

---

## 2. Práctica: Representación del Conocimiento Basada en reglas II. Control

---

### 1.1. Objetivo de la práctica

El objetivo de esta práctica es coger experiencia en el desarrollo de programas en CLIPS separando adecuadamente el conocimiento de control del conocimiento experto. Repetimos el problema de las fichas, pero ahora utilizaremos el modulo de control para implementar una estrategia de búsqueda heurística (véase apuntes *Inteligencia Artificial e Ingeniería del Conocimiento I*. Búsqueda. **Julio J. Rubio, Pedro R. Muro y J.A. Bañares** en la página Web de la asignatura de IA <http://iaaa.cps.unizar.es/docencia/IA/apuntes/iabusq04.pdf>).

Para una adecuada separación del módulo de control (MAIN) del resto de módulos puedes consultar la lección 4 y el ejemplo del 8-puzzle de la lección 5.

### 1.2 El problema de las fichas

- **Enunciado:**

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.

- El programa en esta sesión de prácticas lo haremos sin módulos con las estrategias de control globales de CLIPS. Deberas someter el archivo `P1NXXIII.clp`, en el que aparecerá el programa y un ejemplo de ejecución así como las estadísticas que aporta CLIPS:

```
CLIPS (watch statistics)
CLIPS> (reset)
```

```

CLIPS> (run)
La solución de longitud 15 es:
BBBHVVV
BBBVVVH
...
VVVHBBB
93 rules fired Run time is 1.83 seconds.
50.7272727270043 rules per second.
26 mean number of facts (49 maximum).
1 mean number of instances (1 maximum).
5 mean number of activations (12 maximum).

```

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` ...

NOTA: En la siguiente práctica utilizaremos módulos: MAIN, OPERACIONES, RESTRICCIONES que detecta nodos repetidos y SOLUCION que reconoce la solución y escribe los pasos. El módulo MAIN implementará estrategias de control locales como la búsqueda con heurística. Por ejemplo, se puede utilizar la siguiente **heurística**: La heurística de un estado es la suma de piezas blancas situadas a la izquierda de cada una de las piezas verdes.

Ejemplo: La heurística del siguiente estado es  $1+2+2=5$ .

```

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

```