American University of Beirut
Dept. of Computer Science

CMPS/Math 211

Discrete Structures

Lecture 2

Section 1.1: Propositional Logic

Foundations of Logic: Overview

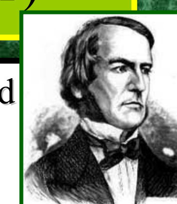
- Propositional logic (§1.1-1.2):
  - Basic definitions. (§1.1)
  - Equivalence rules & derivations. (§1.2)
- Predicate logic (§1.3)
  - Predicates.

Propositional Logic (§1.1)

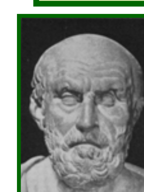
Propositional Logic is the logic of compound statements built from simpler statements using so-called *Boolean connectives*.

Some applications in computer science:

- Design of digital electronic circuits.
- Expressing conditions in programs.
- Queries to databases & search engines.



George Boole (1815-1864)



Chrysippus of Soli

Definition of a *Proposition*

Definition: A *proposition* (denoted p, q, r, …) is simply:

- a *statement* (*i.e.*, a declarative sentence)
  – *with some definite meaning*, (not vague or ambiguous)
- having a *truth value* that's either *true* (**T**) or *false* (**F**)
  – it is **never** both, neither, or somewhere "in between!"
- Later, we will study *probability theory,* in which we assign *degrees of certainty* ("between" **T** and **F**) to propositions.
  – But for now: think True/False only!

Examples of Propositions

- "It is raining." (In a given situation.)
- "Beijing is the capital of China." • "1 + 2 = 3"

<u>But, the following are **NOT** propositions:</u>

- "Who's there?" (interrogative, question)
- "Just do it!" (imperative, command)
- "1 + 2" (expression with a non-true/false value)
- " x = y + 2 "

Operators / Connectives

An *operator* or *connective* combines one or more *operand* expressions into a larger expression. (*E.g.*, "+" in numeric exprs.)

- *Unary* operators take 1 operand (*e.g.,* −3); *binary* operators take 2 operands (*eg* 3 × 4).
- *Propositional* or *Boolean* operators operate on propositions (or their truth values) instead of on numbers.

Some Popular Boolean Operators

Formal Name	**Nickname**	**Arity**	**Symbol**
Negation operator	NOT	Unary	$\neg$
Conjunction operator	AND	Binary	$\wedge$
Disjunction operator	OR	Binary	$\vee$
Exclusive-OR operator	XOR	Binary	$\oplus$
Implication (conditional) operator	IMPLIES	Binary	$\rightarrow$
Biconditional operator	IFF	Binary	$\leftrightarrow$

The Negation Operator

The unary *negation operator* "¬" (*NOT*) transforms a prop. into its logical *negation*.

E.g. If p = "I have brown hair."

then $\neg p$ = "I do **not** have brown hair."

The *truth table* for NOT:

p	$\neg p$
T	F
F	T

Operand column

Result column

T :≡ True; F :≡ False

":≡" means "is defined as"

The Conjunction Operator

The binary *conjunction operator* "∧" (*AND*) combines two propositions to form their logical *conjunction*.

E.g. If p="I will have salad for lunch." and q="I will have steak for dinner.", then $p \wedge q$="I will have salad for lunch **and** I will have steak for dinner."

Conjunction Truth Table

- Note that a conjunction $p_1 \wedge p_2 \wedge \ldots \wedge p_n$ of n propositions will have 2^n rows in its truth table.

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

The Disjunction Operator

The binary *disjunction operator* "∨" (*OR*) combines two propositions to form their logical *disjunction*.

p="My car has a bad engine."

q="My car has a bad carburetor."

$p \vee q$="Either my car has a bad engine, **or** my car has a bad carburetor, or both."

Meaning is like "and/or" in English.

Disjunction Truth Table

- $p \vee q$ means that p is true, or q is true, **or both** are true!
- So, this operation is also called *inclusive or,* because it **includes** the possibility that both p and q are true.

