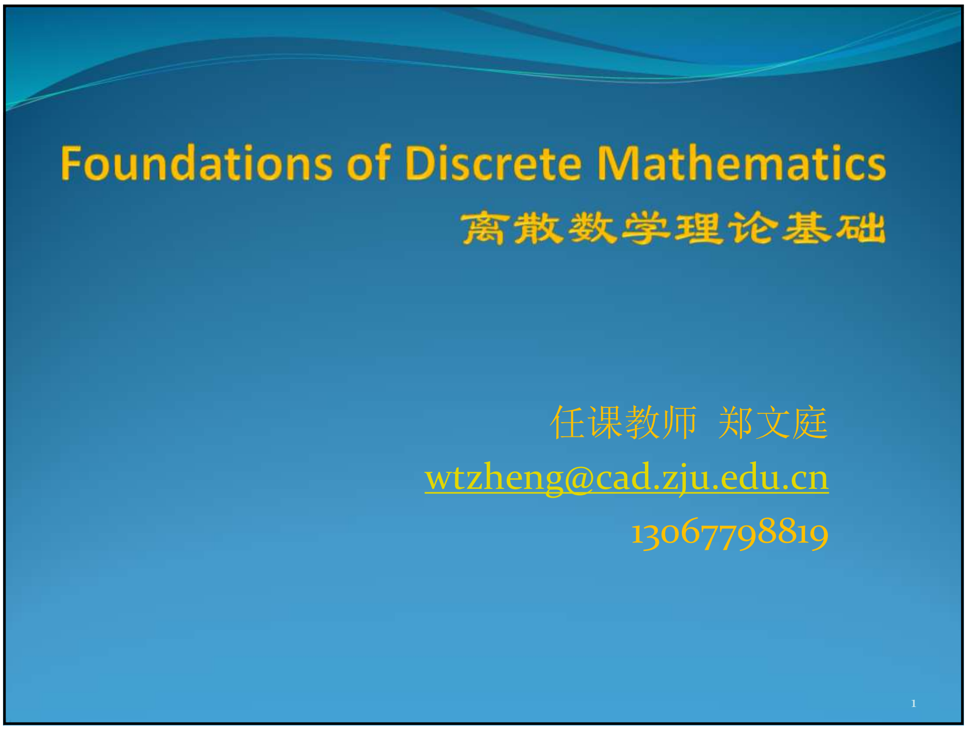


Foundations of Discrete Mathematics

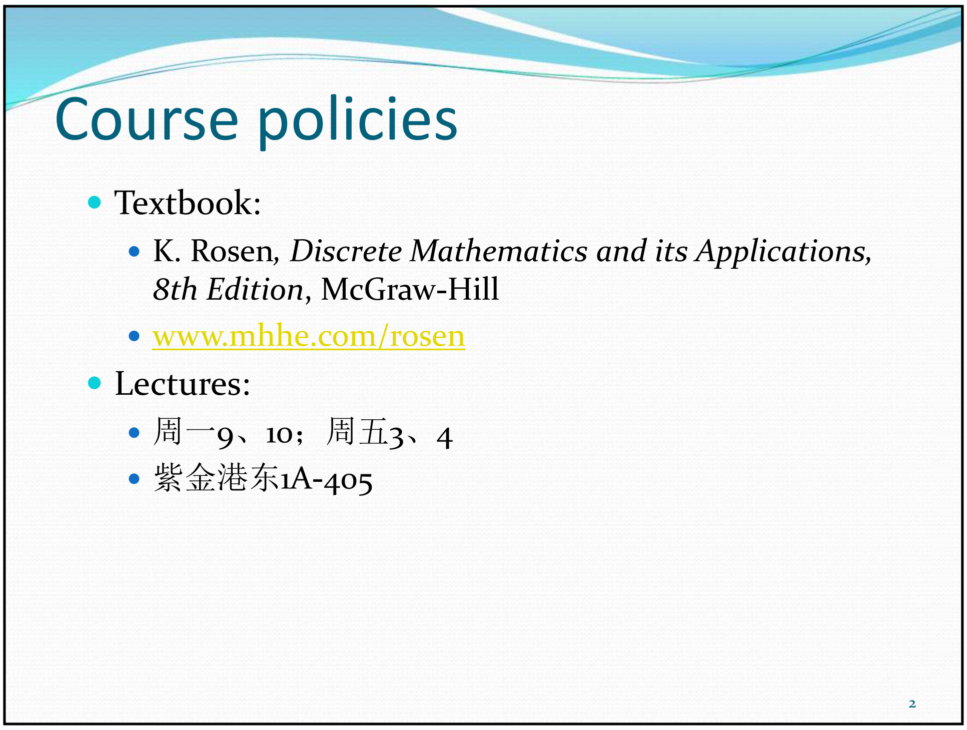
离散数学理论基础

任课教师 郑文庭

wtzheng@cad.zju.edu.cn

13067798819

1



Course policies

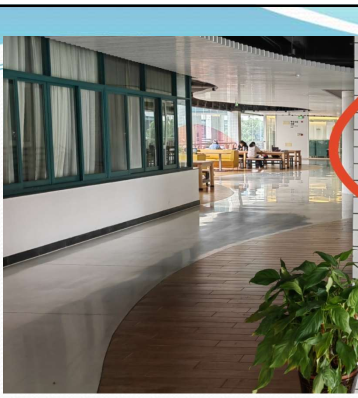
- Textbook:
  - K. Rosen, *Discrete Mathematics and its Applications, 8th Edition*, McGraw-Hill
  - www.mhhe.com/rosen
- Lectures:
  - 周一9、10；周五3、4
  - 紫金港东1A-405

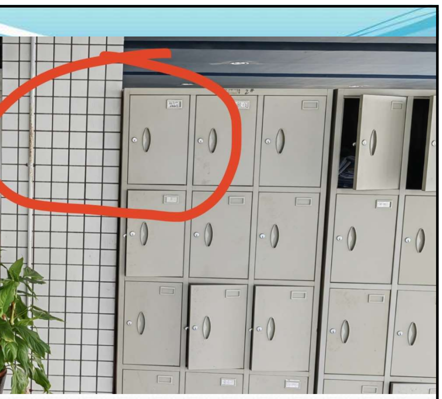
2

Course policies

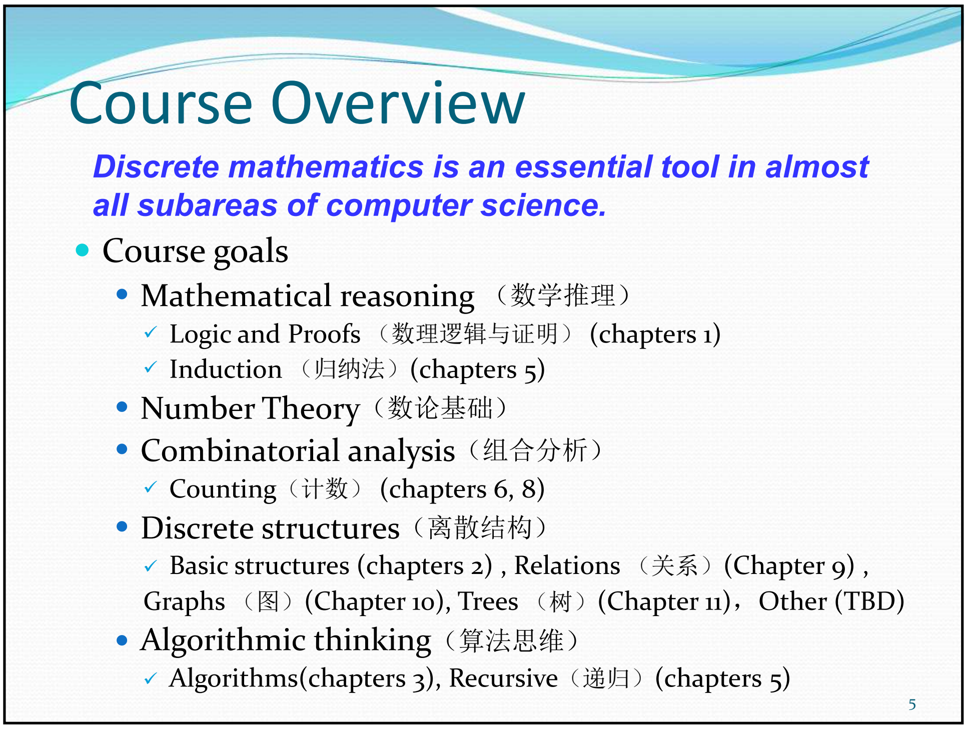
- Grading Scheme
  - Homework (10%):
    每周作业下周一交。除非事先告知，否则不得迟交。
  - Quizzes (40%): 不定期，4次。（有可能根据实际情况调整次数）
  - Final (50%)

- 助教：臧斌，研究生。





- 补交作业可投递：
  东1B 2楼 过道柜子

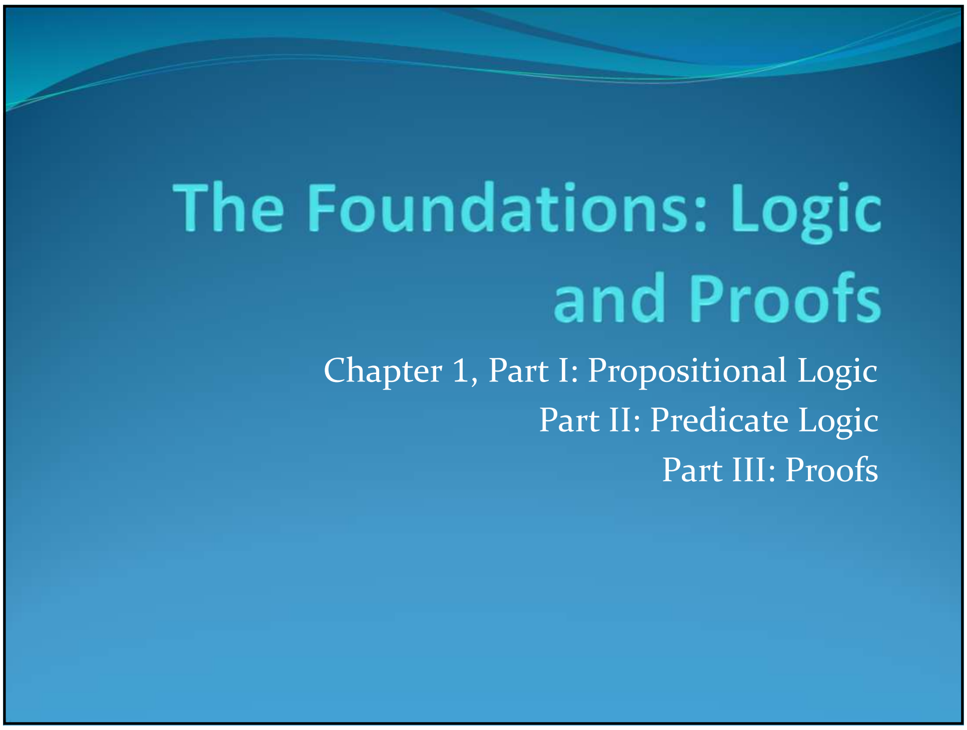


Course Overview

Discrete mathematics is an essential tool in almost all subareas of computer science.

- Course goals
  - Mathematical reasoning （数学推理）
    - ✓ Logic and Proofs （数理逻辑与证明） (chapters 1)
    - ✓ Induction （归纳法）(chapters 5)
  - Number Theory（数论基础）
  - Combinatorial analysis（组合分析）
    - ✓ Counting（计数） (chapters 6, 8)
  - Discrete structures（离散结构）
    - ✓ Basic structures (chapters 2) , Relations （关系）(Chapter 9) , Graphs （图）(Chapter 10), Trees （树）(Chapter 11)， Other (TBD)
  - Algorithmic thinking（算法思维）
    - ✓ Algorithms(chapters 3), Recursive（递归）(chapters 5)

5



The Foundations: Logic and Proofs

Chapter 1, Part I: Propositional Logic
Part II: Predicate Logic
Part III: Proofs

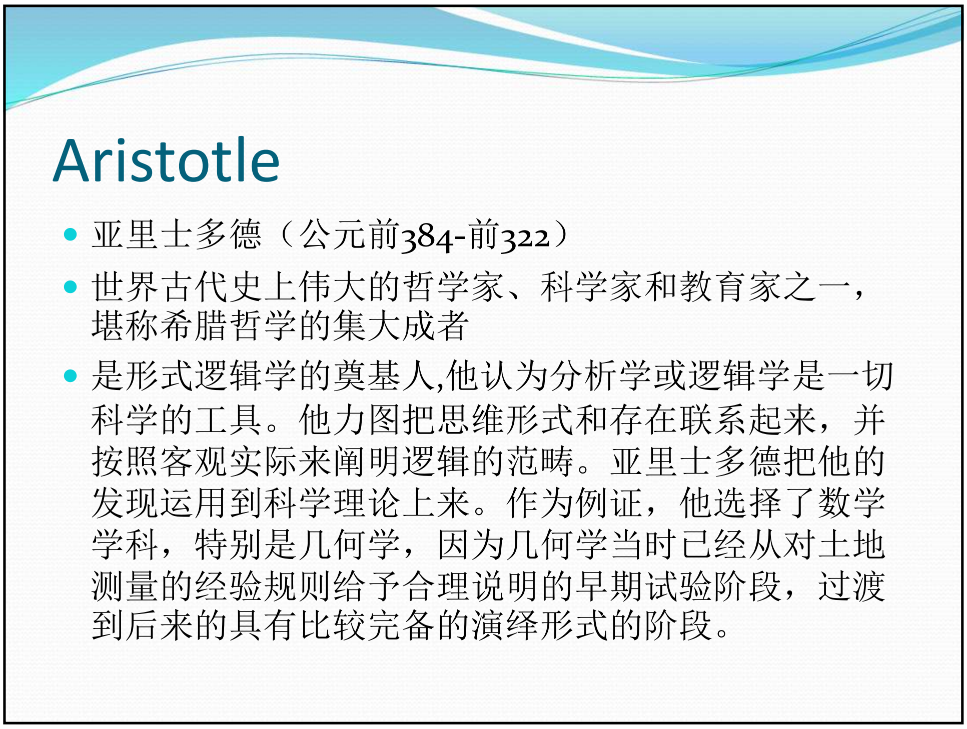
Logic

- 逻辑是探索、阐述和确立有效推理原则的学科，最早由古希腊学者亚里士多德创建的。经典的逻辑表述及推理都是用自然语言来实现的。
- 符号逻辑又称数理逻辑、理论逻辑。用数学的方法研究逻辑的系统思想一般追溯到莱布尼茨，他认为经典的传统逻辑必须改造和发展，使之更为精确和便于演算。后人基本是沿着莱布尼茨的思想进行工作的。 简而言之，数理逻辑就是精确化、数学化的形式逻辑。它是现代计算机技术的基础。新的时代将是数学大发展的时代，而数理逻辑在其中将会起到很关键的作用。用数学的方法研究关于推理、证明等问题的学科就叫做数理逻辑。也叫做符号逻辑。

Kenneth H. Rosen, *Discrete Mathematics and its Applications*, 7e

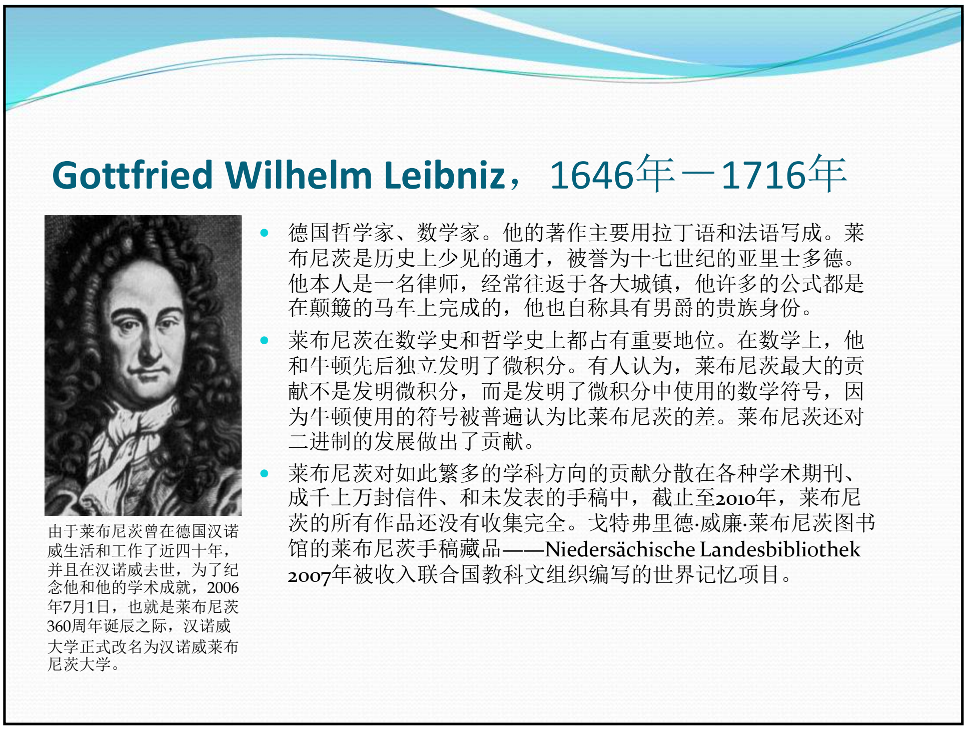
Aristotle

© National Library of Medicine



Aristotle

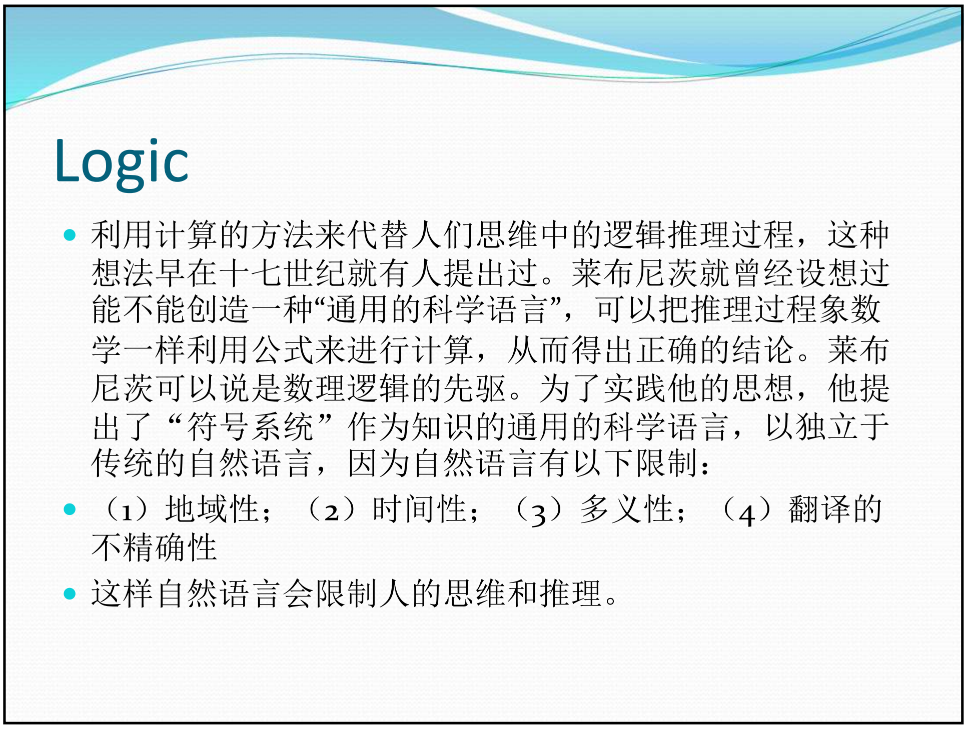
- 亚里士多德（公元前384-前322）
- 世界古代史上伟大的哲学家、科学家和教育家之一，堪称希腊哲学的集大成者
- 是形式逻辑学的奠基人,他认为分析学或逻辑学是一切科学的工具。他力图把思维形式和存在联系起来，并按照客观实际来阐明逻辑的范畴。亚里士多德把他的发现运用到科学理论上来。作为例证，他选择了数学学科，特别是几何学，因为几何学当时已经从对土地测量的经验规则给予合理说明的早期试验阶段，过渡到后来的具有比较完备的演绎形式的阶段。



Gottfried Wilhelm Leibniz，1646年－1716年

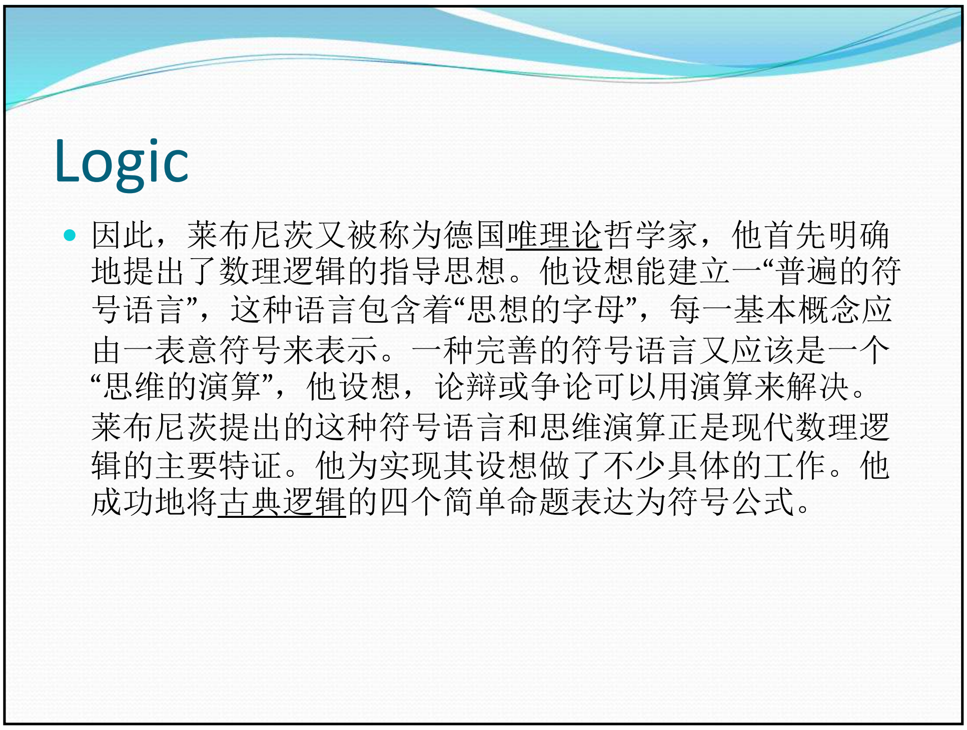
由于莱布尼茨曾在德国汉诺威生活和工作了近四十年，并且在汉诺威去世，为了纪念他和他的学术成就，2006年7月1日，也就是莱布尼茨360周年诞辰之际，汉诺威大学正式改名为汉诺威莱布尼茨大学。

- 德国哲学家、数学家。他的著作主要用拉丁语和法语写成。莱布尼茨是历史上少见的通才，被誉为十七世纪的亚里士多德。他本人是一名律师，经常往返于各大城镇，他许多的公式都是在颠簸的马车上完成的，他也自称具有男爵的贵族身份。
- 莱布尼茨在数学史和哲学史上都占有重要地位。在数学上，他和牛顿先后独立发明了微积分。有人认为，莱布尼茨最大的贡献不是发明微积分，而是发明了微积分中使用的数学符号，因为牛顿使用的符号被普遍认为比莱布尼茨的差。莱布尼茨还对二进制的发展做出了贡献。
- 莱布尼茨对如此繁多的学科方向的贡献分散在各种学术期刊、成千上万封信件、和未发表的手稿中，截止至2010年，莱布尼茨的所有作品还没有收集完全。戈特弗里德·威廉·莱布尼茨图书馆的莱布尼茨手稿藏品——Niedersächische Landesbibliothek 2007年被收入联合国教科文组织编写的世界记忆项目。



