Milestone Report

John DeFalco

March 2019

1 Abstract

This milestone report covers the progress of what stage my semester project is currently at. The introduction covers the overview of the project, mentioning what the project topic is, and what differences I have noticed between what I expected and what the actual outcome has been. The detailed system description includes a diagram of the DFA, along with the formal definition of the automaton. This section also covers what the system does, and what it is used for. The requirements section covers what materials are necessary to practically apply this system into the real world. The literature survey goes over similar systems out there, and how this system is both similar and different to those other systems. The user manual covers how one would interact with this system, and how they may solve any errors they could possibly encounter when using the system. Finally, the conclusion goes over the material required, and I mention how I imagine the rest of the project going, with any changes or tweaks I may put in in the future.

2 Introduction

The project I have been working on is a DFA-implementation of a standard parking meter.

The motivation behind this idea was that a parking meter is a practical system that I've encountered before, and I figure it would help me to more easily understand how finite automata work. The parking meter is rather simple: enter coins into the machine until a minimum amount is reached, and then stop entering coins. The monetary value entered reflects a time period allotted, for example 25 cents can be 30 minutes of parking. Using this idea, I built what I believe to be a comprehensive and concise deterministic finite automaton to represent this system.

3 Detailed System Description

I have formally defined the DFA to be implemented as follows:

- $Q = \{0, 0.25, 0.50, 0.75, 1.00, 1.25, 1.50, 1.75, 2.00, Payment, Error\}$
- $\Sigma = 0.25, 0.50, Enter$
- δ : *See Table 3.1*
- $q_0 = 0$
- $F = Payment$

State	Input (0.25)	Input (0.50)	Input (*Enter*)
0	0.25	0.50	*Error*
0.25	0.50	0.75	*Error*
0.50	0.75	1.00	*Payment*
0.75	1.00	1.25	*Payment*
1.00	1.25	1.50	*Payment*
1.25	1.50	1.75	*Payment*
1.50	1.75	2.00	*Payment*
1.75	2.00	2.00	*Payment*
2.00	2.00	2.00	*Payment*
Payment	*Payment*	*Payment*	*Payment*
Error	*Error*	*Error*	*Error*

Table 3.1: Transition Function

When I started working more in depth on the project and reflected on parking meters I've encountered throughout my life, I realized my alphabet was not completely accurate. I originally included 1.00, but changed the set to only include 0.25, 0.50 and *Enter*, as I've only encounter parking meters that accept coins, not dollar bills. I've also included *Enter* in the alphabet, which is not typically found on a parking meter, but it helps better visualize the system in this scenario.

In theory, a user interacting with this system would enter either quarters or half-dollar coins into the meter until they have entered at least 50 cents or at most 2 dollars, and then would select *Enter* to start the meter. A model of the DFA can be seen in Figure 3.1.

4 Requirements

The requirements the user would have to have to use this machine is just enough money, in quarters or half-dollars, that is at the last 50 or more cents. One could argue the user would also need to have a vehicle to park in the space that the

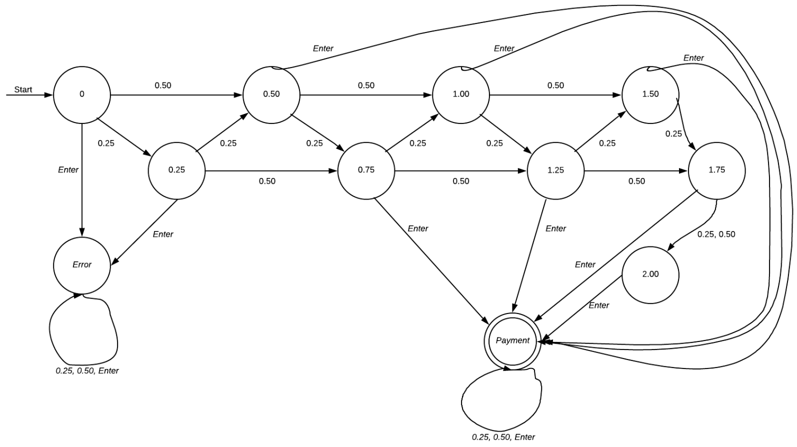

Figure 3.1: Drawing of DFA defined by the 5-tuple

meter covers, however if a user so chooses they could hypothetically use the meter without the vehicle.

5 Literature Survey

From my personal experience, this system is extremely similar to any other parking meter. There are some exceptions, meters that can accept credit/debit cards, and a number of meters that can accept more than 2 dollars, however my intention with this project was not to create a completely unique system.

The concept of deterministic finite automata was foreign to me and hard to understand, so using something familiar I thought I could better understand the lessons being taught in class. Working on the project this far, I can say my initial goal has been either met or close to being met. Working on this system has assisted me in understand the DFA on a deeper level than before, which has also in turn assisted me in better understanding $NFAs$ and how they operate as well.

6 User Manual

The user would park their vehicle in the parking spot, and approach the meter covering the spot.

On the meter, they would see that 25 cents is equal to 30 minutes of parking space. The minimum is an hour of parking, or 50 cents, and the machine will accept either quarters or half-dollar coins. The maximum amount allowed is 2 dollar, or 4 hours of parking time. If the user attempts to enter more than 2 dollars, the machine will not stop the user from entering the money, however the meter **will not go further past 2 dollars.** If the user tries inserts 25 cents into the machine and attempts to select *Enter*, the machine will not accept the payment, and prompt the user to enter more money until they pass the 50 cent threshold.

The user will enter their coins until their desired amount of time, and then select *Enter* to finalize the amount entered. From their, the meter will run until the allotted time is depleted, where the user will either have to enter more money into the meter, or move their vehicle.

7 Conclusion

In conclusion, the background work of this project is complete, and all that is left is for the actual coding implementation. The parking meter has turned out to be an excellent example for me to practice with, as I've mentioned before, because it is helping me understand the practical use of finite automata within our everyday lives. The significance of this project lies within the educational value that it has provided; it has been a perfect supplemental tool to further understand the lessons from class.