Chapter 2: Propositional Calculus: Formulas, Models, Tableaux

August 22, 2008

Outline

2.1 Boolean Operators

- **Boolean type:** T (true), F (false)
- **Boolean operator:** a function on the set $\{T,F\}$.
  These operators can be *unary*, *binary*, etc.
- **Question:** How many *n*-ary Boolean operators are there on $\{T,F\}$?

$$2^{2^n}$$

- We single out the following five operators:

$$\neg \qquad \text{(unary)}$$

$$\vee, \wedge, \rightarrow, \leftrightarrow \qquad \text{(binary)}$$

- There are other binary operators that are sometimes used e.g. in the theory of Boolean circuits:

$$\begin{array}{ll} \oplus & \text{(XOR, exclusive OR)} \\ \uparrow & \text{(NAND)} \\ \downarrow & \text{(NOR, Sheffer's stroke)} \end{array}$$

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

p	q	$p \uparrow q$
T	T	F
T	F	T
F	T	T
F	F	T

p	q	$p \downarrow q$
T	T	F
T	F	F
F	T	F
F	F	T

2.2 Propositional Formulas

BNF (Backus-Naur Form) Grammars:

- rules of the form

$$\mathit{symbol} ::= \mathit{symbol}_1\, \mathit{symbol}_2 \ldots \mathit{symbol}_n, \qquad \text{or}$$
$$\mathit{symbol} ::= \mathit{symbol}_1 | \mathit{symbol}_2 | \ldots | \mathit{symbol}_n$$

- *symbol* is a non-terminal symbol of the grammar.
- symbols that can never occur on the left-hand side of a grammar rule are called **terminal** symbols.

- $\mathcal{P}$ - set of all propositional letters (atoms)

$$\mathcal{P} = \{p, q, r, \ldots\}$$

Definition

A formula in the propositional logic is any string that can be derived from the initial non-terminal *fml* using the following BNF rules:

1. $fml ::= p$, for any $p \in \mathcal{P}$
2. $fml ::= (\neg fml)$
3. $fml ::= (fml \vee fml)$
4. $fml ::= (fml \wedge fml)$
5. $fml ::= (fml \rightarrow fml)$
6. $fml ::= (fml \leftrightarrow fml)$

Remark

If we want to use additional operators such as e.g. $\oplus$, $\uparrow$, $\downarrow$, etc, the BNF grammar can be easily modified by adding appropriate rules to handle these connectives.

Example

Derivation of

$$(p \wedge (r \rightarrow (p \vee (\neg q))))$$

$$\begin{aligned}
\mathit{fml} &::= (\mathit{fml} \wedge \mathit{fml}) \\
&::= (p \wedge \mathit{fml}) \\
&::= (p \wedge (\mathit{fml} \rightarrow \mathit{fml})) \\
&::= (p \wedge (r \rightarrow \mathit{fml})) \\
&::= (p \wedge (r \rightarrow (\mathit{fml} \vee \mathit{fml}))) \\
&::= (p \wedge (r \rightarrow (p \vee \mathit{fml}))) \\
&::= (p \wedge (r \rightarrow (p \vee (\neg \mathit{fml})))) \\
&::= (p \wedge (r \rightarrow (p \vee (\neg q))))
\end{aligned}$$

- **Derivation tree:** tree representing the derivation of the formula using the BNF grammar for propositional logic.
- **Formation tree:** tree representing the structure of the formula; i.e. the tree whose nodes are the connectives occurring in the formula and whose leaves are propositional variables.

Remark

For any formula, the formation tree can be easily obtained from its derivation tree; namely, replace the fml symbol in every node of the derivation tree by the connective used in the rule applied to fml.

Convention: We can omit writing unnecessary pairs of brackets in a propositional formula, if we introduce the following hierarchy (order of priority) of the Boolean connectives:

1. $\neg$
2. $\vee, \quad \wedge$
3. $\rightarrow, \quad \leftrightarrow$

Definition

If the propositional formula A is not an atom (variable), the operator at the root of its formation tree is called the **principal operator** of A.