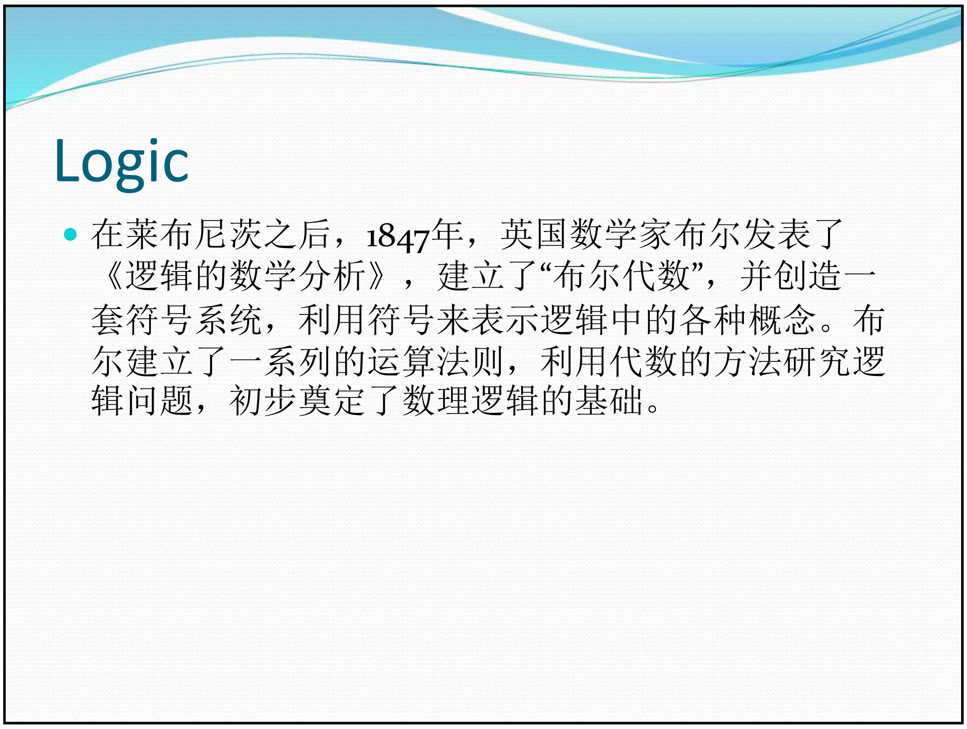
Logic

- 利用计算的方法来代替人们思维中的逻辑推理过程，这种想法早在十七世纪就有人提出过。莱布尼茨就曾经设想过能不能创造一种“通用的科学语言”，可以把推理过程象数学一样利用公式来进行计算，从而得出正确的结论。莱布尼茨可以说是数理逻辑的先驱。为了实践他的思想，他提出了“符号系统”作为知识的通用的科学语言，以独立于传统的自然语言，因为自然语言有以下限制：
- （1）地域性；（2）时间性；（3）多义性；（4）翻译的不精确性
- 这样自然语言会限制人的思维和推理。



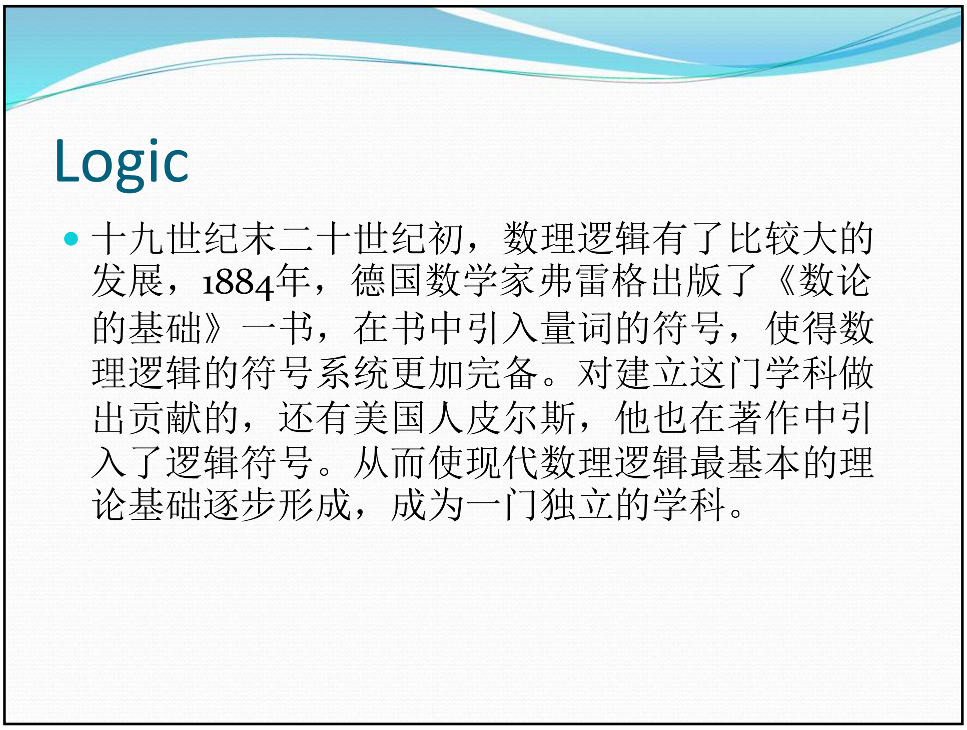
Logic

- 因此，莱布尼茨又被称为德国唯理论哲学家，他首先明确地提出了数理逻辑的指导思想。他设想能建立一“普遍的符号语言”，这种语言包含着“思想的字母”，每一基本概念应由一表意符号来表示。一种完善的符号语言又应该是一个“思维的演算”，他设想，论辩或争论可以用演算来解决。莱布尼茨提出的这种符号语言和思维演算正是现代数理逻辑的主要特征。他为实现其设想做了不少具体的工作。他成功地将古典逻辑的四个简单命题表达为符号公式。



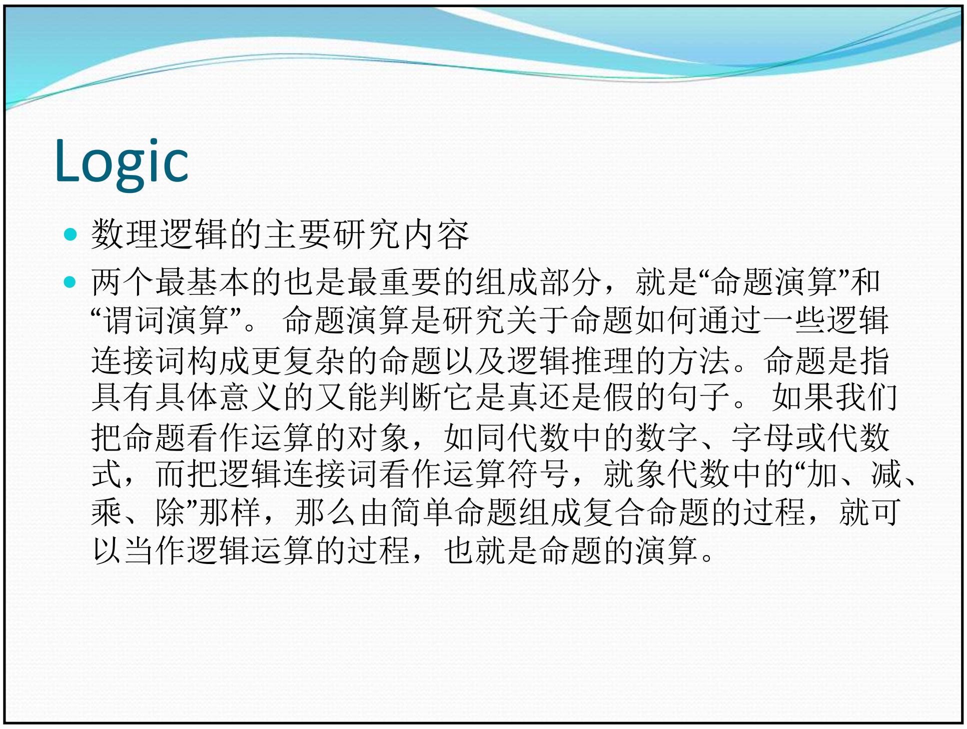
Logic

- 在莱布尼茨之后，1847年，英国数学家布尔发表了《逻辑的数学分析》，建立了“布尔代数”，并创造一套符号系统，利用符号来表示逻辑中的各种概念。布尔建立了一系列的运算法则，利用代数的方法研究逻辑问题，初步奠定了数理逻辑的基础。



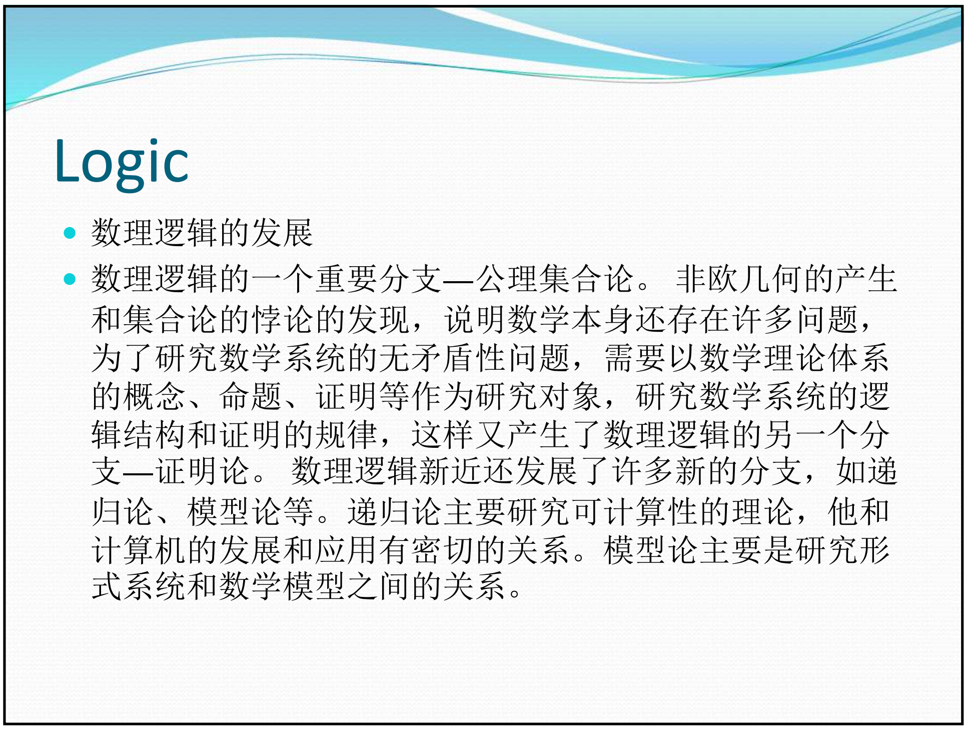
Logic

- 十九世纪末二十世纪初，数理逻辑有了比较大的发展，1884年，德国数学家弗雷格出版了《数论的基础》一书，在书中引入量词的符号，使得数理逻辑的符号系统更加完备。对建立这门学科做出贡献的，还有美国人皮尔斯，他也在著作中引入了逻辑符号。从而使现代数理逻辑最基本的理论基础逐步形成，成为一门独立的学科。



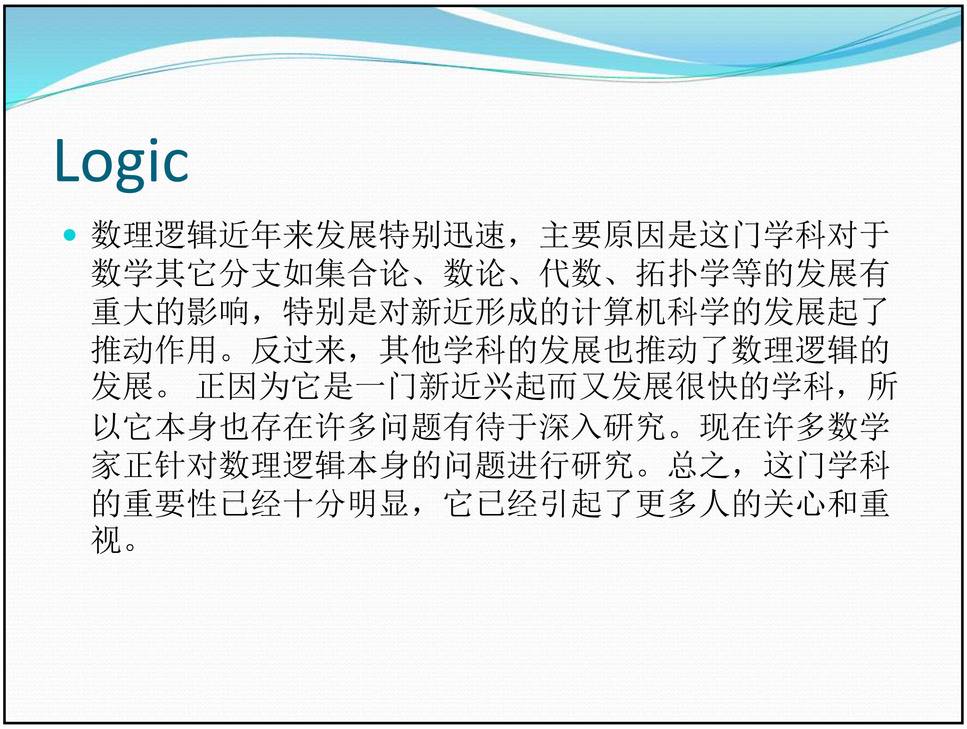
Logic

- 数理逻辑的主要研究内容
- 两个最基本的也是最重要的组成部分，就是“命题演算”和“谓词演算”。 命题演算是研究关于命题如何通过一些逻辑连接词构成更复杂的命题以及逻辑推理的方法。命题是指具有具体意义的又能判断它是真还是假的句子。 如果我们把命题看作运算的对象，如同代数中的数字、字母或代数式，而把逻辑连接词看作运算符号，就象代数中的“加、减、乘、除”那样，那么由简单命题组成复合命题的过程，就可以当作逻辑运算的过程，也就是命题的演算。



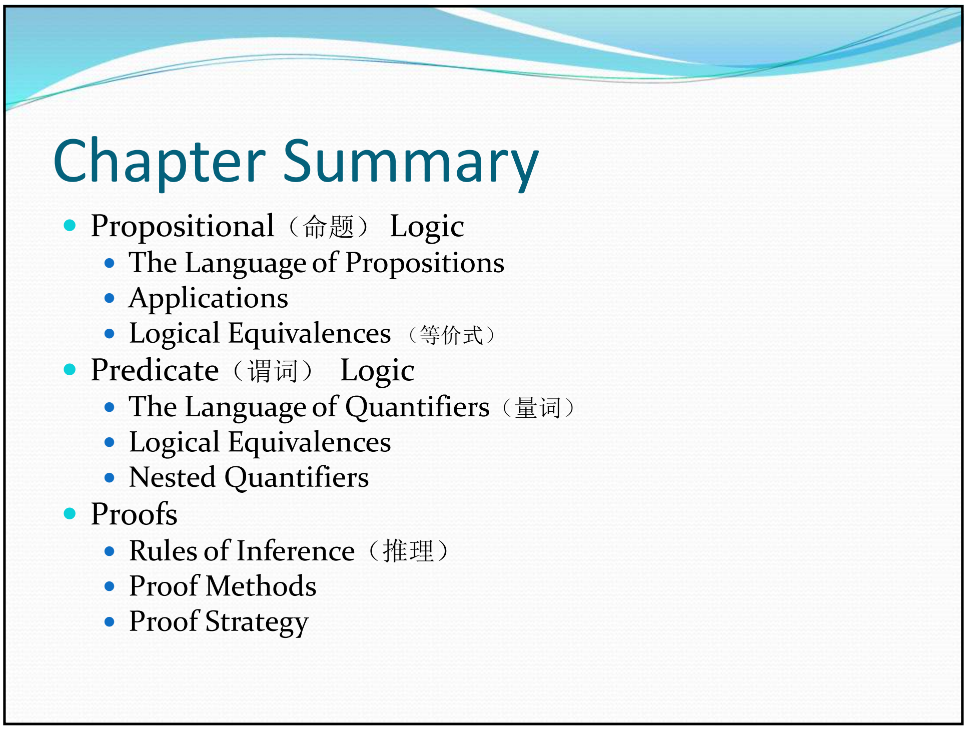
Logic

- 数理逻辑的发展
- 数理逻辑的一个重要分支—公理集合论。 非欧几何的产生和集合论的悖论的发现，说明数学本身还存在许多问题，为了研究数学系统的无矛盾性问题，需要以数学理论体系的概念、命题、证明等作为研究对象，研究数学系统的逻辑结构和证明的规律，这样又产生了数理逻辑的另一个分支—证明论。 数理逻辑新近还发展了许多新的分支，如递归论、模型论等。递归论主要研究可计算性的理论，他和计算机的发展和应用有密切的关系。模型论主要是研究形式系统和数学模型之间的关系。



Logic

- 数理逻辑近年来发展特别迅速，主要原因是这门学科对于数学其它分支如集合论、数论、代数、拓扑学等的发展有重大的影响，特别是对新近形成的计算机科学的发展起了推动作用。反过来，其他学科的发展也推动了数理逻辑的发展。 正因为它是一门新近兴起而又发展很快的学科，所以它本身也存在许多问题有待于深入研究。现在许多数学家正针对数理逻辑本身的问题进行研究。总之，这门学科的重要性已经十分明显，它已经引起了更多人的关心和重视。

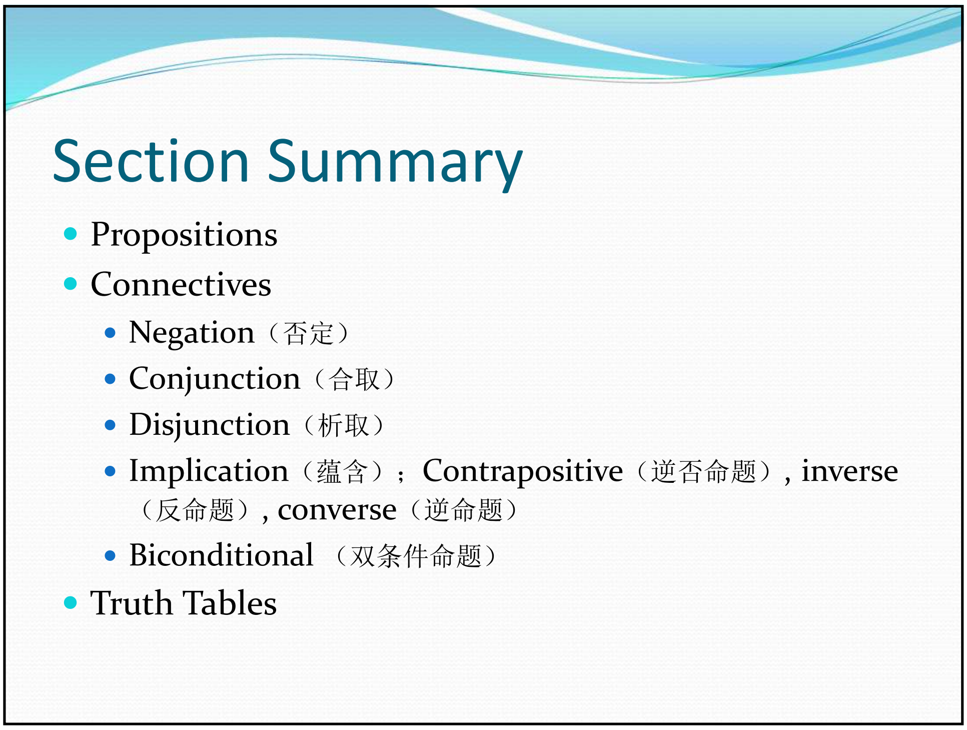


Chapter Summary

- Propositional（命题） Logic
  - The Language of Propositions
  - Applications
  - Logical Equivalences （等价式）
- Predicate（谓词） Logic
  - The Language of Quantifiers（量词）
  - Logical Equivalences
  - Nested Quantifiers
- Proofs
  - Rules of Inference（推理）
  - Proof Methods
  - Proof Strategy

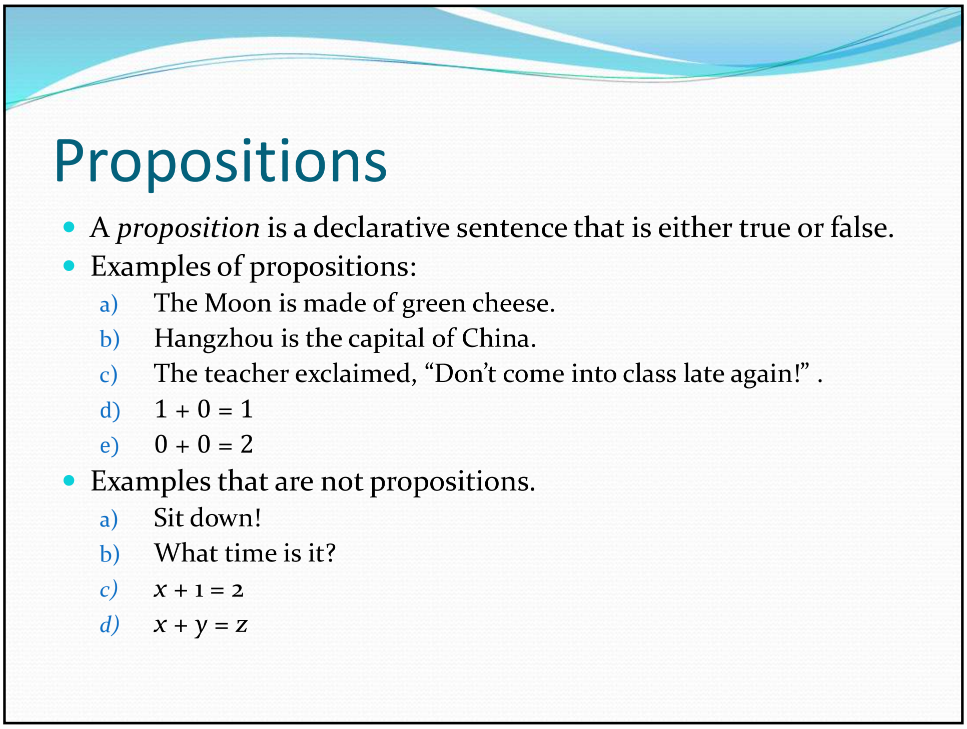
Propositional Logic

Section 1.1



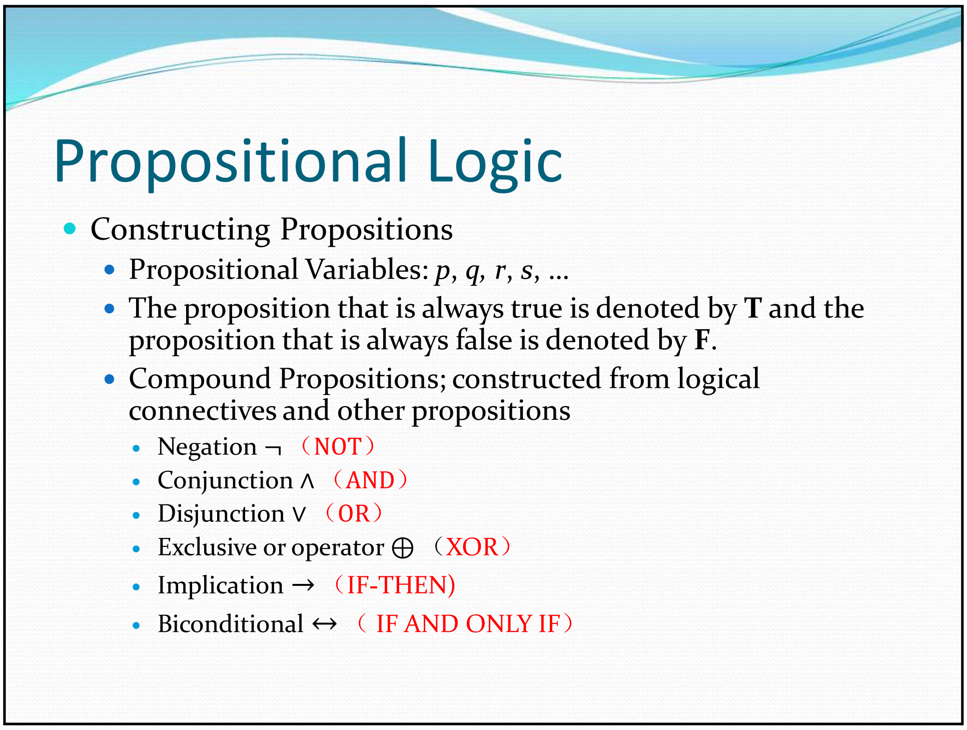
Section Summary

- Propositions
- Connectives
  - Negation（否定）
  - Conjunction（合取）
  - Disjunction（析取）
  - Implication（蕴含）；Contrapositive（逆否命题）, inverse（反命题）, converse（逆命题）
  - Biconditional （双条件命题）
- Truth Tables



