

Trabajo Práctico N° 2. Programación Lógica.

Puntuación

Puntaje Total: 100 puntos.

Aprobación: 60 puntos.

Plazo de entrega: 15/11/2021 a las 23.55 hs.

Condiciones de entrega

1. El presente trabajo práctico deberá resolverse en grupo de hasta 3 (tres) personas.
2. Entrega: Se realizará por medio del campus virtual de la UTN, en la tarea correspondiente al TP N° 2. La extensión del archivo será .zip o .rar, de acuerdo al programa de compresión usado. El nombre del archivo se consigue concatenando un prefijo del número del TP con los apellidos de los integrantes separados por guiones (Ej: Pérez y Abdala, el nombre será tp2-abdala-perez.zip). Note que no hay espacios en blanco ni acentos en el nombre de archivo. Dentro del archivo de entrega, deben constar los siguientes:
 - Fuentes SWI-Prolog: Se debe entregar un archivo denominado tp2.pl.
 - Los casos de prueba se entregarán en un archivo de texto, no deben ser capturas de pantalla. Deberán cubrir diferentes resultados que puedan obtenerse de la evaluación de los predicados solicitados. Se enfatiza que se adjunten casos de prueba que sean claros, válidos y suficientes para poder probar el trabajo. Entregar este archivo con el nombre casos-de-prueba.txt.
 - Archivo de texto (INTEGRANTES.txt) con una línea para cada integrante en la cual figure el nombre del alumno/a y su dirección de email.
4. Penalización por entrega fuera de término: Si el trabajo práctico se entrega después de la fecha indicada, y hasta una semana tarde, tendrá una quita de 15 puntos. No serán recibidos trabajos luego de una semana de la fecha de entrega. Los trabajos que se deban rehacer/corregir tendrán una quita de 30 puntos.

Descripción del problema

Conecta 4 es un juego de estrategia de información perfecta para dos jugadores, incluye un tablero vertical de tamaño fijo 6x7 (6 filas y 7 columnas) y 42 fichas.

Al inicio de la partida cada contrincante recibe 21 fichas de un color. Por turnos deben ingresar una ficha por vez en el tablero por la parte superior (Fig. 1), deslizándose por la columna hasta ocupar la última fila libre.

No hay restricciones en cuanto a qué columna puede elegir un jugador, siempre que exista un lugar disponible.

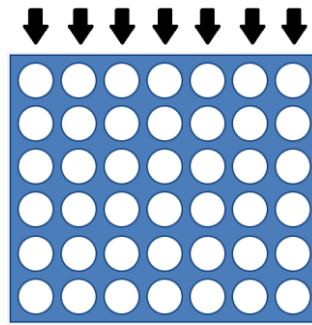


Fig. 1. Tablero e ingreso de fichas.

Cuando una ficha ocupa un casillero ya no puede realizarse ninguna acción sobre la misma, quedando en esa ubicación hasta la finalización de la partida.

La Fig. 2 muestra dos movimientos posibles. En (a) el primer jugador ingresa su ficha por la parte superior, deslizándose hasta ocupar su posición (b). Luego el segundo jugador ingresa su ficha (c). En (d) se observa el tablero luego del primer turno.

