5 Logic

In this part of the course we consider logic. Logic is used in many places in computer science including digital circuit design, relational databases, automata theory and computability, and artificial intelligence.

We start with propositional logic, using symbols to stand for things that can be either true or false. Then we consider the concept of proof in mathematics. Finally we consider some more concepts from symbolic logic.

5.1 Statements

A statement is a sentence that is definitely either true or false but not both.

For example: “Two plus two equals four.” and “Two plus two equals five.” are both statements. The sentence “He is a college student.” is not a statement, since information is missing and it could be either true or false.

A compound statement is a combination of statements using the words not, and, and or.

5.2 Operators and Truth Tables

We represent statements by boolean variables and form compound statements using logical connectives or operators. The three standard operators are not, and, and or.

If $p$ and $q$ are (compound) statements,

the negation of $p$ is not $p$, denoted $\neg p$;

the conjunction of $p$ and $q$ is $p$ and $q$, denoted $p \land q$;

the disjunction of $p$ and $q$ is $p$ or $q$, denoted $p \lor q$, which is true when at least one of $p$ and $q$ are true; and

the exclusive or of $p$ and $q$ is denoted $p \oplus q$, which is true when exactly one of $p$ and $q$ is true.

Note that English is inherently sloppy when using the “or”. So we introduced two operators: the (inclusive) or, where the two parts can happen simultaneously, and the exclusive...
or, where the two parts cannot happen simultaneously.

A truth table gives the truth value of a compound statement for every combination of truth values of its variables.

Here is the truth table for the three binary operators defined above.

<table>
<thead>
<tr>
<th>s</th>
<th>t</th>
<th>s \land t</th>
<th>s \lor t</th>
<th>s \oplus t</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Example 5.1. Here is the truth table for \( \neg p \land \neg q \).

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>\neg p</th>
<th>\neg q</th>
<th>\neg p \land \neg q</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</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>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

5.3 Logical equivalence

Two compound statements are logically equivalent if they have the same truth value for every setting of their variables. The symbol for logical equivalence is \( \equiv \).

The standard/initial method for proving logical equivalence is to construct a truth table for both sides.

Example 5.2. Show that the associative rule holds for conjunction; that is, \( (p \land q) \land r \equiv p \land (q \land r) \).

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>r</th>
<th>p \land q</th>
<th>(p \land q) \land r</th>
<th>q \land r</th>
<th>p \land (q \land r)</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</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>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
Since the two columns corresponding to the two statements are identical, we have shown that they are logically equivalent.

Example 5.3. One can show that:

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

This is called de Morgan’s law. There is another version where the \(\land\)’s and the \(\lor\)’s are interchanged:

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

A tautology is a (compound) statement that is always true. A contradiction is a (compound) statement that is always false.

For example, \(p \lor \neg p\) is a tautology.

5.4 Conditionals and Biconditionals

When we make a logical deduction, we reason from a hypothesis to a conclusion. The aim is to be able to say: “If such and such is known, then something or other must be the case.”

A conditional statement or implication is a statement of the form “If \(p\) then \(q\)”, denoted \(p \Rightarrow q\).

The truth table for implication is:

<table>
<thead>
<tr>
<th>(p)</th>
<th>(q)</th>
<th>(p \Rightarrow q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

This is sometimes translated as “\(p\) is sufficient for \(q\)”, or “\(\neg p\) is necessary for \(\neg q\)”. It is useful to remember that:

- \(p \Rightarrow q \equiv \neg p \lor q\)
- The converse of \(p \Rightarrow q\) is \(q \Rightarrow p\).
• The **contrapositive** of \( p \Rightarrow q \) is \( \neg q \Rightarrow \neg p \). The contrapositive is logically equivalent to the original.

**A biconditional statement** is a statement of the form “\( p \) if and only if \( q \)”, denoted \( p \Leftrightarrow q \).

The truth table for this is:

\[
\begin{array}{ccc}
 p & q & p \Leftrightarrow q \\
 F & F & T \\
 F & T & F \\
 T & F & F \\
 T & T & T \\
\end{array}
\]

This is sometimes translated as “\( p \) is necessary and sufficient for \( q \).”

Note that there is precedence: we assume the “not” applies to the smallest piece it can; and that the arrows apply to the largest piece they can. For example, \( p \lor \neg q \Rightarrow r \land s \) should be read as \( (p \lor (\neg q)) \Rightarrow (r \land s) \). It is common to assume that “and” has higher precedence than “or”, such as in the Java and C programming languages. However, we do not assume that here, using parentheses instead.

**Exercises**

5.1. Construct a truth table for the following compound statement:

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

5.2. Use truth tables to determine which of the following pairs of statements are logically equivalent.

(a) \( p \lor (p \land q) \) versus \( p \)

(b) \( \neg (p \lor q) \) versus \( \neg p \land \neg q \)

(c) \( (p \lor q) \lor r \) versus \( p \lor (q \lor r) \)

(d) \( (p \land q) \lor r \) versus \( p \land (q \lor r) \)

(e) \( ((\neg p \lor q) \land (p \lor \neg r)) \land (\neg p \lor \neg q) \) versus \( \neg (p \lor r) \)

5.3. Assuming \( s \) is a statement, simplify the following:

(a) \( s \land T \)

(b) \( s \lor T \)
5.4. Use De Morgan’s law to write the negation of the compound statement:

\[ 1 < x \leq 3 \]

5.5. Establish which of the following are tautologies and which are contradictions.

(a) \((p \land q) \lor (\neg p \lor (p \land \neg q))\)
(b) \((p \land \neg q) \land (\neg p \lor q)\)

5.6. Give the converse, contrapositive, and negation of: “If Sue is Luiz’s mother then Deana is his cousin.”

5.7. Construct truth tables for the following:

(a) \((p \Rightarrow q) \Rightarrow (q \Rightarrow p)\)
(b) \((p \oplus q) \Rightarrow (\neg r \iff (p \lor q))\)

5.8. Suppose we have a situation where there are \(n\) boolean variables and we want a boolean function where we specify individually the truth value for every setting of the variables. Show that it is always possible to construct such a compound statement using only \(\land\), \(\lor\), and \(\neg\) as operators. Estimate the size of your statement.

5.9. We showed in the previous question that every compound statement has an equivalent statement that uses only \(\land\), \(\lor\), and \(\neg\).

(a) Prove that every compound statement has an equivalent statement that uses only \(\land\) and \(\neg\) as operators.

(b) The **NAND** connective is defined by the following table:

\[
\begin{array}{ccc}
  s & t & s \text{ NAND } t \\
  T & T & F \\
  T & F & T \\
  F & T & T \\
  F & F & T \\
\end{array}
\]

Show that every compound statement has an equivalent statement that uses only NAND as an operator.