CAS 701 Fall 2005

02 Propositional Logic

Instructor: W. M. Farmer

Revised: 25 September 2005

What is Propositional Logic?

- **Propositional logic** is the study of the truth or falsehood of **propositions** or **sentences** constructed using truth-functional **connectives**.
  - Also called **sentential logic**.
  - Began with the work of the Stoic philosophers, particularly Chrysippus, in the late 3rd century BCE.

- Most other logics are extensions of propositional logic.

- Applications:
  - Logical arguments.
  - Logical circuits (e.g., electronic circuits).
  - Boolean constraint modeling.

Syntax

- A **language** of proposition logic is a pair $L = (\mathcal{A}, \mathcal{B})$ where:
  - $\mathcal{A}$ is a set of constants called **propositional symbols** or **propositional letters**.
  - $\mathcal{B}$ is a set of 0-ary, unary, and binary constructors called **propositional connectives**.

- A **formula** of L is a string of symbols inductively defined by the following formation rules:
  1. Each $p \in \mathcal{A}$ is a formula of L.
  2. If $C_0, C_1, C_2 \in \mathcal{B}$ are 0-ary, unary, and binary, respectively, and A_1, A_2 are formulas of L, then C_0, $(C_1\ A_1)$, and $(A_1\ C_2\ A_2)$ are formulas of L.

- Common propositional connectives: T, F (0-ary); $\neg$ (unary); $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$, $|$ (binary).

An Example Language

- Let $L_0 = (\mathcal{A}, \mathcal{B})$ be the propositional language where:

  $\mathcal{A} = \{p_0, p_1, p_2, \ldots\}$.

  $\mathcal{B} = \{\neg, \Rightarrow\}$.

- The following abbreviations are employed:

T	denotes	$(p_0 \Rightarrow p_0)$.
F	denotes	$(\neg \mathsf{T})$.
$(A \vee B)$	denotes	$((\neg A) \Rightarrow B)$.
$(A \wedge B)$	denotes	$(\neg((\neg A) \vee (\neg B)))$.
$(A \Leftrightarrow B)$	denotes	$((A \Rightarrow B) \wedge (B \Rightarrow A))$.
$(A \mid B)$	denotes	$(\neg(A \wedge B))$.

Meaning of Propositional Connectives

- Each n-ary propositional connective denotes an assigned n-ary truth function.

- Examples:

	T
	t

	F
	f

p	$(\neg p)$
t	f
f	t

p	q	$(p \wedge q)$
t	t	t
t	f	f
f	t	f
f	f	f

p	q	$(p \vee q)$
t	t	t
t	f	t
f	t	t
f	f	f

p	q	$(p \Rightarrow q)$
t	t	t
t	f	f
f	t	t
f	f	t

p	q	$(p \Leftrightarrow q)$
t	t	t
t	f	f
f	t	f
f	f	t

p	q	$(p \mid q)$
t	t	f
t	f	t
f	t	t
f	f	t

- A set $\mathcal{C}$ of proposition connectives is **complete** if every truth function can be represented by a formula using only members of $\mathcal{C}$. For example, $\{\neg, \Rightarrow\}$ and $\{|\}$ are complete.

Semantics

- Let $L = (\mathcal{A}, \mathcal{B})$ be a language of propositional logic, and for each $C \in \mathcal{B}$, let f_C be its assigned truth function.

- A **model** for L is an (interpretation) function I that assigns a truth value in $\{\mathsf{t}, \mathsf{f}\}$ to each $p \in \mathcal{A}$.

- The **valuation function** for I is the function V that maps formulas of L to $\{\mathsf{t}, \mathsf{f}\}$ and satisfies the following conditions:

  1. If $p \in \mathcal{A}$, then $V(p) = I(p)$.
  2. If $C \in \mathcal{B}$ is 0-ary, then $V(C) = f_C$.
  3. If $C \in \mathcal{B}$ is unary and A is a formula of L, then $V((C\ A)) = f_C(V(A))$.
  4. If $C \in \mathcal{B}$ is binary and A, B are formulas of L, then $V((A\ C\ B)) = f_C(V(A), V(B))$.

Truth Tables

- Truth tables can be used to analyze the meaning of propositional formulas.

- Example (**Rule of Contraposition**):

p	q	$((p \Rightarrow q) \Leftrightarrow ((\neg q) \Rightarrow (\neg p)))$					
t	t	t	t	f	t	f	t
t	f	f	t	t	f	f	t
f	t	t	t	f	t	t	t
f	f	t	t	t	t	t	t

- A propositional formula A is a **tautology** and is **valid** if all of the final entries in the truth table for A are t.

- A propositional formula A is **satisfiable** if some of the final entries in the truth table for A are t.

- The validity of propositional formulas can be decided with truth tables—hence propositional logic is **decidable**!

Laws of Propositional Logic

- The **laws of propositional logic** are fundamental laws of most other common logics.

- Examples:
  - Law of Double Negation.
  - Law of Excluded Middle.
  - Law of Contraposition.
  - De Morgan's Laws.
  - Associative, commutative, and distributive laws.
  - Idempotent, identity, domination, and absorption laws.

A Hilbert-Style Proof System

Let **H** be the following Hilbert-style proof system for L_0:

- The **logical axioms** of **H** are all formulas of L_0 that are instances of the following three schemas:

  A1: $(A \Rightarrow (B \Rightarrow A))$.
  A2: $((A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C)))$.
  A3: $((\neg A \Rightarrow \neg B) \Rightarrow (B \Rightarrow A))$.

- The single **rule of inference** of **H** is **modus ponens**:

  MP: From A and $(A \Rightarrow B)$, infer B.

Metatheorems of Propositional Logic

- **Deduction Theorem**. $\Sigma \cup \{A\} \vdash_{\mathbf{H}} B$ implies $\Sigma \vdash_{\mathbf{H}} A \Rightarrow B$.

- **Soundness Theorem**. $\Sigma \vdash_{\mathbf{H}} A$ implies $\Sigma \models A$.

- **Completeness Theorem**. $\Sigma \models A$ implies $\Sigma \vdash_{\mathbf{H}} A$.

- **Soundness and Completeness Theorem (second form)**. Σ is consistent in **H** iff Σ is satisfiable.

- **Compactness Theorem**. If Σ is finitely satisfiable, then Σ is satisfiable.