Mathematical Logic

Contents

- Propositional Logic
  - The Language of Propositions
  - Applications
  - Logical Equivalences
- Predicate Logic
  - The Language of Quantifiers
  - Logical Equivalences
  - Nested Quantifiers
- Proofs
  - Rules of Inference
  - Proof Methods
  - Proof Strategy

Propositional Logic

- The Language of Propositions
  - Connectives
  - Truth Values
  - Truth Tables
- Applications
  - Translating English Sentences
  - System Specifications
  - Logic Puzzles
  - Logic Circuits
- Logical Equivalences
  - Important Equivalences
  - Showing Equivalence
  - Satisfiability

Propositional Logic

Contents

- Propositions
- Connectives
  - Negation
  - Conjunction
  - Disjunction
  - Implication; Contrapositive, Inverse, Converse
  - Biconditional
- 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) Trenton is the Capital of New Jersey.
  c) Toronto is the Capital of Canada.
  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*, … (or) *p*, *q*, *r*, *s*, …
  - The Proposition that is always True is denoted by **T** (1) and the Proposition that is always False is denoted by **F** (**0**).
  - Compound Propositions; Constructed from Logical Connectives and other Propositions
    - Negation $\neg$
    - Conjunction $\wedge$
    - Disjunction $\vee$
    - Implication $\rightarrow$
    - Biconditional $\leftrightarrow$

Compound Propositions: Negation

- The *Negation* (NOT) of a Proposition p is denoted by $\neg p$ and has the following 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 (AND)

- 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 (OR)

- The *disjunction* of propositions, namely: p and q is denoted by $p \vee q$ and has the following truth table:

p	q	$p \vee q$
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 $p \vee q$ denotes "I am at home or it is raining."

The Connective OR in English

- In English "OR" has Two Distinct Meanings.
  - "Inclusive OR" - "Students who have taken 1st year Computer Programming and Mathematics courses may take this DM 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 $p \vee q$ to be True, either one or both of p and q must be True.
  - "Exclusive OR" - "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**: If p denotes "I am at home." and q denotes "It is raining." then $p \rightarrow q$ denotes "If I am at home then it is raining."
- In $p \rightarrow q$, p is known as the *Hypothesis* (*Antecedent* or *Premise*) and q is known as the *Conclusion* (or *Consequence*).

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 (contd…)

- One way to view the Logical Conditional is to think of an Obligation or Contract.
  - "If I am elected, then I will lower taxes."
  - "If you get 100% on the final, then you will get an A."

- If the politician is elected and does not lower taxes, then the voters can say that he or she has broken the campaign pledge. Something similar holds for the professor. This corresponds to the case where p is true and q is false.

Different Ways of Expressing $p \to q$

- **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
- 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 $p \rightarrow q$ we can form new conditional statements .
  - $q \rightarrow p$ is the C**onverse** of $p \rightarrow q$
  - $\neg q \rightarrow \neg p$ is the **Contrapositive** of $p \rightarrow q$
  - $\neg p \rightarrow \neg q$ is the **Inverse** of $p \rightarrow q$

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.

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 the following Truth Table:

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

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

Expressing the Biconditional

- 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 (Truth or Falsify) for the Atomic Propositions.
- Columns
  - Need a Column for the Compound Proposition (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

Ex: Show that neither the Converse nor Inverse Implication are not Equivalent to Implication (use the Truth Tables).

Solution:

p	q	$\neg p$	$\neg q$	$p \to q$	$\neg p \to \neg q$	$q \to p$
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

- It 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
$\wedge$ $\vee$	2 3
$\rightarrow$ $\leftrightarrow$	4 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.

Applications of Propositional Logic

Applications of Propositional Logic:

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

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 for the shopping."
  - *p*: I go to Harry's
  - q: I go to the country.
  - *r*: I will go for the shopping.

If *p* or *q* then not *r*.

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

Example

Problem: Translate the following English sentence into the Propositional Logic:

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

One Solution: Let *a*, *c*, and *f* represent respectively "We can access the internet from campus," "We are a computer science major," and "We are a freshman."

