CIS 194 – Introduction to Haskell

April 2012

Team: Alen Kubati, Sanjay Paul, Jiten Suthar

Final Project: Chesskell

1. Introduction

Chesskell is an entirely Haskell-based implementation of chess, incorporating the standard rule set of the game. Our implementation will begin by devising the components that will enable standard gameplay mechanics and a full match to be played between two human opponents. We intend to proceed from there to the development of a basic AI to facilitate human v. machine matches.

1. Architecture
   1. Board and Piece Representation

The board will be represented via a HashMap (Data.HashMap) with 64 key values to represent each of the possible spaces on the board. The HashMap will contain data type Piece as the value for the mapping. The Piece data type allows representation of all possible chess pieces as well as the empty space. It is worthwhile to note that, in building the AI, we will also construct and update an inverted mapping of Pieces to coordinates.

* 1. Control Flow

The game begins with a main function that prompts user to select one of four playing modes (human v. human, human v. m1, human v. m2, or m1 v. m2). So far, we’ve put the architecture of the human v. human game play in place. Specifically, the main function calls a humanVsHuman function which is the main game loop logic.

humanVsHuman proceeds by printing the board (printBoard), prompting the user for an action and translating it to a valid Action type (getAction), and then finally processing that action (processAction). The final function of the series, processAction, recursively calls the main game loop logic function, humanVsHuman, if the Action requires it.

As of now, we have fully implemented all of the logic and data structures except for getAction and processMove (a sub-function of processAction). We expect implementation of both these functions to be fairly challenging as it will involve checking validity of moves, CheckMates, Castling, etc. However, we believe we have overcome a significant hurdle in working out the high-level program logic and architecture.

If time permits, we would also like to tackle the AI portion as well.