1.2 Propositions 命题

Logic is a system based on **propositions**.
A sentence that is either true or false, but not both, is called a **proposition**.

Truth Value of a proposition

True: T

False: F

1.2 Propositions

Logic is a system based on **propositions**.
A sentence that is either true or false, but not both, is called a **proposition**.

Truth Value of a proposition

True: T

False: F

“Elephants are bigger than mice.”

1.2 Propositions

Example 1 Which of the following are propositions? Give the truth value of the propositions.

a. $2 + 3 = 7$.

b. Julius Caesar was president of the United States.

c. What time is it?

d. Be quiet !

e. The difference of two primes.

f. Washington D.C. is the capital of New York.

g. How are you?

1.2 Propositions

The Earth is the only planet where life exists.

1.2 Propositions

Logic is a system based on **propositions**.
A sentence that is either true or false, but not both, is called a **proposition**.

We use variables, such as p, q and r, to represent propositions.

We will also use the notation

$$p\text{: } 1 + 1 = 3$$

to define p to be the proposition 1+1=3.

1.2 Propositions

Definition 1.2.1 Let p and q be propositions.

The conjunction **（合取）** of p and q, denote $\boldsymbol{p \wedge q}$, is the proposition

p **and** q.

The disjunction **（析取）** of p and q, denote $\boldsymbol{p \vee q}$, is the proposition

p **or** q.

1.2 Propositions

Definition 1.2.1 Let p and q be propositions.

The conjunction（合取）of p and q, denote $\boldsymbol{p \wedge q}$, is the proposition

p **and** q.

The disjunction（析取）of p and q, denote $\boldsymbol{p \vee q}$, is the proposition

p **or** q.

Example: If p: It is raining, and q: It is cold, then

$p \wedge q$: ?

$p \vee q$: ?

1.2 Propositions

Definition 1.2.9 The negation of p, denote $\neg p$, is the proposition **not** p.

Example: Tim is a boy.

Logic Operators

$\wedge \quad ::= \ \text{AND} \qquad \vee \quad ::= \ \text{OR} \qquad \neg \quad ::= \ \text{NOT}$

Example:

p: It is hot. q: It is sunny.

It is hot and sunny.

It is not hot but sunny

It is neither hot nor sunny

Logic Operators

$\wedge \ ::= \ \text{AND} \qquad \vee \ ::= \ \text{OR} \qquad \neg \ ::= \ \text{NOT}$

Truth Table

p	q	$p \wedge q$

p	q	$p \vee q$

Logic Operators

$\wedge \;::=\;$ AND $\quad \vee \;::=\;$ OR $\quad \neg \;::=\;$ NOT

Truth Table

p	$\neg p$

Logic Operators

$\wedge$ ::= AND $\vee$ ::= OR $\neg$ ::= NOT

Truth Table

Think about a general version of a truth table.

p	$\neg p$

p_1 p_2 p_3 P

Logic Operators

$\wedge$ ::= AND $\vee$ ::= OR $\neg$ ::= NOT

Truth Table

Think about a general version of a truth table.

p	$\neg p$

$p_1\ p_2\ p_3 \cdots p_n$ P

Truth table of (p∨q)^r

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

Truth table of (p∨q)^r

p	q	r	$p \vee q$	$(p \vee q) \wedge r$
T	T	T	T	T
T	T	F	T	F
T	F	T	T	T
T	F	F	T	F
F	T	T	T	T
F	T	F	T	F
F	F	T	F	F
F	F	F	F	F

Logic Operators

$\wedge$ ::= AND $\vee$ ::= OR $\neg$ ::= NOT

Operator Precedence 操作符的优先级

In the absence of parentheses,
we first evaluate $\neg$,
then $\wedge$,
and then $\vee$.

Example： Given that proposition p is false, proposition q is true, and proposition r is false, determine whe ther the proposition $\neg p \vee q \wedge r$ is true or false.

Operator Precedence 操作符的优先级

In the absence of parentheses,
we first evaluate $\neg$,
then $\wedge$,
and then $\vee$.

Example：

Given that proposition p is false, proposition q is true, and proposition r is false, determine whether each proposition in Exercises 17–22 is true or false.

