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: \( p \lor \neg p \)
- A contradiction is a proposition which is always false.
  - Example: \( p \land \neg p \)
- A contingency is a proposition which is neither a tautology nor a contradiction, such as \( p \)

<table>
<thead>
<tr>
<th>( P )</th>
<th>( \neg p )</th>
<th>( p \lor \neg p )</th>
<th>( p \land \neg p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>
Logically Equivalent

- Two compound propositions $p$ and $q$ are logically equivalent if $p \leftrightarrow q$ is a tautology.
- We write this as $p \iff q$ 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 $\neg p \lor q$ is equivalent to $p \to q$.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$\neg p$</th>
<th>$\neg p \lor q$</th>
<th>$p \to q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
De Morgan’s Laws

\[ \neg(p \land q) \equiv \neg p \lor \neg q \]

\[ \neg(p \lor q) \equiv \neg p \land \neg q \]

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

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(p)</td>
<td>(q)</td>
<td>(\neg p)</td>
<td>(\neg q)</td>
<td>(p \lor q)</td>
<td>(\neg(p \lor q))</td>
<td>(\neg p \land \neg q)</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
Key Logical Equivalences

- **Identity Laws:** \[ p \land T \equiv p \ , \ p \lor F \equiv p \]
- **Domination Laws:** \[ p \lor T \equiv T \ , \ p \land F \equiv F \]
- **Idempotent laws:** \[ p \lor p \equiv p \ , \ p \land p \equiv p \]
- **Double Negation Law:** \[ \neg(\neg p) \equiv p \]
- **Negation Laws:** \[ p \lor \neg p \equiv T \ , \ p \land \neg p \equiv F \]
Key Logical Equivalences (cont)

- **Commutative Laws:** \( p \lor q \equiv q \lor p \), \( p \land q \equiv q \land p \)
- **Associative Laws:**
  - \( (p \land q) \land r \equiv p \land (q \land r) \)
  - \( (p \lor q) \lor r \equiv p \lor (q \lor r) \)
- **Distributive Laws:**
  - \( (p \lor (q \land r)) \equiv (p \lor q)) \land (p \lor r) \)
  - \( (p \land (q \lor r)) \equiv (p \land q) \lor (p \land r) \)
- **Absorption Laws:**
  - \( p \lor (p \land q) \equiv p \)
  - \( p \land (p \lor q) \equiv p \)
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.**

- \( p \iff q \equiv (p \to q) \land (q \to p) \)
- \( p \iff q \equiv \neg p \iff \neg q \)
- \( p \iff q \equiv (p \land q) \lor (\neg p \land \neg q) \)
- \( \neg(p \iff q) \equiv p \iff \neg q \)
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 equivalences listed earlier, it may be replaced by an arbitrarily complex compound proposition.
Equivalence Proofs

Example: Show that \( \neg(p \lor (\neg p \land q)) \)

is logically equivalent to \( \neg p \land \neg q \)

Solution:

\[
\begin{align*}
\neg(p \lor (\neg p \land q)) & \equiv \neg p \land \neg(\neg p \land q) & \text{by the second De Morgan law} \\
& \equiv \neg p \land [\neg(\neg p) \lor \neg q] & \text{by the first De Morgan law} \\
& \equiv \neg p \land (p \lor \neg q) & \text{by the double negation law} \\
& \equiv (\neg p \land p) \lor (\neg p \land \neg q) & \text{by the second distributive law} \\
& \equiv F \lor (\neg p \land \neg q) & \text{because } \neg p \land p \equiv F \\
& \equiv (\neg p \land \neg q) \lor F & \text{by the commutative law for disjunction} \\
& \equiv (\neg p \land \neg q) & \text{by the identity law for } F
\end{align*}
\]
Equivalence Proofs

**Example:** Show that \((p \land q) \rightarrow (p \lor q)\) is a tautology.

**Solution:**

\[
(p \land q) \rightarrow (p \lor q) \equiv \neg(p \land q) \lor (p \lor q)
\]

by truth table for \(\rightarrow\)

\[
\equiv (\neg p \lor \neg q) \lor (p \lor q)
\]

by the first De Morgan law

\[
\equiv (\neg p \lor p) \lor (\neg p \lor \neg q)
\]

by associative and

commutative laws

laws for disjunction

\[
\equiv T \lor T
\]

by truth tables

\[
\equiv T
\]

by the domination law
Disjunctive Normal Form (optional)

- A propositional formula is in disjunctive normal form if it consists of a disjunction of \((1, \ldots, n)\) disjuncts where each disjunct consists of a conjunction of \((1, \ldots, m)\) atomic formulas or the negation of an atomic formula.
  - Yes \((p \land \lnot q) \lor (\lnot p \lor q)\)
  - No \(p \land (p \lor q)\)

- 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.
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 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:
\[-(p \rightarrow q) \lor (r \rightarrow p)\]

Solution:
1. Eliminate implication signs:
   \[-(\neg p \lor q) \lor (\neg r \lor p)\]
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:

\[(p \lor \neg q) \land (q \lor \neg r) \land (r \lor \neg p)\]

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

\[(p \lor q \lor r) \land (\neg p \lor \neg q \lor \neg r)\]

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

\[(p \lor \neg q) \land (q \lor \neg r) \land (r \lor \neg p) \land (p \lor q \lor r) \land (\neg p \lor \neg q \lor \neg r)\]

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

\[ \bigvee_{j=1}^{n} p_j \text{ is used for } p_1 \lor p_2 \lor \ldots \lor p_n \]

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

Needed for the next example.
Sudoku

- A **Sudoku puzzle** is represented by a $9 \times 9$ grid made up of nine $3 \times 3$ 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
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 $9 \times 9 \times 9 = 729$ such propositions.
- In the sample puzzle $p(5,1,6)$ is true, but $p(5,j,6)$ is false for $j = 2,3,...9$
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.
  $$\bigwedge_{j=1}^{9} \bigwedge_{i=1}^{9} \bigvee_{n=1}^{9} p(i, j, n)$$
Encoding (cont)

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

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

(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 satisfiability problems as many practical problems can be translated into satisfiability problems.