Propositional logic : Semantics

The semantics of propositional logic is concerned with determining the truth value of a statement, that is, a formula. This involves the interpretation of a formula, which more concretely means assigning a true or false value to each of the propositional variables that compose it. For a formula with n variables, there are 2*n possibl interpretations . To achieve this, we use what are called truth tables. (See Chapter 01.)

1. **An Interpretation:**

An interpretation is a function that assigns a truth value to each propositional variable.

2. **A truth table:**

is a table with several columns. The values in the cells of this table are called "truth values" (1 or T for true, 0 or F for false) in logic. The left columns define the truth values of different propositions in mathematics (propositional logic). The right column indicates the truth value of the logical expression in mathematics or logic.

3. Categories of Formulae

- **Model**: A model is an interpretation for which a formula is true.
- **Consistency**: A formula AAA is said to be consistent, or satisfiable, or verifiable, if there exists an interpretation of its propositional variables that makes it true. In other words, the formula AAA is consistent if and only if the formula AAA has a model.
- **Inconsistency**: A formula for which there is no interpretation that makes it true is said to be inconsistent, or unsatisfiable, or unverifiable, or more simply false. The formula $(A \wedge \neg A)$ is inconsistent.
- **Tautology**: A valid formula, or tautology, is a formula that is true regardless of the truth values of the atoms that compose it (true in every interpretation). It is denoted as $\models A$. The formula $(A \vee \neg A)$ is a tautology.
- **Antilogies**: An unsatisfiable, falsifiable, or semantically inconsistent formula, also known as an antilogy, is a formula that is false in every interpretation.
- **Invalid Formulas**: An invalid formula is false in at least one interpretation.
- **Contingent Formulas**: A contingent formula is true in some interpretations and false in others.
- **Logical Consequence**: Let A and Bbe two formulae. We say that B is a valid consequence of A, denoted $(A \models B)$, if every model of A is a model of B.

Exercise

1. Draw the truth table of the following formula

$A \Rightarrow (B \wedge C)$

2. Determine the categories of the formula $A \Rightarrow (B \wedge C)$
3. What we can conclude about this formula $B \wedge C \models A \Rightarrow (B \wedge C)$

4. CNF (Conjunctive Normal Form) and DNF (Disjunctive Normal Form)

CNF

A formula in conjunctive normal form (CNF) is a conjunction of clauses, where a clause is a disjunction of literals. CNF formulae are used in the context of automated theorem proving and in solving the SAT problem (particularly in the DPLL algorithm).

Example
All of the following expressions are in CNF:

- A ∧ B
- A
- (A ∨B) ∧C
- (A ∨ ¬B ∨¬C ∨ ¬D) ∧ (¬D∨ E∨F)

We obtained the CNF of a formula by applying one of the following methods

4.1 CNF Conversion Algorithm

Start

1. Eliminate all occurrences of the connectors → and ↔ by replacing:
   F→G with ¬F∨G
   and F↔G with (¬F∨G)∧(¬G∨F)
2. Apply De Morgan's laws to push negations inward by replacing:
   ¬(F∨G)\ with ¬F∧¬G
   ¬(F∧G) with ¬F∨¬G
3. Eliminate double negations by replacing ¬¬F with F.
4. Apply the distributive laws by replacing:
   F∨(G∧H)F with (F∨G)∧(F∨H)
   (F∧G)∨Hwith (F∨H)∧(G∨H)

4.2 Using truth table

To obtain the Conjunctive Normal Form (CNF) of a logical formula using a truth table, follow these steps:

Steps to Generate CNF from a Truth Table

1. **Construct the Truth Table**:
   - List all possible combinations of truth values for the variables in the formula.
   - Determine the output of the formula for each combination.
2. **Identify Rows for Output False** :
   - Focus on the rows where the output of the formula is **false** (i.e., where the formula evaluates to 0). These rows will help us form the clauses for CNF.
3. **Create Clauses**:
   - For each row where the output is false, create a clause that represents that row:

        - If a variable is true (1) in that row, include its negation (¬) in the clause.
        - If a variable is false (0) in that row, include the variable itself in the clause.
    - For example, if the row is P=T,Q=F,R=T (which gives an output of false), the corresponding clause would be ¬P∨Q∨¬R.
4. **Combine Clauses**:
    - Combine all the clauses from the false output rows using the logical AND operator (∧). This results in the CNF.

Example

Consider a formula F(P,Q)=(P∧Q)∨¬Q.

1. **Truth Table**:

P	**Q**	**F(P,Q)**
T	T	T
T	F	T
F	T	F
F	F	F

2. **Identify False Outputs**:
    - The outputs are false for the following rows:
        - Row 3: P=F,Q=T
        - Row 4: P=F,Q=F
3. **Create Clauses**:
    - For Row 3: P=F,Q=T → Clause: P∨ ¬Q
    - For Row 4: P=F,Q=F → Clause: P∨Q
4. **Combine Clauses**:
    - CNF: (P∨ ¬Q)∧(P∨Q)

5. DNF

A disjunctive normal form (DNF) is a normalization of a logical expression that is a disjunction of conjunctive clauses. It is used in automated theorem proving. A logical expression is in DNF if and only if it is a disjunction of one or more conjunctions of one or more literals.

Example

All of the following expressions are in DNF:

- A ∨ B
- A
- (A ∧ B) ∨C
- (A ∧ ¬B ∧¬C ∧ ¬D) ∨(¬D∧ E∧F)

5.1 DNF Conversion Algorithm

The same as the CNF conversion method, except that in steps 2 and 4, the ∧ is replaced by ∨ and the ∨ is replaced by ∧.

5.2. Using truth table

Steps to Generate DNF from a Truth Table

1. **Construct the Truth Table**:
   - List all possible combinations of truth values for the variables in the formula.
   - Determine the output of the formula for each combination.
2. **Identify Rows for Output True**:
   - Focus on the rows where the output of the formula is **true** (i.e., where the formula evaluates to 1). These rows will help us form the terms for DNF.
3. **Create Minterms**:
   - For each row where the output is true, create a minterm that represents that row:
     - If a variable is true (1) in that row, include the variable itself in the minterm.
     - If a variable is false (0) in that row, include its negation (¬) in the minterm.
   - For example, if the row is P=T,Q=F,R=TP = T, Q = F, R = TP=T,Q=F,R=T (which gives an output of true), the corresponding minterm would be P∧¬Q∧RP \land \neg Q \land RP∧¬Q∧R.
4. **Combine Minterms**:
   - Combine all the minterms from the true output rows using the logical OR operator (∨). This results in the DNF.

Example

Consider a formula F(P,Q)=(P∧Q)∨¬Q.

1. **Truth Table**:

P	**Q**	**F(P,Q)**
T	T	T
T	F	T
F	T	F
F	F	F

2. **Identify True Outputs**:
   - The outputs are true for the following rows:
     - Row 1: P=T,Q=T
     - Row 2: P=T,Q=F
3. **Create Minterms**:
   - For Row 1: P=T,Q=T → Minterm: P∧Q
   - For Row 2: P=T,Q=F → Minterm: P∧¬Q
4. **Combine Minterms**:

- DNF: (P∧Q)∨(P∧¬Q)