Propositional logic

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Course administration

Homework 1
• First homework assignment is out today will be posted on the course web page
• Due next week on Thursday

Recitations:
• today at 4:00pm SENSQ 5313
• tomorrow at 11:00 SENSQ 5313

Course web page:
http://www.cs.pitt.edu/~milos/courses/cs441/
Propositional logic: review

- **Propositional logic**: a formal language for representing knowledge and for making logical inferences.
- A **proposition** is a statement that is either true or false.
- A **compound proposition** can be created from other propositions using logical connectives.
- The **truth of a compound proposition** is defined by truth values of elementary propositions and the meaning of connectives.
- The **truth table for a compound proposition**: table with entries (rows) for all possible combinations of truth values of elementary propositions.

Compound propositions

- Let \( p: 2 \text{ is a prime} \) ..... \( T \)
  \( q: 6 \text{ is a prime} \) ..... \( F \)
- Determine the **truth value** of the following statements:
  \( \neg p: F \)
  \( p \land q: F \)
  \( p \land \neg q: T \)
  \( p \lor q: T \)
  \( p \oplus q: T \)
  \( p \rightarrow q: F \)
  \( q \rightarrow p: T \)
Constructing the truth table

- **Example:** Construct the truth table for
  \((p \rightarrow q) \land (\neg p \leftrightarrow q)\)

---

**Rows:** all possible combinations of values for elementary propositions: \(2^n\) values
Constructing the truth table

• Example: Construct the truth table for 
  \((p \rightarrow q) \land (\neg p \leftrightarrow q)\)

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

Typically the target (unknown) compound proposition and its values

Auxiliary compound propositions and their values

Constructing the truth table

• Examples: Construct a truth table for 
  \((p \rightarrow q) \land (\neg p \leftrightarrow q)\)

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>\neg p</th>
<th>p \rightarrow q</th>
<th>\neg p \leftrightarrow q</th>
<th>(p \rightarrow q) \land (\neg p \leftrightarrow q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</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>T</td>
</tr>
<tr>
<td>F</td>
<td>T</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>F</td>
<td>F</td>
</tr>
</tbody>
</table>
Computer representation of True and False

We need to encode two values True and False:

- Computers represents data and programs using 0s and 1s
- Logical truth values – True and False
- A bit is sufficient to represent two possible values:
  - 0 (False) or 1 (True)
- A variable that takes on values 0 or 1 is called a **Boolean variable**.

- **Definition:** A **bit string** is a sequence of zero or more bits. The length of this string is the number of bits in the string.

Bitwise operations

- T and F replaced with 1 and 0

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>( p \lor q )</th>
<th>( p \land q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>p</th>
<th>( \neg p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Bitwise operations

- Examples:

\[
\begin{array}{c}
1011 0011 \\
\lor 0110 1010 \\
1111 1011
\end{array}
\begin{array}{c}
1011 0011 \\
\land 0110 1010 \\
0010 0010
\end{array}
\begin{array}{c}
1011 0011 \\
\oplus 0110 1010 \\
1101 1001
\end{array}
\]

Applications of propositional logic

- Translation of English sentences
- Inference and reasoning:
  - new true propositions are inferred from existing ones
  - Used in Artificial Intelligence:
    - Rule based (expert) systems
    - Automatic theorem provers
- Design of logic circuit
Translation

Assume a sentence:
If you are older than 13 or you are with your parents then you can attend a PG-13 movie.

Parse:
• If (you are older than 13 or you are with your parents) then (you can attend a PG-13 movie)

Atomic (elementary) propositions:
– A= you are older than 13
– B= you are with your parents
– C= you can attend a PG-13 movie

• Translation: A \lor B \to C

Translation

• General rule for translation.
• Look for patterns corresponding to logical connectives in the sentence and use them to define elementary propositions.

• Example:
You can have free coffee if you are senior citizen and it is a Tuesday

\textit{Step 1 find logical connectives}
Translation

- **General rule for translation.**
  - Look for patterns corresponding to logical connectives in the sentence and use them to define elementary propositions.

- **Example:**
  
  You can have free coffee if you are senior citizen and it is a Tuesday

  *Step 1 find logical connectives*

---

Translation

- **General rule for translation.**
  - Look for patterns corresponding to logical connectives in the sentence and use them to define elementary propositions.

- **Example:**
  
  You can have free coffee if you are senior citizen and it is a Tuesday

  *Step 2 break the sentence into elementary propositions*
Translation

• General rule for translation.
• Look for patterns corresponding to logical connectives in the sentence and use them to define elementary propositions.

