**CSCE 4323 / CSCE 5173 – Formal Languages and Computability**

**Assignment 2 – Due 02/07/2024**

**Instructions:**

The focus of this assignment is the creation of NFAs to recognize specific regular languages. Many of the problems below will be similar to examples discussed in class and in chapter 2 of Sipser’s text book. In order to complete this assignment, you must use JFLAP to create and test your DFAs. This software is available at no cost from http://www.jflap.org. Please see their online documentation to learn how to use this application.

For each of the questions below, create the specified NFA or DFA using JFLAP and perform a sequence of tests to illustrate that the FA recognizes the specified language (accepts strings in the language, rejects strings not in language). Save the FA as a “jff” file and save the testing results as a “jpg” file with names corresponding to the problem number. For question 3, the file names should be “3.jff” and “3.jpg”. Save these files to be included in your homework document and uploaded in Blackboard.

**Questions:**

1. [4 points] Create an NFA with  = {a, b} to recognize all strings in the language L = {w | w contains “bb” or “aaa”}

A screenshot of a computer

Description automatically generated

1. [4 points] Create an NFA with  = {a, b} to recognize all strings in the language L = {w | w is either the string “abba” or a string with exactly 4 a’s and any number of b’s}

A screenshot of a computer

Description automatically generated

1. [4 points] Create an NFA with  = {a, b} to recognize all strings in the language L = {w | w starts with “aa” and ends with “aa”}

A screenshot of a computer

Description automatically generated

1. [4 points] Create an NFA with  = {a, b} to recognize all strings in the language L = {w | w is a sequence of a’s with even length followed by a sequence of b’s with even length}. For example, the strings “aaaabb” and “bbbbbb” are in the language.

A screenshot of a computer

Description automatically generated

1. [4 points] Create an NFA with  = {0, 1} to recognize all strings in the L = L1\* where L1 = {w | w = is either “0” or “111”}

A screenshot of a computer

Description automatically generated

1. [4 points] Create an NFA with  = {0,1} to recognize all strings in the language L = L1 U L2 where L1 = {w | w begins with a 00 and ends with 00} and L2 = {w | w contains the substring “0101”}

A screenshot of a computer

Description automatically generated

1. [4 points] Create an NFA with  = {0,1} to recognize all strings in the language L = L1 U L2 where L1 = {w | w contains three or more 0’s} and L2 = {w | w does not contain the substring “1001”}

A screenshot of a computer

Description automatically generated

1. [4 points] Create an NFA with  = {0,1} to recognize all strings in the language L = {w | w represents a binary number with a value that is not a multiple of 3} (assume the most significant bit is on the left so 110 represents 6)

A diagram of a diagram of a diagram

Description automatically generated

1. [4 points] Use the method described on class to manually convert the following NFA into an equivalent DFA. Draw your DFA using JFLAP and include the jpg image and your DFA transition table in your homework document.

M = {Q={q0,q1,q2,q3},  = {a,b}, q0=q0, F = {q3}}

|  |  |  |  |
| --- | --- | --- | --- |
|  | a | b |  |
| q0 | {q2} | {} | {} |
| q1 | {q2} | {q3} | {q0} |
| q2 | {q0,q1} | {} | {} |
| q3 | {q3} | {q3} | {} |

