CAS 701

Logic and Discrete Mathematics

Fall 2017

3 Propositional Logic

William M. Farmer

Department of Computing and Software
McMaster University

October 16, 2017

What is Logic?

1. The study of the principles underlying sound reasoning.
   - Central idea: logical consequence.
2. The branch of mathematics underlying mathematical reasoning and computation.

Fundamental Distinctions

Logic makes several fundamental distinctions:

- Syntax vs. semantics.
- Language vs. metalanguage.
- Theory vs. model.
- Truth vs. proof.

What is a Logic?

- Informally, a logic is a system of reasoning.
- Formally, a logic is a family of formal languages with:
  1. A common syntax.
  2. A common semantics.
  3. A notion of logical consequence.
- A logic may include one or more proof systems for mechanically deriving that a given formula is a logical consequence of a given set of formulas.
- Examples:
  - Propositional logic.
  - First-order logic.
  - Simple type theory (higher-order logic).

What is Propositional Logic?

- Propositional logic is the study of the truth or falsehood of propositional formulas (or propositions) formed using propositional 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.
- Main applications:
  - Basic logical reasoning.
  - Design of logical circuits.
  - Solving problems by encoding them as satisfiability (SAT) or tautology (TAUT) problems.
- We will develop propositional logic following Capter 3 of J. H. Gallier, *Logic for Computer Science*, Dover, 2015.

Syntax

- Let $\mathbf{PS} = \{P_1, P_2, \ldots\}$ be a countable set of symbols calle propositional symbols.
  - These are also called propositional variables or propositional letters.
- Let $\bot$ and $\top$ be propositional constants.
- Let $\neg$, $\wedge$, $\vee$, and $\supset$ be propositional connectives.
- The set PROP of propositional formulas is the set of strings defined inductively by:
  1. Each $P_i \in \mathbf{PS}$ is in PROP.
  2. $\bot$ and $\top$ are in PROP.
  3. If A and B are in PROP, then $\neg A$, $(A \wedge B)$, $(A \vee B)$, and $(A \supset B)$ are also in PROP.
- PROP is defined by a more formal inductive closure (i.e., inductive type) in Gallier 2015.

Semantics [1/3]

- Let BOOL $= \{\mathbf{T}, \mathbf{F}\}$ be the set of standard truth values.
- The propositional constants $\bot$ and $\top$ are interpreted by **F** and **T**, resprectively.
- Each propositional connective X is interpreted as a function H_X on BOOL as shown by the following truth table:

P	Q	$H_\neg(P)$	$H_\wedge(P,Q)$	$H_\vee(P,Q)$	$H_\supset(P,Q)$
T	**T**	**F**	**T**	**T**	**T**
T	**F**	**F**	**F**	**T**	**F**
F	**T**	**T**	**F**	**T**	**T**
F	**F**	**T**	**F**	**F**	**T**

Semantics [2/3]

- A truth assignment or valuation is a total function $v : \mathbf{PS} \rightarrow \mathrm{BOOL}$.
- Every valuation v uniquely extends to a total function $\hat{v} : \mathrm{PROP} \rightarrow \mathrm{BOOL}$ that satisfies the following conditions for all $A, B \in \mathrm{PROP}$:
  1. $\hat{v}(\bot) = \mathbf{F}$.
  2. $\hat{v}(\top) = \mathbf{T}$.
  3. $\hat{v}(P) = v(P)$ for all $P \in \mathbf{PS}$.
  4. $\hat{v}(\neg A) = H_{\neg}(\hat{v}(A))$.
  5. $\hat{v}(A \wedge B) = H_{\wedge}(\hat{v}(A), \hat{v}(B))$.
  6. $\hat{v}(A \vee B) = H_{\vee}(\hat{v}(A), \hat{v}(B))$.
  7. $\hat{v}(A \supset B) = H_{\supset}(\hat{v}(A), \hat{v}(B))$.

Semantics [3/3]

- Let v be a valuation, $A \in \mathrm{PROP}$, and $\Gamma \subseteq \mathrm{PROP}$.
- v satisfies A, written $v \models A$, if $\hat{v}(A) = \mathbf{T}$.
- v satisfies Γ if $v \models B$ for all $B \in \Gamma$.
- A is satisfiable if $v \models A$ for some valuation v.
- A is valid, written $\models A$, if $v \models A$ for all valuations v.
  - A valid propositional formula is called a tautology.
- Lemmas. A is valid iff $\neg A$ is unsatisfiable.
- Truth tables can be used to decide whether A is satisfiable, unsatisfiable, valid, or invalid (falsifiable).
- A is a semantic consequence of Γ, written $\Gamma \models A$, if, for all valuations v, $v \models \Gamma$ implies $v \models A$.
  - Semantic consequence is a form of logical consequence.

