Discrete Math

Contents

Part I Discrete Math: Logic

Chapter I Propositional Logic

§ 1.1 Connectives and Truth Assingments

Define 1.1.1 (Truth table of Connectives) (Omitted)

Define 1.1.2 (Truth Assingments) Suppose Σ is the set of propositional variables. A mapping from Σ to $\{\mathbf{T}, \mathbf{F}\}$ called a truth assignment.

Define 1.1.3 Suppose Σ is the set of propositional variables and $\mathcal{J} : \Sigma \to \{\mathbf{T}, \mathbf{F}\}$ is a truth assignment. The truth value of the compond proposition on $\mathcal{J}$ …
(Omitted)

Define 1.1.4 (Tautology, contradiction) (Omitted)

Define 1.1.5 (Contingency, Satisfiable) A contingency is a compound proposition that is neither a tautology nor a contradiction.
A compound proposition is satisfiable if it is true under some truth assignment.

§ 1.2 Consequence and Equivalent

1 The definition of consequence and logically equivalent

Define 1.2.1 (Consequence) Suppose Φ is a set of propositions and ψ is one single proposition. We say that ψ is a consequence of Φ, written as $\Phi \models \psi$. if Φ 's being all true implies that ψ is also true.
In other words, $\Phi \models \psi$ if for any truth assignment $\mathcal{J}$, $[\![\phi]\!]_{\mathcal{J}} = \mathbf{T}$ for any $\phi \in \Phi$

implies $[\![\psi]\!]_{\mathcal{J}} = \mathbf{T}$.

Define 1.2.2 (Logically Equivalent) ϕ is a logically equivalent to ψ, written as $\phi \equiv \psi$, if ϕ 's truth value and ψ 's truth value are the same under any situation. In other words, $\phi \equiv \psi$ if $[\![\phi]\!]_{\mathcal{J}} = [\![\psi]\!]_{\mathcal{J}}$ for any truth assignment $\mathcal{J}$.

Example 1.2.1 $\Phi = \{\ \}, \psi = p \vee \neg p, \Phi \models \psi$

2 Important properties

Theorem 1.2.1

- $\phi \vee \neg\phi$ is an tautology
- $\phi \wedge \neg\phi$ is a contradiction
- $\phi, \psi \models \phi \wedge \psi$ ($\wedge$-Introduction)
- $\phi \wedge \psi \models \phi$ ($\wedge$-Elimination)
- $\phi \models \phi \vee \psi$ ($\vee$-Introduction)
- If $\Phi, \phi_1 \models \psi, \Phi, \phi_2 \models \psi$, then $\Phi, \phi_1 \vee \phi_2 \models \psi$ ($\vee$-Elimination)

Proof (Proof of the last one) Suppose $[\![\phi]\!]_{\mathcal{J}} = \mathbf{T}$, $[\![\phi_1 \vee \phi_2]\!]_{\mathcal{J}} = \mathbf{T}$.
Then at least one of the following holds: $[\![\phi_1]\!]_{\mathcal{J}} = \mathbf{T}$, $[\![\phi_2]\!]_{\mathcal{J}} = \mathbf{T}$. $\square$

Theorem 1.2.2 (Contrapositive) If $\Phi, \neg\phi \models \psi$, then $\Phi, \neg\psi \models \phi$

Theorem 1.2.3

- $\neg(\neg q) \equiv q$ (Double Negation)
- $\phi \wedge \phi \equiv \phi, \quad \phi \vee \phi \equiv \phi$ (Idempotent Laws)
- $\phi \wedge \psi \equiv \psi \wedge \psi, \quad \phi \vee \psi \equiv \psi \vee \psi$ (Commutative Laws)

