

# **Truth and Logic**

## **Brief Introduction to Boolean Variables**

INFO-1301, Quantitative Reasoning 1  
University of Colorado Boulder

**February 13, 2017**

Prof. Michael Paul

Prof. William Aspray

# Overview

This lecture will...

- introduce boolean variables and operations,
- provide another way to think about sets, and
- teach you how to describe what's true in this world.

Boolean logic plays an important role in modern information systems and computational design.

# Another type of variable

The variables we've seen so far describe properties of the world:

- *ColorOfSky* = Blue

We can also create variables that correspond to **statements** about the world

- *The sky is blue* = True

# Another type of variable

A **boolean** variable represents a statement about the world called a **proposition**

Propositions have values like other variables, usually called **truth values**

- Domain of a boolean variable: {True, False}

Example:

Let  $P$  be the proposition “the Broncos won Super Bowl 50”

- $P = \text{True}$

# Another type of variable

A **boolean** variable represents a statement about the world called a **proposition**

Propositions have values like other variables, usually called **truth values**

- Domain of a boolean variable: {True, False}

Example:

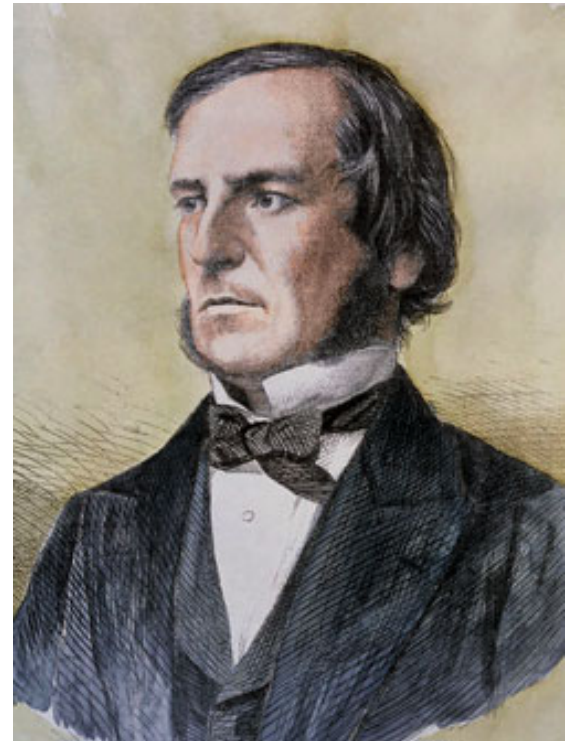
Let  $Q$  be the proposition “the Panthers won Super Bowl 50”

- $Q = \text{False}$

# Another type of variable

Boolean variables were invented by philosophers to reason about the world

If we know that certain statements are true or false, we can make conclusions about the truth value of other statements



George Boole, 1815-1864

# Boolean operations

With numerical variables, we have operations like addition and multiplication

With sets, we have operations like union, intersection, and complement

Boolean variables have their own operations:

- AND, OR, NOT

# Boolean operations

$P$  = “the Broncos won Super Bowl 50” = True

$Q$  = “the Panthers won Super Bowl 50” = False

We can combine these two propositions to create a new proposition with its own truth value



# Boolean operations: AND

$P$  = “the Broncos won Super Bowl 50” = True

$Q$  = “the Panthers won Super Bowl 50” = False

Notation:

$P \wedge Q$  = “the Broncos won Super Bowl 50 AND  
the Panthers won Super Bowl 50”

= False

AND propositions are only true if every part of the proposition is true

- $P = \text{True AND } Q = \text{True}$

# Boolean operations: AND

$P = \text{“the Broncos won Super Bowl 50”} = \text{True}$

$Q = \text{“the Broncos won Super Bowl XXXIII”} = \text{True}$

Notation:

$P \wedge Q = \text{“the Broncos won Super Bowl 50 AND  
the Broncos won Super Bowl XXXIII”}$   
 $= \text{True}$

# Boolean operations: AND

$P$  = “the Broncos won Super Bowl 50” = True

$Q$  = “the Broncos won Super Bowl XLVIII” = False

Notation:

$P \wedge Q$  = “the Broncos won Super Bowl 50 AND  
the Broncos won Super Bowl XLVIII”  
= False

# Boolean operations: OR

$P =$  “the Broncos won Super Bowl 50” = True

$Q =$  “the Panthers won Super Bowl 50” = False

Notation:

$P \vee Q =$  “the Broncos won Super Bowl 50 OR  
the Panthers won Super Bowl 50”

= True

OR propositions are true if either  
part of the proposition is true

- $P = \text{True OR } Q = \text{True}$

# Boolean operations: OR

$P$  = “the Broncos won Super Bowl XLVIII” = False

$Q$  = “the Panthers won Super Bowl 50” = False

Notation:

$P \vee Q$  = “the Broncos won Super Bowl XLVIII OR  
the Panthers won Super Bowl 50”  
= False

