Propositional Logic

(Based on [Gallier 1986], [Goubault-Larrecq and Mackie 1997], and [Huth and Ryan 2004])

Yih-Kuen Tsay

Department of Information Management
National Taiwan University

Introduction

- Logic concerns two concepts:
  - truth (in a specific or general context)
  - provability (of truth from assumed truth)
- *Formal (symbolic) logic* approaches logic by rules for manipulating symbols:
  - syntax rules: for writing statements (or formulae).
    (There are also semantic rules determining whether a statement is true or false in a context or mathematical structure.)
  - inference rules: for obtaining true statements from other true statements.
- We shall introduce two main branches of formal logic:
  - *propositional logic*
  - *first-order logic* (predicate logic/calculus)
- This lecture covers propositional logic.

Why We Need Logic

- Correctness of software hinges on a precise statement of its requirements.
- Logical formulae give the most precise kind of statements about software requirements.
- The fact that "a software program satisfies a requirement" is very much the same as "a mathematical structure satisfies a logical formula":

$$prog \models req \quad \text{vs.} \quad M \models \varphi$$

- To prove (formally verify) that a software program is correct, one may utilize the kind of inferences seen in formal logic.
- The verification may be done manually, semi-automatically, or fully automatically.

Propositions

- A *proposition* is a statement that is either *true* or *false* such as the following:
  - Leslie is a teacher.
  - Leslie is rich.
  - Leslie is a pop singer.
- Simplest (atomic) propositions may be combined to form compound propositions:
  - Leslie is *not* a teacher.
  - *Either* Leslie is not a teacher *or* Leslie is not rich.
  - *If* Leslie is a pop singer, *then* Leslie is rich.

Inferences

- We are given the following assumptions:
  - Leslie is a teacher.
  - Either Leslie is not a teacher or Leslie is not rich.
  - If Leslie is a pop singer, then Leslie is rich.
- We wish to conclude the following:
  - Leslie is not a pop singer.
- The above process is an example of *inference* (deduction). Is it correct?

Symbolic Propositions

- Propositions are represented by *symbols*, when only their truth values are of concern.
  - P: Leslie is a teacher.
  - Q: Leslie is rich.
  - R: Leslie is a pop singer.
- Compound propositions can then be more succinctly written.
  - *not* P: Leslie is not a teacher.
  - *not* P *or not* Q: Either Leslie is not a teacher or Leslie is not rich.
  - R *implies* Q: If Leslie is a pop singer, then Leslie is rich.

Symbolic Inferences

- We are given the following assumptions:
  - *P* (Leslie is a teacher.)
  - *not P or not Q* (Either Leslie is not a teacher or Leslie is not rich.)
  - *R implies Q* (If Leslie is a pop singer, then Leslie is rich.)
- We wish to conclude the following:
  - *not R* (Leslie is not a pop singer.)
- Correctness of the inference may be checked by asking:
  - Is (*P and* (*not P or not Q*) *and* (*R implies Q*)) *implies* (*not R*) a tautology (valid formula)?
  - Or, is (*A and* (*not A or not B*) *and* (*C implies B*)) *implies* (*not C*) a tautology (valid formula)?

Propositional Logic: Syntax

- Vocabulary:
  - A countable set $\mathcal{P}$ of *proposition symbols* (variables): $P, Q, R, \ldots$ (also called *atomic propositions*);
  - *Logical connectives* (operators): $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$ and sometimes the constant $\perp$ (*false*);
  - Auxiliary symbols: "(", ")".
- How to read the logical connectives:
  - $\neg$ (negation): not
  - $\wedge$ (conjunction): and
  - $\vee$ (disjunction): or
  - $\rightarrow$ (implication): implies (or if . . . , then . . .)
  - $\leftrightarrow$ (equivalence): is equivalent to (or if and only if)
  - $\perp$ (*false* or bottom): false (or bottom)

Propositional Logic: Syntax (cont.)

