

DFA Implementation of a Chess Practice Program

By Alex Smith

1. Abstract

The Chess Openings Practice program allows users to test their knowledge of common chess openings and attempts to be a learning and practice tool for users looking to advance their chess strategy. The program uses a DFA to validate if a sequence of moves will result in a set of common opening chess moves. Using the states of the DFA, the program will also offer advice on how to properly complete openings if the user enters an incorrect sequence of moves. This program is designed to as an easy to use practice tool that benefits from the DFA's ability to read input and transfer to a different state depending on its current state.

2. Introduction

My project is a practice tool for those who want to advance their chess strategy. This tool allows the user to practice their knowledge of 6 common chess openings (three white openings and three black openings) by inputting the moves necessary to complete the opening. A DFA is used to verify if the sequence of moves given by the user is correct for the opening chosen, with each move acting as a transition between states. The user will be notified if they have correctly completed the selected opening or if they have not correctly given the selected opening. In addition, if the user gives an incorrect sequence of moves, the program will record the last move that was on track to complete the selected opening and use this to tell the user what move it was expecting next, helping the user learn how to complete the opening. This project idea came from my desire to get better at chess and learn more openings to have a greater variety in strategies I can implement while playing.

This paper will cover how the program operates given user input and how the DFAs are designed. Then there will be a section about the requirements to properly run the program on a system. Following the requirements section, there will be a brief survey of other chess practice tools and DFA implementations. There will be a detailed explanation of how to use the program, including all necessary commands and the correct opening solutions for testing purposes. Finally, the paper will conclude with a summary of the project and the implications of this project.

3. Detailed System Description

This program starts by presenting the user with a main menu and informing the user of the possible commands to use the program. If the user inputs an invalid command, the user will be notified and will be informed again that a command can be used to see all other valid commands. When the user starts 'practice' mode, they will be directed to a screen that requires the user to select if they wish to practice white openings or black openings. This is used to select which DFA will be used to complete the state transitions when the user inputs the sequence of moves to complete an opening. The program also uses states to keep track of each decision made by the user in order to display the proper instructions on the screen. After the user has selected a side, they are then prompted to choose one of the three given openings for each side. The white openings available are the Ruy Lopez, the King's Gambit, or the Italian Game opening while the black openings available are the Slav Defense, the King's Indian, or the Bogo Indian opening. Once the user has selected the opening they wish to practice, they are prompted to input a sequence of moves to complete the openings, with specific instructions for the format of a valid move. The program then takes the given input

and separates each move using spaces in the input until it reaches the 'end' given by the user. Each move is then put through the transition function of the DFA which will be further elaborated on in the detailed DFA section later in this paper. After every move has been validated by the DFA, the DFA will return the current state to the program which is then checked against the expected accepting state for that opening. If the opening is accepted, the user is informed and is returned to side selection. If the opening is not accepted, the user is informed and is given their last move on track to complete the opening with the expected next move for future reference. The user is then returned to side selection and this cycle continues until the user quits the program.

For this program, there are two DFAs to validate input, one for white openings and one for black openings. The white DFA is shown below in figure 1 and has a total of 10 states.