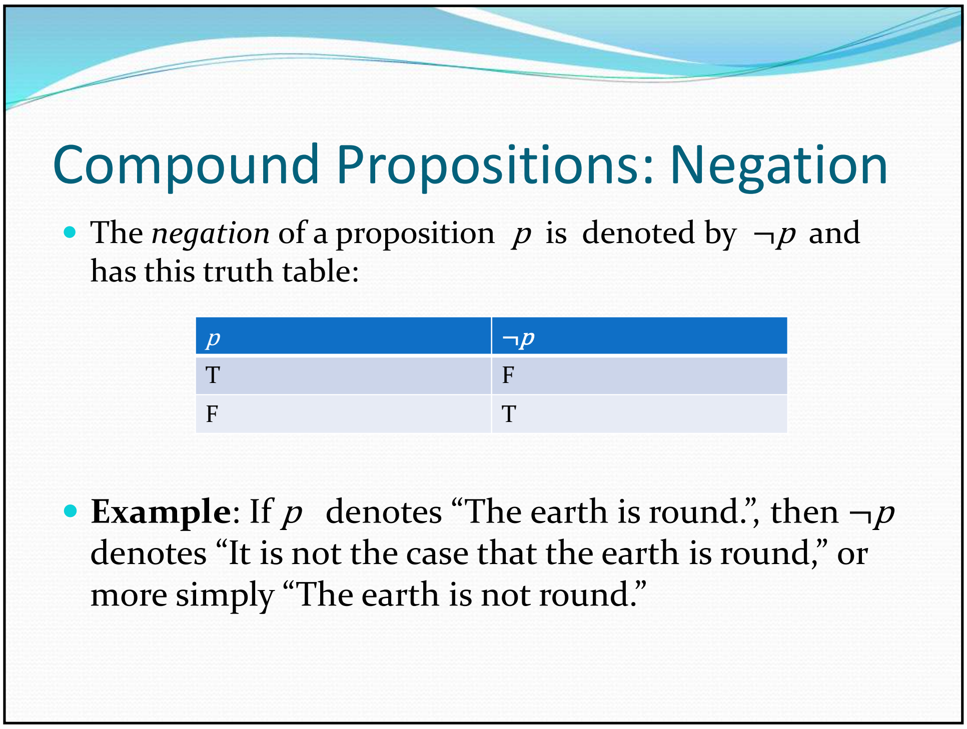
Propositions

- A *proposition* is a declarative sentence that is either true or false.
- Examples of propositions:
  a) The Moon is made of green cheese.
  b) Hangzhou is the capital of China.
  c) The teacher exclaimed, “Don’t come into class late again!” .
  d) $1 + 0 = 1$
  e) $0 + 0 = 2$
- Examples that are not propositions.
  a) Sit down!
  b) What time is it?
  c) $x + 1 = 2$
  d) $x + y = z$



Propositional Logic

- 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**.
  - Compound Propositions; constructed from logical connectives and other propositions
    - Negation ¬ （NOT）
    - Conjunction ∧ （AND）
    - Disjunction ∨ （OR）
    - Exclusive or operator ⊕ （XOR）
    - Implication → （IF-THEN)
    - Biconditional ↔ （ IF AND ONLY IF）

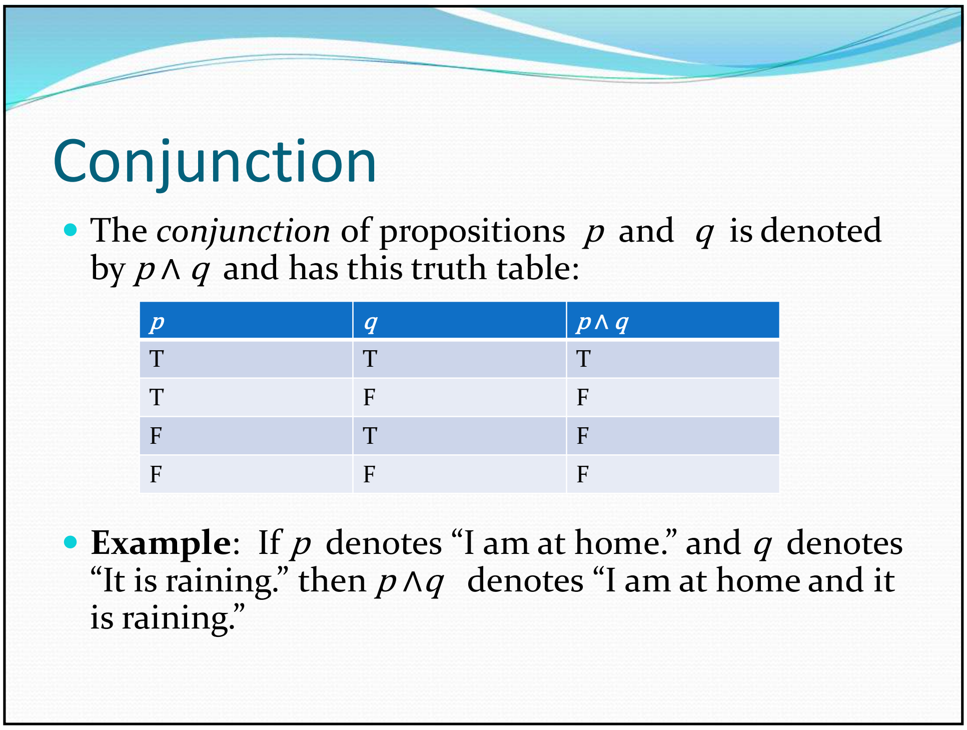


Compound Propositions: Negation

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

p	$\neg p$
T	F
F	T

- **Example**: If p denotes “The earth is round.”, then $\neg p$ denotes “It is not the case that the earth is round,” or more simply “The earth is not round.”

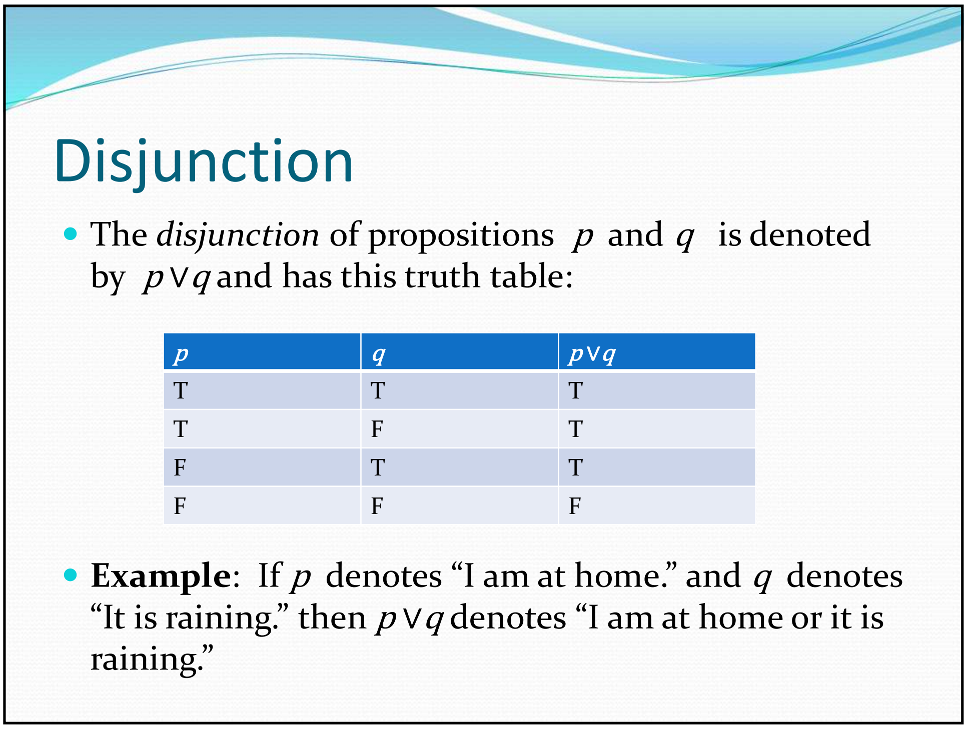


Conjunction

- 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.”

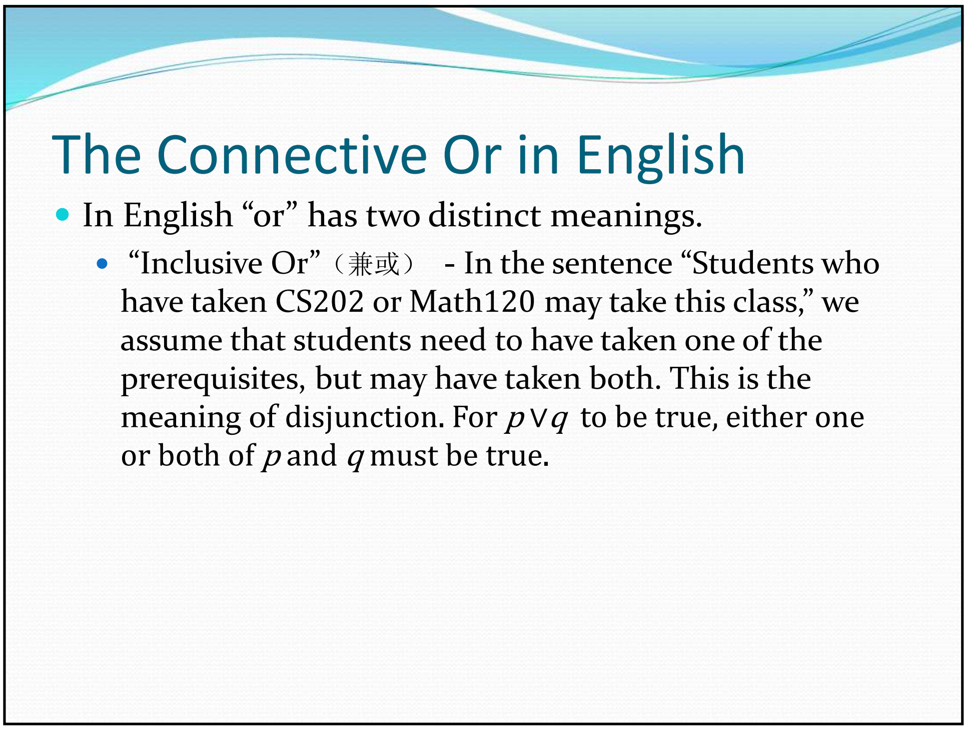


Disjunction

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

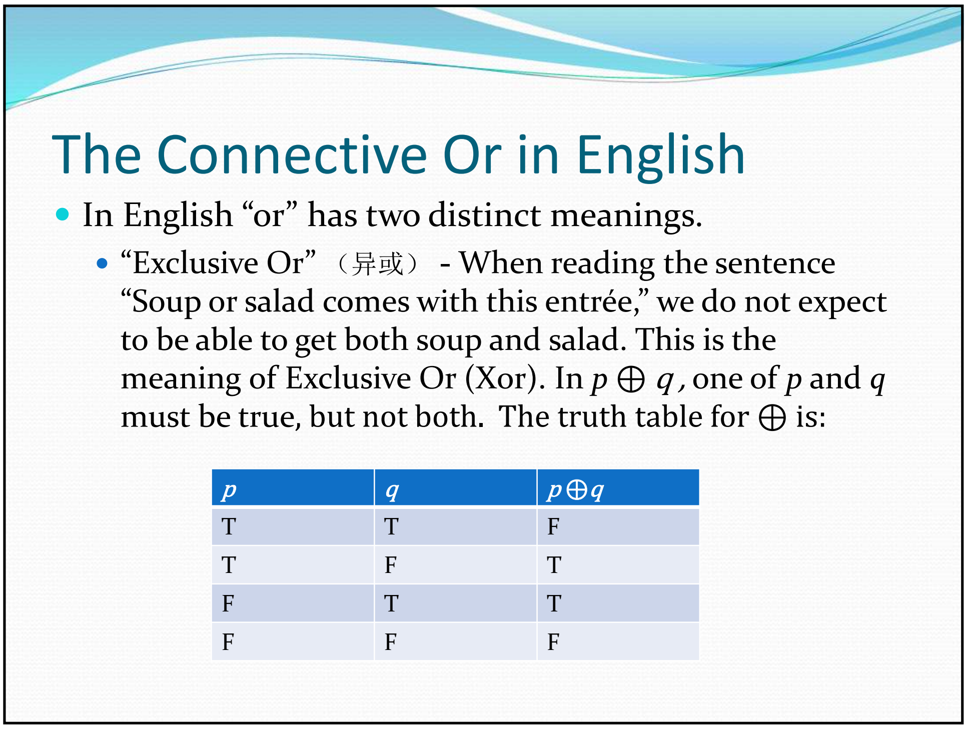
p	q	$p \lor q$
T	T	T
T	F	T
F	T	T
F	F	F

- **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 Connective Or in English

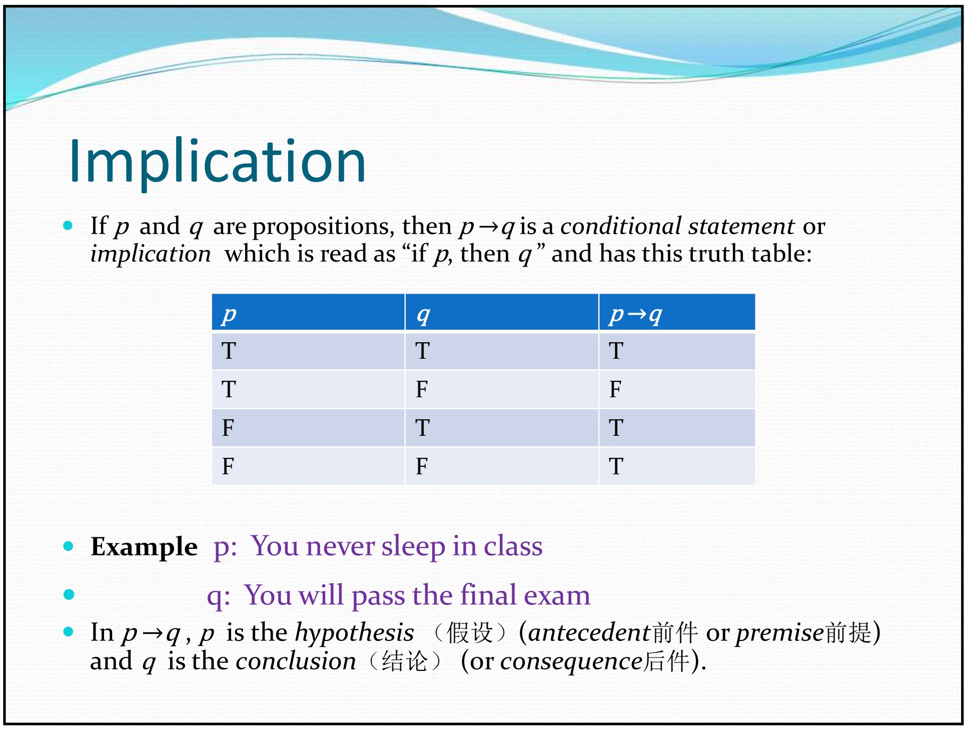
- In English “or” has two distinct meanings.
  - “Inclusive Or”（兼或） - In the sentence “Students who have taken CS202 or Math120 may take this class,” we assume that students need to have taken one of the prerequisites, but may have taken both. This is the meaning of disjunction. For $p \lor q$ to be true, either one or both of p and q must be true.



The Connective Or in English

- In English “or” has two distinct meanings.
  - “Exclusive Or”（异或） - 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 Or (Xor). In $p \oplus q$, one of p and q must be true, but not both. The truth table for $\oplus$ is:

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

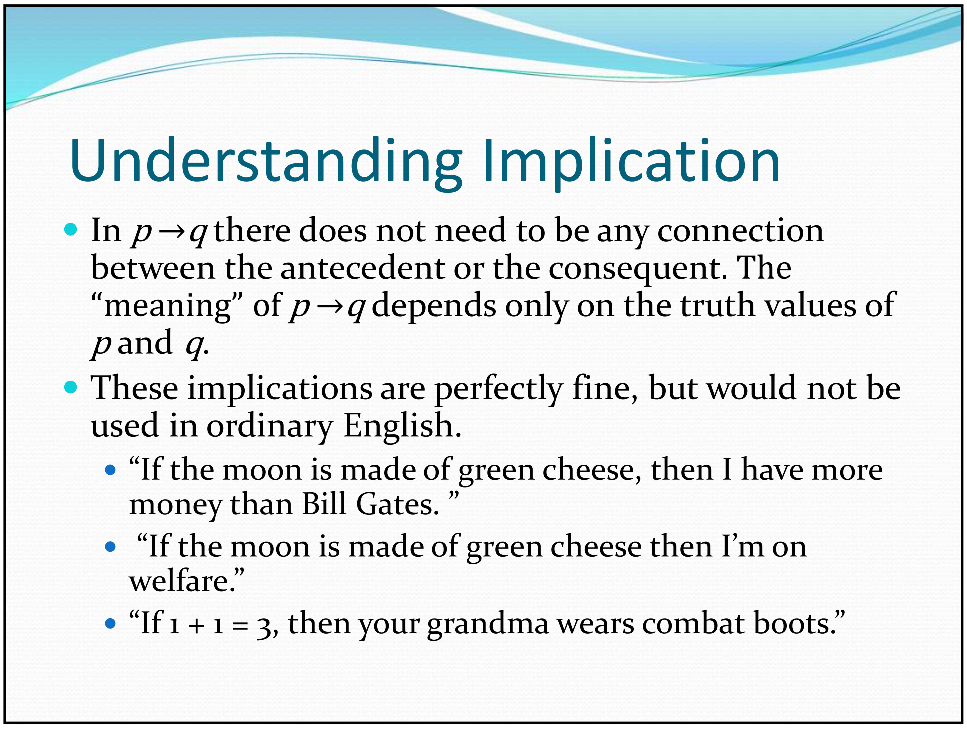


Implication

- 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:

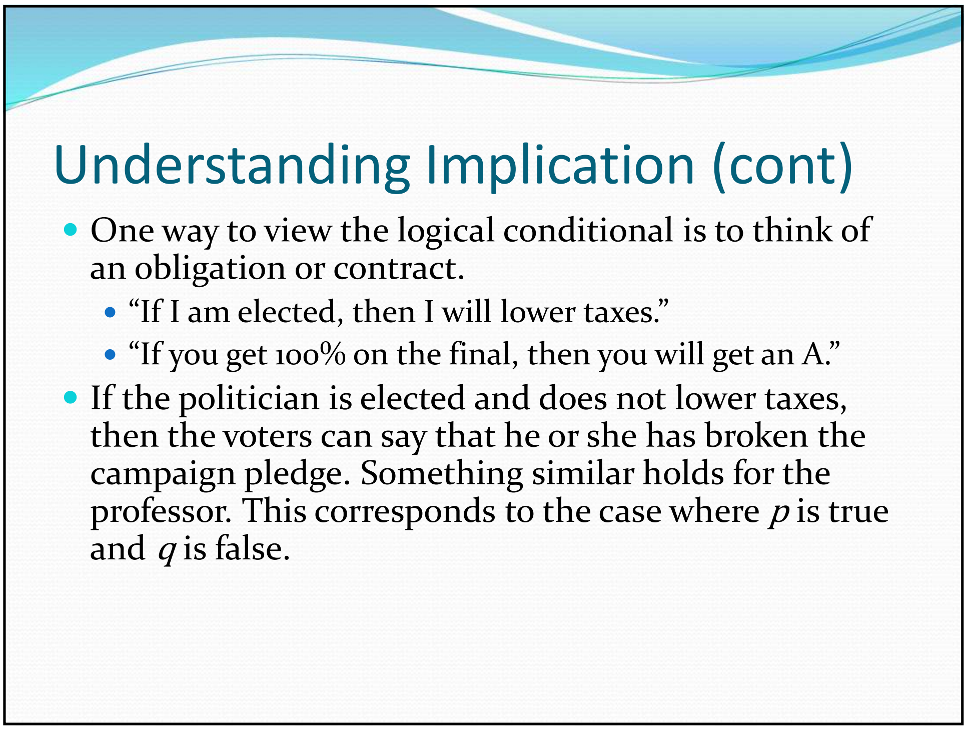
p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

- **Example** p: You never sleep in class
- q: You will pass the final exam
- In $p \rightarrow q$, p is the *hypothesis* （假设）(*antecedent*前件 or *premise*前提) and q is the *conclusion*（结论） (or *consequence*后件).



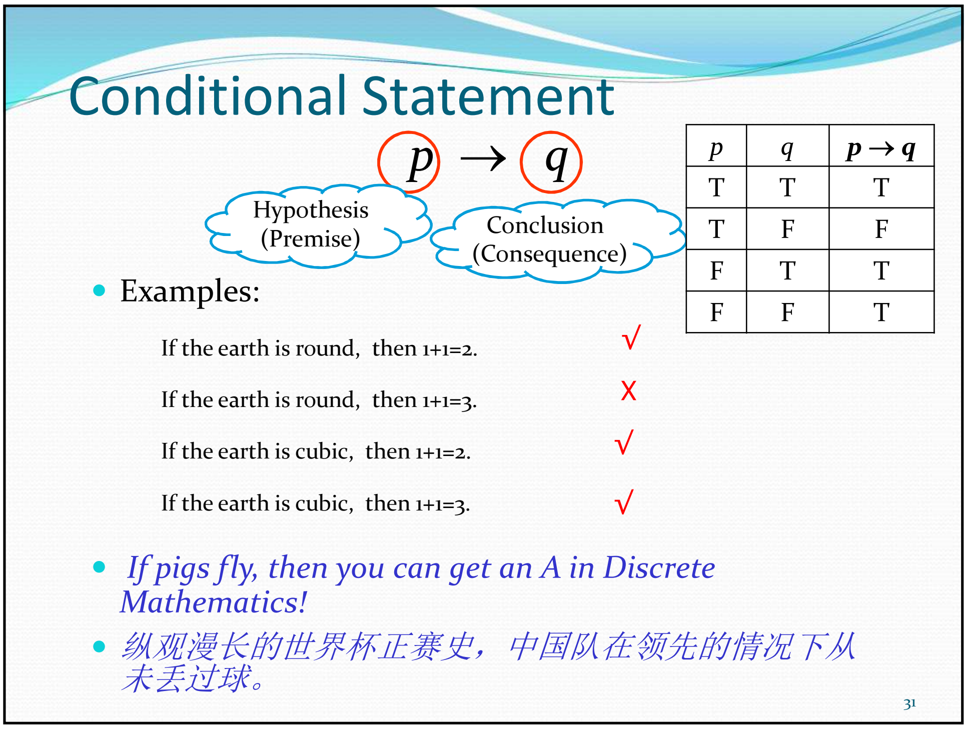
Understanding Implication

- In $p \rightarrow q$ there does not need to be any connection between the antecedent or the consequent. The "meaning" of $p \rightarrow q$ 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 Implication (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.