- *Propositional Formulae*:
  - Any $A \in \mathcal{P}$ is a formula and so is $\perp$ (these are the "atomic" formula).
  - If A and B are formulae, then so are $\neg A$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, and $(A \leftrightarrow B)$.
- A is called a *subformula* of $\neg A$, and A and B subformulae of $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, and $(A \leftrightarrow B)$.
- Precedence (for avoiding excessive parentheses):
  - $A \wedge B \rightarrow C$ means $((A \wedge B) \rightarrow C)$.
  - $A \rightarrow B \vee C$ means $(A \rightarrow (B \vee C))$.
  - $A \rightarrow B \rightarrow C$ means $(A \rightarrow (B \rightarrow C))$.
  - More about this later ...

About Boolean Expressions

- *Boolean expressions* are essentially propositional formulae, though they may allow more things as atomic formulae.
- Boolean expressions in various styles:
  - $(x \vee y \vee \overline{z}) \wedge (\overline{x} \vee \overline{y}) \wedge x$
  - $(x + y + \overline{z}) \cdot (\overline{x} + \overline{y}) \cdot x$
  - $(a \vee b \vee \overline{c}) \wedge (\overline{a} \vee \overline{b}) \wedge a$
  - etc.
- Propositional formula: $(P \vee Q \vee \neg R) \wedge (\neg P \vee \neg Q) \wedge P$

Propositional Logic: Semantics

- The meanings of propositional formulae may be conveniently summarized by the truth table:

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

  The meaning of $\perp$ is always F (false).
- There is an implicit inductive definition in the table. We shall try to make this precise.

Truth Assignment and Valuation

- The semantics of propositional logic assigns a truth function to each propositional formula.
- Let $BOOL$ be the set of truth values $\{T, F\}$.
- A *truth assignment* (valuation) is a function from $\mathcal{P}$ (the set of proposition symbols) to $BOOL$.
- Let $PROPS$ be the set of all propositional formulae.
- A truth assignment v may be extended to a *valuation* function $\hat{v}$ from $PROPS$ to $BOOL$ as follows:

Truth Assignment and Valuation (cont.)

$$
\begin{array}{lcl}
\hat{v}(\bot) & = & F \\
\hat{v}(P) & = & v(P) \;\; \text{for all } P \in \mathcal{P} \\
\hat{v}(P) & = & \text{as defined by the table below, otherwise}
\end{array}
$$

$\hat{v}(A)$	$\hat{v}(B)$	$\hat{v}(\neg A)$	$\hat{v}(A \wedge B)$	$\hat{v}(A \vee B)$	$\hat{v}(A \rightarrow B)$	$\hat{v}(A \leftrightarrow B)$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

So, the truth value of a propositional formula is completely determined by the truth values of its subformulae.

Truth Assignment and Satisfaction

- We say $v \models A$ (*v satisfies A*) if $\hat{v}(A) = T$.
- So, the symbol $\models$ denotes a binary relation, called *satisfaction*, between truth assignments and propositional formulae.
- $v \not\models A$ (*v falsifies A*) if $\hat{v}(A) = F$.

Satisfaction

- Alternatively (in a more generally applicable format), the satisfaction relation $\models$ may be defined as follows:

$$
\begin{array}{lcl}
v \not\models \bot & & \\
v \models P & \iff & v(P) = T, \quad \text{for all } P \in \mathcal{P} \\
v \models \neg A & \iff & v \not\models A \text{ (it is \textit{not} the case that } v \models A) \\
v \models A \wedge B & \iff & v \models A \textit{ and } v \models B \\
v \models A \vee B & \iff & v \models A \textit{ or } v \models B \\
v \models A \rightarrow B & \iff & v \not\models A \textit{ or } v \models B \\
v \models A \leftrightarrow B & \iff & (v \models A \textit{ and } v \models B) \\
 & & \textit{or } (v \not\models A \textit{ and } v \not\models B)
\end{array}
$$

Object vs. Meta Language

- The language that we study is referred to as the *object* language.
- The language that we use to study the object language is referred to as the *meta* language.
- For example, *not*, *and*, and *or* that we used to define the satisfaction relation $\models$ are part of the meta language.