17. $p \vee q$

18. $\neg p \vee \neg q$

19. $\neg p \vee q$

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

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

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

Operator Precedence 操作符的优先级

In the absence of parentheses,
we first evaluate $\neg$,
then $\wedge$,
and then $\vee$.

Logic Operators: Exclusive-Or (异或)

$\wedge$ ::= AND $\quad\vee$ ::= OR (兼或) $\quad\neg$ ::= NOT

$\oplus$ exclusive-or（异或）

p	q	$p \vee q$

p	q	$p \oplus q$

Logic Operators: Exclusive-Or (异或)

$\wedge$ ::= AND $\vee$ ::= OR (兼或) $\neg$ ::= NOT

Does this definition make sense? ← $\oplus$ exclusive-or（异或）

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

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

组合电路

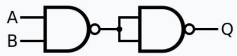
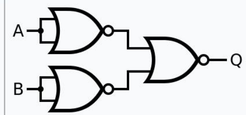
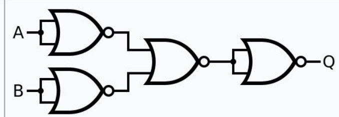
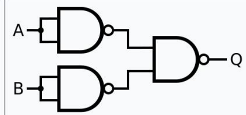
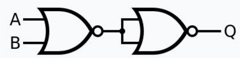
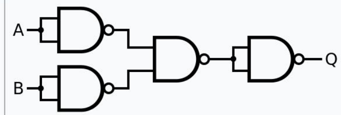
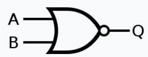
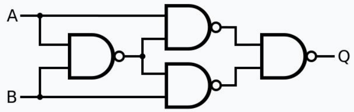
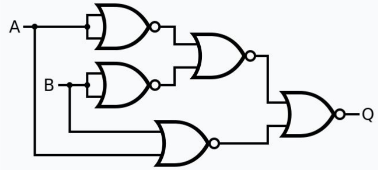
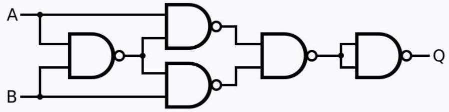
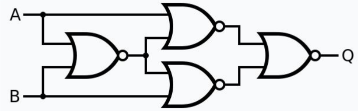
T: has power

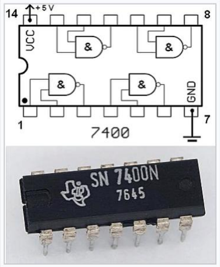
F: do not has power

Universal logic gates [edit]

Further information on the theoretical basis: Functional completeness

Charles Sanders Peirce (during 1880–81) showed that NOR gates alone (or alternatively NAND gates alone) can be used to reproduce the functions of all the other logic gates, but his work on it was unpublished until 1933.[15] The first published proof was by Henry M. Sheffer in 1913, so the NAND logical operation is sometimes called Sheffer stroke; the logical NOR is sometimes called *Peirce's arrow*.[16] Consequently, these gates are sometimes called *universal logic gates*.[17]

type	NAND construction	NOR construction
NOT		
AND		
NAND		
OR		
NOR		
XOR		
XNOR		



The 7400 chip, containing four NANDs. The two additional pins supply power (+5 V) and connect the ground.

How to construct a compound proposition for exclusive-or?

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

How to construct a compound proposition for exclusive-or?

Definition 1.3.10 Suppose that the propositions P and Q are made up of the propositions $p_1, p_2, p_3, \ldots, p_n$. We said that P and Q are **logically equivalent (逻辑等价)**, and write

$$P \equiv Q,$$

provided that given any truth value of $p_1, p_2, p_3, \ldots, p_n$, either P and Q are both true, or P and Q are both false.

How to construct a compound proposition for exclusive-or?

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

How to construct a compound proposition for exclusive-or?

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

Problem-Solving Tips

Although there may be a shorter way to determine the truth values of a proposition P formed by combining propositions $p_1, \ldots, p_n$ using operators such as $\neg$ and $\vee$, a truth table will always supply all possible truth values of P for various truth values of the constituent propositions $p_1, \ldots, p_n$.