A Reinforcement Learning Approach for Solving KRK Chess Endgames

Zacharias Georgiou S3009874
Evangelos Karountzos S2705532
Matthia Sabatelli S2847485
Yaroslav Shkarupa S3017125

November 19, 2015

The game of chess has always been an important subject in Artificial Intelligence. The most famous examples of this go from The Turk, a mechanical automaton chess player constructed in the late 18th century to the more recent super computer Deep Blue that defeated one of the strongest players of all time, Garry Kasparov in 1997.
The goal of our project is to have an insight on this research area by focusing on a particular moment of a chess game called the endgame. In fact the idea is to build a set of agents able to solve the KRK endgames by using reinforcement learning to improve their performances on the board. A KRK endgame regards a particular position on the chess board in which only 2 type of pieces appear, these are namely the 2 Kings and 1 Rook, an example of it is shown in the figure hereafter:

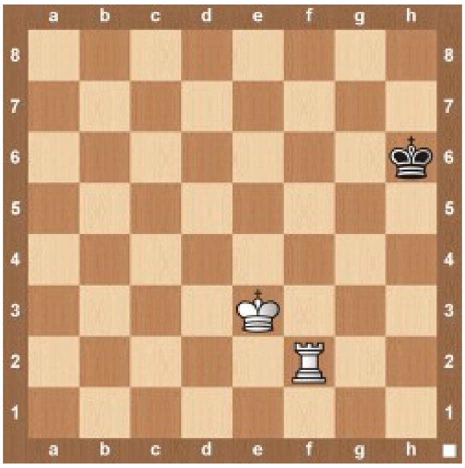

Figure 1: Example of a KRK endgame

These positions can be either wins, defeats or draws and a dataset of all these positions can be found online at the UCI Machine Learning Repository where each possible position has its relative ending state. In our project we want

to make the white agents able to win or draw the relative positions by making them play against a random opponent representing the black pieces. This is done to make them learn to play the endgames in the best way by getting results as closest as possible to the one presented in the database. According to the database the white agents will have to deal with a total of 28056 positions from which we are sure to use all the tied ones and only one part of the winning ones since part of these positions need more then 20 moves to get to the theoretical win.
As already mentioned the focus will be put on the white pieces which will play against an opponent playing randomly representing the black ones in order to be sure they're playing against something following the rules. We hope that after making them play n games, our white pieces will be able to win the positions on the board by using our reinforcement learning strategy. At the end we would like to compare if the strategy developed by our agents is similar to the one presented into the database. Our whole implementation will be made in python since there are quite different chess libraries that may turn out to be useful and we would like to respect the following timetable in which all the steps necessary to finish the project are mentioned.

- 20 21 22 November: Review of the relative literature.
- 23-30 November: Representing the different endgames on the board and implementing the movements of the pieces + the relative rules of chess.
- 1-6 December: Connect the black King to a working chess opponent and test if the white pieces are able to play against it.
- 7-17 December: Here we have 10 days to implement the learning algorithm and define properly the functions the pieces will use to play. We hope to have a possible demo running before the Christmas break.
- 18 December-4 January: Time to get back in our hometown for all of us.
- 4-13 January: Improve the learning process as much as possible and hope to get some nice results for the final presentation.