Discrete Mathematics[1]
http://lcs.ios.ac.cn/˜znj/DM2017

Naijun Zhan

February 22, 2017

[1] Special thanks to Profs Hanpin Wang (PKU) and Lijun Zhang (ISCAS) for their courtesy of the slides on this course.

Contents

The Foundations: Logic and Proofs Logic in Computer Science

During the past fifty years there has been extensive, continuous, and growing interaction between logic and computer science. In many respects, logic provides computer science with both a unifying foundational framework and a tool for modeling computational systems. In fact, logic has been called the calculus of computer science. The argument is that logic plays a fundamental role in computer science, similar to that played by calculus in the physical sciences and traditional engineering disciplines. Indeed, logic plays an important role in areas of computer science as disparate as machine architecture, computer-aided design, programming languages, databases, artificial intelligence, algorithms, and computability and complexity. **Moshe Vardi**

- The origins of logic can be dated back to Aristotle's time.
- The birth of **mathematical logic:**
  - Leibnitz's idea
  - Russell paradox
  - Hilbert's plan
  - Three schools of modern logic:
    - logicism (Frege, Russell, Whitehead)
    - formalism (Hilbert)
    - intuitionism (Brouwer)
- One of the central problem for logicians is that: "why is this proof correct/incorrect?"
- Boolean algebra owes to George Boole.
- Now, we are interested in: "is the program correct?"

Outline

Proposition

A *proposition* is a declarative sentence that is either true or false, but not both.

Propositional Logic

Fix a countable proposition set AP. Syntax of propositional formulas in BNF (Backus-Naur form) is given by:

$$\varphi ::= p \in AP \mid \neg\varphi \mid \varphi \wedge \varphi$$

Accordingly,

- Atomic proposition $p \in AP$ is a formula.
- Compound formulas: $\neg\varphi$ (negation) and $\varphi \wedge \psi$ (conjunction), provided that φ and ψ are formulas.

For $p \in AP$, negation and conjunction, we can construct the truth tables. We define the following derived operators:

- Disjunction: $\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi)$
- Implication: $\varphi \rightarrow \psi := \neg\varphi \vee \psi$
- Bi-implication: $\varphi \leftrightarrow \psi := (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$
- Exclusive Or: $\varphi \oplus \psi := (\varphi \vee \psi) \wedge (\neg(\varphi \wedge \psi))$

Precedence of Logical Operators

Operators $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ have precedence $1, 2, 3, 4, 5$, respectively.

Logic and Bit Operators

- A *bit* is a symbol with possible values 0 and 1. A *Boolean variable* is a variable with value true or false.
- Computer *bit operations* correspond to logic connectives: OR, AND, XOR in various programming languages correspond to $\vee, \wedge, \oplus$, respectively.
- A *bit string* is a sequence of zero or more bits. The *length* of this string is the number of bits in the string.
- Bitwise OR, bitwise AND and bitwise XOR of two strings of the same length are the strings that have as their bits the OR, AND and XOR of the corresponding bits in the two strings, respectively.

Outline

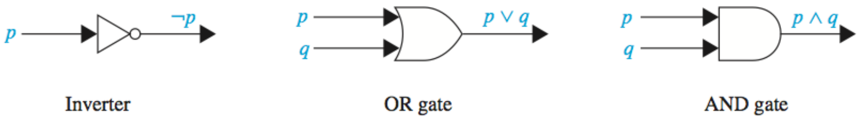
Applications

- System Specifications: The automated reply cannot be sent when the file system is full.
- Boolean Searches: (one | two) - (three)
- Logic Puzzles. Knights always tell the truth, and the opposite knaves always lie. A says: "B is a knight". B says "The two of us are opposite types"

- Propositional logic can be applied to the design of computer hardware.

Claude Shannon

- A logic circuit receives input signals $p_1, p_2, \ldots, p_n$ and produces an output s. Complicated digital circuits are constructed from three basic circuits, called *gates*.



- Build a digital circuit producing $(p \vee \neg r) \wedge (\neg p \vee (q \vee \neg r))$.

Outline

A formula φ is called a

- *tautology* if it is always true, no matter what the truth values of the propositional variables are;
- *contradiction* if it is always false;
- *contingency* if it is neither a tautology nor a contradiction.

