Herbert Glaser
3/31/19
Final Writeup

I: Abstract

Punch-Out! is a boxing game where you must do things such as punch and dodge to defeat your opponent and become the champion. In Punch-Out! the final boss Mike Tyson has been known as one of the hardest boss fights in all of video gaming history. To simulate this fight, I have decided to allow the person to choose from 4 different commands being jab, cross-punch, dodge, and block. These are symbolized as J,C,D, and B. To simulate the original Punch-Out! final boss I gave the player guidelines to defeat him listed down below. This fight is very difficult so to defeat him you must follow the rules. If any of these guidelines are broken, then the user will get attacked and immediately lose the match. In the original Punch-Out! if Mike Tyson hits you even once within the first minute and a half of the fight you get knocked out. I made this mock-up of the game similar to the man, goat, wolf, cabbage puzzle and that you must make decisions based on the situation. In total there are 4 different paths you can take. Three of the paths are longer and one of them is a minor split as shown in the diagram. In order to win the game, you must hit Mike Tyson five times. If you make any incorrect move, you instantly get knocked out. Depending on your playstyle you can manipulate Mike into doing things that will lead you to hit him how you want to such as using crosses instead of jabs more often makes Mike use specific attacks making it so you know to dodge rather than trying to remember to block or not. The fight is complicated but that's how it was in the original game as well. You really must think about your approach to defeating the most terrifying person in the ring.

II: Introduction

The focus of this DFA project is to simulate how AI manipulation works in video games. Many talented Punch-Out! players use AI manipulation to win this difficult fight. AI manipulation is when the player does certain actions that cause the opponents to do certain actions as a response. For example, like in the DFA if you punch at a certain time or use a certain attack then Mike will counter appropriately. If you were to use certain AI manipulation guidelines than it will make defeating Mike Tyson much easier. In the normal Punch-Out! a large aspect of the game is timing, but I decided on focusing on the second most important aspect in this game.

III: Detailed System Description

As stated previously, this DFA is very similar to the man, wolf, cabbage, goat puzzle game. It makes the user consider all the options that they have and make decisions based on the scenario. The way commands are input in that DFA are very similar to the Punch-Out! DFA being a string of commands such as cmwgcmw etc. When reviewing the man, wolf, cabbage, and goat game I planned on doing something similar because the concept was interesting. So I decided to create a similar DFA for a game that is interesting. The Punch-Out! DFA is different from the man, wolf, cabbage, and goat DFA because it uses many more guidelines and rules. It also allows the user to create different paths for themselves based on how the user likes to play. This allows the user to also simulate AI manipulation so they may use these types of guidelines in the actual Punch-Out! game.

IV: Literature Survey

This DFA uses a very similar format to the man, wolf, cabbage, and goat puzzle game. To use this DFA you simply type out the commands that you want to input in a string. The commands are J for jab, B for block, C for cross, and D for dodge. They can be input in a string such as jbcdjbcdjbc. Once the accepted state is reached the DFA will say that Mike Tyson has been defeated. If a string of commands is input that doesn't reach the accepted state, the DFA will say that you have been knocked out and you will have to try again. If a command such as A or P are input, it will immediately go to the error state and you will lose the game. There are a total of 22 states in this DFA. 21 of the states are used for mapping of your progress and the 22nd state is used as an error state. At states q1, q3, q10, q14, and q17 they branch into different paths that allow the player to decide on different actions with different results of those actions.