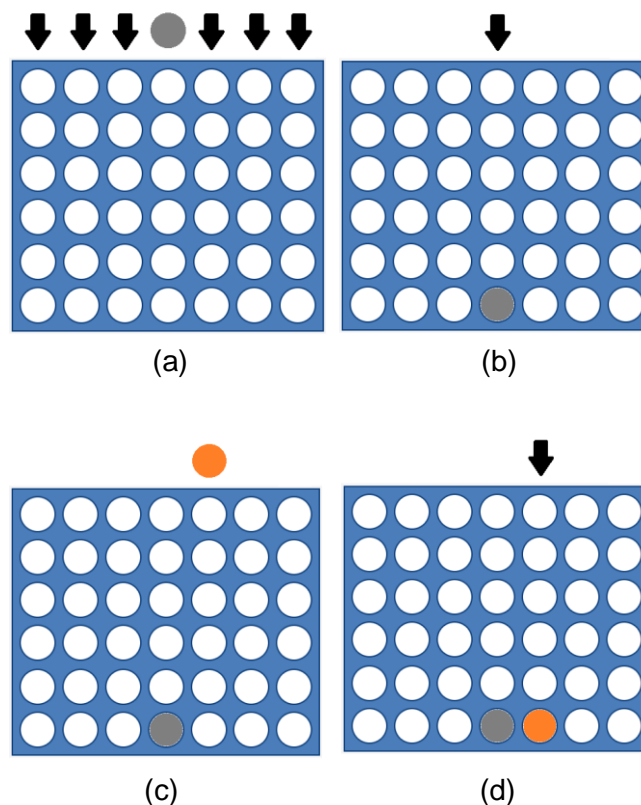


Fig. 2. Ingreso de las primeras dos fichas de la partida.

El ganador es quien primero consiga conectar (o alinear) consecutivamente al menos cuatro fichas de su color de forma horizontal, vertical o diagonal. La Fig. 3 muestra el movimiento ganador, la ficha ingresada por el segundo jugador (a) consigue conectar cuatro fichas consecutivas (b).

Si ambos jugadores ingresaron todas sus fichas y no hay ningún ganador, se considera un empate.

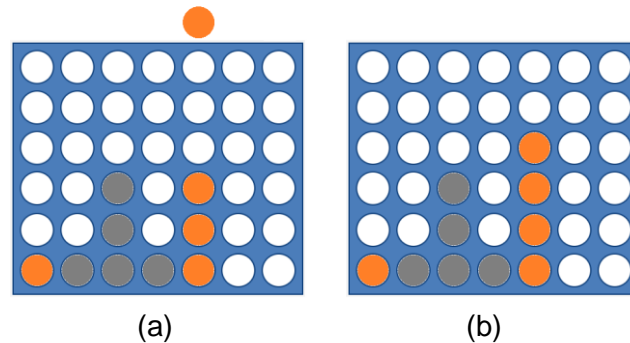


Fig. 3. Movimiento ganador de la partida.

Implementación

Se solicita escribir un programa lógico en Prolog para implementar el juego Conecta 4 de acuerdo a las reglas planteadas y las funcionalidades solicitadas a continuación.

Representación

El tablero será representado por una lista, en donde cada elemento será una sublista que corresponde a una fila. Las sublistas incluirán los siguientes valores (en lugar de colores) indicando el contenido del casillero:

- -: casillero vacío.
- a: casillero ocupado por una ficha del jugador A.
- b: casillero ocupado por una ficha del jugador B.

Ejemplos:

Tablero inicial (Fig. 1):

```
[[ -, -, -, -, -, -, - ], [ -, -, -, -, -, -, - ], [ -, -, -, -, -, -, - ], [ -, -, -, -, -, -, - ],
[ -, -, -, -, -, -, - ], [ -, -, -, -, -, -, - ]]
```

Tablero Fig. 2 (d):

```
[[ -, -, -, -, -, -, - ], [ -, -, -, -, -, -, - ], [ -, -, -, -, -, -, - ], [ -, -, -, -, -, -, - ],
[ -, -, -, -, -, -, - ], [ -, -, -, a, b, -, - ]]
```

Tablero ganador Fig. 3 (b):

```
[[ -, -, -, -, -, -, - ],
[ -, -, -, -, -, -, - ],
[ -, -, -, b, -, -, - ],
[ -, -, a, -, b, -, - ],
[ -, -, a, -, b, -, - ],
[ b, a, a, a, b, -, - ]]
```

Aclaración: Cuando los elementos de la lista se muestren encolumnados es solo a los fines de favorecer la legibilidad de los mismos.

Predicados solicitados

A continuación se indican los predicados a implementar, para todos ellos debe respetar la cantidad y uso de sus argumentos de acuerdo a la definición de los mismos.

Puede además incluir los predicados auxiliares que considere necesarios.

