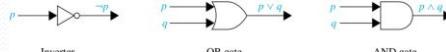


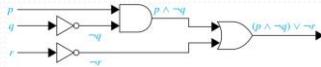
Logic Circuits

(Studied in depth in Chapter 12)

- 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:



Propositional Equivalences

Section 1.3

Section Summary

- Tautologies, Contradictions, and Contingencies.
- Logical Equivalence
 - Important Logical Equivalences
 - Showing Logical Equivalence
- Normal Forms (*optional, covered in exercises in text*)
 - 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 the columns in a truth table giving their truth values agree.
- This truth table shows $\neg p \vee q$ 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



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

Augustus De Morgan

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

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

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 (cont)

- Commutative Laws: $p \vee q \equiv q \vee p$, $p \wedge q \equiv q \wedge p$
- 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: $p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$

More Logical Equivalences

TABLE 7 Logical Equivalences Involving Conditional Statements.

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

TABLE 8 Logical Equivalences Involving Biconditional Statements.

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

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.

$$A \equiv A_1$$

⋮

$$A_n \equiv B$$

- 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:

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

Equivalence Proofs

Example: Show that $(p \wedge q) \rightarrow (p \vee q)$
is a tautology.

Solution:

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

Disjunctive Normal Form (*optional*)

- A propositional formula is in *disjunctive normal form* if it consists of a disjunction of $(1, \dots, n)$ disjuncts where each disjunct consists of a conjunction of $(1, \dots, m)$ atomic formulas or the negation of an atomic formula.
 - Yes $(p \wedge \neg q) \vee (\neg p \wedge q)$
 - No $p \wedge (p \vee q)$
- Disjunctive Normal Form is important for the circuit design methods discussed 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

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

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

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

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 artificial Intelligence (AI).
- A compound proposition can be put in conjunctive normal form through repeated application of the logical equivalences covered earlier.

Conjunctive Normal Form (optional)

Example: Put the following into CNF:

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

Solution:

1. Eliminate implication signs:

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

2. Move negation inwards; eliminate double negation:

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

3. Convert to CNF using associative/distributive laws

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

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 $p_1 \vee p_2 \vee \dots \vee p_n$

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

The Foundations: Logic and Proofs

Chapter 1, Part II: Predicate Logic

With Question/Answer Animations

Summary

- Predicate Logic (First-Order Logic (FOL), Predicate Calculus)
 - The Language of Quantifiers
 - Logical Equivalences
 - Nested Quantifiers
 - Translation from Predicate Logic to English
 - Translation from English to Predicate Logic

Predicates and Quantifiers

Section 1.4

Section Summary

- Predicates
- Variables
- Quantifiers
 - Universal Quantifier
 - Existential Quantifier
- Negating Quantifiers
 - De Morgan's Laws for Quantifiers
- Translating English to Logic
- Logic Programming (*optional*)

Propositional Logic Not Enough

- If we have:
“All men are mortal.”
“Socrates is a man.”
- Does it follow that “Socrates is mortal?”
- Can’t be represented in propositional logic. Need a language that talks about objects, their properties, and their relations.
- Later we’ll see how to draw inferences.

Introducing Predicate Logic

- Predicate logic uses the following new features:
 - Variables: x, y, z
 - Predicates: $P(x), M(x)$
 - Quantifiers (*to be covered in a few slides*):
- *Propositional functions* are a generalization of propositions.
 - They contain variables and a predicate, e.g., $P(x)$
 - Variables can be replaced by elements from their *domain*.

Propositional Functions

- Propositional functions become propositions (and have truth values) when their variables are each replaced by a value from the *domain* (or *bound* by a quantifier, as we will see later).
- The statement $P(x)$ is said to be the value of the propositional function P at x .
- For example, let $P(x)$ denote “ $x > 0$ ” and the domain be the integers. Then:
 - $P(-3)$ is false.
 - $P(0)$ is false.
 - $P(3)$ is true.
- Often the domain is denoted by U . So in this example U is the integers.

Examples of Propositional Functions

- Let “ $x + y = z$ ” be denoted by $R(x, y, z)$ and U (for all three variables) be the integers. Find these truth values:
 - $R(2, -1, 5)$
Solution: F
 - $R(3, 4, 7)$
Solution: T
 - $R(x, 3, z)$
Solution: Not a Proposition
- Now let “ $x - y = z$ ” be denoted by $Q(x, y, z)$, with U as the integers. Find these truth values:
 - $Q(2, -1, 3)$
Solution: T
 - $Q(3, 4, 7)$
Solution: F
 - $Q(x, 3, z)$
Solution: Not a Proposition

Compound Expressions

- Connectives from propositional logic carry over to predicate logic.
- If $P(x)$ denotes “ $x > 0$,” find these truth values:

$P(3) \vee P(-1)$	Solution: T
$P(3) \wedge P(-1)$	Solution: F
$P(3) \rightarrow P(-1)$	Solution: F
$P(3) \rightarrow P(-1)$	Solution: T
- Expressions with variables are not propositions and therefore do not have truth values. For example,

$$P(3) \wedge P(y)$$

$$P(x) \rightarrow P(y)$$
- When used with quantifiers (to be introduced next), these expressions (propositional functions) become propositions.



