# 1.2
# Applications of Propositional Logic

# Section Summary

**Logic has many important applications to mathematics, computer science, and numerous other disciplines.**

**Some of these applications:**

- ✓ **Translating English to Propositional Logic**
- ✓ **System Specifications**
- ✓ **Boolean Searching**
- ✓ **Logic Puzzles**
- ✓ **Logic Circuits**

# Translating Sentences

**Why translate English sentence into the language of logic?**

- **Remove the inaccuracy and ambiguity of statements in natural language**
- **Analyze the logic expressions to determine their truth values**
- **Manipulate**
- **Use rules of inference to reason**

# Translating Sentences

**How to translate an English sentence to a statement in propositional logic?**

- Steps
  - **Identify atomic propositions and represent using propositional variables.**
  - **Determine appropriate logical connectives**
  - **Express the sentence in logical expression**

Example,

- "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*.**

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

# Another 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 $a$, $c$ and $f$ represent "you can access the Internet from campus","you are a computer science major" and "you are a freshman".**

**This sentence can be represented as**

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

# System Specification

**System and Software engineers take requirements in English and express them in a precise specification language based on logic.**

**Example: Express in propositional logic:**

**"It is necessary to scanned the message for viruses whenever the message was sent from an unknown system ."**

***Solution:***

**Let *p* denote "The message is scanned for viruses "**

**Let *q* denote "the message was sent from an unknown system".**

**This specification can be represented as**

$$q \rightarrow p$$

# 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 $p$ denote "The diagnostic message is stored in the buffer." Let $q$ denote "The diagnostic message is retransmitted" The specification can be written as: $p \vee q$, $p \rightarrow q$, $\neg p$. When $p$ is false and $q$ is true all three statements are true. So the specification is consistent.

- **What if "The diagnostic message is not retransmitted" is added.**

**Solution**: Now we are adding $\neg q$ and there is no satisfying assignment. So the specification is not consistent.

# Logic Puzzles

Logic puzzle: can be solved using logical reasoning.

Solving logic puzzles is an excellent way to practice working with the rules of logic.

Computer programs designed to carry out logical reasoning often use well-known logic puzzles to illustrate their capabilities.

# Examples of Logic Puzzle

**Example**: As a reward for saving his daughter from pirates, the King has given you the opportunity to win a treasure hidden inside one of three trunks. The two trunks that do not hold the treasure are empty. To win, you must select the correct trunk. Trunks 1 and 2 are each inscribed with the message "This trunk is empty," and Trunk 3 is inscribed with the message "The treasure is in Trunk 2." The Queen, who never lies, tells you that only one of these inscriptions is true, while the other two are wrong. Which trunk should you select to win?

**Solution:**

Let $p_i$ be the proposition that the treasure is in Trunk $i$, for $i = 1, 2, 3$.

The inscriptions on Trunk 1, Trunk 2, and Trunk 3, are $\neg p_1$, $\neg p_2$, and $p_2$, respectively

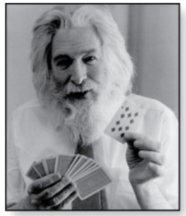
The Queen's statement can be translated to

$$(\neg p_1 \wedge \neg(\neg p_2) \wedge \neg p_2) \vee (\neg(\neg p_1) \wedge \neg p_2 \wedge \neg p_2) \vee (\neg(\neg p_1) \wedge \neg(\neg p_2) \wedge p_2))$$

This is then equivalent to $p_1$ .

So the treasure is in Trunk 1.

# Examples of Logic Puzzle



Raymond Smullyan (Born 1919)

- 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."

**Example**: 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 $(p \land \neg q) \lor (\neg p \land q)$ 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.