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

System Specifications

- Computer System and Software Engineers take requirements in English and express them in a precise specification language based on logic.

Example: Express in Propositional Logic:

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

Solution: One possible solution: Let *p* denotes "The automated reply can be sent" and *q* denotes "The file system is full."

$$q \rightarrow \neg p$$

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.

Exercise: Are these specifications consistent?

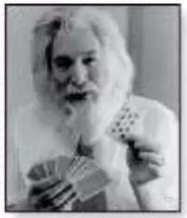
- "The diagnostic message is stored in the buffer or it is retransmitted."
- "The diagnostic message is not stored in the buffer."
- "If the diagnostic message is stored in the buffer, then it is retransmitted."

Solution: Let p denote "The diagnostic message is stored in the buffer." Let q denotes "The diagnostic message is retransmitted". The specification can be written as: $p \lor q$, $\neg p$, $p \rightarrow q$. When p is false and q is true then all three statements are true. Hence, 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 the assignment. Hence, the specification is not consistent.

Logic Puzzles



Raymond Smullyan (1919-2017)

- An island has two kinds of inhabitants, *Knights* (Truth Tellers), who always tell the truth, and *Knaves (Liars)*, 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."

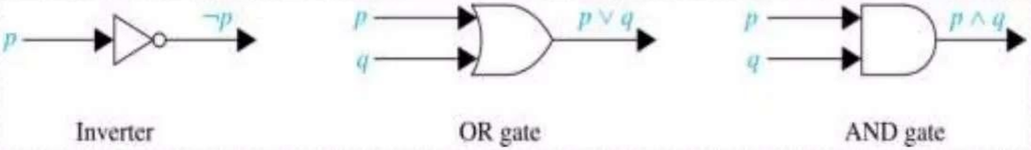
Example: What are the types of A and B?

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.

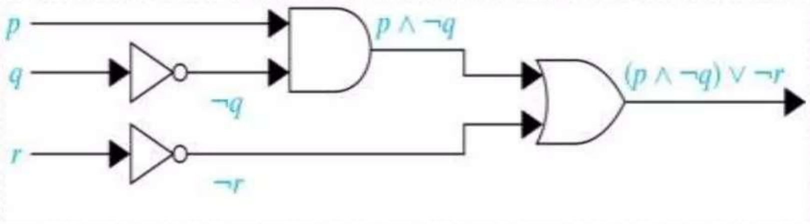
- If A is a Knight, then p is true. Since Knights tell the truth, q must also be true. Then $(p \wedge \neg q) \vee (\neg p \wedge q)$ would have to be true, but it is not. So, A is not a Knight and therefore $\neg p$ must be true.
- If A is a Knave, then B must not be a Knight since Knaves always lie.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

Logic Circuits

- Digital Electronic Circuits; each input/output signal can be viewed as a 0 or 1.
  - 0 represents **False**
  - 1 represents **True**
- Complicated circuits are constructed from three basic circuits called gates.