Theorem

(Structural Induction) To show that some property holds for all propositional formulas A, it suffices to show the following:

1. *every atom (variable) p has the property.*
2. *assuming that a formula A has the required property, show that*

   $$\neg A$$

   has the property.
3. *assuming that the formulas A and B have the required property, show that the formulas*

   $$A \vee B, \qquad A \wedge B, \qquad A \rightarrow B, \qquad A \leftrightarrow B$$

   have the property.

Example

Prove that every formula A, formed using BNF form for propositional formulas, is *balanced*; i.e. A contains the same number of left and right brackets.

Proof.

We will prove this by structural induction.
1. any atom (variable) p is trivially balanced, since it contains no left or right brackets.
2. assume A is a balanced propositional formula, i.e. A contains the same number of left and right brackets.
Consider $\neg A$. Since A is balanced, so is $\neg A$.
3. suppose A and B are both balanced formulas. Consider, say, $A \vee B$. Clearly, the number of left brackets in $A \vee B$ is the sum of the left brackets in A and B, and similarly for right brackets. Since both A and B are balanced, it is easy to see that this holds for $A \vee B$, too. [Similarly for other three connectives $\wedge, \rightarrow, \leftrightarrow$.] $\square$

2.3 Interpretations

- $\mathcal{P}$ - set of all propositional variables (atoms)

Definition

An assignment is a function

$$v : \mathcal{P} \rightarrow \{\text{T,F}\}$$

v assigns a truth value to any atom in a given formula.

Suppose $\mathcal{F}$ denotes the set of all propositional formulas. We can extend an assignment v to a function

$$v : \mathcal{F} \rightarrow \{\text{T,F}\},$$

which assigns the truth value $v(A)$ to any formula $A \in \mathcal{F}$.

Example

Suppose v is an assignment for which

$$v(p) = \mathrm{F}, \qquad v(q) = \mathrm{T}.$$

If

$$A = (\neg p \to q) \leftrightarrow (p \vee q)$$

what is $v(A)$?

Solution:

$$\begin{aligned}
v(A) &= v((\neg p \to q) \leftrightarrow (p \vee q)) \\
&= v(\neg p \to q) \leftrightarrow v(p \vee q) \\
&= (v(\neg p) \to v(q)) \leftrightarrow (v(p) \vee v(q)) \\
&= (\neg v(p) \to v(q)) \leftrightarrow (v(p) \vee v(q)) \\
&= (\neg \mathrm{F} \to \mathrm{T}) \leftrightarrow (\mathrm{F} \vee \mathrm{T}) \\
&= (\mathrm{T} \to \mathrm{T}) \leftrightarrow (\mathrm{F} \vee \mathrm{T}) \\
&= \mathrm{T} \leftrightarrow \mathrm{T} \\
&= \mathrm{T}
\end{aligned}$$

Theorem

An assignment can be extended to exactly one interpretation. In other words: for a given set of truth values of atoms, the truth value of a formula is uniquely determined.

- In fact: if two assignments agree on all atoms that appear in the formula, the interpretations they induce also agree on that formula.

Suppose

$$S = \{A_1, A_2, \ldots, A_n\}$$

is a set of formulas and v is an assignment which assigns truth values to all atoms that appear in the set of formulas S. Any interpretation that extends v to all propositional atoms $\mathcal{P}$ will be called an **interpretation** for S.

Example

The assignment

$$v(p) = \mathsf{F}, \qquad v(q) = \mathsf{T}, \qquad v(r) = \mathsf{T}$$

determines the following interpretation of the set of formulas

$$S = \{p \vee \neg q, \quad q, \quad p \wedge r \leftrightarrow (r \rightarrow q)\}$$

$$v(p \vee \neg q) = \mathsf{F}, \qquad v(q) = \mathsf{T}, \qquad v(p \wedge r \rightarrow (r \rightarrow q)) = \mathsf{F}$$

2.4 Equivalence and Substitution

Definition

If $A, B \in \mathcal{F}$ are such that

$$v(A) = v(B)$$

for **all** interpretations v, A is (logically) equivalent to B.

$$A \equiv B$$

Example

$$\neg p \vee q \equiv p \rightarrow q$$

since both formulas are true in all interpretations except when

$$v(p) = \mathrm{T}, \qquad v(q) = \mathrm{F}$$

and are false for that particular interpretation.