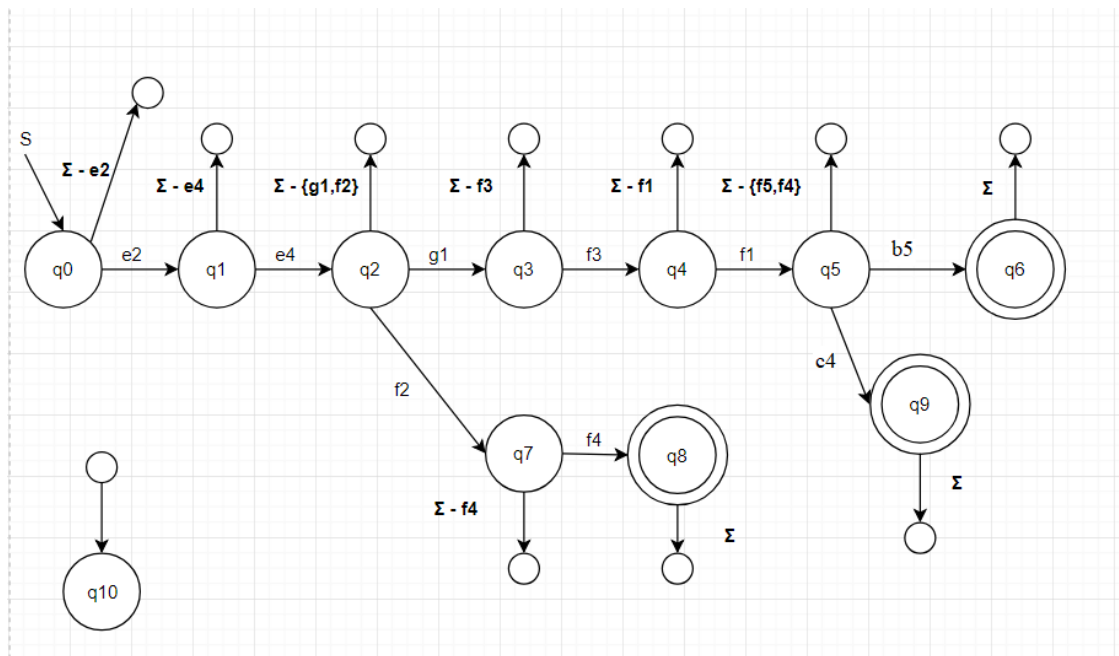


Figure 1: DFA for the white openings

The starting state is q0 and every transition in this DFA is a part of a given move by the user (either a starting position of a piece or an ending position of a piece). If the given input does not

advance the board state toward any valid opening, it is an error and transitions to the error state q10. The DFA stops validating input once the error state is reached due to the inability to leave the error state once at it. This DFA uses a 2d array to serve as a transition table for the transition function and it is possible to reach the error state from an accepting state if the user attempts to move after completing an opening successfully. The three accepting states (q6, q8, and q9) represent the correct board state for each of the respective openings (Ruy Lopez, King's Gambit, and the Italian Game).

The second DFA is the DFA to validate input for black openings and is shown below in figure 2. This DFA has a total of 13 states, starting at state q0, accepting states q4, q8, and q12,

