

CPSC-406 Report

Marc Nathaniel Domingo
Chapman University

February 23, 2023

Abstract

The purpose of this course is to understand how the Theory of Algorithms coincide with the application of functions and algorithms in Software Engineering. In particular, this course places an emphasis on the concepts of Automata, Logic, and Concurrency in regard to algorithms in our eventual application of these topics to a final project.

Contents

1	Introduction	1
1.1	Week 1	1
1.2	Week 2	1
1.3	Week 3	1
1.4	Week 4	2
2	Homework	2
2.1	Week 2	2
2.2	Week 3	3
3	Paper	5
4	Conclusions	5

1 Introduction

1.1 Week 1

During this week, the overall outline and "roadmap" of topics and project deadlines were covered during class, as well as an introduction to the concept of Definitive Finite Automata (DFA).

1.2 Week 2

Lectures covered Definitive Finite Automata (DFA) and Non-Definitive Finite Automata (NFA) in more detail, as well as how to convert NFA to DFA.

1.3 Week 3

During this week, the lectures covered the concepts of querying a database, as well as algorithms for Unification and Resolution.

1.4 Week 4

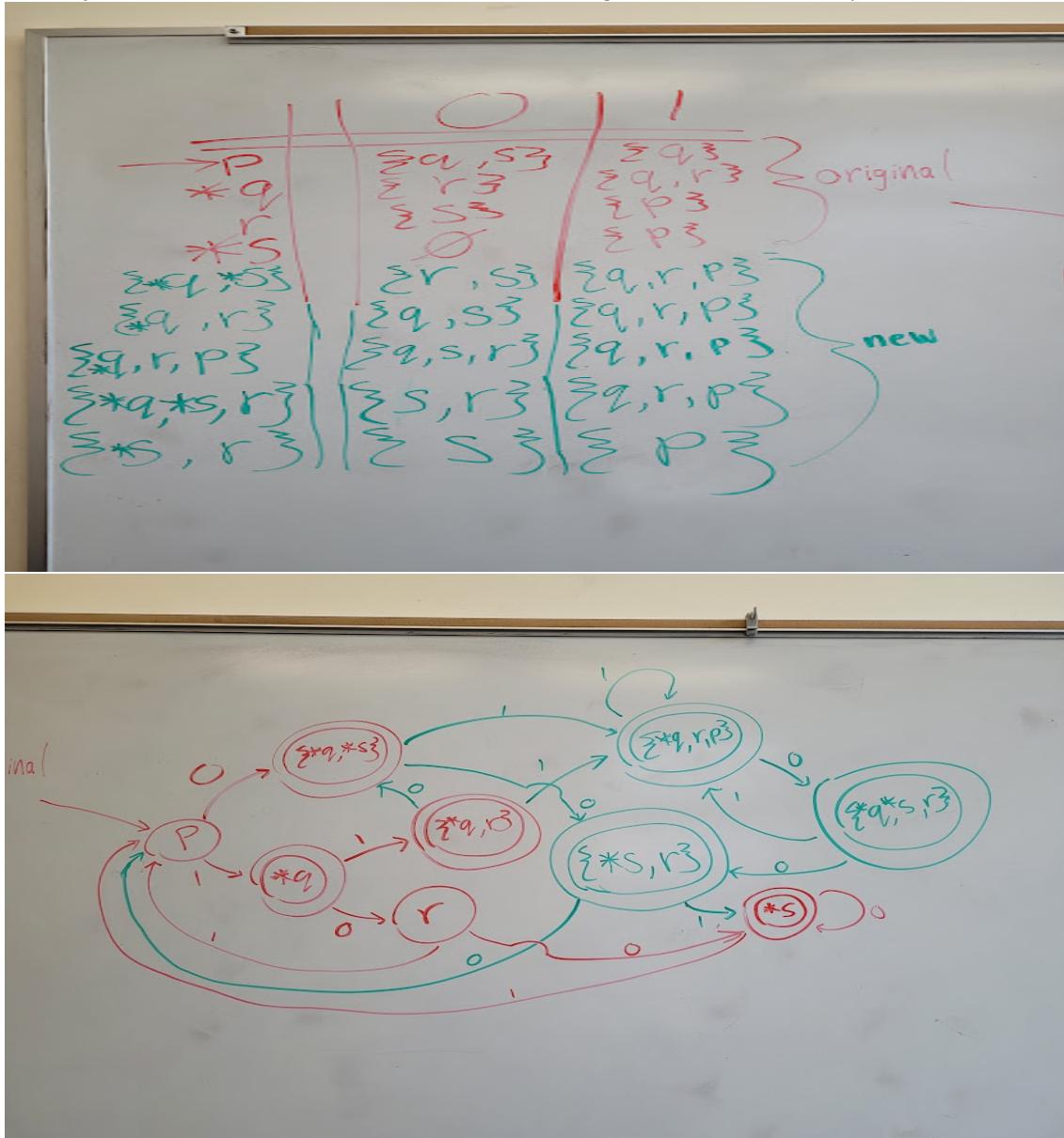
During this week, lectures covered the concept of Distributed Hash Tables (DHT) in the application of Decentralized Systems.

2 Homework

This section contains solutions to homework.

2.1 Week 2

The objective of this week's homework is to convert a given NFA to a DFA by hand.



2.2 Week 3

The objective of this week's homework is to solve Unification Algorithm problems in addition to practicing drawing an SLD-tree solution to a program given a particular input.