Caution: $\equiv$ **does not** mean the same thing as $\leftrightarrow$:

- $A \leftrightarrow B$ is a formula (syntax)
- $A \equiv B$ is a relation between two formula (semantics)

Theorem

$A \equiv B$ if and only if $A \leftrightarrow B$ is true in every interpretation; i.e. $A \leftrightarrow B$ is a tautology.

Definition

A is a subformula of B if it is a formula occurring within B; i.e. the formation tree for A is a subtree of the formation tree for B.

Example

The subformulas of

$$p \wedge (r \leftrightarrow p \vee \neg q)$$

are

$$p \wedge (r \leftrightarrow p \vee \neg q), \quad p, \quad r \leftrightarrow p \vee \neg q, \quad r, \quad p \vee \neg q, \quad \neg q, \quad q$$

Definition

Suppose A is a subformula of B, and A' is any formula. Then, we say that B' is a formula that results from substitution of A' for A in B, and we write it as

$$B' = B\{A \leftarrow A'\}$$

if we obtain B' from B by replacing all occurrences of A in B with A'.

Example

Suppose

$$B = (p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p), \quad A = p \rightarrow q, \quad A' = \neg p \vee q$$

Then,

$$B' = B\{A \leftarrow A'\} = (\underline{\neg p \vee q}) \leftrightarrow (\neg q \rightarrow \neg p)$$

Theorem

Let A be a subformula of B, and let A' be a formula such that $A \equiv A'$. Then

$$B \equiv B\{A \leftarrow A'\}$$

Proof.

By induction on the depth of the highest occurrence of the formation tree of A as a subtree of B. $\square$

Logically Equivalent Formulas

$A \equiv \neg\neg A$

$A \vee B \equiv B \vee A$ $\qquad$ $A \wedge B \equiv B \wedge A$

$(A \vee B) \vee C \equiv A \vee (B \vee C)$ $\qquad$ $(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$

$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$

$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$

$\neg(A \wedge B) \equiv \neg A \vee \neg B$ $\qquad$ $\neg(A \vee B) \equiv \neg A \wedge \neg B$

$A \wedge \mathit{true} \equiv A$ $\qquad$ $A \rightarrow \mathit{false} \equiv \neg A$

Example

Simplify

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

Solution:

$$\begin{aligned} p \vee (\neg p \wedge q) &\equiv (p \vee \neg p) \wedge (p \vee q) \\ &\equiv \mathsf{T} \wedge (p \vee q) \\ &\equiv p \vee q \end{aligned}$$

Adequate Sets of Connectives

Definition

A set of connectives is adequate if it generates all possible Boolean functions.

Example

The usual set of connectives

$$\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$$

is adequate for propositional logic, since every Boolean function can be generated from these five operators. [A nontrivial fact!]

Example

1. The set $\{\neg, \wedge, \vee\}$ is adequate.

$$
\begin{aligned}
A \rightarrow B &\equiv \neg A \vee B \\
A \leftrightarrow B &\equiv (A \rightarrow B) \wedge (B \rightarrow A) \\
&\equiv (\neg A \vee B) \wedge (\neg B \vee A)
\end{aligned}
$$

2. $\{\neg, \wedge\}$ is adequate
   We know that $\{\neg, \wedge, \vee\}$ is adequate, so it would suffice to show that $\vee$ can be expressed using $\neg, \wedge$ only:

$$A \vee B \equiv \neg(\neg A \wedge \neg B)$$

3. $\{\neg, \vee\}$ is adequate [Exercise.]
4. $\{\neg, \rightarrow\}$ is adequate.

$$A \vee B \equiv \neg A \rightarrow B$$
$$A \wedge B \equiv \neg(A \rightarrow \neg B)$$

5. $\{\neg, \leftrightarrow\}$ is **not** adequate. [This will be proved in the lab.]

   Hint: Proving the following fact would be useful in order to show inadequacy:
   If *A* is a formula involving at least two atoms, then the number of truth assignments that make *A* true is even and the same is true of the number of truth assignments that make *A* false.

6. $\{\uparrow\}$ is adequate.