A screenshot of a computer

Description automatically generated

A blackboard with white writing on it

Description automatically generated

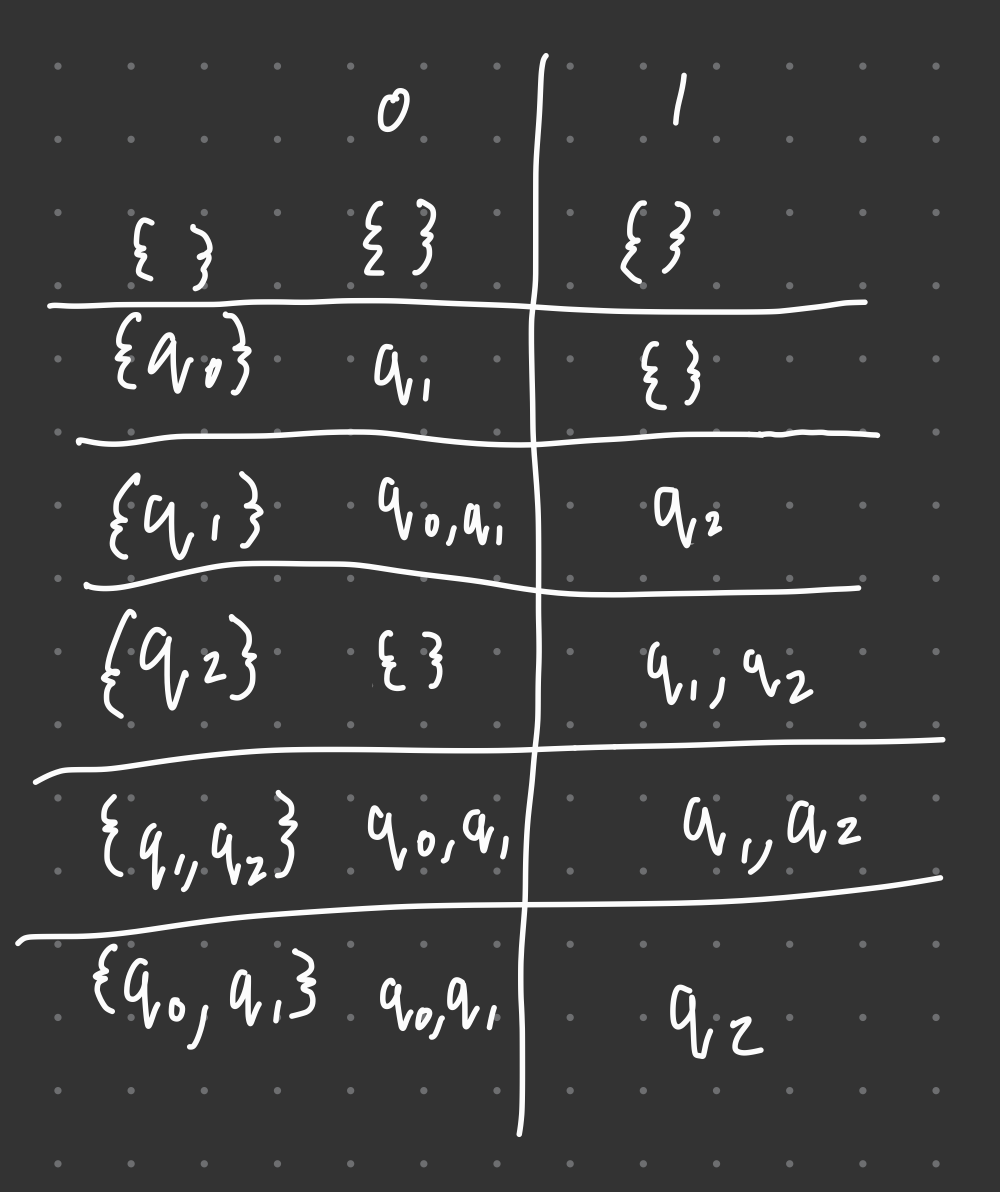
1. [4 points] Use the method described on class to manually convert the following NFA into an equivalent DFA. Draw your DFA using JFLAP and include the jpg image and your DFA transition table in your homework document.

M = {Q={q0,q1,q2},  = {0,1}, q0=q0, F = {q1}}

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0 | 1 |  |
| q0 | {q1} | {} | {q1} |
| q1 | {q0} | {q2} | {} |
| q2 | {} | {q1,q2} | {} |

A screenshot of a computer

Description automatically generated



**Challenge Question: (required for graduate students, optional for undergraduates)**

1. [4 points] Use the method described on class to manually convert the following NFA into an equivalent DFA. Draw your DFA using JFLAP and include the jpg image and your DFA transition table in your homework document.

M = {Q={q0,q1,q2,q3},  = {0,1}, q0=q0, F = {q1,q3}}

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0 | 1 |  |
| q0 | {q1} | {q3} | {} |
| q1 | {q2} | {q1,q2} | {} |
| q2 | {q1,q3} | {q0} | {q0} |
| q3 | {} | {q3} | {} |

**Assignment Submission:**

Once you have answered all of the questions above, create a document called “hw2.docx” that has a copy of the questions above and copy/paste the “jpg” images you created into this document. Create a folder called “hw2” that contains all of your “jff” files and compress this folder to create “hw2.zip”. Finally, upload “hw2.zip” and “hw2.docx” into the “hw2” link in Blackboard. In order to receive full credit for each question, the “jff” files must be named correctly, and they must load and operate correctly in JFLAP. No credit will be given for missing or incorrectly named files.

The dates on your electronic submission will be used to verify that you met the due date above. All late assignments will receive reduced credit:

* 10% off if less than 1 day late,
* 20% off if less than 2 days late,
* 30% off if less than 3 days late,
* no credit if more than 3 days late.

**7. Academic Honesty Statement:**

Students are expected to submit their own work on all assignments, unless group projects have been explicitly assigned. Students are NOT allowed to distribute solutions to each other, or copy solutions from another individual or website. Students ARE allowed to use any materials on the class website, or in the textbook, or ask the instructor and/or GTAs for assistance.

Violations of the policies above will be reported to the Provost's office and may result in a ZERO on the assignment, an F in the class, or suspension from the university, depending on the severity of the violation and any history of prior violations.