Satisfiability

- A proposition A is *satisfiable* if there exists an assignment v such that $v \models A$.
  - $v(P) = F, v(Q) = T \models (P \lor Q) \land (\neg P \lor \neg Q)$
- A proposition is *unsatisfiable* if no assignment satisfies it.
  - $(\neg P \lor Q) \land (\neg P \lor \neg Q) \land P$ is unsatisfiable.
- The problem of determining whether a given proposition is satisfiable is called the *satisfiability problem*.

Tautology and Validity

- A proposition A is a *tautology* if every assignment satisfies A, written as $\models A$.
  - $\models A \vee \neg A$
  - $\models (A \wedge B) \rightarrow (A \vee B)$
- The problem of determining whether a given proposition is a tautology is called the *tautology problem*.
- A proposition is also said to be *valid* if it is a tautology.
- So, the problem of determining whether a given proposition is valid (a tautology) is also called the *validity problem*.

Note: the notion of a tautology is restricted to propositional logic. In first-order logic, we also speak of valid formulae.

Validity vs. Satisfiability

Theorem

A proposition A is valid (a tautology) if and only if $\neg A$ is unsatisfiable.

So, there are two ways of proving that a proposition A is a tautology:

- A is satisfied by every truth assignment (or *A cannot be falsified* by any truth assignment).
- $\neg A$ is unsatisfiable.

Relating the Logical Connectives

IM NTU

Lemma

$$\models (A \leftrightarrow B) \leftrightarrow ((A \rightarrow B) \wedge (B \rightarrow A))$$

$$\models (A \rightarrow B) \leftrightarrow (\neg A \vee B)$$

$$\models (A \vee B) \leftrightarrow \neg(\neg A \wedge \neg B)$$

$$\models \bot \leftrightarrow (A \wedge \neg A)$$

Note: these equivalences imply that some connectives could be dispensed with. We normally want a smaller set of connectives when analyzing properties of the logic and a larger set when actually using the logic.

Normal Forms

- A *literal* is an atomic proposition or its negation.
- A propositional formula is in Conjunctive Normal Form (CNF) if it is a conjunction of disjunctions of literals.
  - $(P \lor Q \lor \neg R) \land (\neg P \lor \neg Q) \land P$
  - $(P \lor Q \lor \neg R) \land (\neg P \lor \neg Q \lor R) \land (P \lor \neg Q \lor \neg R)$
- A propositional formula is in Disjunctive Normal Form (DNF) if it is a disjunction of conjunctions of literals.
  - $(P \land Q \land \neg R) \lor (\neg P \land \neg Q) \lor P$
  - $(\neg P \land \neg Q \land R) \lor (P \land Q \land \neg R) \lor (\neg P \land Q \land R)$
- A propositional formula is in Negation Normal Form (NNF) if negations occur only in literals.
  - CNF or DNF is also NNF (but not vice versa).
  - $(P \land \neg Q) \land (P \lor (Q \land \neg R))$ in NNF, but not CNF or DNF.
- Every propositional formula has an equivalent formula in each of these normal forms.

Semantic Entailment

- Consider two sets of propositions Γ and Δ.
- We say that $v \models \Gamma$ (v satisfies Γ) if $v \models B$ for every $B \in \Gamma$; analogously for Δ.
- We say that Δ is a *semantic consequence* of Γ if every assignment that satisfies Γ also satisfies Δ, written as $\Gamma \models \Delta$.
  - $A, A \to B \models A, B$
  - $A \to B, \neg B \models \neg A$
- We also say that Γ *semantically entails* Δ when $\Gamma \models \Delta$.

Sequents

- A (propositional) *sequent* is an expression of the form $\Gamma \vdash \Delta$, where $\Gamma = A_1, A_2, \cdots, A_m$ and $\Delta = B_1, B_2, \cdots, B_n$ are finite (possibly empty) sequences of (propositional) formulae.
- In a sequent $\Gamma \vdash \Delta$, Γ is called the *antecedent* (also *context*) and Δ the *consequent*.