Conditional Statement

$p \to q$

Hypothesis (Premise)

Conclusion (Consequence)

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

- Examples:

If the earth is round, then 1+1=2. √

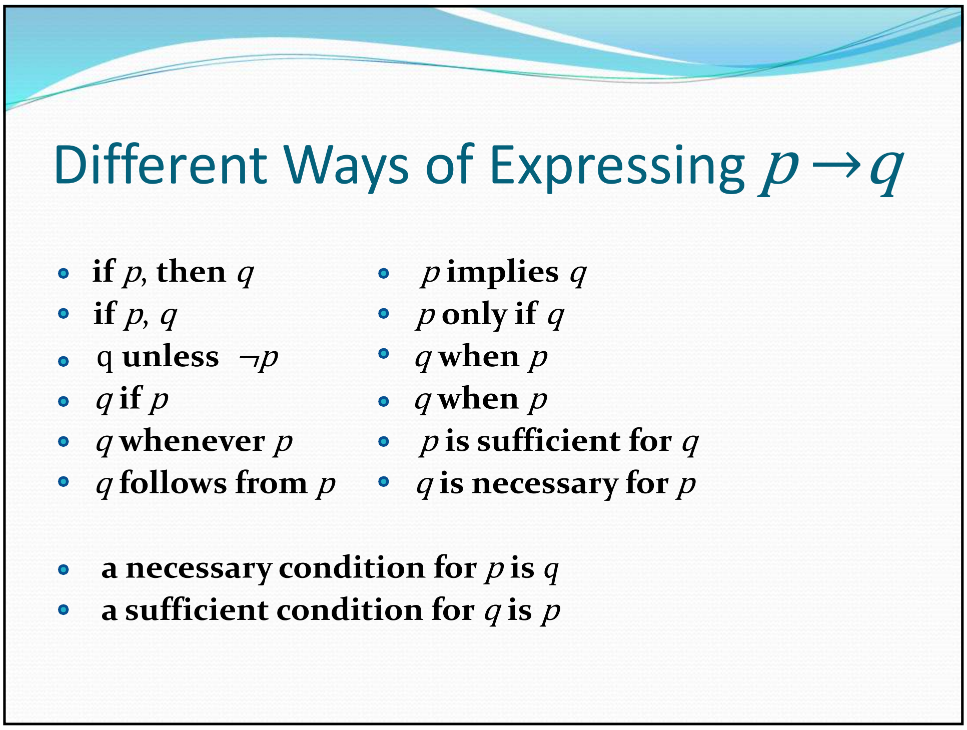
If the earth is round, then 1+1=3. X

If the earth is cubic, then 1+1=2. √

If the earth is cubic, then 1+1=3. √

- *If pigs fly, then you can get an A in Discrete Mathematics!*
- *纵观漫长的世界杯正赛史，中国队在领先的情况下从未丢过球。*

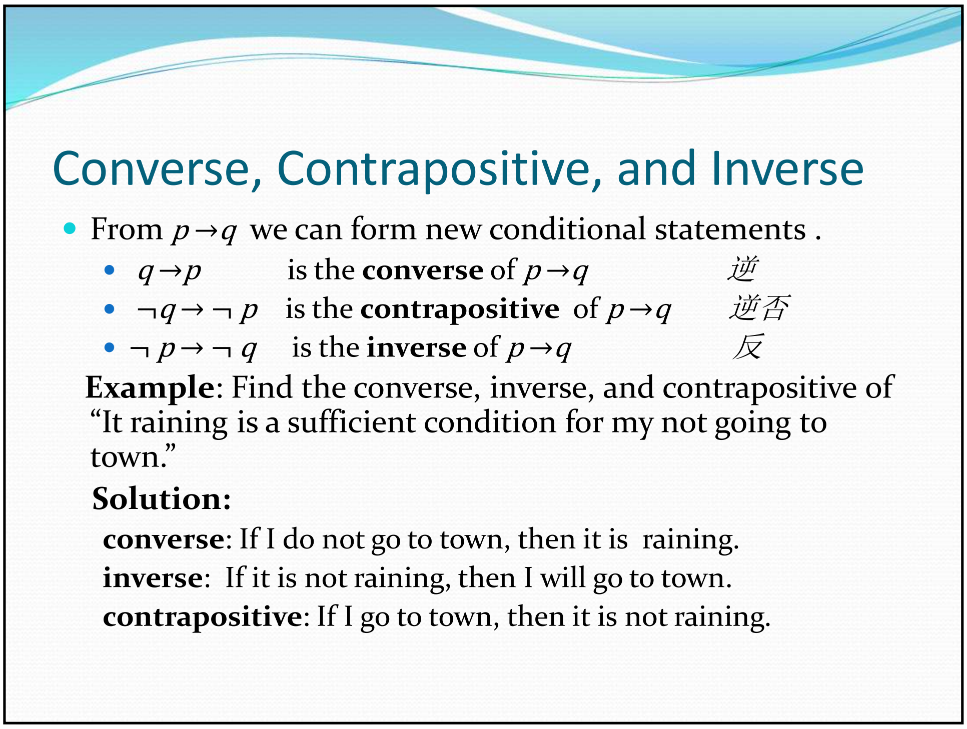
31



Different Ways of Expressing $p \to q$

- **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
- q **when** p
- 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



Converse, Contrapositive, and Inverse

- From $p \to q$ we can form new conditional statements .
  - $q \to p$ is the **converse** of $p \to q$ 逆
  - $\neg q \to \neg p$ is the **contrapositive** of $p \to q$ 逆否
  - $\neg p \to \neg q$ is the **inverse** of $p \to q$ 反

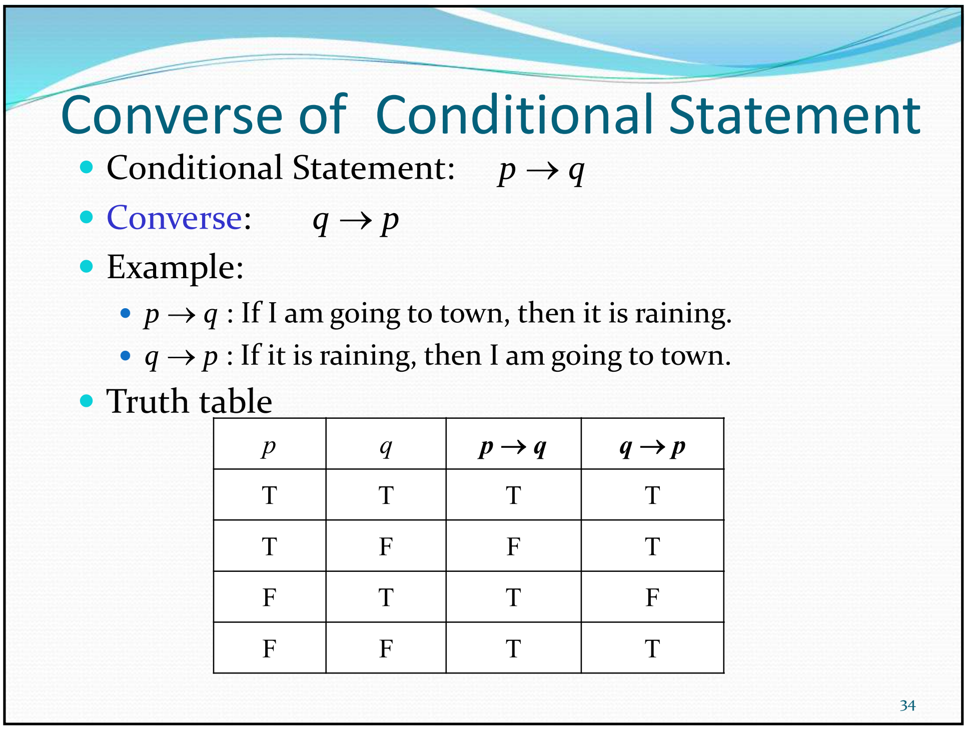
Example: Find the converse, inverse, and contrapositive of "It raining is a sufficient condition for my not going to town."

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.

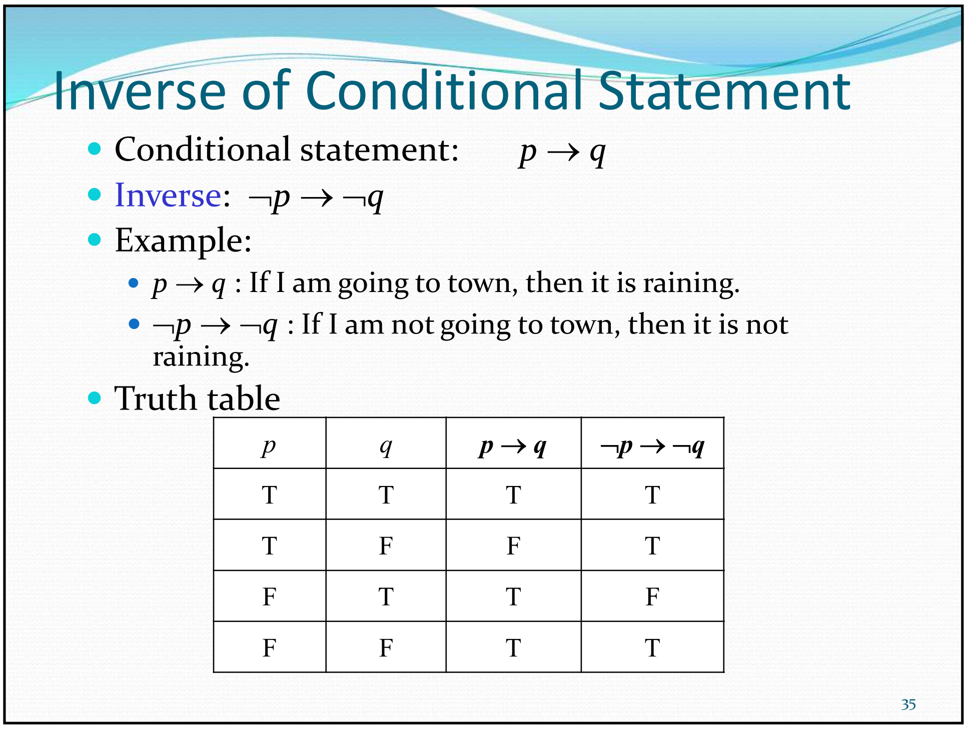
contrapositive: If I go to town, then it is not raining.



Converse of Conditional Statement

- Conditional Statement: $p \to q$
- Converse: $q \to p$
- Example:
  - $p \to q$: If I am going to town, then it is raining.
  - $q \to p$: If it is raining, then I am going to town.
- Truth table

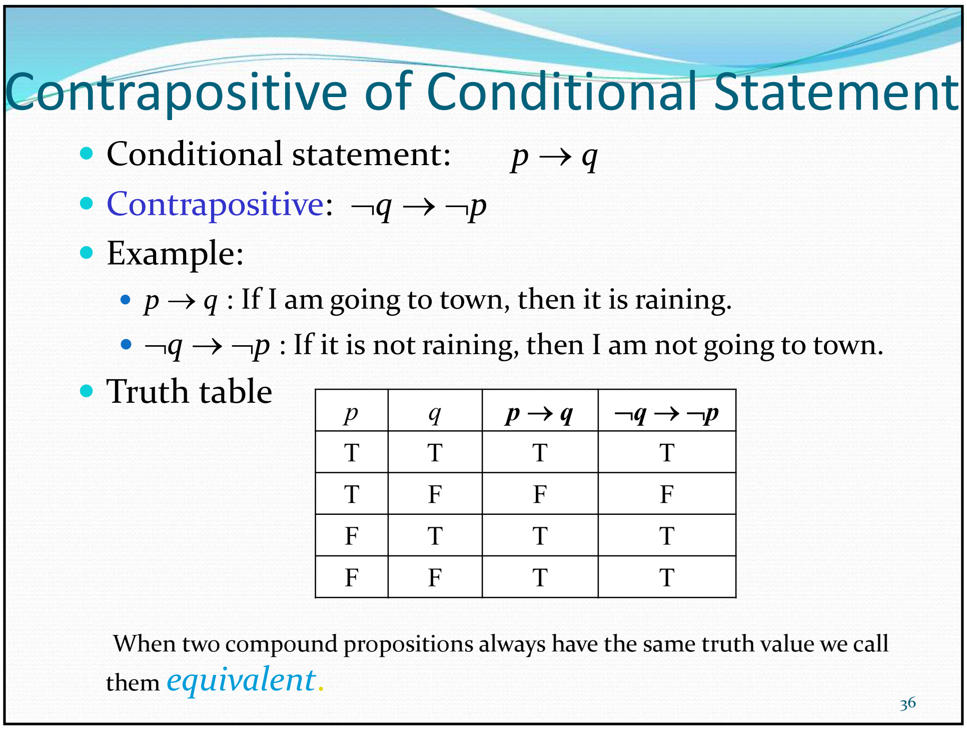
p	q	$p \to q$	$q \to p$
T	T	T	T
T	F	F	T
F	T	T	F
F	F	T	T



Inverse of Conditional Statement

- Conditional statement: $p \rightarrow q$
- Inverse: $\neg p \rightarrow \neg q$
- Example:
  - $p \rightarrow q$: If I am going to town, then it is raining.
  - $\neg p \rightarrow \neg q$: If I am not going to town, then it is not raining.
- Truth table

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

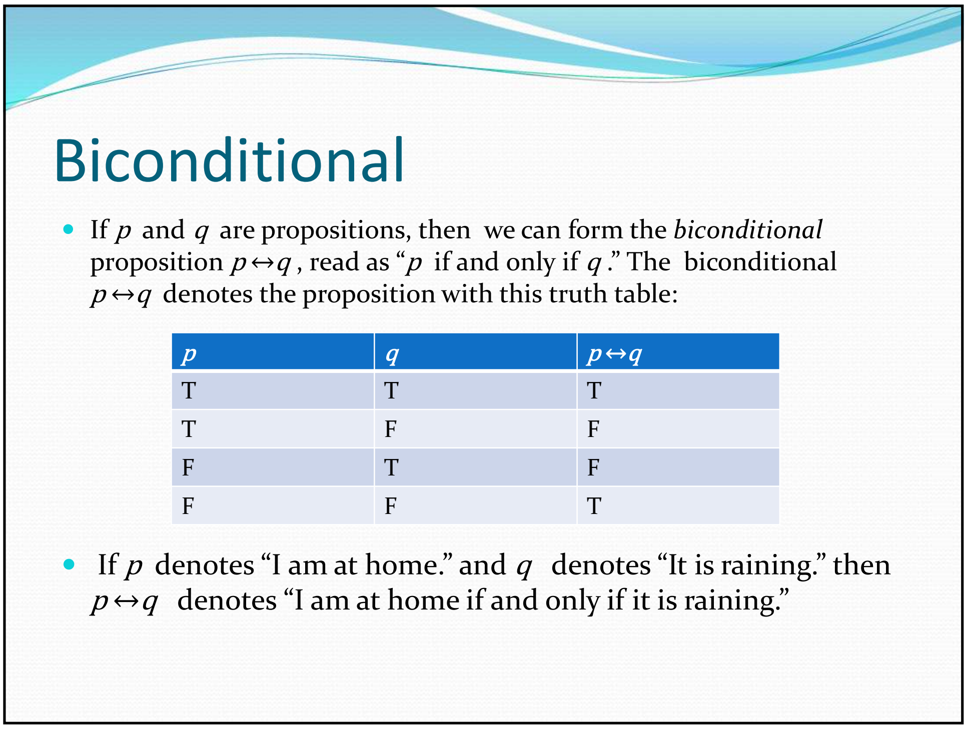


Contrapositive of Conditional Statement

- Conditional statement: $p \rightarrow q$
- Contrapositive: $\neg q \rightarrow \neg p$
- Example:
  - $p \rightarrow q$: If I am going to town, then it is raining.
  - $\neg q \rightarrow \neg p$: If it is not raining, then I am not going to town.
- Truth table

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

When two compound propositions always have the same truth value we call them *equivalent*.

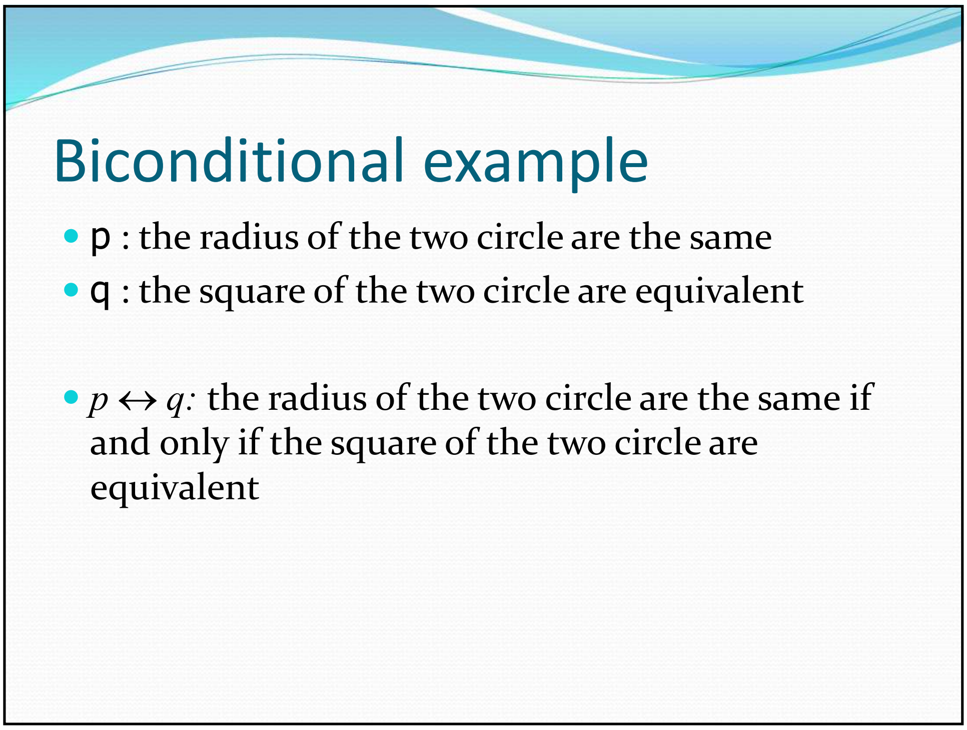


Biconditional

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

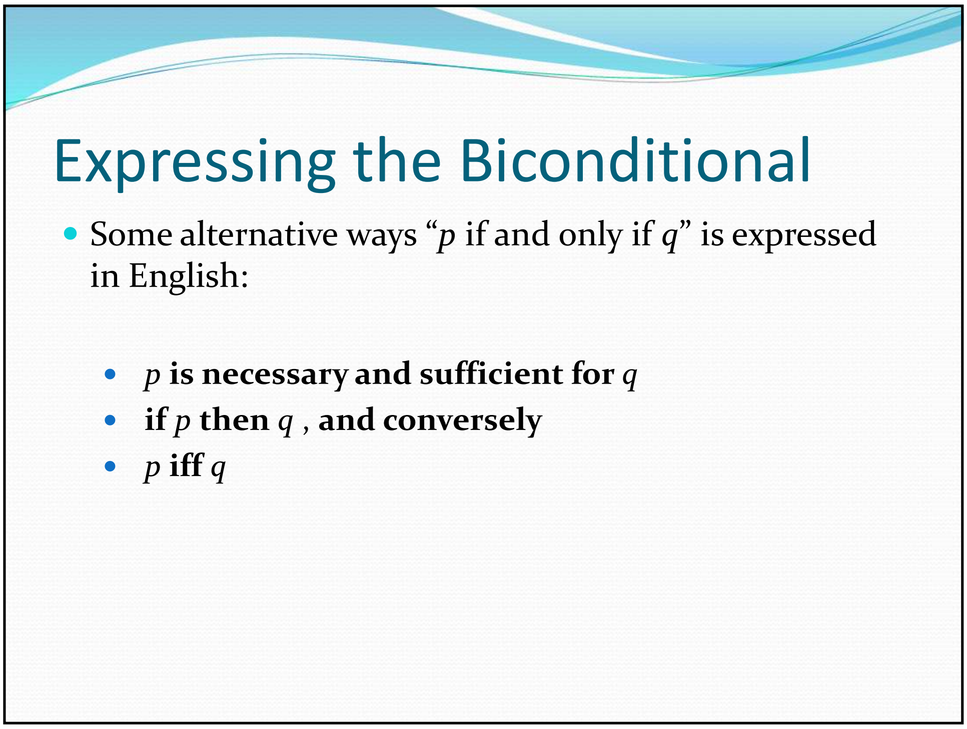
p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

- 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."



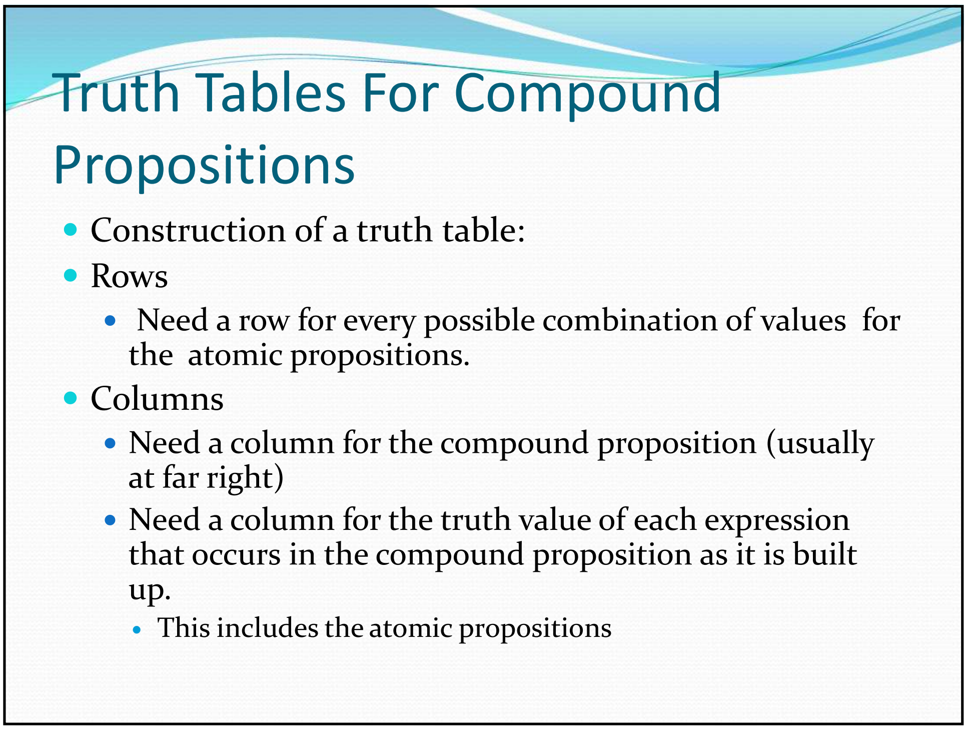
Biconditional example

- p : the radius of the two circle are the same
- q : the square of the two circle are equivalent
- $p \leftrightarrow q$: the radius of the two circle are the same if and only if the square of the two circle are equivalent



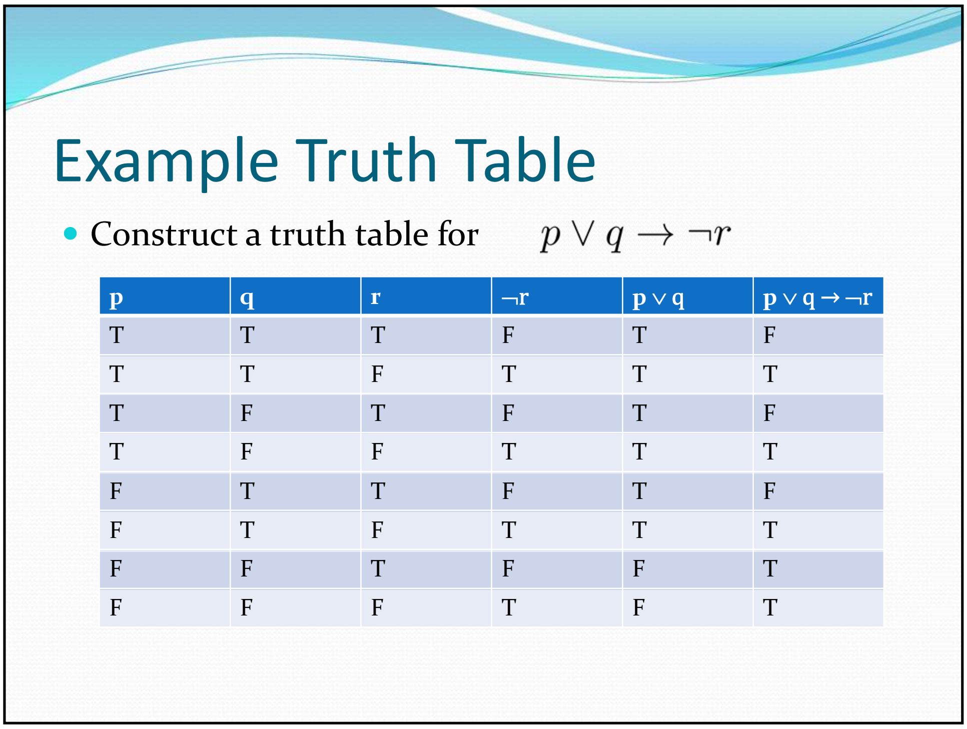
Expressing the Biconditional

- 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*



Truth Tables For Compound Propositions

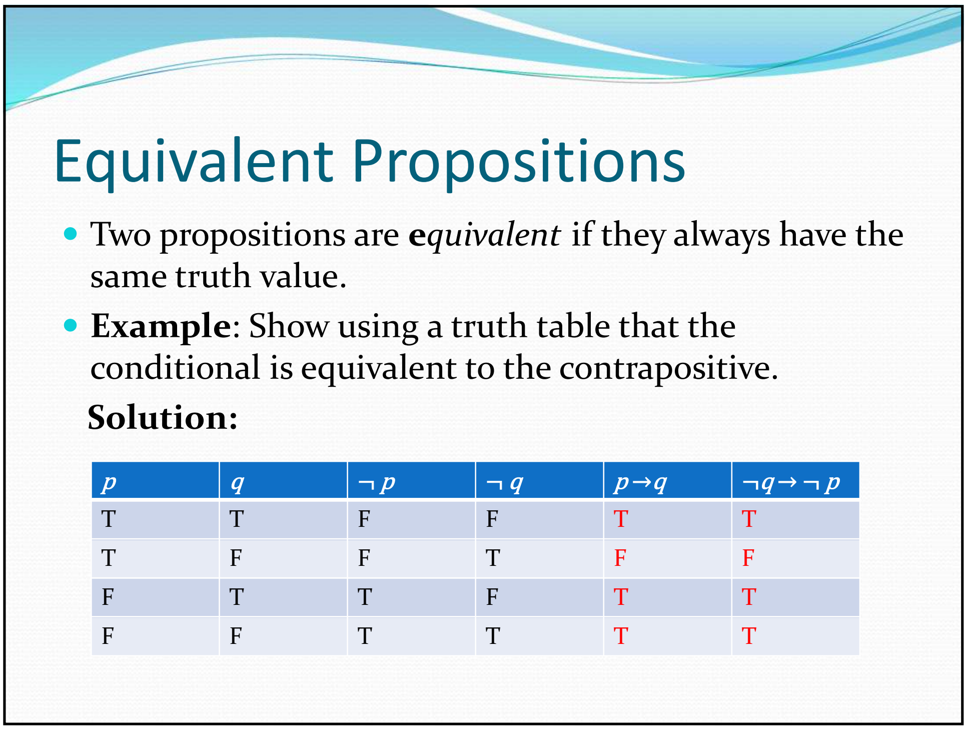
- Construction of a truth table:
- Rows
  - Need a row for every possible combination of values for the atomic propositions.
- Columns
  - Need a column for the compound proposition (usually at far right)
  - Need a column for the truth value of each expression that occurs in the compound proposition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