- $\phi \vee (\psi \wedge \chi) \equiv (\phi \vee \psi) \wedge (\phi \vee \chi), \quad \phi \wedge (\psi \vee \chi) \equiv (\phi \wedge \psi) \vee (\phi \wedge \chi)$ (Distributive Laws)
- $\neg(q \wedge q) \equiv \neg p \vee \neg q, \quad \neg(q \vee q) \equiv \neg p \wedge \neg q$ (De Morgan's Laws)
- $\phi \wedge (\neg\phi) \equiv \mathbf{F}, \quad \phi \vee (\neg\phi) \equiv \mathbf{T}$ (Negation Laws)
- $\phi \wedge \mathbf{T} \equiv \phi, \quad \phi \vee \mathbf{F} \equiv \phi, \quad \phi \wedge \mathbf{F} \equiv \mathbf{F}, \quad \phi \vee \mathbf{T} \equiv \mathbf{T}$ (Laws of logical constants)
- $\phi \vee (\phi \wedge \psi) \equiv \phi, \quad \phi \wedge (\phi \vee \psi) \equiv \phi$ (Absorption Laws)

3 Prove Logical Equivalence

Theorem 1.2.4 (Transitivity) If $\phi \equiv \psi$ and $\psi \equiv \chi$, then $\phi \equiv \chi$.

Theorem 1.2.5 (Congruence Property)

- If $\phi \equiv \psi$, then $\neg\phi \equiv \neg\psi$
- If $\phi_1 \equiv \phi_2, \psi_1 \equiv \psi_2$, then $\phi_1 \wedge \psi_1 \equiv \phi_2 \wedge \psi_2$
- If $\phi_1 \equiv \phi_2, \psi_1 \equiv \psi_2$, then $\phi_1 \vee \psi_1 \equiv \phi_2 \vee \psi_2$

Theorem 1.2.6 (Reflexivity) $\phi \equiv \phi$

4 Relation among tautologies, contradictions, satisfiable assertions, consequence relations and logic equivalence

Theorem 1.2.7

- $\phi_1, \phi_2, \cdots \phi_n \models \psi$ iff. $\left(\bigwedge_{k=1}^{n}\right) \wedge \neg\psi$ is not satisfiable.
- $\{\ \} \models \phi$ iff. ϕ is an tautology.

- $\phi \equiv \psi$ iff. $\phi \models \psi$ and $\psi \models \phi$.

Theorem 1.2.8 If $\phi \models \psi$ and $\psi \models \chi$, then $\phi \models \chi$.

§ 1.3 Normal Forms

Define 1.3.1 (Disjunctive Normal Form, DNF)

- A **literal** is a propositional variable or its negation.
- A **conjunctive clause** is a conjunctions of literals.
- A **compound proposition** is in disjunctive normal form if it is a disjunction of conjunctive clauses.

Define 1.3.2 (Conjunctive Normal Form, CNF)
(Similar as above)

Example 1.3.1

- literals $x, y, z, p, q, r, \neg q$
- conjunctive clauses $p, p \wedge q, \neg p \wedge q$
- DNF $p, p \vee (\neg q \wedge r), \neg p \vee (q \wedge p \wedge r)$

Theorem 1.3.1 Every compound proposition is logically equivalent to some compound proposition in DNF.

Proof (Proof 1) Suppose that the compound proposition ϕ consists of the literals $p_1, p_2, \cdots, p_n$.
For all $\mathcal{J}$ as a interpretation, we only need to prove that

$$\phi \equiv \bigvee_{[\![\phi]\!]_{\mathcal{J}}=\mathbf{T}} \left(\bigwedge_{\mathcal{J}(p_i)=\mathbf{T}} p_i \wedge \bigwedge_{\mathcal{J}(p_i)=\mathbf{F}} \neg p_i \right) = \mathbf{T}$$

Consider a specific interpretation $\mathcal{J}_0$, if $[\![\phi]\!]_{\mathcal{J}} = \mathbf{T}$, then

$$\left[\!\!\left[\bigvee_{[\![\phi]\!]_{\mathcal{J}}=\mathbf{T}}\left(\bigwedge_{\mathcal{J}(p_i)=\mathbf{T}} p_i \wedge \bigwedge_{\mathcal{J}(p_i)=\mathbf{F}} \neg p_i\right)\right]\!\!\right]_{\mathcal{J}_0} = \left[\!\!\left[\left(\bigwedge_{\mathcal{J}_0(p_i)=\mathbf{T}} p_i \wedge \bigwedge_{\mathcal{J}_0(p_i)=\mathbf{F}} \neg p_i\right)\right]\!\!\right]_{\mathcal{J}_0}$$