Moreover, φ is *satisfiable* if it is either a tautology or a contingency, *unsatisfiable* if it is a contradiction.
Formulas φ and ψ are called *logically equivalent* if $\varphi \leftrightarrow \psi$ is a tautology. This is denoted by $\varphi \equiv \psi$.

Logical Equivalence

Show the following logical equivalences:

1. Identity laws:
   $\varphi \wedge \mathbf{T} \equiv \varphi$, $\varphi \vee \mathbf{F} \equiv \varphi$
2. Dominations laws
   $\varphi \vee \mathbf{T} \equiv \mathbf{T}$, $\varphi \wedge \mathbf{F} \equiv \mathbf{F}$
3. Idempotent laws
   $\varphi \vee \varphi \equiv \varphi, \varphi \wedge \varphi \equiv \varphi$
4. Double negation law
   $\neg(\neg\varphi) \equiv \varphi$
5. Commutative laws
   $\varphi \vee \psi \equiv \psi \vee \varphi, \varphi \wedge \psi \equiv \psi \wedge \varphi$
6. Associative laws
   $(\varphi_1 \vee \varphi_2) \vee \varphi_3 \equiv \varphi_1 \vee (\varphi_2 \vee \varphi_3), (\varphi_1 \wedge \varphi_2) \wedge \varphi_3 \equiv \varphi_1 \wedge (\varphi_2 \wedge \varphi_3)$
7. Distributive laws
   $\varphi_1 \vee (\varphi_2 \wedge \varphi_3) \equiv (\varphi_1 \vee \varphi_2) \wedge (\varphi_1 \vee \varphi_3),$
   $\varphi_1 \wedge (\varphi_2 \vee \varphi_3) \equiv (\varphi_1 \wedge \varphi_2) \vee (\varphi_1 \wedge \varphi_3)$
8. De Morgan's laws
   $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi, \neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$
9. Absorption laws
   $\varphi \vee (\varphi \wedge \psi) \equiv \varphi, \varphi \wedge (\varphi \vee \psi) \equiv \varphi$
10. Negation laws
    $\varphi \vee \neg\varphi \equiv \mathbf{T}, \varphi \wedge \neg\varphi \equiv \mathbf{F}$

Logical equivalences involving conditional statements:

1. $\varphi \rightarrow \psi \equiv \neg\varphi \lor \psi$
2. $\varphi \rightarrow \psi \equiv \neg\psi \rightarrow \neg\varphi$
3. $\varphi \lor \psi \equiv \neg\varphi \rightarrow \psi$
4. $\varphi \land \psi \equiv \neg(\varphi \rightarrow \neg\psi)$
5. $\varphi \land \neg\psi \equiv \neg(\varphi \rightarrow \psi)$
6. $(\varphi_1 \rightarrow \varphi_2) \land (\varphi_1 \rightarrow \varphi_3) \equiv \varphi_1 \rightarrow (\varphi_2 \land \varphi_3)$
7. $(\varphi_1 \rightarrow \varphi_3) \land (\varphi_2 \rightarrow \varphi_3) \equiv (\varphi_1 \lor \varphi_2) \rightarrow \varphi_3$
8. $(\varphi_1 \rightarrow \varphi_2) \lor (\varphi_1 \rightarrow \varphi_3) \equiv \varphi_1 \rightarrow (\varphi_2 \lor \varphi_3)$
9. $(\varphi_1 \rightarrow \varphi_3) \lor (\varphi_2 \rightarrow \varphi_3) \equiv (\varphi_1 \land \varphi_2) \rightarrow \varphi_3$
10. $\varphi \leftrightarrow \psi \equiv \neg\varphi \leftrightarrow \neg\psi$
11. $\varphi \leftrightarrow \psi \equiv (\varphi \land \psi) \lor (\neg\varphi \land \neg\psi)$
12. $\neg(\varphi \leftrightarrow \psi) \equiv \varphi \leftrightarrow \neg\psi$

Outline

PRINCIPLE OF MATHEMATICAL INDUCTION

To prove that $P(n)$ is true for all positive integers n, where $P(n)$ is a propositional function, we complete two steps:

- BASIS STEP: We verify that $P(1)$ is true.
- INDUCTIVE STEP: We show that the conditional statement $P(k) \rightarrow P(k+1)$ is true for all positive integers k.