**Homework:**

**SE：P.22 6, 9, 24, 35**

**EE：P.23 6, 9, 28, 39**

# 1.3 Propositional Equivalences

# Section Summary

- ✓Tautologies, Contradictions, and Contingencies
- ✓Logical Equivalence
  - Important Logical Equivalences
  - Showing Logical Equivalence
- ✓Propositional Satisfiability
  - The n-Queens Problem
  - Sudoku puzzle
- ✓Other logical operators
- ✓The Dual of a Compound Proposition
- ✓Functionally Complete Collection of Logical Operators
- ✓Normal Forms (covered in exercises in text)
  - DNF & Full DNF
  - CNF & Full CNF

# 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 | ¬p | p ∨¬p | p ∧¬p |
|---|---|---|---|
| T | F | T | F |
| F | T | T | F |

# Logically Equivalent

Compound propositions that have the same values in all possible cases are called logically equivalent. We can also define as:

**The propositions $p$ and $q$ are called logically equivalent if $p \leftrightarrow q$ is a tautology.**

***Notation:*** $p \Leftrightarrow q$ **or** $p \equiv q$

***Remark:***

**The symbol $\equiv$ is not a logical connective, and $p \equiv q$ is not a compound proposition but rather is the statement that $p \leftrightarrow q$ is a tautology.**

## How to determine whether two compound propositions are equivalent?

**Method 1: to use a truth table**

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 | ¬p | ¬p ∨ q | p→ q |
|---|---|---|---|---|
| T | T | F | T | T |
| T | F | F | F | F |
| F | T | T | T | T |
| F | F | T | T | T |

**Implication law:** $\boxed{p \rightarrow q \equiv \neg p \vee q}$

# De Morgan's Laws

Augustus De Morgan
1806-1871

$$\neg(p \land q) \equiv \neg p \lor \neg q$$

$$\neg(p \lor q) \equiv \neg p \land \neg q$$

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

| $p$ | $q$ | $\neg p$ | $\neg q$ | $(p \lor q)$ | $\neg(p \lor q)$ | $\neg p \land \neg q$ |
|---|---|---|---|---|---|---|
| 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 |

# Key Logical Equivalences

| Name | Equivalences | |
|---|---|---|
| Identity laws | $p \wedge T \equiv p$ | $p \vee F \equiv p$ |
| Domination laws | $p \vee T \equiv T$ | $p \wedge F \equiv F$ |
| Idempotent laws | $p \vee p \equiv p$ | $p \wedge p \equiv p$ |
| Double negation law | $\neg\neg p \equiv p$ | |
| Commutative laws | $p \vee q \equiv q \vee p$ | $p \wedge q \equiv q \wedge p$ |
| Associative laws | $(p \vee q) \vee r \equiv p \vee (q \vee r)$<br>$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$ | |
| Distributive laws | $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$<br>$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$ | |
| De Morgan's laws | $\neg(p \wedge q) \equiv \neg p \vee \neg q$<br>$\neg(p \vee q) \equiv \neg p \wedge \neg q$ | |

| Name | Equivalences |
|---|---|
| Negation laws | $p \vee \neg p \equiv T, p \wedge \neg p \equiv F$ |
| Absorption laws | $p \vee (p \wedge q) \equiv p \quad p \wedge (p \vee q) \equiv p$ |
| Contrapositive law | $p \rightarrow q \equiv \neg q \rightarrow \neg p$ |
| Exportation law | $(p \wedge q) \rightarrow r \equiv p \rightarrow (q \rightarrow r)$ |
| Absurdity law | $(p \rightarrow q) \wedge (p \rightarrow \neg q) \equiv \neg p$ |
| Implication law | $p \rightarrow q \equiv \neg p \vee q$ |
| Equivalence law | $p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$ |

# More Logical Equivalences

| TABLE 7 Logical Equivalences Involving Conditional Statements. |
| --- |
| $p \to q \equiv \neg p \lor q$ |
| $p \to q \equiv \neg q \to \neg p$ |
| $p \lor q \equiv \neg p \to q$ |
| $p \land q \equiv \neg(p \to \neg q)$ |
| $\neg(p \to q) \equiv p \land \neg q$ |
| $(p \to q) \land (p \to r) \equiv p \to (q \land r)$ |
| $(p \to r) \land (q \to r) \equiv (p \lor q) \to r$ |
| $(p \to q) \lor (p \to r) \equiv p \to (q \lor r)$ |
| $(p \to r) \lor (q \to r) \equiv (p \land q) \to r$ |

| TABLE 8 Logical Equivalences Involving Biconditional Statements. |
| --- |
| $p \leftrightarrow q \equiv (p \to q) \land (q \to p)$ |
| $p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$ |
| $p \leftrightarrow q \equiv (p \land q) \lor (\neg p \land \neg q)$ |
| $\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$ |

