

## Evaluación

### Ejercicio 1 [evaluación individual del laboratorio]

Responda las siguientes preguntas en base a la implementación del *Scrabblog* realizada por su grupo:

- ¿Cómo se realiza la validación de una jugada del jugador humano?
- ¿Cómo representa las letras que cada jugador tiene en el atril?
- ¿Qué consideraciones se tienen en cuenta para la manipulación de los atriles cuando la bolsa de letras está vacía y los jugadores tienen menos de siete letras?

### Ejercicio 2 [20 puntos]

- Sea  $P$  un programa lógico y  $G$  un objetivo. Indique si las siguientes afirmaciones son verdaderas o falsas. Justifique.
  - Existen programas lógicos que no tienen ningún modelo.
  - Si  $P \cup \{G\}$  es insatisfactible, entonces  $P$  es insatisfactible.
  - $\theta$  es una respuesta computada de  $P \cup \{G\} \Leftrightarrow \theta$  es una respuesta correcta de  $P \cup \{G\}$
- Dé una interpretación con dominio natural que sea modelo y otra que no lo sea, para el siguiente programa lógico:

$$\begin{aligned}
 & p(X, n, X). \\
 & p(X, Y, Z) \leftarrow p(Y, X, Z) \\
 & q(X, Y) \leftarrow p(Z1, Z2, I1), \\
 & \quad p(I1, Z3, X), \\
 & \quad p(Z2, Z3, I2), \\
 & \quad p(Z1, I2, Y). \\
 & q(X, Y) \leftarrow p(X, Z, I1), \\
 & \quad p(Y, Z, I2), \\
 & \quad q(I1, I2).
 \end{aligned}$$

### Ejercicio 3 [20 puntos]

Sea el siguiente programa *Prolog*:

$$\begin{aligned}
 \text{foo}([X|Xs], Ys, [X|Zs]) & \text{ :- foo}(Xs, Ys, Zs). \\
 \text{foo}([X|Xs], Ys, []) & \text{ :- oof}([X|Xs], Ys). \\
 \text{oof}([X|Xs], [X|Xs]). \\
 \text{oof}([X|Xs], Ys) & \text{ :- oof}(Xs, Ys).
 \end{aligned}$$

- Dé los valores de  $As$  y  $Bs$  que son solución de la consulta:  $? \text{foo}([b,a,b], As, Bs)$ .
- ¿Es posible probar la negación de  $\text{foo}(Xs, [], [])$ ? Justifique.
- Para las siguientes variantes con *cut* del programa anterior, indique los valores de  $As$  y  $Bs$  que son solución de la consulta de la parte (a). Justifique sus respuestas.

- $$\begin{aligned}
 \text{foo}([X|Xs], Ys, [X|Zs]) & \text{ :- foo}(Xs, Ys, Zs). \\
 \text{foo}([X|Xs], Ys, []) & \text{ :- !, oof}([X|Xs], Ys).
 \end{aligned}$$

$$\begin{aligned}
 & \text{oof}([X|Xs], [X|Xs]). \\
 & \text{oof}([X|Xs], Ys) \text{ :- oof}(Xs, Ys).
 \end{aligned}$$

- $$\begin{aligned}
 \text{foo}([X|Xs], Ys, [X|Zs]) & \text{ :- foo}(Xs, Ys, Zs). \\
 \text{foo}([X|Xs], Ys, []) & \text{ :- oof}([X|Xs], Ys).
 \end{aligned}$$

$$\begin{aligned}
 & \text{oof}([X|Xs], [X|Xs]) \text{ :- !}. \\
 & \text{oof}([X|Xs], Ys) \text{ :- oof}(Xs, Ys).
 \end{aligned}$$

- Indique de qué «color» es el *cut* en cada variante de la parte (c). Justifique.

**Ejercicio 4** [20 puntos]

a) Implemente el siguiente metaintérprete para *Prolog* puro:

```
resArbol(+G, -R) ← resuelve G, siendo R la correspondiente rama de prueba del árbol SLD.
```

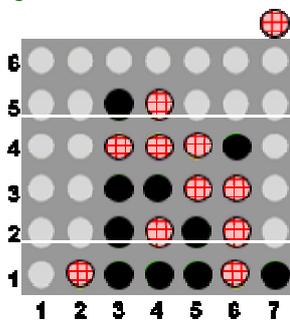
b) Defina una gramática DCG que reconozca el lenguaje de las sentencias *while*, *assign* y expresiones lógicas (operadores *and*, *or* y *not*) de un lenguaje imperativo.

Por ejemplo, el siguiente es un programa válido:

```
X: = true or Y;
while (X and Z) do
    Z: = X or Y;
    while (false) do endw;
endw;
```

Nota: Suponga que la entrada es una lista de *tokens*.

**Ejercicio 5** [25 puntos]



K-en-línea es un juego muy parecido al ta-te-ti, en donde dos jugadores alternadamente colocan sus fichas en un tablero. Gana aquel jugador que coloque primero k de sus fichas contiguas en una línea, ya sea en forma horizontal, vertical o diagonal.

Sin embargo, a diferencia del ta-te-ti, el tablero se encuentra en posición «vertical»: una ficha sólo puede ocupar una posición en una columna, si todas las filas anteriores están ocupadas por una ficha (en caso contrario, se «cae» hasta la primera fila libre).

En la figura, el juego es cuatro-en-línea en un tablero de seis por siete, y las fichas claras ganan jugando en la última columna.

Para resolver el problema en *Prolog*, se modela el tablero con el funtor **tab**:

**tab**(NFilas, NColumnas, Ocupadas)

*NFilas* y *NColumnas* son las dimensiones del tablero, y *Ocupadas* es la lista de posiciones ocupadas. Estas posiciones se representan con el funtor **oc**:

**oc**(Fila, Columna, Color)

en donde (Fila,Columna) es la coordenada de la ficha y *Color* toma el valor **n** (negro) o **b** (blanco).

Se pide implementar los siguientes predicados:

- novuelan**(+Tablero) ← *Tablero* no contiene fichas sin apoyo (no hay fichas «suspendidas en el aire»).
- siguientes**(+Tab, -Ps) ← *Ps* es una lista de posiciones *pos*(Fila,Col) en donde se puede jugar una ficha en el tablero *Tab*.
- klinea**(+Tab, +K, ?Color) ← Hay en el tablero *Tab* una línea horizontal, vertical o diagonal de *K* fichas de *Color*.
- jugada**(+T, +K, +Color, ?Pos) ← *Pos* es una «buena» jugada a realizar por el jugador *Color* en el tablero *T*. Se entiende por una buena jugada aquella que le permite obtener *K* de sus fichas en línea o que, en caso de no poder ganar, no le permite a su contrincante formar *K* en línea en el siguiente turno.

Nota: Puede utilizar, sin implementarlos, todos los predicados extralógicos, metalógicos, aritméticos y de 2do. orden vistos en el curso. Cualquier otro predicado auxiliar utilizado en la resolución del problema debe ser implementado.