

boolalg: Boolean Logic Truth Tables, Simplification, and Algebraic Proofs

Param Srivastava

Version 1.0.0 — October 22, 2026

1 Overview

The `boolalg` package provides pure- $\text{\LaTeX}$ macros for generating truth tables, simplifying Boolean expressions step-by-step, and proving algebraic equivalence (or finding counterexamples) directly in your document without external scripts or shell-escape.

2 Commands

- `\btable{<expr>}` — Truth table with intermediate sub-expression columns (aliases: `\btruth`, `\truthtable`, `\btruthtable`, `\booltruthtable`).
- `\btable*{<expr>}` — Truth table followed by the simplified expression.
- `\bsimplify{<expr>}` — Step-by-step algebraic simplification with law justifications (alias: `\boolsimplify`).
- `\bsimplify*{<expr>}` — Final simplified expression only.
- `\bprove{<lhs> = <rhs>}` — Step-by-step algebraic proof or counterexample (alias: `\boolprove`).

3 Input Syntax

Operators are listed from highest to lowest precedence:

Precedence	Operator	Accepted Input Syntax
Highest	Grouping	(...)
	Variables / Constants	A--Z, a--z, 0, 1
1	NOT	<code>not</code> , <code>\not</code> , <code>\lnot</code> , <code>\neg</code> , ' (postfix), <code>!</code> , <code>~</code>
2	AND	Implicit (AB), <code>*</code> , <code>.</code> , <code>&</code> , <code>and</code> , <code>\and</code> , <code>\land</code> , <code>\wedge</code> , <code>\cdot</code>
	NAND	<code>nand</code> , <code>\nand</code> , <code>\uparrow</code> , <code>\barwedge</code>
3	OR	<code>+</code> , <code>—</code> , <code>or</code> , <code>\or</code> , <code>\lor</code> , <code>\vee</code>
	NOR	<code>nor</code> , <code>\nor</code> , <code>\downarrow</code>
4	XOR	<code>^</code> , <code>xor</code> , <code>\xor</code> , <code>\oplus</code> , <code>\veebar</code>
	XNOR	<code>xnor</code> , <code>\xnor</code> , <code>\odot</code> , <code>\equiv</code> , <code>\leftrightarrow</code> , <code>\iff</code>

Note: Place a space after word operators (e.g., `not A`, `A and B`).

4 Configuration

You can customize output symbols and parameters using standard `\def`:

```
\def\boolfname{f}           % Name of the truth table output column (\empty to hide)
\def\booltablemaxvars{5}    % Maximum number of variables before table is omitted
\def\boolandsym{\land}      % Display symbol for AND
\def\boolorsym{\lor}        % Display symbol for OR
\def\boolnotsym{\lnot}      % Display symbol for NOT
\def\boolxorsym{\oplus}     % Display symbol for XOR
\def\boolxnorsym{\odot}     % Display symbol for XNOR
\def\boolnandsym{\uparrow}  % Display symbol for NAND
\def\boolnorsym{\downarrow}% Display symbol for NOR
```

5 Examples

5.1 Truth Table with Simplification

```
\btable*[A'B + AB']
```

Truth table: $f = (\neg A \wedge B) \vee (A \wedge \neg B)$

m	A	B	$\neg A \wedge B$	$A \wedge \neg B$	f
0	0	0	0	0	0
1	0	1	1	0	1
2	1	0	0	1	1
3	1	1	0	0	0

Simplified: $(\neg A \wedge B) \vee (A \wedge \neg B) = (\neg A \wedge B) \vee (A \wedge \neg B)$

5.2 Step-by-Step Simplification

```
\bsimplify{(A + B)(A + B')}
```

Simplify: $(A \vee B) \wedge (A \vee \neg B)$

$$\begin{aligned}
 (A \vee B) \wedge (A \vee \neg B) &= A \vee (B \wedge \neg B) \\
 &= A \vee 0 \\
 &= A
 \end{aligned}$$

Distributivity
Complement
Identity

Simplified: $(A \vee B) \wedge (A \vee \neg B) = A$

5.3 Algebraic Equivalence Proof

```
\bprove{A \oplus B = (A + B)(AB)'}%
```

$$\begin{aligned}
 A \oplus B &= (A \wedge \neg B) \vee (\neg A \wedge B) \\
 &= (B \wedge \neg A) \vee (A \wedge \neg B) \\
 &= (B \wedge \neg A) \vee ((A \wedge \neg B) \vee 0) \\
 &= (B \wedge \neg A) \vee ((A \wedge \neg B) \vee (B \wedge \neg B)) \\
 &= (B \wedge \neg A) \vee ((A \vee B) \wedge \neg B)
 \end{aligned}$$

Definition of XOR
Commutativity
Identity
Complement
Distributivity

$= (0 \vee (B \wedge \neg A)) \vee ((A \vee B) \wedge \neg B)$	Identity
$= ((A \wedge \neg A) \vee (B \wedge \neg A)) \vee ((A \vee B) \wedge \neg B)$	Complement
$= ((A \vee B) \wedge \neg A) \vee ((A \vee B) \wedge \neg B)$	Distributivity
$= (A \vee B) \wedge (\neg A \vee \neg B)$	Distributivity
$= (A \vee B) \wedge \neg(A \wedge B)$	De Morgan's law

5.4 Counterexample Generation for Non-Equivalent Expressions

`\bprove{A + B = AB}`

The two sides are not equal: taking $A=0$, $B=1$ the left side is 1 but the right side is 0.