CPSC-406 Report

Marc Nathaniel Domingo
Chapman University

March 19, 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 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.

1.5 Week 5

During this week, the lectures covered the concepts of Turing Machines and their relation to Polynomial and Non-Polynomial Time Problems.

1.6 Week 6

During this week, the lectures reviewed Satisfiability as covered during our lectures on Non-Polynomial Time Problems in their application to a Sudoku Solver.

1.7 Week 7

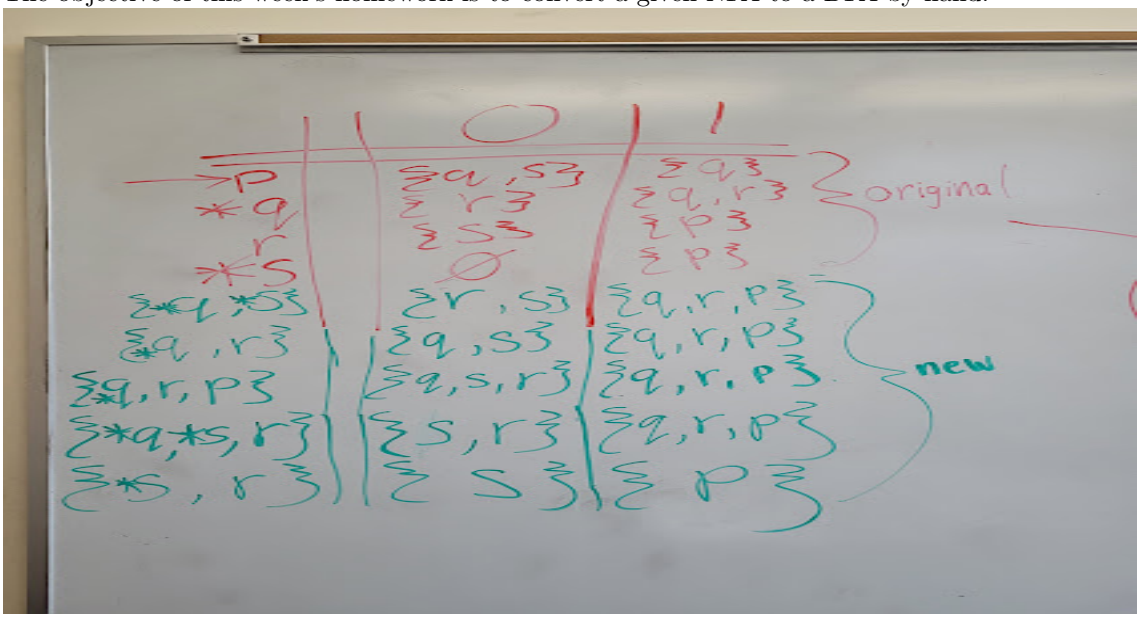
During this week, the lectures covered the concept of Distributed Systems in terms of how to create a Logical Clock.