If $\mathcal{J}_0(p_i) = \mathbf{T}$, then $[\![p_i]\!]_{\mathcal{J}_\prime} = \mathbf{T}$,
if $\mathcal{J}_0(p_i) = \mathbf{F}$, then $[\![\neg p_i]\!]_{\mathcal{J}_\prime} = \mathbf{T}$.
So

$$\left[\!\!\left[\left(\bigwedge_{\mathcal{J}_0(p_i)=\mathbf{T}} p_i \wedge \bigwedge_{\mathcal{J}_0(p_i)=\mathbf{F}} \neg p_i\right)\right]\!\!\right]_{\mathcal{J}_0} = \mathbf{T}$$

□

Proof (Proof 2) Define $DNF(\phi)$ as follow and prove that $DNF(\phi) \equiv \phi$.

Define 1.3.3

- $DNF(\phi) \triangleq DNF_2(DNF_1(\phi))$

- $DNF_1(\neg\neg\phi) = DNF_1(\phi)$.
  (The De Morgan's law)
  $DNF_1(\phi \wedge \psi) = DNF_1(\phi) \wedge DNF_1(\psi)$ ($\vee$ is the same)
  $DNF_1(l) = l$ l is a literal.

- $DNF_2(l) = l$ l is a literal,
  $DNF_2(\phi \vee \psi) = DNF_2(\phi) \vee DNF_2(\psi)$
  If $\phi = \bigvee_{i=1}^{n} \phi_i, \psi = \bigvee_{j=1}^{m} \psi_j$, then

$$DNF_2(\phi \wedge \psi) = \bigvee_{i=1}^{n}\bigvee_{j=1}^{m}(\phi_i \wedge \psi_j)$$

Then it's obvious that $\phi \equiv DNF(\phi)$ and $DNF(\phi)$ is a DNF. □

Theorem 1.3.2 Every compound proposition is logically equivalent to some compound proposition in CNF.

Proof (Similar as above)

Example 1.3.2 (*) The CDCL algorithm.
(Suspended now)

Chapter II First Order Logic, FOL

§ 2.1 The syntax of first order language

Define 2.1.1

- Predicate Logic's Language
  - Variables $x, y, z, \cdots$
  - Constants $c_1, c_2, \cdots$
  - Prelicates $P, Q, R, \cdots$
  - Functions $f, g, h, \cdots$
  - Logic patterns $\exists, \forall, \wedge, \vee, \neg$
- Terms $x, y, c_1, c_2, f(x), g(x, y), \cdots$
- propositions $P(x), Q(f(x, g(x, y))), \exists x \forall y R(x, g(y)), \cdots$

§ 2.2 The semantics of first order language

1 Structure

Define 2.2.1 (S-structure)
Given a sumbol set S, an S-structure $\mathcal{A} = (A, \alpha)$ contains

- a domain A, which is a non-empty set.
- an interpretation of every predicate symbol.
  Example 2.2.1 if P is a symbol of binary predicate, then $\alpha(P)$ is a mapping from $A \times A$ to $\{\mathbf{T}, \mathbf{F}\}$.

- an interpretation of every function symbol.
  Example 2.2.2 if f is a symbol of unary function, then $\alpha(f)$ is a mapping from A to A.

- an interpretation of every constant symbol.
  Example 2.2.3 if s is a constant symbol, $\alpha(c)$ is an element in domain A.

With a structure, we can determine the truth of an closed proposition.

2 Interpretation

Define 2.2.2 (S-interpretation)
Given a symbol set S, a S-interpretation $\mathcal{J} = (\mathcal{A}, \beta)$ is

- a S-structure $\mathcal{A} = (A, \alpha)$
- a S-assignment β: a mapping from variables to elements in the domain A

For $\mathcal{J} = (\mathcal{A}, \beta)$ and $\mathcal{A} = (A, \alpha)$, we usually use $\mathcal{J}(P)$ and $\mathcal{A}(P)$ to represent $\alpha(P)$, use $\mathcal{J}(f)$ and $\mathcal{A}(f)$ to represent $\alpha(f)$, use $\mathcal{J}(c)$ and $\mathcal{A}(c)$ to represent $\alpha(c)$, and use $\mathcal{J}(x)$ to represent $\beta(x)$.