# The extended version of De Morgan's Laws

$$\neg(p_1 \vee p_2 \vee \ldots \vee p_n) \equiv \neg p_1 \wedge \neg p_2 \wedge \ldots \wedge \neg p_n$$

$$\neg(p_1 \wedge \neg p_2 \wedge \ldots \wedge \neg p_n) \equiv \neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_n$$

# Using De Morgan's Laws

**Example : Use De Morgan's laws to express the negation of "Miguel has a cellphone and he has a laptop computer" and "Heather will go to the concert or Steve will go to the concert".**

***Solution:***

**Miguel does not have a cellphone or he does not have a laptop computer.**

**Heather will not go to the concert and Steve will not go to the concert.**

# Constructing New Logical Equivalences

- The logical equivalence can be used to constructing new logical equivalences
  - **Reason: a proposition in a compound proposition can be replaced by a compound proposition that is logically equivalent to it without changing the truth value of the original compound proposition.**
- 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.**

$$A \equiv A_1$$
$$\vdots$$
$$A_n \equiv B$$

# Equivalence Proofs

**Example : Show that $\neg(p \vee (\neg p \wedge q))$ and $\neg p \wedge \neg q$ are logically equivalent.**

*Solution:*

$\neg(p \vee (\neg p \wedge q)) \equiv \neg p \wedge \neg(\neg p \wedge q)$ **by the De Morgan' law**

$\equiv \neg p \wedge (\neg\neg p \vee \neg q)$ **by the De Morgan' law**

$\equiv \neg p \wedge (p \vee \neg q)$ **by the double negation law**

$\equiv (\neg p \wedge p) \vee (\neg p \wedge \neg q)$ **by the distributive law**

$\equiv F \vee (\neg p \wedge \neg q)$ **by the negation law**

$\equiv \neg p \wedge \neg q$ **by the identity law**

**Example : Show that** $((P \to Q) \to R) \to ((R \to P) \to (S \to P))$

**is a tautology.**

*Solution:*

$$((p \to q) \to r) \to ((r \to p) \to (s \to p))$$

$$\equiv \neg(\neg(\neg p \vee q) \vee r) \vee (\neg(\neg r \vee p) \vee (\neg s \vee p))$$

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

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

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

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

$$\equiv T$$

# Propositional Satisfiability

- A compound proposition is satisfiable if there is an assignment of truth values to its variables that makes it true.
- A compound proposition is unsatisfiable when it is false for all assignments of truth values to its variables.
- Show that a compound proposition is satisfiable
  - Find a particular assignment of truth values that make this compound proposition true.
  - Such an assignment is called a solution of this particular satisfiability problem

**Example: Determine the satisfiability of the following compound propositions:**

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

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

# Application of Satisfiability

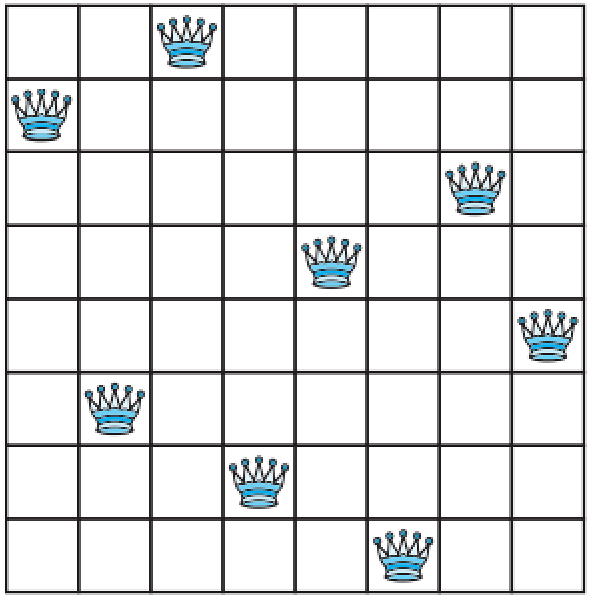
- **Many problems, in diverse areas such as robotics, software testing, artificial intelligence planning, computer-aided design, machine vision, integrated circuit design, scheduling, computer networking, and genetics, can be modeled in terms of propositional satisfiability.**

