

INFORME PRÀCTICA HASKELL

Introducció

Per la realització d'aquesta pràctica en Haskell i amb el paradigma funcional hem implementat un verificador de partides d'escacs que segueix el següent procediment:

1. Demanem el fitxer que conté els moviments que s'han fet durant la partida.
2. Mentre hi hagi moviments:
 - a. Llegim el moviment i comprovem que la notació és correcta.
 - i. Si la notació és correcta, invoquem el mètode *fesJugada* amb el tauler actual i la jugada que hem llegit del fitxer.
 1. Si la jugada és legal, modifiquem el tauler actual.
 2. Altrament, llencem una excepció "jugada il·legal".
 - ii. Altrament, llencem una excepció "notació incorrecta".
 - b. Modifiquem la variable que ens indica de qui és el torn.
3. Fi mentre.
4. Mostrem el guanyador.

No obstant, el verificador que hem implementat no està complet, sinó que només accepta els moviments bàsics del joc ni tampoc detecta si la partida ha acabat en taules. Per tant, la nostra nota màxima és un 8.

Per facilitar la correcció del codi i també la seva claredat, l'hem separat en diferents mòduls.

Descripció dels tipus de dades

Els tipus que hem fet servir són:

- Color

El Color és un tipus molt senzill amb només dos valors, Blanc i Negre, per saber a quin bàndol pertany la peça o qui li toca jugar. Deriva d'*Equals* per poder fer comparacions.

- TipusDePeça

El TipusDePeça té una funció molt semblant al Color ja que ens permet diferenciar les peces d'escac. Poden ser: P de peó, T de torre, C de cavall, A d'alfil, D de dama i R de rei. Deriva d'*Equals* i sobreescriu el *Show*.

- Peça

El tipus Peça ja és més interessant: conté no només el Color i el TipusDePeça, sinó també la posició d'aquesta. Deriva d'*Equals* i sobreescriu el *Show*: les peces blanques les mostra en majúscula i les peces negres en minúscula.

Un aspecte interessant a comentar de la nostra implementació de Peça és que hem definit la possibilitat de que una Peça sigui “Buida”, per simplificar algunes funcions que podem trobar a Tauler com són *pecaA* o el propi *Show*.

- Tauler

Com s’ens va exigir, el Tauler és un tipus que té una llista de peces. No hem d’incloure les posicions perquè cada Peça ja sap on es troba. Sobreescrivim la instància *Show* per mostrar el tauler per pantalla amb les negres sempre a dalt i les blanques a baix. No hem fet que el tauler hereti d’*Equals* perquè no hem implementat una detecció de taules.

- Jugada

Jugada és el tipus central de dades, juntament amb el Tauler, ja que s’encarrega de tota la lògica pròpiament del joc d’escacs. Una Jugada és una Peça, que té la Posició origen, i una Posició destí.

- Partida

L’últim tipus de dada que hem fet servir és la Partida que guarda l’estat de la partida que s’està jugant, això vol dir, el tauler i qui juga.

A més, també hem declarat un sinònim Posició per la parella (Integer, Integer) on el primer element és la fila i el segon element la columna. S’ha de tenir en compte que la posició (1,1) és la cantonada inferior esquerra del tauler i la posició (8,8) és la cantonada superior dreta del tauler.

Llistat del codi

Hem separat el codi en diferents mòduls per facilitar la correcció del codi i també la seva claredat.

- Mòdul Posició

Està al fitxer “posicio.hs” i inclou el *type* Posició i la funció *valida* que retorna cert si la posició que se li passa per paràmetre està dintre del tauler o fals altrament.

- Mòdul Color

Està en el fitxer “color.hs” i inclou el *data* Color i la funció *contrari* que rep per paràmetre un color i retorna el contrari d’aquest. És a dir, si rep Blanc, diu Negre; i al revés.

- Mòdul Peça

Es troba en el fitxer “peca.hs” i inclou el tipus `TipusDePeca` i el tipus `Peca`. Té diverses funcions que són:

- *Show* de `TipusDePeca`.
- *Show* de `Peca`.
- *posicioInicial* que rep una peça i retorna cert si es troba a la posició inicial o fals altrament. L'utilitzem per saber si un peó té disponible el moviment doble.
- *sonDelMateixColor* que rep dos peces i retorna cert si són del mateix color, fals altrament.

- Mòdul Tauler

Està en el fitxer “tauler.hs” i té el tipus `Tauler` que guarda l'estat actual del tauler de joc. Aquest ja és un mòdul més complex i interessant on podem trobar la implementació de les següents funcions per mostrar el tauler:

- *Show* de `Tauler`.

Per inicialitzar un tauler:

- *taulerBuit*.
- *taulerInicial* amb totes les peces a les seves posicions inicials.

