REPRESENTACIÓN DEL CONOCIMIENTO - CLIPS

Práctica 3

Objetivos:

1. Estructuras de control
2. Instrucciones para archivos de datos

1. ESTRUCTURAS DE CONTROL

If...then...else

(**if** <expression>
then
<action>*
[**else**
<action>*])
Pueden anidarse las sentencias if, else es opcional en CLIPS, aunque no siempre debe usarse.

```
(defrule valvulas_cerradas
   (temp Alta)
   (valvula ?v cerrada)
=>
  (if (= ?v 6)  then
        (printout t "ATENCION: La valvula especial " ?v " esta cerrada" crlf)
        (assert (realizar operacion especial))
  else
        (printout t "La valvula " ?v " esta cerrada" crlf)
  )
)
```

*Esta regla **DEBE DESCOMPONERSE** en dos reglas:*

```
(defrule valvula6_cerrada
(temp Alta)
(valvula 6 cerrada)
=>
(printout t "ATENCION: La valvula especial " ?v " esta cerrada"  crlf)
(assert (realizar operacion especial))
)

(defrule valvulas_normales_cerradas
(temp Alta)
(valvula ?v&~6 cerrada)
=>
(printout t "La valvula " ?v " esta cerrada" crlf)
)
```

Switch

(**switch** <test-expression>
<case-statement>*
[<default-statement>])
<case-statement> ::=
(case <comparison-expression> then <action>*)
<default-statement> ::= (default <action>*)

```
(defglobal ?*x* = 0)   ;; define una variable global
(defglobal ?*y* = 1)
(deffunction foo (?val)
      (switch ?val
         (case ?*x* then ?*y*)
         (case ?*y* then ?*x*)
         (default none)))
```

```
CLIPS> (foo 0)
1
CLIPS> (foo 1)
0
CLIPS> (foo 2)
None
```

While

(**while** <expression> [do]
<action>*)

Ejemplo

```
(defrule valvulas
(valvulas_abiertas ?v)
=>
(while (> ?v 0)
         (printout t "La válvula " ?v " está abierta" crlf)
         (bind ?v (- ?v 1))))
```

Loop-for-count

(**loop-for-count** <range-spec> [do] <action>*)
<range-spec> ::= <end-index> |
(<loop-variable> <start-index> <end-index>) |
(<loop-variable> <end-index>)
<start-index> ::= <integer-expression>
<end-index> ::= <integer-expression>

Ejemplos

```
(loop-for-count 2 (printout t "Hello world" crlf))


 (loop-for-count (?conta 2 4) do
          (loop-for-count (?contb 1 3) do
               (printout t ?conta " " ?contb crlf)))
```

Return

(return [<expression>])

Ejemplo

```
CLIPS>
(deffunction sign (?num)
          (if (> ?num 0)
              then (bind ?res 1)
              else (if (< ?num 0)
                       then (bind ?res -1)
                       else (bind ?res 0)))
          (return ?res))

CLIPS> (sign 5)
1
CLIPS> (sign -10)
-1
CLIPS> (sign 0)
0
CLIPS>
```

Read

(read [<logical-name>])

Donde <logical-name> es un parámetro opcional para leer desde el lugar asociado a ese nombre, que puede ser un archivo o la entrada estándar cuando se escribe **t** o cuando no se especifica nada (de stdin).
Devuelve un tipo de datos primitivo, por lo que espacios, retornos de carro, tabuladores sólo actúan como delimitadores y no forma parte del valor devuelto a menos que se incluyan entre comillas dobles.

Ejemplo

```
CLIPS> (open "datos.txt" misdatos "w")
TRUE
CLIPS> (printout misdatos "rojo verde")
CLIPS> (close)
TRUE
CLIPS> (open "datos.txt" misdatos)
TRUE
CLIPS> (read misdatos)
red
CLIPS> (read misdatos)
```

```
green
CLIPS> (read misdatos)
EOF
CLIPS> (close)
TRUE
CLIPS>
```

Readline

Similar a **read** pero permite introducir una cadena de texto completa en vez de un solo campo. **read** suele detenerse cuando se encuentra un delimitador mientras que **readline** lo hace cuando se encuentra un retorno de carro, un punto y como el símbolo EOF. Los tabuladores y los espacios forman parte de la cadena devuelta por **readline.**
(readline [<logical-name>])

Ejemplo

```
CLIPS> (open "datos.txt" misdatos "w")
TRUE
CLIPS> (printout misdatos "rojo verde")
CLIPS> (close)
TRUE
CLIPS> (open "datos.txt" misdatos)
TRUE
CLIPS> (readline misdatos)
" rojo verde "
CLIPS> (readline misdatos)
EOF
CLIPS> (close)
TRUE
```

Implementa el siguiente SBR para la gestión de válvulas

Cada válvula se identifica por un nombre, pueden tener 2 estados, abierta o cerrada. Además poseen un valor de presión y dos valores de temperatura, el primero hace referencia a la temperatura interna y el otro a la temperatura externa. Por defecto cualquier valor está a 0, y la válvula cerrada.

En la base de hechos se encuentran los siguientes hechos iniciales:

(valvula (nombre Entrada) (T1 101) (T2 35) (presion 1))
(valvula (nombre Salida) (T1 101) (T2 155) (presion 5))
(valvula (nombre Pasillo1) (T1 99) (T2 37) (estado cerrada))

Las tres reglas que componen la base de conocimientos es la siguiente:

- **R1:** Si una válvula está abierta con un valor de presión de 5, entonces la válvula se cierra y se baja la presión a 0.

- **R2:** Si una válvula cerrada tiene un valor de presión menor de 10 y una temperatura T1 mayor de 35 grados entonces esta válvula deberá abrirse y aumentar la presión en función de la temperatura T1.

  ⇒ Para aumentar la presión crea una función que reciba como argumentos la presión y la temperatura 1 de la válvula: mientras T1 sea mayor de 35 grados aumenta la presión en una unidad, y decrementa la temperatura en 5 grados.

- **R3:** Si dos válvulas distintas, v1 y v2, tienen la misma temperatura T2, y la temperatura T1 de la válvula v2, es menor que T2, entonces se decrementa la temperatura T2 de la válvula v2 y se abren ambas válvulas.

  ⇒ Para decrementar la temperatura crea una función que reciba como argumentos las dos temperaturas, T1 y T2, si la temperatura T2 es mayor que la temperatura T1 entonces T2= T2-T1