Katie Bartolotta

CMPT 440L

Project Milestone

Credit Card Validation

3/31/19

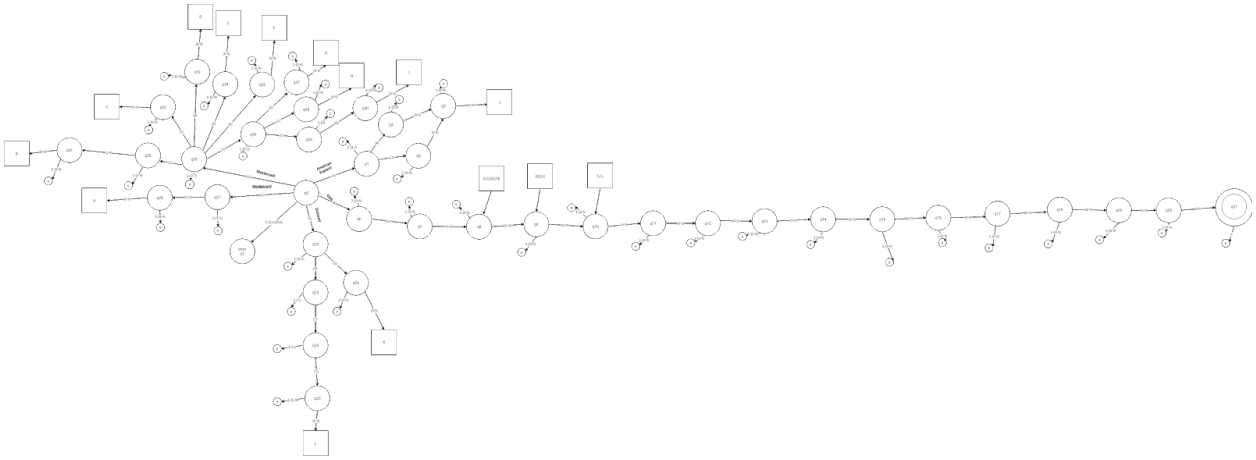
Abstract

This project will explore the realm of credit card processing by recreating the validation process for a credit card number. Through a java program, a user will be able to enter a credit card number and discover whether it is valid as well as what type of card it is. On the surface, this process may seem quite simple, but through the DFA diagram provided it is clear that this is in fact quite complicated. The diagram follows the pattern for each valid credit card for each of the four major credit card companies. Each is unique and will need to be handled accordingly in the java program. Ultimately, through the use of the DFA diagram and the java program, I will be able to represent the DFA created through a viable user interface.

Introduction

It has grown more and more uncommon for people to carry around paper money with them, replacing this with a plastic card. Credit cards have become an easy and efficient way to pay for items on a daily basis. There is no need to get change back or do mathematical calculations of any kind. Instead, one must only hand over their credit card and the money is taken out of their account automatically. Each credit card is unique to the holder, having their own fifteen or sixteen digit number. One individual may have multiple credit cards, but they will all have different numbers. There are four major credit cards that are used which include: Mastercard, Visa, Discover, and American Express. Each company uses their own way of generating numbers. Thus, it is necessary to figure out which type of credit card is being processed before evaluating the digits to see if they represent a valid card, a process that is more intricate than it seems.

DFA Diagram



DFA Transition Table

	0	1	2	3	4	5	6	7	8	9
q0	q2	q2	q29	q1	q6	q27	q22	q2	q2	q2
q1	q2	q2	q2	q2	q3	q2	q2	q4	q2	q2
q2	q2	q2	q2	q2	q2	q2	q2	q2	q2	q2
q3	q5	q5	q5	q5	q5	q5	q5	q5	q5	q5
q4	q5	q5	q5	q5	q5	q5	q5	q5	q5	q5
q5	q10	q10	q10	q10	q10	q10	q10	q10	q10	q10
q6	q7	q7	q7	q7	q7	q7	q7	q7	q7	q7
q7	q8	q8	q8	q8	q8	q8	q8	q8	q8	q8
q8	q9	q9	q9	q9	q9	q9	q9	q9	q9	q9
q9	q10	q10	q10	q10	q10	q10	q10	q10	q10	q10
q10	q11	q11	q11	q11	q11	q11	q11	q11	q11	q11
q11	q12	q12	q12	q12	q12	q12	q12	q12	q12	q12
q12	q13	q13	q13	q13	q13	q13	q13	q13	q13	q13
q13	q14	q14	q14	q14	q14	q14	q14	q14	q14	q14
q14	q15	q15	q15	q15	q15	q15	q15	q15	q15	q15
q15	q16	q16	q16	q16	q16	q16	q16	q16	q16	q16
q16	q17	q17	q17	q17	q17	q17	q17	q17	q17	q17
q17	q18	q18	q18	q18	q18	q18	q18	q18	q18	q18
q18	q19	q19	q19	q19	q19	q19	q19	q19	q19	q19
q19	q20	q20	q20	q20	q20	q20	q20	q20	q20	q20
q20	q21	q21	q21	q21	q21	q21	q21	q21	q21	q21
q21	q2	q2	q2	q2	q2	q2	q2	q2	q2	q2
q22	q23	q2	q2	q2	q2	q26	q2	q2	q2	q2
q23	q2	q24	q2	q2	q2	q2	q2	q2	q2	q2
q24	q2	q25	q2	q2	q2	q2	q2	q2	q2	q2
q25	q10	q10	q10	q10	q10	q10	q10	q10	q10	q10
q26	q8	q8	q8	q8	q8	q8	q8	q8	q8	q8
q27	q2	q28	q28	q28	q28	q28	q2	q2	q2	q2
q28	q8	q8	q8	q8	q8	q8	q8	q8	q8	q8
q29	q2	q2	q30	q32	q33	q34	q35	q36	q2	q2
q30	q2	q2	q31	q31	q31	q31	q31	q31	q31	q31
q31	q9	q9	q9	q9	q9	q9	q9	q9	q9	q9
q32	q8	q8	q8	q8	q8	q8	q8	q8	q8	q8
q33	q8	q8	q8	q8	q8	q8	q8	q8	q8	q8
q34	q8	q8	q8	q8	q8	q8	q8	q8	q8	q8