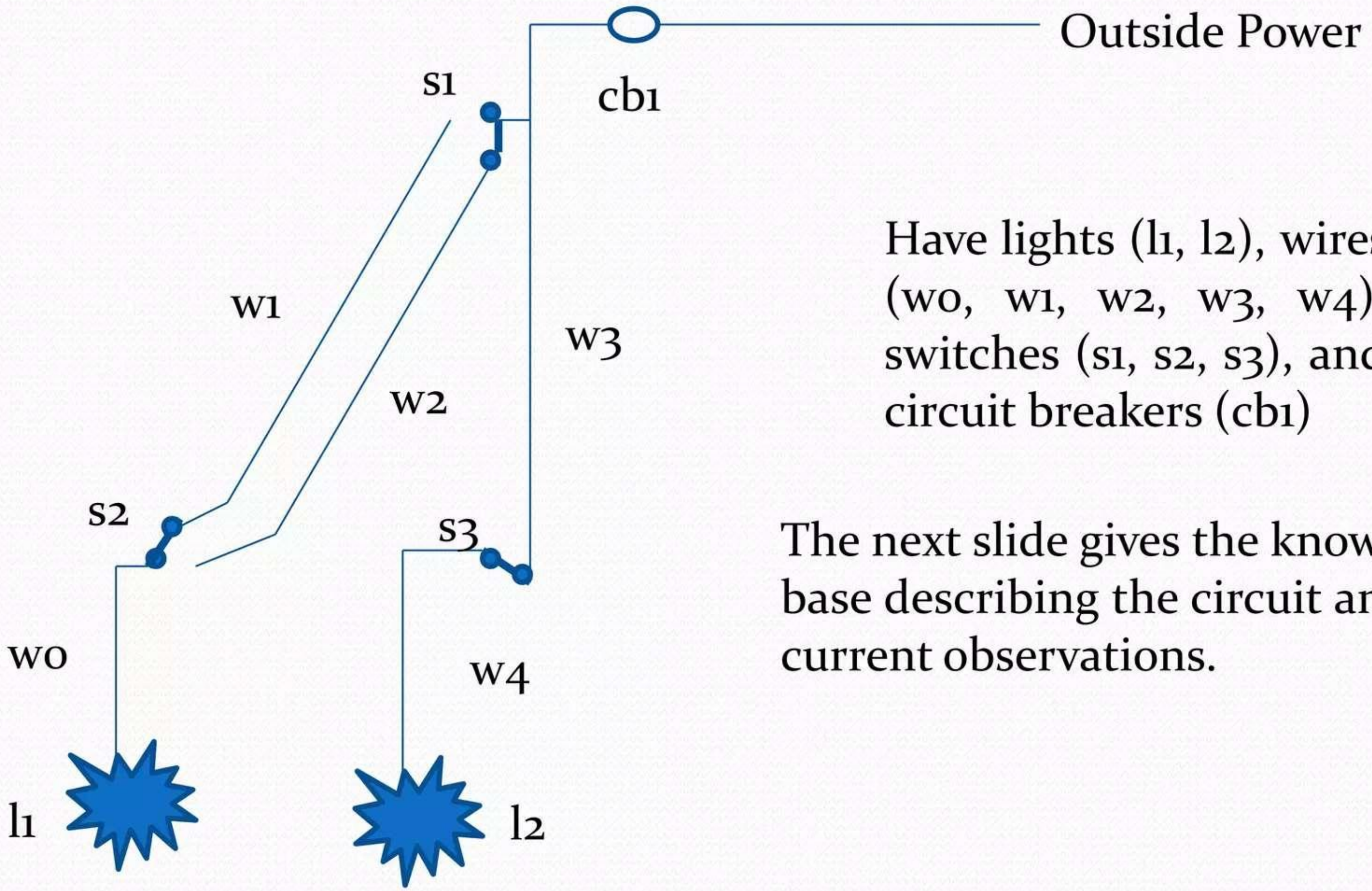
  - The inverter (**NOT gate**)takes an input bit and produces the negation of that bit.
  - The **OR gate** takes two input bits and produces the value equivalent to the disjunction of the two bits.
  - The **AND gate** takes two input bits and produces the value equivalent to the conjunction of the two bits.
- More complicated digital circuits can be constructed by combining these basic circuits to produce the desired output given the input signals by building a circuit for each piece of the output expression and then combining them. For example:



Faults Diagnosis in an Electrical System

- AI Example (Extracted From *Artificial Intelligence: Foundations of Computational Agents* by David Poole and Alan Mackworth, 2010)
- Need to represent in propositional logic the features of a piece of machinery or circuitry that are required for the operation to produce observable features. This is called the **Knowledge Base (KB)**.
- We also have observations representing the features that the system is exhibiting now.

Electrical System Diagram



Have lights (l1, l2), wires (w0, w1, w2, w3, w4), switches (s1, s2, s3), and circuit breakers (cb1)

The next slide gives the knowledge base describing the circuit and the current observations.