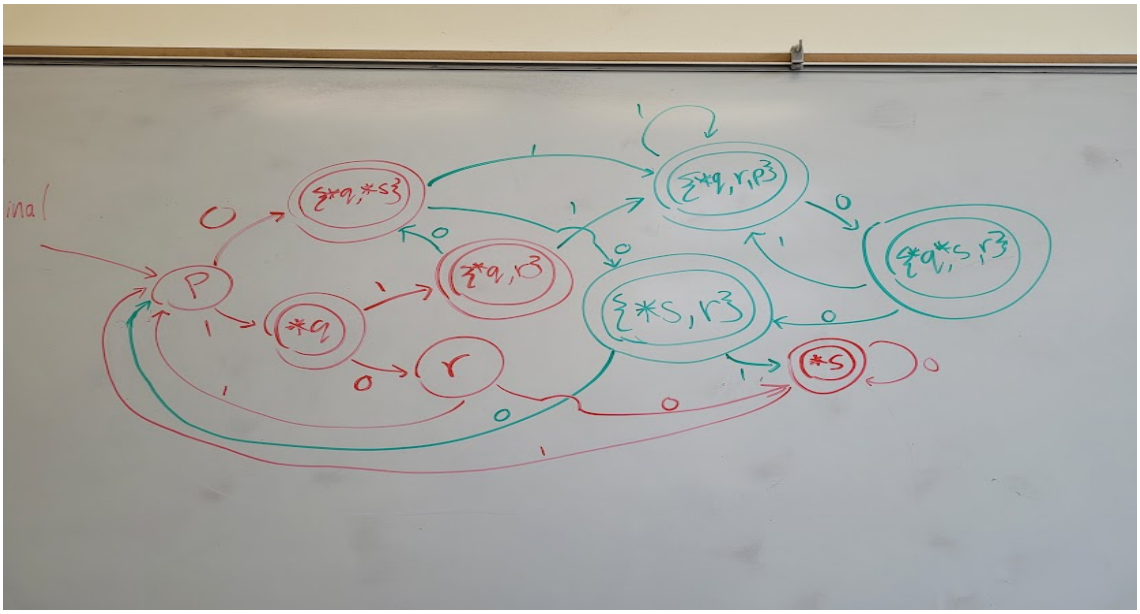
2 Homework

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. $f(x, f(x,Y)) \stackrel{?}{=} f(f(Y,a), f(U,b))$

1. $X \stackrel{?}{=} f(Y,a)$

$\sigma_1 = [f(Y,a)/X]$

2. $f(X,Y) \stackrel{?}{=} f(U,b)$

3. $X \stackrel{?}{=} U$

$X\sigma_1 \stackrel{?}{=} U \rightarrow f(Y,a) \stackrel{?}{=} U$

$\sigma_3 = [f(Y,a)/U]$

4. $Y \stackrel{?}{=} b$

$\sigma_4 = [b/Y]$

$\sigma_2 = \sigma_3 \circ \sigma_4 = [f(Y,a)/U, b/Y]$

$\sigma = \sigma_1 \circ \sigma_2 = [f(Y,a)/X, f(Y,a)/U, b/Y]$

$MGU\ \sigma = [f(b,a)/X, f(b,a)/U, b/Y]$

2. $f(g(U), f(X,Y)) \stackrel{?}{=} f(X, f(Y,U))$

1. $g(U) \stackrel{?}{=} X$

$\sigma_1 = [g(U)/X]$

2. $f(X,Y) \stackrel{?}{=} f(Y,U)$

3. $X \stackrel{?}{=} Y$

$X\sigma_1 \stackrel{?}{=} Y \rightarrow g(U) \stackrel{?}{=} Y$

$\sigma_3 = [g(U)/Y]$

4. $Y \stackrel{?}{=} U$

$Y\sigma_3 \stackrel{?}{=} U \rightarrow g(U) \stackrel{?}{=} U$ Fail Check Occurs

3. $h(U, f(g(V), W), g(W)) \stackrel{?}{=} h(f(X,b), U, Z)$

1. $U \stackrel{?}{=} f(X,b)$

$\sigma_1 = [f(X,b)/U]$

2. $f(g(V),W) \stackrel{?}{=} U$

$f(g(V),W) \stackrel{?}{=} U\sigma_1 \rightarrow f(g(V),W) \stackrel{?}{=} f(X,b)$

