

Boolean Logic

Derrick Hasterok

January 26, 2005

1 Introduction

This document is a introduction to Boolean logic, specifically for Matlab. The introduction of logic into mathematics began with Aristotle, who stated that A cannot be both B and not B (Law of Contradition) and A must either be B or not B (Law of Bivariance). Boolean logic, first conceived by George Boole (1815–1864), is a mathematical description of simple logic. Boole proposed that a number may belong the the universal set (denoted by 1) or the empty set (denoted by 0) and introduced the logical operators AND, OR and NOT. This is a very useful concept for computing because computers work in binary, i.e. ones and zeros.

2 Examples

The logical operators AND, OR, XOR (exclusive OR) and NOT have the following properties:

1	AND	1	=	1		1	OR	1	=	1
1	AND	0	=	0		1	OR	0	=	1
0	AND	0	=	0		0	OR	0	=	0
	NOT	1	=	0		1	XOR	1	=	0
	NOT	0	=	1		1	XOR	0	=	1
						0	XOR	0	=	0

The universal and empty sets, may also be thought of as TRUE (1) or FALSE (0). This can simplify the thought process for a human. However, for a computer the 1 and 0 are actually used. Boolean operations may be combined with comparison operators such as =, >, <, ≥, ≤ and ≠. They may also include arithmetic statements. Some examples:

$$\begin{array}{rcccl} (5 + 4 > 3) & \text{AND} & (12/2 \leq 6) & & \\ \text{true} & \text{AND} & \text{true} & = & \text{true} \end{array}$$

$$\begin{array}{rcccl} \text{NOT } (5 + 4 > 3) & \text{AND} & (12/2 \leq 6) & & \\ \text{NOT } \text{true} & \text{AND} & \text{true} & = & \text{false} \end{array}$$

$$\begin{array}{rcccl} \text{NOT } (5 + 4 < 3) & \text{OR} & (12/2 > 6) & & \\ \text{NOT } \text{false} & \text{OR} & \text{false} & = & \text{true} \end{array}$$

Although the importance of logical math is generally not apparent on paper, Boolean logic is incredibly useful for computer programming. Computers reduce all numbers to binary and carry out mathematical operations as Boolean operations (sounds simple... not really).

3 Boolean Logic in Matlab

Matlab uses the symbols `&`, `|` *xor* and `~` for AND, OR, XOR and NOT. The examples from above may be used in exactly the same way, but would be written:

$$\begin{array}{rcccl} (5 + 4 > 3) & \& (12/2 \leq 6) & & \\ 1 & \& 1 & = & 1 \\ \\ \sim (5 + 4 > 3) & \& (12/2 \leq 6) & & \\ \sim 1 & \& 1 & = & 0 \\ \\ \sim (5 + 4 < 3) & | & (12/2 > 6) & & \\ \sim 0 & | & 0 & = & 1 \end{array}$$

Boolean statements are used as expressions to initiate an *if/else* statement, *while* loop and for a *switch/case*. Boolean expressions are also occasionally used to set flags.

For more detailed information on Boolean logic, see: [Wikipedia.org](http://www.wikipedia.org) at <http://www.wikipedia.org/> and [Mathworld](http://mathworld.wolfram.com/) at <http://mathworld.wolfram.com/>.