type SYMBOL)(allowed-symbols si no nil)(default nil))*

*(slot vacunado (type SYMBOL)(allowed-symbols si no nil)(default nil))*

*(slot id (type SYMBOL))*

*(slot diagnostico (type SYMBOL))*

*(slot temperatura)*

*)*

Para inferir el diagnóstico de cada paciente, pida sólo la información que sea necesaria. Por ejemplo, no tiene sentido preguntar si el paciente está vacunado o no, si no hay ninguna posibilidad de que padezca sarampión. También contemple la posibilidad de partir de un hecho paciente inicial con algunos datos.

**Cuestión 5:** ¿Cuál es la traza que se debe seguir para hacer las preguntas? Indíquela

**Cuestión 6:** ¿Qué reglas debe añadir? Indíquelas. Pruébelas para la siguiente base de hechos inicial y diferentes valores de entrada de datos. En caso de que no se llegue a un diagnóstico, el sistema debe indicarlo en pantalla.

*(def facts BHinicial*

*(Paciente (id P4) (exantema si))*

*(Paciente (id P5))*

*(Paciente (id P3) (vacunado si)))*

**Cuestión 7:** Diseñe un juego de pruebas (BH inicial + datos a introducir) suficientemente exhaustivo para que cubra todas las posibilidades. Anótelo en una tabla, junto a las respuestas de su programa Clips.