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 $\Phi = \{\ \}$, $\psi = p \lor \neg p$, $\Phi \models \psi$

2 Important properties

Theorem 1.2.1

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

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

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 asser-tions, 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