Propositional Logic (命题逻辑)

Content

- The language of propositions (命题)
  - Logic operators (逻辑运算符)
  - Logic formula (逻辑公式)
  - Truth table (真值表)
- Applications of propositional logic

The Language of Propositions

The language of propositions

- Propositions (命题)
- Logic operators (逻辑运算符)
  - Negation (否定)
  - Connectives (连接词)
    - Conjunction (合取)
    - Disjunction (析取)
    - Conditional statement (条件)
    - Biconditional statement (双条件)
- Truth tables (真值表)

Proposition (Statement)

A Proposition is a sentence that is either **True** or **False**

Examples:

$2 + 2 = 4$ True

$3 \times 3 = 8$ False

787009911 is a prime (素数)

Today is Tuesday.

Non-examples:

$x+y>0$

$x^2+y^2=z^2$

They are true for some values of x and y
but are false for some other values of x and y.

Proposition

- A declarative sentence that is either true or false, but not both
  - "Washington, D.C., is the capital of USA."
  - "California is adjacent to New York."
  - "1+1=2"
  - "2+2=5"
  - *"What time is it?"*
  - *"Read this carefully."*
  - *"x+y=z"*
- Constructing propositions
  - Propositional variables: p, q, r, s, …
  - The proposition that is always true is denoted by **T** and the Proposition that is always false is denoted by **F**.
- Atomic proposition (原子命题)
- Compound proposition (复合命题)

Logic operators

- Negation (否定) (NOT, ¬)
- Conjunction (合取) (AND, ∧)
- Disjunction (析取) (OR, ∨)
  - Inclusive Or (可兼或)
  - Exclusive Or (不可兼或) (XOR, ⊕)
- Conditional statement (条件) (Implication) (IF THEN→)
  - Converse (逆换) , Inverse (反换), Contrapositive (逆反)
- Biconditional statement (双条件) (Equivalence) (IF AND ONLY IF↔)

Logic operators are used to construct new propositions from old propositions

Compound propositions (propositions + logic operators)

p = "it is hot" q = "it is sunny"

It is hot and sunny	$p \wedge q$
It is not hot but sunny	$\neg p \wedge q$
It is neither hot nor sunny	$\neg p \wedge \neg q$

We can also define logic operators on three or more statements, e.g. OR(P,Q,R)

Negation

- The **negation** of a proposition p is denoted by ¬p and has this truth table:

p	¬p
T	F
F	T

- Example: If p denotes "The earth is round.", then ¬p denotes "It is not the case that the earth is round," or more simply "The earth is not round."

Conjunction

- The **conjunction** of propositions p and q is denoted by $p \wedge q$ and has this truth table:

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

- Example: If p denotes "I am at home." and q denotes "It is raining." then $p \wedge q$ denotes "I am at home and it is raining."

Disjunction

- The **disjunction** of propositions p and q is denoted by $p \lor q$ and has this truth table:

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

- Example: If p denotes "I am at home." and q denotes "It is raining." then $p \lor q$ denotes "I am at home or it is raining."

The operator “Or” in English

- In English “Or” has two distinct meanings.
  - “Inclusive Or”: In the sentence “Students who have taken CS202 or Math120 may take this class,” we assume that students need to have taken one of the prerequisites, but may have taken both. This is the meaning of inclusive disjunction. For $p \vee q$ to be true, either one or both of p and q must be true.
  - “Exclusive Or”: When reading the sentence “Soup or salad comes with this entrée,” we do not expect to be able to get both soup and salad. This is the meaning of exclusive disjunction. In $p \oplus q$, one of p and q must be true, but not both. The truth table for $\oplus$ is:

p	**q**	$\mathbf{p \oplus q}$
T	T	F
T	F	T
F	T	T
F	F	F

Conditional Statement (Implication)

- If p and q are propositions, then $p \rightarrow q$ is a conditional statement or **implication** which is read as "if p, then q" and has this truth table:

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

- Example: If p denotes "I am at home." and q denotes "It is raining." then $p \rightarrow q$ denotes "If I am at home then it is raining."
- In $p \rightarrow q$, p is the hypothesis (antecedent or premise) and q is the conclusion (or consequence).

Understanding Conditional Statement

- In $p \to q$ there does not need to be any connection between the antecedent or the consequent. The "meaning" of $p \to q$ depends only on the truth values of p and q.
- These implications are perfectly fine, but would not be used in ordinary English.
  - "If the moon is made of green cheese, then I have more money than Bill Gates. "
  - "If the moon is made of green cheese then I'm on welfare."
  - "If 1 + 1 = 3, then your grandma wears combat boots."

