CSE 191: Discrete Structures
Introduction to Propositional Logic

Nasrin Akhter

Some of the slides are either directly used or adapted from Prof. Andrew Hughes's CSE 191 presentations and are also based on the textbook "Discrete Mathematics and it's applications" (7th edition) by Kenneth H. Rosen.

The Foundations: Logic and Proofs

- Logic is the basis of all correct mathematical arguments (i.e., proofs).
- Important in all of CS and CEN:
  - Problem solving
  - Software engineering (requirements specification, verification)
  - Databases (relational algebras, SQL)
  - Computer architecture (logic gates, verification)
  - AI (automated theorem proving, rule-based ML)
  - Computer security (threat modeling)
  - …

Outline

- Propositional logic
  - **Propositions [are declarations]**
  - Logical operators
  - Truth tables

Proposition

Definition: A **proposition** is a declarative statement.

- Must be either TRUE (T) or FALSE (F).
  - Cannot be both TRUE and FALSE.
- An opinion of a specific person is a proposition.
  - Their opinion would determine the truth value.
- The bits 0/1 are used for F/T, respectively.
  - Digital logic uses 0/1 or LOW/HIGH or OFF/ON.
  - Computers use bits and logic gates for **all** computation.

Propositional Logic

Examples of Propositions	
Washington, D.C., is the capital of the USA	TRUE Proposition
$2 \times 2 = 3$	FALSE Proposition
Snow is blue	FALSE Proposition
$1 + 1 = 2$	TRUE Proposition

Propositional Logic

Examples of Non-Propositions	
What time is it?	Questions are not declarations.
Please do your homework.	Requests or commands are not declarations.
2 + 3	Not a de~~al~~ation
x + 1 = 2	Neither true nor false; truth value depends on x.
Wow!	Neither true nor false

Propositions vs Non-Propositions

Propositions	Non-Propositions

• Declarative Statements
• Either true or false, but not both
  • Has truth values.
  • Can be either a true proposition or a false proposition.
	• Questions
	• Commands/requests
	• Statements with unassigned variables
	• Exclamations
	• …

Propositional Variables

g : Grass is green

r: T

r

- Propositional variables are variables that represent propositions.
- Commonly used letter for propositional variables are **p**, **q**, **r**, **s**, …
  - Or the first letter of what we mean to represent
- Truth value of a proposition
  - **T** (true) for true propositions
  - **F** (false) for false propositions
- A propositional variable may be associated with a specific proposition or left as a placeholder for an arbitrary proposition.
- Compound propositions are formed by using propositional variables and logical operators.
  - Each compound proposition is a new proposition itself.

