CALIFORNIA STATE UNIVERSITY, NORTHRIDGE

Survey of Imperative Style Turing Complete proof techniques and an application to prove Proteus Turing Complete

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science in
Computer Science

By

Isaiah Martinez

December 2024

The thesis of Isaiah Martinez is approved:

Kyle Dewey, PhD., Chair	Date
John Noga, PhD.	Date
Name	Date

Acknowledgements

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus lacinia odio vitae vestibulum vestibulum. Cras venenatis euismod malesuada. Maecenas vehicula felis quis eros auctor, sed efficitur erat suscipit. Curabitur vel lacus velit. Proin a lacus at arcu porttitor vehicula. Mauris non velit vel lectus tincidunt ullamcorper at id risus. Sed convallis sollicitudin purus a scelerisque. Phasellus faucibus purus at magna tempus, sit amet aliquet nulla cursus.

Table of Contents

List of Figures

This list must reference the figure, page it appears, and subject matter.

List of Tables

This list must reference the table, page it appears, and subject matter.

List of Illustrations

This list must reference the illustration, page it appears, and subject matter.

Abstract

TITLE GOES HERE

By

Isaiah Martinez

Master of Science in Computer Science

Abstract which will cover the contents of the entire paper in less than 350 words or so. 1.5 pgs double spaced

state research probem briefly describe methods and procedures used in gathering data or studying the problem give a condensed summary of the findings of the study

Chapter 1

Introduction

1.1 Turing Machines

What is a TM?

What are some examples?

Why are they important?

1.2 Turing Complete

What does TC mean?

How is it different from a TM?

Are all TC things equivalent?

What are the results of being TC?

1.2.1 Example of Turing Complete Systems

Some classical/practical examples

Link some funny/interesting examples of TC that have been proven

Chapter 2

Different Proof Techniques for demonstrating Turing Completeness

2.1 Turing Complete Proofs Overview

We will be exploring the different approaches to demonstrate TC

All proofs are equivalent in goal. The difference lies in their usability for specific domains.

2.2 Computer Engineering

2.2.1 Gates & Wiring

NOR, ADDER, OR, NOT gates

2.3 Computer Science

2.3.1 State Machines

2.3.1.1 Formal Technical Writing

$\lambda\gamma\tau$ etc. being used to describe TC

2.3.1.2 Gamified Writing

$\lambda\gamma\tau$ etc. being used to describe TC, but based on a very verbose and interactive style

2.3.2 Software Implementation

Demonstrate an example of a problem/scenario which shows TC.

2.3.2.1 Conway's Game of Life

Classic example to implement

2.3.2.2 Rule 110

Simpler example to implement

2.3.2.3 Calculator with Store Value

More challenging to implement than the previous two.

2.3.3 Interpreter for TC lang

Implement an interpreter for a known TC language in a given language.

2.3.3.1 Brainfuck

Describe Brainfuck

2.3.3.2 C

Is C TC?

Show how this

2.4 Mathematics

2.4.1 Lambda Calculus

Whatever the fuck this is, is TC

Chapter 3

Proteus is Turing Complete

This section will describe how we will construct 1+ proofs for showing that Proteus is TC.

3.1 The Proteus Grammar

List out the grammar and describe how it can be read, line by line. Probably have a small example of a program that shows most of (maybe all) of the functionality with Proteus

3.2 Initial Thoughts based off of the grammar

List out some ideas that led to the idea that Proteus could be TC

3.3 Proof Outline

Theorems that would be used Steps for proof outline

3.4 Proof v1

Formal proof that follows the outline

3.5 Implementing Rule 110

Implementation of Rule 110 that is a demonstration of TC.

Chapter 4

Conclusion

Succinctly describe what was the several techniques. Compare and constrast them? Perhaps a discussion section?

Bibliography

[1] Baker, N. 1966, in Stellar Evolution, ed. R. F. Stein & A. G. W. Cameron (Plenum, New York) 333