Define 2.2.3 (Terms' denotation)
For S-interpretation $\mathcal{J}$ and a S-term t,

- $[\![x]\!]_{\mathcal{J}} = \mathcal{J}(x)$
- $[\![c]\!]_{\mathcal{J}} = \mathcal{J}(c)$
- $[\![f(t_1, t_2, \ldots, t_n)]\!]_{\mathcal{J}} = \mathcal{J}(f)\big([\![t_1]\!]_{\mathcal{J}}, [\![t_2]\!]_{\mathcal{J}}, \ldots, [\![t_n]\!]_{\mathcal{J}}\big)$

Define 2.2.4 (Propositions' truth)
For S-interpretation $\mathcal{J}$ and a S-proposition t,

- $[\![P(t_1, t_2, \ldots, t_n)]\!]_{\mathcal{J}} = \mathcal{J}(P)\big([\![t_1]\!]_{\mathcal{J}}, [\![t_2]\!]_{\mathcal{J}}, \ldots, [\![t_n]\!]_{\mathcal{J}}\big)$

- $[\![\varphi \wedge \psi]\!]_{\mathcal{J}} = [\![\wedge]\!]\big([\![\varphi]\!]_{\mathcal{J}}, [\![\psi]\!]_{\mathcal{J}}\big)$
- $[\![\neg\varphi]\!]_{\mathcal{J}} = [\![\neg]\!]\big([\![\varphi]\!]_{\mathcal{J}}\big)$
- $[\![\forall x\varphi]\!]_{\mathcal{J}} = \mathbf{T}$ if and only if for every a in $\mathcal{A}$'s domain, $[\![\varphi]\!]_{\mathcal{J}[x\mapsto a]} = \mathbf{T}$
- $[\![\exists x\varphi]\!]_{\mathcal{J}} = \mathbf{T}$ if and only if for at least one a in $\mathcal{A}$'s domain, $[\![\varphi]\!]_{\mathcal{J}[x\mapsto a]} = \mathbf{T}$

where $\mathcal{J}[x \mapsto a]$ is a S-interpretation which keeps all other interpretations in $\mathcal{J}$ and interprets x by a.

§ 2.3 Quantiers with restricted domains

1 The truth of "if-then"

Theorem 2.3.1

- $\phi \to (\psi \to \phi) \equiv \mathbf{T}$.
- $(\phi \to \psi \to \chi) \to (\phi \to \psi) \to (\phi \to \chi) \equiv \mathbf{T}$.
- $\phi \to \psi \equiv \neg\phi \vee \psi$

Part II Discrete Math: Set Theory

Chapter III The definition of set

(Omitted)

Chapter IV Relations

§ 4.1 Relations

1 Properties of relations

Define 4.1.1 Given R, a relation on A,

- **Reflexive** on A if it holds that $\forall a \in A, (aRa) \Leftrightarrow I_A \subseteq R$
- **Symmetric** on A if it holds that $\forall a, b \in A$ if aRb, then $bRa \Leftrightarrow R^{-1} = R$
- **Transitive** on A if it holds that $\forall a, b, c \in A$ if aRb, bRc, then $aRc \Leftrightarrow R \circ R \subseteq R$
- **Antisymmetric** on A if it holds that $\forall a, b \in A$ if aRb, bRa, then $a = b \Leftrightarrow R \cap R^{-1} = I_A$

2 Equivalence relations

Define 4.1.2 If $R \subseteq A \times A$ is reflexive, symmetric and transitive, then R is called a **equivalence relation** on A

§ 4.2 Relations and Sets

1 Equivalence classes and Partitions

Define 4.2.1 R is an equivalence relation on A, $a \in A$, then we define the equivalence class $[a]_R$ of A by

$$[a]_R = \{b \in A | bRa\}$$

Theorem 4.2.1 aRb iff. $[a]_R = [b]_R$

2 Transitive Closures and Reflexive Transitive Closures

Define 4.2.2 (Transitive Closures) Suppose R is a relation on A, R' is a transitive closure of R if

- $R \subseteq R'$
- R' is transitive
- $\forall T$, T is transitive, $R \subseteq T$, then $R' \subseteq T$.

