Outline of Lecture 2

1. Propositional Logic
2. Syntax and Meaning of Propositional Logic
Logic defines the ground rules for establishing truths.

Mathematical logic spells out these rules in complete detail, defining what constitutes a *formal proof*.

Learning mathematical logic is a good way to learn logic because it puts you on a firm foundation.

Writing formal proofs in mathematical logic is a lot like computer programming. The rules of the game are clearly defined.
Propositional logic is a language that abstracts away from content and focuses on the logical connectives.

- Uppercase letters like $P$ and $Q$ are *meta-variables* that are placeholders for propositions.
- The following rules define what is a *proposition*.
Propositional logic is a language that abstracts away from content and focuses on the logical connectives.

Uppercase letters like $P$ and $Q$ are *meta-variables* that are placeholders for propositions.

The following rules define what is a *proposition*.

- A propositional variable (lowercase letters $p$, $q$, $r$) is a proposition. These variables model true/false statements.
Propositional Logic

- Propositional logic is a language that abstracts away from content and focuses on the logical connectives.
- Uppercase letters like $P$ and $Q$ are *meta-variables* that are placeholders for propositions.
- The following rules define what is a *proposition*.
  - A propositional variable (lowercase letters $p$, $q$, $r$) is a proposition. These variables model true/false statements.
  - The negation of a proposition $P$, written $\neg P$, is a proposition.
Propositional logic is a language that abstracts away from content and focuses on the logical connectives.

- Uppercase letters like $P$ and $Q$ are *meta-variables* that are placeholders for propositions.
- The following rules define what is a *proposition*.
  - A propositional variable (lowercase letters $p$, $q$, $r$) is a proposition. These variables model true/false statements.
  - The negation of a proposition $P$, written $\neg P$, is a proposition.
  - The conjunction (and) of two propositions, written $P \land Q$, is a proposition.
  - The disjunction (or) of two propositions, written $P \lor Q$, is a proposition.
  - The conditional statement (implies), written $P \rightarrow Q$, is a proposition.
  - The Boolean values True and False are propositions.
Propositional logic is a language that abstracts away from content and focuses on the logical connectives.

Uppercase letters like $P$ and $Q$ are *meta-variables* that are placeholders for propositions.

The following rules define what is a *proposition*.

- A propositional variable (lowercase letters $p$, $q$, $r$) is a proposition. These variables model true/false statements.
- The negation of a proposition $P$, written $\neg P$, is a proposition.
- The conjunction (and) of two propositions, written $P \land Q$, is a proposition.
- The disjunction (or) of two propositions, written $P \lor Q$, is a proposition.
Propositional logic is a language that abstracts away from content and focuses on the logical connectives.

- Uppercase letters like $P$ and $Q$ are meta-variables that are placeholders for propositions.

The following rules define what is a proposition.

- A propositional variable (lowercase letters $p$, $q$, $r$) is a proposition. These variables model true/false statements.
- The negation of a proposition $P$, written $\neg P$, is a proposition.
- The conjunction (and) of two propositions, written $P \land Q$, is a proposition.
- The disjunction (or) of two propositions, written $P \lor Q$, is a proposition.
- The conditional statement (implies), written $P \implies Q$, is a proposition.

The Boolean values True and False are propositions.
Propositional Logic

- Propositional logic is a language that abstracts away from content and focuses on the logical connectives.
- Uppercase letters like P and Q are *meta-variables* that are placeholders for propositions.
- The following rules define what is a *proposition*.
  - A propositional variable (lowercase letters \( p, q, r \)) is a proposition. These variables model true/false statements.
  - The negation of a proposition \( P \), written \( \neg P \), is a proposition.
  - The conjunction (and) of two propositions, written \( P \land Q \), is a proposition.
  - The disjunction (or) of two propositions, written \( P \lor Q \), is a proposition.
  - The conditional statement (implies), written \( P \rightarrow Q \), is a proposition.
  - The Boolean values True and False are propositions.
Different authors include different logical connectives in their definitions of Propositional Logic. However, these differences are not important.

