

Evaluación

Duración: 2:30 hs.

Ejercicio 1 [evaluación individual del laboratorio]

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

- ¿Cómo se representa la estructura del tablero?
- ¿Cómo funciona la estrategia implementada?
- ¿Qué condiciones se tienen que cumplir para que la partida finalice?

Ejercicio 2 [20 puntos]

Sea el siguiente programa *Prolog*:

```

arbol(arb(prolog,
           arb(3,
               h(fing),
               h(4)
            ),
       arb(fing,
           arb(prolog,
               h(1),
               h(2)
            ),
           h(inco)
        )
      ).
achatable(fing).
achatable(prolog).
achata(h(X), [X|Xs], Xs).
achata(arb(X,Y,Z), [X|Ys], Rs) :-
    achatable(X),
    achata(Y, Ys, Zs),
    achata(Z, Zs, Rs).
achata(arb(X,Y,Z), Ys, Rs) :-
    achata(Y, Ys, Rs).
```

y la consulta: `?arbol(Arbol), achata(Arbol, As, []).`

- Dé los valores de *As* que son solución de la consulta. Justifique.
- Cree variantes del programa anterior, de forma tal que, agregando uno y solamente un *cut* a las cláusulas de *achata*/3, la consulta tenga por resultado:
 - una única solución
 - ninguna solución
 - las mismas soluciones
- Si se agrega un *cut* a la primera cláusula de *achatable*/1, ¿se modifica la respuesta a la consulta anterior? En caso afirmativo, muestre cuál es. En caso negativo, dé una consulta para la cual el resultado cambia.

Ejercicio 3 [20 puntos]

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

resDerecha(+G) ← resuelve *G* con la siguiente regla de computación: siempre se selecciona el átomo de más a la derecha para la sustitución.

- Construya una gramática DCG que reconozca el lenguaje $L = \{w \in \Sigma^* / w = w^R\}$ utilizando únicamente reglas DCG.

Ejercicio 4 [20 puntos]

Sea el siguiente programa lógico P:

```
arco(a, b)
arco(a, c)
camino(X, X)
camino(X, Y) ← arco(X, Z), camino(Z, Y)
```

y el siguiente objetivo G: $\leftarrow \text{camino}(a, X), \text{camino}(X, c)$. El intérprete selecciona el átomo de más a la derecha como regla de computación.

- Construya el árbol SLD correspondiente a $P \cup \{G\}$. El árbol debe tener, siempre que sea posible, al menos 3 niveles.
- Dé las soluciones que muestra el intérprete, si éste selecciona las cláusulas en el orden inverso de aparición en el programa y realiza una búsqueda DFS.
- ¿Cambia su respuesta anterior si ahora el intérprete selecciona las cláusulas en el orden de aparición en el programa?
- ¿Existe alguna respuesta correcta para $P \cup \{G\}$ que no sea computada? Justifique.

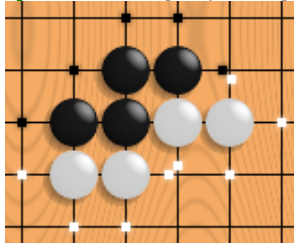
Ejercicio 5 [25 puntos]

Figura 1

El Go es un juego en donde dos oponentes luchan por ver quién gana más territorio. El juego se desarrolla en un tablero de N por N casillas, en donde las fichas —piedras blancas o negras, según el jugador— se colocan en las intersecciones.

En este juego es importante la noción de bloque. Un bloque es una secuencia de piedras conectadas de forma vertical u horizontal. Por ejemplo, en la figura 1 hay tres bloques: 1 negro y 2 blancos.

La libertad de un bloque es la cantidad de jugadas con que se lo puede extender. Por ejemplo, el bloque negro de la figura 1 tiene cinco libertades; y ambos bloques blancos, cuatro.

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 **pos**:

pos(Fila, Columna, Color)

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

Se pide implementar los siguientes predicados:

conectadas(+Tab, +Ps1, ?Ps2) $\leftarrow$ *Ps1* y *Ps2* son las coordenadas de dos piedras que pertenecen a un mismo bloque del tablero *Tab*.

libertades(+Tab, +Pos, ?N) $\leftarrow$ El bloque al que pertenece la piedra en el lugar *Pos* tiene *N* libertades.

bloques(+Tab, +Color, -Bs) $\leftarrow$ *Bs* es una lista con todos los bloques de *Color* en el tablero *Tab*. Cada bloque se representa como una lista de coordenadas.

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.