Chapter 1. The Foundations: Logic and Proofs

Items italic and with an * (an asterisk) are ones unimportant.

Items with an * are ones not mentioned by teacher in the class.

Items with two *s are ones mentioned by teacher but cannot be found on the Internet.

Proposition

- A proposition is a declarative sentence that is either true or false, but not both.
- e.g.
  1. Beijing is the capital of China.
  2. $1 + 1 = 3$
  3. Is Beijing the capital of China?
  4. $x + 2 = 3$
  - 1 and 2 are propositions. Proposition 1 is true while 2 is false.
  - 3 is not a proposition since it is a question, not a declarative sentence.
  - 4 is not a proposition as it can be either true or false. However, it would become a proposition as we assign a value to x.
- We also need variables to denote propositions: propositional variables, or sentential variables.
- Typically we use letters $p, q, r, s, \ldots$ for propositional variables.
- For a proposition p, we can ask if p is true or false. And it is called the truth value of p.
- Propositions that cannot be expressed in terms of simpler propositions are called atomic propositions (or atomic formulas, atom for short).
- We can use connectives, or called logical operators, to combine propositions into new propositions, called compound propositions.
- Above all, what we are studying is to deal with propositions, called propositional calculus or propositional logic.

Logic Basis

- Describing the truth value of a compound proposition with all of the possible combinations of values can be annoying. We shall use a truth table.
- A proposition p can be negated. The negation of p is denoted by $\neg p$ or $\bar{p}$, with a meaning of "not p". It has a truth table of

p	$\neg p$
T	F
F	T

- The conjunction of p and q is denoted by $p \wedge q$, with a meaning of "p and q".
  - Note that in logic the word "but" sometimes used instead of "and" in conjunction.
  - $p \wedge q$ is true if when both p and q are true and is false otherwise.
- The disjunction of p and q is denoted by $p \vee q$, with a meaning of "p or q".
  - The word "or" has two meanings in English. The disjunction corresponds to the inclusive or.
  - $p \vee q$ is true when at least one in p and q is true and is false otherwise.
- The truth table of conjunction and disjunction:

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

Extra Logical Operators

- The exclusive or of p and q is denoted by $p \oplus q$ (or $p \operatorname{XOR} q$ or $p \dot{\vee} q$).
  - $p \oplus q$ is true when exactly one of p and q is true and is false otherwise.
- The conditional statement is denoted by $p \rightarrow q$, with a meaning of "if p then q". It is also called implication.
  - In the conditional statement, p is called the hypothesis (or antecedent or premise) and q is called the conclusion (or consequence).
  - $p \rightarrow q$ is false when p is true and q is false. It is logically equivalent to $\neg p \vee q$.
    - If two propositions always share the same truth values, regardless of the truth values of their propositional variables, we say that they are equivalent, or logically equivalent.
  - There are various ways to express conditional statement:

"if p, then q"	"p implies q"
"if p, q"	"p only if q"
"p is sufficient for q"	"a sufficient condition for q is p"
"q if p"	"q whenever p"
"q when p"	"q is necessary for p"
"a necessary condition for p is q"	"q follows from p"
"q unless $\neg p$"	"q provided that p"

- The biconditional statement is denoted by $p \leftrightarrow q$, with a meaning of "p if and only if q" ("p iff q" for short).
  - $p \leftrightarrow q$ is true when p and q have the same truth values.
- The truth table:

p	q	$p \oplus q$	$p \rightarrow q$	$p \leftrightarrow q$
T	T	F	T	T
T	F	T	F	F
F	T	T	T	F
F	F	F	T	T

Based on Conditionals

- For a conditional proposition $p \rightarrow q$
  - $q \rightarrow p$ is called its converse.
  - $\neg p \rightarrow \neg q$ is called its inverse.
  - $\neg q \rightarrow \neg p$ is called its contrapositive.
- A conditional statement and its contrapositive are equivalent.

