Propositional logic (§1.1-1.2): Review from Mat 1348

Propositional logic: Review

Mathematical Logic is a tool for working with elaborate *compound* statements. It includes:

- A formal language for expressing them.
- A concise notation for writing them.
- A methodology for objectively reasoning about their truth or falsity.
- It is the foundation for expressing formal proofs in all branches of mathematics.

Propositional logic: Review

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!"
    - However, you might not *know* the actual truth value,
    - and, the truth value might *depend* on the situation or context.

Propositional logic: Review

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

But, the following are **NOT** propositions:

- "Who's there?" (interrogative, question)
- "La la la la la." (meaningless interjection)
- "Just do it!" (imperative, command)
- "Yeah, I sorta dunno, whatever..." (vague)
- "1 + 2" (expression with a non-true/false value)

Operators / Connectives

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

Formal Name	**Nickname**	**Arity**	**Symbol**
Negation operator	NOT	Unary	¬
Conjunction operator	AND	Binary	∧
Disjunction operator	OR	Binary	∨
Exclusive-OR operator	XOR	Binary	⊕
Implication operator	IMPLIES	Binary	→
Biconditional operator	IFF	Binary	?

The Negation Operator

The unary *negation operator* "¬" (*NOT*)

p	$\neg p$
T	F
F	T

The binary *conjunction operator* "∧" (*AND*)

∧ND

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

¬ and ∧ operations together are sufficient to express *any* Boolean truth table!

The Disjunction Operator

The binary *disjunction operator* "∨" (*OR*).

Meaning is like "and/or" in English.

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

So, this operation is also called *inclusive or,* because it **includes** the possibility that both *p* and *q* are true.

"¬" and "∨" together are also universal.

Nested Propositional Expressions

- Use parentheses to *group sub-expressions*: "I just saw my old *f*riend, and either he's *g*rown or I've *s*hrunk." = $f \wedge (g \vee s)$
  - $(f \wedge g) \vee s$ 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

Exclusive-or operator “⊕” (*XOR*).

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**

p = “I will earn an A in this course,”

q = “I will drop this course,”

p ⊕ *q* = “I will either earn an A in this course, or I will drop it (but not both!)”

“¬” and “⊕” together are **not** universal.

Natural Language is Ambiguous

Note that <u>English</u> "or" can be <u>ambiguous</u> regarding the "both" case!

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

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

p	q	p "or" q
F	F	F
F	T	T
T	F	T
T	T	?

Need context to disambiguate the meaning!

For this class, assume "or" means <u>inclusive</u>.

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!

How do we know for sure?

Proving the equivalence of $p \rightarrow q$ and its contrapositive $\neg q \rightarrow \neg p$ 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

The *biconditional* operator

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

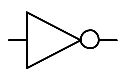
p	q	$p \leftrightarrow q$
F	F	T
F	T	F
T	F	F
T	T	T

- $p \leftrightarrow q$ 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.

Boolean Operations Summary

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

<table>
<tr><th>Name:</th><th>not</th><th>and</th><th>or</th><th>xor</th><th>implies</th><th>iff</th></tr>
<tr><td>Propositional logic:</td><td>$\neg$</td><td>$\wedge$</td><td>$\vee$</td><td>$\oplus$</td><td>$\rightarrow$</td><td>$\leftrightarrow$</td></tr>
<tr><td>Boolean algebra:</td><td>$\overline{p}$</td><td>pq</td><td>$+$</td><td>$\oplus$</td><td></td><td></td></tr>
<tr><td>C/C++/Java (wordwise):</td><td>!</td><td>&&</td><td>||</td><td>!=</td><td></td><td>==</td></tr>
<tr><td>C/C++/Java (bitwise):</td><td>~</td><td>&</td><td>|</td><td>^</td><td></td><td></td></tr>
<tr><td>Logic gates:</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Propositional Equivalence (§1.2)

Two *syntactically* (*i.e.,* textually) different compound propositions may be the *semantically* identical (*i.e.,* have the same meaning). We call them *equivalent*. Learn:

- Various *equivalence rules* or *laws*.
- How to *prove* equivalences using *symbolic derivations*.

Tautologies and Contradictions

A *tautology* is a compound proposition that is **true** *no matter what* the truth values of its atomic propositions are!

Ex. $p \vee \neg p$ [What is its truth table?]

A *contradiction* is a compound proposition that is **false** no matter what!

Ex. $p \wedge \neg p$ [Truth table?]

Other compound props. are *contingencies*.

Logical Equivalence

Compound proposition *p* is *logically equivalent* to compound proposition *q*, written $p \Leftrightarrow q$, **IFF** the compound proposition $p \leftrightarrow q$ is a tautology.

THAT IS:

IFF *p* and *q* contain the same truth values as each other in <u>all</u> rows of their truth tables.

Proving Equivalence via Truth Tables

Ex. Prove that $p \lor q \Leftrightarrow \neg(\neg p \land \neg q)$.

p	q	$p \lor q$	$\neg p$	$\neg q$	$\neg p \land \neg q$	$\neg(\neg p \land \neg q)$
F	F	F	T	T	T	F
F	T	T	T	F	F	T
T	F	T	F	T	F	T
T	T	T	F	F	F	T

Equivalence Laws

- These are similar to the arithmetic identities you may have learned in algebra, but for propositional equivalences instead.
- They provide a pattern or template that can be used to match all or part of a much more complicated proposition and to find an equivalence for it.