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

Note difference from AND

Order of Precedence

- Use parentheses to *group sub-expressions*: "I just saw my old friend, and either he's grown or I've shrunk." $= f \wedge (g \vee s)$
  - $(f \wedge g) \vee s$ would mean something different
  - $f \wedge g \vee s$ would be ambiguous
- By convention, "¬" takes *precedence* over both "∧" and "∨".
  - $\neg s \wedge f$ means $(\neg s) \wedge f$, **not** $\neg (s \wedge f)$

A Simple Exercise

Let p="It rained last night",
q="The sprinklers came on last night,"
r="The lawn was wet this morning."

Translate each of the following into English:

$\neg p$ = "It didn't rain last night."

$r \wedge \neg p$ = "The lawn was wet this morning, and it didn't rain last night."

$\neg r \vee p \vee q$ = "Either the lawn wasn't wet this morning, or it rained last night, or the sprinklers came on last night."

The *Exclusive Or* Operator

The binary *exclusive-or operator* "⊕" (*XOR*) combines two propositions to form their logical "exclusive or"

p = "I will earn an A in this course,"

q = "I will drop this course,"

$p \oplus q$ = "I will either earn an A in this course, or I will drop it (but not both!)"

Exclusive-Or Truth Table

- Note that $p \oplus q$ means that p is true, or q is true, but **not both**!
- This operation is called *exclusive or,* because it **excludes** the possibility that both p and q are true.

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

Note difference from OR.

Natural Language is Ambiguous

Note that English "or" can be ambiguous regarding the "both" case!

"Pat is a singer or Pat is a writer." - $\vee$

"Pat is a man or Pat is a woman." - $\oplus$

Need context to understand the meaning!

For this class, assume "or" means inclusive.

The *Implication* Operator

antecedent consequent

The *implication* $p \rightarrow q$ states that p implies q.

I.e., If p is true, then q is true; but if p is not true, then q could be either true or false.

E.g., let p = "You study hard."
q = "You will get a good grade."

$p \rightarrow q$ = "If you study hard, then you will get a good grade." (else, it could go either way)

Implication Truth Table

- $p \rightarrow q$ is **false** only when p is true but q is **not** true.
- $p \rightarrow q$ does **not** say that p causes q!
- $p \rightarrow q$ does **not** require that p or q **are ever true**!
- *E.g.* "(1=0) $\rightarrow$ pigs can fly" is TRUE!

p	q	$p \rightarrow q$
F	F	T
F	T	T
T	F	**F**
T	T	T

The only False case!

English Phrases Meaning $p \rightarrow q$

- "p implies q"
- "if p, then q"
- "if p, q"
- "when p, q"
- "whenever p, q"
- "q if p"
- "q when p"
- "q whenever p"

- "p only if q"
- "p is sufficient for q"
- "q is necessary for p"
- "q follows from p"
- "q is implied by p"
- ``q unless $\neg p$''

More Examples

- ``If you get 100% on the final, then you will get an A''

Interpretation: If you manage to get a 100% on the final, then you would expect to receive an A. If you do not get a 100%, you may or may not receive an A, depending on other factors (such as…). However, if you do get 100%, but the professor does not give you an A, you will feel cheated.

Intricacies

- ``*if $p$ then $q$*'' expresses the same thing as ``*p only if q*'' (*which really says q is necessary for p*)
- To remember this, note that ``*p only if q*'' says that p cannot be true when q is not true (check the truth table of implication)

p	q	$p \rightarrow q$
F	F	T
F	T	T
T	F	**F**
T	T	T

Intricacies

- ``*if $p$ then $q$*'' expresses the same thing as ``*p only if q*''.
- In other words, if q is false, p must be false.
- Do not mistake this for p if q: this says, if q is false, p may or may not be false.

Intricacies

- ``*if p then q*'' expresses the same thing as ``*q unless ¬ q*'':

Ex: If Maria learns discrete Mathematics, then she will find a good job.

Maria will find a good job unless she does not learn discrete mathematics.