Charles Peirce (1839-1914)

Quantifiers

- We need *quantifiers* to express the meaning of English words including *all* and *some*:
 - “All men are Mortal.”
 - “Some cats do not have fur.”
- The two most important quantifiers are:
 - *Universal Quantifier*, “For all,” symbol: \forall
 - *Existential Quantifier*, “There exists,” symbol: \exists
- We write as in $\forall x P(x)$ and $\exists x P(x)$.
- $\forall x P(x)$ asserts $P(x)$ is true for every x in the *domain*.
- $\exists x P(x)$ asserts $P(x)$ is true for some x in the *domain*.
- The quantifiers are said to bind the variable x in these expressions.

Universal Quantifier

- $\forall x P(x)$ is read as “For all x , $P(x)$ ” or “For every x , $P(x)$ ”

Examples:

- 1) If $P(x)$ denotes “ $x > 0$ ” and U is the integers, then $\forall x P(x)$ is false.
- 2) If $P(x)$ denotes “ $x > 0$ ” and U is the positive integers, then $\forall x P(x)$ is true.
- 3) If $P(x)$ denotes “ x is even” and U is the integers, then $\forall x P(x)$ is false.

Existential Quantifier

- $\exists x P(x)$ is read as “For some x , $P(x)$ ”, or as “There is an x such that $P(x)$,” or “For at least one x , $P(x)$.”

Examples:

1. If $P(x)$ denotes “ $x > 0$ ” and U is the integers, then $\exists x P(x)$ is true. It is also true if U is the positive integers.
2. If $P(x)$ denotes “ $x < 0$ ” and U is the positive integers, then $\exists x P(x)$ is false.
3. If $P(x)$ denotes “ x is even” and U is the integers, then $\exists x P(x)$ is true.

Uniqueness Quantifier (*optional*)

- $\exists!x P(x)$ means that $P(x)$ is true for one and only one x in the universe of discourse.
- This is commonly expressed in English in the following equivalent ways:
 - “There is a unique x such that $P(x)$.”
 - “There is one and only one x such that $P(x)$ ”
- Examples:
 1. If $P(x)$ denotes “ $x + 1 = 0$ ” and U is the integers, then $\exists!x P(x)$ is true.
 2. But if $P(x)$ denotes “ $x > 0$,” then $\exists!x P(x)$ is false.
- The uniqueness quantifier is not really needed as the restriction that there is a unique x such that $P(x)$ can be expressed as:

$$\exists x (P(x) \wedge \forall y (P(y) \rightarrow y=x))$$

Thinking about Quantifiers

- When the domain of discourse is finite, we can think of quantification as looping through the elements of the domain.
- To evaluate $\forall x P(x)$ loop through all x in the domain.
 - If at every step $P(x)$ is true, then $\forall x P(x)$ is true.
 - If at a step $P(x)$ is false, then $\forall x P(x)$ is false and the loop terminates.
- To evaluate $\exists x P(x)$ loop through all x in the domain.
 - If at some step, $P(x)$ is true, then $\exists x P(x)$ is true and the loop terminates.
 - If the loop ends without finding an x for which $P(x)$ is true, then $\exists x P(x)$ is false.
- Even if the domains are infinite, we can still think of the quantifiers this fashion, but the loops will not terminate in some cases.

Properties of Quantifiers

- The truth value of $\exists x P(x)$ and $\forall x P(x)$ depend on both the propositional function $P(x)$ and on the domain U .
- **Examples:**
 1. If U is the positive integers and $P(x)$ is the statement " $x < 2$ ", then $\exists x P(x)$ is true, but $\forall x P(x)$ is false.
 2. If U is the negative integers and $P(x)$ is the statement " $x < 2$ ", then both $\exists x P(x)$ and $\forall x P(x)$ are true.
 3. If U consists of 3, 4, and 5, and $P(x)$ is the statement " $x > 2$ ", then both $\exists x P(x)$ and $\forall x P(x)$ are true. But if $P(x)$ is the statement " $x < 2$ ", then both $\exists x P(x)$ and $\forall x P(x)$ are false.

Precedence of Quantifiers

- The quantifiers \forall and \exists have higher precedence than all the logical operators.
- For example, $\forall x P(x) \vee Q(x)$ means $(\forall x P(x)) \vee Q(x)$
- $\forall x (P(x) \vee Q(x))$ means something different.
- Unfortunately, often people write $\forall x P(x) \vee Q(x)$ when they mean $\forall x (P(x) \vee Q(x))$.

Translating from English to Logic

Example 1: Translate the following sentence into predicate logic: “Every student in this class has taken a course in Java.”

Solution:

First decide on the domain U .

Solution 1: If U is all students in this class, define a propositional function $J(x)$ denoting “ x has taken a course in Java” and translate as $\forall x J(x)$.

Solution 2: But if U is all people, also define a propositional function $S(x)$ denoting “ x is a student in this class” and translate as $\forall x (S(x) \rightarrow J(x))$.

$\forall x (S(x) \wedge J(x))$ is not correct. What does it mean?