Example Truth Table

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

p	q	r	¬r	p ∨ q	p ∨ q → ¬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

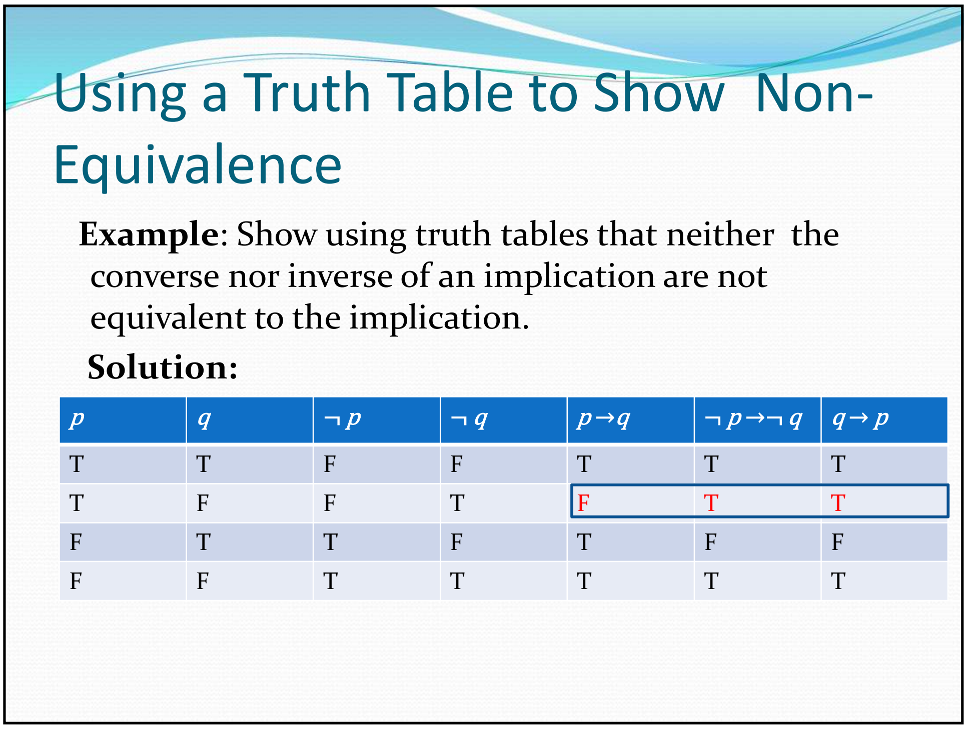


Equivalent Propositions

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

Solution:

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

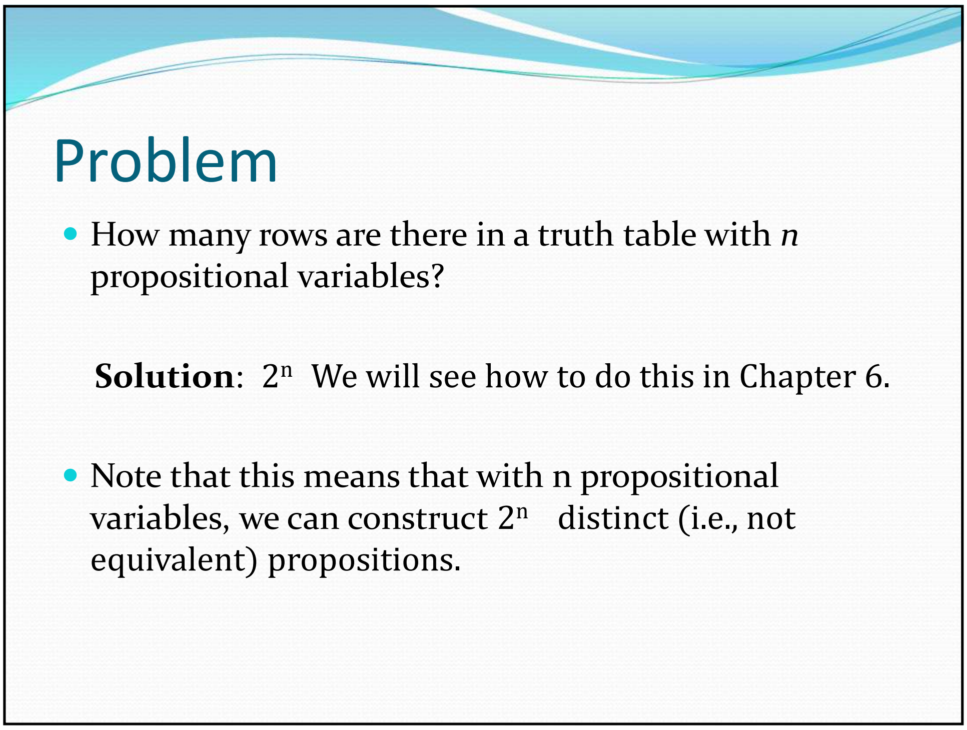


Using a Truth Table to Show Non-Equivalence

Example: Show using truth tables that neither the converse nor inverse of an implication are not equivalent to the implication.

Solution:

p	q	$\neg p$	$\neg q$	$p \rightarrow q$	$\neg p \rightarrow \neg q$	$q \rightarrow p$
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	T	T	T

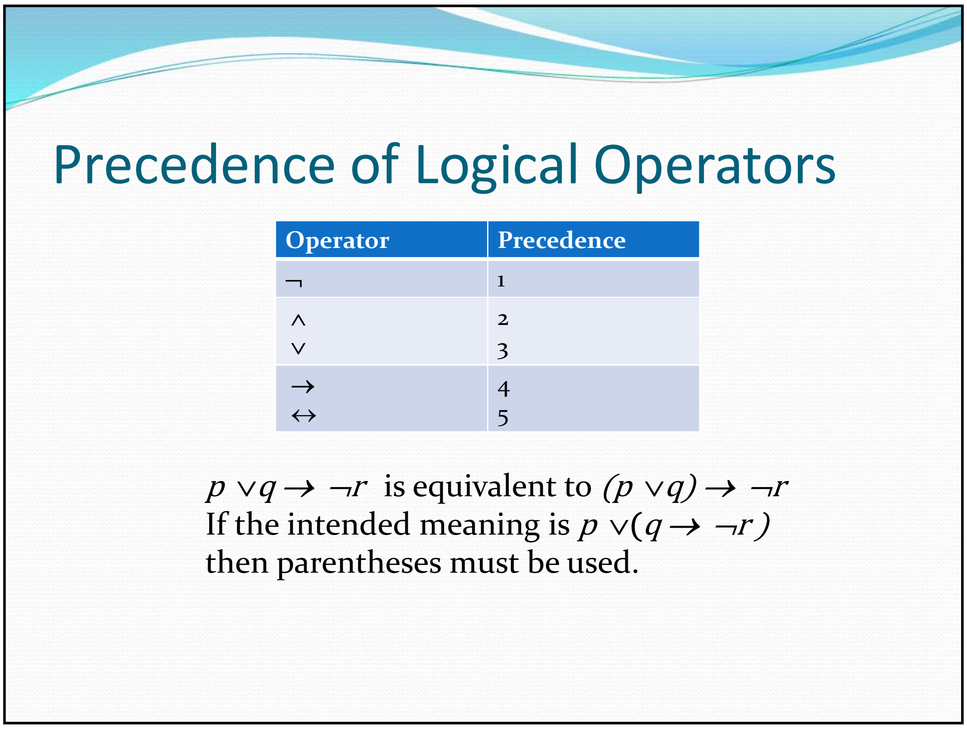


Problem

- How many rows are there in a truth table with *n* propositional variables?

Solution: 2^n We will see how to do this in Chapter 6.

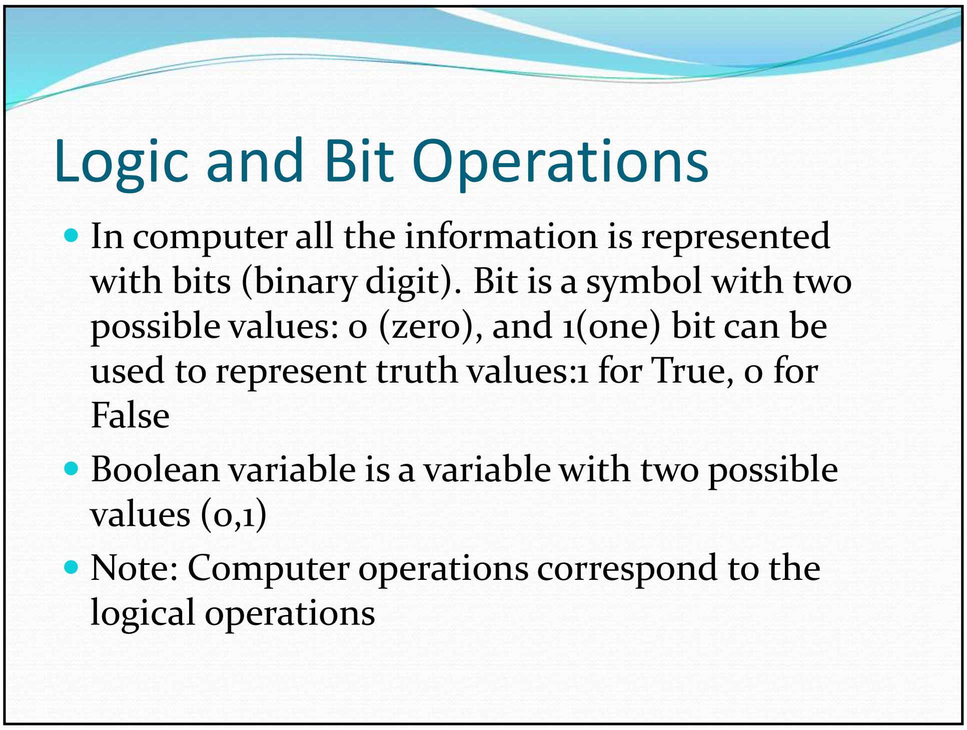
- Note that this means that with n propositional variables, we can construct 2^n distinct (i.e., not equivalent) propositions.



Precedence of Logical Operators

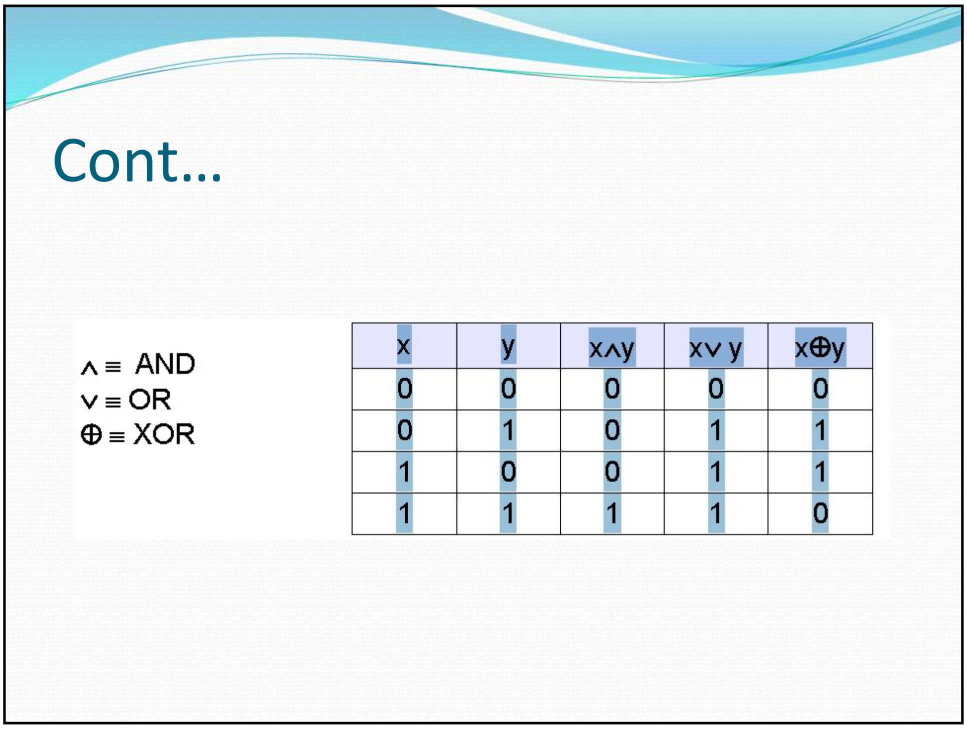
Operator	Precedence
$\neg$	1
$\wedge$	2
$\vee$	3
$\rightarrow$	4
$\leftrightarrow$	5

$p \vee q \rightarrow \neg r$ 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.



Logic and Bit Operations

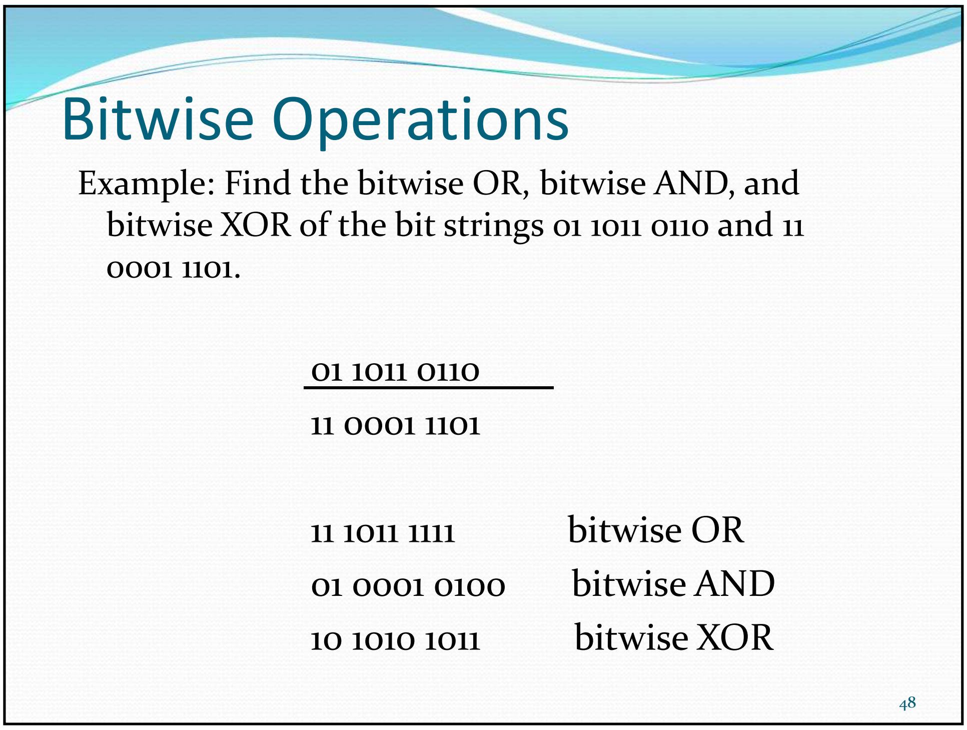
- In computer all the information is represented with bits (binary digit). Bit is a symbol with two possible values: 0 (zero), and 1(one) bit can be used to represent truth values:1 for True, 0 for False
- Boolean variable is a variable with two possible values (0,1)
- Note: Computer operations correspond to the logical operations



Cont…

∧ ≡ AND
∨ ≡ OR
⊕ ≡ XOR

x	y	x∧y	x∨ y	x⊕y
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0



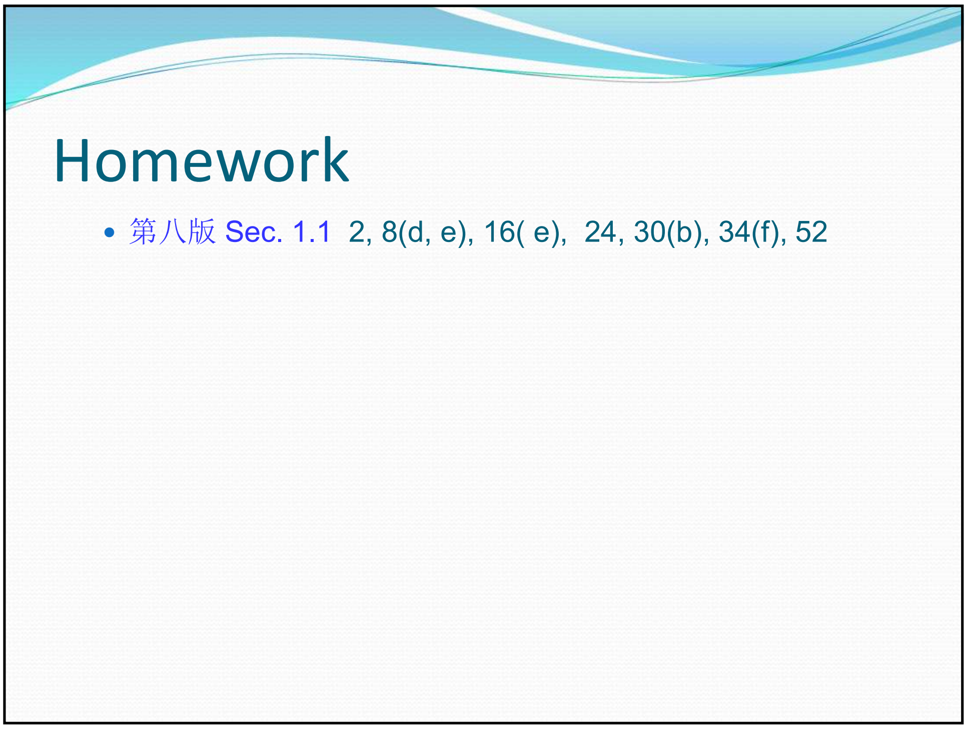
Bitwise Operations

Example: Find the bitwise OR, bitwise AND, and bitwise XOR of the bit strings 01 1011 0110 and 11 0001 1101.

01 1011 0110

11 0001 1101

11 1011 1111	bitwise OR
01 0001 0100	bitwise AND
10 1010 1011	bitwise XOR



Homework

- 第八版 Sec. 1.1 2, 8(d, e), 16(e), 24, 30(b), 34(f), 52

Applications of Propositional Logic

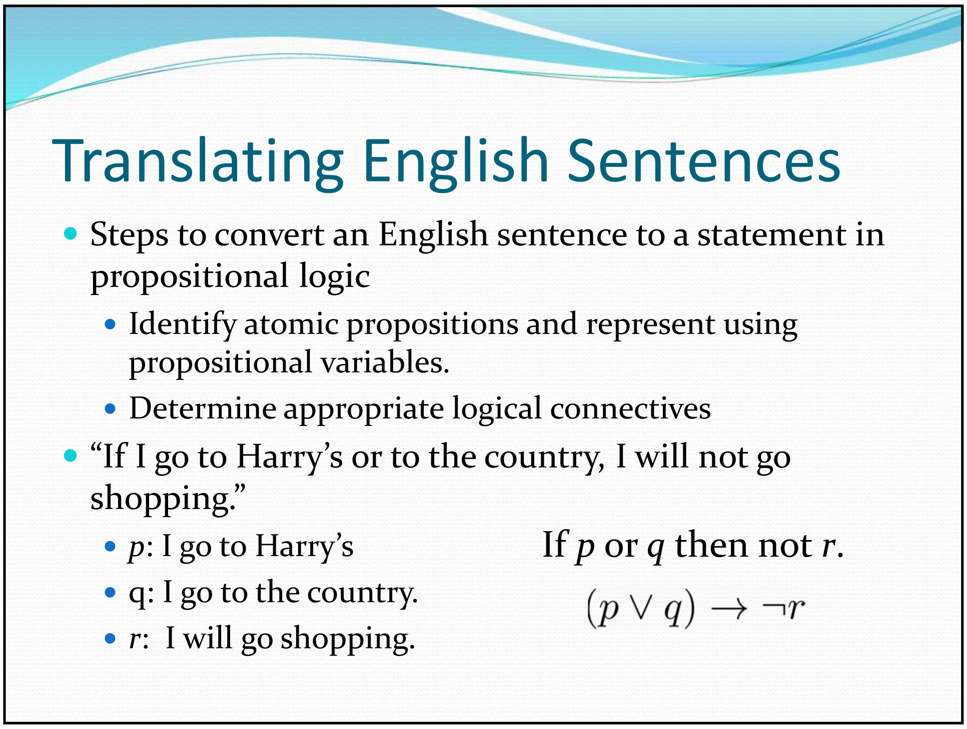
Section 1.2

Applications of Propositional Logic: Summary

- Translating English to Propositional Logic
- System Specifications
- Boolean Searching
- Logic Puzzles
- Logic Circuits
- AI Diagnosis Method (Optional)

Overview

- Logic has many important applications to mathematics, computer science, and numerous other disciplines. Statements in mathematics and the sciences and in natural language often are imprecise or ambiguous. To make such statements precise, they can be translated into the language of logic.

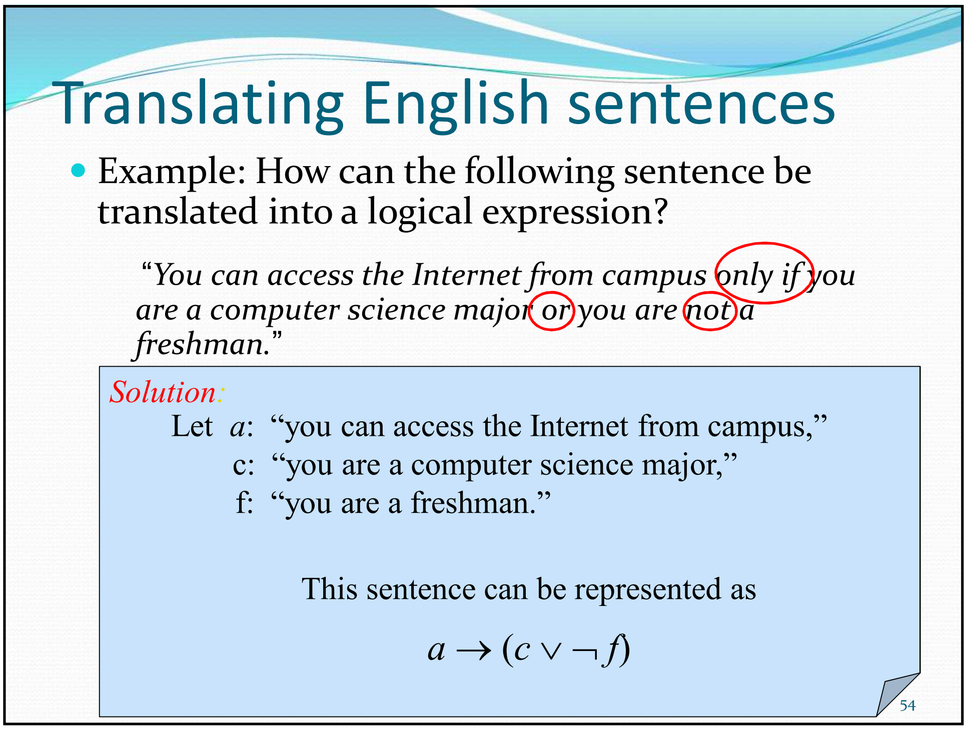


Translating English Sentences

- 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*.

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



Translating English sentences

- Example: How can the following sentence be translated into a logical expression?

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

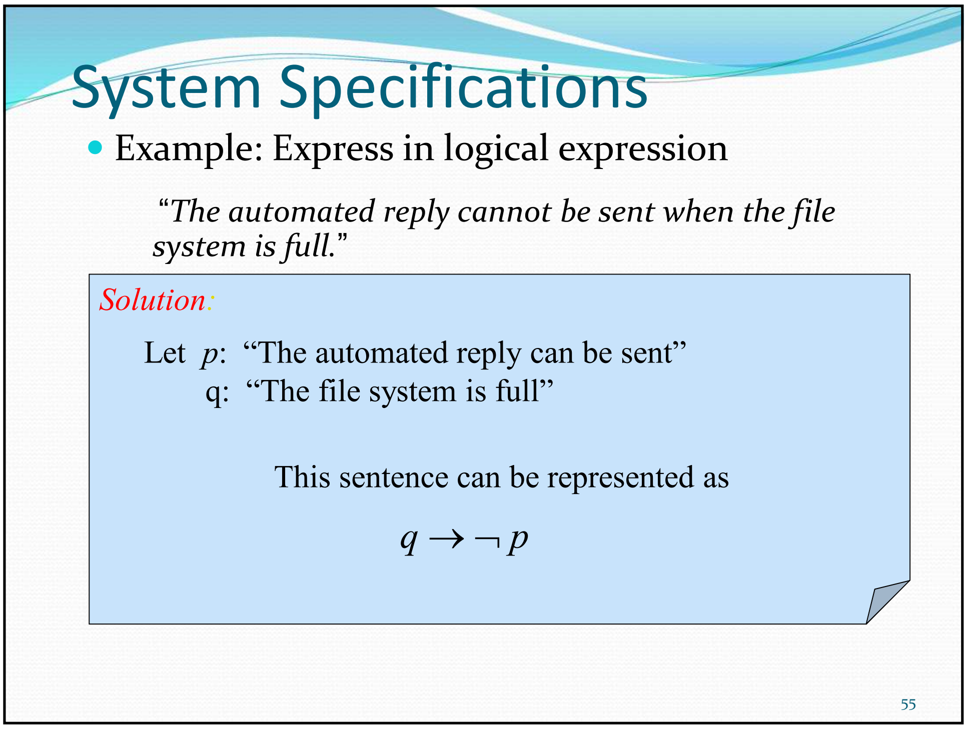
Solution:

Let *a*: "you can access the Internet from campus,"
c: "you are a computer science major,"
f: "you are a freshman."

