

Programación Lógica

Laboratorio 2 - 2015

**Facultad de Ingeniería
Instituto de Computación
Grupo de Procesamiento de Lenguaje Natural**

El objetivo de este obligatorio es implementar un juego de mesa en Prolog y una inteligencia artificial para que lo juegue.

Nota previa - IMPORTANTE

Se debe cumplir íntegramente el “Reglamento del Instituto de Computación ante Instancias de No Individualidad en los Laboratorios”, disponible en:

https://www.fing.edu.uy/sites/default/files/2011/3090/proc_casos_copia.pdf

En particular está prohibido utilizar documentación de otros grupos o de otros años, de cualquier índole, o hacer público código a través de cualquier medio (EVA, correo, papeles sobre la mesa, etc.).

Juego de los ceritos

El *juego de los ceritos*, también conocido como *cuadritos*, *timbiriche* [1], o en inglés *dots and boxes* [2], es un juego de estrategia por turnos para dos jugadores que se juega sobre una grilla de tamaño $N \times N$ (aunque también existen variantes rectangulares), donde el objetivo es capturar la mayor cantidad de casilleros o celdas de la grilla.

Se comienza con una grilla vacía donde N representa la cantidad de vértices, por lo tanto la grilla contará con $(N-1) \times (N-1)$ casilleros. En cada turno, un jugador puede elegir para marcar una línea (borde) de cualquier casillero que no haya sido marcada hasta el momento. Luego le toca el turno a otro jugador, y se irán alternando marcando líneas (ver Fig. 1).

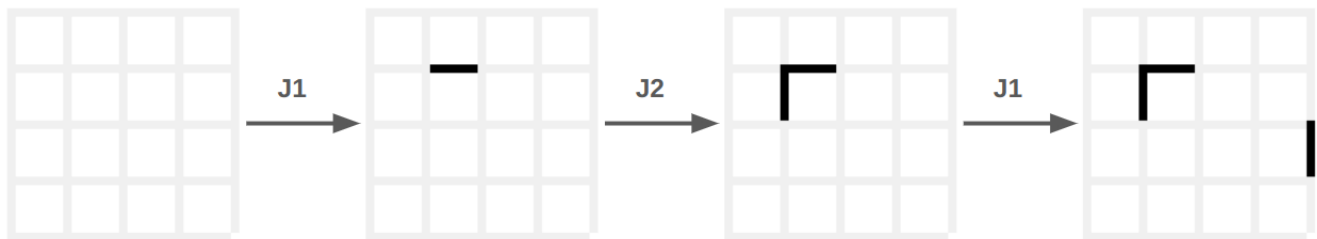


Fig. 1: La grilla de tamaño 5 comienza vacía, en el primer turno el jugador 1 marca la línea 2-2-h (veremos esta notación de coordenadas de la jugada más abajo), en el segundo turno el jugador 2 marca la línea 2-2-v, y en el tercer turno el jugador 1 marca la línea 3-5-v.

Cuando un jugador marca la última línea que rodea un casillero, se dice que capturó el casillero y le pone su nombre (ver Fig. 2). Ese casillero ahora es un punto para el jugador. Notar que no importa qué líneas marcó cada jugador, el casillero será capturado por el jugador que dibuje la última línea que lo cierra. Además, es posible que se cierre más de un casillero a la vez marcando una sola línea (máximo 2). Cada vez que un jugador captura un casillero, tiene un turno más para marcar otra línea, y esta operación se repite cada vez que capture un casillero, por lo que pueden formarse “cadenas” de cerrado de casilleros.

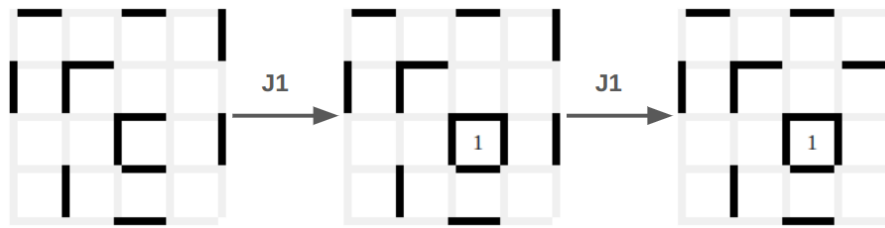


Fig. 2: El jugador 1 marca la línea 3-4-v, lo cual completa el casillero (3,3). En ese momento el jugador marca el casillero con su número. Como completó un casillero, juega un turno más eligiendo la línea 2-4-h.

