

**Evaluación (1ª instancia)****Duración:** 3 horas**Ejercicio 0: Pregunta individual sobre el laboratorio**

Describa la heurística utilizada para evaluar el tablero de juego en su estrategia.

**Ejercicio 1** [30 puntos]

Considere el problema del puente y la linterna. Hay cuatro personas intentando cruzar un puente sobre un río en la noche, y tienen una sola linterna. El puente soporta como máximo dos personas. Las personas demoran tiempos diferentes en cruzar el puente: la persona A demora 1 minuto, la persona B 2 minutos, la persona C 5 minutos y la persona D 8 minutos. Cuando dos personas cruzan el puente, demoran lo que demore la persona más lenta. La pregunta es: ¿existe una forma de cruzar el puente antes de que agote la batería de la linterna, que dura 15 minutos?

Implemente en Prolog los siguientes predicados:

**a)** `cruce(?Origen, ?Destino, ?T)`: existe un cruce posible (de una o dos personas, con la linterna) partiendo de `Origen` y llegando a `Destino`, que insume un tiempo `T`. Tanto `Origen` como `Destino` están representados por el término `estado(Izq,Der,Linterna)`, donde `Izq` y `Der` son listas ordenadas con las personas que están, respectivamente, a la izquierda y a la derecha del puente, y `Linterna` puede ser `left` o `right`, según el lado en el que se encuentra la linterna. Para que el cruce sea posible, la linterna debe estar del lado del lado del puente donde el cruce comienza.

```
cruce(estado([a,b,c],[d],left),estado([a,b],[c,d],right),5).
```

```
cruce(estado([a,b,c],[d],left),estado([a],[b,c,d],right),5).
```

```
cruce(estado([a,b],[c,d],right),estado([a,b,c,d],[],left),8).
```

Ayuda: para asegurar que la lista está siempre ordenada, puede utilizar el predicado `sort` de Prolog (`sort(+Lista,-ListaOrd)`: `ListaOrd` es la versión en orden lexicográfico de `Lista`).

**b)** `recorrido(?Camino,?T)`: existe una secuencia de cruces que resuelve el problema en un tiempo `T`, a partir de la configuración inicial. `Camino` es la lista de estados en la secuencia.

```
recorrido([estado([a,b,c,d],[],left),estado([a,b],[c,d],right),
```

```
estado([a,b,c],[d],left),estado([a],[b,c,d],right),
```

```
estado([a,b],[c,d],left),estado([],[a,b,c,d],right)],22).
```

**c)** `problema(?Camino)`: existe una secuencia de cruces que resuelve el problema del puente y la linterna (es decir, encuentra una secuencia de estados que resuelve el problema en un tiempo menor o igual a 15 minutos).

**d)** `recorrido2`: muestra en consola todos los caminos que resuelven el cruce (sin importar el tiempo), utilizando un mecanismo `loop-fail`.

**Ejercicio 2** [15 puntos]

**a)** Indique si son verdaderas o falsas las siguientes afirmaciones. Fundamente.

- Una cláusula de Horn contiene uno y solo un literal positivo.
- Un conjunto de fórmulas cerradas es insatisfactible si no hay ninguna interpretación posible que sea modelo del conjunto.
- Dado un programa lógico y un objetivo, el árbol SLD resultante solo puede contener ramas de éxito y ramas infinitas.
- Existe al menos un m.g.u. para todo par de expresiones.

**b)** Aplique el algoritmo para hallar el m.g.u. de las siguientes dos expresiones, mostrando su aplicación paso a paso:

```
siguiente(X,g(X,Y),a(W))
```

```
siguiente(a(Z),T,Z)
```

**Ejercicio 3** [25 puntos]

Considere el siguiente programa Prolog:

a(1).  
b(2).  
b(3).

pq(f(X),f(Y),f(Z)) :- a(X),a(Y),b(Z).  
pq(X,Y,Z) :- pq(f(Z),Z,Y).  
pq(1,1,1).

- Dibuje el árbol SLD correspondiente a la consulta  $?- p(A,B,C)$  suponiendo que la regla de computación toma el átomo de más a la izquierda para la resolución.
- ¿Qué respuestas dará el intérprete Prolog para el objetivo anterior?
- ¿Cuáles son las respuestas si cambiamos la primera cláusula del predicado pq/3 por la siguiente? Justifique.

pq(f(X),f(Y),f(Z)) :- a(X),a(Y),!,b(Z).

- ¿Qué respuestas devolvería la consulta de parte a) si el intérprete realizara un recorrido en amplitud del árbol SLD?

**Ejercicio 4** [15 puntos]

**a)** Considere un escenario donde se lanzan 3 dados, todos ellos perfectamente balanceados. Construya programas en Prolog que permitan calcular:

- La probabilidad de que los tres dados salgan iguales.
- La probabilidad de que salga una escalera (es decir, tres valores seguidos en los dados)
- La probabilidad de que salga una escalera, sabiendo que el primer dado salió 5 (utilice la capacidad de Prolog de especificar evidencia).

**b)** Utilizando DCG, defina un programa Prolog para reconocer el siguiente lenguaje:

$$L = \{a^n b^m c^n d^m / n, m \geq 0\}$$