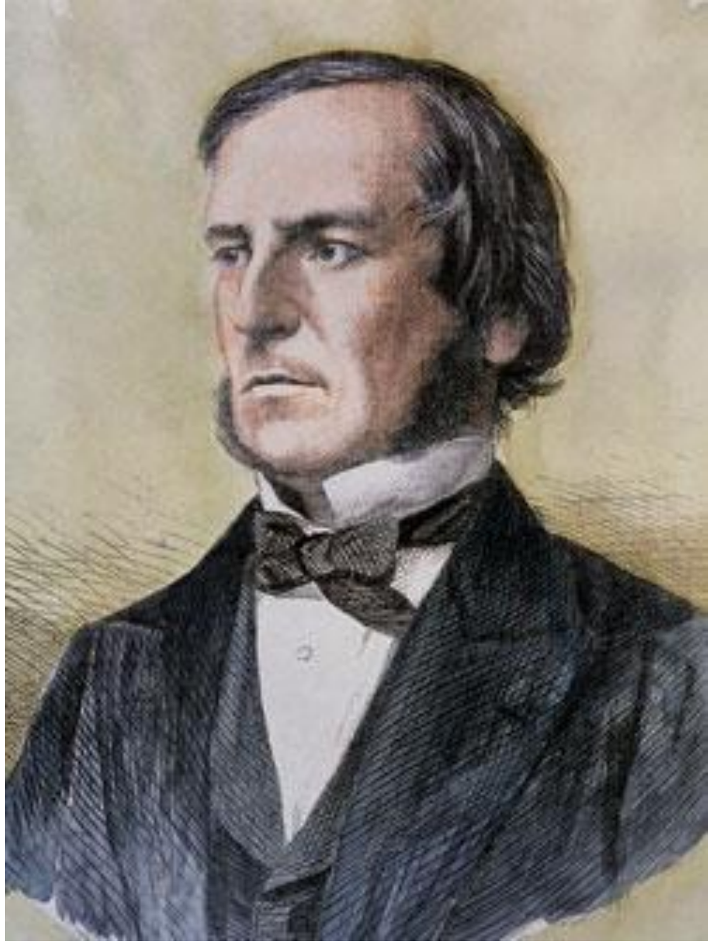
Propositional Variables

- Examples
  - ***p***: Chicago is the capital of the USA
  - ***q***: Albany is the capital of NYS
- "*p*: " has the meaning "let *p* be the statement '……' "
- Now, we can substitute *p* with the statement "Chicago is the capital of the US" and its English equivalents..

- Then we can ask –
  - Is *p* true?
  - Is *q* true?
- What about the following sentences?
  - *p* and *q*
  - *q* or *q*

Outline

- Propositional logic
  - Propositions
  - **Logical operators [form new propositions]**
  - Truth tables



George Boole

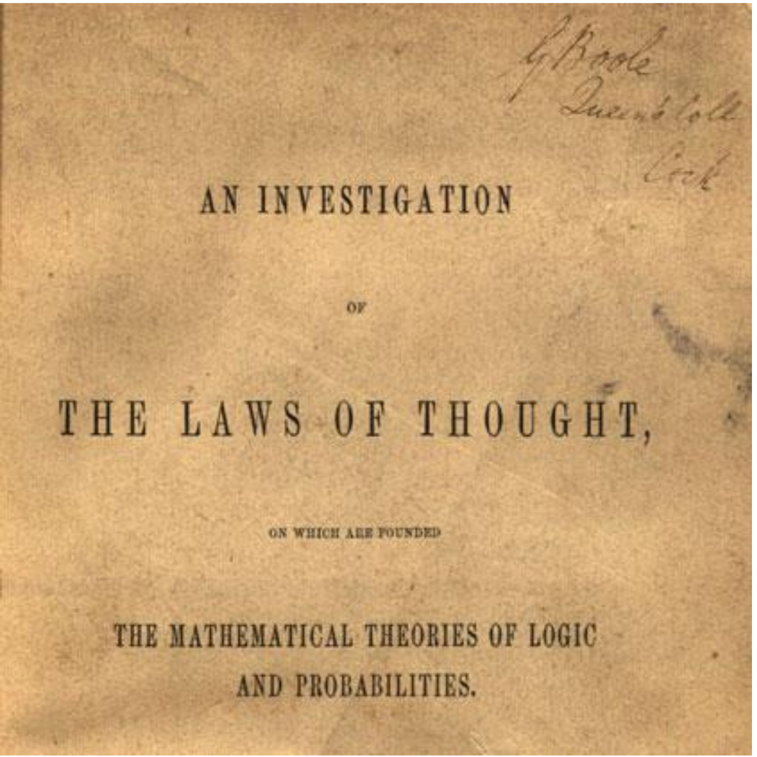


Image Sources:
Wikipedia, "George Boole", https://en.wikipedia.org/wiki/George_Boole,
George Boole 200, "Publication of the Laws of Thought", https://georgeboole.com/boole/life/ucc/lawsofthought/

Logical Operators

P and Q

- Logical operators allow combining propositions.
  - *Going forward*: combine propositions to form new propositions.
  - *Going backwards*: decompose proposition into **atomics.**

- The combined/compound proposition
  - If it is snowing, then I am not driving.
- The atomics
  - It is snowing [proposition]
  - I am not driving [proposition]
- The logical operator
  - If …, then …

Outline

- Propositional logic
  - Propositions
  - Logical operators
    - **Unary operator [operates on only one proposition]**
    - Binary logical operators
  - Truth tables

Negation Operator

Definition:

$\tilde{p}$

Let p be a proposition. The *negation of p*, denoted by $\neg p$ (also denoted by $\bar{p}$), is the statement

"It is not the case that p."

- The proposition $\neg p$ is read as "not p."
- The truth value of $\neg p$ is the opposite of the truth value of p.