This sentence can be represented as

$$a \rightarrow (c \lor \neg f)$$

54



System Specifications

- Example: Express in logical expression

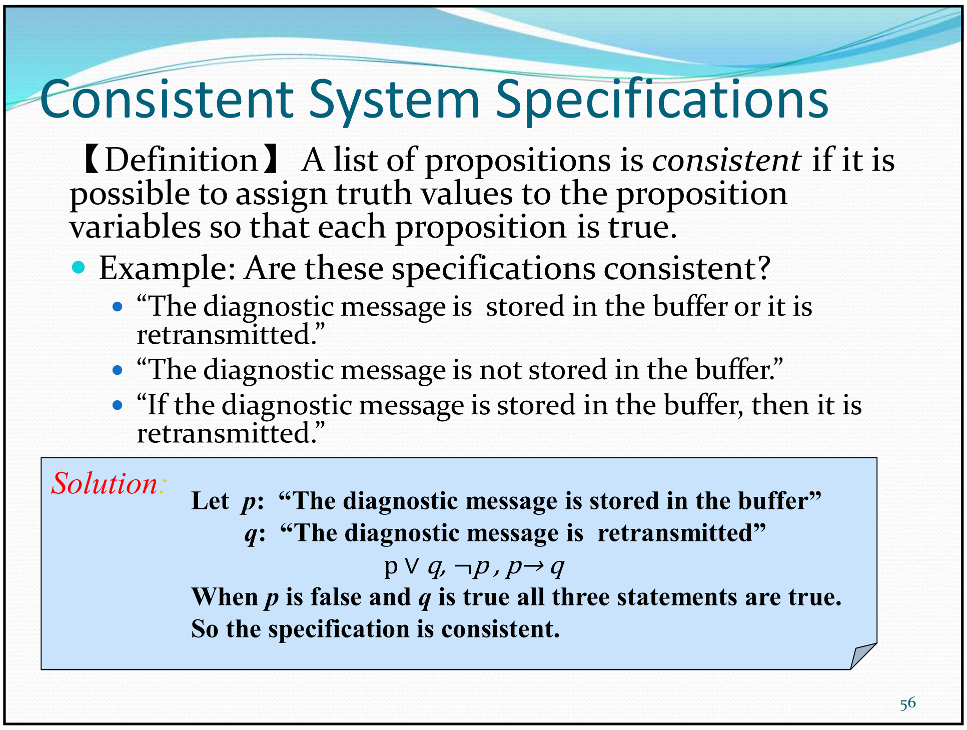
"The automated reply cannot be sent when the file system is full."

Solution:

Let p: "The automated reply can be sent"
q: "The file system is full"

This sentence can be represented as

$$q \rightarrow \neg p$$



Consistent System Specifications

【Definition】 A list of propositions is *consistent* if it is possible to assign truth values to the proposition variables so that each proposition is true.

- Example: Are these specifications consistent?
  - "The diagnostic message is stored in the buffer or it is retransmitted."
  - "The diagnostic message is not stored in the buffer."
  - "If the diagnostic message is stored in the buffer, then it is retransmitted."

Solution:

Let p: "The diagnostic message is stored in the buffer"
q: "The diagnostic message is retransmitted"

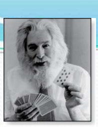
$p \lor q, \neg p , p \to q$

When p is false and q is true all three statements are true.
So the specification is consistent.

Boolean searches

- EXAMPLE 6 web search
- to find pages that deal with universities in New Mexico or Arizona, we can search for pages matching
- (NEW AND MEXICO OR ARIZONA) AND UNIVERSITIES.

Logic Puzzles



Raymond Smullyan (Born 1919)

- Example: An island has two kinds of inhabitants, *knights*, who always tell the truth, and *knaves*, 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."

What are the types of A and B?

Solution:

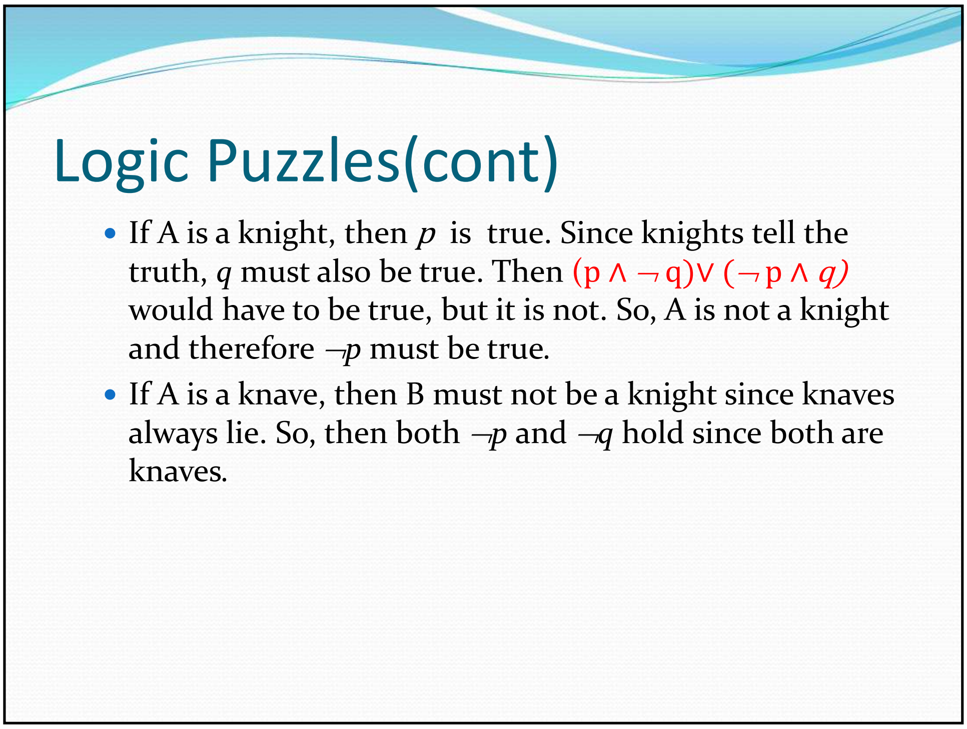
Let p: "A is a knight"

q: "B is a knight"

Then $\neg p$: A is a knave

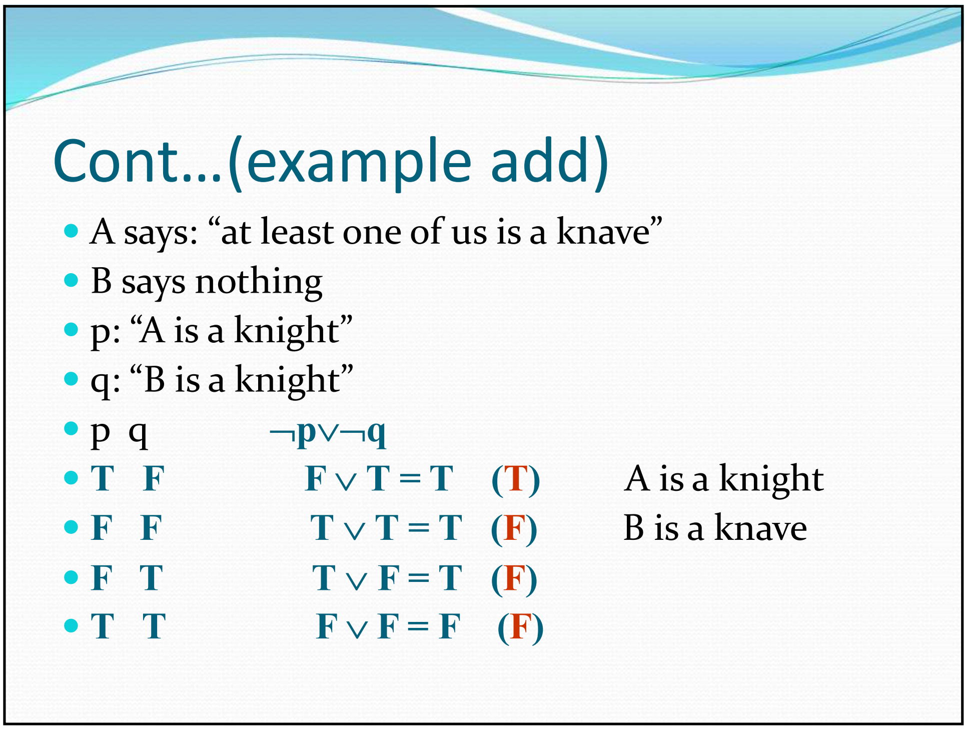
$\neg q$: B is a knave

$q, \quad (p \wedge \neg q) \vee (\neg p \wedge q)$



Logic Puzzles(cont)

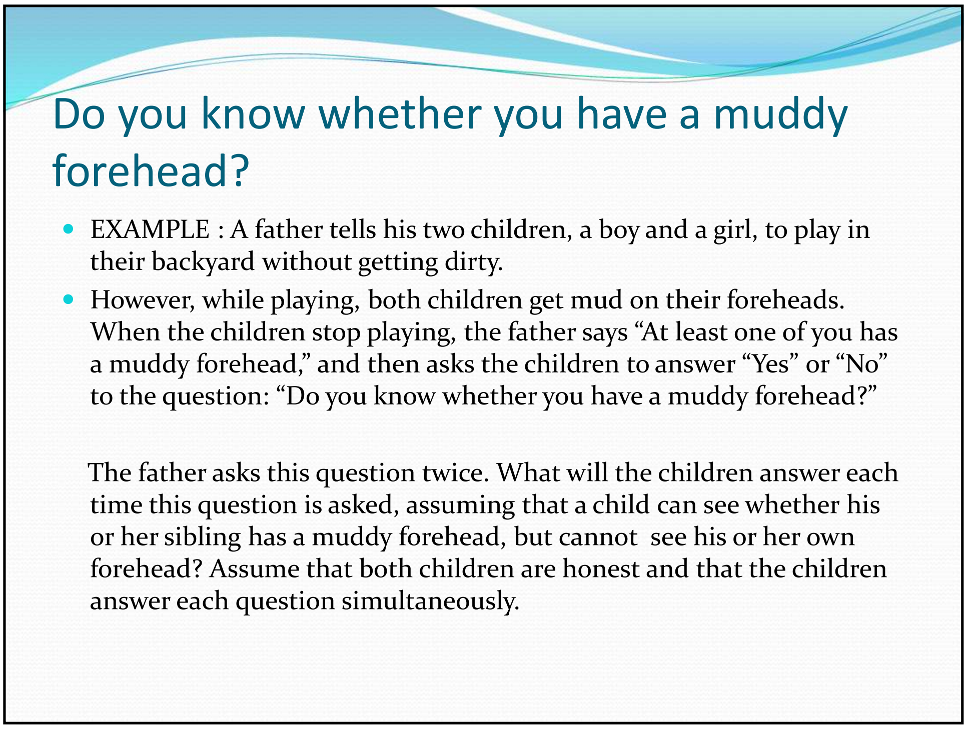
- If A is a knight, then p is true. Since knights tell the truth, q must also be true. Then $(p \wedge \neg q) \vee (\neg p \wedge q)$ would have to be true, but it is not. So, A is not a knight and therefore $\neg p$ must be true.
- If A is a knave, then B must not be a knight since knaves always lie.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



Cont…(example add)

- A says: “at least one of us is a knave”
- B says nothing
- p: “A is a knight”
- q: “B is a knight”

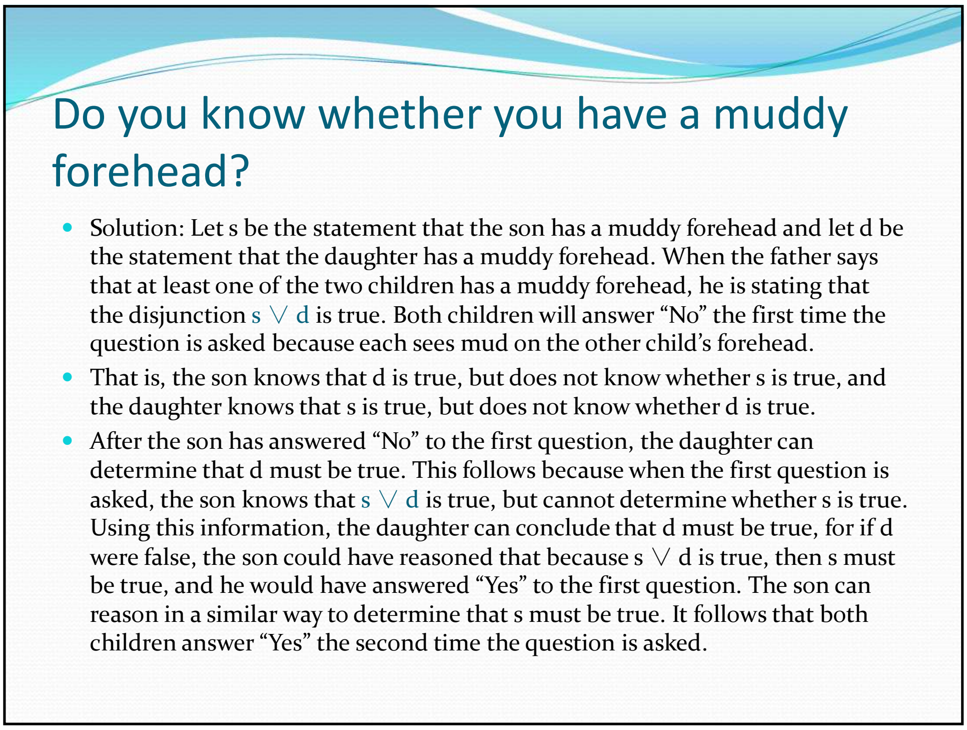
p	q	¬p∨¬q		
T	F	F ∨ T = T	(T)	A is a knight
F	F	T ∨ T = T	(F)	B is a knave
F	T	T ∨ F = T	(F)	
T	T	F ∨ F = F	(F)	



Do you know whether you have a muddy forehead?

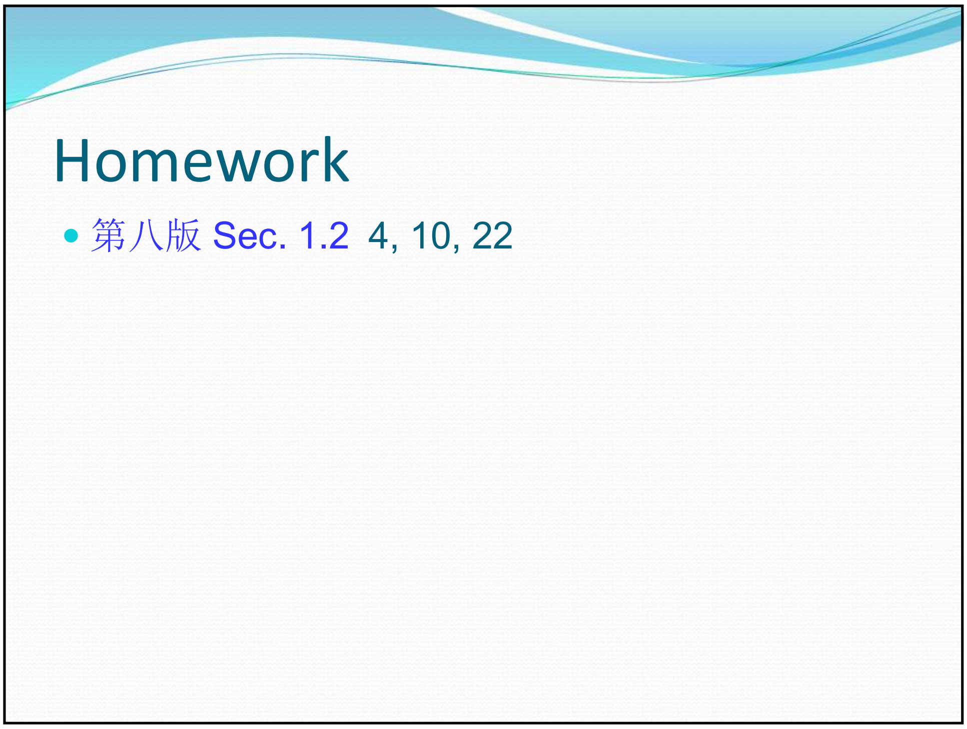
- EXAMPLE : A father tells his two children, a boy and a girl, to play in their backyard without getting dirty.
- However, while playing, both children get mud on their foreheads. When the children stop playing, the father says “At least one of you has a muddy forehead,” and then asks the children to answer “Yes” or “No” to the question: “Do you know whether you have a muddy forehead?”

The father asks this question twice. What will the children answer each time this question is asked, assuming that a child can see whether his or her sibling has a muddy forehead, but cannot see his or her own forehead? Assume that both children are honest and that the children answer each question simultaneously.



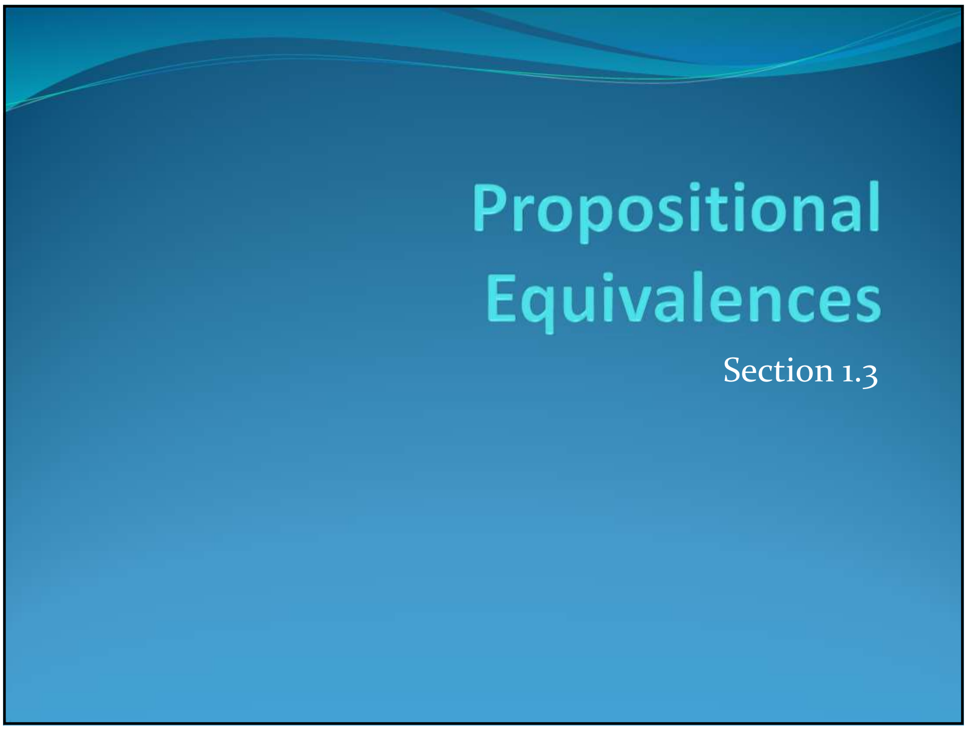
Do you know whether you have a muddy forehead?

- Solution: Let s be the statement that the son has a muddy forehead and let d be the statement that the daughter has a muddy forehead. When the father says that at least one of the two children has a muddy forehead, he is stating that the disjunction $s \vee d$ is true. Both children will answer “No” the first time the question is asked because each sees mud on the other child’s forehead.
- That is, the son knows that d is true, but does not know whether s is true, and the daughter knows that s is true, but does not know whether d is true.
- After the son has answered “No” to the first question, the daughter can determine that d must be true. This follows because when the first question is asked, the son knows that $s \vee d$ is true, but cannot determine whether s is true. Using this information, the daughter can conclude that d must be true, for if d were false, the son could have reasoned that because $s \vee d$ is true, then s must be true, and he would have answered “Yes” to the first question. The son can reason in a similar way to determine that s must be true. It follows that both children answer “Yes” the second time the question is asked.



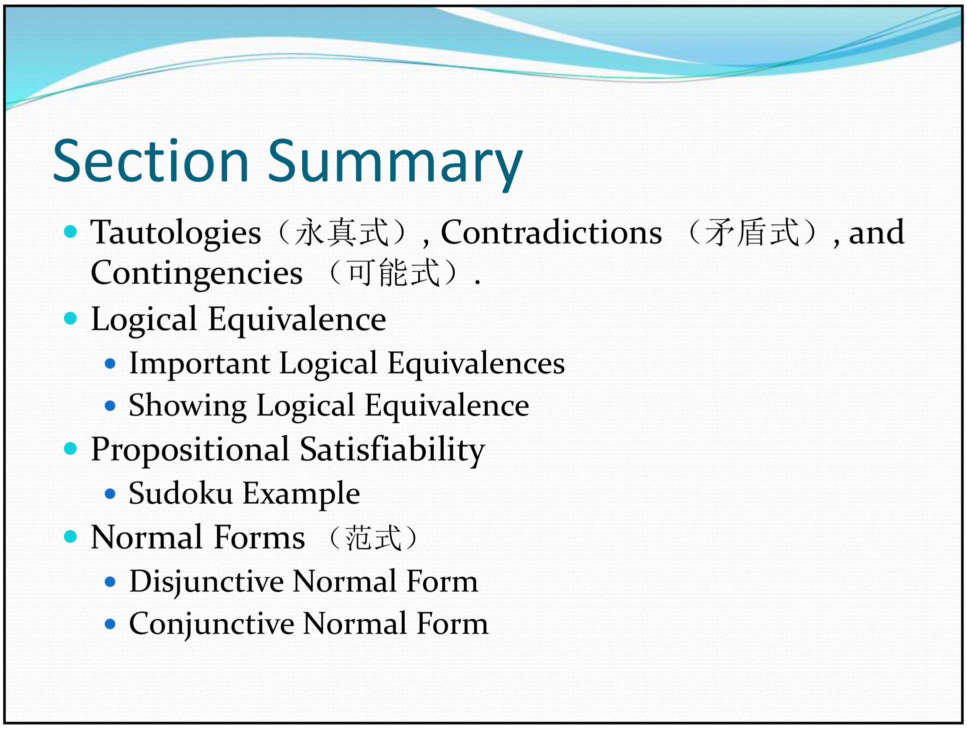
Homework

- 第八版 Sec. 1.2 4, 10, 22



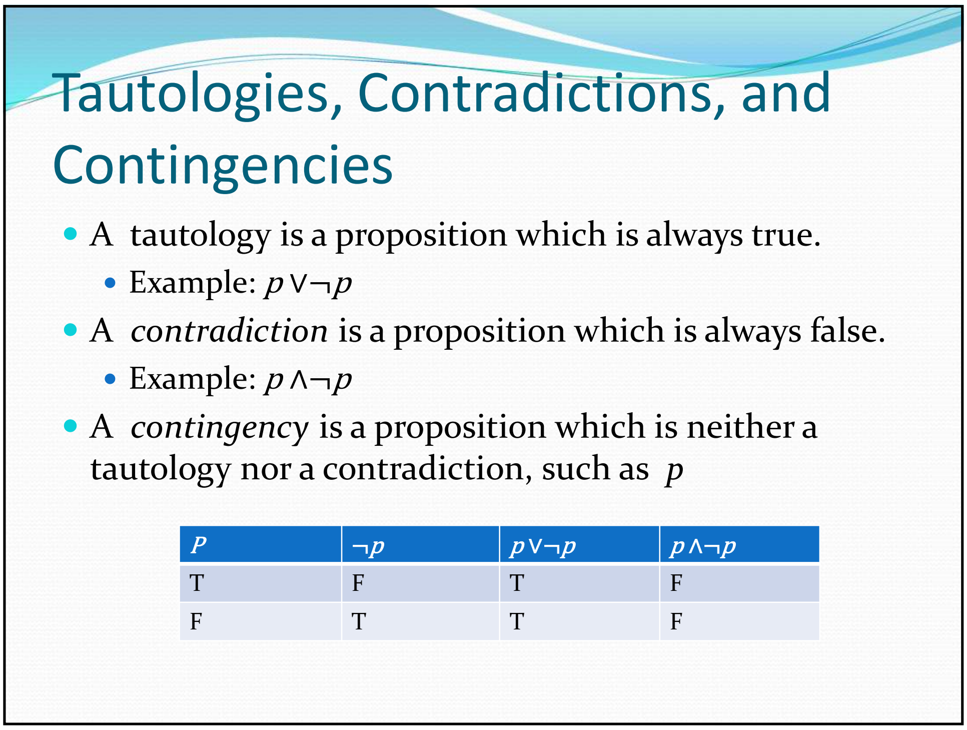
Propositional Equivalences

Section 1.3



Section Summary

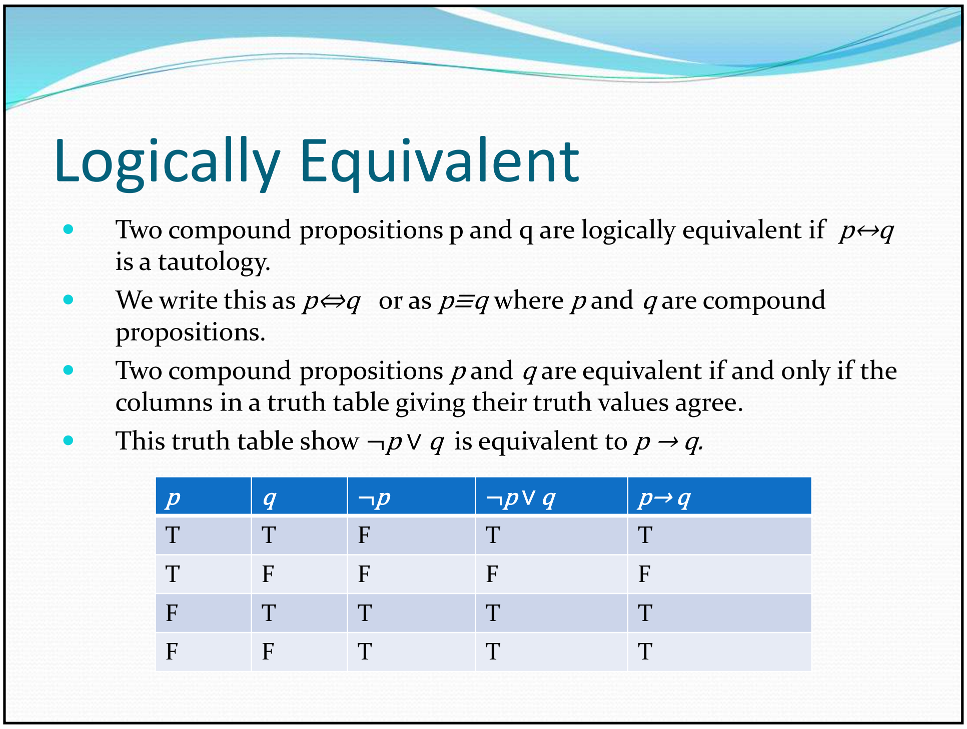
- Tautologies（永真式）, Contradictions （矛盾式）, and Contingencies （可能式） .
- Logical Equivalence
  - Important Logical Equivalences
  - Showing Logical Equivalence
- Propositional Satisfiability
  - Sudoku Example
- Normal Forms （范式）
  - Disjunctive Normal Form
  - Conjunctive Normal Form