1) **tableroInicial(T)**

Este predicado permitirá obtener el tablero inicial de una partida. Ejemplo:

?- tableroInicial(T).

```
T = [[-, -, -, -, -, -, -],
      [-, -, -, -, -, -, -],
      [-, -, -, -, -, -, -],
      [-, -, -, -, -, -, -],
      [-, -, -, -, -, -, -],
      [-, -, -, -, -, -, -],
      [-, -, -, -, -, -, -]]
```

2) **ingresarFicha(Tablero, Columna, Ficha, Tablero2)**

Tablero unifica con un tablero, Columna con un número de columna, Ficha con una ficha del juego y Tablero2 con el tablero resultante de ingresar la ficha en la columna. Ejemplo:

?- tableroInicial(T), ingresarFicha(T, 3, a, T2).

```
T2 = [[-, -, -, -, -, -, -],
       [-, -, -, -, -, -, -],
       [-, -, -, -, -, -, -],
       [-, -, -, -, -, -, -],
       [-, -, -, -, -, -, -],
       [-, -, -, -, -, -, -],
       [-, -, a, -, -, -, -]]
```

?- tableroInicial(T), ingresarFicha(T, 5, a, T2), ingresarFicha(T2, 5, b, T3).

```
T3 = [[-, -, -, -, -, -, -],
       [-, -, -, -, -, -, -],
       [-, -, -, -, -, -, -],
       [-, -, -, -, -, -, -],
       [-, -, -, -, b, -, -],
       [-, -, -, -, a, -, -]]
```

3) **columnaDisp(Tablero, Columna)**

Tablero unifica con un tablero y Columna con un número de columna que cuenta con al menos un casillero libre.

?- tableroInicial(T), columnaDisp(T, C).

C = 1 ;

```
C = 2 ;
. . .
C = 7
?- tableroInicial(T),columnaDisp(T,6).
true
```

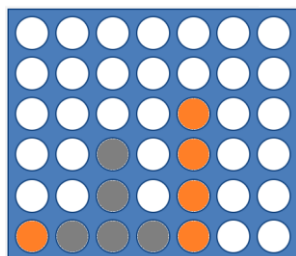
4) contenido(Tablero,Casillero,Ficha)

Tablero unifica con un tablero, Casillero con una lista de dos elementos que indica la fila y columna, y Ficha con una ficha del juego. Ejemplo:

```
?- tableroInicial(T),contenido(T,[2,4],F).
F = -
?- tableroInicial(T),ingresarFicha(T,3,a,T2),contenido(T2,[6,3],F).
F = a
```

5) conecta4(Tablero,Ficha,ListaPos)

Tablero unifica con un tablero que tiene un ganador, Ficha con la ficha del ganador y ListaPos con la lista de posiciones ganadoras. El predicado fallará en caso de no haber un ganador. Ejemplo:



```
T = [[-, -, -, -, -, -, -],
      [-, -, -, -, -, -, -],
      [-, -, -, -, b, -, -],
      [-, -, a, -, b, -, -],
      [-, -, a, -, b, -, -],
      [b, a, a, a, b, -, -]]
```

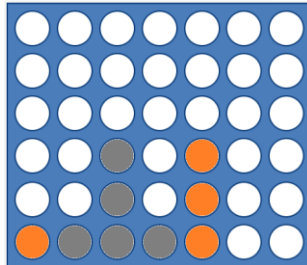
```
?- conecta4(T,F,L).
F = b
L = [[3,5],[4,5],[5,5],[6,5]]
?- tableroInicial(T),conecta4(T,F,L).
false
```

6) empate(Tablero)

Tablero unifica con un tablero que no tiene ganador y ninguna columna disponible para ingresar fichas.

7) jugadaGanadora(Tablero,Ficha,Columna)

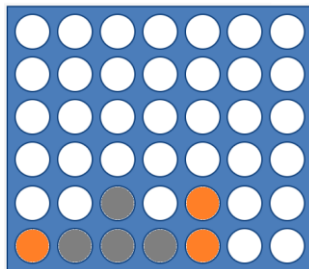
Tablero unifica con un tablero, Ficha con una ficha y Columna con la columna sobre la cual ingresar una ficha que gane el juego. Si no existe un movimiento para ganar el juego el predicado falla. Ejemplo:



```
T=[ [-,-,-,-,-,-,-],
     [-,-,-,-,-,-,-],
     [-,-,-,-,-,-,-],
     [-,-,a,-,b,-,-],
     [-,-,a,-,b,-,-],
     [b,a,a,a,b,-,-]]
```

?- jugadaGanadora(T,b,C).

