

# CPSC-402 Report

Stephen White

February 5, 2022

## Abstract

In short, the goal is to search for all occurrences of a small string within a larger string. We are trying to emulate the text searching abilities of Regular Expressions.

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Homework</b>	<b>1</b>
2.1	Week 1	1
2.1.1	Homework 1.1	1
2.1.2	Homework 1.2	3
2.2	Week 2	4
<b>3</b>	<b>Project</b>	<b>4</b>
<b>4</b>	<b>Conclusions</b>	<b>4</b>

## 1 Introduction

My name is Stephen White. I am a Computer Science major with a minor in Business Administration at Chapman University.

## 2 Homework

This section contains the solutions to the homework.

### 2.1 Week 1

#### 2.1.1 Homework 1.1

The homework discussed in this section can be seen [here](#). Part 1 of the homework is exercise 2.2.4 item B. The problem can be found on page 54 of the textbook (page 69 of the pdf).

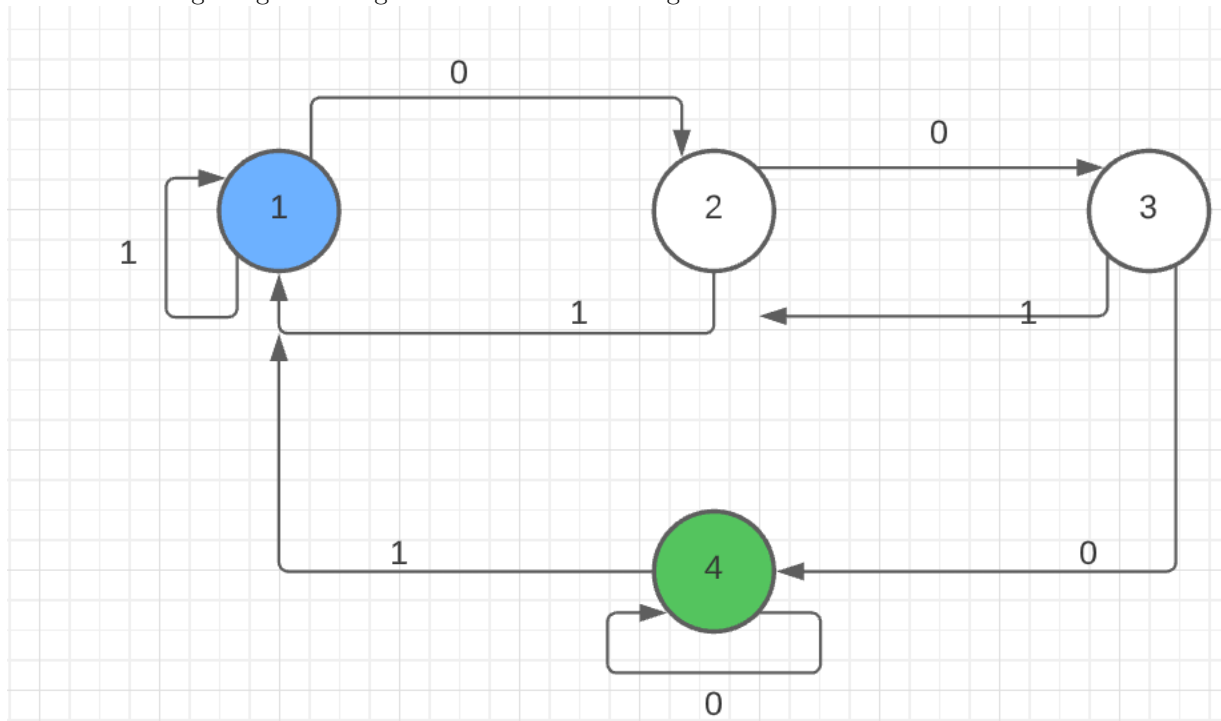
The first question is as follows: "Give DFA's accepting the following languages over the alphabet  $\{0,1\}$ : The set of all strings with three consecutive 0's (not necessarily at the end)."

My answer to this question is as follows:

Since we know that this DFA's alphabet can only consist of 0's and 1's, and we are searching for three consecutive 0's, looking like this: 000, we can formally defined the DFA as

$$\{\omega \mid \omega \text{ is of the form } x000y \text{ for some strings } x \text{ and } y \text{ consisting of } 0's \text{ and } 1's \text{ only}\}$$

The following image is a diagram of this DFA showing its states and transitions.



(I have chosen blue to signify the initial state and green to signify the success state)

