

Pràctica complementària de Haskell.

Tres en ratlla

1 Presentació

Considerem el conegut joc del “tres en ratlla” en la seva variant en la que cada un dels dos jugadors disposa de tres fitxes d’un mateix color. Hi ha un color per a cada jugador.

Els tauler és de tres per tres i només pot haver una fitxa per casella. Hi ha dos tipus de moviments:

- Inicials. Posar un fitxa al tauler que encara no havíem posat.
- Canvi (només es realitzen quan les tres fitxes del jugador són al tauler). Permet moure una de les seves fitxes a una altra posició (diferent) que estigui lliure (no ocupada per cap altre fitxa).

Els jugadors alternen els seus moviments fins que un aconsegueix formar una línia en vertical, horitzontal o en diagonal amb 3 fitxes del seu color.



2 Es demana

Fer un programa Haskell que admeti dues formes de joc

1. L'usuari juga contra el programa.
2. Simular una partida del joc entre dos jugadors.

La primera opció llegeix els *moviments* de l'usuari i dona els seus alternativament fins que acaba la partida. En la segona opció s'han de mostrar els moviments dels dos jugadors d'un cop i el resultat de la partida.

En els dos casos s'ha de poder limitar el nombre de rondes. Si no hi ha guanyador en les rondes indicades el resultat és empat.

Per a definir la forma de jugar del programa s'usarà una *estratègia*, que és una funció que rep la informació de l'estat i el jugador a qui toca moure i retorna l'acció a fer. Noteu que en els moviment *inicials* només es posen fitxes i en els de *canvi* s'ha de treure i posar una fitxa.

Una partida simulada necessita el nombre de rondes (o 0 si és il·limitada) i les dues estratègies que usarà el programa. Noteu que aquests tipus de partida no requereix cap interacció amb l'usuari.

Una partida contra l'usuari necessita el nombre de rondes (o 0 si és il·limitada) i l'estratègia que usarà el programa. En aquest cas s'ha d'usar l'entrada/sortida per introduir els moviments de l'usuari.

Cal que programeu diverses estratègies, com a mínim una d'aleatòria i una mica intel·ligent. Es valorarà la qualitat de les estratègies.

Per a obtenir nota d'aquesta part és indispensable implementar la simulació de partida. Per obtenir la màxima nota heu de fer els dos tipus de partida i que es puguin escollir les estratègies d'una llista (posant, per exemple, un número). Podeu separar els dos tipus de partides en dos programes Haskell diferents o fer-ne un de sol que inclogui les dues funcions.