Teaching London Computing

Programming for GCSE Topic 3.3: Boolean Logic and Truth Tables

Queen Mary University of London

King's College London

Computing at School — Educate · Engage · Encourage

Department for Education

Supported by Mayor of London

Network of Excellence Computer Science Teaching — University Partner

William Marsh
School of Electronic Engineering and Computer Science
Queen Mary University of London

Aims

- Introduce the study of logic
  - Introduction to logic
  - Truth tables
  - Logic and programming
  - Writing logic – Boolean algebra
  - …
- 'Logic gates' are covered later

Teaching Issue

- How to provide a coherent, joined up view
  - Some curricula include logic circuits but it is not related to operation of a computer
- Abstraction e.g. 'X and Y'
  - Best way in?
- Notation?
  - Unfortunately, several used

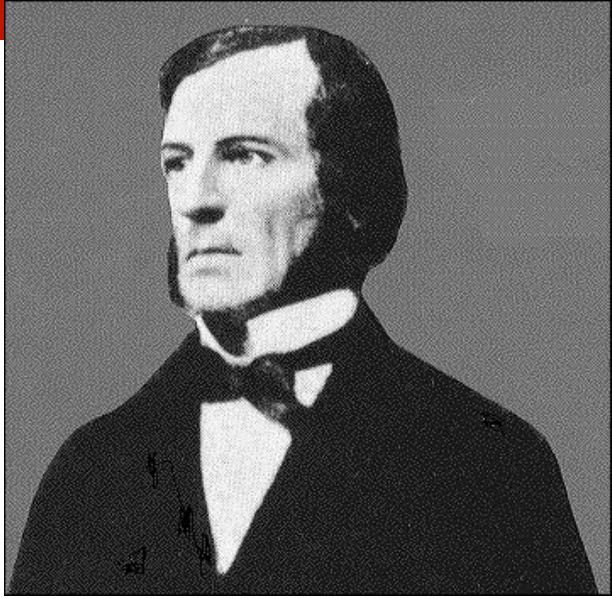
Introduction to Logic

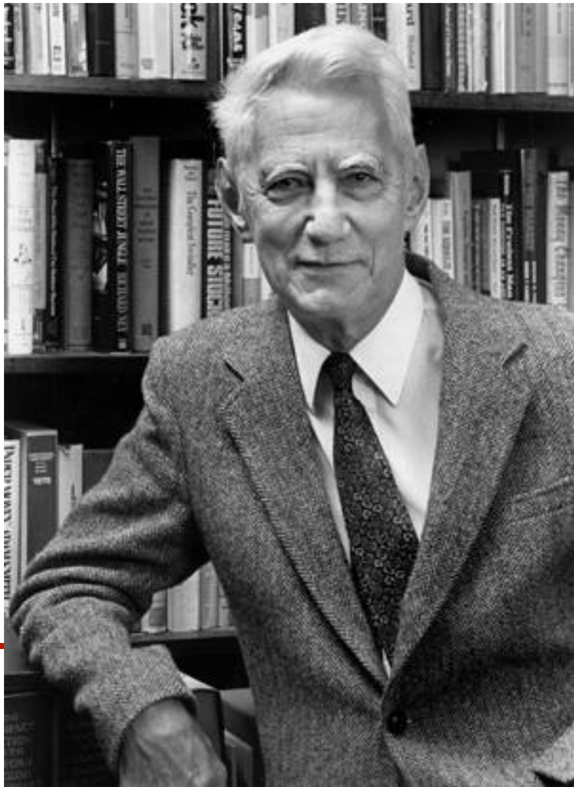
True and False

Some History

- George Boole
  - Invented ‘Boolean Algebra’
  - Makes 'logic' into mathematics
  - "An Investigation of the Laws of Thought", 1854

- Claude Shannon
  - A Symbolic Analysis of Relay and Switching Circuits, 1938
  - Based on his master’s thesis.





True or False?