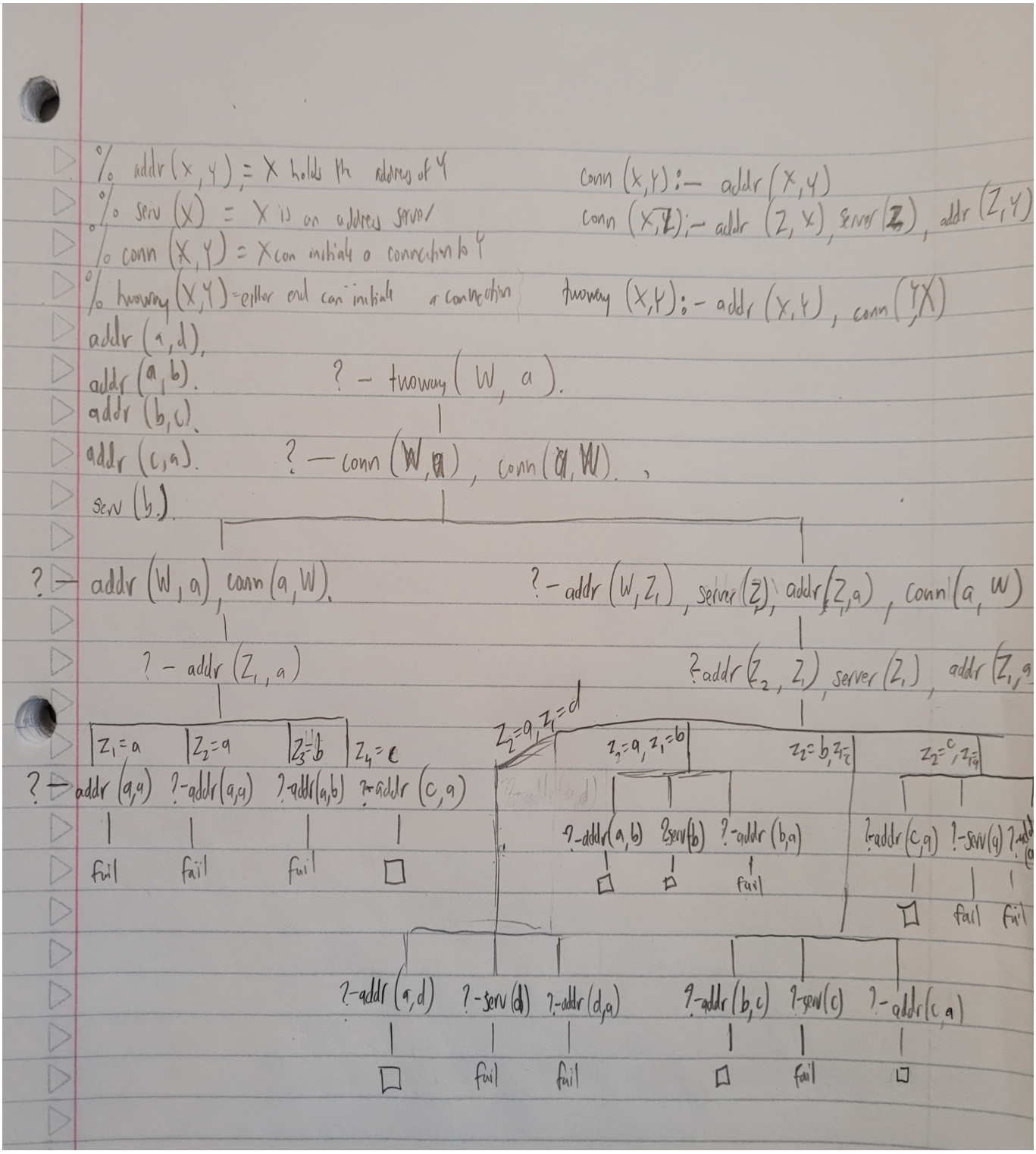
3. $g(V) \stackrel{?}{=} X \rightarrow \sigma_3 = [g(V)/X]$

4. $W \stackrel{?}{=} b \rightarrow \sigma_4 = [b/W]$ $\sigma_2 = \sigma_3 \circ \sigma_4 = [g(U)/X, b/W]$

5. $g(W) \stackrel{?}{=} Z \Rightarrow \sigma_5 = [g(W)/Z]$

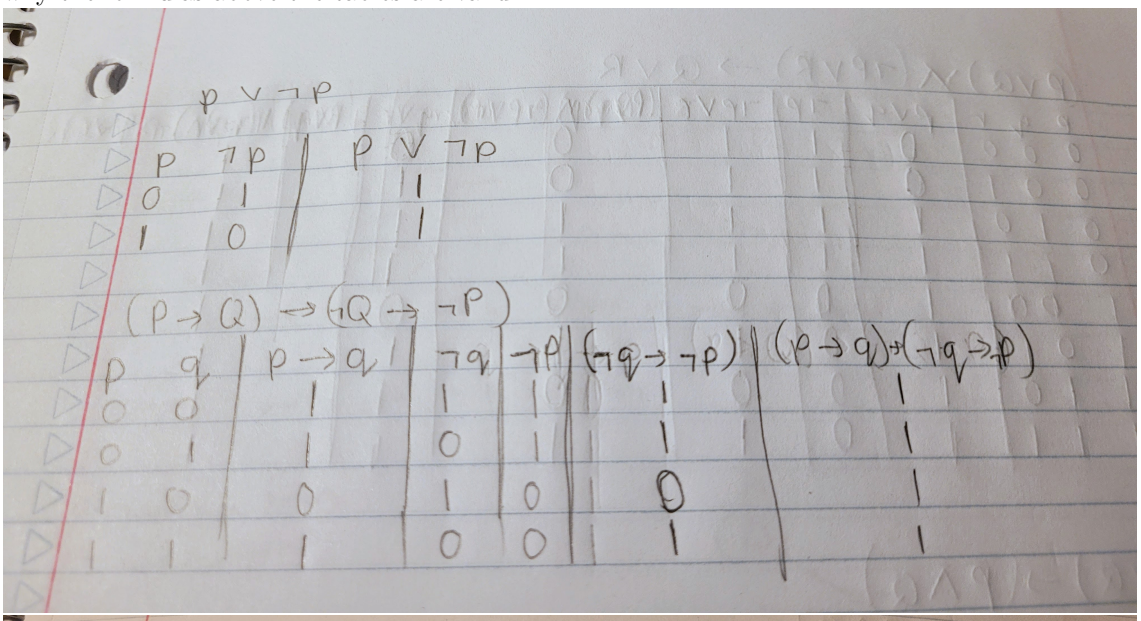
$MGU\ \sigma = \sigma_1 \circ \sigma_2 \circ \sigma_5 = [f(X,b)/U, g(U)/X, b/W, g(W)/Z]$

