Propositional Logic – Semantic (part 2)

1. Satisfiability of a set of formulas

- Let Σ be a set of formulas$\{P_1, P_2, \ldots, P_n\}$
- A set Σ is said to be satisfiable if and only if given the truth table of all formulas, there exists at least one row where all these formulas $P_1, P_2, \ldots, P_n$are true simultaneously.

Example 1:

- Let Σ ={p⇒q, p∧q, p∨q}
- The set Σ is satisfiable by I(p)=1 and I(q)=1.

p	q	p ⇒ q	p ∧ q	p ∨ q
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	**1**	**1**	**1**

Example 2:

- Let Σ ={p⇒q, q⇒p, q}¬
- The set Σ is satisfiable by I(p)=0, I(q)=1

p	q	p ⇒ q	q ⇒ ¬p
0	0	1	1
0	1	1	1
1	0	0	1
1	1	1	0

2. Logical consequence

- Q is said to be a semantic consequence of P if every model of P is a model of Q.
- We note: P⊨ Q

Example:

- Show that: $p \wedge q \vDash p \vee q$

p	q	$p \wedge q$	$p \vee q$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	**1**	**1**

Property:

- Let P⊨Q be if and only if P ⇒Q is valid

Example:

- Show that:$p \wedge q \vDash p \vee q$

p	q	$p \wedge q$	$p \vee q$	$(p \wedge q) \Rightarrow (p \vee q)$
0	0	0	0	**1**
0	1	0	1	**1**
1	0	0	1	**1**
1	1	**1**	**1**	**1**

- We say that Q is a semantic consequence of (or of the set Σ) if in the truth table of Q is true whenever $P_1, P_2, \ldots, P_n$ are true.
- We note: $P_1, P_2, \ldots, P_n$ ⊨Q or Σ ⊨Q.
- To check the consequence, we must check the truth value of Q is true on all lines where the formulas $P_1, P_2, \ldots, P_n$ are simultaneously true.

Example 1:

- We check using the truth table if {(p⇒¬q) , q }⊨¬p
- Σ ={p⇒¬q, q}
- Q: ¬p

p	q	p ⇒ ¬q	¬p
0	0	1	1
0	**1**	**1**	**1**
1	0	1	0
1	1	0	0

Example 2:

- Socrates said: "If I am guilty, I must be punished;

I am guilty. Therefore, I must be punished."

Is the argument logical?

- p: I am guilty
- q: I must be punished
- {p⇒q, p}⊨ q
- So, this consequence is correct.

p	q	p ⇒ q
0	0	1
0	1	1
1	0	0
1	**1**	**1**

Properties:

- Let $P_1, P_2, \ldots, P_n$ ⊨ Q be if and only if ⊨ $\boldsymbol{P_1 \wedge P_2 \wedge \ldots \wedge P_n \Rightarrow Q}$

Example:

- {(p⇒ ¬q) , q} ⊨¬p

p	q	p ⇒ ¬q	¬p	(p ⇒ ¬q) ∧ q	(p ⇒ ¬q) ∧ q ⇒ ¬p
0	0	1	1	0	1
0	1	1	1	1	1
1	0	1	0	0	1
1	1	0	0	0	1

Properties:

- Σ ⊨Q if and only if $\Sigma \cup \{\neg Q\}$ is unsatisfiable.

Example:

- {(p⇒¬q) , q} ⊨¬ p

The set {(p⇒ ¬q), q,¬ ¬p} is unsatisfiable.

p	q	p ⇒ ¬q
0	0	1
0	1	1
1	0	1
1	1	0

3. Conjunctive and disjunctive normal forms

- Literal: is either an atom or its negation, p or $\neg p$.
- Clause: A clause is a disjunction of literals. $(l_1 \vee \ldots \vee\ l_n)$
    - Example: $p_1 \vee p_2 \vee \neg p_3$
- Monomial: A monomial is a conjunction of literals. $(l_1 \wedge \ldots \wedge\ l_n)$
    - Example: $p_1 \wedge \neg p_2$

Conjunctive Normal Form (CNF):

- A formula is said to be in conjunctive normal form if it is a conjunction of clauses.