<u>Example:</u>

- *"p: CSE116 is a prerequisite for CSE 191"*.
- The negation of p:
  - *"$\neg p$: It is not the case that CSE 116 is a prerequisite for CSE 191"*
- More simply, *"$\neg p$: CSE 116 is not a prerequisite for CSE 191"*

Negation Operator

$s: \neg p$

- $\neg p$ is a **new proposition** generated from p.
- We have generated one proposition from another proposition.
- We call ¬ the **negation operator.**

Outline

- Propositional logic
  - Propositions
  - Logical operators
    - Unary operator
    - **Binary logical operators [operate on two propositions]**
  - Truth tables

Binary Logical Operators

What other connectives do we have in English?

- *. . .* **and** *. . .*
- *. . .* **or** *. . .*
- **If** *. . .***, then** *. . .*
- *. . .* **if and only if** *. . .*

Unary Vs. Binary Logical Operators

- Unary operators transform **one proposition** into another.
- Binary operators combine **two propositions** into one **compound proposition.**

Binary Logical Operators: Conjunction Example

Definition:

Let p and q be propositions. The **conjunction** of p and q, denoted by $p \wedge q$, is the statement:

"p and q"

$S: p \wedge q$

and is only TRUE when p and q are both TRUE, and is FALSE otherwise.

Binary Logical Operators: Conjunction Example

- ***r***: It is rainy **and** windy. [conjunction of two propositions]
  - p: it is rainy.
  - q: it is windy.
  - $p \land q$: it is rainy and it is windy.
    - Simplified: $p \land q$: it is rainy and windy.
  - r and "$p \land q$" are interchangeable.

Note: when converting to English from symbols, try to use natural wordings.

$p \land q$ $T \land F$

- What if it turns out that today it is rainy indeed, but not windy?
  - p is TRUE, q is FALSE, so "$p \land q$" is "$T \land F$"
  - $T \land F$ is FALSE ➔ r is FALSE

Binary Logical Operators: Disjunction Operator

Definition:

Let *p* and *q* be propositions. The **disjunction** of *p* and *q*, denoted by $p \vee q$, is the statement:

"p or q."

and is TRUE whenever *p is* TRUE, *q is* TRUE, *or* both are TRUE.

The **disjunction** $p \vee q$ is false when both *p* and *q* are FALSE.

Binary Logical Operators: Disjunction Operator

- ***r***: students who took CSE115 **or** EAS 230 can take this class. [disjunction of two propositions]
  - *p*: students who took CSE115 can take this class.
  - *q*: students who took EAS230 can take this class.
  - $p \lor q$: students who took CSE115 can take this class or students who took EAS230 can take this class.
    - Simplified: $p \lor q$: students who took CSE115 or EAS230 can take this class.
    - *r* and "$p \lor q$" are interchangeable

$p \lor q \quad T \lor T$

- What if a student took both CSE115 and EAS230?
  - *p* is TRUE, *q* is TRUE, so "$p \lor q$" is "T V T"
  - T V T is TRUE ➔ *r* is TRUE

Note: known as **inclusive or.**

Binary Logical Operators: "Exclusive or" Operator

<u>Definition:</u>

Let p and q be propositions. The **exclusive or** of p and q, denoted by $p \oplus q$ (read: *p XOR q*), is the statement:

"*p or q,* but not both"

and is TRUE when exactly one of p and q is TRUE, but not both, and is FALSE otherwise.

Binary Logical Operators: “Exclusive or” Operator

- ***r***: I will go to park or I will go to movie.
  - *p*: I will go to park.
  - *q*: I will go to movie.
  - $p \oplus q$: I will go to park or I will go to movie.
    - *r* and “$p \oplus q$” are interchangeable

But not both!

Binary Logical Operators: Implication Operator

<u>Definition:</u>

Conditional

Let p and q be propositions. The **implication of** p **on** q, denoted by $p \to q$ (read: p *implies* q), is the statement:

drive to work → buy lunch

"p implies q"

or,

"if p, then q"

$p \to q$

$T \to T$ T

and is FALSE when p is TRUE and q is FALSE, and TRUE otherwise.

$\boxed{T \to F}$ F

$F \to F$ T

$F \to T$ T

- p is called the *hypothesis* or *antecedent* or *premise.*
- q is called the *conclusion* or *consequence.*

Binary Logical Operators: Implication Example

p → q

- If the night sky is clear, then stars are visible. [implication between two propositions] [Assuming there's no air or light pollution!]
  - *p*: The night sky is clear.
  - *q*: stars are visible.
  - $p \rightarrow q$: If the night sky is clear, stars are visible. [a conditional statement]

T → F q F

- What if the night sky is clear but the stars are not visible?
  - *p* is TRUE, *q* is FALSE, so "$p \longrightarrow q$" is "T $\longrightarrow$ F"
  - T $\longrightarrow$ F is FALSE

Terminology for Implication

- Implication statements are expressed in many ways.
- Common expressions of $p \to q$.