Define 4.2.3 (Another definition) $R^+ = \bigcup\limits_{n=1}^{\infty} R^n$ is the transitive closure

Proof Let's prove that the two definitions are equivalent.

- $R \subseteq R^+$
- If aR^+b, bR^+c, then there exists m, n, aR^mb, bR^nc, then $aR^{m+n}c$, R^+ is transitive.
- If $R \subseteq T$ and T is transitive, if $R^n \subseteq T$, then $R^{n+1} = R^n \circ R \subseteq T \circ T \subseteq T$, so $R^+ = \bigcup\limits_{n=1}^{\infty} R^n \subseteq T$.

So such R^+ is a transitive closure. $\square$

Chapter V Functions

§ 5.1 Functions

§ 5.2 Funcions and Sets

Define 5.2.1

$F : A \to B$,

- **Injection**(one-to-one map): $\forall a, a' \in A$, if $F(a) = F(a')$, then $a = a'$.
- **Surjection**(onto map): $\forall b \in B, \exists a \in A, F(a) = b$.
- **Bijection**(one-to-one correspondence): both one-to-one and onto.

Theorem 5.2.1

- If F, G are both injections, then $F \circ G$ is also an injection.
- If F, G are both surjection, then $F \circ G$ is also a surjection.
- If $F \circ G$ is an injection, then G is also an injection.
- If F is an bijection, then F^{-1} is also a bijection.

Theorem 5.2.2 (Berstern's Theorem) If there exist an injection from A to B and an injection from B to A, then there exists a bijection between A and B

Proof Suppose F is an one-to-one function from A to B, G is an one-to-one function from B to A.
Then we can construct a sequence of set as follow:

$$\begin{aligned} C_0 &= \{a \in A | \forall b \in B, G(b) \neq a\} = A \setminus \{a | \exists b \in B, G(b) = a\}, \\ D_0 &= \{F(a) | a \in C_0\} = B \setminus \{b \in B | \exists a \in A \setminus C_0, b = F(a)\} \end{aligned}$$

$\forall n \geqslant 1$,

$$\begin{aligned} C_n &= \left\{ a \in A | \forall b \in B \setminus \bigcup_{i=0}^{n-1} D_i, G(b) \neq a \right\} \\ D_n &= \{F(a) | a \in C_n\} \end{aligned}$$

Now we define a function H, where

$$H(a) = \begin{cases} F(a), & a \in \bigcup_{n=0}^{\infty} C_n \\ b\ (a = G(b)), & a \notin \bigcup_{n=0}^{\infty} C_n \end{cases}$$

Let $C = A \setminus \bigcup_{n=0}^{\infty} C_n$, $D = B \setminus \bigcup_{n=0}^{\infty} D_n$

Now we prove that H is well-defined and is a bijection.

- Firstly we prove that such b exists.

  $\forall a \in C, a \notin C_0$, so $\exists b \in B$, $G(b) = a$. If $b \in D_n$, then $a = G(b) \in C_{n+1}$, contradictive! So $b \in D$. Due to G is an injection, such b is unique.

- Then we prove that H is an injection.

  $\forall a \in \bigcup_{n=0}^{\infty} C_n, F(a) \in \bigcup_{n=0}^{\infty} D_n$, and due to F is an injection, on $\bigcup_{n=0}^{\infty} C_n$ H is an injection.

  $\forall a \in C, \exists b \in D, a = G(b)$, due to G is an injection, on C H is an injection.

- Finallty we prove that H is a surjection.

  $\forall b \in \bigcup_{n=0}^{\infty} D_n$ according to the define. $\forall b \in D, \exists a \in A, G(b) = a$, so $a \notin C_0$. If $a \in C_n, n \geqslant 1$, then $b \in D_{n-1}$, contradictive! So $a \in C$.

$\square$