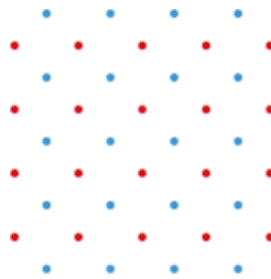


Pràctica complementària de Haskell.

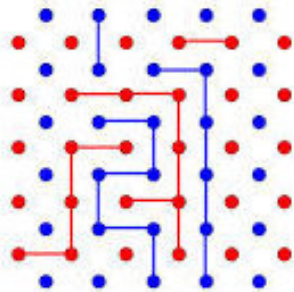
El Bridgit

1 Presentació

Considerem el joc de paper i llapis del Bridgit (creat per David Gale, a finals dels anys 50). Es parteix d'una quadricula amb punts de dos colors posats com s'indica a la figura. Noteu que hi ha els mateixos punts de cada color.



Cada jugador té un color. Un dels dos comença i es van alternant els moviments. Cada moviment consisteix en unir dos punts del color del jugador que el fa i que siguin adjacents horitzontalment o vertical, però no en diagonal. No està permès creuar una línia ja dibuixada al taulell. El joc acaba quan un dels jugadors aconsegueix construir una línia contínua que uneixi dos costats oposats del tauler, un de dalt a baix (en l'exemple el blau) i l'altre de dreta a esquerra. En el següent exemple, el blau guanyarà si uneix els dos punts que li falten en la segona fila.



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: qui ha guanyat i els punts que formen la línia que uneix els extrems. No pot haver empat.

En els dos casos s'ha de definir inicialment la mida de la quadricula, indicant el nombre de files i de columnes.

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 (i alguna cosa més, si us cal). Per tant, en la vostra implementació, la funció *estratègia* s'ha de passar per paràmetre a la funció que juga.

Una partida simulada necessita la mida de la quadrícula i les dues estratègies que usará el programa. Noteu que aquests tipus de partida no requereix cap interacció amb l'usuari.

Una partida contra l'usuari necessita la mida de la quadrícula i l'estratègia que usará el programa i qui comença. 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 altra 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).

Heu d'entregar un únic programa amb una opció inicial per triar el tipus de partida. Si només heu implementat la partida simulada, el programa ha d'escriure un missatge inicial que ho indiqui.

La pràctica s'ha de poder compilar amb `ghc` i l'executable que es generi s'ha de poder executar per línia de comandes i tenir les funcionalitats que s'han descrit anteriorment.