Expressed as a rule of inference for *first-order logic*, this proof technique can be stated as:

$$\Phi := (P(1) \land \forall k.(P(k) \rightarrow P(k+1))) \rightarrow \forall n.P(n)$$

Exercise

Prove $1 + 2 + 2^2 + \ldots + 2^n = 2^{n+1} - 1.$

STRONG INDUCTION (Second principle of mathematical induction)

To prove that $P(n)$ is true for all positive integers n, where $P(n)$ is a propositional function, we complete two steps:

- BASIS STEP: We verify that the proposition $P(1)$ is true.
- INDUCTIVE STEP: We show that the conditional statement $(P(1) \wedge P(2) \wedge \ldots \wedge P(k)) \rightarrow P(k+1)$ is true for all positive integers k.

Expressed as a rule of inference for *first-order logic*, this proof technique can be stated as:

$$\Psi := (P(1) \wedge \forall k.(\wedge_{i=1}^{k} P(i) \rightarrow P(k+1))) \rightarrow \forall n.P(n)$$

Exercise

Prove that if n is a natural number greater than 1*, then n can be written as the product of primes.*

Strings

The set Σ^* of *strings* over the alphabet Σ is defined recursively by

- BASIS STEP: $\lambda \in \Sigma^*$ (where λ is the empty string containing no symbols).
- RECURSIVE STEP: If $w \in \Sigma^*$ and $x \in \Sigma$, then $wx \in \Sigma^*$.

We define the set of *well-formed formulas in propositional logic*, denoted by L, from alphabet $\Sigma := AP \cup \{\neg, \rightarrow, (,)\}$.

- BASIS STEP: each $p \in AP$ is a well-formed formula.
- RECURSIVE STEP: If φ and ψ are well-formed formulas, i.e., $\varphi, \psi \in L$, then $(\neg\varphi)$, $(\varphi \rightarrow \psi)$ are well-formed formulas.

Thus, the set of well-formed formulas is a subset of $L \subseteq \Sigma^*$.

STRUCTURAL INDUCTION

A proof by structural induction consists of two parts.

- BASIS STEP: Show that the result holds for all elements specified in the basis step.
- RECURSIVE STEP: Show that if the statement is true for each of the elements used to construct new elements in the recursive step of the definition, the result holds for these new elements.

Remark: The validity of structural induction follows from the principle of mathematical induction for the nonnegative integers.

Exercise

Show that every well-formed formula for compound propositions contains an equal number of left and right parentheses.

Outline

Negation Normal Form

- **Literal:** An atomic proposition p or its negation $\neg p$;
- **Negation Normal Form (NNF):** A formula built up with "$\wedge$", "$\vee$", and literals.
- Using repeated DeMorgan and Double Negation, we can transform any formula into a formula with Negation Normal Form.
- **Example:**

$$
\begin{array}{lll}
\neg((A \vee B) \wedge \neg C) & \leftrightarrow & \text{(DeMorgan)} \\
\neg(A \vee B) \vee \neg\neg C & \leftrightarrow & \text{(Double Neg, DeMorgan)} \\
(\neg A \wedge \neg B) \vee C & &
\end{array}
$$

Disjunction Normal Form

- **Disjunction Normal Form (DNF):** A generalized disjunction of generalized conjunctions of literals.
- Using repeated distribution of $\wedge$ over $\vee$, any NNF formula can be rewritten in DNF (exercise).
- **Example:**

$$
\begin{array}{lcl}
(A \vee B) \wedge (C \vee D) & \leftrightarrow & \text{(Distribution)} \\
[(A \vee B) \wedge C] \vee [(A \vee B) \wedge D] & \leftrightarrow & \text{(Distribution)} \\
(A \wedge C) \vee (B \wedge C) \vee (A \wedge D) \vee (B \wedge D) & &
\end{array}
$$

Conjunction Normal Form

- **Conjunction Normal Form (CNF):** A generalized conjunction of generalized disjunctions of literals.
- Using repeated distribution of $\vee$ over $\wedge$, any NNF formula can be rewritten in CNF (exercise).
- **Example:**

$$
\begin{array}{lcl}
(A \wedge B) \vee (C \wedge D) & \leftrightarrow & \text{(Distribution)} \\
[(A \wedge B) \vee C] \wedge [(A \wedge B) \vee D] & \leftrightarrow & \text{(Distribution)} \\
(A \vee C) \wedge (B \vee C) \wedge (A \vee D) \wedge (B \vee D) & &
\end{array}
$$