Tautologies, Contradictions, and Contingencies

- A tautology is a proposition which is always true.
  - Example: $p \vee \neg p$
- A *contradiction* is a proposition which is always false.
  - Example: $p \wedge \neg p$
- A *contingency* is a proposition which is neither a tautology nor a contradiction, such as p

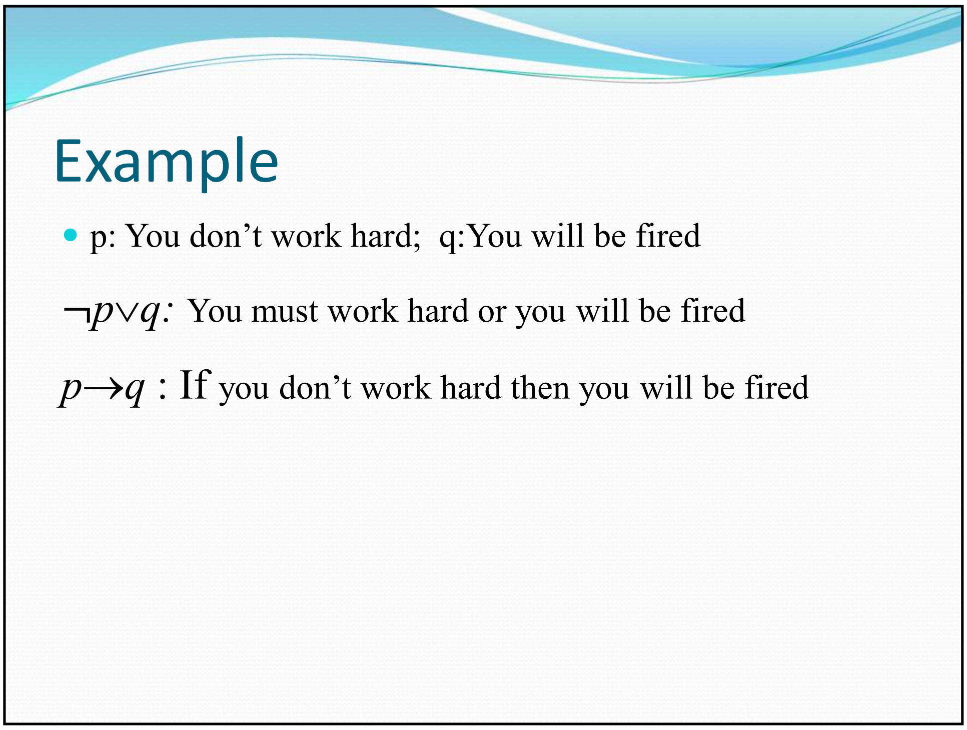
P	$\neg p$	$p \vee \neg p$	$p \wedge \neg p$
T	F	T	F
F	T	T	F



Logically Equivalent

- Two compound propositions p and q are logically equivalent if $p \leftrightarrow q$ is a tautology.
- We write this as $p \Leftrightarrow q$ or as $p \equiv q$ where p and q are compound propositions.
- Two compound propositions p and q are equivalent if and only if the columns in a truth table giving their truth values agree.
- This truth table show $\neg p \vee q$ is equivalent to $p \rightarrow q$.

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

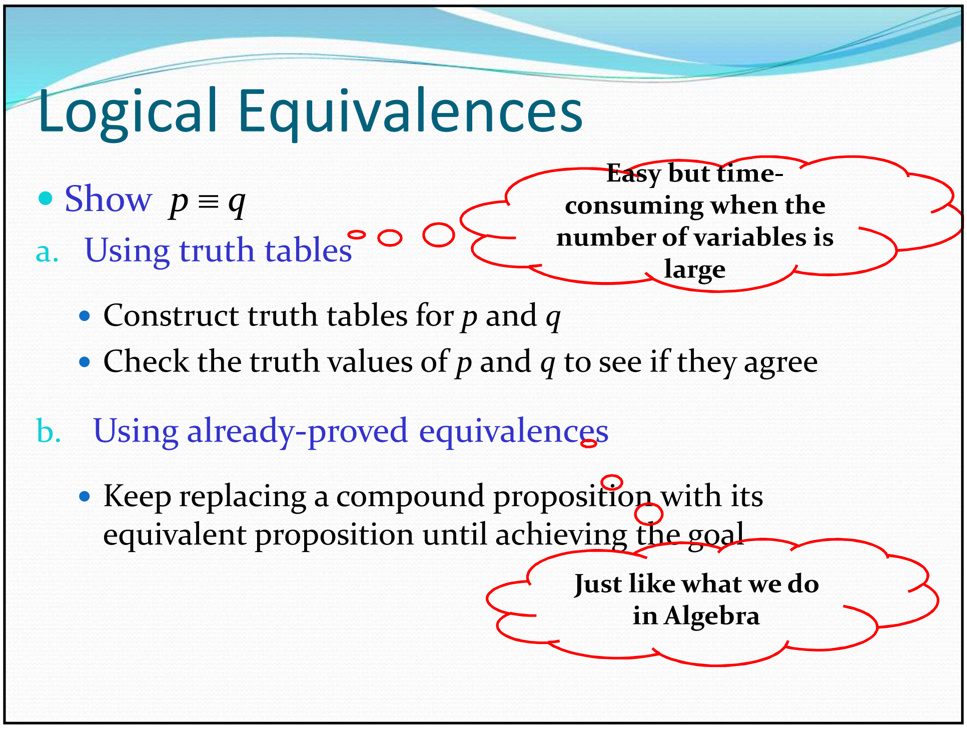


Example

- p: You don’t work hard; q:You will be fired

$\neg p \vee q$: You must work hard or you will be fired

$p \rightarrow q$: If you don’t work hard then you will be fired



Logical Equivalences

- Show $p \equiv q$

a. Using truth tables

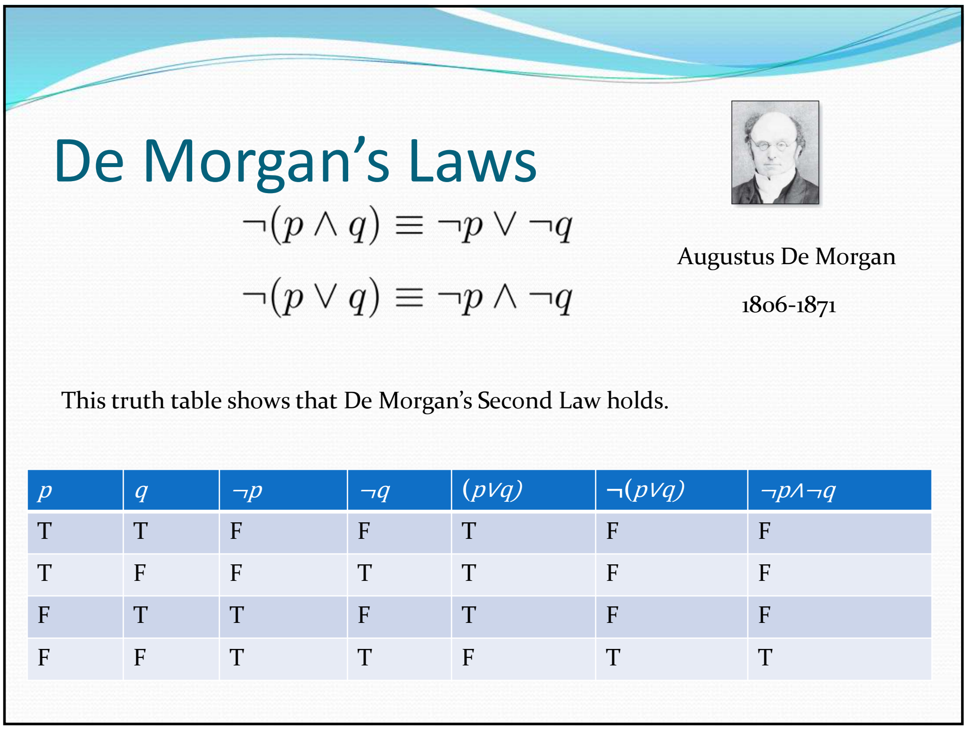
Easy but time-consuming when the number of variables is large

- Construct truth tables for *p* and *q*
- Check the truth values of *p* and *q* to see if they agree

b. Using already-proved equivalences

- Keep replacing a compound proposition with its equivalent proposition until achieving the goal

Just like what we do in Algebra



De Morgan's Laws

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

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

Augustus De Morgan

1806-1871

This truth table shows that De Morgan's Second Law holds.

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

Using De Morgan's Laws

- Use De Morgan's laws to express the negation of "Mike has a cellphone and he has a laptop computer" and "Jim will go to the concert or Steve will go to the concert."

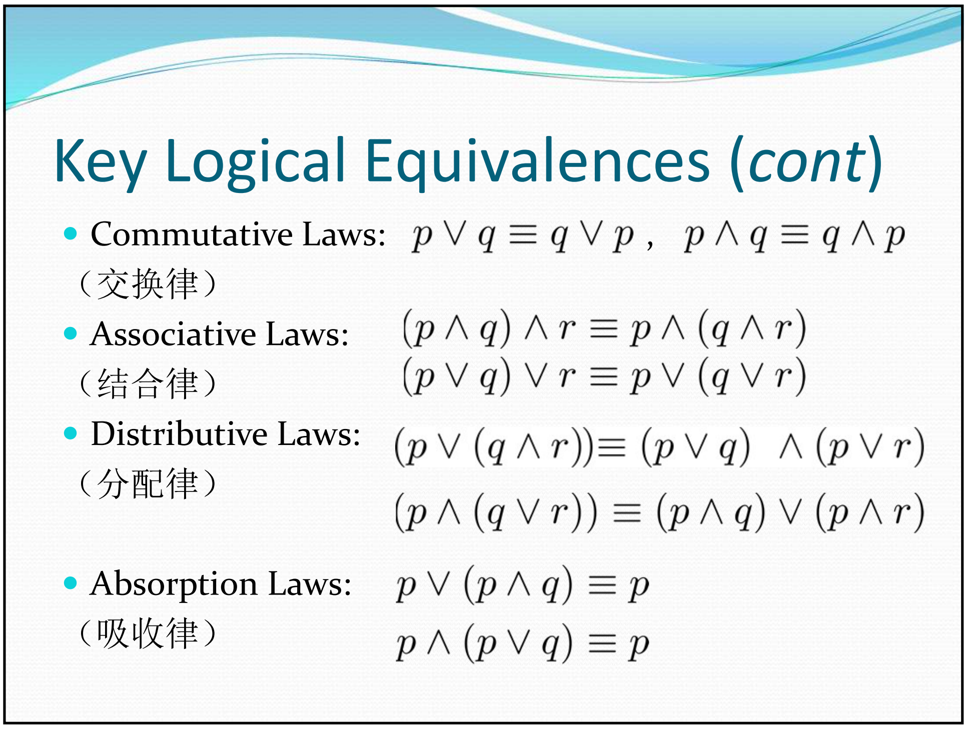
Solution:

Mike does not have a cellphone or he does not have a laptop computer.

Jim will not go to the concert and Steve will not go to the concert.

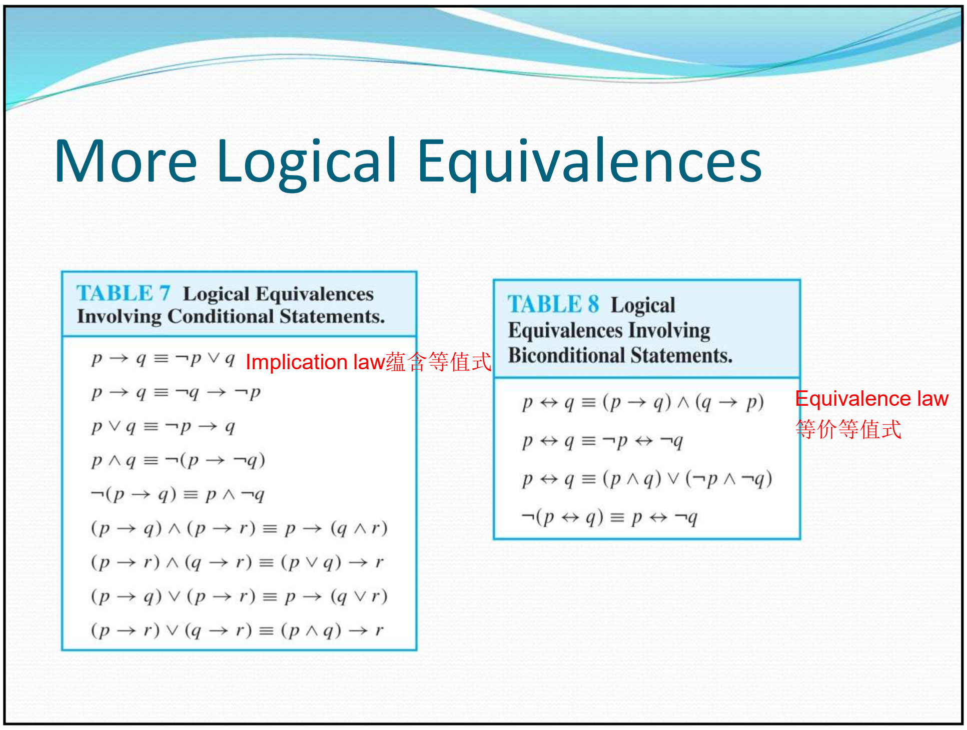
Key Logical Equivalences

- Identity Laws: $p \wedge T \equiv p$, $p \vee F \equiv p$
  （恒等律）
- Domination Laws: $p \vee T \equiv T$, $p \wedge F \equiv F$
  （支配律）
- Idempotent laws: $p \vee p \equiv p$, $p \wedge p \equiv p$
  （幂等律）
- Double Negation Law: $\neg(\neg p) \equiv p$
  （双重否定律）
- Negation Laws: $p \vee \neg p \equiv T$, $p \wedge \neg p \equiv F$
  （否定律）



Key Logical Equivalences (*cont*)

- Commutative Laws: $p \vee q \equiv q \vee p$, $p \wedge q \equiv q \wedge p$
（交换律）
- Associative Laws: $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
（结合律） $(p \vee q) \vee r \equiv p \vee (q \vee r)$
- Distributive Laws: $(p \vee (q \wedge r)) \equiv (p \vee q) \wedge (p \vee r)$
（分配律）
$(p \wedge (q \vee r)) \equiv (p \wedge q) \vee (p \wedge r)$
- Absorption Laws: $p \vee (p \wedge q) \equiv p$
（吸收律） $p \wedge (p \vee q) \equiv p$



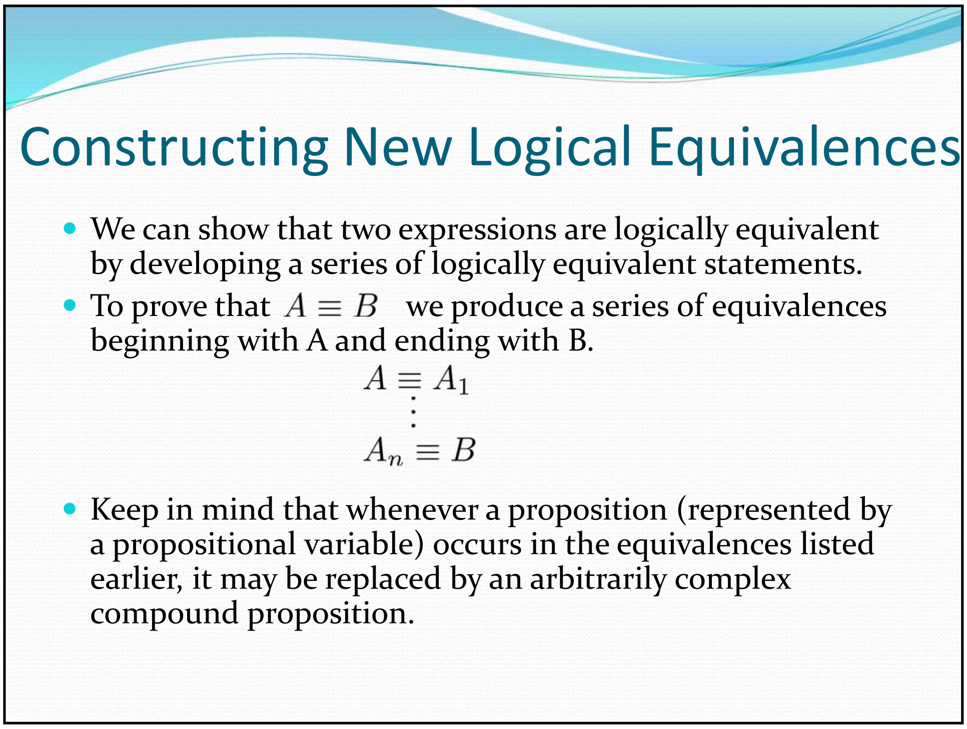
More Logical Equivalences

TABLE 7 Logical Equivalences Involving Conditional Statements.
$p \rightarrow q \equiv \neg p \vee q$
$p \rightarrow q \equiv \neg q \rightarrow \neg p$
$p \vee q \equiv \neg p \rightarrow q$
$p \wedge q \equiv \neg(p \rightarrow \neg q)$
$\neg(p \rightarrow q) \equiv p \wedge \neg q$
$(p \rightarrow q) \wedge (p \rightarrow r) \equiv p \rightarrow (q \wedge r)$
$(p \rightarrow r) \wedge (q \rightarrow r) \equiv (p \vee q) \rightarrow r$
$(p \rightarrow q) \vee (p \rightarrow r) \equiv p \rightarrow (q \vee r)$
$(p \rightarrow r) \vee (q \rightarrow r) \equiv (p \wedge q) \rightarrow r$

Implication law蕴含等值式

TABLE 8 Logical Equivalences Involving Biconditional Statements.
$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$
$p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$
$p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$
$\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$

Equivalence law
等价等值式

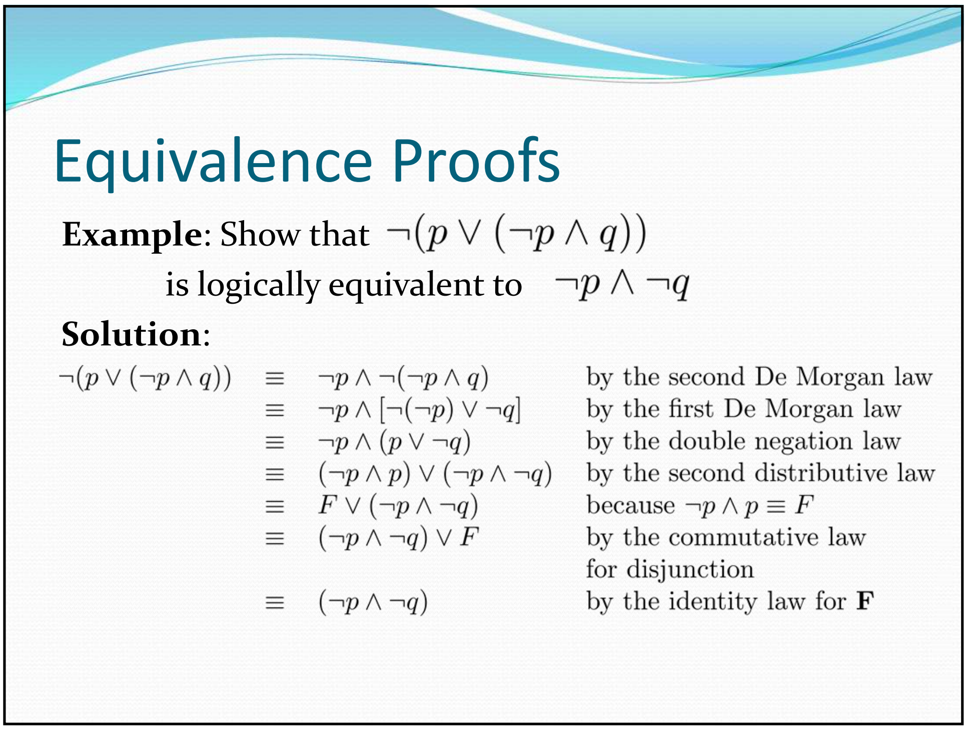


Constructing New Logical Equivalences

- We can show that two expressions are logically equivalent by developing a series of logically equivalent statements.
- To prove that $A \equiv B$ we produce a series of equivalences beginning with A and ending with B.

$$\begin{aligned} A &\equiv A_1 \\ &\vdots \\ A_n &\equiv B \end{aligned}$$

- Keep in mind that whenever a proposition (represented by a propositional variable) occurs in the equivalences listed earlier, it may be replaced by an arbitrarily complex compound proposition.

