# 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.
  - Ex: $p \vee \neg p$
- A *contradiction* is a proposition which is always false.
  - Ex: $p \wedge \neg p$
- A *contingency* is a proposition which is neither a tautology nor a contradiction
  - Ex: $p$

| $P$ | $\neg p$ | $p \vee \neg p$ | $p \wedge \neg p$ |
|---|---|---|---|
| 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 $p \equiv q$
- 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 $\neg p \vee q \equiv p \rightarrow q$

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

# De Morgan's Laws

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

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

Augustus De Morgan
1806-1871

Show using a truth table 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 |

# Use De Morgan's laws to find the negation of the statement

- Jan is rich and happy.

*p*: Jan is rich

*q*: Jan is happy

$(p \land q)$

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

Jan is **not** rich **or not** happy.

- Carlos will bicycle or run tomorrow.

*p*: Carlos will bicycle tomorrow

*q*: Carlos will run tomorrow

$(p \lor q)$

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

Carlos will **not** bicycle **and** will **not** run tomorrow.

# Key Logical Equivalences

- Double Negation Law: $\neg(\neg p) \equiv p$
- Negation Laws: $p \vee \neg p \equiv T$ $p \wedge \neg p \equiv F$
- Identity Laws: $p \vee F \equiv p$ , $p \wedge T \equiv p$
- Domination Laws: $p \vee T \equiv T$ , $p \wedge F \equiv F$
- Idempotent laws: $p \vee p \equiv p$ , $p \wedge p \equiv p$

# 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.**

$\boxed{\begin{array}{l} p \to q \equiv \neg p \vee q \\ p \to q \equiv \neg q \to \neg p \end{array}}$

$p \vee q \equiv \neg p \to q$

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

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

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

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

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

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

**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.

$$
\begin{aligned}
A &\equiv A_1 \\
&\equiv A_2 \\
&\equiv A_3 \\
&\vdots \\
&\equiv B
\end{aligned}
$$

# 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**:

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

# DNF (*optional*)

- A propositional formula is in *disjunctive normal form* if it consists of a disjunction of conjunctive clauses
  - Yes $(p \land \neg q \land r) \lor (r \land s)$
  - No $p \land (p \lor q)$

- Disjunctive Normal Form is important for the circuit design methods discussed in Chapter 12.

# DNF (*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 from.

# DNF (*optional*)

**Example**: Find the Disjunctive Normal Form (DNF) of

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

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

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

# CNF (*optional*)

- A compound proposition is in *Conjunctive Normal Form* (CNF) if it is a conjunction of disjunctions.
  - Yes $(F \lor \neg p) \land (\neg q \lor r)$
  - No $p \lor (q \land r)$
- Every proposition can be put in an equivalent CNF, through repeated application of the logical equivalences covered earlier (eliminating implications, moving negation inwards, and using distributive/associative laws).
- Important in resolution theorem proving used in AI.

# CNF (*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 a contradiction (i.e., always false).

# 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 $p_1 \lor p_2 \lor \ldots \lor p_n$

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

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.

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

- 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.

# 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 $i$th row and the $j$th 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,…9

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | 2 | 9 | | | | 4 | | |
| 2 | | | | 5 | | | 1 | | |
| 3 | | 4 | | | | | | | |
| 4 | | | | | 4 | 2 | | | |
| 5 | 6 | | | | | | | 7 | |
| 6 | 5 | | | | | | | | |
| 7 | 7 | | | 3 | | | | | 5 |
| 8 | | 1 | | | 9 | | | | |
| 9 | | | | | | | | 6 | |

# Encoding (cont)

column

row value

- For each cell with a given value *n*, assert *p*(*i*,*j*,*n*).
- 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 (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.
  - 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.