Equivalence Laws - Examples

- *Identity*: $p \wedge \mathbf{T} \Leftrightarrow p$ $\quad p \vee \mathbf{F} \Leftrightarrow p$
- *Domination*: $p \vee \mathbf{T} \Leftrightarrow \mathbf{T}$ $\quad p \wedge \mathbf{F} \Leftrightarrow \mathbf{F}$
- *Idempotent*: $p \vee p \Leftrightarrow p$ $\quad p \wedge p \Leftrightarrow p$
- *Double negation:* $\neg\neg p \Leftrightarrow p$
- *Commutative:* $p \vee q \Leftrightarrow q \vee p$ $\quad p \wedge q \Leftrightarrow q \wedge p$
- *Associative:* $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$

  $(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$

More Equivalence Laws

- *Distributive*:

  $$p\vee(q\wedge r) \Leftrightarrow (p\vee q)\wedge(p\vee r)$$
  $$p\wedge(q\vee r) \Leftrightarrow (p\wedge q)\vee(p\wedge r)$$

- *De Morgan's*:

  $$\neg(p\wedge q) \Leftrightarrow \neg p \vee \neg q$$
  $$\neg(p\vee q) \Leftrightarrow \neg p \wedge \neg q$$

- *Trivial tautology/contradiction*:

  $$p \vee \neg p \Leftrightarrow \mathbf{T} \qquad p \wedge \neg p \Leftrightarrow \mathbf{F}$$

Defining Operators via Equivalences

Using equivalences, we can *define* operators in terms of other operators.

- Exclusive or: $p \oplus q \Leftrightarrow (p \vee q) \wedge \neg(p \wedge q)$
  $p \oplus q \Leftrightarrow (p \wedge \neg q) \vee (q \wedge \neg p)$
- Implies: $p \rightarrow q \Leftrightarrow \neg p \vee q$
- Biconditional: $p \leftrightarrow q \Leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p)$
  $p \leftrightarrow q \Leftrightarrow \neg(p \oplus q)$

An Example Problem

- Check using a symbolic derivation whether $(p \wedge \neg q) \rightarrow (p \oplus r) \Leftrightarrow \neg p \vee q \vee \neg r$.

$(p \wedge \neg q) \rightarrow (p \oplus r)$ [Expand definition of $\rightarrow$]

$\Leftrightarrow$ Ø$(p \wedge \neg q)$ Ú $(p \oplus r)$ [Expand defn. of $\oplus$]

$\Leftrightarrow \neg(p \wedge \neg q) \vee ((p \vee r) \wedge \neg(p \wedge r))$

[DeMorgan's Law]

$\Leftrightarrow (\neg p \vee q) \vee ((p \vee r) \wedge \neg(p \wedge r))$

cont.

Example Continued…

$(\neg p \vee q) \vee ((p \vee r) \wedge \neg(p \wedge r)) \Leftrightarrow$ [$\vee$ commutes]

$\Leftrightarrow (q \vee \neg p) \vee ((p \vee r) \wedge \neg(p \wedge r))$ [$\vee$ associative]

$\Leftrightarrow q \vee (\neg p \vee ((p \vee r) \wedge \neg(p \wedge r)))$ [distrib. $\vee$ over $\wedge$]

$\Leftrightarrow q \vee (((\neg p \vee (p \vee r)) \wedge (\neg p \vee \neg(p \wedge r)))$ [assoc.]

$\Leftrightarrow q \vee (((\neg p \vee p) \vee r) \wedge (\neg p \vee \neg(p \wedge r)))$ [trivail taut.]

$\Leftrightarrow q \vee ((\mathbf{T} \vee r) \wedge (\neg p \vee \neg(p \wedge r)))$ [domination]

$\Leftrightarrow q \vee (\mathbf{T} \wedge (\neg p \vee \neg(p \wedge r)))$ [identity]

$\Leftrightarrow q \vee (\neg p \vee \neg(p \wedge r)) \Leftrightarrow$ *cont.*

End of Long Example

$q \vee (\neg p \vee \neg(p \wedge r))$

[DeMorgan's] $\Leftrightarrow q \vee (\neg p \vee (\neg p \vee \neg r))$

[Assoc.] $\Leftrightarrow q \vee ((\neg p \vee \neg p) \vee \neg r)$

[Idempotent] $\Leftrightarrow q \vee (\neg p \vee \neg r)$

[Assoc.] $\Leftrightarrow (q \vee \neg p) \vee \neg r$

[Commut.] $\Leftrightarrow \neg p \vee q \vee \neg r$

Q.E.D. (quod erat demonstrandum)

(Which was to be shown.)

Review: Propositional Logic (§§1.1-1.2)

- Atomic propositions: *p*, *q*, *r*, ...
- Boolean operators: $\neg \wedge \vee \oplus \rightarrow \leftrightarrow$
- Compound propositions: $s :\equiv (p \wedge \neg q) \vee r$
- Equivalences: $p \wedge \neg q \Leftrightarrow \neg(p \rightarrow q)$
- Proving equivalences using:
  - Truth tables.
  - Symbolic derivations. $p \Leftrightarrow q \Leftrightarrow r \ldots$