Per consultar l'estat del tauler:

- *pecesDeUnColor* que rep un tauler i un color i retorna una llista de totes les peces d'aquell color actualment en joc.
- *pecaA* que rep un tauler i una posició i retorna la peça que ocupa aquella posició en el tauler. En cas de que no hi hagi cap retorna una peça “Buida”.
- *posicioRey* que rep un tauler i un color i retorna la posició del rei d'aquell color.
- *casellaBuida* que rep un tauler i una posició i retorna cert si aquella posició està buida, fals altrament.
- *alguEntre* que rep un tauler i dos posicions i retorna cert si el moviment lineal, per torre, alfil, dama i sortida doble de peó, està bloquejat per una altre peça, fals altrament.

Per modificar l'estat del tauler:

- *eliminarPeca* que rep un tauler i una posició i retorna un tauler sense la peça que ocupa aquella posició.
- *afegirPeca* que rep un tauler i una peça i retorna el tauler amb la nova peça.
- *modificarPosicioPeca* que rep un tauler, una peça i una posició destí i retorna el tauler amb la posició d'aquella peça modificada.

Seguidament explicarem breument com hem implementat *alguEntre*:

- Una vegada hem identificat la direcció i el sentit en el que s'ha de moure la peça, mirem totes les posicions entre la posició origen i la posició destí, que generem amb una llista de compressió, i retornem cert si alguna d'aquestes cel·les està ocupada.

- Mòdul Jugada

El mòdul Jugada es troba al fitxer “jugada.hs” i conté tota la lògia pròpiament dita del joc d’escacs, des de les regles d’aquest al codi necessari per saber si una jugada és legal o no. A més de les funcions que ens permeten saber si el rei està en escac o escac i mat.

- *fesJugada* que rep un tauler i una jugada i retorna el tauler modificat si la jugada és legal, altrament llença una excepció.
- *jugadaLegal* que rep un tauler i una jugada i retorna cert si la jugada és legal, fals altrament.
- *movimentLegal* que rep un tauler i una peça i retorna totes les posicions a les que aquella peça pot arribar legalment, assegurant-se que després del seu moviment el rei no està amenaçat mirant recursivament tots els moviments legals del contrari.
- *moviment* que rep una peça i retorna totes les posicions a les que aquella peça pot arribar en un tauler buit.
- *posicionsDesde* que rep una posició, un enter que indica quantes caselles pot moure d’una vegada la peça, i una llista de direccions en les quals es pot moure i retorna totes les posicions des de les quals es pot arribar.
- *posicionsDesdeCavall* que rep una posició i una llista de direccions a les que pot arribar i retorna aquelles que són vàlides. És una funció especial per la peça de tipus Cavall.
- *Escac* que rep un tauler i un color i retorna cert si el restà amenaçat en aquell tauler, fals altrament.
- *escacIMat* que rep un tauler i un color i retorna cert si aquell color ha rebut un escac i mat, fals altrament.
- *movimentLegalNoComplet* que rep un tauler i una peça i retorna totes les posicions que aquella peça pot arribar legalment, sense tenir en compte si el seu rei estarà amenaçat després del moviment. Fem servir aquesta versió menys segura per veure si el moviment d’una peça pot provocar escac o escac i mat en el següent torn.

Explicarem la funció *jugadaLegal* i *escacIMat* i el que hauríem de fer en els dos casos per solucionar els errors.

- Per saber si una jugada és legal, primer generem una llista amb totes les posicions finals legals d’aquella peça en aquell tauler i busquem en aquesta llista la posició destí que ens passen com a paràmetre. Si la trobem, és legal, altrament no ho és i llencem excepció. Per fer aquesta feina, tenim diferents funcions que ens ajuden i que estan estretament relacionades: primer, trobem totes les posicions dintre del tauler en el que una peça pot arribar en un tauler buit, és la funció *moviment*; després, passem diferents filtres a aquesta llista per eliminar aquelles que no són accessibles. Per exemple, aquelles posicions en les que hi ha una peça enmig (això no ho hem de fer pel Cavall) o en les que la posició està ocupada per una peça del mateix color. En aquest punt també afegim les posicions de captura de peó si existeix la possibilitat. Per últim, filtrem les posicions a les que podem arribar per a aquelles que no provoquen que el nostre rei està amenaçat.

- Per la seva banda, en la funció *escac/Mat* seguim el procediment següent: primer mirem si el rei està en escac en la posició actual, després generem una llista amb tots els moviments legals de les peces del mateix color i que impedeixen l'escac. Si aquesta llista està buida i el rei està en escac en la posició actual, llavors és escac i mat.

- Mòdul Partida

El mòdul Partida es troba en el fitxer “partida.hs” i és un mòdul molt reduït que ens permet guardar l'estat de la partida. Té dos funcions, la primera per inicialitzar la partida, sempre començaran les blanques, *iniciPartida*; i la segona per canviar el jugador que juga, *canviJugador*.

Execució del codi