Note: many authors prefer to write a sequent as $\Gamma \longrightarrow \Delta$ or $\Gamma \Longrightarrow \Delta$, while reserving the symbol $\vdash$ for provability (deducibility) in the proof (deduction) system under consideration.

Sequents (cont.)

- A sequent $A_1, A_2, \cdots, A_m \vdash B_1, B_2, \cdots, B_n$ is falsifiable if there exists a valuation v such that $v \models (A_1 \wedge A_2 \wedge \cdots \wedge A_m) \wedge (\neg B_1 \wedge \neg B_2 \wedge \cdots \wedge \neg B_n)$.
  - $A \vee B \vdash B$ is falsifiable, as $v(A) = T, v(B) = F \models (A \vee B) \wedge \neg B$.
- A sequent $A_1, A_2, \cdots, A_m \vdash B_1, B_2, \cdots, B_n$ is valid if, for every valuation v, $v \models A_1 \wedge A_2 \wedge \cdots \wedge A_m \to B_1 \vee B_2 \vee \cdots \vee B_n$.
  - $A \vdash A, B$ is valid.
  - $A, B \vdash A \wedge B$ is valid.
- A sequent is valid if and only if it is not falsifiable.
- In the following, we will use only sequents of this simpler form: $A_1, A_2, \cdots, A_m \vdash C$, where C is a formula.

Inference Rules

- Inference rules allow one to obtain true statements from other true statements.
- Below is an inference rule for conjunction.

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\,(\wedge I)$$

- In an inference rule, the upper sequents (above the horizontal line) are called the *premises* and the lower sequent is called the *conclusion*.

Proofs

- A deduction tree is a tree where each node is labeled with a sequent such that, for every internal (non-leaf) node,
  - the label of the node corresponds to the conclusion and
  - the labels of its children correspond to the premises

  of an instance of an inference rule.
- A proof tree is a deduction tree, each of whose leaves is labeled with an axiom.
- The root of a deduction or proof tree is called the conclusion.
- A sequent is provable if there exists a proof tree of which it is the conclusion.

Detour: Another Style of Proofs

- Proofs may also be carried out in a calculational style (like in algebra); for example,

$$
\begin{array}{ll}
 & (A \vee B) \to C \\
\equiv & \{\ A \to B \ \equiv\ \neg A \vee B\ \} \\
 & \neg(A \vee B) \vee C \\
\equiv & \{\ \text{de Morgan's law}\ \} \\
 & (\neg A \wedge \neg B) \vee C \\
\equiv & \{\ \text{distributive law}\ \} \\
 & (\neg A \vee C) \wedge (\neg B \vee C) \\
\equiv & \{\ A \to B \ \equiv\ \neg A \vee B\ \} \\
 & (A \to C) \wedge (B \to C) \\
\Rightarrow & \{\ A \wedge B \ \Rightarrow\ A\ \} \\
 & (A \to C)
\end{array}
$$

- Here, $\Rightarrow$ corresponds to semantical entailment and $\equiv$ to mutual semantical entailment. Both are transitive.

Detour: Some Laws for Calculational Proofs

- Equivalence is commutative and associative
  - $A \leftrightarrow B \quad \equiv \quad B \leftrightarrow A$
  - $A \leftrightarrow (B \leftrightarrow C) \quad \equiv \quad (A \leftrightarrow B) \leftrightarrow C$
- $\bot \vee A \quad \equiv \quad A \vee \bot \quad \equiv \quad A$
- $\neg A \wedge A \quad \equiv \quad \bot$
- $A \rightarrow B \quad \equiv \quad \neg A \vee B$
- $A \rightarrow \bot \quad \equiv \quad \neg A$
- $(A \vee B) \rightarrow C \quad \equiv \quad (A \rightarrow C) \wedge (B \rightarrow C)$
- $A \rightarrow (B \rightarrow C) \quad \equiv \quad (A \wedge B) \rightarrow C$
- $A \rightarrow B \quad \equiv \quad A \leftrightarrow (A \wedge B)$
- $A \wedge B \quad \Rightarrow \quad A$

