



## Práctica 3 :

# Resolución de problemas y búsqueda con sistemas de producción

## 1. Objetivo de la práctica

El objetivo de esta práctica es coger experiencia en el desarrollo de programas en CLIPS tanto escribiendo código como depurando programas. Para ello, deberás escribir un programa CLIPS que resuelva el problema de las fichas. Utilizaremos el modulo de control para implementar la búsqueda A\*. Para una adecuada separación del módulo de control (MAIN) del resto de módulos puedes consultar el ejemplo del 8-puzzle de la lección de control en sistemas de producción.

## 2. Tareas

La situación inicial es

```
+---+---+---+---+---+---+---+
| B | B | B |   | V | V | V |
+---+---+---+---+---+---+---+
```

La situación final es

```
+---+---+---+---+---+---+---+
| V | V | V |   | B | B | B |
+---+---+---+---+---+---+---+
```

Los movimientos permitidos consisten en desplazar una ficha al hueco, saltando como máximo, sobre otras dos. Puedes partir del programa Puzzle.clp presentado en las transparencias y modificar la representación del estado y los operadores. Puedes utilizar la siguiente representación:

```
(deftemplate nodo
  (multislot estado)
  (multislot camino)
  (slot heuristica)
  (slot coste)
  (slot clase (default abierto)))

(defglobal MAIN
  ?*estado-inicial* = (create$ B B B H V V V))
```

---

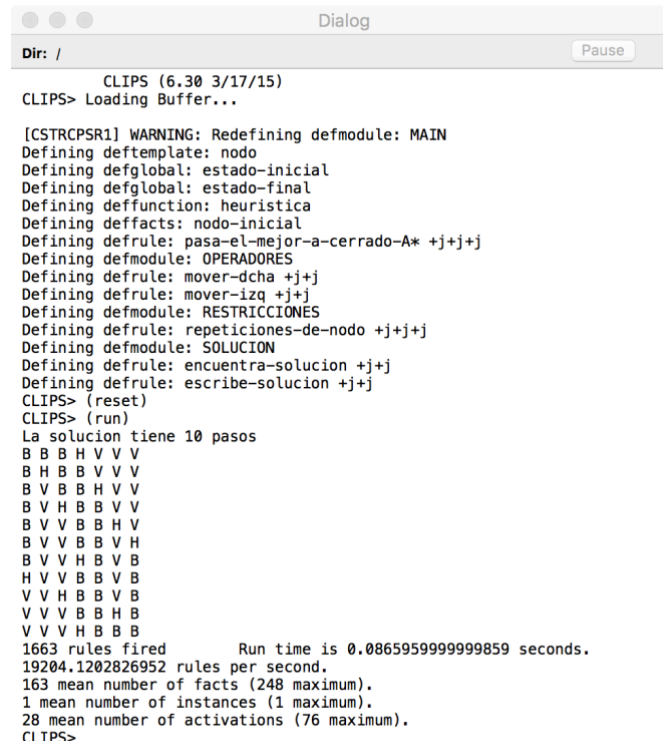
Funciones que pueden ser útiles: `implode`, `explode`, `create`, `duplicate`, `loop-for-count` ... En <http://clipsrules.sourceforge.net/OnlineDocs.html> encontrarás la documentación de CLIPS, aunque es suficiente con los ejemplos que encontrarás en las transparencias de clase.

En esta versión utilizaremos módulos: MAIN, OPERACIONES, RESTRICCIONES que detecta nodos repetidos y SOLUCION que reconoce la solución y escribe los pasos. El módulo MAIN implementa la búsqueda con heurística. Por ejemplo, se puede utilizar las siguientes **heurísticas**:  $h_1$ , La heurística de un estado es la suma de piezas blancas situadas a la izquierda de cada una de las piezas verdes, o bien  $h_2$  el número de fichas descolocadas.

Por ejemplo: La heurística del siguiente estado para  $h_1$  es  $1+2+2=5$ .

```
+---+---+---+---+---+---+---+
| B | V | B |   | V | V | B |
+---+---+---+---+---+---+---+
```

Debes entregar un único fichero `fichas.clp` con los módulos mencionados y listo para ser ejecutado como en el ejemplo que se muestra a continuación:



```
Dialog
Dir: / Pause
CLIPS (6.30 3/17/15)
CLIPS> Loading Buffer...

[CSTRCPSR1] WARNING: Redefining defmodule: MAIN
Defining deftemplate: nodo
Defining defglobal: estado-inicial
Defining defglobal: estado-final
Defining deffunction: heuristica
Defining deffacts: nodo-inicial
Defining defrule: pasa-el-mejor-a-cerrado-A* +j+j+j
Defining defmodule: OPERADORES
Defining defrule: mover-dcha +j+j
Defining defrule: mover-izq +j+j
Defining defmodule: RESTRICCIONES
Defining defrule: repeticiones-de-nodo +j+j+j
Defining defmodule: SOLUCION
Defining defrule: encuentra-solucion +j+j
Defining defrule: escribe-solucion +j+j
CLIPS> (reset)
CLIPS> (run)
La solucion tiene 10 pasos
B B B H V V V
B H B B V V V
B V B B H V V
B V H B B V V
B V V B B H V
B V V B B V H
B V V H B V B
H V V B B V B
V V H B B V B
V V B B H B
V V H B B B
1663 rules fired      Run time is 0.086595999999859 seconds.
19204.1202826952 rules per second.
163 mean number of facts (248 maximum).
1 mean number of instances (1 maximum).
28 mean number of activations (76 maximum).
CLIPS>
```