• Example:

You can have free coffee if you are senior citizen and it is a Tuesday

\[ a \land b \land c \]

*Step 2 break the sentence into elementary propositions*

Step 3 rewrite the sentence in propositional logic

\[ b \land c \rightarrow a \]
Translation

• Assume two elementary statements:
  – p: you drive over 65 mph ; q: you get a speeding ticket

• Translate each of these sentences to logic
  – you do not drive over 65 mph. (¬p)
  – you drive over 65 mph, but you don’t get a speeding ticket. (p ∧ ¬q)
  – you will get a speeding ticket if you drive over 65 mph. (p → q)
  – if you do not drive over 65 mph then you will not get a speeding ticket. (¬p → ¬q)
  – driving over 65 mph is sufficient for getting a speeding ticket. (p → q)
  – you get a speeding ticket, but you do not drive over 65 mph. (q ∧ ¬p)

Application: inference

Assume we know the following sentences are true:
If you are older than 13 or you are with your parents then you can attend a PG-13 movie. You are older than 13.

Translation:
• If (you are older than 13 or you are with your parents) then (you can attend a PG-13 movie). (You are older than 13).
  – A= you are older than 13
  – B= you are with your parents
  – C=you can attend a PG-13 movie
• (A ∨ B → C), A
• (A ∨ B → C) ∧ A is true
• With the help of the logic we can infer the following statement (proposition):
  – You can attend a PG-13 movie or C is True
Application: inference

The field of Artificial Intelligence:
- Builds programs that act intelligently
- Programs often rely on symbolic manipulations

Expert systems:
- Encode knowledge about the world in logic
- Support inferences where new facts are inferred from existing facts following the semantics of logic

Theorem provers:
- Encode existing knowledge (e.g. about math) using logic
- Show that some hypothesis is true

Example: MYCIN

- MYCIN: an expert system for diagnosis of bacterial infections
- It represents
  - Facts about a specific patient case
  - Rules describing relations between entities in the bacterial infection domain

<table>
<thead>
<tr>
<th>If</th>
<th>Then</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The stain of the organism is gram-positive, and</td>
<td>the identity of the organism is streptococcus</td>
</tr>
<tr>
<td>2. The morphology of the organism is coccus, and</td>
<td></td>
</tr>
<tr>
<td>3. The growth conformation of the organism is chains</td>
<td></td>
</tr>
</tbody>
</table>

- Inferences:
  - manipulates the facts and known relations to answer diagnostic queries (consistent with findings and rules)
Tautology and Contradiction

• Some propositions are interesting since their values in the truth table are always the same

Definitions:
• A compound proposition that is always true for all possible truth values of the propositions is called a tautology.
• A compound proposition that is always false is called a contradiction.
• A proposition that is neither a tautology nor contradiction is called a contingency.

Example: \( p \lor \neg p \) is a tautology.

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

Tautology and Contradiction

• Some propositions are interesting since their values in the truth table are always the same

Definitions:
• A compound proposition that is always true for all possible truth values of the propositions is called a tautology.
• A compound proposition that is always false is called a contradiction.
• A proposition that is neither a tautology nor contradiction is called a contingency.

Example: \( p \land \neg p \) is a contradiction.

<table>
<thead>
<tr>
<th>( p )</th>
<th>( \neg p )</th>
<th>( p \land \neg p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>
Equivalence

- We have seen that some of the propositions are equivalent. Their truth values in the truth table are the same.
- Example: \( p \rightarrow q \) is equivalent to \( \neg q \rightarrow \neg p \) (contrapositive)

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

- Equivalent statements are important for logical reasoning since they can be substituted and can help us to:
  1) make a logical argument and
  2) infer new propositions

Logical equivalence

**Definition:** The propositions \( p \) and \( q \) are called logically equivalent if \( p \leftrightarrow q \) is a tautology (alternately, if they have the same truth table). The notation \( p \leftrightarrow q \) denotes \( p \) and \( q \) are logically equivalent.

**Example of important equivalences**
- **DeMorgan's Laws:**
  1) \( \neg( p \lor q ) \leftrightarrow \neg p \land \neg q \)
  2) \( \neg( p \land q ) \leftrightarrow \neg p \lor \neg q \)

**Example:** Negate "The summer in Mexico is cold and sunny" with DeMorgan's Laws

**Solution:** ?
Equivalence

• **Definition**: The propositions p and q are called **logically equivalent** if p ⇔ q is a tautology (alternately, if they have the same truth table). The notation p ↔ q denotes p and q are logically equivalent.