"if p, then q"	"q whenever p"
"if p, q"	"p is sufficient for q"
"q if p"	"q is necessary for p"
"q when p"	"a necessary condition for p is q"
"q unless ¬p"	"a sufficient condition for q is p"
"p implies q"	"q follows from p"
"p only if q"	

p: T
Stars are visible
whenever
sky is clear, q: F

$p \to q$
$T \to F \} F$

$q \to p$
$F \to T \} T$

Converse, Contrapositive, and Inverse

$p \to q$

- Converse of $p \to q$: $q \to p$ swap
- Contrapositive of $p \to q$: $\neg q \to \neg p$ swap + negate
- Inverse of $p \to q$: $\neg p \to \neg q$ negate

<u>Example:</u>

$r: p \to q$: If it is $\overset{p}{\text{sunny}}$, $\overset{\to}{\text{then}}$ we will $\overset{q}{\text{go}}$ to beach.

- p: it is sunny, q: we will go to beach

- Contrapositive: if we do not go to beach, then it is not sunny.
- Converse: if we go to beach, then it is sunny.
- Inverse: if it is not sunny, then we will not go to beach.

- Which of the above three is equivalent to r? [Equivalent propositions have the same truth value].

Binary Logical Operators: Bidirectional Implication Operator

<u>Definition:</u>

$p \to q$ AND $q \to p$

Let p and q be propositions. The **bidirectional implication between** p **on** q, denoted by $p \Leftrightarrow q$, is the statement:

"p if and only if q"

$T \leftrightarrow T$ } T

$F \leftrightarrow F$ } T

and is only TRUE when p and q have the same truth value, and is FALSE otherwise.

$T \leftrightarrow F$ } F

$F \leftrightarrow T$ } F

Binary Logical Operators: Bidirectional Implication Example

$p \quad \Leftrightarrow \quad q$

r • You can take the flight if and only if you buy a ticket.
  - p: you can take the flight.
  - q: you buy a ticket.
  - $p \Leftrightarrow q$: you can take the flight if and only if you buy a ticket.

$T \Leftrightarrow F \quad F$

- You could take the flight but you didn't buy a ticket!!?
  - p is TRUE, q is FALSE, so "$p \Leftrightarrow q$" is "T ⇔ F".
  - T ⇔ F is FALSE.

Terminology for Bidirectional Implication

- Common expressions of $p \Leftrightarrow q$.
  - "p is necessary and sufficient for q".
  - " if p then q, and conversely".
  - "p iff q".

Outline

- Propositional logic
  - Propositions
  - Logical operators
    - Unary operator
    - Binary logical operators [operate on two propositions]
  - Truth tables
    - Truth table construction process
    - Compound propositions
    - Order of operations

Truth Tables

- How can we formally specify the behavior or an operator?
  - How to clearly show the results of applying an operator (e.g., the negation operator) on one or more propositions?

- Build truth tables
  - List all possible combinations of truth values of the operands.
  - List the resulting truth value in the rightmost column.

Truth Tables: Negation Operator

- The truth value of $\neg p$ is the opposite of the truth value of p.

<u>Two cases</u>

- Original proposition p is FALSE
  - New proposition $\neg p$ is a TRUE proposition
- Original proposition p is TRUE
  - New proposition $\neg p$ is a FALSE proposition

p	$\neg p$
F	T
T	F

Example: Truth table for negation

Truth Tables: Binary Logical Operators

p	q	$p \wedge q$
F	F	F
F	T	F
T	F	F
T	T	T

Conjunction/AND

Truth Tables: Binary Logical Operators

p	q	$p \vee q$
F	F	F
F	T	T
T	F	T
T	T	T

Disjunction/OR

p	q	$p \oplus q$
F	F	F
F	T	T
T	F	T
T	T	F

Exclusive or/XOR

Truth Tables: Binary Logical Operators

p	q	$p \to q$
F	F	T
F	T	T
T	F	F
T	T	T

Implication/ if …, then …

p	q	$p \Leftrightarrow q$
F	F	T
F	T	F
T	F	F
T	T	T

Bidirectional Implication/IFF

Outline

- Propositional logic
  - Propositions
  - Logical operators
    - Unary operator
    - Binary logical operators [operate on two propositions]
  - Truth tables
    - Truth table construction process
    - Compound propositions
    - Order of operations

How do we construct a truth table?

