Discrete Structures

Propositional Logic

Foundations of Logic: Overview

- Propositional logic:
  - Basic definitions.
  - Equivalence rules & derivations.
- Predicate logic
  - Predicates.
  - Quantified predicate expressions.
  - Equivalences & derivations.

Propositional Logic

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

Applications:

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

Definition of a *Proposition*

A *proposition* (*p*, *q*, *r*, …) is simply a *statement* (*i.e.*, a declarative sentence) *with a definite meaning*, having a *truth value* that's either *true* (T) or *false* (F) (**never** both, neither, or somewhere in between).

Examples of Propositions

- "It is raining." (Given a situation.)
- "Beijing is the capital of China."
- "1 + 2 = 3"
- The following are **NOT** propositions:
- "Who's there?" (interrogative, question)
- "La la la la la." (meaningless interjection)
- "Just do it!" (imperative, command)
- "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.)

Unary operators take 1 operand (*e.g.*, -3);
Binary operators take 2 operands (*eg* 3×4).

Propositional or *Boolean* operators operate on propositions or truth values instead of on numbers.

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."

Truth table for NOT:

p	$\neg p$
T	F
F	T

The Conjunction Operator

The binary *conjunction operator* "$\wedge$" (*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.
- $\neg$ and $\wedge$ operations together are universal, i.e., sufficient to express *any* 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* "$\lor$" (*OR*) combines two propositions to form their logical *disjunction*.

E.g. p = "That car has a bad engine."
q = "That car has a bad carburetor."
$p \lor q$ = "Either that car has a bad engine, **or** that car has a bad carburetor."

Disjunction Truth Table

- Note that $p \vee q$ 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.
- "¬" and "∨" together are also universal.

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

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* "$\oplus$" (*XOR*) combines two propositions to form their logical "exclusive or" (exjunction?).

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

q = "I will drop this course,"

$p \oplus q$ = "I will either earn an *A* for 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.
- "¬" and "⊕" together are **not** universal.

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

Natural Language is Ambiguous

Note that <u>English</u> "or" is *by itself* ambiguous regarding the "both" case!

"Pat is a singer or Pat is a writer." -

"Pat is a man or Pat is a woman." -

Need context to disambiguate the meaning!

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

p	q	p or q
F	F	F
F	T	T
T	F	T
T	T	undef.

The *Implication* Operator

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

It is FALSE <u>only</u> in the case that p is TRUE but q is FALSE.

E.g., p = "I am elected."
q = "I will lower taxes."

$p \rightarrow q$ = "If I am elected, then I will lower taxes" (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** imply that p causes q!
- $p \rightarrow q$ does **not** imply that p or q **are ever true**!
- *E.g.* "(1=0) $\rightarrow$ cows can fly" is TRUE!

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

Different Ways to Express $p \to q$

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

Examples of Implications

- "If this lecture ends, then the sun will rise tomorrow." *True* or *False*?
- "If Tuesday is a day of the week, then I am a penguin." *True* or *False*?
- "If 1+1=6, then George passed the exam." *True* or *False*?
- "If the moon is made of green cheese, then I am richer than Bill Gates." *True* or *False*?

Inverse, Converse, Contrapositive

Some terminology:

- The *inverse* of $p \rightarrow q$ is: $\neg p \rightarrow \neg q$
- The *converse* of $p \rightarrow q$ is: $q \rightarrow p$.
- The *contrapositive* of $p \rightarrow q$ is: $\neg q \rightarrow \neg p$.
- One of these 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

The *biconditional* operator

The *biconditional* $p \leftrightarrow q$ states that p is true *if and only if (IFF)* q is true.
It is TRUE when both $p \rightarrow q$ and $q \rightarrow p$ are TRUE.

E.g. p = "It is raining."
q = "The home team wins."
$p \leftrightarrow q$ = "If and only if it is raining, the home team wins."

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!
  - $p \leftrightarrow q$ means $\neg(p \oplus q)$
- $p \leftrightarrow q$ does **not** imply p and q are true, or cause each other.