- **Examples:**
  - **The n-Quees Problem**
  - **Sudoku puzzle**

# The *n*-Quees Problem

**The *n*-Quees Problem asks for a placement of *n* queens on an *n* × *n* chessboard so that no queen can attack another queen. This means that no two queens can be placed in the same row, in the same column, or on the same diagonal.**

**Example,**



# The *n*-Quees Problem

- **Modeling as a satisfiability Problem**

**Let $p(i,j)$ denote the proposition that is true when there is a queen on the square in the $i$th row and $j$th column, and is false otherwise.**

- **Assert that no row contains more than one queen.**

$$Q_1 = \bigwedge_{i=1}^{n} \bigvee_{j=1}^{n} p(i,j) \qquad Q_2 = \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n-1} \bigwedge_{k=j+1}^{n} (\neg p(i,j) \vee \neg p(i,k)).$$

- **Assert that no column contains more than one queen.**

$$Q_3 = \bigwedge_{j=1}^{n} \bigwedge_{i=1}^{n-1} \bigwedge_{k=i+1}^{n} (\neg p(i,j) \vee \neg p(k,j)).$$

# The *n*-Quees Problem

- **Modeling as a Satisfiability Problem**

  - **Assert that no diagonal contains two queens**

$$Q_4 = \bigwedge_{i=2}^{n} \bigwedge_{j=1}^{n-1} \bigwedge_{k=1}^{\min(i-1,n-j)} (\neg p(i,j) \vee \neg p(i-k,k+j))$$

$$Q_5 = \bigwedge_{i=1}^{n-1} \bigwedge_{j=1}^{n-1} \bigwedge_{k=1}^{\min(n-i,n-j)} (\neg p(i,j) \vee \neg p(i+k,j+k)).$$

  - **the solutions of the *n*-queens problem are given by the assignments of truth values to the variables $p(i, j)$, $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, n$ that make Q true**

$$Q = Q_1 \wedge Q_2 \wedge Q_3 \wedge Q_4 \wedge Q_5$$

# Solving Satisfiability Problems

- Using this and other approaches, the number of ways $n$ queens can be placed on a chessboard so that no queen can attack another has been computed for $n \leq 27$. When $n = 8$ there are 92 such placements, while for $n = 16$ this number grows to 14,772,512.

- 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.

# Other logical operators

**Sheffer stroke |:**

$$p|q \equiv \neg(p \wedge q) \quad \textbf{NAND}$$

**Peirce arrow ↓:**

$$p \downarrow q \equiv \neg(p \vee q) \quad \textbf{NOR}$$

# The Dual of a Compound Proposition

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**.

**For example,**

(1) $S=(p \vee \neg q) \wedge r \vee T \quad S^{*}= (p \wedge \neg q) \vee r \wedge F$

(2) $S= (p \wedge q) \rightarrow (p \vee q) \equiv \neg(p \wedge q) \vee (p \vee q)$

$S^{*}= \neg(p \vee q) \wedge (p \wedge q)$

【Theorem】 let *s and t* are two compound propositions, $s \equiv t$ if and only if $s^{*} \equiv t^{*}$ .