C = 5



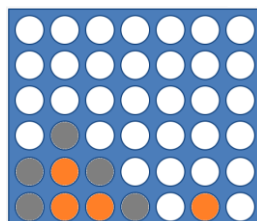
```
T=[ [-,-,-,-,-,-,-],
     [-,-,-,-,-,-,-],
     [-,-,-,-,-,-,-],
     [-,-,-,-,-,-,-],
     [-,-,a,-,b,-,-],
     [b,a,a,a,b,-,-]]
```

?- jugadaGanadora(T,b,C).

false

8) jugadaSegura(Tablero,Ficha,Columna)

Tablero unifica con un tablero, Ficha con una ficha y Columna con la columna sobre la cual ingresar una ficha garantizando que no se habilita una jugada ganadora al oponente. Ejemplo:



```
T=[ [-,-,-,-,-,-,-],
     [-,-,-,-,-,-,-],
     [-,-,-,-,-,-,-],
     [-,a,-,-,-,-,-],
     [a,b,a,-,-,-,-],
     [a,b,b,a,-,b,-]]
```

En este caso si el jugador B ingresa una ficha en la primera columna, el jugador A ganará el juego ingresando una ficha también en la primera columna (obtiene cuatro

fichas consecutivas en forma diagonal). Las jugadas seguras para el jugador B son ingresar la ficha en las columnas 2, 3, 4, 5, 6 y 7.

?- `jugadaSegura(T,b,C).`

`C = 2 ;`

`...`

`C = 7`

9) `jugadaDefinitiva(Tablero,Ficha,Columna)`

Tablero unifica con un tablero, Ficha con una ficha y Columna con la columna sobre la cual ingresar una ficha que garantice ganar el juego en el próximo turno sin importar el movimiento del oponente.

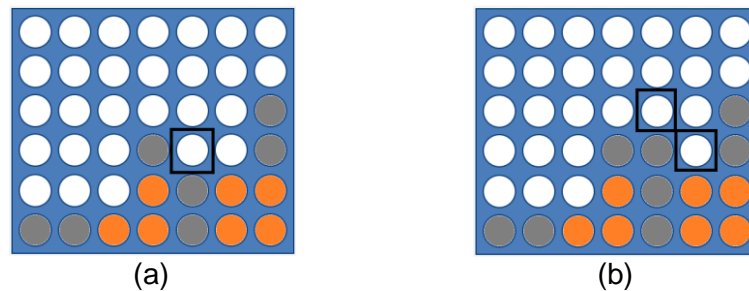
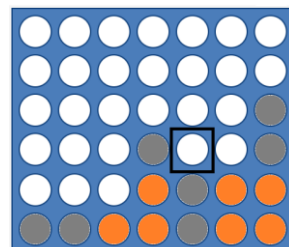


Fig. 4. Jugada definitiva.

En la Fig.4 (a) el casillero marcado muestra una jugada en la cual, si el jugador A ingresa su ficha en la columna número 5, en el próximo turno tendrá al menos una jugada ganadora sin importar la ficha ingresada por su oponente. Luego de la jugada (ingreso en la columna número 5) en la Fig. 4 (b) se observa que el jugador B no tiene jugada segura, ya que podrá elegir solo una de las columnas 5 o 6. Ejemplo:



```
T=[ [-,-,-,-,-,-],
    [-,-,-,-,-,-],
    [-,-,-,-,-,a],
    [-,-,-,a,-,-,a],
    [-,-,-,b,a,b,b],
    [a,a,b,b,a,b,b]]
```

?- `jugadaDefinitiva(T,b,C).`

`C = 5`

?- `tableroInicial(T),jugadaDefinitiva(T,a,C).`

`false`