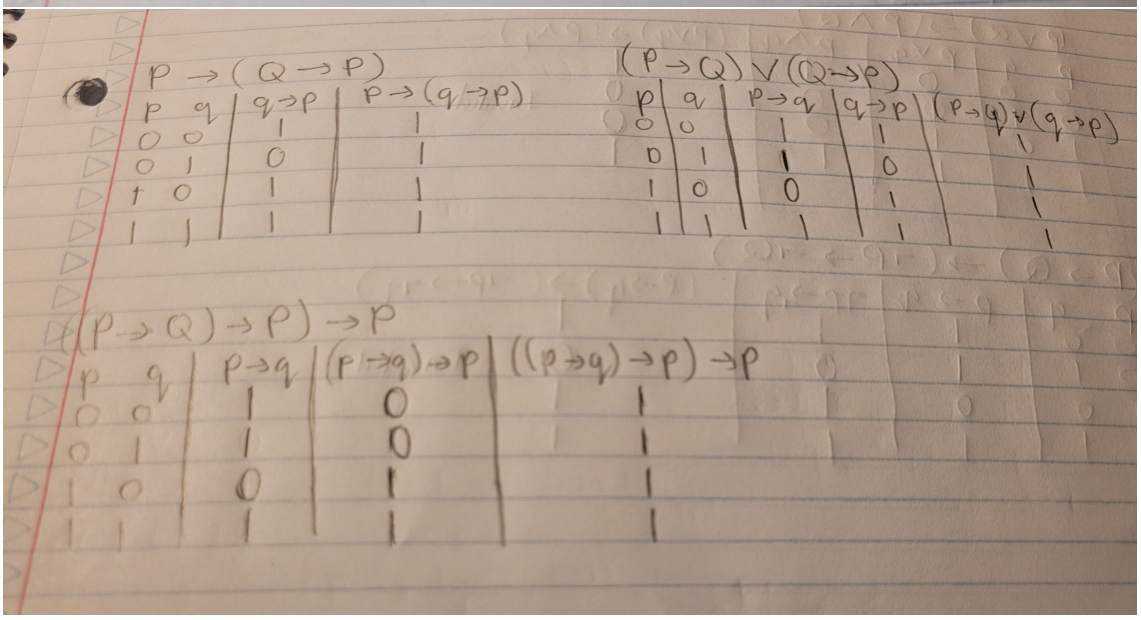
$MGU_\sigma = [f(g(V),b)/U, g(V)/X, b/W, g(W)/Z]$