# Functionally Complete Collection of Logical 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.**

**For example,**
$\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$, $\{\neg, \wedge, \vee\}$, $\{\neg, \wedge\}$, $\{\neg, \vee\}$, $\{\mid\}$, $\{\downarrow\}$
**are all functionally complete operators.**

# Propositional Normal Forms

**There are two types of normal forms in propositional calculus:**

**disjunctive normal form(DNF) and conjunctive normal form(CNF)**

**Why is it needed to introduce propositional normal forms?**

**Formulas can be transformed into standard forms**

- **Make identification and comparison of two formulas**
- **Become more convenient to determine whether a formula is tautology, contradiction or contingency**
- **Become more convenient to find the assignments for which the formula is true or false**
- **Can simplify the formula**

# Disjunctive normal form

A ***literal*** is a variable or its negation.

**Conjunctions with literals as conjuncts are called *conjunctive clauses (clauses)*.**

**For example,**

$$p \wedge q, \quad p \wedge \neg q, \quad \neg p \wedge q, \quad \neg p \wedge \neg q$$

**A formula is said to be in disjunctive normal form if it is written as a disjunction, in which all the terms are conjunctions of literals.**

$$(p \wedge q) \vee (p \wedge \neg q) \qquad \checkmark$$

$$p \wedge (p \vee q) \qquad \times$$

# More DNF or CNF

| | |
|---|---|
| $\neg p \vee (q \wedge \neg r)$ | **DNF** |
| $\neg p \wedge (q \vee \neg r) \wedge (\neg q \vee r)$ | **CNF** |
| $p$ | **DNF & CNF** |
| $\neg p \vee q$ | **DNF & CNF** |
| $\neg p \wedge q \wedge \neg r$ | **DNF & CNF** |

# Existence of normal form

【 Theorem 1 】 Any formula *A* is tautologically equivalent to some formula in DNF (CNF).

***Proof*:**

**Construct the truth table for the proposition. Then an equivalent proposition is the disjunction with *n* disjuncts (where *n* is the number of rows for which the formula evaluates to T). Each disjunct has m conjuncts where *m* is the number of distinct propositional variables. Each conjunct includes the positive form of the propositional variable if the variable is assigned T in that row and the negated form if the variable is assigned F in that row.  This proposition is in disjunctive normal from.**

# How to obtain normal form

**(1) Use of the following logical equivalences to eliminate $\rightarrow$,$\leftrightarrow$.**

$$p \rightarrow q \equiv \neg p \vee q$$

$$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$$

**(2) Use of the following logical equivalences to eliminate $\neg$ ,$\vee$, $\wedge$ from the scope of $\neg$ such that any $\neg$ has only an atom as its scope.**

$$\neg (p_1 \vee p_2 \vee \ldots \vee p_n) \equiv \neg p_1 \wedge \neg p_2 \wedge \ldots \wedge \neg p_n$$

$$\neg\neg p \equiv p$$

**(3) Use of the commutative laws, the distributive laws and the associative laws to obtain normal form.**

**Example : Convert the following formula into conjunctive and disjunctive normal forms.**

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

***solution:***

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

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

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

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

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

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

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

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

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

**Example : Find the assignments of *p* and *q* for which the following formula is true.**

$$(p \to q) \to p$$

***solution:***

$$(p \to q) \to p$$

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

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

$$\equiv p$$

**The assignments of *p* and *q* for which the formula is true:**

| ***p*** | ***q*** |
|---|---|
| **T** | **T** |
| **T** | **F** |

# Conjunctive normal form

- A compound proposition is in Conjunctive Normal Form (CNF) if it is a conjunction of disjunctions.
- Every proposition can be put in an equivalent CNF.
- Conjunctive Normal Form (CNF) can be obtained by eliminating implications, moving negation inwards and using the distributive and associative laws.
- Important in resolution theorem proving used in artificial Intelligence (AI).
- A compound proposition can be put in conjunctive normal form through repeated application of the logical equivalences covered earlier.