Unary Connectives

- What other unary connectives are there besides '¬'?
- Thinking about this in terms of truth tables, we see that there are 4 different unary connectives:

P	*P
T	T
F	T

P	*P
T	T
F	F

P	*P
T	F
F	T

P	*P
T	F
F	F

Binary Connectives

- The truth table below shows that there are $2^4 = 16$ binary connectives:

P	Q	P*Q
T	T	T/F
T	F	T/F
F	T	T/F
F	F	T/F

In general:
n sentences $\Rightarrow$

2^n truth value combinations
(i.e. 2^n rows in truth table) $\Rightarrow$

2^{2^n} different n-ary connectives!

Expressive completeness

- What are the truth tables for $(p \wedge q) \vee r$ and $p \wedge (q \vee r)$?
- Truth table for n-ary Boolean function.
- A set of logical connectives is called **functionally complete** if any n-ary Boolean function is definable with it, e.g. $\{\neg, \wedge\}$, $\{\neg, \vee\}$.
- How about $\{\neg, \rightarrow\}$ and $\{\neg, \leftrightarrow\}$?

Exercise. Exercise 49 of page 16, Exercise 15 of page 23, Exercise 39 of page 35, Exercise 45 of page 36.

Outline

This section considers a complete axiomatization system such that, a formula is a tautology if and only if it can be derived by means of the axioms and the deduction rules of the system.

Syntax and Semantics of Propositional Logic

Fix a countable proposition set AP, then formulas of propositional logic are defined by:

Definition (Syntax)

Syntax of propositional formulas in BNF (Backus-Naur form) is given by:

$$\varphi ::= p \in AP \mid \neg\varphi \mid \varphi \rightarrow \varphi$$

It generates recursively the set of well-formed formulas, denoted by L:

- Atomic formula: $p \in AP$ implies $p \in L$.
- Compound formulas: $(\neg\varphi)$ and $(\varphi \rightarrow \psi)$, provided that $\varphi, \psi \in L$.

We omit parentheses if it is clear from the context.

Semantics

Semantics of a formula φ is given w.r.t. an assignment $\sigma \in 2^{AP}$, which is a subset of AP. Intuitively, it assigns true (or, $\mathbf{T}$) to propositions belonging to it, and assigns false (or, $\mathbf{F}$) to others. Thus, it can also be viewed as a function from AP to $\{\mathbf{T}, \mathbf{F}\}$.

Definition (Semantics)

Inductively, we may define the relation $\Vdash \subseteq 2^{AP} \times L$ as follows:

- $\sigma \Vdash p$ iff $p \in \sigma$.
- $\sigma \Vdash \neg\varphi$ iff not $\sigma \Vdash \varphi$ (denoted by $\sigma \nVdash \varphi$).
- $\sigma \Vdash \varphi \rightarrow \psi$ iff either $\sigma \nVdash \varphi$ or $\sigma \Vdash \psi$.

where $(\sigma, \varphi) \in \Vdash$ is denoted as $\sigma \Vdash \varphi$.

The formula φ is called a *tautology* if $\sigma \Vdash \varphi$ for all assignment, it is *satisfiable* if $\sigma \Vdash \varphi$ for some assignment.

The Axiom System: the Hilbert's System

Axioms

1. $\varphi \to (\psi \to \varphi)$.
2. $(\varphi \to (\psi \to \eta)) \to ((\varphi \to \psi) \to (\varphi \to \eta))$.
3. $(\neg\varphi \to \neg\psi) \to (\psi \to \varphi)$.

MP Rule

1. $\dfrac{\varphi \to \psi \quad \varphi}{\psi}$

Given a formula set Γ, a deductive sequence of φ from Γ is a sequence

$$\varphi_0, \varphi_1, \ldots, \varphi_n = \varphi$$

where each φ_i should be one of the following cases:

1. $\varphi_i \in \Gamma$.
2. φ_i is an instance of some axiom.
3. There exists some $j, k < i$, such that $\varphi_k = \varphi_j \to \varphi_i$.

And, we denote by $\Gamma \vdash \varphi$ if there exists such deductive sequence. We write $\Gamma, \psi \vdash \varphi$ for $\Gamma \cup \{\psi\} \vdash \varphi$.