In each case, the missing connectives can be defined in terms of the connectives that are present.

For example, I left out exclusive or, $P \oplus Q$, but

\[ P \oplus Q = (P \land \neg Q) \lor \neg P \land Q \]
Propositional Logic

- How expressive is Propositional Logic?
- Can you write down the rules for Sudoku in Propositional Logic?

It's rather difficult if not impossible to express the rules of Sudoku in Propositional Logic. But Propositional Logic is a good first step towards more powerful logics.
How expressive is Propositional Logic?

Can you write down the rules for Sudoku in Propositional Logic?

It’s rather difficult if not impossible to express the rules of Sudoku in Propositional Logic.

But Propositional Logic is a good first step towards more powerful logics.
A truth assignment maps propositional variables to True or False. The following is an example:

\[ A \equiv \{ p \mapsto \text{True}, q \mapsto \text{False}, r \mapsto \text{True} \} \]

\[ A(p) = \text{True} \quad A(q) = \text{False} \quad A(r) = \text{True} \]

The meaning of a proposition is a function from truth assignments to True or False. We use the notation \([P]\) for the meaning of proposition \(P\).

\[ [p](A) = A(p) \]

\[ \neg P(A) = \begin{cases} \text{True} & \text{if } [P](A) = \text{False} \\ \text{False} & \text{otherwise} \end{cases} \]
Meaning of Propositions, cont’d

\[ [P \land Q](A) = \begin{cases} 
  \text{True} & \text{if } [P](A) = \text{True}, [Q](A) = \text{True} \\
  \text{False} & \text{otherwise}
\end{cases} \]

\[ [P \lor Q](A) = \begin{cases} 
  \text{False} & \text{if } [P](A) = \text{False}, [Q](A) = \text{False} \\
  \text{True} & \text{otherwise}
\end{cases} \]

\[ [P \rightarrow Q](A) = \begin{cases} 
  \text{False} & \text{if } [P](A) = \text{True}, [Q](A) = \text{False} \\
  \text{True} & \text{otherwise}
\end{cases} \]
Example Propositions

Suppose $A = \{ p \mapsto \text{True}, q \mapsto \text{False} \}$. 
Example Propositions

Suppose $A = \{p \mapsto \text{True}, q \mapsto \text{False}\}$.

▶ $[p](A) = \text{True}$
Example Propositions

Suppose $A = \{p \mapsto \text{True}, q \mapsto \text{False}\}$.

- $\llbracket p \rrbracket (A) = \text{True}$
- $\llbracket q \rrbracket (A) = \text{False}$

Jeremy Siek
Discrete Mathematics 9/12
Example Propositions

Suppose $A = \{ p \mapsto \text{True}, q \mapsto \text{False} \}$.

- $[p](A) = \text{True}$
- $[q](A) = \text{False}$
- $[p \land p](A) = \text{True}$
- $[p \lor q](A) = \text{True}$
- $[p \rightarrow p](A) = \text{True}$
- $[q \rightarrow p](A) = \text{True}$
- $[(p \lor q) \rightarrow q](A) = \text{False}$
Example Propositions

Suppose $A = \{p \mapsto \text{True}, q \mapsto \text{False}\}$.

- $\llbracket p \rrbracket (A) = \text{True}$
- $\llbracket q \rrbracket (A) = \text{False}$
- $\llbracket p \land p \rrbracket (A) = \text{True}$
- $\llbracket p \land q \rrbracket (A) = \text{False}$
Example Propositions

Suppose $A = \{ p \mapsto \text{True}, q \mapsto \text{False} \}$.