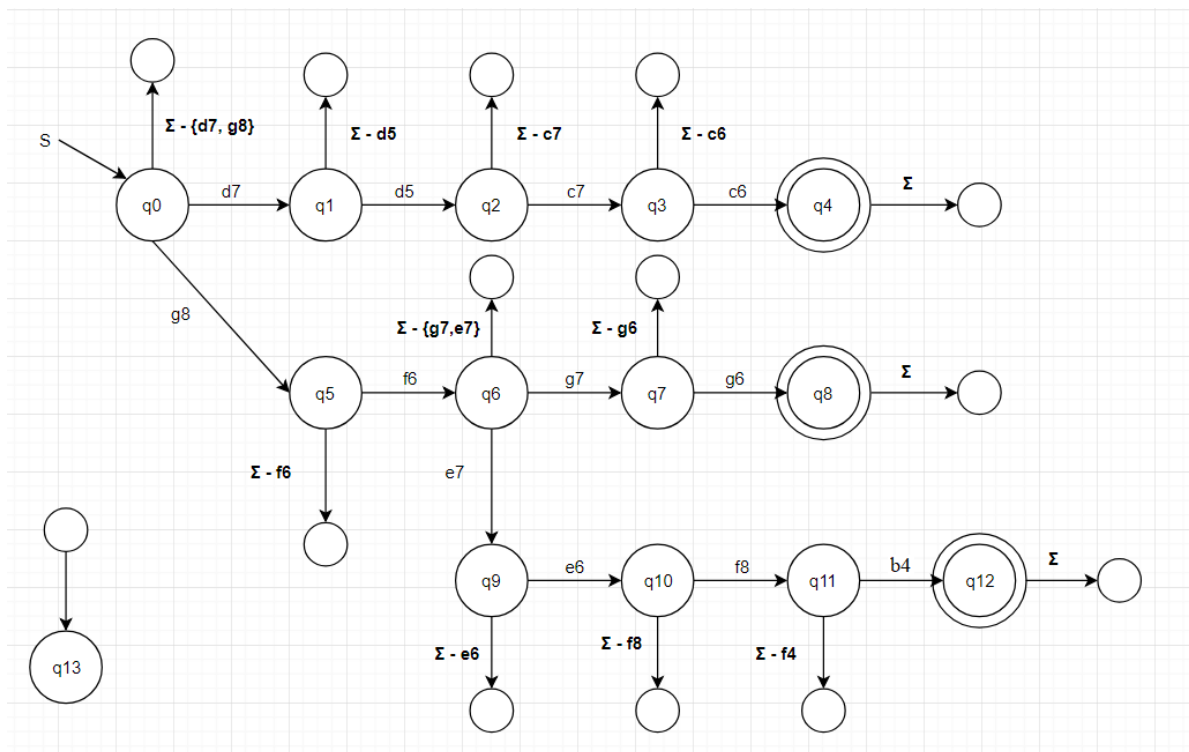


Figure 2: DFA for the black openings

and the error state q13. This DFA operates almost identically to the white openings DFA but the transitions between states are a different set of moves to lead to different openings (q4 is the

accepting state for the Slav Defense, q8 is the accepting state for the King's Indian, and q12 is the accepting state for the Bogo Indian). This DFA also uses a 2d array, with the moves serving as the columns and the current state serving as the rows, to be used as a transition table for the transition function. PDFs of both DFAs are available on GitHub and transition tables for each DFA are also available on GitHub.

4. Requirements

This program does not have many requirements, only that the user has a working Java environment on their system to run this program. The program does not require a large amount of memory and operates quickly on any average system.

5. Literature Survey

There are a variety of chess teaching tools available through various forms of media. Similar chess opening teaching tools include physical devices that use circuits to store potential moves. Instead of having the user input moves through a keyboard, a pressure sensitive board is used to record moves from the user (Hegener). This has the advantage of displaying a physical board in front of the user that may be easier to visualize but has the disadvantage of requiring the physical device instead of running a program on a computer. Websites such as Chess.com offer tools that allow the user to see a visual representation of an opening on the board and show variations the user can take after the opening to further their strategy. This website also displays win-rates of professional chess matches using the associated opening (Chess.com). Although this website displays a large amount of

information about the given openings, it does not allow the user to practice moving the pieces on the visual display, making it more of a learning tool than a practicing tool. Other chess computer programs are usually computer opponents rather than opening practice tools and fulfill a different purpose than the program presented in this project.

6. User Manual

The system requires user input and the valid commands are explained within the program. The user can type 'help' to view a list of valid commands at the start of the program. When presented with a decision, such as picking either white or black or picking an opening, instructions are on the screen of what to type to proceed to the intended opening (for example, the user will type 'w' to proceed to white openings). It is important to input commands exactly as shown on the screen to avoid any errors. When inputting a sequence of moves to complete an opening, it is important to use conventional chess board positions with A through H representing columns and 1 through 8 representing rows on a chess board. It is also essential to put a single space between each board position and to end each sequence with the word 'end' to avoid any errors.

7. Conclusion

This program is a DFA based chess opening learning tool that can help players learn how to complete common openings. Other pieces of chess software present a computer opponent that is ideal for practicing when a user has a good foundation of chess fundamentals, but this program seeks to help users establish those fundamentals by giving information as well as practicing opening sequences. The implemented DFA allows for a simple way to progress through a sequence and validate if a sequence of moves will lead to a completed opening.

Bibliography

1. Hegener, Manfred. "Chess teaching computer." U.S. Patent No. 5,098,106. 24 Mar. 1992.
2. "Chess Openings." Chess.com, www.chess.com/.