$F\ = C_1 \wedge C_2 \wedge \ldots\ \wedge C_n$ such as is a clause C_i

- Example:
    - F $= (p_1 \vee p_2 \vee \neg p_3) \wedge (\neg p_1 \vee p_4) \wedge (\neg p_1 \vee p_3 \vee \neg p_4)$

Disjunctive Normal Form (DNF):

- A formula is said to be in disjunctive normal form if it is a monomial disjunction.

$F\ = M_1 \vee M_2 \vee \ldots\ \vee M_n$ such that is a monomial M_i

- Example:
    - F $= (\neg p_1 \wedge p_2 \wedge p_4) \vee (p_2 \wedge \neg p_3)$

Features:

- The formula in normal form (CNF/DNF) is equivalent to the initial formula.
- The conjunctive/disjunctive normal form of a formula is not unique.
- If formula F is an antilogy then CNF(F) ≡0.

Transformation methods:

There are two techniques for determining a conjunctive or disjunctive normal form of a given formula:

- Use equivalence rules
- Use the minterms / maxterms method

Method 1: Use equivalence rules

- We use equivalence chains by systematically applying the following rules:
  - Replacing the connectors ⇒ and ⇔ byh ∧ Or ∨.
  - Get rid of double negations.
  - Distributivity: Convert p∨ (q ∧ r) to (p ∨ q) ∧ (p ∨ r) for CNF or inverse for DNF.
- **Example 1:**Convert the following formulas into CNF: $\big((q \vee r) \Rightarrow s\big) \vee t$
  - $(\neg(q \vee r) \vee s) \vee t$(Material implication)
  - $\big((\neg q \wedge \neg r) \vee s\big) \vee t$(DeMorgan's Law)
  - $\big((\neg q \vee s) \wedge (\neg r \vee s)\big) \vee t$(distributivity)
  - $\big((\neg q \vee s) \vee t\ ) \wedge ((\neg r \vee s) \vee t\ \big)$ (distributivity)
  - $()\neg q \vee s \vee t) \wedge (\neg r \vee s \vee t)$ (associativity) CNF
- **Example 2:** Convert the following formulas into CNF: $(\neg p \Rightarrow q) \Rightarrow r$
  - $(\neg\neg p \vee q) \Rightarrow r$(Material implication)
  - $(p \vee q) \Rightarrow r$(law of double negative)
  - $\neg(p \vee q) \vee r$(Material implication)
  - $(\neg p \wedge \neg q) \vee r$ (De Morgan's law) DNF
  - $(\neg p \vee r) \wedge (\neg q \vee r)$(distributivity) CNF
  - $(\neg p \wedge \neg q\ ) \vee (\neg p \wedge r\ ) \vee (r \wedge \neg q\ ) \vee r$ (distributivity, Idempotence) DNF

Method 2: Use the minterms / maxterms method

Every truth table (Boolean function) can be written as either a conjunctive normal form (CNF) or disjunctive normal form (DNF).

- First, we must construct the truth table of the initial form.
- Minterme: is a conjunction of literals which the interpretations I(P) = 1 (for each True line).
- Maxterm: is a disjunction of negations of literals which the interpretations I(P) = 0 (for each False line).
- DNF: Formula as a disjunction of minterms.
- CNF: Formula as conjunction of maxterms.

Example 1:

p	q	Formule	Min terme	Max terme
0	0	**1**	$\neg p \wedge \neg q$	
0	1	**1**	$\neg p \wedge q$	
1	0	**0**		$\neg(p \wedge \neg q) = \neg p \vee q$
1	1	**0**		$\neg(p \wedge q) = \neg p \vee \neg q$

DNF: $(\neg p \wedge \neg q) \vee (\neg p \wedge q)$

CNF: $(\neg p \vee q) \wedge (\neg p \vee \neg q)$

Example 2: Convert the following formula into DNF: $\neg(p \wedge q)$

p	q	$p \wedge q$	$\neg(p \wedge q)$	Minterme	Maxterme
0	0	**0**	**1**	¬p ∧ ¬q	
0	1	**0**	**1**	¬p ∧q	
1	0	**0**	**1**	p ∧ ¬q	
1	1	**1**	**0**		¬p ∨ ¬q

- **DNF:** $(\neg p \wedge \neg q) \vee (\neg p \wedge q) \vee (p \wedge \neg q)$

Exercise: Convert the following formulas into CNF

- (p ∨ q) ⇒ (p ∧ q)
- (p ∧ q) ⇔ (¬p ⇒ q)