- $\llbracket p \rrbracket (A) = \text{True}$
- $\llbracket q \rrbracket (A) = \text{False}$
- $\llbracket p \land p \rrbracket (A) = \text{True}$
- $\llbracket p \land q \rrbracket (A) = \text{False}$
- $\llbracket p \lor q \rrbracket (A) = \text{True}$
Example Propositions

Suppose $A = \{p \mapsto \text{True}, q \mapsto \text{False}\}$.

- $[p](A) = \text{True}$
- $[q](A) = \text{False}$
- $[p \land p](A) = \text{True}$
- $[p \land q](A) = \text{False}$
- $[p \lor q](A) = \text{True}$
- $[p \rightarrow p](A) = \text{True}$
- $[q \rightarrow p](A) = \text{True}$
- $[(p \lor q) \rightarrow q](A) = \text{False}$
Example Propositions

Suppose $A = \{ p \mapsto \text{True}, q \mapsto \text{False} \}$.

- $\llbracket p \rrbracket(A) = \text{True}$
- $\llbracket q \rrbracket(A) = \text{False}$
- $\llbracket p \land p \rrbracket(A) = \text{True}$
- $\llbracket p \land q \rrbracket(A) = \text{False}$
- $\llbracket p \lor q \rrbracket(A) = \text{True}$
- $\llbracket p \rightarrow p \rrbracket(A) = \text{True}$
- $\llbracket q \rightarrow p \rrbracket(A) = \text{True}$
- $\llbracket p \rightarrow q \rrbracket(A) = \text{False}$
- $\llbracket (p \lor q) \rightarrow q \rrbracket(A) = \text{False}$
Example Propositions

Suppose \( A = \{ p \mapsto \text{True}, q \mapsto \text{False} \} \).

\[
\begin{align*}
\lbrack p \rbrack(A) &= \text{True} \\
\lbrack q \rbrack(A) &= \text{False} \\
\lbrack p \land p \rbrack(A) &= \text{True} \\
\lbrack p \land q \rbrack(A) &= \text{False} \\
\lbrack p \lor q \rbrack(A) &= \text{True} \\
\lbrack p \rightarrow p \rbrack(A) &= \text{True} \\
\lbrack q \rightarrow p \rbrack(A) &= \text{True} \\
\lbrack p \rightarrow q \rbrack(A) &= \text{False}
\end{align*}
\]
Suppose \( A = \{ p \mapsto \text{True}, q \mapsto \text{False} \} \).

- \( [p](A) = \text{True} \)
- \( [q](A) = \text{False} \)
- \( [p \land p](A) = \text{True} \)
- \( [p \land q](A) = \text{False} \)
- \( [p \lor q](A) = \text{True} \)
- \( [p \rightarrow p](A) = \text{True} \)
- \( [q \rightarrow p](A) = \text{True} \)
- \( [p \rightarrow q](A) = \text{False} \)
- \( [(p \lor q) \rightarrow q](A) = \text{False} \)
Tautologies

Definition
A tautology is a proposition that is true in any truth assignment.

Examples:
- \( p \implies p \)
- \( q \lor \neg q \)
- \( (p \land q) \implies (p \lor q) \)

There are two ways to show that a proposition is a tautology:

1. Check the meaning of the proposition for every possible truth assignment. This is called model checking.
2. Construct a proof that the proposition is a tautology.
One way to simplify the checking is to only consider truth assignments that include the variables that matter. For example, to check $p \rightarrow p$, we only need to consider two truth assignments.

1. $A_1 = \{p \mapsto \text{True}\}$, $[p \rightarrow p](A_1) = \text{True}$
2. $A_2 = \{p \mapsto \text{False}\}$, $[p \rightarrow p](A_2) = \text{True}$

However, in real systems there are many variables, and the number of possible truth assignments grows quickly: it is $2^n$ for $n$ variables.

There are many researchers dedicated to discovering algorithms that speed up model checking.
Stuff to Remember

Propositional Logic:

- The kinds of propositions.
- The meaning of propositions.
- How to check that a proposition is a tautology.