Understanding Conditional Statement (cont)

- One way to view the logical conditional is to think of an obligation or contract.
  - "If I am elected, then I will lower taxes."
  - "If you get 100% on the final, then you will get an A."
- If the politician is elected and does not lower taxes, then the voters can say that he or she has broken the campaign pledge. Something similar holds for the professor. This corresponds to the case where p is true and q is false.

Different ways of expressing $p \rightarrow q$

- if p, then q
- if p, q
- **q unless $\neg$p**
- **q if p**
- q whenever p
- q follows from p
- p implies q
- **p only if q**
- p is sufficient for q
- q when p
- q is necessary for p

- A necessary condition for p is q
- A sufficient condition for q is p

Common mistake for $p \rightarrow q$
- Correct: p only if q
- Mistake to think "q only if p"

Converse, Inverse and Contrapositive

- From $p \to q$ we can form new conditional statements.
  - $q \to p$ is the **converse**（逆） of $p \to q$
  - $\neg p \to \neg q$ is the **inverse**（否） of $p \to q$
  - $\neg q \to \neg p$ is the **contrapositive**（逆否） of $p \to q$
- Example: Find the converse, inverse, and contrapositive of "It is raining is a sufficient condition for I am not going to town."
- Solution:
  - Converse: If I do not go to town, then it is raining.
  - Inverse: If it is not raining, then I will go to town.
  - Contrapositive: If I go to town, then it is not raining.

Biconditional Statement

- If p and q are propositions, then we can form the **biconditional statement** proposition $p \leftrightarrow q$, read as "p if and only if q."
- The biconditional statement $p \leftrightarrow q$ denotes the proposition with this truth table:

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

- If p denotes "I am at home." and q denotes "It is raining." then $p \leftrightarrow q$ denotes "I am at home if and only if it is raining."

Expressing the Biconditional Statement

- Some alternative ways “p if and only if q” is expressed in English:
  - p is necessary and sufficient for q
  - if p then q, and conversely
  - p iff q

Precedence (优先级) of logic operators

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

- From left to right, parentheses "(,)" has the highest precedence
- $p \vee q \rightarrow \neg r$ is equivalent to $(p \vee q) \rightarrow \neg r$
- If the intended meaning is $p \vee (q \rightarrow \neg r)$, then parentheses must be used.

Truth tables for compound propositions

- Construction of a truth table:
- Rows
  - Need a row for every possible combination of values for the atomic propositions.
- Columns
  - Need a column for the compound proposition (usually at far right)
  - Need a column for the truth value of each expression that occurs in the compound proposition as it is built up.
    - This includes the atomic propositions

Problem

- How many rows are there in a truth table with n propositional variables?
- **Solution**: 2^n
- Note that this means that with n propositional variables, we can construct 2^n distinct (i.e., not equivalent) propositions.

Example truth table

- Construct a truth table for $p \vee q \rightarrow \neg r$

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

Equivalent propositions

- Two propositions are equivalent if they always have the same truth value.
- **Example**: Show using a truth table that the conditional statement is equivalent to the contrapositive.
- **Solution**:

p	q	¬p	¬q	p→q	¬q→¬p
T	T	F	F	T	T
T	F	F	T	F	F
F	T	T	F	T	T
F	F	T	T	T	T

Using a truth table to show non-equivalence

- **Example**: Show using truth tables that neither the converse nor inverse of an conditional statement are not equivalent to the conditional statement .
- **Solution**:

p	q	¬p	¬q	p→q	¬p→¬q	q→p
T	T	F	F	T	T	T
T	F	F	T	F	T	T
F	T	T	F	T	F	F
F	F	T	T	F	T	T

Applications of Propositional Logics

Translating English to logical expressions

Why?

- English is often ambiguous and translating sentences into compound propositions removes the ambiguity
- Using logical expressions, we can analyze them and determine their truth values
- We can use rules of inferences to reason about them

Translating English sentences

- Steps to convert an English sentence to a statement in propositional logic
  - Identify atomic propositions and represent using propositional variables.
  - Determine appropriate logical connectives
- "If I *go* to Harry's or to the country, I will not *go* shopping."
  - p: I *go* to Harry's
  - q: I *go* to the country.
  - r: I will *go* shopping.

If p or q then not r.

$p \vee q \rightarrow \neg r$

Examples

Problem: Translate the following sentence into propositional logic:

- "You can access the Internet from campus only if you are a computer science major or you are not a freshman."
- One solution:
  - p: "You can access the internet from campus,"
  - q: "You are a computer science major,"
  - r: "You are a freshman."