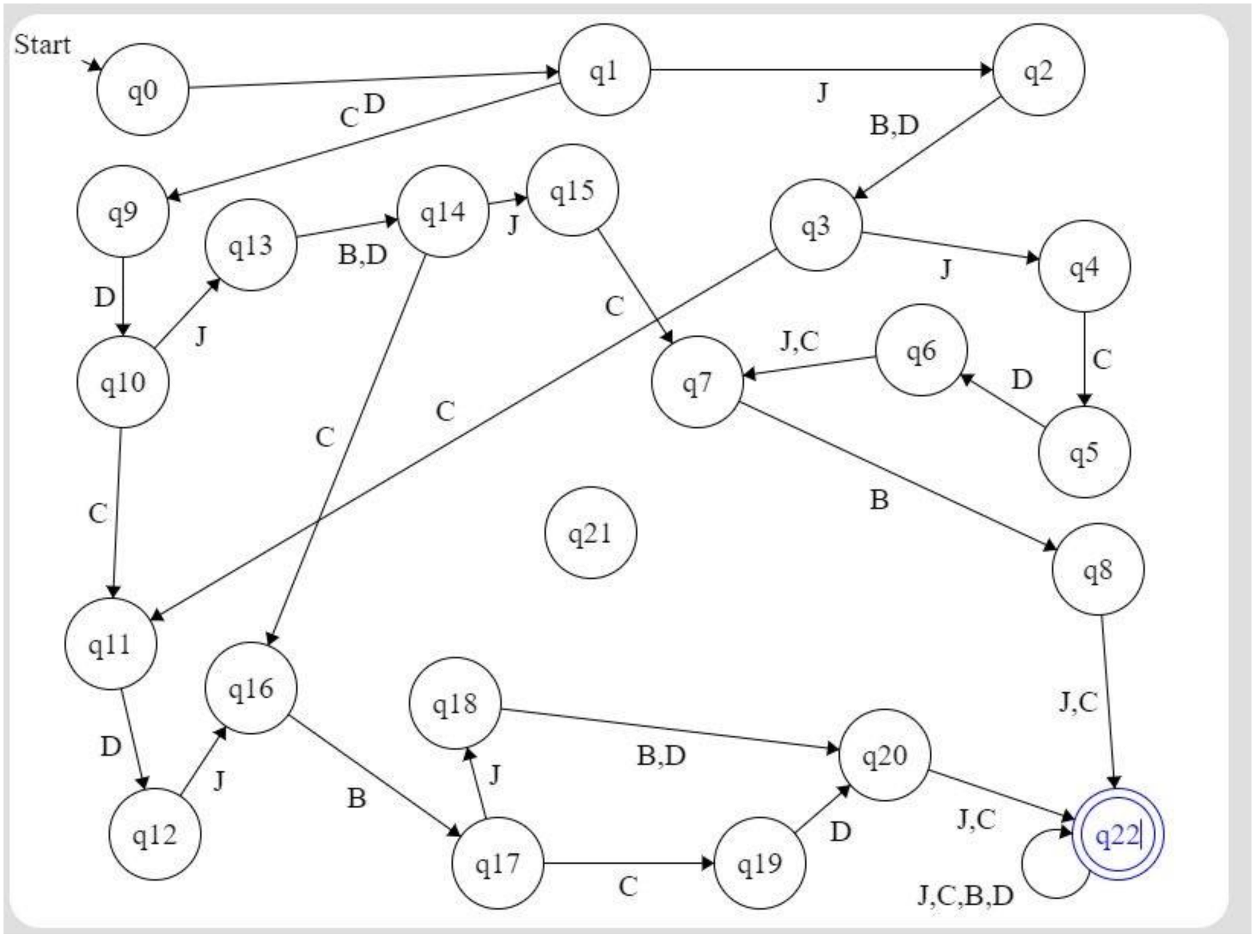
V: Conclusion

In conclusion, the Punch-Out! DFA simulates the AI manipulation portion of the final boss in the game Punch-Out! Using AI manipulation is a big part of defeating the final boss because of the sheer difficult that he showcases. If players learn similar guidelines to his attack patterns and how he acts when you input certain actions, then defeating him will be much easier. The main purpose of creating this DFA was to create a game that I enjoy and teach players a lesson on how to defeat on of the most difficult bosses in video game history.

