Logical operators full truth table

Input		Output				
		Conjunction	Exclusive or	Disjunction	Conditional	Biconditional
p	q	$p \wedge q$	$p \oplus q$	$p \vee q$	$p \to q$	$p \leftrightarrow q$
T	T	T	F	T	T	T
T	F	F	T	T	F	F
F	T	F	T	T	T	F
F	F	F	F	F	T	T
		"p and q"	"p xor q"	"p or q"	"if p then q"	"p if and only if q"

Hypothesis conclusion

The only way to make the conditional statement $p \to q$ false is to _____

The **hypothesis** of $p \to q$ is _____ The **antecedent** of $p \to q$ is _____

The **conclusion** of $p \to q$ is _____ The **consequent** of $p \to q$ is _____

Compound propositions recursive definition

We can use a recursive definition to describe all compound propositions that use propositional variables from a specified collection. Here's the definition for all compound propositions whose propositional variables are in $\{p, q\}$.

Basis Step:	p and q are each a compound proposition
Recursive Step:	If x is a compound proposition then so is $(\neg x)$ and if x and y are both compound propositions then so is each of $(x \wedge y), (x \oplus y), (x \vee y), (x \to y), (x \leftrightarrow y)$

Compound propositions precedence

Order of operations (Precedence) for logical operators:

Negation, then conjunction / disjunction, then conditional / biconditionals.

Example: $\neg p \lor \neg q$ means $(\neg p) \lor (\neg q)$.

Consistency example

Consistency:

> Whenever the system software is being upgraded, users cannot access the file system. If users can access the file system, then they can save new files. If users cannot save new files, then the system software is not being upgraded.

1. Translate to symbolic compound propositions

2. Look for some truth assignment to the propositional variables for which all the compound propositions output T

Logical operators

Logical operators aka propositional connectives

Conjunction	AND	$\land$	`\land`	2 inputs	Evaluates to T exactly when **both** inputs are T
Exclusive or	XOR	$\oplus$	`\oplus`	2 inputs	Evaluates to T exactly when **exactly one** of inputs is T
Disjunction	OR	$\lor$	`\lor`	2 inputs	Evaluates to T exactly when **at least one** of inputs is T
Negation	NOT	$\neg$	`\lnot`	1 input	Evaluates to T exactly when its input is F

Logical operators truth tables

Truth tables: Input-output tables where we use T for 1 and F for 0.

Input		Output		
		Conjunction	**Exclusive or**	**Disjunction**
p	q	$p \wedge q$	$p \oplus q$	$p \vee q$
T	T	T	F	T
T	F	F	T	T
F	T	F	T	T
F	F	F	F	F
				

Input	Output
	Negation
p	$\neg p$
T	F
F	T
	

Logical operators example truth table

Input			Output	
p	q	r	$(p \wedge q) \oplus (\ (p \oplus q) \wedge r\ )$	$(p \wedge q) \vee (\ (p \oplus q) \wedge r\ )$
T	T	T		
T	T	F		
T	F	T		
T	F	F		
F	T	T		
F	T	F		
F	F	T		
F	F	F		

Truth table to compound proposition

Given a truth table, how do we find an expression using the input variables and logical operators that has the output values specified in this table?

Application: design a circuit given a desired input-output relationship.

Input		Output	
p	q	$mystery_1$	$mystery_2$
T	T	T	F
T	F	T	F
F	T	F	F
F	F	T	T

Expressions that have output $mystery_1$ are

Expressions that have output $mystery_2$ are

Idea: To develop an algorithm for translating truth tables to expressions, define a convenient **normal form** for expressions.

Compound proposition definitions

Proposition: Declarative sentence that is true or false (not both).

Propositional variable: Variable that represents a proposition.

Compound proposition: New proposition formed from existing propositions (potentially) using logical operators. *Note*: A propositional variable is one example of a compound proposition.

Truth table: Table with one row for each of the possible combinations of truth values of the input and an additional column that shows the truth value of the result of the operation corresponding to a particular row.