Propositional Logic
Introduction and Syntax

Alice Gao

Lecture 2
Outline

Learning goals

Propositions and Connectives

Propositional Language

Revisiting the learning goals
Learning goals

By the end of the lecture, you should be able to

▶ Determine whether an English sentence is a proposition.
▶ Determine whether an English sentence is a simple or compound proposition.
▶ Determine whether a propositional formula is atomic and/or well-formed.
▶ Draw the parse tree of a well-formed propositional formula.
▶ Given a propositional formula with no parentheses, make it a well-formed formula by adding parentheses according to the precedence rules.
Outline

Learning goals

Propositions and Connectives

Propositional Language

Revisiting the learning goals
Propositions

A proposition is a statement that is either true or false.

Meaningless statements, commands, and questions are not propositions.
CQ on proposition
Examples of propositions

- The sum of 3 and 5 is 8.
- The sum of 3 and 5 is 35.
- Goldbach’s conjecture: Every even number greater than 2 is the sum of two prime numbers.
Examples of non-propositions

- Question: Where shall we go to eat?
- Command: Please pass the salt.
- Sentence fragment: The dogs in the park
- Non-sensical: Green ideas sleep furiously.
- Paradox: This sentence is false.
A \textbf{compound} proposition is formed by means of logical connectives.

The commonly used logical connectives are “not”, “and”, “or”, “if, then”, and “iff”.

A \textbf{simple} proposition is not compound and cannot be further divided.
Interpreting a compound proposition

To interpret a compound proposition, we need to understand the meanings of the connectives.

Let $A$ and $B$ be arbitrary propositions.

We will use $1$ and $0$ to denote true and false respectively.
"Not A" is true if and only if A is false.

<table>
<thead>
<tr>
<th>$A$</th>
<th>not $A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Conjunction

“$A$ and $B$” is true if and only if both $A$ and $B$ are true.

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$A$ and $B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Disjunction

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>$B$</td>
<td>$A$ or $B$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

“Or” may be interpreted in two ways

▶ The inclusive sense of “A or B or both”
▶ The exclusive sense of “A or B but not both”

In mathematics, the inclusive sense of “or” is commonly used.
Implication

The only circumstance in which “if A then B” is false is when A is true and B is false.

Whenever A is false, “if A then B” is vacuously true.
Equivalence

"A iff B" is the same as "if A then B, and if B then A".

iff is pronounced as if and only if.

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$A$ iff $B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
CQ on compound or simple propositions
Remarks on connectives

The arity of a connective:

▶ The negation is a unary connective. It only applies to one proposition.
▶ All other connectives are binary connectives. They apply to two propositions.

Is a connective symmetric?

▶ And, Or, and Equivalence are symmetric. The order of the two propositions does not affect the truth value of the compound proposition.
▶ Implication is not symmetric. If A then B, and if B then A have different truth values.
Outline

Learning goals

Propositions and Connectives

Propositional Language

Revisiting the learning goals
Propositional language $L^p$

The propositional language $L^p$ consists of three classes of symbols:

- **Proposition symbols:** $p, q, r, \ldots$
- **Connective symbols:** $\neg, \land, \lor, \rightarrow, \leftrightarrow$.

**Oral reading of logical connectives**

- $\neg$ not, negation
- $\land$ and, conjunction
- $\lor$ or, (inclusive) disjunction
- $\rightarrow$ if, then (imply), implication
- $\leftrightarrow$ iff (equivalent to), equivalence

- **Punctuation symbols:** ( and ).
Expressions of $L^p$

- expressions are finite strings of symbols. Examples: $p$, $pq$, $(r)$, $p \land \rightarrow q$ and $\neg(p \land q)$.
- The length of an expression is the number of occurrences of symbols in it.
- empty expression: an expression of length 0, denoted by $\lambda$.
- two expressions $u$ and $v$ are equal if they are of the same length and have the same symbols in the same order.
- an expression is read from left to right.
Expression terminologies

- $uv$ denotes the result of **concatenating** two expressions $u$, $v$ in this order. Note that $\lambda u = u \lambda = u$.

- $v$ is a **segment** of $u$ if $u = w_1vw_2$ where $u, v, w_1, w_2$ are expressions.

  $v$ is a **proper segment** of $u$ if $v$ is non-empty and $v \neq u$.

  If $u = vw$, where $u, v, w$ are expressions, then $v$ is an **initial segment** (prefix) of $u$. Similarly, $w$ is a **terminal segment** (suffix) of $u$. 
Definition \((\text{Atom}(L^p))\)

\(\text{Atom}(L^p)\) is the set of expressions of \(L^p\) consisting of a proposition symbol only.
Well-formed propositional formulas

Definition ($Form(L^p)$)

An expression of $L^p$ is a member of $Form(L^p)$ if and only if its being so follows from (1) - (3):

1. $Atom(L^p) \subseteq Form(L^p)$.
2. If $A \in Form(L^p)$, then $\neg A \in Form(L^p)$.
3. If $A, B \in Form(L^p)$, then $(A \ast B) \in Form(L^p)$ where $\ast$ is one of the four binary connectives.

Note that $Form(L^p)$ is the minimum set that satisfies the three conditions above.
CQ on the first symbol in a well-formed formula
CQ on well-formed propositional formulas
Example: Generating Formulas

The following expression is a formula.

$$((p \lor q) \rightarrow ((\neg p) \leftrightarrow (q \land r)))$$

How is it generated using the definition of well-formed propositional formulas? One can use parse trees to analyze formulas.
Example: Parse Tree

Draw the parse tree for the following formula.

$$((p \lor q) \rightarrow ((\neg p) \leftrightarrow (q \land r)))$$

Parse tree:
Draw the parse tree for the following formula.

$(((\neg p) \land q) \rightarrow (p \land (q \lor (\neg r))))$
Precedence rules: for humans

Consider the following sequence of connectives:

\[ \neg, \land, \lor, \rightarrow, \leftrightarrow \]

Each connective on the left has priority over those on the right.

Examples: Add back the brackets based on the precedence rules.

- \( \neg p \lor q \)
- \( p \land q \lor r \)
- \( p \rightarrow q \leftrightarrow p \)
- \( \neg p \rightarrow p \land \neg q \lor r \leftrightarrow q \)
By the end of the lecture, you should be able to

▶ Determine whether an English sentence is a proposition.
▶ Determine whether an English sentence is a simple or compound proposition.
▶ Determine whether a propositional formula is atomic and/or well-formed.
▶ Draw the parse tree of a well-formed propositional formula.
▶ Given a propositional formula with no parentheses, make it a well-formed formula by adding parentheses according to the precedence rules.