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

Boolean Operations Summary

- We have seen 1 unary operator (4 possible) and 5 binary operators (16 possible).

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

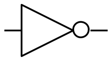
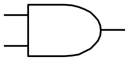
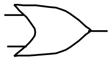
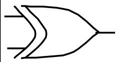
Precedence of Logical Operators

Operator	Precedence
$\neg$	1
$\wedge$	2
$\vee$	3
$\rightarrow$	4
$\leftrightarrow$	5

Expressions

- 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, "$\neg$" takes *precedence* over both "$\wedge$" and "$\vee$".
  - $\neg s \wedge f$ means $(\neg s) \wedge f$, **not** $\neg (s \wedge f)$

Some Alternative Notations

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

Translating English Sentences

- *You can access the Internet from campus only if you are a computer science major or you are not a freshman.*

a = "You can access the Internet from campus"

c = "You are a computer science major"

f = "You are a freshman"

Answer: $a \rightarrow (c \vee \neg f)$

Translating English Sentences

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

q = "You can ride the roller coaster "

r = "You are under 4 feet tall"

s = "You are older than 16 years old"

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

Bits and Bit Operations

- A *bit* is a binary (base 2) digit: 0 or 1.
- Bits may be used to represent truth values.
- By convention: 0 represents "true";
  1 represents "false".
- *Boolean algebra* is like ordinary algebra except that variables stand for bits, + means "or", and multiplication means "and".

Bit Strings

- A *Bit string* of *length n* is an ordered series or sequence of $n \geq 0$ bits.
- By convention, bit strings are written left to right: *e.g.* the first bit of "1001101010" is 1.
- When a bit string represents a base-2 number, by convention the first bit is the *most significant* bit. *Ex.* $1101_2=8+4+1=13$.

Bitwise Operations

- Boolean operations can be extended to operate on bit strings as well as single bits.
- E.g.:
  01 1011 0110
  <u>11 0001 1101</u>
  11 1011 1111 Bit-wise OR
  01 0001 0100 Bit-wise AND
  10 1010 1011 Bit-wise XOR

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 the truth values of its atomic propositions are!

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

Propositional Equivalence

Two *syntactically* (*i.e.,* textually) different compound propositions may be *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*.

Proving Equivalences

Compound propositions p and q are logically equivalent to each other **IFF** p and q contain the same truth values as each other in all rows of their truth tables.

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

Proving Equivalence via Truth Tables

Ex. Prove that $p \vee q \Leftrightarrow \neg(\neg p \wedge \neg q)$.

p	q	$p \vee q$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$	$\neg(\neg p \wedge \neg q)$
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 much more complicated propositions and to find equivalences for them.

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

More Equivalence Laws

- *Absorption*:

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

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

- *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.

- Implication: $p \to q \Leftrightarrow \neg p \lor q$
- Biconditional: $p \leftrightarrow q \Leftrightarrow (p \to q) \land (q \to p)$
  $p \leftrightarrow q \Leftrightarrow \neg(p \oplus q)$
- Exclusive or: $p \oplus q \Leftrightarrow (p \lor q) \land \neg(p \land q)$
  $p \oplus q \Leftrightarrow (p \land \neg q) \lor (q \land \neg p)$

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 $\rightarrow$] $\Leftrightarrow \neg(p \wedge \neg q) \vee (p \oplus r)$

[Defn. of $\oplus$] $\Leftrightarrow \neg(p \wedge \neg q) \vee ((p \vee r) \wedge \neg(p \wedge r))$

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

Example Continued...

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

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

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

$\Leftrightarrow q \vee (((\underline{\neg p \vee} (p \vee r)) \wedge (\underline{\neg p \vee} \neg(p \wedge r)))$

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

[trivial taut.] $\Leftrightarrow q \vee ((\underline{\mathbf{T}} \vee r) \wedge (\neg p \vee \neg(p \wedge r)))$

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

[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)

References

- *Sections 1.1 and 1.2 of the text book "Discrete Mathematics and its Applications" by Rosen, 6th edition.*
- The original slides were prepared by Bebis