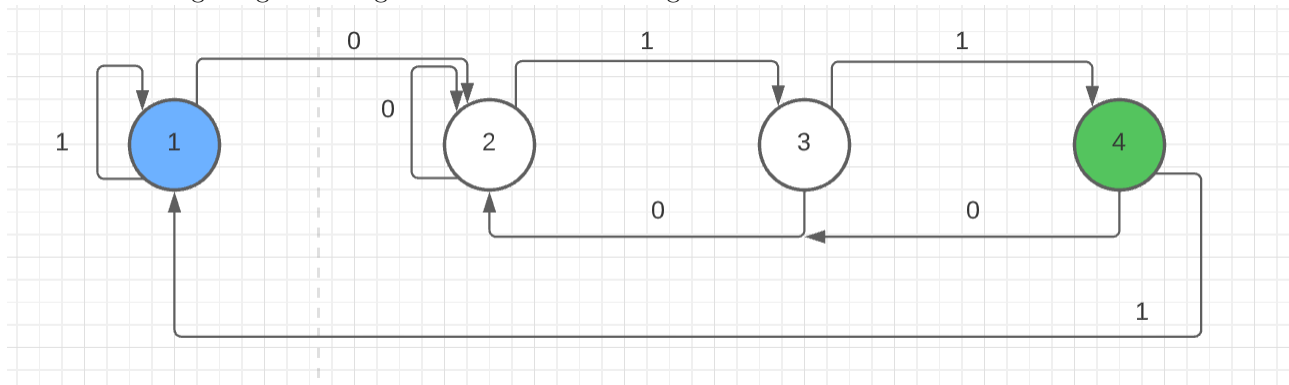
The next question asks "Give DFA's accepting the following languages over the alphabet {0,1}: The set of strings with 011 as a substring."

My answer to this question is as follows:

Since we know that this DFA's alphabet can only consist of 0's and 1's, and we are searching for the string 011, we can formally defined the DFA as

$$\{\omega \mid \omega \text{ is of the form } x011y \text{ for some strings } x \text{ and } y \text{ consisting of } 0's \text{ and } 1's \text{ only}\}$$

The following image is a diagram of this DFA showing its states and transitions.



(I have chosen blue to signify the initial state and green to signify the success state)

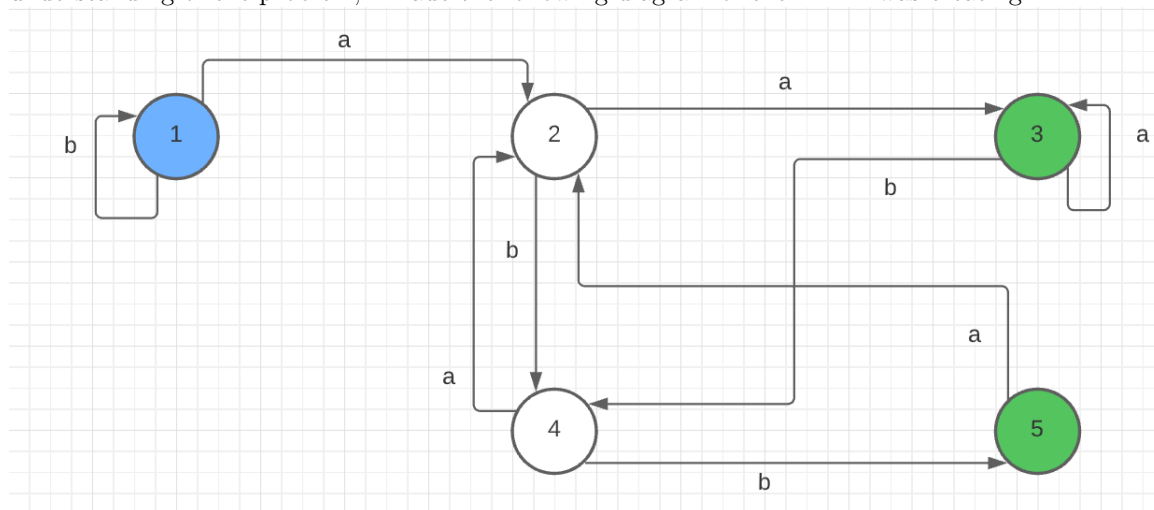
### 2.1.2 Homework 1.2

The homework discussed in this section can be seen [here](#).

Part 1 of this assignment is to answer the following question:

”Does the file contain the sequence aa or abb? For example, if the file is aaabbaabb, the output should be [2,3,5,7,9]. The challenge here is to not write a program that runs twice through the file, once to search for aa and once for abb, but to find a way of running through the file only once and to simultaneously search for both strings.”

I chose to write the program in C# because I am most familiar with it, and I believe it has a good mix between control and programming speed. That code can be found [here on my GitHub](#). To aid me in my understanding of the problem, I made the following diagram of the DFA I was creating:



Part 2 of the homework was to complete exercise 2.2.10 (the proof being optional) on page 54 of the textbook (page 70 of the pdf).

The question reads as follows:

	0	1
$\rightarrow A$	A	B
$*B$	B	A

”Informally describe the language accepted by this DFA.”

I tackled this problem by first breaking down the chart, to understand what I was being asked. The chart is similar to a Punnett Square. The left column is a list of the possible current states (an input). The top row is a list of possible character inputs. The rest of the chart is the next state (the output) determined by that row’s current state and current character. Since the only possible characters are 1 and 0, the language accepted by this DFA is a set of strings or words created by any combination of 0’s and 1’s.

## 2.2 Week 2

...

## 3 Project

The project will be to compile code from a programming language of your choice to assembly and to explain the assembly code and compilation process using the knowledge you learned in this course.

Pro tips:

- Choose a compiled (not interpreted) programming language.
- Choose a language that you find interesting anyway.
- Start early.
- Come to office hours. I have not run this part of the course before and I am really interested what you will find.

## 4 Conclusions

## References

[HMU] John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman: [Introduction to automata theory, languages, and computation](#), 3rd Edition. Pearson international edition, Addison-Wesley 2007