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 mutable HashMap (Data.HashMap) with 64 key values to represent each of the possible spaces on the board. This will be coupled with a function to convert coordinates to values that key into the map to retrieve the value corresponding to the board space. The HashMap will, furthermore, 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. Rule Enforcement

Each individual Piece will maintain a rule-set dictating how to build a legal set of possible moves at a given location. Some difficulties that arise in this regard are edge moves (e.g. castling) and other such variable movement options (e.g. the pawn’s option to hop on first move).

* 1. Control Flow

The game’s control flow, in keeping with the functional paradigm, is governed superficially by a recursive loop that proceeds from initiation until a specific state (i.e. checkmate) is achieved. Input-dependent functions are called per iteration to simulate the equivalent of a player turn. Proceeding in sequence, the turn consists of obtaining the desired player move, converting it to a game-based action, evaluating the legality of the move, making the necessary the adjustments to the board, and printing the board to the screen on completion. The board’s state itself is passed along the recursive chain and updated on each iteration.