Boolean Algebra Calculator

Software Requirements Specifications

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 19/03/2024 | 0.1 | Completed Section 2 | Benjamin Weinzirl |
| 20/03/2024 | 0.2 | Completed Section 1 | Ibrahim Sufi |
| 22/03/2024 | 0.3 | Completed Section 3 | Del Endecott |
| 24/03/2024 | 0.4 | Completed Section 4 | Maya Routh |
| 24/03/2024 | 1.0 | Combined sections into single document | Benjamin Weinzirl |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Overall Description 4

2.1 Product perspective 4

2.1.1 User Interfaces 4

2.1.2 Software Interfaces 4

2.1.3 Memory Constraints 4

2.2 Product functions 4

2.3 User characteristics 4

2.4 Constraints 5

2.5 Assumptions and dependencies 5

3. Specific Requirements 5

3.1 Functionality 5

3.1.1 Language and Structure 5

3.1.2 Operators 5

3.1.3 Variable Definition, Result Output 5

3.1.4 Error handling 5

3.2 Use-Case Specifications 6

3.3 Supplementary Requirements 6

4. Classification of Functional Requirements 6

5. Appendices 7

Software Requirements Specifications

# Introduction

## Purpose

The purpose of this Software Requirements Specification document is to describe the external behavior of the Boolean Algebra Calculator that will be built for the final project of EECS 348. This document describes the nonfunctional requirements and the design constraints of the Boolean Algebra Calculator as well as proves a comprehensive description of the software requirements. This will facilitate effective collaboration between team members and ensure the final project meets required specifications.

## Scope

The Boolean Algebra Project should allow the user to input a Boolean expression either with Trues and Falses or with variables and should output either the simplified result of the expression (True or False) or it should output a truth table for the expression in terms of the variable. A use case model for this system is available in section 3.2.

## Definitions, Acronyms, and Abbreviations

No unfamiliar terms, abbreviations or acronyms are used

## References

There are no major references.

## Overview

This *Software Requirements Specification* document contains the following information:

Overall Description — describes the general factors that affect the product and its requirements including functionality and use case specification.

Specific Requirements — contains all software requirements to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements.

Classification of Functional Requirements— lists all function requirements ordered by type (Essential, Desirable, and Optional)

# Overall Description

## Product perspective

### User Interfaces

The user will interface with the command line in their powershell of choice. This program will specifically be able to run on KU cycle servers.

### Software Interfaces

The software will run on a Linux-based operating system.

### Memory Constraints

The program should be able to provide memory for defining variables throughout the project. The overall program should take less than 1MB.

## Product functions

The product will function as a Boolean algebra calculator, taking expressions from the user and outputting either true or false based on defined variables along with a truth table for the expression for undefined variables.

## User characteristics

Users will be on KU cycle servers and interested in the outputs of Boolean expressions. Users are expected to value functionality over quality.

## Constraints

The program must function properly on KU cycle servers and must follow the guidelines provided by the class. There is a time constraint present, where the program must be finished by the end of the 2024 Spring semester.

## Assumptions and dependencies

This project assumes that the users require only a text-based output from their text-based input. Users are assumed to input one expression into the program for each run-through. A minimum power requirement is assumed, as the program should not be draining on the user’s hardware.

# Specific Requirements

## Functionality

### Language and Structure

The program will be coded in C++ using object orientation principles. The code must include detailed comments describing the purpose of code segments, user-defined functions, and classes, etc.

### Operators

The program must support expressions containing the Boolean logical operations AND, OR, NOT, NAND, and XOR. The program must be able to parse expressions with multiple different operators in infix notation, respective parentheses, and operator precedence. The program must also be able to correct for excessive paratheses use.

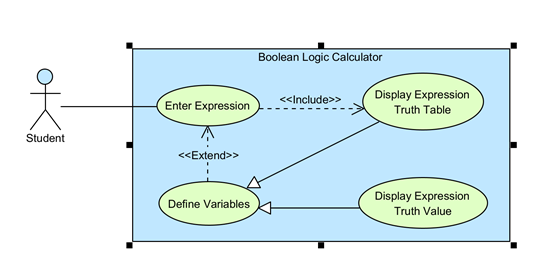
### Variable Definition, Result Output

Any letter present in the expression will be considered a variable. The user must be able to define truth values for each variable. If no truth values are defined for the variables, the program must correctly calculate and display the expression’s entire truth table. If the variables’ truth values are known, the user can choose for the program to display either the entire truth table or only the truth value of the expression.

### Error handling

The program must be able to handle multiple types of error and display a descriptive error message. Errors include, but are not limited to, missing parentheses and unknown operators. Any other types of errors discovered during program testing must be accounted for and handled by the program in its next iteration.

## Use-Case Specifications



## Supplementary Requirements

Program should be compatible with the KU EECS cycle servers.

# Classification of Functional Requirements

|  |  |
| --- | --- |
| **Functionality** | **Type** |
| Contain Boolean operators: NOT, AND, OR, XOR | Essential |
| Compute truth value of Boolean expressions containing TRUE and FALSE | Essential |
| Compute truth table of Boolean expressions containing variable values | Essential |
| Allow user input of Boolean expressions | Essential |
| Sleek User Interface to display input and output of calculator | Essential |
| Contain Boolean operator complements: NAND, NOR, XNOR | Desirable |
| Display the input as a logical circuit | Desirable |
| Display results as a Venn diagram | Desirable |
| Compute minimal forms of Boolean expressions | Optional |
| Mode switch to display Boolean operators as words or symbols | Optional |

# Appendices

No appendices applicable to this document.