Tech Titans

Boolean Logic Simulator

User's Manual

Version 1.1

Revision History

Date	Version	Description	Author
01/05/2024	1.0	Initial document work, everything but section 4	Theodora Okhagbuzo
01/05/2024	1.1	Section 4	Caden LeCluyse

Table of Contents

Test Case

1. Purpose

This manual serves as a comprehensive guide to using the Boolean Simulator software, providing clear instructions for users of all skill levels on the functionality and features of the tool that is designed for evaluating boolean expressions.

2. Introduction

The Boolean Simulator is a user-friendly application for evaluating Boolean logic. It interprets expressions involving logical constants and operators, yielding results that aid in learning and debugging logical operations.

Features include:

- Multiple logical operators: **AND**, **OR** , **NOT**, **NAND**, **XOR**
- Continuous mode for uninterrupted expression evaluation
- Clear error messages without exiting the program

Installation is straightforward across Linux, macOS, and Windows (via **WSL**), following the provided instructions for each operating system.

3. Getting started

Installation and Running the Program:

Linux:

```
sudo apt-get install cmake
sudo apt-get install clang
cmake -B build && make -C build
./build/boolean_simulator
```

macOS:

```
brew install cmake
brew install clang
cmake -B build && make -C build
./build/boolean_simulator
```

Windows:

Install **WSL** and follow Linux instructions.
As an alternative, you can use your own installation method for cmake, clang, and make.

Usage

Run the executable to begin evaluating boolean expressions. Enter your expression when prompted. Encapsulate expressions in single quotes if not in continuous mode and passing in the expression at runtime.

Expression Entry:

- Input **T** for a true value, and **F** for a false value.
- Use operators like **&** (AND), **|** (OR), **!** (NOT) , **$** (XOR), and **@** (NAND).
    - For any confusion on **|** (OR), it is the key above the enter key.
- Combine expressions with parentheses for precedence.

Interpreting Results:
The result will be **True** or **False**, reflecting the logical outcome of the evaluated expression.

4. Advanced features

- History:
  - This program will store a history of all previously evaluated expressions while in continuous mode.
  - Entering **history** while in continuous mode will print all of the previously evaluated expressions.
- Test mode (mainly intended for developer use):
  - This program has a test mode that can be utilized with the **-t** and **–test** flags.
  - Entering the program in test mode expects a file **expressions.txt** to be present in the current working directory, with one boolean expression per line.
  - The results of each evaluated expression will be placed in **results.txt**

5. Troubleshooting

Common Issues:

- Syntax Errors: Ensure expressions are correctly formatted, with proper operators and operands.
- If you are passing in an expression at runtime, make sure the expression is wrapped in single quotes.
- Installation Issues: verify that all dependencies are installed and paths are set correctly.
  If issues persist, consult the FAQs or reach out to support.

6. Examples

Evaluate the AND operation:

$ boolean_simulator 'T & F'
Result: False!

Evaluating expressions with continuous mode:

$ boolean_simulator -c
Please enter your boolean expression, or enter history to see all prior evaluated expressions (enter exit, quit, or q to exit the program): T | F
Result: True!

Please enter your boolean expression, or enter history to see all prior evaluated expressions (enter exit, quit, or q to exit the program): T @ T
Result: False!

Please enter your boolean expression, or enter history to see all prior evaluated expressions (enter exit, quit, or q to exit the program): (T @ T) & F
Result: False!

Please enter your boolean expression, or enter history to see all prior evaluated expressions (enter exit, quit, or q to exit the program): history
Expression: T | F
Result: True!
Expression: T @ T
Result: False!
Expression: (T @ T) & F
Result: False!

Please enter your boolean expression, or enter history to see all prior evaluated expressions (enter exit, quit, or q to exit the program): quit
exiting…

7. Glossary of terms

- **Boolean/operand**: A binary variable, having two possible values **True** and **False**.
- **Operator**: Symbol that denotes operations such as AND, OR, NOT, NAND, XOR.
    - **AND &**: Returns true if both values are true.
    - **OR |**: Returns true if one value is true.
    - **NOT !**: Negates the value it is currently in front of. False becomes true and vice versa
    - **NAND @**: Returns true if both values are not true.
    - **XOR $**: Returns true if both values are different. True xor false returns true.
- **Expression**: Combination of variables and operators forming a logical statement.
- **WSL**: Windows subservice for Linux. This utility runs a full instance of Linux inside windows.
- **Apt**: The package manager for Debian based distributions of Linux.
- **Brew**: The package manager for macOS.
- **CMake**: A cross-platform build system generator that allows for easy building across a wide variety of platforms and compilers.
- **Clang**: A C++ compiler. It turns C++ code into an executable that you can actually run.

8. FAQ

Q: Can I use lowercase **t** and **f** for true and false?
A: No, use uppercase **T'** and **F**.

Q: Can I use **True** and **False** instead?
A: No, use **T** and **F.**

Q: How can I exit continuous mode?
A: Type exit, quit, or q and press enter.

Q: Does it handle parentheses for expression grouping?
A: Yes, you can use parentheses to group expressions for evaluation.

Q: Can I use brackets?
A: No, just use parentheses.

Q: How to create a truth table?
A: This feature is planned for future updates.

Q: How do I use test mode?
A: Refer to the advanced features section.

Q: I'm getting a bash error when passing in an expression at runtime?
A: Wrap the expression in single quotes.

Q: Do spaces matter?
A: No, spaces are completely ignored by this program.