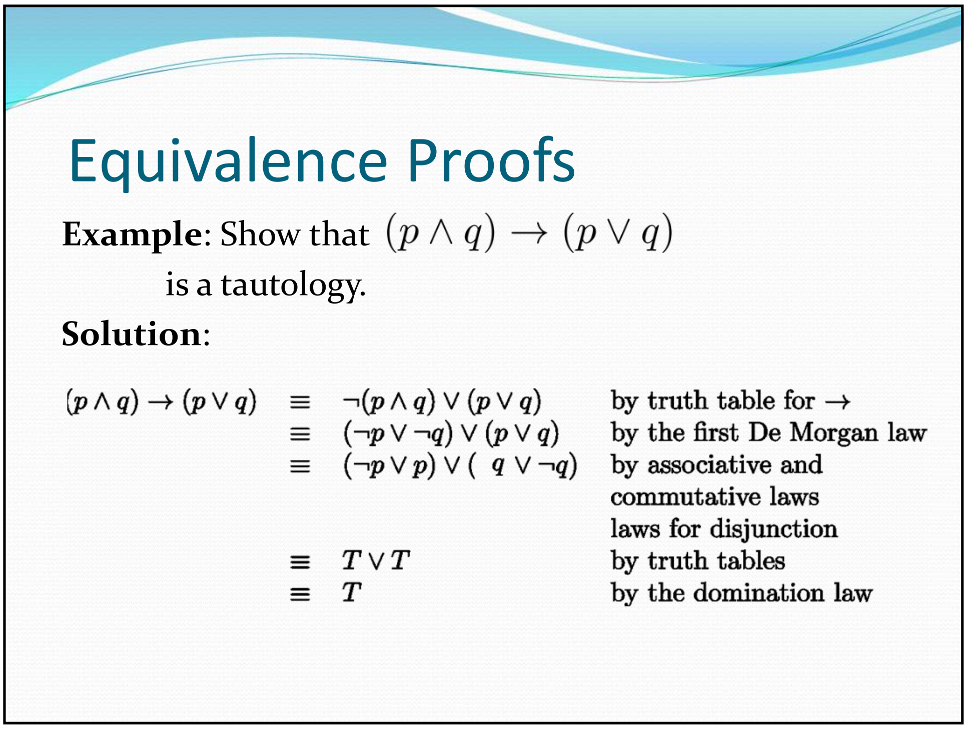


Equivalence Proofs

Example: Show that $\neg(p \vee (\neg p \wedge q))$ is logically equivalent to $\neg p \wedge \neg q$

Solution:

$$\begin{aligned}
\neg(p \vee (\neg p \wedge q)) &\equiv \neg p \wedge \neg(\neg p \wedge q) && \text{by the second De Morgan law} \\
&\equiv \neg p \wedge [\neg(\neg p) \vee \neg q] && \text{by the first De Morgan law} \\
&\equiv \neg p \wedge (p \vee \neg q) && \text{by the double negation law} \\
&\equiv (\neg p \wedge p) \vee (\neg p \wedge \neg q) && \text{by the second distributive law} \\
&\equiv F \vee (\neg p \wedge \neg q) && \text{because } \neg p \wedge p \equiv F \\
&\equiv (\neg p \wedge \neg q) \vee F && \text{by the commutative law} \\
& && \text{for disjunction} \\
&\equiv (\neg p \wedge \neg q) && \text{by the identity law for } \mathbf{F}
\end{aligned}$$

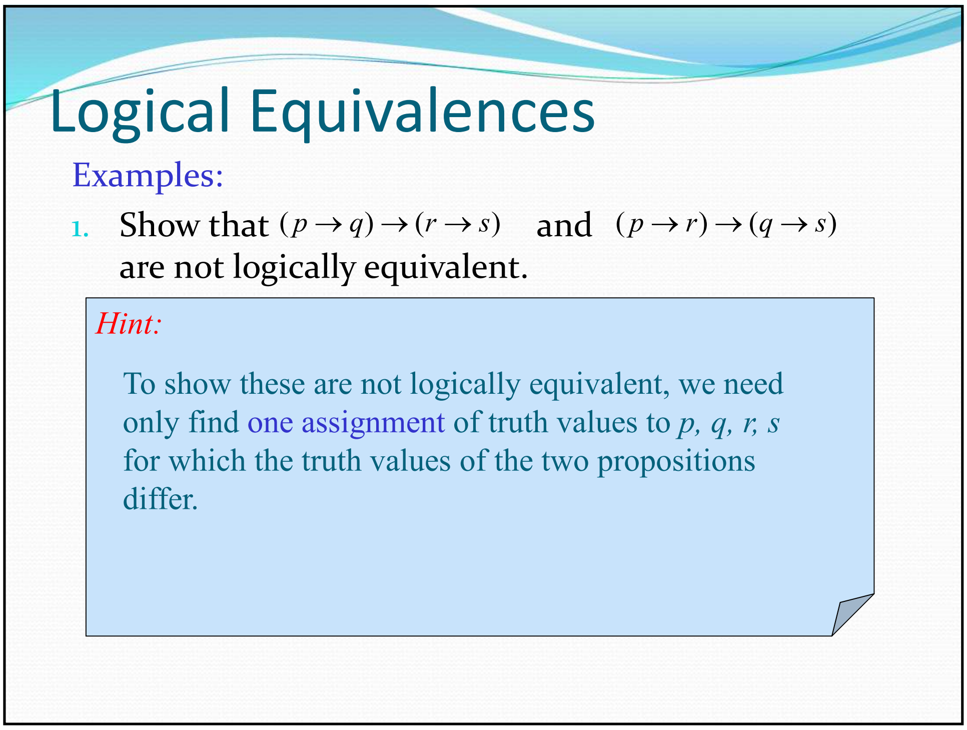


Equivalence Proofs

Example: Show that $(p \land q) \to (p \lor q)$ is a tautology.

Solution:

$$\begin{array}{llll}
(p \land q) \to (p \lor q) & \equiv & \neg(p \land q) \lor (p \lor q) & \text{by truth table for} \to \\
 & \equiv & (\neg p \lor \neg q) \lor (p \lor q) & \text{by the first De Morgan law} \\
 & \equiv & (\neg p \lor p) \lor (\ q \lor \neg q) & \text{by associative and} \\
 & & & \text{commutative laws} \\
 & & & \text{laws for disjunction} \\
 & \equiv & T \lor T & \text{by truth tables} \\
 & \equiv & T & \text{by the domination law}
\end{array}$$



Logical Equivalences

Examples:

1. Show that $(p \to q) \to (r \to s)$ and $(p \to r) \to (q \to s)$ are not logically equivalent.

Hint:

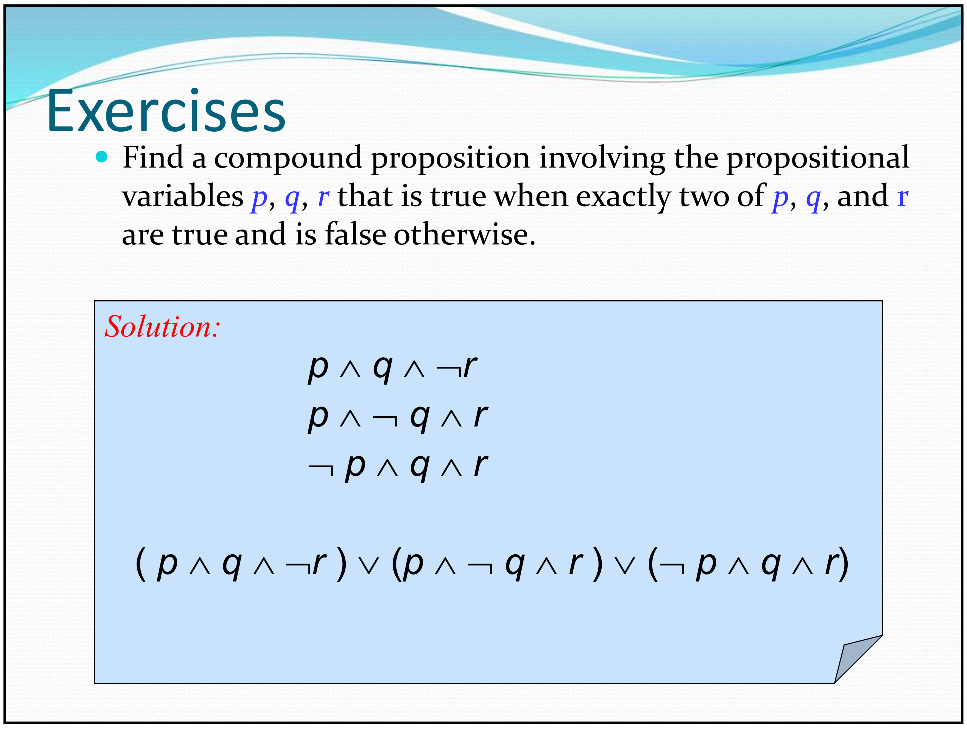
To show these are not logically equivalent, we need only find one assignment of truth values to *p, q, r, s* for which the truth values of the two propositions differ.

Dual （对偶式）

- The dual of compound proposition that contains only the logical operators $\vee$, $\wedge$ and $\neg$ is the proposition obtained by replacing each $\vee$ by $\wedge$,each $\wedge$ by $\vee$,each T by F and each F by T. the dual of S is denoted by S*.
- $S=(p \vee \neg q) \wedge r \vee T$ $S^*= (p \wedge \neg q) \vee r \wedge F$

Dual (cont…)

- $S= (p \wedge q) \rightarrow (p \vee q) = \neg(p \wedge q) \vee (p \vee q)$
- $S^*= \neg(p \vee q) \wedge (p \wedge q)$
- Theorem: let s and t are two compound propositions, s $\Leftrightarrow$ t if and only if s* $\Leftrightarrow$ t*



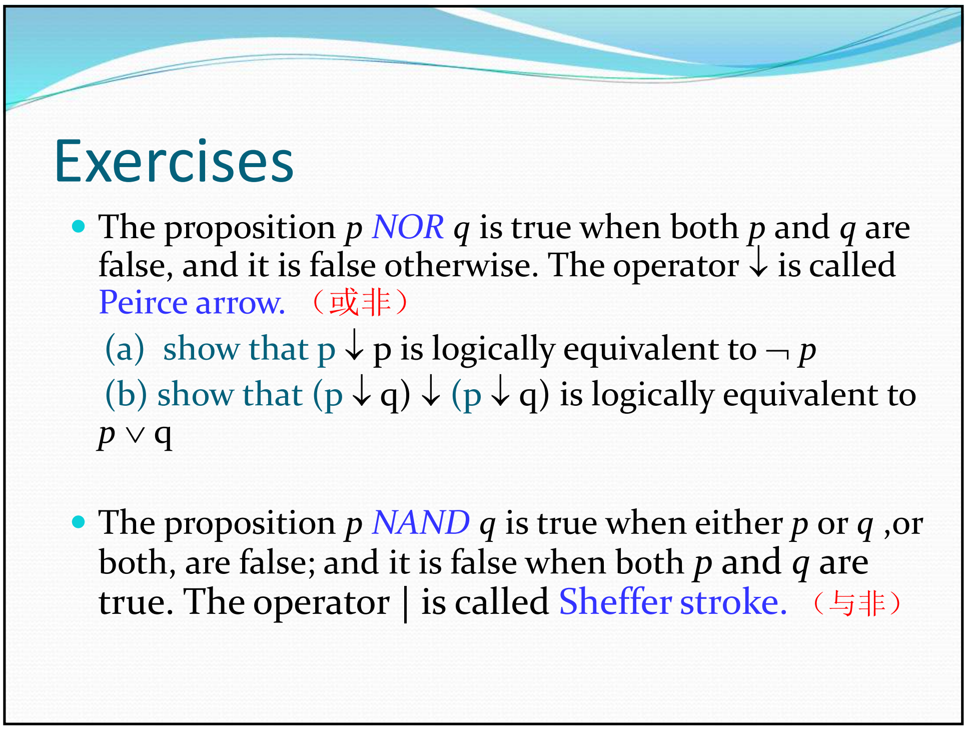
Exercises

- Find a compound proposition involving the propositional variables *p*, *q*, *r* that is true when exactly two of *p*, *q*, and r are true and is false otherwise.

Solution:

$$p \wedge q \wedge \neg r$$
$$p \wedge \neg q \wedge r$$
$$\neg p \wedge q \wedge r$$

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



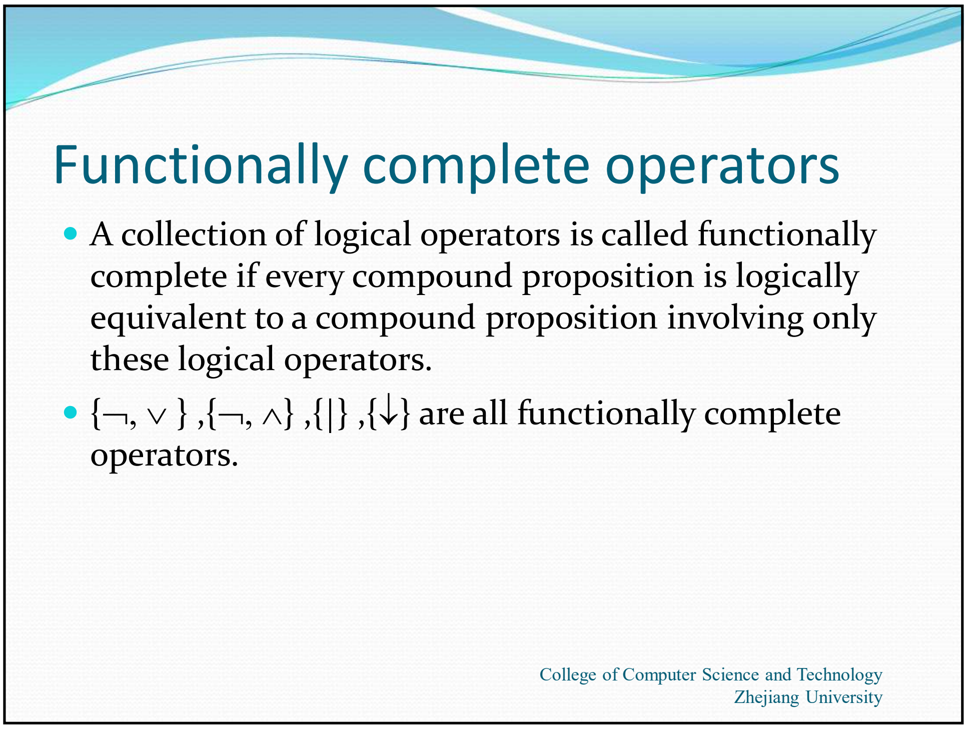
Exercises

- The proposition *p NOR q* is true when both *p* and *q* are false, and it is false otherwise. The operator ↓ is called Peirce arrow. （或非）

  (a) show that p ↓ p is logically equivalent to ¬ *p*

  (b) show that (p ↓ q) ↓ (p ↓ q) is logically equivalent to *p* ∨ q

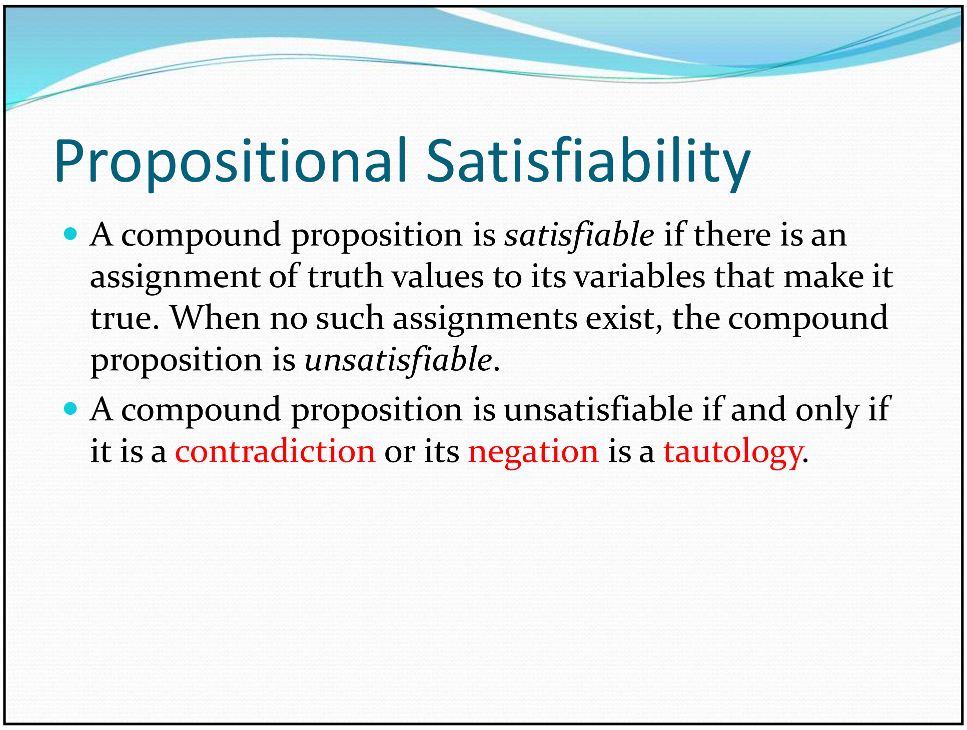
- The proposition *p NAND q* is true when either *p* or *q* ,or both, are false; and it is false when both *p* and *q* are true. The operator | is called Sheffer stroke. （与非）



Functionally complete operators

- A collection of logical operators is called functionally complete if every compound proposition is logically equivalent to a compound proposition involving only these logical operators.
- $\{\neg, \vee\}$,$\{\neg, \wedge\}$,$\{|\}$,$\{\downarrow\}$ are all functionally complete operators.

College of Computer Science and Technology
Zhejiang University



Propositional Satisfiability

- A compound proposition is *satisfiable* if there is an assignment of truth values to its variables that make it true. When no such assignments exist, the compound proposition is *unsatisfiable*.
- A compound proposition is unsatisfiable if and only if it is a contradiction or its negation is a tautology.

Questions on Propositional Satisfiability

Example: Determine the satisfiability of the following compound propositions:

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

Solution: Satisfiable. Assign **T** to *p, q,* and *r.*

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

Solution: Satisfiable. Assign **T** to *p* and ***F*** to *q*.

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

Solution: Not satisfiable. Check each possible assignment of truth values to the propositional variables and none will make the proposition true.

Notation

$$\bigvee_{j=1}^{n} p_j \text{ is used for } p_1 \vee p_2 \vee \ldots \vee p_n$$

$$\bigwedge_{j=1}^{n} p_j \text{ is used for } p_1 \wedge p_2 \wedge \ldots \wedge p_n$$

Needed for the next example.

Sudoku

- A **Sudoku puzzle** is represented by a 9×9 grid made up of nine 3×3 subgrids, known as **blocks**. Some of the 81 cells of the puzzle are assigned one of the numbers 1,2, …, 9.
- The puzzle is solved by assigning numbers to each blank cell so that every row, column and block contains each of the nine possible numbers.
- Example

	2	9				4		
			5			1		
	4							
				4	2			
6							7	
5								
7			3					5
	1			9				
							6	

Encoding as a Satisfiability Problem

- Let $p(i,j,n)$ denote the proposition that is true when the number n is in the cell in the ith row and the jth column.
- There are 9×9 × 9 = 729 such propositions.
- In the sample puzzle $p(5,1,6)$ is true, but $p(5,j,6)$ is false for $j = 2,3,\ldots 9$

	2	9				4		
			5			1		
	4							
				4	2			
6							7	
5								
7			3					5
	1			9				
							6	

Encoding (cont)

- For each cell with a given value, assert *p*(*i*,*j*,*n*), when the cell in row *i* and column *j* has the given value.
- Assert that every row contains every number.

$$\bigwedge_{i=1}^{9}\bigwedge_{n=1}^{9}\bigvee_{j=1}^{9} p(i,j,n)$$

- Assert that every column contains every number.

$$\bigwedge_{j=1}^{9}\bigwedge_{n=1}^{9}\bigvee_{i=1}^{9} p(i,j,n)$$

	2	9				4		
			5			1		
	4							
				4	2			
6							7	
5								
7			3					5
	1			9				
							6	

Encoding (cont)

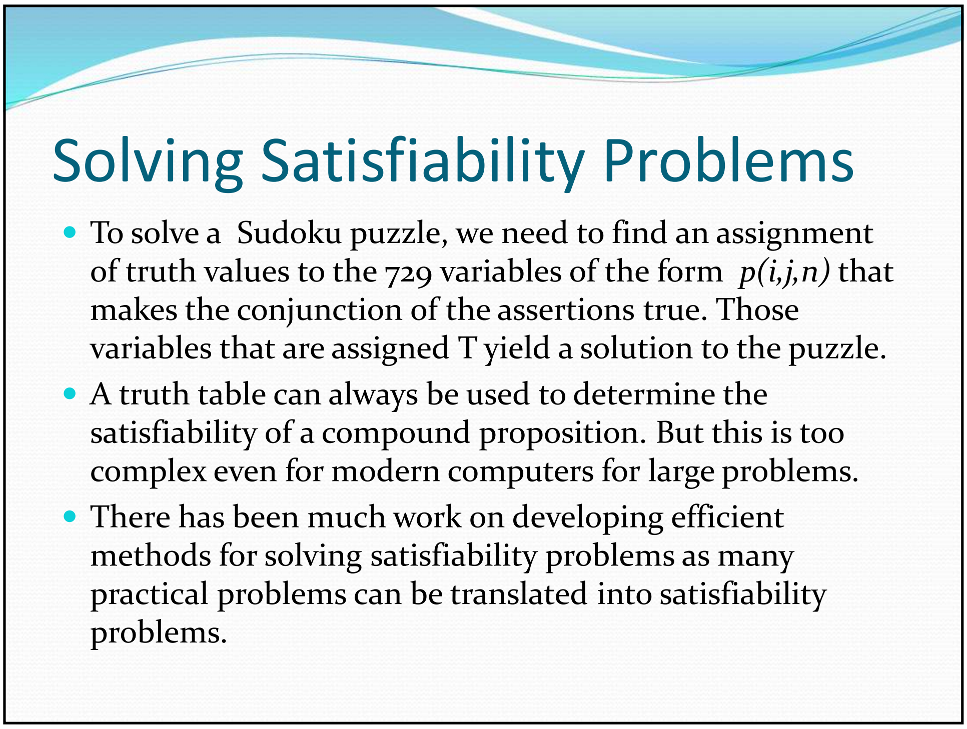
- Assert that each of the 3 x 3 blocks contain every number.

$$\bigwedge_{r=0}^{2}\bigwedge_{s=0}^{2}\bigwedge_{n=1}^{9}\bigvee_{i=1}^{3}\bigvee_{j=1}^{3} p(3r+i,3s+j,n)$$

(this is tricky - ideas from chapter 4 help)

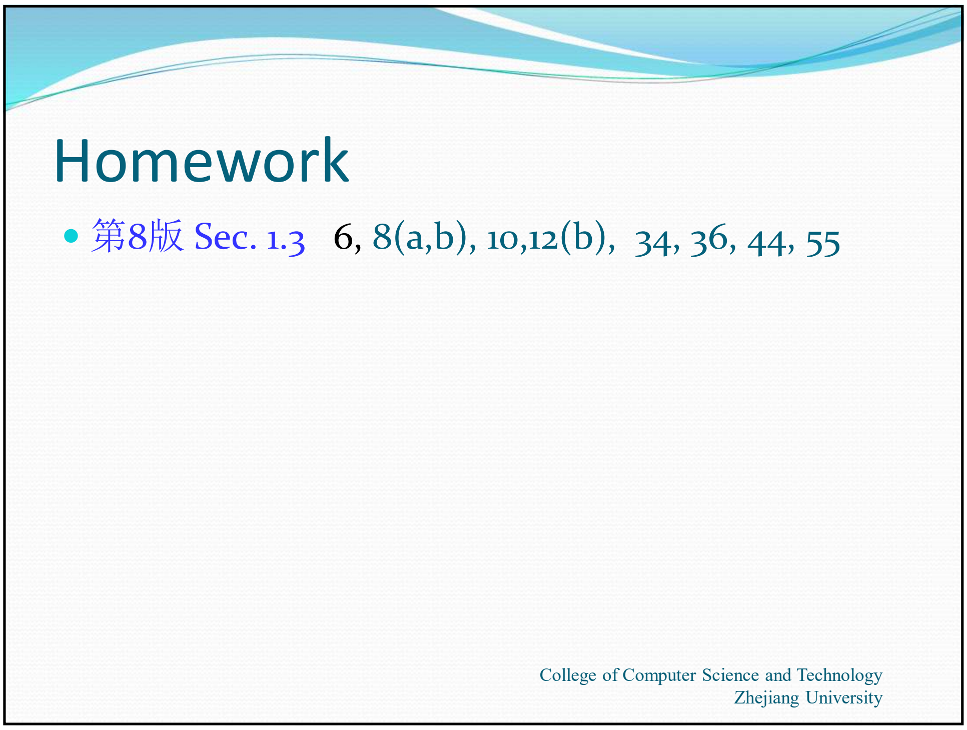
- Assert that no cell contains more than one number. Take the conjunction over all values of *n*, *n'*, *i*, and j, where each variable ranges from 1 to 9 and $n \neq n'$, of

$$p(i,j,n) \rightarrow \neg p(i,j,n')$$



Solving Satisfiability Problems

- To solve a Sudoku puzzle, we need to find an assignment of truth values to the 729 variables of the form *p(i,j,n)* that makes the conjunction of the assertions true. Those variables that are assigned T yield a solution to the puzzle.
- A truth table can always be used to determine the satisfiability of a compound proposition. But this is too complex even for modern computers for large problems.
- There has been much work on developing efficient methods for solving satisfiability problems as many practical problems can be translated into satisfiability problems.



Homework

- 第8版 Sec. 1.3 6, 8(a,b), 10,12(b), 34, 36, 44, 55

College of Computer Science and Technology
Zhejiang University