Natural Deduction in the Sequent Form

$$\frac{}{\Gamma, A \vdash A}\ (Ax)$$

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\ (\wedge I)$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\ (\wedge E_1)$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\ (\wedge E_2)$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ (\vee I_1)$$

$$\frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ (\vee I_2)$$

$$\frac{\Gamma \vdash A \vee B \qquad \Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma \vdash C}\ (\vee E)$$

Natural Deduction (cont.)

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B}\ (\rightarrow I) \qquad \frac{\Gamma \vdash A \rightarrow B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ (\rightarrow E)$$

$$\frac{\Gamma, A \vdash B \wedge \neg B}{\Gamma \vdash \neg A}\ (\neg I) \qquad \frac{\Gamma \vdash A \qquad \Gamma \vdash \neg A}{\Gamma \vdash B}\ (\neg E)$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash \neg\neg A}\ (\neg\neg I) \qquad \frac{\Gamma \vdash \neg\neg A}{\Gamma \vdash A}\ (\neg\neg E)$$

These inference rules collectively are called System *ND* (the propositional part).

A Proof in Propositional *ND*

Below is a partial proof of the validity of
$P \wedge (\neg P \vee \neg Q) \wedge (R \to Q) \to \neg R$ in *ND*,
where γ denotes $P \wedge (\neg P \vee \neg Q) \wedge (R \to Q)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\gamma, R \vdash R \to Q} \quad \dfrac{}{\gamma, R \vdash R}\,(Ax)}{\gamma, R \vdash Q}\,(\to E) \quad \dfrac{\dfrac{\vdots}{\gamma, R, Q \vdash P \wedge \neg P}}{\gamma, R \vdash \neg Q}\,(\neg I)}{\dfrac{\gamma, R \vdash Q \wedge \neg Q}{P \wedge (\neg P \vee \neg Q) \wedge (R \to Q) \vdash \neg R}\,(\neg I)}\,(\wedge I)}{\vdash P \wedge (\neg P \vee \neg Q) \wedge (R \to Q) \to \neg R}\,(\to I)$$

Soundness and Completeness

Theorem

System ND is sound, i.e., if a sequent $\Gamma \vdash C$ *is provable in ND, then* $\Gamma \vdash C$ *is valid.*

Theorem

System ND is complete, i.e., if a sequent $\Gamma \vdash C$ *is valid, then* $\Gamma \vdash C$ *is provable in ND.*

Compactness

A set Γ of propositions is satisfiable if some valuation satisfies every proposition in Γ. For example, $\{A \vee B, \neg B\}$ is satisfiable.

Theorem

For any (possibly infinite) set Γ of propositions, if every finite non-empty subset of Γ is satisfiable then Γ is satisfiable.

Proof hint: by contradiction and the completeness of *ND*.

Consistency

- A set Γ of propositions is *consistent* if there exists some proposition B such that the sequent $\Gamma \vdash B$ is not provable.
- Otherwise, Γ is *inconsistent*; e.g., $\{A, \neg(A \vee B)\}$ is inconsistent.

Lemma

For System ND, a set Γ of propositions is inconsistent if and only if there is some proposition A such that both $\Gamma \vdash A$ and $\Gamma \vdash \neg A$ are provable.

Theorem

For System ND, a set Γ of propositions is satisfiable if and only if Γ is consistent.

References

- J.H. Gallier. *Logic for Computer Science: Foundations of Automatic Theorem Proving*, Harper & Row Publishers, 1985.
- J. Goubault-Larrecq and I. Mackie. *Proof Theory and Automated Deduction*, Kluwer Academic Publishers, 1997.
- M. Huth and M. Ryan. *Logic in Computer Science: Modelling and Reasoning about Systems*, Cambridge University Press, 2004.