Incomplete Inc.

Boolean Algebra Calculator

Test Case

Version 1.0

Revision History

Date	Version	Description	Author
02/05/2024	1.0	Document created	Del Endecott

Table of Contents

Test Case

1. Purpose

This Test Case Specification document for the Boolean Algebra Calculator defines a test case for an item that should be tested. Each test case has six categories: Test case ID, test description, test data, expected results, actual results, and pass/fail status. The test case ID consists of the class of test case (either STC for standard or ATC for additional) and the number of the test. The test case description is a summary of the program's behavior that the input is intended to test. The test data is the contents of the input. The expected result is what the program is intended to output. The actual result is what the program outputs after the conclusion of the test. The pass/fail status indicates if the program outputs the correct value with no errors caused.

2. Standard Test Cases

Test Case ID	**STC01**
Test Case Description	Verify operators are parsed in the correct order
Test Data	(T | F) $ F
Expected Results	True
Actual Results	True
Pass/Fail Status	Pass

Test Case ID	**STC02**
Test Case Description	Ensure NOT operator functions correctly on parenthesized expression
Test Data	!(T & T)
Expected Results	False
Actual Results	False
Pass/Fail Status	Pass

Test Case ID	**STC03**
Test Case Description	Ensure sets of multiple parenthesized expressions are read left to right
Test Data	(F @ T) | (T @ F)
Expected Results	True
Actual Results	True
Pass/Fail Status	Pass

Test Case ID	**STC04**
Test Case Description	Verify AND operator only returns True when both operands are True
Test Data	(T $ T) & F

Expected Results	False
Actual Results	False
Pass/Fail Status	Pass

Test Case ID	**STC05**
Test Case Description	Ensure NOT operator functions correctly within expression
Test Data	!F | !T
Expected Results	True
Actual Results	True
Pass/Fail Status	Pass

Test Case ID	**STC06**
Test Case Description	Check nested expressions evaluate in correct order
Test Data	(((((T | F) & F) | (T & (T | F))) @ (T @ T)) $ (! (T | F)))
Expected Results	True
Actual Results	True
Pass/Fail Status	Pass

Test Case ID	**STC07**
Test Case Description	Check that extraneous parentheses are handled correctly when True
Test Data	((F $ ((T | F) & (F @ (T | F)))) | (T $ (T & F)))
Expected Results	True
Actual Results	True
Pass/Fail Status	Pass

Test Case ID	**STC08**
Test Case Description	Check that extraneous parentheses are handled correctly when False
Test Data	(((! (T $ F)) & (T @ T)) | ((F | T) & (T $ T)))
Expected Results	False
Actual Results	False
Pass/Fail Status	Pass

Test Case ID	**STC09**
Test Case Description	Check that extraneous parentheses around operands with NOT are handled correctly when True
Test Data	(((T @ T) $ (F @ T)) | ((!T) & (T | (!T))))
Expected Results	True
Actual Results	True
Pass/Fail Status	Pass

Test Case ID	**STC10**
Test Case Description	Check that extraneous parentheses around operands with NOT are handled correctly when False
Test Data	((F @ T) $ (T | (F & F))) & (T & (T @ (!T)))
Expected Results	False
Actual Results	False
Pass/Fail Status	Pass

Test Case ID	**STC11**
Test Case Description	Ensure missing operand is handled correctly
Test Data	! & T
Expected Results	Error – invalid expression
Actual Results	Error – invalid expression
Pass/Fail Status	Pass

Test Case ID	**STC12**
Test Case Description	Ensure unknown operator is handled correctly
Test Data	T ? T
Expected Results	Error – invalid expression
Actual Results	Error – invalid expression
Pass/Fail Status	Pass

Test Case ID	**STC13**
Test Case Description	Ensure mismatched parentheses are handled correctly
Test Data	(T |) F
Expected Results	Error – invalid expression

Actual Results	Error – invalid expression
Pass/Fail Status	Pass

Test Case ID	**STC14**
Test Case Description	Ensure circular logic is handled correctly
Test Data	T = !(T & T)
Expected Results	Error – invalid expression
Actual Results	Error – invalid expression
Pass/Fail Status	Pass

Test Case ID	**STC15**
Test Case Description	Ensure an empty input is handled correctly
Test Data	<empty input>
Expected Results	Error – invalid expression
Actual Results	<empty output>
Pass/Fail Status	Pass

Test Case ID	**STC16**
Test Case Description	Ensure extra operators are handled correctly
Test Data	T &&& F
Expected Results	Error – Adjacent inner operations
Actual Results	Error – Adjacent inner operations
Pass/Fail Status	Pass

Test Case ID	**STC17**
Test Case Description	Ensure unassigned variables are handled correctly
Test Data	X | Y
Expected Results	Error - invalid expression
Actual Results	Error - invalid expression
Pass/Fail Status	Pass

Test Case ID	**STC18**
Test Case Description	Ensure incorrectly positioned NOT is handled correctly

Test Data	T!
Expected Results	Error – Expression cannot end in '!'
Actual Results	Error – Expression cannot end in '!'
Pass/Fail Status	Pass

Test Case ID	**STC19**
Test Case Description	Ensure calculator is case-sensitive
Test Data	t & f
Expected Results	Error – invalid expression
Actual Results	Error – invalid expression
Pass/Fail Status	Pass

3. Additional Test Cases

Test Case ID	**ATC01**
Test Case Description	Error when parsing adjacent Truth values
Test Data	TT
Expected Results	Error – invalid expression
Actual Results	Error – invalid expression
Pass/Fail Status	Pass

Test Case ID	**ATC02**
Test Case Description	Parses left to right when there are no ()
Test Data	T|F&T&T
Expected Results	T
Actual Results	T
Pass/Fail Status	Pass

Test Case ID	**ATC03**
Test Case Description	Parses from left to right inside innermost parenthesis
Test Data	T | (T|F&T&T)
Expected Results	Error – invalid expression
Actual Results	Error – invalid expression
Pass/Fail Status	Pass

Test Case ID	**ATC04**
Test Case Description	Not Not cancels out
Test Data	!(!T)
Expected Results	T
Actual Results	T
Pass/Fail Status	Pass

Test Case ID	**ATC05**
Test Case Description	Two operators in a row are handled correctly
Test Data	T||T
Expected Results	Error – invalid expression
Actual Results	Error – invalid expression
Pass/Fail Status	Pass

Test Case ID	**ATC06**
Test Case Description	Large expressions with unknown variables
Test Data	(T|F) & X
Expected Results	Error – invalid expression
Actual Results	Error – invalid expression
Pass/Fail Status	Pass

Test Case ID	**ATC07**
Test Case Description	NOR returns False when both sides are True
Test Data	T@T
Expected Results	F
Actual Results	F
Pass/Fail Status	Pass

Test Case ID	**ATC08**
Test Case Description	Single character expression
Test Data	T
Expected Results	T
Actual Results	T
Pass/Fail Status	Pass

Test Case ID	**ATC09**
Test Case Description	Operator Symbols are required
Test Data	T AND F
Expected Results	Error – invalid expression
Actual Results	Error – invalid expression
Pass/Fail Status	Pass

Test Case ID	**ATC010**
Test Case Description	Typing the words those an error
Test Data	True | True
Expected Results	Error – invalid expression
Actual Results	Error – invalid expression
Pass/Fail Status	Pass