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

Different ways of expressing $p \rightarrow q$

- if p, then q
- if p, q
- **q unless ¬p**
- **q if p**
- q whenever p
- q follows from p
- p implies q
- **p only if q**
- q when p
- p is sufficient for q
- q is necessary for p

- A necessary condition for p is q
- A sufficient condition for q is p

Common mistake for $p \rightarrow q$

- Correct: p only if q
- Mistake to think "q only if p"

System specification (系统规范说明)

- Translating sentences in natural language into logical expressions is an essential part of specifying both hardware and software systems.
- Consistency of system specification.

Example

Express the specification "The automated reply cannot be sent when the file system is full"

- Let p denote "The automated reply can be sent"
- Let q denote "The file system is full"
- The logical expression for the sentence "The automated reply cannot be sent when the file system is full" is

$$q \rightarrow \neg p$$

Different ways of expressing p→q

- if p, then q
- if p, q
- **q unless ¬p**
- **q if p**
- q whenever p
- q follows from p
- p implies q
- **p only if q**
- q when p
- p is sufficient for q
- q is necessary for p

- A necessary condition for p is q
- A sufficient condition for q is p

Common mistake for p→q
- Correct: p only if q
- Mistake to think "q only if p"

Boolean searches

- Logical connectives are used extensively in searches of large collections of information, such as indexes of webpages: Such searches are called **Boolean Searches**
- The connective AND is used to match records that contain both of the two search terms
- The connective OR is used to match one or both of the search terms
- The connective NOT (sometimes written as AND NOT) is used to exclude a search term

Web page searching

- Most Web search engines support Boolean searching techniques
- Using Boolean searching to find Web pages about universities in British Columbia we can look for pages matching

  British AND Columbia AND Universities
- **The AND operator**: Note that in Google the word "AND" is not needed, although it is implicit
- Google also supports the use of quotation marks (" ") to search for specific phrases: It may be more effective to search for

  "British Columbia" Universities

Web page searching

- **The OR operator:** In Google "The OR operator, for which you may also use | (vertical bar), applies to the search terms immediately adjacent to it."
- In Google, the terms used for searching olympics locations for 2014 or 2018 would be

  olympics (2014 OR 2018)

  olympics (2014 | 2018)

  interpreted as olympics AND (2014 OR 2018)

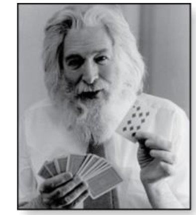
  To find universities in British Columbia or Ontario we would use

  "British Columbia" OR Ontario universities
- **The NOT operator**: To find webpages that deal with universities in Columbia (but not in British Columbia) we search for

  (Columbia AND Universities) NOT British
- In Google, the word NOT is replaced by the symbol "-" (minus)

  Columbia Universities -British

Logic puzzles (逻辑谜题)



Raymond Smullyan
(Born 1919)

- An island has two kinds of inhabitants, knights (骑士), who always tell the truth, and knaves (流氓), who always lie.
- You go to the island and meet A and B.
  - A says "B is a knight."
  - B says "The two of us are of opposite types."

Puzzle: What are the types of A and B?

Solution:

Let p and q be the statements that A is a knight and B is a knight, respectively. So, then $\neg p$ represents the proposition that A is a knave and $\neg q$ that B is a knave.

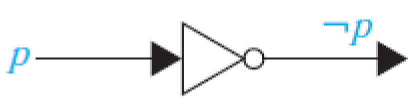
- If A is a knight, then p is true. Since knights tell the truth, q must also be true. Then $(p \wedge \neg q) \vee (\neg p \wedge q)$ would have to be true, but it is not. So, A is not a knight and therefore $\neg p$ must be true and we explore that possibility.
- If A is a knave, then B must not be a knight since knaves always lie. So, then both $\neg p$ and $\neg q$ hold, and thus **both are knaves.**

Logic circuits

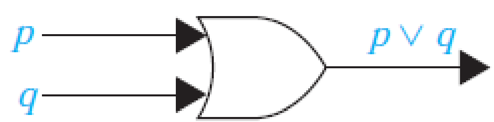
- Propositional logic can be applied to the design of computer hardware.
- A logic circuit (or digital circuit) receives input signals $p_1, p_2, \dots, p_n$, each a bit [either 0 (off) or 1 (on)], and produces output signals $s_1, s_2, \dots, s_n$, each a bit.
- Here we restrict our attention to logic circuits with a single output signal; in general, digital circuits may have multiple outputs.