The Axiom System: Soundness

For a formula set Γ and an assignment σ, the *satisfaction relation* $\Vdash$ is defined by: $\sigma \Vdash \Gamma$ iff $\sigma \Vdash \varphi$ for every $\varphi \in \Gamma$.
Observe $\sigma \Vdash \emptyset$ always holds. We say φ is a logical consequent of Γ, denoted as $\Gamma \models \varphi$, if $\sigma \Vdash \Gamma$ implies $\sigma \Vdash \varphi$ for each assignment σ.
Thus, φ is a tautology if φ is the logical consequent of $\emptyset$, denoted as $\models \varphi$.

Theorem (Soundness)

Regard **Hilbert's axiom system**, *we have that $\Gamma \vdash \varphi$ implies $\Gamma \models \varphi$.*

Proof.

By induction of the length of deductive sequence of $\Gamma \vdash \varphi$. □

Corollary

If $\vdash \varphi$, then $\models \varphi$.

The Axiom System: Completeness

With Hilbert's axiom system, we have the following elementary properties:

- **(Fin)** If $\Gamma \vdash \varphi$, then there exists some finite subset Γ' of Γ, such that $\Gamma' \vdash \varphi$.
- **($\in$)** If $\varphi \in \Gamma$, then $\Gamma \vdash \varphi$.
- **($\in_+$)** If $\Gamma \vdash \varphi$ and $\Gamma \subseteq \Gamma'$ then $\Gamma' \vdash \varphi$.
- **($\overline{\text{MP}}$)** If $\Gamma_1 \vdash \varphi$ and $\Gamma_2 \vdash \varphi \to \psi$, and $\Gamma_1, \Gamma_2 \subseteq \Gamma$, then $\Gamma \vdash \psi$.

The Axiom System: Examples of Theorems

Example

(Ide): $\vdash \varphi \to \varphi$

Solution

1. $\varphi \to (\varphi \to \varphi)$
2. $\varphi \to ((\varphi \to \varphi) \to \varphi)$
3. $(\varphi \to ((\varphi \to \varphi) \to \varphi)) \to ((\varphi \to (\varphi \to \varphi)) \to (\varphi \to \varphi))$
4. $(\varphi \to (\varphi \to \varphi)) \to (\varphi \to \varphi)$
5. $\varphi \to \varphi$

Example

$(\to_{-})$: If $\Gamma \vdash \varphi \to \psi$ then $\Gamma, \varphi \vdash \psi$.

Solution

A simple application of $\overline{MP}$ and $(\in)$.

Example

$(\to_+)$ (**Deduction Theorem**) : If $\Gamma, \varphi \vdash \psi$ then $\Gamma \vdash \varphi \to \psi$.

Solution

By induction of the deductive sequence of $\Gamma, \varphi \vdash \psi$.

Example

(τ): If $\Gamma \vdash \varphi \to \psi$ and $\Gamma \vdash \psi \to \eta$, then $\Gamma \vdash \varphi \to \eta$.

Solution

By $(\to_-)$, $(\to_+)$ *and* $(\in_+)$.

Example

(Abs): $\vdash \neg\varphi \to (\varphi \to \psi)$.

Solution

1. $\vdash \neg\varphi \to (\neg\psi \to \neg\varphi)$
2. $\vdash (\neg\psi \to \neg\varphi) \to (\varphi \to \psi)$
3. $\vdash \neg\varphi \to (\varphi \to \psi)$

Example

(Abs'): $\vdash \varphi \to (\neg\varphi \to \psi)$

Example

$(\neg_w)$: $\neg\varphi \to \varphi \vdash \varphi$

Solution

1. $\neg\varphi \to \varphi \vdash \neg\varphi \to \varphi$
2. $\vdash \neg\varphi \to (\varphi \to \neg(\neg\varphi \to \varphi))$
3. $\vdash (\neg\varphi \to (\varphi \to \neg(\neg\varphi \to \varphi))) \to ((\neg\varphi \to \varphi) \to (\neg\varphi \to \neg(\neg\varphi \to \varphi)))$
4. $\neg\varphi \to \varphi \vdash \neg\varphi \to \neg(\neg\varphi \to \varphi)$
5. $\vdash (\neg\varphi \to \neg(\neg\varphi \to \varphi)) \to ((\neg\varphi \to \varphi) \to \varphi)$
6. $\neg\varphi \to \varphi \vdash (\neg\varphi \to \varphi) \to \varphi$
7. $\neg\varphi \to \varphi \vdash \varphi$