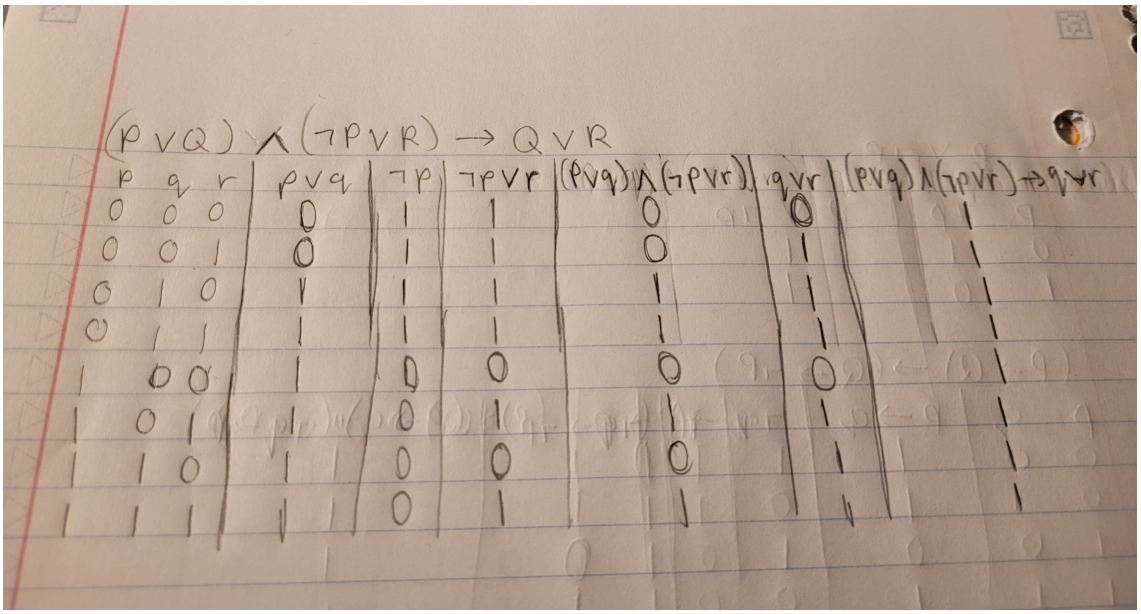

% addr(X,Y) = X holds the address of Y
% serv(X) = X is an address server
% conn(X,Y) = X can initiate a connection to Y
% twoway(X,Y) = either end can initiate a connection
addr(a,d).
addr(a,b).
addr(b,c).
addr(c,a).
serv(b).
conn(X,Y) :- addr(X,Y)
conn(X,Y) :- addr(Z,X), serv(Z), addr(Z,Y)
twoway(X,Y) :- addr(X,Y), conn(Y,X)
?- twoway(W,a).
?- conn(W,a), conn(a,W).
?- addr(W,a), conn(a,W).
?- addr(W,Z1), serv(Z1), addr(Z1,a), conn(a,W)
?- addr(Z1,a)
?- addr(Z2,Z1), serv(Z1), addr(Z1,
Z1=a
Z2=a
Z3=b
Z4=c
Z1=a, Z1=d
Z2=a, Z1=b
Z2=b, Z1=c
Z2=c, Z1=a
?- addr(a,a)
?- addr(a,a)
?- addr(a,b)
?- addr(c,a)
?- addr(a,b)
?- serv(b)
?- addr(b,a)
?- addr(c,a)
?- serv(a)
fail
fail
fail
fail
?- addr(a,d)
?- serv(d)
?- addr(d,a)
?- addr(b,c)
?- serv(c)
?- addr(c,a)
fail
fail
fail
fail

2.3 Week 6

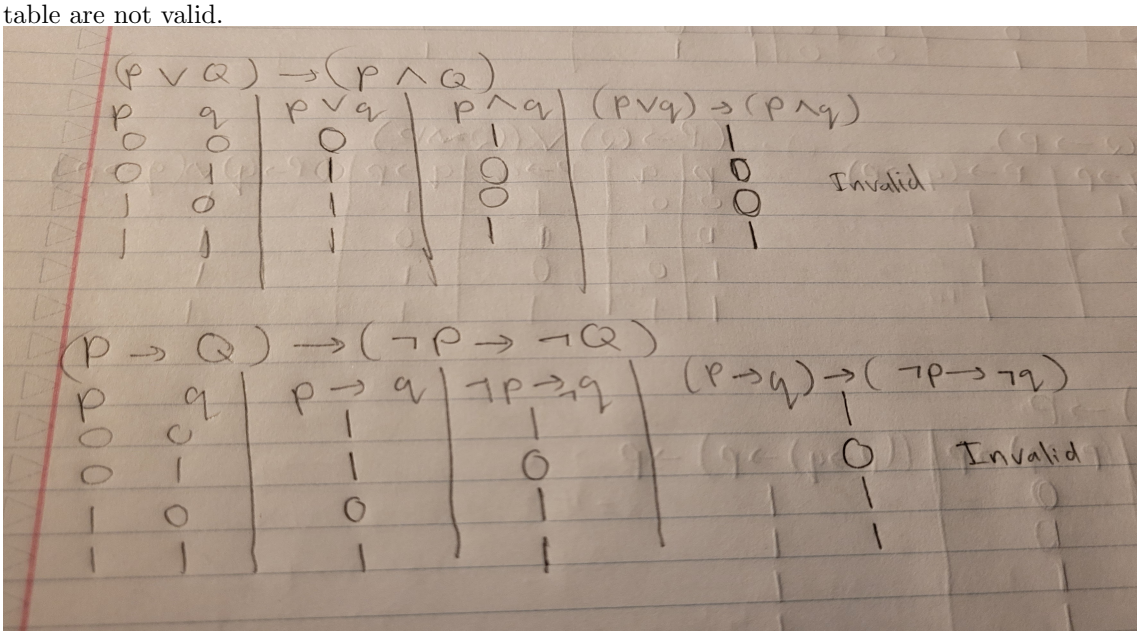
The objective of this week's homework was to create Indirect Truth Tables to understand Satisfiability in terms of Propositional Logic. For the following tables, we used the Indirect Truth Table method to prove why the formulas above the tables are valid.







For the following tables, the Indirect Truth Table method was used to prove why the formulas above the table are not valid.



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.