$2^2 = 4$

- Need 2^n rows, where n is the number of propositional variables.
  - For $p \lor q$ we have 2 variables, so we need $2^2 = 4$ rows.
- Fill half of the first column with ***F*** values, remainder with ***T***.

p	q	$p \lor q$
F		
F		
T		
T		

We need a row for each possible combination of truth values

How do we construct a truth table?

- Need 2^n rows, where n is the number of propositional variables.
  - For $p \vee q$ we have 2 variables, so we need $2^2 = 4$ rows.
- Fill half of the first column with ***F*** values, remainder with ***T***.
- In the second column:
  - For each group of rows in first column: fill half with ***F*** and half with ***T***.

p	q	$p \vee q$
F	F	F
F	T	T
T	F	T
T	T	T

We need a row for each possible combination of truth values

F F
F T
T F
T T

How do we construct a truth table?

- Need 2^n rows, where n is the number of propositional variables.
  - For $p \vee q$ we have 2 variables, so we need $2^2 = 4$ rows.
- Fill half of the first column with ***F*** values, remainder with ***T***.
- In the second column:
  - For each group of rows in first column: fill half with ***F*** and half with ***T***.
- Determine truth value of new proposition in the last column.

p	q	$p \vee q$
F	F	F
F	T	T
T	F	T
T	T	T

We need a row for each possible combination of truth values

How do we construct a truth table?

- What to do if we have more than two variables?

$$p \lor q \lor r$$

- Here we have 3 variables, so we should end up with $2^3 = 8$ rows.
- (optionally) Add additional columns to handle partial propositions.
  - We evaluate it as the proposition $(p \lor q) \lor r$ and handle $(p \lor q)$ first.

3 variables $\rightarrow 2^3 = 8$ rows

p	q	r	$p \lor q$	$p \lor q \lor r$
F				
F				
F				
F				
T				
T				
T				
T				

How do we construct a truth table?

- What to do if we have more than two variables?

$$p \lor q \lor r$$

- Here we have 3 variables, so we should end up with $2^3 = 8$ rows.
- (optionally) Add additional columns to handle partial propositions.
  - We evaluate it as the proposition $(p \lor q) \lor r$ and handle $(p \lor q)$ first.

p	q	r	$p \lor q$	$p \lor q \lor r$
F	F			
F	F			
F	T			
F	T			
T	F			
T	F			
T	T			
T	T			

How do we construct a truth table?

- What to do if we have more than two variables?

$$p \lor q \lor r$$

- Here we have 3 variables, so we should end up with $2^3 = 8$ rows.
- (optionally) Add additional columns to handle partial propositions.
  - We evaluate it as the proposition $(p \lor q) \lor r$ and handle $(p \lor q)$ first.

p	q	r	$p \lor q$	$p \lor q \lor r$
F	F	F		
F	F	T		
F	T	F		
F	T	T		
T	F	F		
T	F	T		
T	T	F		
T	T	T		

How do we construct a truth table?

- What to do if we have more than two variables?

$$p \vee q \vee r$$

- Here we have 3 variables, so we should end up with $2^3 = 8$ rows.
- (optionally) Add additional columns to handle partial propositions.
  - We evaluate it as the proposition $(p \vee q) \vee r$ and handle $(p \vee q)$ first.

p	q	r	$p \vee q$	$p \vee q \vee r$
F	F	F	F	
F	F	T	F	
F	T	F	T	
F	T	T	T	
T	F	F	T	
T	F	T	T	
T	T	F	T	
T	T	T	T	

How do we construct a truth table?

- What to do if we have more than two variables?

$$p \vee q \vee r$$

- Here we have 3 variables, so we should end up with $2^3 = 8$ rows.
- (optionally) Add additional columns to handle partial propositions.
  - We evaluate it as the proposition $(p \vee q) \vee r$ and handle $(p \vee q)$ first.

$\vee$

p	q	r	$p \vee q$	$p \vee q \vee r$
F	F	F	F	F F
F	F	T	F	T T
F	T	F	T	T
F	T	T	T	T
T	F	F	T	T
T	F	T	T	T
T	T	F	T	T
T	T	T	T	T

Outline

- Propositional logic
  - Propositions
  - Logical operators
    - Unary operator
    - Binary logical operators [operate on two propositions]
  - Truth tables
    - Truth table construction process
    - Compound propositions
    - Order of operations

Compound Propositions

A **compound proposition** is created by using one or more logical operators.