El juego finaliza cuando todos los casilleros fueron capturados por alguno de los jugadores (o lo que es lo mismo, todas las líneas están marcadas), y gana el jugador que tenga más casilleros capturados (ver Fig. 3).

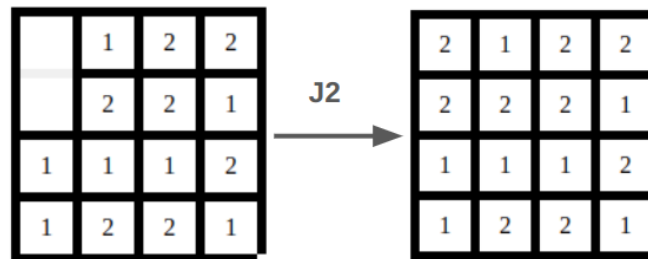


Fig. 3: El jugador 2 marca la última línea, 2-1-h, capturando dos casilleros y finalizando el juego. En este caso gana el jugador 2, ya que capturó en total 9 casilleros, contra 7 del jugador 1.

Representación del tablero

En nuestra implementación del juego, consideraremos que el tablero es una matriz de tamaño $N \times N$, donde cada celda contendrá información sobre las líneas marcadas y si la celda fue capturada. La forma de referirse a cada línea de la grilla será mediante una tupla de tres componentes F-C-D, donde F es la fila, C es la columna y D es la dirección *h* o *v*, dependiendo de si es la línea horizontal superior o vertical izquierda (ver Fig. 4).

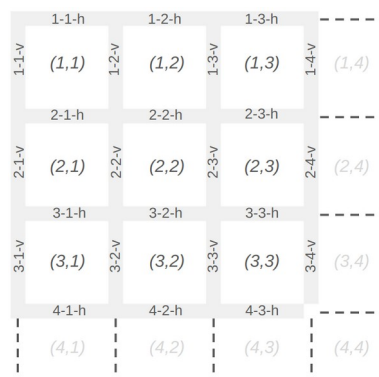


Fig. 4: Tablero de tamaño 4, indicando las coordenadas de todas las líneas válidas y todas las celdas. Notar que en realidad las filas y columnas válidas para el tablero de juego son hasta $N-1$. La fila N y la columna N solo se utilizan para representar el borde, pero no son casilleros capturables.

Por lo tanto, para capturar el casillero (F,C) , deben estar marcadas las siguientes cuatro líneas: **F-C-h**, **F-C-v**, **(F+1)-C-h**, **F-(C+1)-v**. De esta manera hay que tener en cuenta que en la matriz de $N \times N$ celdas, solo $(N-1) \times (N-1)$ celdas son capturables, y las restantes se utilizarán solo para poder marcarles una línea de borde horizontal o vertical.

Los estudiantes pueden elegir la forma de implementar la matriz en Prolog, pero por motivos de

eficiencia se sugiere utilizar una matriz de funtores, donde cada celda sea una 3-tupla $c(H,V,J)$ con información de si su borde horizontal superior (H) o vertical izquierdo (V) está marcado, y si la celda está capturada por algún jugador (J).

Requerimientos a implementar

Se debe implementar en Prolog el juego de los ceritos, denominado *Ceritolog*, para dos jugadores y cualquier tamaño de tablero N. Cada jugador puede estar controlado por el humano o la computadora (máquina). Para eso se proveerá una implementación gráfica básica en HTML+JS que invoca a los predicados Prolog a implementar. Los predicados son los siguientes:

tablero(+N, ?Tablero) ← Devuelve un tablero de tamaño N vacío, o sea una matriz que representa un tablero vacío de juego como la descrita anteriormente.

fin_del_juego(+Tablero,?P1,?P2,?Ganador) ← Dado un tablero, el predicado es verdadero si el tablero representa un juego finalizado, y devuelve la cantidad de puntos del jugador 1 en P1, la cantidad de puntos del jugador 2 en P2, y un string que indica si alguno ganó, en el formato: “Gana el jugador 1”, “Gana el jugador 2”, o “Empate”. En caso de que no sea el fin del juego, el predicado falla.

