

Proyecto II

(jueves semana 5 \leadsto miércoles semana 7)

El objetivo de este proyecto es que Ud. desarrolle en Prolog una implementación del juego Batalla Naval (*Battleship*), versión solitario. En esta versión Ud. debe proveer las posiciones de los barcos, y su solución debe generar una secuencia de jugadas que conlleven a hundir todos los barcos. Los barcos pueden estar ubicados de forma vertical u horizontal, pudiendo estar en posiciones contiguas a otros barcos, pero no solaparse. Se entiende que los movimientos realizados por su solución no pueden usar la información provista por Ud. más allá de para verificar que los movimientos realizados han logrado dañar un barco, o informar que los movimientos realizados han logrado hundir un barco, o conocer la información sobre número de barcos totales y dimensiones de cada uno de ellos (pero NO su ubicación).

Para resolver este problema Ud. debe implementar los siguientes predicados:

- **jugar**: se satisface una vez se ha obtenido las dimensiones del tablero, el número de balas disponibles y la información de los barcos, se ha almacenado en la base de datos la información de los barcos, y se han mostrado paso a paso cada uno de los movimientos realizados por su programa hasta conseguir hundir todos los barcos colocados con el número de balas disponibles. Puede que la secuencia de movimientos incluya movimientos “hacia atrás” en caso de que se agoten las balas antes de alcanzar un estado final.
- **tableroInicial(F, C, T)**: se satisface si T representa un tablero de F filas y C columnas donde todas sus posiciones corresponden a *agua* (posiciones donde no se sabe si existe o no barcos).
- **mostrarTablero(T)**: muestra en pantalla una representación clara del estado actual del tablero T, diferenciando claramente las casillas con *agua*, *fallos*, *barcos golpeados*, y *barcos hundidos* (esto de acuerdo al formato indicado más adelante).
- **estadoFinal(T)**: se satisface si en T se han hundido todos los barcos originalmente ocultos.
- **ataque(T0, T1, F, C)**: se satisface los tableros T0 y T1 corresponden a tableros con F filas y C columnas, y T1 corresponde a realizar un disparo sobre el tablero T0 (sólo difieren en una casilla).

Ud. puede definir otros predicados que considere necesarios al momento de construir su solución.

Ud. puede decidir cuál estrategia de ataque utilizar, sin embargo considere que su solución debe poder lidiar con tableros de cualquier dimensión (variable entre 1x1 y 20x20), y cualquier número de barcos (de cualquier tamaño) que sean colocados en posiciones apropiadas del tablero, en un tiempo razonable.

Este proyecto involucra entrada-salida y modificación de la base de datos durante ejecución, sin embargo se recomienda que no espere a ver estos temas en clase para comenzar a trabajar en su solución. Ud. puede comenzar a trabajar en los predicados **tableroInicial**, **estadoFinal** y

ataque así como en su esquema de solución. Para probar que sus predicados funcionan apropiadamente puede cargar un archivo con información de los barcos o usar un predicado auxiliar que tenga como argumento la información de los barcos (esto sólo para sus pruebas). Adicionalmente se recomienda comience construyendo una versión que considere un número ilimitado de balas, y luego haga las modificaciones apropiadas para considerar el número de balas, y usando el *backtracking* realizado por Prolog se obtenga una solución que use como máximo el número de balas indicado, si este número no es suficiente el predicado `jugar` debe fallar.

Representación del Tablero

El tablero se representará con una lista de listas, cuyos elementos están en $\{a, h, f, g\}$. Donde `a` representa *agua*, `h` representa una casilla perteneciente a un *barco hundido*, `f` representa un *fallo*, y `g` representa una casilla perteneciente a un *barco golpeado* (pero no hundido aún).

Detalles de la entrega

La fecha de entrega de este proyecto es hasta el **miércoles 2 de noviembre** hasta las 12 de la medianoche, y debe realizarse a través de **Aula Virtual**, para ello suba los archivos a los documentos asociados a su grupo.

Su implementación debe ser realizada en **Swi-Prolog**, y debe ser ejecutable en cualquiera de las maquinas del ldc usando **swipl**.

La entrega se realizará de forma digital, y debe incluir los archivos fuente de su proyecto debidamente documentados, versión de impresión de sus archivos fuentes (en formato PDF). Si considera necesario incluir alguna aclaratoria sobre su solución incluyala como comentarios en los archivos fuente de su solución.

La implementación que Ud. realice debe cumplir con todos los principios de programación lógica, el que su código esté bien escrito y siga estos principios será parte **importante** de su evaluación.

Cualquier posible aclaratoria o error detectado en este enunciado será anunciado en la página del curso.

Ejemplo

?- `jugar`.

Num. de Filas: **3**.

Num. de Columnas: **5**.

Cant. de Barcos: **3**.

Informacion de Barco:

Tamano: **3**.

Direccion: **v**.

Fila Inicial: **0**.

Columna Inicial: **1**.

Informacion de Barco:

Tamano: **2**.

Direccion: **h**.

Fila Inicial: **0**.

Columna Inicial: **3**.

Informacion de Barco:

Tamano: 1.

Direccion: h.

Fila Inicial: 2.

Columna Inicial: 3.

Cant. de proyectiles disponibles: 10.

a a a a a

a a a a a

a a a a a

a g a a a

a a a a a

a a a a a

a g a a a

a g a a a

a a a a a

a h a a a

a h a a a

a h a a a

a h a a a

a h f a a

a h a a a

a h a a a

a h f a a

a h f a a

a h a a a

a h f a a

a h f h a

a h a a a

a h f a f

a h f h a

a h a g a

a h f a f

a h f h a

a h a g a

a h f f f

a h f h a

a h a h h
a h f f f
a h f h a