Example of important equivalences

• **DeMorgan's Laws**:
  1) ¬( p ∨ q ) ⇔ ¬p ∧ ¬q
  2) ¬( p ∧ q ) ⇔ ¬p ∨ ¬q

Example: Negate "The summer in Mexico is cold and sunny"
with DeMorgan's Laws

Solution: "The summer in Mexico is not cold or not sunny."

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>¬p</th>
<th>¬q</th>
<th>¬(p ∨ q)</th>
<th>¬p ∧ ¬q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example of important equivalences

- **DeMorgan's Laws:**
  - 1) \( \neg(p \lor q) \iff \neg p \land \neg q \)
  - 2) \( \neg(p \land q) \iff \neg p \lor \neg q \)

To convince us that two propositions are logically equivalent
use the truth table

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

Equivalence
Equivalence

Example of important equivalences

- **DeMorgan's Laws:**
  - 1) \( \neg(p \lor q) \iff \neg p \land \neg q \)
  - 2) \( \neg(p \land q) \iff \neg p \lor \neg q \)

To convince us that two propositions are logically equivalent use the truth table

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

Important logical equivalences

- **Identity**
  - \( p \land T \iff p \)
  - \( p \lor F \iff p \)

- **Domination**
  - \( p \lor T \iff T \)
  - \( p \land F \iff F \)

- **Idempotent**
  - \( p \lor p \iff p \)
  - \( p \land p \iff p \)
Important logical equivalences

- **Double negation**
  - \( \neg(\neg p) \iff p \)

- **Commutative**
  - \( p \lor q \iff q \lor p \)
  - \( p \land q \iff q \land p \)

- **Associative**
  - \( (p \lor q) \lor r \iff p \lor (q \lor r) \)
  - \( (p \land q) \land r \iff p \land (q \land r) \)

- **Distributive**
  - \( p \lor (q \land r) \iff (p \lor q) \land (p \lor r) \)
  - \( p \land (q \lor r) \iff (p \land q) \lor (p \land r) \)

- **De Morgan**
  - \( \neg(p \lor q) \iff \neg p \land \neg q \)
  - \( \neg(p \land q) \iff \neg p \lor \neg q \)

- **Other useful equivalences**
  - \( p \lor \neg p \iff T \)
  - \( p \land \neg p \iff F \)
  - \( p \rightarrow q \iff (\neg p \lor q) \)
Using logical equivalences

- Equivalences can be used in proofs. A proposition or its part can be transformed using equivalences and some conclusion can be reached.

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

- **Proof:** (we must show \((p \land q) \rightarrow p \iff T)\)

  \[(p \land q) \rightarrow p \iff \neg(p \land q) \lor p \quad \text{Useful} \]

  - \[\iff \neg p \lor \neg q \lor p \quad \text{DeMorgan}\]
  - \[\iff \neg q \lor \neg p \lor p \quad \text{Commutative}\]
  - \[\iff \neg q \lor [\neg p \lor p] \quad \text{Associative}\]
  - \[\iff \neg q \lor [T] \quad \text{Useful}\]
  - \[\iff T \quad \text{Domination}\]

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p \land q</th>
<th>(p \land q) \rightarrow p</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>
Using logical equivalences

- Equivalences can be used in proofs. A proposition or its part can be transformed using equivalences and some conclusion can be reached.

- Example 2: Show \( (p \rightarrow q) \iff (\neg q \rightarrow \neg p) \)

Proof:
- \( (p \rightarrow q) \iff (\neg q \rightarrow \neg p) \)
- \( \iff \ ? \)
Using logical equivalences

- Equivalences can be used in proofs. A proposition or its part can be transformed using equivalences and some conclusion can be reached.

- **Example 2:** Show \((p \rightarrow q) \iff (\neg q \rightarrow \neg p)\)

  **Proof:**
  
  - \((p \rightarrow q) \iff (\neg q \rightarrow \neg p)\)
  - \(\iff \neg(\neg q) \lor (\neg p)\) Useful
  - \(\iff q \lor (\neg p)\) Double negation
  - \(\iff ?\)
Using logical equivalences

- Equivalences can be used in proofs. A proposition or its part
  can be transformed using equivalences and some conclusion
  can be reached.

- **Example 2:** Show \((p \rightarrow q) \iff (\neg q \rightarrow \neg p)\)

  **Proof:**
  
  \[
  \begin{align*}
  & (p \rightarrow q) \iff (\neg q \rightarrow \neg p) \\
  \iff & (\neg (\neg q) \lor \neg p) \quad \text{Useful} \\
  \iff & q \lor (\neg p) \quad \text{Double negation} \\
  \iff & \neg p \lor q \quad \text{Commutative} \\
  \iff & p \rightarrow q \quad \text{Useful}
  \end{align*}
  \]

  End of proof