jugada_humano(+Tablero,+Turno,+F,+C,+D,?Tablero2,?Turno2,?Celdas) ← Se le envía un tablero, de quién es el turno (1 o 2) y la línea elegida por el jugador humano con las variables F-C-D, y devuelve: el tablero modificado con la línea marcada (y celdas marcadas en caso de que sea necesario), de quién es el siguiente turno (Turno2), y una lista de celdas que se capturaron con esta acción en formato [Fila,Columna]. Notar que esta lista puede tener 0, 1 o 2 celdas como máximo para cualquier acción. Si el movimiento que se quiere hacer no es válido, el predicado falla.

jugada_maquina(+Tablero,+Turno,+Nivel,?F,?C,?D,?Tablero2,?Turno2,?Celdas) ← Se le envía un tablero y de quién es el turno (1 o 2), y debe elegir una jugada a realizar por el jugador controlado por la computadora. Esta debe utilizar una estrategia *buena y eficiente*. Se recomienda utilizar el algoritmo minimax [3] para implementar la estrategia de la máquina, y el nivel de minimax a utilizar se indicará en el argumento Nivel. El predicado devuelve: el tablero modificado luego de la jugada, de quién es el siguiente turno (Turno2), y una lista de celdas que se cerraron con esta acción en formato [Fila,Columna], de la misma forma que en el predicado anterior.

sugerencia_jugada(+Tablero,+Turno,+Nivel,?F,?C,?D) ← Utiliza la estrategia de minimax para calcular una buena jugada para sugerirle a un jugador humano.

El algoritmo minimax es la estrategia sugerida para resolver este juego, sin embargo los estudiantes tienen la libertad de probar también con otras estrategias, ya sea modificando el minimax, escribiendo reglas heurísticas, usando estrategias híbridas, o cualquier otra técnica. La idea es que intenten construir el mejor jugador de Ceritolog que puedan, *siempre que el tiempo de espera para calcular la siguiente jugada no sea mayor a 10 segundos en un tablero de tamaño 6*. O sea: para aprobar el laboratorio se necesita implementar *al menos* la estrategia minimax.

Insumos

Se proveen los siguientes archivos:

- **server_ceritolog.pl**: Contiene todos los predicados de manejo del servidor web que controla las invocaciones de la web hacia Prolog. Al ejecutarlo con 'prolog server_ceritolog.pl' o 'swipl server_ceritolog.pl' levantará un servidor web en el puerto 9999. Luego se puede ir al browser y acceder <http://localhost:9999/> o <http://localhost:9999/static/ceritolog.html> para iniciar el juego.
- **ceritolog.pl**: Módulo principal de la solución. Se provee los predicados vacíos con el fin de que la implementación sea completada por los estudiantes para resolver el laboratorio.
- **ceritolog.html, .js, .css**: Implementación del cliente web de ceritolog que invoca los predicados

Prolog a implementar para resolver el juego.

Observaciones

La implementación debe realizarse de manera que pueda ser ejecutada en la plataforma SWI-Prolog.

Forma de entrega

La entrega se realizará a través del espacio eva del curso. Se debe entregar un solo archivo '**grupo##.zip**', donde **##** es el número del grupo que realiza la entrega, conteniendo todos los módulos de la solución y el informe, ítems indicados en el apartado **Entregable**.

Fecha de entrega

Los trabajos deberán ser entregados siguiendo el procedimiento descrito anteriormente antes del viernes 13/6/2025 a las 23:55, sin excepciones. No se aceptará ningún trabajo pasada la citada fecha.

Entregable

El archivo a entregar debe contener:

1. Implementación del juego Ceritolog, con todos los módulos necesarios para la ejecución del juego. Incluir también el módulo **servidor_ceritolog.pl** y los archivos del cliente.
2. Informe en formato pdf detallando la estructura de los módulos, los predicados principales y las decisiones de diseño tomadas.

Referencias

- [1] [https://es.wikipedia.org/wiki/Timbiriche_\(juego\)](https://es.wikipedia.org/wiki/Timbiriche_(juego))
- [2] https://en.wikipedia.org/wiki/Dots_and_boxes
- [3] <http://en.wikipedia.org/wiki/Minimax>