Example

$(\neg\neg\_)$: $\neg\neg\varphi \vdash \varphi$

Solution

1. $\vdash \neg\neg\varphi \to (\neg\varphi \to \varphi)$
2. $\vdash (\neg\varphi \to \varphi) \to \varphi$
3. $\vdash \neg\neg\varphi \to \varphi$
4. $\neg\neg\varphi \vdash \varphi$

Example

$(\neg_s)$: $\varphi \to \neg\varphi \vdash \neg\varphi$

Solution

1. $\neg\neg\varphi \vdash \varphi$
2. $\varphi \to \neg\varphi \vdash \varphi \to \neg\varphi$
3. $\varphi \to \neg\varphi, \neg\neg\varphi \vdash \varphi$
4. $\varphi \to \neg\varphi \vdash \neg\neg\varphi \to \neg\varphi$
5. $\vdash (\neg\neg\varphi \to \neg\varphi) \to \neg\varphi$
6. $\varphi \to \neg\varphi \vdash \neg\varphi$

Example

$(\neg\neg_+)$: $\varphi \vdash \neg\neg\varphi$

Solution

1. $\vdash \varphi \to (\neg\varphi \to \neg\neg\varphi)$
2. $\vdash (\neg\varphi \to \neg\neg\varphi) \to \neg\neg\varphi$
3. $\vdash \varphi \to \neg\neg\varphi$
4. $\varphi \vdash \neg\neg\varphi$

Example

(R0) $\varphi \to \psi \vdash \neg\psi \to \neg\varphi$

(R1) $\varphi \to \neg\psi \vdash \psi \to \neg\varphi$

(R2) $\neg\varphi \to \psi \vdash \neg\psi \to \varphi$

(R3) $\neg\varphi \to \neg\psi \vdash \psi \to \varphi$

Solution

1. $\varphi \to \psi, \neg\neg\varphi \vdash \varphi$
2. $\varphi \to \psi, \neg\neg\varphi \vdash \psi$
3. $\vdash \psi \to \neg\neg\psi$
4. $\vdash (\neg\neg\varphi \to \neg\neg\psi) \to (\neg\psi \to \neg\varphi)$
5. $\varphi \to \psi \vdash \neg\psi \to \neg\varphi$

Consistency

Consistency

We say a formula set Γ is consistent, iff there is some φ such that $\Gamma \nvdash \varphi$. Moreover, we say φ is consistent w.r.t. Γ iff $\Gamma \cup \{\varphi\}$ is consistent.

Note that we have the theorem $\neg\varphi, \varphi \vdash \psi$ and hence, Γ is consistent iff for each φ, either $\Gamma \nvdash \varphi$ or $\Gamma \nvdash \neg\varphi$.

Further, φ is consistent w.r.t. Γ iff $\Gamma \nvdash \neg\varphi$. Suppose that $\Gamma \nvdash \neg\varphi$ and $\Gamma \cup \{\varphi\}$ is inconsistent, then we have $\Gamma, \varphi \vdash \neg\varphi$ hence $\Gamma \vdash \varphi \to \neg\varphi$. Recall that we have $\varphi \to \neg\varphi \vdash \neg\varphi$, and this implies $\Gamma \vdash \neg\varphi$, contradiction!

Lemma

If the formula set Γ is inconsistent, then it has some finite inconsistent subset Δ.

Theorem

Γ *is consistent iff* Γ *is satisfiable.*

Proof sketch.

The "if" direction is easy: suppose that $\sigma \Vdash \Gamma$ but $\Gamma \vdash \varphi$ and $\Gamma \vdash \neg\varphi$, then $\sigma \Vdash \varphi$ and $\sigma \Vdash \neg\varphi$, contradiction.

For the "only if" direction, let us enumerate all propositional formulas as following (note the cardinality of all such formulas is $\aleph_0$):

$$\varphi_0, \varphi_1, \ldots, \varphi_n, \ldots$$

Let $\Gamma_0 = \Gamma$ and

$$\Gamma_{i+1} = \begin{cases} \Gamma_i \cup \{\varphi_i\} & \text{if } \Gamma_i \nvdash \neg\varphi_i \\ \Gamma_i \cup \{\neg\varphi_i\} & \text{otherwise} \end{cases}$$

and finally let $\Gamma^* = \lim_{i\to\infty} \Gamma_i$.