   It is enough to show that $\{\uparrow\}$ generates $\neg$ and $\wedge$, since we know that these form an adequate set of connectives:

$$\begin{aligned} \neg A &\equiv A \uparrow A \\ A \wedge B &\equiv \neg(A \uparrow B) \\ &\equiv (A \uparrow B) \uparrow (A \uparrow B) \end{aligned}$$

2.5 Satisfiability, Validity, and Consequence

Definition

We say that a propositional formula A is satisfiable if and only if $v(A) = \mathrm{T}$ in **some** interpretation v. Such an interpretation is called a model for A.

- A is valid (or, a tautology) if $v(A) = \mathrm{T}$, for **all** interpretations v

$$\models A$$

- A is unsatisfiable (or, contradictory) if it is false in every interpretation.
- A is not valid (or, falsifiable), if we can find **some** interpretation v, such that $v(A) = \mathrm{F}$

$$\not\models A$$

Examples

1. $(p \to q) \to (\neg q \to \neg p)$

   Valid (tautology).

2. $q \to (q \to p)$
   Not valid (take $v(p) = \mathrm{F}$, $v(q) = \mathrm{T}$), but it is satisfiable (take e.g. $v(p) = v(q) = \mathrm{T}$).

3. $(p \wedge \neg p) \vee (q \wedge \neg q)$

   False (contradiction).

Theorem

(a) *A is valid if and only if $\neg A$ is unsatisfiable.*

(b) *A is satisfiable if and only if $\neg A$ is falsifiable.*

Definition

Suppose $\mathcal{V}$ is a set of formulas. An algorithm is a decision procedure for $\mathcal{V}$ if, given an arbitrary formula A, the algorithm terminates and returns as the answer either

(a) 'yes, $A \in \mathcal{V}$'; or

(b) 'no, $A \notin \mathcal{V}$'

Main Problem: develop an algorithm which decides whether a propositional formula A is valid or not (So, the set $\mathcal{V}$ in this particular problem is the set of all valid propositional formulas.)

- **Truth-Table Method:** provides a decision algorithm but it is too time-consuming; in general, it requires exponential time for the majority of propositional formulas.
- A different approach: instead of A, consider $\neg A$ and try to decide whether $\neg A$ is satisfiable or not.

Definition

Let $U = \{A_1, A_2, \ldots, A_n\}$ be a set of formulas.
We say that U is satisfiable if we can find an interpretation v such that

$$v(A_1) = v(A_2) = \ldots = v(A_n) = \mathrm{T}$$

Such an interpretation is called a model for U. U is unsatisfiable if no such interpretation exists.

Facts

1. If U is satisfiable, then so is $U - \{A_i\}$ for any $i = 1, 2, \ldots, n$.
2. If U is satisfiable and B is valid, then $U \cup \{B\}$ is also satisfiable.
3. If U is unsatisfiable and B is **any** formula, $U \cup \{B\}$ is also unsatisfiable.
4. If U is unsatisfiable and some A_i is valid, then $U - \{A_i\}$ is also unsatisfiable.

Definition

Let U be a set of formulas and A a formula. We say that A is a (logical) consequence of U, if any interpretation v which is a model of U is also a model for A.

$$U \models A$$

Example

$$\{p \wedge r, \neg q \vee (p \wedge \neg p)\} \models (p \wedge \neg q) \rightarrow r$$

If some interpretation v is a model for the set $\{p \wedge r, \neg q \vee (p \wedge \neg p)\}$, it must satisfy

$$v(p) = v(r) = \mathrm{T}, \qquad v(q) = \mathrm{F}$$

but in this interpretation, we also have

$$v((p \wedge \neg q) \rightarrow r) = \mathrm{T}$$

Theorem

1. $U \models A$ *if and only if*

$$\models (A_1 \land A_2 \land \ldots \land A_n) \rightarrow A$$

2. *If* $U \models A$*, then* $U \cup \{B\} \models A$*, for any formula* B*.*
3. *If* $U \models A$ *and* B *is valid, then*

$$U - \{B\} \models A$$

Theories

Definition

A set f formulas $\mathcal{T}$ is a theory if it closed under logical consequence. This means that, for every formula A, if

$$\mathcal{T} \models A,$$

then $A \in \mathcal{T}$.