Representing the Electrical System using the Propositional Logic

- We need to represent our common-sense understanding of how an electrical system works in propositional logic.
- For example: "If l_1 is a light and if l_1 is receiving current, then l_1 is lit.
  - $light\_l_1 \wedge live\_l_1 \wedge ok\_l_1 \rightarrow lit\_l_1$
- Also: "If w_1 has current, and switch s_2 is in the up position, and s_2 is not broken, then w_0 has current."
  - $live\_w_1 \wedge up\_s_2 \wedge ok\_s_2 \rightarrow live\_w_0$
- This task of representing a piece of our common-sense world in logic is a common one in logic-based AI.

Knowledge Base

- live_outside
- light_l1
- light_l2
- live_w0 → live_l1
- live_w1 ∧ up_s2 ∧ ok_s2 → live_w0
- live_w2 ∧ down_s2 ∧ ok_s2 → live_w0 ←
- live_w3 ∧ up_s1 ∧ ok_s1 → live_w1
- live_w3 ∧ down_s1 ∧ ok_s1 → live_w2
- live_w4 → live_l2
- live_w3 ∧ up_s3 ∧ ok_s3 → live_w4
- live_outside ∧ ok_cb1 → live_w3
- light_l1 ∧ live_l1 ∧ ok_l1 → lit_l1
- light_l2 ∧ live_l2 ∧ ok_l2 → lit_l2

We have outside power.

Both l1 and l2 are lights.

If s2 is ok and s2 is in a down position and w2 has current, then w0 has current.

Observations (Key Takeaways)

- Observations need to be added to the Knowledge Base
  - Both Switches up
    - up_s1
    - up_s2
  - Both lights are dark
    - ¬ lit_l1
    - ¬ lit_l2

Diagnosis of Electrical System

- We assume that the components are working ok, unless we are forced to assume otherwise. These atoms are called *assumables*.
- The assumables (ok_cb1, ok_s1, ok_s2, ok_s3, ok_l1, ok_l2) represent the assumption that the switches, lights, and circuit breakers are ok.
- If the system is working correctly (all assumables are true), the observations and the knowledge base are consistent (i.e., satisfiable).
- The augmented knowledge base is clearly not consistent if the assumables are all true. The switches are both up, but the lights are not lit. Some of the assumables must then be false. This is the basis for the method to diagnose possible faults in the system.
- A diagnosis is a minimal set of assumables which must be false to explain the observations of the system.

Electrical System's Diagnostic Results

- *Artificial Intelligence: Foundations of Computational Agents (*by David Poole and Alan Mackworth, 2010)(for details on this problem and how the method of consistency based diagnosis can find possible diagnoses for an electrical system.

- This yields 7 possible faults in the system. At least one of these must hold:
  - Circuit Breaker 1 is not ok.
  - Both Switch 1 and Switch 2 are not ok.
  - Both Switch 1 and Light 2 are not ok.
  - Both Switch 2 and Switch 3 are not ok.
  - Both Switch 2 and Light 2 are not ok.
  - Both Light 1 and Switch 3 are not ok.
  - Both Light 1 and Light 2 are not ok.

Propositional Equivalences

Contents

- Tautologies, Contradictions, and Contingencies.
- Logical Equivalence
  - Important Logical Equivalences
  - Showing Logical Equivalence
- Normal Forms (*optional)*
  - Disjunctive Normal Form
  - Conjunctive Normal Form
- Propositional Satisfiability
  - Sudoku Example

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 Columns in a Truth Table giving its Truth Values agree.
- This truth table shows that $\neg p \vee q$ 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

De Morgan's Laws

Augustus De Morgan
(1806-1871)

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

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

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

Key Logical 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: $\neg(\neg p) \equiv p$
- Negation Laws: $p \vee \neg p \equiv T$, $p \wedge \neg p \equiv F$

Key Logical Equivalences (*contd…*)

- Commutative Laws: $p \vee q \equiv q \vee p\ ,\quad p \wedge q \equiv q \wedge p$
- Associative Laws: $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
  $(p \vee q) \vee r \equiv p \vee (q \vee r)$
- Distributive Laws: $(p \vee (q \wedge r)) \equiv (p \vee q) \wedge (p \vee r)$
  $(p \wedge (q \vee r)) \equiv (p \wedge q) \vee (p \wedge r)$
- Absorption Laws: $p \vee (p \wedge q) \equiv p \quad p \wedge (p \vee q) \equiv p$

More Logical Equivalences

TABLE 7 Logical Equivalences Involving Conditional Statements.