- Suppose ***p*** and ***q*** are propositions.
  - Compound proposition: $(p \vee q) \wedge \neg(p \wedge q)$
  - This is a new proposition formed by using a combination of AND, OR, and NOT.
- Suppose we have propositions ***s*** and ***t*** where:
  - ***s*** : Jim eats pie
  - ***t*** : Jim eats cake
- What is $u = (s \vee t) \wedge \neg(s \wedge t)$ in natural language (e.g., English)?
  - ***u*** : Jim eats pie or cake **but** Jim doesn't eat pie and cake.

AND

How to build construct compound propositions?

- Consider how we build u from s and t: $\neg(s \wedge t)$

  $s \quad \vee \quad t \quad \wedge$

  u : Jim eats pie or cake but Jim doesn't eat pie and cake.

- Identify atomic propositions:

  s : Jim eats pie. $\qquad$ t : Jim eats cake

- Then we have:

  $(s \vee t)$: Jim eats pie or cake

  $(s \wedge t)$: Jim eats pie and cake

- The last piece is

  $\neg(s \wedge t)$: Jim doesn't eat pie and cake

- Finally, we get:

  $(s \vee t) \wedge \neg(s \wedge t)$: Jim eats pie or cake but Jim doesn't eat pie and cake

Examples

- p: The window is closed; q: It is raining; r: I will run the air conditioner.
- ¬p : ? The window is not closed.
- p V ¬q: ? The window is closed or it is not raining

  $q \wedge \neg p$
- It is raining but the window is not closed: ?

  $\neg q \rightarrow \neg p$
- If it is not raining then the window is open: ?
- p ⇔ q: ?
- (q ∧ ¬p) → ¬r

How to evaluate compound propositions?

- Recall that each logical operator creates a new proposition.
  - The outcome is a new proposition.
  - New statement must be TRUE or FALSE.

- Two ways to view a compound proposition.
  - Start with the smaller propositions and build up.
  - Start with the compound proposition and decompose.

How to evaluate compound propositions?

- Is "*u* : Jim eats pie or cake but Jim doesn't eat pie and cake." true?
- $(s \vee t) \wedge \neg(s \wedge t)$

  $s : T \quad t : T$

- Scenario 1: Jim eats cake and Jim eats pie are both true.
- We can evaluate *u* from the ground up (as a function of *s* and *t*).

$(T \vee T) \wedge \neg(T \wedge T)$

$T \wedge \neg(T)$

$T \wedge F \equiv F$

F

We have that s : T, t : T

So $(s \vee t)$ and $(s \wedge t)$ are both T

$\neg(s \wedge t)$ is F

Finally, we can say that $(s \vee t) \wedge \neg(s \wedge t)$ is F.

$(T \wedge F)$

$\neg(s \wedge t)$ is F, so the AND evaluates to F.

Therefore, *u* is FALSE.

How to evaluate compound propositions?

- What if we don't know anything about the atomics that form u?
  - Consider the general case, a.k.a. build the truth table.

$2^2 = 4$

s	t	(s ∨ t)	(s ∧ t)	¬(s ∧ t)	(s ∨ t) ∧ ¬(s ∧ t)

How to evaluate compound propositions?

- What if we don't know anything about the atomics that form u?
  - Consider the general case, a.k.a. build the truth table.

s	t	$(s \vee t)$	$(s \wedge t)$	$\neg(s \wedge t)$	$(s \vee t) \wedge \neg(s \wedge t)$
F					
F					
T					
T					

How to evaluate compound propositions?

- What if we don't know anything about the atomics that form u?
  - Consider the general case, a.k.a. build the truth table.

s	***t***	***(s ∨ t)***	***(s ∧ t)***	**¬(*s* ∧ *t*)**	**(*s* ∨ *t*) ∧ ¬(*s* ∧ *t*)**
F	F				
F	T				
T	F				
T	T				

How to evaluate compound propositions?

- What if we don’t know anything about the atomics that form u?
    - Consider the general case, a.k.a. build the truth table.

s	t	$(s \vee t)$	$(s \wedge t)$	$\neg(s \wedge t)$	$(s \vee t) \wedge \neg(s \wedge t)$
F	F	F			
F	T	T			
T	F	T			
T	T	T			

How to evaluate compound propositions?