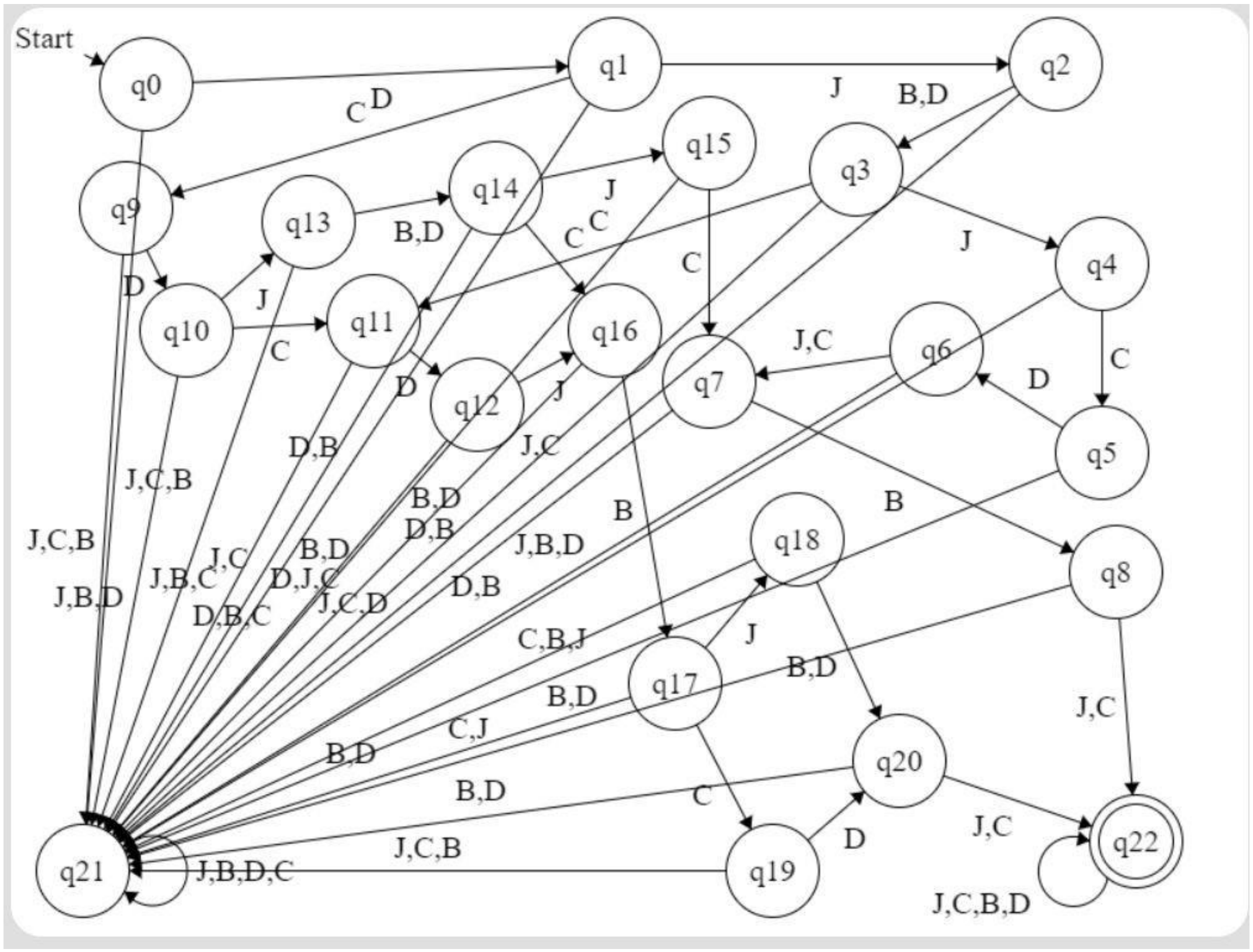
Rules

-Mike gets the first attack and always starts with an uppercut
-Mike will cross you every fourth attack
-After an uppercut Mike will go for two jabs unless it’s the fourth attack
-If you cross mike while he isn’t knocked off balance, he will get mad and Uppercut you even (not if it's the fourth attack)
-Every time you Jab Mike twice, he will be knocked off balance allowing you to cross him
-Never Block an uppercut
-Do not jab while mike is knocked off balance
-Don’t block two jabs in a row
-If Mike isn’t knocked off balance you only have enough time for 1 attack
-If you try to dodge or block during an attack Mike will catch you off guard and knock you out ----
-You can't dodge a cross

This is the DFA diagram with the error state invisible for a clearer view of the diagram



This is the DFA diagram with the error state



This is the state Transition table for the final writeup

State Transition Table		B	D	J	C
	q0	q21	q1	q21	q21
	q1	q21	q21	q2	q9
	q2	q3	q3	q21	q21
	q3	q21	q21	q4	q11
	q4	q21	q21	q21	q5
	q5	q21	q6	q21	q21
	q6	q21	q21	q7	q7
	q7	q8	q21	q21	q21
	q8	q21	q21	q22	q22
	q9	q21	q10	q21	q21
	q10	q21	q21	q13	q11
	q11	q21	q12	q21	q21
	q12	q21	q21	q16	q21
	q13	q14	q14	q21	q21
	q14	q21	q21	q15	q16
	q15	q21	q21	q21	q7
	q16	q17	q21	q21	q21
	q17	q21	q21	q18	q19
	q18	q20	q20	q21	q21
	q19	q21	q20	q21	q21
	q20	q21	q21	q22	q22
	q21	q21	q21	q21	q21
	q22	q22	q22	q22	q22