Logic gates

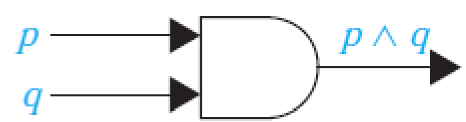
- Complicated digital circuits can be constructed from three basic circuits, called **gates**
- The **inverter**, or **NOT gate**, takes an input bit p, and produces as output $\neg p$.
- The **OR gate** takes two input signals p and q, each a bit, and produces as output the signal $p \vee q$.
- The **AND gate** takes two input signals p and q, each a bit, and produces as output the signal $p \wedge q$



Inverter



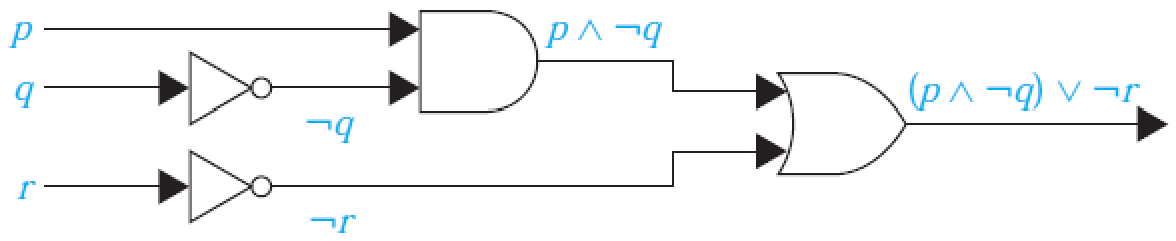
OR gate



AND gate

Examples

- Determine the output for the combinatorial circuit as follows



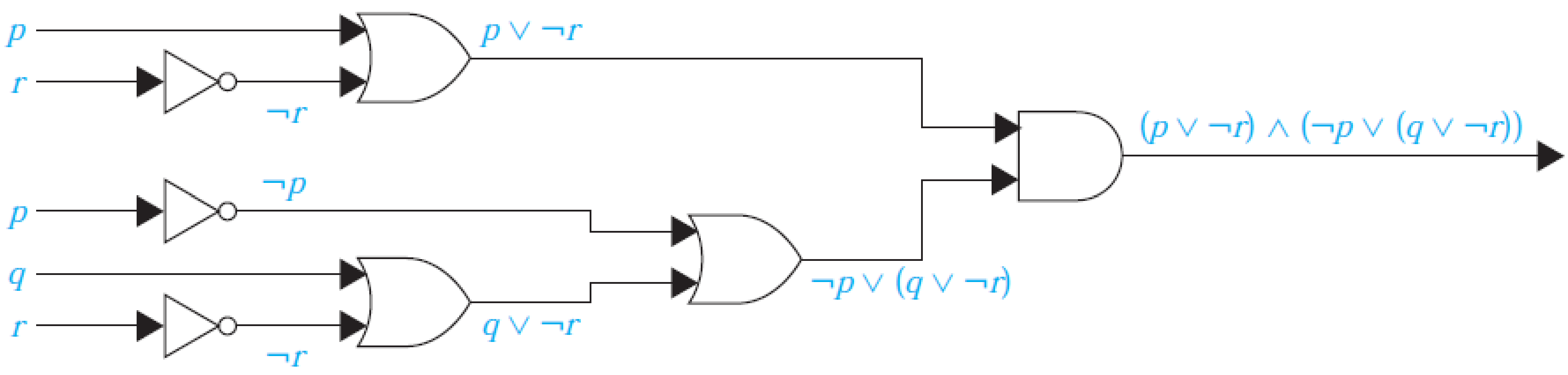
- We see that the AND gate takes input of p and $\neg$q, the output of the inverter with input q, and produces p $\wedge \neg$q.
- Next, we note that the OR gate takes input p $\wedge \neg$q and $\neg$r, the output of the inverter with input r, and produces the final output (p $\wedge \neg$q)$\vee \neg$r.

Examples

Problem: Build a digital circuit that produces the output $(p \vee \neg r) \wedge (\neg p \vee (q \vee \neg r))$ when given input bits p, q, and r.

Solution:

- To construct the desired circuit, we build separate circuits for $p \vee \neg r$ and for $\neg p \vee (q \vee \neg r)$ and combine them using an AND gate.
- To construct a circuit for $p \vee \neg r$, we use an inverter to produce $\neg r$ from the input r. Then, we use an OR gate to combine p and $\neg r$.
- To build a circuit for $\neg p \vee (q \vee \neg r)$, we first use an inverter to obtain $\neg r$. Then we use an OR gate with inputs q and $\neg r$ to obtain $q \vee \neg r$.
- Finally, we use another inverter and an OR gate to get $\neg p \vee (q \vee \neg r)$ from the inputs p and $q \vee \neg r$.
- To complete the construction, we employ a final AND gate, with inputs $p \vee \neg r$ and $\neg p \vee (q \vee \neg r)$.



p
r
¬r
p ∨ ¬r
p
¬p
q
r
¬r
q ∨ ¬r
¬p ∨ (q ∨ ¬r)
(p ∨ ¬r) ∧ (¬p ∨ (q ∨ ¬r))