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

TABLE 8 Logical Equivalences Involving Biconditional Statements.

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

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.

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

- Keep in mind that whenever a Proposition (represented by a Propositional Variable) occurs in the Listed Equivalences, 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 $(p \wedge q) \rightarrow (p \vee q)$ is a tautology.

Solution:

$(p \wedge q) \rightarrow (p \vee q)$	$\equiv$	$\neg(p \wedge q) \vee (p \vee q)$	by truth table for $\rightarrow$
	$\equiv$	$(\neg p \vee \neg q) \vee (p \vee q)$	by the first De Morgan law
	$\equiv$	$(\neg p \vee p) \vee (\neg q \vee q)$	by associative and commutative laws laws for disjunction
	$\equiv$	$T \vee T$	by truth tables
	$\equiv$	T	by the domination law

Show that $(P \to Q) \wedge (P \to R) \equiv P \to (Q \wedge R)$

Soln:

(P→Q)∧(P→R)*(P→Q)∧(P→R)*

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

$\equiv \neg P \vee (Q \wedge R)$ (Distributive Law)

$\equiv P \to (Q \wedge R)$ (Implication equivalence)

P	Q*Q*	R*R*	P→Q	P→R	(P→Q)∧(P→R)	Q∧R	P→(Q∧R)
T	T	T	T	T	T	T	T
T	T	F	T	F	F	F	F
T	F	T	F	T	F	F	F
T	F	F	F	F	F	F	F
F	T	T	T	T	T	T	T
F	T	F	T	T	T	F	T
F	F	T	T	T	T	F	T
F	F	F	T	T	T	F	T

Disjunctive Normal Form (*optional*)

- A propositional formula is in *Disjunctive Normal Form* if it consists of a Disjunction of (1, … ,*n*) disjuncts where each disjunct consists of a Conjunction of (1, …, *m*) Atomic Formulas or the Negation of an Atomic Formula.
  - Yes $(p \wedge \neg q) \vee (\neg p \vee q)$
  - No $p \wedge (p \vee q)$
- Disjunctive Normal Form is important for Circuit Design Methods discussed later in Chapter 12.

Disjunctive Normal Form (optional)

Example: Show that every Compound Proposition can be put in Disjunctive Normal Form.

Solution: 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 Form.

Disjunctive Normal Form (optional)

Example: Find the Disjunctive Normal Form (DNF) of

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

Solution: This proposition is true when r is false or when both p and q are false.

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

Conjunctive Normal Form (optional)

- 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 AI.
- A Compound Proposition can be put in CNF through repeated application of logical equivalences covered earlier.

Conjunctive Normal Form (optional)

Example: Put the following into CNF:

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

Solution:

1. Eliminate Implication signs:

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

2. Move Negation inwards; Eliminate Double Negation:

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

3. Convert to CNF using Associative/Distributive Laws

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

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 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

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

$\bigwedge_{j=1}^{n} p_j$ 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 can be solved by assigning the numbers to each blank cell so that every row, column and the 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 $p(i, j, n)$ denotes the Proposition that is True when the number n is in the cell in the ith row and the jth column.
- There are $9\times9 \times 9 = 729$ such Propositions.
- In the sample puzzle $p(5, 1, 6)$ is true, but $p(5, j, 6)$ is False for $j = 2, 3, \ldots, 9$

Encoding (contd…)

- 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.

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

Encoding (contd…)

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

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

(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 the Satisfiability Problems as many practical problems can be translated into Satisfiability Problems.