- The truth values of the converse and the inverse are not dependent on the original proposition. However, they are equivalent as they are contrapositives with each other.

Precedence of Logical Operators

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

Note that the table does not contain exclusive or (operator $\oplus$). Actually there is not a general agreement on this problem. In practice, it is better to use parentheses or expand the exclusive or by definition to avoid ambiguity.

*Bitwise Operations

- A bit is a symbol with a value of either `0` or `1`.
- Computer bit operations correspond to the logical connectives by replacing true by a `1` and false by a `0` in the truth table for the operators $\vee, \wedge$ and $\oplus$. We also use notation OR, AND and XOR respectively.
- Bit operations can be extend to bit strings. We define the bitwise OR, bitwise AND, and bitwise XOR of two bit strings of the same length to be the strings that have as their bits the OR, AND, and XOR of the corresponding bits in the two strings.

*System Specifications

- System and software engineers often need to translate natural language into precise and unambiguous specifications. Symbolic propositions are often used as system specifications.
- A set of system specifications is consistent iff it is possible to assign truth values to the proposition variables so that each proposition is true. (remember that "iff" is the contraction of "if and only if"?)

*Boolean Search

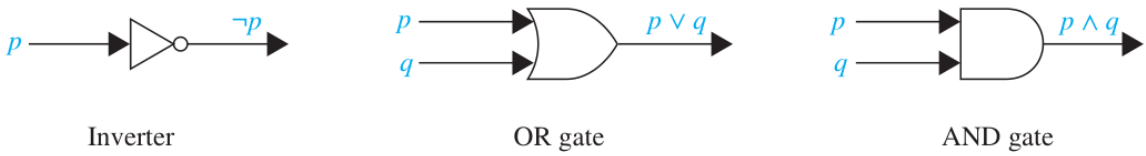
- When searching for something, Boolean search can be used to precisely locate the desired item. That is, using connectives AND, OR, and NOT to match or exclude particular items.

*Logic Puzzles

- Puzzles that can be solved using logical reasoning are called logic puzzles. Propositional logic can be used to solve them.

Logic Circuits

- A logic circuit (or digital circuit) receives input signals and produces output signals by bits.
- Complicated digital circuits can be constructed from three basic gates:



Inverter OR gate AND gate

  - 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 $p \wedge q$.
- The digital circuit can be many students' nightmare.

Propositional Equivalence

- A compound proposition that is always true is called a tautology. A compound proposition that is always false is called a contradiction. A compound proposition that is neither a tautology nor a contradiction is called a contingency.
- Propositions p and q are equivalent iff $p \leftrightarrow q$ is a tautology.
- The notations $p \equiv q$ and $p \Leftrightarrow q$ are used to denote that p and q are logically equivalent.

Important Equivalences

- Some equivalences are important enough to have a name.

Equivalence	Name
$p \wedge \mathrm{T} \equiv p$ $p \vee \mathrm{F} \equiv p$	Identity laws
$p \vee \mathrm{T} \equiv \mathrm{T}$ $p \wedge \mathrm{F} \equiv \mathrm{F}$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv \mathrm{T}$ $p \wedge \neg p \equiv \mathrm{F}$	Negation laws

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

$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$
$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

Normal Forms of Compound Propositions

- An atom proposition or its negation is called a *literal.
- A disjunction of literals is called a *clause (or simply a "disjunction").
- A conjunction of literals is called a *term (or simply a "conjunction").
- A disjunction of terms, namely an $\vee$ of $\wedge$s, is called a disjunctive normal form (or DNF for short).
- A conjunction of clauses, namely an $\wedge$ of $\vee$s, is called a conjunctive normal form (or CNF for short).
- A clause contains all variables is called a **maxterm.
- A term contains all variables is called a **minterm.
- For a DNF, if its every term is a minterm, the DNF is called a **major DNF (or **principal DNF or *full DNF or *canonical DNF).
- For a CNF, if its every clause is a maxterm, the CNF is called a **major CNF (or **principal CNF or *full CNF or *canonical CNF).