The formula set Γ^* has the following properties:

1. Each Γ_i is consistent, and Γ^* is also consistent.
2. Γ^* is a maximal set, i.e., for each formula φ, either $\varphi \in \Gamma^*$ or $\neg\varphi \in \Gamma^*$.
3. For each formula φ, we have $\Gamma^* \models \varphi$ iff $\varphi \in \Gamma^*$.

Then we have $\sigma \Vdash \Gamma^*$, where $\sigma = \Gamma^* \cap AP$. □

Theorem (Completeness)

If $\Gamma \models \varphi$, *then* $\Gamma \vdash \varphi$.

Proof.

Assume by contradiction that $\Gamma \nvdash \varphi$, then there is an assignment σ such that $\sigma \Vdash \Gamma \cup \{\neg\varphi\}$. However, this implies that $\sigma \Vdash \Gamma$ and $\sigma \nVdash \varphi$, which violates the assumption $\Gamma \models \varphi$. □

Corollary

$\models \varphi$ *implies that* $\vdash \varphi$.

Theorem

Given a formula set Γ, we have

1. Γ *is consistent iff each of its finite subsets is consistent;*
2. Γ *is satisfiable iff each of its finite subsets is satisfiable.*

Proof.

1. The first property has been proven (see the previous lemma).
2. With the aforementioned theorem: for propositional logic, a set is satisfiable iff it is consistent.

□

Rules of Inference for Propositional Logic (cf. page 72):

Rule of Inference	*Tautology*	*Name*
p $p \to q$ $\therefore q$	$(p \land (p \to q)) \to q$	Modus ponens
$\neg q$ $p \to q$ $\therefore \neg p$	$(\neg q \land (p \to q)) \to \neg p$	Modus tollens
$p \to q$ $q \to r$ $\therefore p \to r$	$((p \to q) \land (q \to r)) \to (p \to r)$	Hypothetical syllogism
$p \lor q$ $\neg p$ $\therefore q$	$((p \lor q) \land \neg p) \to q$	Disjunctive syllogism
p $\therefore p \lor q$	$p \to (p \lor q)$	Addition
$p \land q$ $\therefore p$	$(p \land q) \to p$	Simplification
p q $\therefore p \land q$	$((p) \land (q)) \to (p \land q)$	Conjunction
$p \lor q$ $\neg p \lor r$ $\therefore q \lor r$	$((p \lor q) \land (\neg p \lor r)) \to (q \lor r)$	Resolution

Exercise

Exercise 1

Show, by applying the rules of the deduction system presented in Section 6, the following statements:

1. $\vdash (\varphi \to \psi) \to ((\neg\varphi \to \neg\psi) \to (\psi \to \varphi))$
2. $\vdash ((\varphi \to (\psi \to \eta)) \to (\varphi \to \psi)) \to ((\varphi \to (\psi \to \eta)) \to (\varphi \to \eta))$
3. $\vdash (\varphi \to (\varphi \to \psi)) \to (\varphi \to \psi)$
4. $\vdash \varphi \to (\psi \to (\varphi \to \psi))$
5. $\{\varphi \to \psi, \neg(\psi \to \eta) \to \neg\varphi\} \vdash \varphi \to \eta$
6. $\varphi \to (\psi \to \eta) \vdash \psi \to (\varphi \to \eta)$
7. $\vdash ((\varphi \to \psi) \to \varphi) \to \varphi$
8. $\vdash \neg(\varphi \to \psi) \to (\psi \to \varphi)$

Exercise

Exercise 2

Find a deduction showing the correctness of some of the following equivalences, that is, if $\varphi \equiv \psi$, then provide a deduction for $\vdash \varphi \to \psi$ and for $\vdash \psi \to \varphi$.

1. $\varphi \vee (\varphi \wedge \psi) \equiv \varphi$,
2. $(\varphi_1 \to \varphi_2) \vee (\varphi_1 \to \varphi_3) \equiv \varphi_1 \to (\varphi_2 \vee \varphi_3)$.

Exercise 3 [* not required]

Fill the missing parts of the proofs of the soundness and completeness theorems in Section 6.