Converse, Inverse, Contrapositive

Some terminology, for an implication $p \rightarrow q$:

- Its *converse* is: $q \rightarrow p$.
- Its *inverse* is: $\neg p \rightarrow \neg q$.
- Its *contrapositive*: $\neg q \rightarrow \neg p$.
- One of these three has the *same meaning* (same truth table) as $p \rightarrow q$. Can you figure out which?

How do we know for sure?

Proving the equivalence of $p \rightarrow q$ and its contrapositive using truth tables:

p	q	$\neg q$	$\neg p$	$p \rightarrow q$	$\neg q \rightarrow \neg p$
F	F	T	T	T	T
F	T	F	T	T	T
T	F	T	F	F	F
T	T	F	F	T	T

Exercise

- The converse and the inverse of a conditional statement are also equivalent.
- But neither is equivalent to the original conditional statement.
- What are the contrapositive, the converse, and the inverse of the conditional statement: ``If it is raining, then the home team wins.''?

Intricacies Re-visited

- Why ``*if p then q*'' expresses the same thing as ``*p only if q*''.
- We know that $p \to q$ is equivalent to $\neg q \to \neg p$
- Then this must express *p only if q,* because if $\neg q$, then $\neg p$, a contradiction.
- Notice we are not claiming *p if q* (because here, if $\neg q$, then we have p or $\neg p$.

The *biconditional* operator

The *biconditional* $p \leftrightarrow q$ states that p is true *if and only if (IFF)* q is true.

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."

Biconditional Truth Table

- $p \leftrightarrow q$ means that p and q have the **same** truth value.
- Note this truth table is the exact **opposite** of $\oplus$'s!
  Thus, $p \leftrightarrow q$ means $\neg(p \oplus q)$
- $p \leftrightarrow q$ does **not** imply that p and q are true, or that either of them causes the other, or that they have a common cause.

p	q	$p \leftrightarrow q$
F	F	T
F	T	F
T	F	F
T	T	T

Intricacies

$$p \to q$$

- p is sufficient but not necessary for q
- q is necessary but not sufficient for p

$$p \leftrightarrow q$$

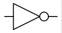
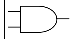
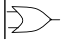
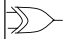
- p is necessary and sufficient for q
- q is necessary and sufficient for p

Boolean Operations Summary

- We have seen 1 unary operator (out of the 4 possible) and 5 binary operators (out of the 16 possible). Their truth tables are below.

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \oplus q$	$p \rightarrow q$	$p \leftrightarrow q$
F	F	T	F	F	F	T	T
F	T	T	F	T	T	T	F
T	F	F	F	T	T	F	F
T	T	F	T	T	F	T	T

Some Alternative Notations

Name:	not	and	or	xor	implies	iff
Propositional logic:	$\neg$	$\wedge$	$\vee$	$\oplus$	$\rightarrow$	$\leftrightarrow$
Boolean algebra:	$\bar{p}$	pq	$+$	$\oplus$		
Logic gates:						

Truth Tables of Compound Propositions

- Compound propositions involve any number of propositional variables and logical connectors.
- Construct the truth table of the compound proposition: $(p \vee \neg q) \rightarrow (p \wedge q)$

Precedence of Logical Operators

- Example, how do you interpret $\neg p \wedge q$?
- In order of most dominating:

  $\neg$

  $\wedge$

  $\vee$

  $\rightarrow$

  $\leftrightarrow$

Translating English Sentences

- ``You can access the internet from campus only if you are a computer science major or you are not a freshman''.

Translating English Sentences

- ``You can access the internet from campus only if you are a computer science major or you are not a freshman''.

Let a, c and f, represent `` You can access the internet from campus'', ``you are a computer science major'', and ``you are a freshman'', respectively.

We then have: $a \rightarrow (c \vee \neg f)$

Translating English Sentences

- ``You cannot ride the roller coaster if you are under 4 feet tall unless you are older than 16 years old.''

Translating English Sentences

- ``You cannot ride the roller coaster if you are under 4 feet tall unless you are older than 16 years old.''

q = *``You can ride the roller-coaster''*

r = *``You are under 4 feet tall''*

S = *``You are older than 16 years old''*

$(r \wedge \neg s) \rightarrow \neg q$