- Theorem: Any proposition can be converted into at least a DNF or at least a CNF.
- Theorem: Any proposition can be converted into a unique DNF or a unique CNF.
- Methods to convert a proposition into a major DNF
  - Using a truth table:
    1. Write the truth table of the proposition.
    2. Take a disjunction of all satisfying truth assignments.
  - Using propositional equivalence
    1. Eliminate implication signs.
    2. Moving negation inwards and eliminate double negation.
    3. Convert to DNF using associative and distributive laws.
    4. If there is a term P that is not a minterm and misses proposition variable p, replace it with $(p \wedge \neg p) \vee P$.
    5. Expand the new term using distributive law.
    6. Repeat steps 4 and 5 until there are no more terms that are not minterms. Merge the same minterm, and we finally get the major DNF.
- Methods to convert a proposition into a major CNF
  - Using a truth table:
    1. Write the truth table of the proposition.
    2. Take a conjunction of all falsifying truth assignments.
  - Using propositional equivalence
    1. Eliminate implication signs.
    2. Moving negation inwards and eliminate double negation.
    3. Convert to CNF using associative and distributive laws.
    4. If there is a clause P that is not a maxterm and misses proposition variable p, replace it with $(p \vee \neg p) \wedge P$.
    5. Expand the new clause using distributive law.
    6. Repeat steps 4 and 5 until there are no more clauses that are not maxterms. Merge the same maxterm, and we finally get the major CNF.
- The methods to convert a proposition into a major normal form ("major NF" below) are tedious and fallible. There would be better an example to present how to use the methods. But I'm lazy and tired.
- The major NF we get can sometimes grow very big. In fact, if there are n propositional variables, the number of minterms or maxterms in the major NF can up to 2^n.

Satisfiability of a Proposition

- A proposition is satisfiable if there is an assignment of truth values to its variables that makes it true. Otherwise it is unsatisfiable. Such an assignment to make it true is called a solution.
- A proposition is satisfiable iff it is a tautology or a contingency.
- The Satisfiability of a proposition can be used to solve problems such as The n-Queens Problem or Sudoku.

Appendix: Glossary

English terms with an * are ones not mentioned by teacher and textbook. (their translation can be mentioned or not)

Translations or terms with two *s are ones mentioned by teacher but cannot be found on the Internet.

terms in English	中文译名
proposition	命题
propositional variable / sentential variable	命题变量
truth value	真值
atomic proposition / *atomic formula (atom)	原子命题 / 原子公式
connectives / logic operator	逻辑运算（符）
compound proposition	复合命题
propositional calculus / propositional logic	命题演算 / 命题逻辑
truth table	真值表
negation	否定
conjunction	合取
disjunction	析取
inclusive or	或 / 包含或
exclusive or	异或 / 互斥或
conditional statement	条件语句
implication	蕴含
logically equivalent	逻辑等价
equivalent	等价
biconditional	
converse	逆命题
inverse	否命题
contrapositive	逆否命题
bit	位
bit operation	位运算
system specifications	
consistent	
Boolean search	
inverter	
gate	门
NOT gate	非门
OR gate	或门
AND gate	与门
tautology	重言式 / 永真式
contradiction	
contingency	

terms in English	中文译名
*literal	字面量
*clause	子句 / 简单析取式 / 基本和
*term	短语 / 简单合取式 / 基本积
disjunctive normal form (DNF)	析取范式
conjunctive normal form (CNF)	合取范式
**minterm	最小项
**maxterm	最大项
**major DNF / **principal DNF	主析取范式
*full DNF / *canonical DNF	主析取范式
**major CNF / **principal CNF	主合取范式
*full CNF / *canonical CNF	主合取范式
satisfiable	可满足
unsatisfiable	不可满足
solution	解
The n-Queens Problem	n 皇后问题
Sudoku	数独