q35	q8	q8	q8	q8	q8	q8	q8	q8	q8	q8
q36	q37	q38	q39	q2	q2	q2	q2	q2	q2	q2
q37	q9	q9	q9	q9	q9	q9	q9	q9	q9	q9
q38	q9	q9	q9	q9	q9	q9	q9	q9	q9	q9
q39	q40	q2	q2	q2	q2	q2	q2	q2	q2	q2
q40	q10	q10	q10	q10	q10	q10	q10	q10	q10	q10

DFA Transition Descriptions

q0	The start state that takes valid input of 2-6 otherwise, enters error state
q1	If credit card begins with a 3, moves to American Express validation
q2	The start state that takes valid input of 2-6 otherwise, enters error state
q3	If credit card begins with a 3, moves to American Express validation
q4	The start state that takes valid input of 2-6 otherwise, enters error state
q5	If credit card begins with a 3, moves to American Express validation
q6	The start state that takes valid input of 2-6 otherwise, enters error state
q7	If credit card begins with a 3, moves to American Express validation
q8	The start state that takes valid input of 2-6 otherwise, enters error state
q9	If credit card begins with a 3, moves to American Express validation
q10	The start state that takes valid input of 2-6 otherwise, enters error state
q11	If credit card begins with a 3, moves to American Express validation
q12	The start state that takes valid input of 2-6 otherwise, enters error state
q13	If credit card begins with a 3, moves to American Express validation
q14	The start state that takes valid input of 2-6 otherwise, enters error state
q15	If credit card begins with a 3, moves to American Express validation
q16	The start state that takes valid input of 2-6 otherwise, enters error state
q17	If credit card begins with a 3, moves to American Express validation
q18	The start state that takes valid input of 2-6 otherwise, enters error state
q19	If credit card begins with a 3, moves to American Express validation
q20	The start state that takes valid input of 2-6 otherwise, enters error state
q21	If credit card begins with a 3, moves to American Express validation

q22	The start state that takes valid input of 2-6 otherwise, enters error state
q23	If credit card begins with a 3, moves to American Express validation
q24	The start state that takes valid input of 2-6 otherwise, enters error state
q25	If credit card begins with a 3, moves to American Express validation
q26	The start state that takes valid input of 2-6 otherwise, enters error state
q27	If credit card begins with a 3, moves to American Express validation
q28	The start state that takes valid input of 2-6 otherwise, enters error state
q29	If credit card begins with a 3, moves to American Express validation
q30	The start state that takes valid input of 2-6 otherwise, enters error state
q31	If credit card begins with a 3, moves to American Express validation
q32	The start state that takes valid input of 2-6 otherwise, enters error state
q33	If credit card begins with a 3, moves to American Express validation
q34	The start state that takes valid input of 2-6 otherwise, enters error state
q35	If credit card begins with a 3, moves to American Express validation
q36	The start state that takes valid input of 2-6 otherwise, enters error state
q37	If credit card begins with a 3, moves to American Express validation
q38	The start state that takes valid input of 2-6 otherwise, enters error state
q39	If credit card begins with a 3, moves to American Express validation
q40	The start state that takes valid input of 2-6 otherwise, enters error state

Detailed System Description

The DFA will allow for the processing of any of the four major credit card types: Mastercard, Visa, Discover, and American Express. It begins at a start state in which the user is able to enter their first number. If the number is between two and six, then that will signal to the program that this is the beginning of a valid credit card number, otherwise it will go into an error state and the user must start over from the beginning. At every state there is a possibility for an error state to occur where the user must begin again from the start of their card number. Each of the four credit card types has a different sequence. Mastercards can begin with 51 through 55, or with 2221-2720. This card has the most possibilities. Visa cards are simpler as they only have to begin with a 4. Discover cards must begin with either 6011 or 65. These three types of cards each have digits after the start sequence that are 0 through 9 until it reaches sixteen total digits. American Express, however, is a little different, having only fifteen digits. These cards must start with 34 or 37. The diagram is somewhat messy due to the number of possible start sequences for these cards as well as the possibility for user error.

In terms of the java program that will represent this DFA, it will need to validate each number individually and makes a choice at each stage. This makes for a lengthy code file, but it is necessary to make sure that neither user error nor intended sabotage can break the system. While piecing through each part of the string, the program will jump from state to state of the DFA. If an error state is reached, the user will need to re-enter their card number once again.