# Full disjunctive normal form

## 1. Minterm & Maxterm

**A *minterm* is a conjunctive of literals in which each variable is represented exactly once.**

**For example,**

**If a formula has the variables $p$, $q$, $r$, then $p \wedge \neg q \wedge r$ is a minterm, but $p \wedge \neg q$ and $p \wedge \neg p \wedge r$ are not.**

**Question:**

**If a formula has $n$ variables, how many minterms are there?**

## 2. Full disjunctive normal form

**If a formula is expressed as a disjunction of minterms, it is said to be in *full disjunctive normal form*.**

**For example,**

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

## 3. How to obtain full disjunctive normal form

**Any formula *A* is tautologically equivalent to a formula in full disjunctive normal form.**

**First, obtain disjunctive normal form, then use of negation law and distributive laws to obtain full disjunctive forms**

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

**Example : Convert the following formula into full disjunctive normal form.**

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

***solution:***

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

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

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

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

- **How to find the assignments for which a given formula is True?**
- **How to determine whether a given formula is tautology, contradiction or contingency**

**From the above expression, we can see**

**(1) The formula is true for the following four assignments**

| *p* | *q* | *r* |
|---|---|---|
| T | T | T |
| T | T | F |
| F | T | T |
| F | F | T |

**(2) The formula is a contingency.**

# Full disjunctive normal form from truth tables

**What is the Full DNF of the formula *f* ?**

| $p$ | $q$ | $r$ | $f$ |
|---|---|---|---|
| T | T | T | T |
| T | T | F | T |
| T | F | T | F |
| T | F | F | F |
| F | T | T | T |
| F | T | F | F |
| F | F | T | T |
| F | F | F | F |

# Analyze the minterms with two propositional variables

| $p$ | $q$ | $\neg p \wedge \neg q$ | $\neg p \wedge q$ | $p \wedge \neg q$ | $p \wedge q$ |
|---|---|---|---|---|---|
| T | T | F | F | F | T |
| T | F | F | F | T | F |
| F | T | F | T | F | F |
| F | F | T | F | F | F |

**Note:**

(1) Each minterm is true for exactly one assignment.

(2) There is exactly one minterm which truth value is T for each assignment.

(3) If $A$ and $B$ are two distinct minterms, then $A \wedge B \equiv$F.

(4) A disjunction of minterms is true only if at least one of its constituents minterms is true.

## What is the Full DNF of the formula *f* ?

| $p$ | $q$ | $r$ | $f$ |
|---|---|---|---|
| T | T | T | T |
| T | T | F | T |
| T | F | T | F |
| T | F | F | F |
| F | T | T | T |
| F | T | F | F |
| F | F | T | T |
| F | F | F | F |

$$f \equiv (p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r) \vee (\neg p \wedge q \wedge r) \vee (\neg p \wedge \neg q \wedge r)$$ **Why?**

**Example : Find the full disjunctive normal form for *f* given by the table.**

| $p$ | $q$ | $r$ | $f$ |
|---|---|---|---|
| T | T | T | T |
| T | T | F | F |
| T | F | T | T |
| T | F | F | F |
| F | T | T | F |
| F | T | F | F |
| F | F | T | T |
| F | F | F | F |

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

**Example : Find the full conjunctive normal form for *f* given by the table.**

| $p$ | $q$ | $r$ | $f$ |
|---|---|---|---|
| T | T | T | T |
| T | T | F | T |
| T | F | T | F |
| T | F | F | F |
| F | T | T | T |
| F | T | F | T |
| F | F | T | F |
| F | F | F | T |

***Solution:***

(1) $f \equiv (\neg p \vee q \vee \neg r) \wedge (\neg p \vee q \vee r) \wedge (p \vee q \vee \neg r)$

(2) **Find the full disjunctive normal form of $\neg f$**

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

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

Homework:

(1) SE： P.34 10(d), 24, 30, 40, 51, 59

EE： P.38 12(d), 28, 34, 45, 55, 63

(2)Give the simplest DNF and CNF of the following formulas:

1) $((P \to Q) \leftrightarrow (\neg Q \to \neg P)) \wedge R$

2) $(P \wedge (Q \wedge S)) \vee (\neg P \wedge (Q \wedge S))$

(3) Give the full DNF of the following formulas, Find the assignments of p , q and r for which the formula is true.

1) $(\neg R \wedge (Q \to P)) \to (P \to (Q \vee R))$

2) $(P \to (Q \wedge R)) \wedge (\neg P \to (\neg Q \wedge \neg R))$