# Boolean operations: NOT

$P = \text{“the Broncos won Super Bowl 50”} = \text{True}$

Notation:

$\neg P = \text{NOT “the Broncos won Super Bowl 50”}$   
 $= \text{False}$

NOT propositions are the opposite of the original proposition

- If  $P = \text{True}$  then  $\neg P = \text{False}$
- If  $P = \text{False}$  then  $\neg P = \text{True}$

# Boolean operations

You should become familiar with the 3 operations:

- AND:  $P \wedge Q$
- OR:  $P \vee Q$
- NOT:  $\neg P$

Doing algebra with boolean variables is called boolean **logic**

You can combine the operations to create even longer propositions

- $(P \wedge R) \vee (Q \wedge S)$
- $\neg(P \vee Q \vee R)$

# Boolean operations

You should become familiar with the 3 operations:

- AND:  $P \wedge Q$
- OR:  $P \vee Q$
- NOT:  $\neg P$

Do these operations look familiar?



# Boolean operations

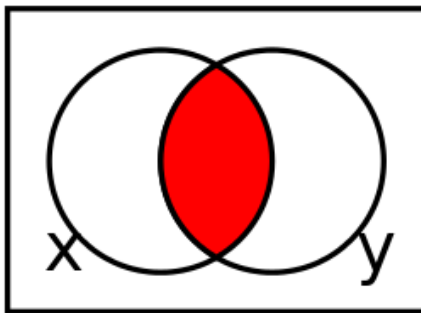
- AND:  $P \wedge Q$
- OR:  $P \vee Q$
- NOT:  $\neg P$

**Similar to set operations:**

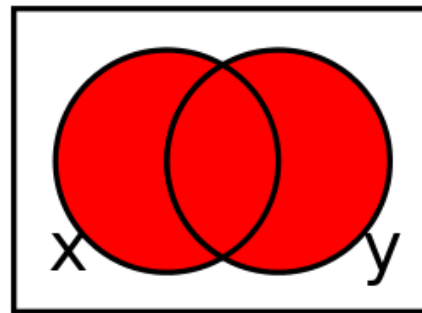
Intersection:  $\mathbf{A \cap B}$

Union:  $\mathbf{A \cup B}$

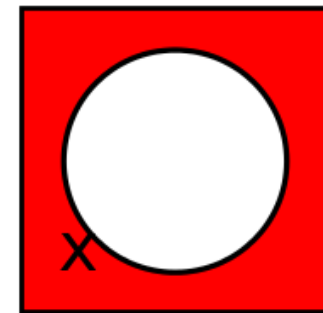
Complement:  $\mathbf{A'}$



$x \wedge y$



$x \vee y$



$\neg x$

# Describing set membership

$P$  = the number 1 is in the set of evens, **E**  
= False

$Q$  = the number 1 is in the set of odds, **O**  
= True

Is the number 1 in **E**  $\cup$  **O**?

- $P \vee Q = \text{True}$

Is the number 1 in **E**  $\cap$  **O**?

- $P \wedge Q = \text{False}$

# Describing set membership

$P$  = the number 1 is in the set of evens, **E**  
= False

$Q$  = the number 1 is in the set of odds, **O**  
= True

Is the number 1 in **E'** ?

- $\neg P$  = True

Is the number 1 in **O'** ?

- $\neg Q$  = False

# Boolean logic in data

Boolean expressions can help you **query** for data

Suppose you have a dataset of cats, and you want to test a theory that male cats that are orange and heavier than 13 lbs are friendlier than all other cats

Want to compare average temperament of two sets:

- Cats that are male, orange, and weight  $\geq 13$  lbs
- All other cats (the **complement** of the first set)

# Boolean logic in data

Want to compare average temperament of two sets:

- Cats that are male, orange, and weight  $\geq 13$  lbs
- All other cats (the **complement** of the first set)

The way you typically select subsets of data in software is using boolean expressions

Set 1: cat is male AND  
cat is orange AND  
cat's weight is  $\geq 13$  lbs

# Boolean logic in data

Want to compare average temperament of two sets:

- Cats that are male, orange, and weight  $\geq 13$  lbs
- All other cats (the **complement** of the first set)

The way you typically select subsets of data in software is using boolean expressions

Set 2: NOT (cat is male AND  
cat is orange AND  
cat's weight is  $\geq 13$  lbs)

# Boolean logic in data

Important to understand boolean logic so that you can construct the correct query for what you want

For example, understand that these statements all have the same meaning:

- NOT(cat is male AND cat is orange AND weight  $\geq$  13)
- cat is not male OR cat is not orange OR weight is not  $\geq$  13
- cat is not male OR cat is not orange OR weight is  $<$  13

# Some rules to know

$$\neg(P \vee Q) = \neg P \wedge \neg Q$$

$$\neg(P \wedge Q) = \neg P \vee \neg Q$$

$$P \vee Q = \neg(\neg P \wedge \neg Q)$$

$$P \wedge Q = \neg(\neg P \vee \neg Q)$$

$$P = P \vee (P \wedge Q)$$