Complete Sets of Propositional Connectives

- A set $\mathcal{C}$ of propositional connectives is complete if every truth function can be represented by a propositional formula formed using only members of $\mathcal{C}$.
- Examples of complete sets of propositional connectives:
  - $\{\neg, \wedge\}$.
  - $\{\neg, \vee\}$.
  - $\{\neg, \supset\}$.
  - $\{\mathsf{nand}\}$ (nand is known as the Sheffer stroke).
  - $\{\mathsf{nor}\}$ (nor is known as the Peirce arrow).

Sequents

- A sequent is an ordered pair $\{\Gamma, \Delta\}$ where $\Gamma = A_1, \ldots, A_m$ and $\Delta = B_1, \ldots, B_n$ are finite (possibly empty) sequences of proprositional formulas in PROP.
  - Γ is called the antecedent and Δ the succedent.
- A sequent $\{\Gamma, \Delta\}$ is written as $\Gamma \to \Delta$.
- A valuation v satisfies a sequent

$$A_1, \ldots, A_m \to B_1, \ldots, B_n$$

  if

$$v \models (A_1 \wedge \cdots \wedge A_m) \supset (B_1 \vee \cdots \vee B_n).$$

- A sequent $\Gamma \to \Delta$ is satisfiable [valid] if $v \models \Gamma \to \Delta$ for some [all] valuations v.

Gentzen System G'

The Gentzen system G' is a proof system whose formulas are sequents where:

1. The axioms of G' are sequents $\Gamma \rightarrow \Delta$ such that Γ and Δ contain a common propositional formula.
2. G' has the following rules of inference:

$$\frac{\Gamma, A, B, \Delta \rightarrow \Lambda}{\Gamma, A \wedge B, \Delta \rightarrow \Lambda}\ (\wedge : \text{left}) \qquad \frac{\Gamma \rightarrow \Delta, A, \Lambda \quad \Gamma \rightarrow \Delta, B, \Lambda}{\Gamma \rightarrow \Delta, A \wedge B, \Lambda}\ (\wedge : \text{right})$$

$$\frac{\Gamma, A, \Delta \rightarrow \Lambda \quad \Gamma, B, \Delta \rightarrow \Lambda}{\Gamma, A \vee B, \Delta \rightarrow \Lambda}\ (\vee : \text{left}) \qquad \frac{\Gamma \rightarrow \Delta, A, B, \Lambda}{\Gamma \rightarrow \Delta, A \vee B, \Lambda}\ (\vee : \text{right})$$

$$\frac{\Gamma, \Delta \rightarrow A, \Lambda \quad B, \Gamma, \Delta \rightarrow \Lambda}{\Gamma, A \supset B, \Delta \rightarrow \Lambda}\ (\supset : \text{left}) \qquad \frac{A, \Gamma \rightarrow B, \Delta, \Lambda}{\Gamma \rightarrow \Delta, A \supset B, \Lambda}\ (\supset : \text{right})$$

$$\frac{\Gamma, \Delta \rightarrow A, \Lambda}{\Gamma, \neg A, \Delta \rightarrow \Lambda}\ (\neg : \text{left}) \qquad \frac{A, \Gamma \rightarrow \Delta, \Lambda}{\Gamma \rightarrow \Delta, \neg A, \Lambda}\ (\neg : \text{right})$$

Proofs and Counterexamples

- The set of deduction trees of G' is the least set of trees containing all one-node trees labeled by sequents and closed under the rules of inference of G'.
- The conclusion of a deduction tree is the sequent that labels the root of the tree.
- A deduction tree of G' is a proof tree if all the leaves of the tree are labeled by an axiom of G'.
- A deduction tree of G' is a counterexample tree if some leaf of the tree is labeled by a sequent $\Gamma \rightarrow \Delta$ where Γ and Δ are disjoint sequences of propositional symbols.
- A sequent $\Gamma \rightarrow \Delta$ is provable in G', written $\vdash \Gamma \rightarrow \Delta$, if there is a proof tree whose conclusion is $\Gamma \rightarrow \Delta$.

Metatheorems about G'

- Theorem (Soundness). $\vdash \Gamma \rightarrow \Delta$ implies $\models \Gamma \rightarrow \Delta$.
- Theorem (Search Procedure). There is a procedure that terminates on every input sequent $\Gamma \rightarrow \Delta$. If $\Gamma \rightarrow \Delta$ is valid, the procedure produces a proof tree for $\Gamma \rightarrow \Delta$. If $\Gamma \rightarrow \Delta$ is invalid, it produces a counterexample tree the encodes all that falsifying valuations for $\Gamma \rightarrow \Delta$.
- Corollary (Completeness). $\models \Gamma \rightarrow \Delta$ implies $\vdash \Gamma \rightarrow \Delta$.
- Corollary (Decidability). The satisfiabilty and tautology problems for for PROP are decidable.
  - The satisfiablity and tautology problems for PROP are NP-complete.