- What if we don’t know anything about the atomics that form u?
  - Consider the general case, a.k.a. build the truth table.

s	***t***	***(s ∨ t)***	***(s ∧ t)***	**¬(*s* ∧ *t*)**	**(*s* ∨ *t*) ∧ ¬(*s* ∧ *t*)**
F	F	F	F		
F	T	T	F		
T	F	T	F		
T	T	T	T		

How to evaluate compound propositions?

- What if we don’t know anything about the atomics that form u?
    - Consider the general case, a.k.a. build the truth table.

s	***t***	***(s ∨ t)***	***(s ∧ t)***	***¬(s ∧ t)***	**(*s ∨ t*) ∧ ¬(*s ∧ t*)**
F	F	F	F	T	
F	T	T	F	T	
T	F	T	F	T	
T	T	T	T	F	

How to evaluate compound propositions?

- What if we don't know anything about the atomics that form u?
  - Consider the general case, a.k.a. build the truth table.

s	t	$(s \vee t)$	$(s \wedge t)$	$\neg(s \wedge t)$	$(s \vee t) \wedge \neg(s \wedge t)$
F	F	F	F	T	F
F	T	T	F	T	T
T	F	T	F	T	T
T	T	T	T	F	F

How to evaluate compound propositions?

- What if we don't know anything about the atomics that form u?
    - Consider the general case, a.k.a. build the truth table.

s	t	$(s \vee t)$	$(s \wedge t)$	$\neg(s \wedge t)$	$(s \vee t) \wedge \neg(s \wedge t)$
F	F	F	F	T	F
F	T	T	F	T	T
T	F	T	F	T	T
T	T	T	T	F	F

- <u>Scenario 2</u>: *u* is TRUE and *s* (Jim eats pie) is TRUE.
    - What is the truth value of *t*?

Outline

- Propositional logic
  - Propositions
  - Logical operators
    - Unary operator
    - Binary logical operators [operate on two propositions]
  - Truth tables
    - Truth table construction process
    - Compound propositions
    - Order of operations

Precedence or Operators

p : T
q : F
r : T

Operator	Precedence
()	0
$\neg$	1
$\wedge$	2
$\vee$	3
$\oplus$	4
$\rightarrow$	5
$\Leftrightarrow$	6

<u>Examples</u>

$((\neg \mathbf{p}) \wedge \mathbf{q})$ means $(\neg \mathbf{p}) \wedge \mathbf{q}$

$((\mathbf{p} \wedge \mathbf{q}) \rightarrow \mathbf{r}) \rightarrow$ means $(\mathbf{p} \wedge \mathbf{q}) \rightarrow \mathbf{r}$

$\mathbf{p} \vee (\mathbf{q} \wedge \mathbf{r}) \leftrightarrow \mathbf{p} \rightarrow \mathbf{q} \oplus \mathbf{r}$ means

$T \vee (F \wedge T) \longleftrightarrow T \rightarrow F \oplus T$

$(T \vee F) \longleftrightarrow T \rightarrow F \oplus T$

$T \longleftrightarrow T \rightarrow (F \oplus T)$

$T \leftrightarrow (T \rightarrow T)$

$T \leftrightarrow T$

T

Precedence or Operators

Operator	Precedence
()	0
$\neg$	1
$\wedge$	2
$\vee$	3
$\oplus$	4
$\rightarrow$	5
$\Leftrightarrow$	6

p: T q: T r: F

<u>Examples</u>

$\neg \mathbf{p} \wedge \mathbf{q}$ means $(\neg \mathbf{p}) \wedge \mathbf{q}$

$\mathbf{p} \wedge \mathbf{q} \rightarrow \mathbf{r} \rightarrow$ means $(\mathbf{p} \wedge \mathbf{q}) \rightarrow \mathbf{r}$

$\mathbf{p} \vee \mathbf{q} \wedge \mathbf{r} \leftrightarrow \mathbf{p} \rightarrow \mathbf{q} \oplus \mathbf{r}$ means

$(\mathbf{p} \vee (\mathbf{q} \wedge \mathbf{r})) \leftrightarrow (\mathbf{p} \rightarrow (\mathbf{q} \oplus \mathbf{r}))$

When in doubt, use parenthesis.