- Logic view: true or false
  - 'My name is David'
  - 'Today is a Tuesday'
- Proposition: a statement that is true or false

- In reality, some statements are more complex
  - 'That colour suits you'
  - 'You are the most beautiful girl in the world'

Writing Logic

Boolean Algebra

Logical / Boolean Variables

- X, Y - (an uppercase letter)
- A variable that may be true of false
- A proposition

Writing Logic

Python	Logic Gates	Maths
X and Y	$X \cdot Y$	$x \wedge y$
X or Y	$X + Y$	$x \vee y$
not X	$\overline{X}$	$\neg x$
True	1	true
False	0	false

- 'X', 'Y' are Boolean variables

TRUTH TABLES:

AND, OR, NOT

Truth Tables

- Table of all variables in a Boolean formula
- 2 variables

X	Y	Result
0	0	
0	1	
1	0	
1	1	

- Each row has a possible combination of X and Y
- Table covers all possibilities

AND, OR

- OR

X	Y	X or Y
0	0	0
0	1	1
1	0	1
1	1	1

True when either X or Y true

- AND

X	Y	X and Y
0	0	0
0	1	0
1	0	0
1	1	1

True when both X and Y true

NOT

x	not x
0	1
1	0

Logic and Boolean Expressions

… Python programming

Boolean Expressions

- Combine conditions
- If A, B are conditions – true or false

Expression	Description
`A and B`	Both A true and B true
`A or B`	Either A true or B true
`not A`	True when A is false

Examples

- What are the values of the following?

Expression	True or False?
`10 > 5 or "hello"[1] == "e"`	
`2 + 3 < 10 and 6 != 6`	
`10 != 11 or 5 != 5`	
`10 < 10 and "world"[1:3] == "or"`	
`not (10 >= 11)`	

Compare these programs:

```
if age >= 21 and age <= 25:
   print("Great age")
else:
   print("Not so good! ")
```

Equivalent

Nested 'if' statement

```
if age >= 21:
   if age <= 25:
      print("Great age")
   else:
      print("Not so good!")
else:
   print("Not so good!")
```

Using Truth Tables to Check Equivalence

Truth Tables

- Table of all variables in a boolean formula
- 2 variables
  - 4 row
- 3 variables
  - 8 rows
- 4 variables
  - 16 rows
- Two formula same if (and only if) same truth table

X	**Y**	**Z**	**Result**
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Comparing Formulae using Truth Table

- Use a truth table to check equivalence

$$\overline{(A . B)} = \overline{A} + \overline{B}$$

Are the two formula the same:
- not (A and B)
- (not A) or (not B)

A	B	NOT(A . B)	(NOT A) + (NOT B)
0	0		
0	1		
1	0		
1	1		

Comparing Formulae using Truth Table

- Use a truth table to check equivalence

$$\overline{(A . B)} = \overline{A} + \overline{B}$$

A	B	NOT(A . B)	(NOT A) + (NOT B)
0	0	0	1	1
0	1	0	1	0
1	0	0	0	1
1	1	1	0	0

Comparing Formulae using Truth Table

- Use a truth table to check equivalence

$$(\overline{A . B}) = \overline{A} + \overline{B}$$

A	B	NOT	(A . B)	(NOT A)	+	(NOT B)
0	0	1	0	1	1	1
0	1	1	0	1	1	0
1	0	1	0	0	1	1
1	1	0	1	0	0	0

QUIZ: Are these the same?

```
if age >= 20 and age <= 25:
    print("Great age")
else:
    print("Not so good! ")
```

```
if age < 20 or age > 25:
    print("Not so good!")
else:
    print("Great age")
```

Comparing Formulae using Truth Table

- QUIZ: use a truth table to check

$$\overline{(A + B)} = \overline{A} . \overline{B}$$

A	B	NOT(A + B)	(NOT A) . (NOT B)
0	0		
0	1		
1	0		
1	1		

Rules of Boolean Algebra

Boolean Algebra Rules

- Rules for NOT

$A \cdot \overline{A} = 0$	*never A and not A*
$A + \overline{A} = 1$	*always A or not A*

- Associative rules

$A.(B.C) = (A.B).C$	*and associative*
$A+(B+C) = (A+B)+C$	*or associative*

De-Morgan's Laws

- Important law for exchanging AND with OR

$$\overline{(A . B)} = \overline{A} + \overline{B}$$

'A and B' is false when either A is false or B is false

$$\overline{(A + B)} = \overline{A} . \overline{B}$$

'A or B' is false when both A is false and B is false

Summary

- Logical expressions
  - AND, OR, NOT
- Logic and programming
  - Reasoning about conditions
- Truth table
  - Equivalence of expressions
- Boolean expression (formula)
  - Algebraic rules

- Next stop: logic circuits