1. $f(x, f(x, y)) \stackrel{?}{=} f(f(y, a), f(v, b))$

1. $X \stackrel{?}{=} f(y, a)$
 $\sigma_1 = [f(y, a) / X]$
2. $f(x, y) \stackrel{?}{=} f(v, b)$
3. $X \stackrel{?}{=} v$
 $X_{\sigma_1} \stackrel{?}{=} v \rightarrow f(y, a) \stackrel{?}{=} v$
 $\sigma_3 = [f(y, a) / v]$
4. $y \stackrel{?}{=} b$
 $\sigma_4 = [b / y]$

 $\sigma_2 = \sigma_3 \circ \sigma_4 = [f(y, a) / v, b / y]$
 $\sigma = \sigma_1 \circ \sigma_2 = [f(y, a) / X, f(y, a) / v, b / y]$
 $MGU \sigma = [f(b, a) / X, f(b, a) / v, b / y]$

2. $f(g(v), f(x, y)) \stackrel{?}{=} f(x, f(y, v))$

1. $g(v) \stackrel{?}{=} x$
 $\sigma_1 = [g(v) / X]$
2. $f(x, y) \stackrel{?}{=} f(y, v)$
3. $X \stackrel{?}{=} y$
 $X_{\sigma_1} \stackrel{?}{=} y \rightarrow g(v) \stackrel{?}{=} y$
 $\sigma_3 = [g(v) / y]$
4. $y \stackrel{?}{=} v$
 $Y_{\sigma_3} \stackrel{?}{=} v \rightarrow g(v) \stackrel{?}{=} v$ Fail check occurs

3. $h(u, f(g(v), w), g(w)) \stackrel{?}{=} h(f(x, b), v, z)$

1. $v \stackrel{?}{=} f(x, b)$
 $\sigma_1 = [f(x, b) / v]$
2. $f(g(v), w) \stackrel{?}{=} u$
 $f(g(v), w) \stackrel{?}{=} u_{\sigma_1} \rightarrow f(g(v), w) \stackrel{?}{=} f(x, b)$
 $g(v) \stackrel{?}{=} x \rightarrow \sigma_2 = [g(v) / x]$
 $w \stackrel{?}{=} b \rightarrow \sigma_4 = [b / w]$
 $\sigma_2 = \sigma_1 \circ \sigma_4 = [g(v) / x, b / w]$
3. $g(w) \stackrel{?}{=} z \Rightarrow \sigma_5 = [g(w) / z]$
4. $MGU \sigma = \sigma_1 \circ \sigma_2 \circ \sigma_4 = [f(x, b) / v, g(v) / x, b / w, g(w) / z]$

 $MGU \sigma = [f(g(v), b) / v, g(v) / x, b / w, g(w) / z]$

▷	$\% \text{addr}(x,y) :- X \text{ holds the address of } Y$	$\text{conn}(x,y) :- \text{addr}(x,y)$
▷	$\% \text{serv}(x) :- X \text{ is an address serv/}$	$\text{conn}(x,z) :- \text{addr}(z,x), \text{serv}(z), \text{addr}(z,y)$
▷	$\% \text{conn}(x,y) :- X \text{ can initiate a connection to } Y$	
▷	$\% \text{knowing}(x,y) :- \text{either end can initiate a connection}$	$\text{knowing}(x,y) :- \text{addr}(x,y), \text{conn}(y,x)$
▷	$\text{addr}(a,d)$	
▷	$\text{addr}(a,b).$? - $\text{knowing}(w,a).$
▷	$\text{addr}(b,c).$	
▷	$\text{addr}(c,a).$? - $\text{conn}(w,a), \text{conn}(a,w).$
▷	$\text{serv}(b).$	
▷		
▷	$? - \text{addr}(w,a), \text{conn}(a,w).$? - $\text{addr}(w,z), \text{serv}(z), \text{addr}(z,a), \text{conn}(a,w)$
▷		
▷	$? - \text{addr}(z,a)$? - $\text{addr}(z_2, z_1), \text{server}(z_1), \text{addr}(z_1, a), \text{conn}(a, w)$
▷		
▷	$ z_1 = a z_2 = a z_3 = b z_4 = c$	$ z_2 = a, z_1 = d z_2 = a, z_1 = b z_2 = b, z_1 = c z_2 = c, z_1 = d$
▷	$? - \text{addr}(a,a) ? - \text{addr}(a,a) ? - \text{addr}(a,b) ? - \text{addr}(c,a)$	$? - \text{addr}(a,b) ? - \text{serv}(b) ? - \text{addr}(b,a) ? - \text{addr}(c,a) ? - \text{serv}(a) ? - \text{addr}(c,a)$
▷	fail	fail
▷	fail	fail
▷	fail	fail
▷		
▷	$? - \text{addr}(a,d) ? - \text{serv}(d) ? - \text{addr}(d,a)$	$? - \text{addr}(b,c) ? - \text{serv}(c) ? - \text{addr}(c,a)$
▷	fail	fail
▷	fail	fail
▷		

3 Paper

...

4 Conclusions

(approx 400 words) A critical reflection on the content of the course. Step back from the technical details. How does the course fit into the wider world of software engineering? What did you find most interesting or useful? What improvements would you suggest?

References

[ALG] [Algorithm Analysis](#), Chapman University, 2023.