- Let U be a set of formulas. Then, the set of all consequences of U

  $$\mathcal{T}(U) = \{A \mid U \models A\}$$

  is called the theory of U.
  The formulas in U are called the axioms for the theory $\mathcal{T}(U)$.

2.6 Semantic Tableaux

- a more efficient method for deciding satisfiability of a propositional formula than using truth-tables.

Definition

A literal is an atom or its negation.

- atom: **positive** literal
- negation of an atom: **negative** literal

- $\{p, \neg p\}$ - complementary pair of literals
- $\{A, \neg A\}$ - complementary pair of formulas

Example

Consider the formula

$$A = p \wedge (\neg q \vee \neg p).$$

When is $v(A) = \mathrm{T}$?

1. First, we must have

$$v(p) = \mathrm{T}, \qquad v(\neg q \vee \neg p) = \mathrm{T}$$

2. This splits into two cases; either
   (a) $v(p) = \mathrm{T}$, $v(\neg q) = \mathrm{T}$; or
   (b) $v(p) = \mathrm{T}$, $v(\neg p) = \mathrm{T}$.

   and the second case is clearly impossible.

So, the truth assignment

$$v(p) = \mathrm{T}, \qquad v(q) = \mathrm{F}$$

makes A true, showing that A is satisfiable.

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

$$p, \neg q \vee \neg p$$

$$\begin{array}{cc} p, \neg q & p, \neg p \\ \odot & \times \end{array}$$

General Idea: Given a formula A, first transform it into an equivalent formula, which is a disjunction of conjunctions of literals.
After this, we can analyze this new form of A to see if we can construct a truth assignment v, such that $v(A) = \mathrm{T}$. If there is one, A is satisfiable; if there is no such v, A is not satisfiable.

Example

Determine if

$$B = (p \vee q) \wedge (\neg p \wedge \neg q)$$

is satisfiable.

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

$$p \vee q, \neg p \wedge \neg q$$

$$p, \neg p \wedge \neg q \qquad q, \neg p \wedge \neg q$$

$$p, \neg p, \neg q \qquad q, \neg p, \neg q$$

$$\times \qquad \times$$

- This is another example of a semantic tableau.

- In order to use this method, we had to rewrite the formula using $\neg$, $\vee$, and $\wedge$ only.
- The method can be made more general if we can also eliminate the connectives $\rightarrow$ and $\leftrightarrow$ within a tableau.

α	α_1	α_2
$\neg\neg A$	A	
$A_1 \wedge A_2$	A_1	A_2
$\neg(A_1 \vee A_2)$	$\neg A_1$	$\neg A_2$
$\neg(A_1 \rightarrow A_2)$	A_1	$\neg A_2$
$A_1 \leftrightarrow A_2$	$A_1 \rightarrow A_2$	$A_2 \rightarrow A_1$

β	β_1	β_2
$\neg(B_1 \wedge B_2)$	$\neg B_1$	$\neg B_2$
$B_1 \vee B_2$	B_1	B_2
$B_1 \rightarrow B_2$	$\neg B_1$	B_2
$\neg(B_1 \leftrightarrow B_2)$	$\neg(B_1 \rightarrow B_2)$	$\neg(B_2 \rightarrow B_1)$

Algorithm (Construction of a Semantic Tableau)

- **INPUT:** formula A
- **OUTPUT:** a tableau $\mathcal{T}$ for A, all of whose leaves are marked as open or closed.

- Initially, $\mathcal{T}$ is a single node (root) labeled $\{A\}$.
- We build the tableau inductively, by choosing an unmarked leaf l which is labeled by a set of formulas $U(l)$, and apply one of the following rules:

1. If $U(l)$ is just a set of literals, check if it contains a pair of complementary literals. If it does, mark the leaf as closed $(\times)$; if not, mark it as open $(\odot)$
2. If $U(l)$ is not just a set of literals, choose one formula in $U(l)$ which is not a literal.
   (a) if one of the α-rules applies, replace $U(l)$ with $(U(l) - \{\alpha\}) \cup \{\alpha_1, \alpha_2\}$.
   (b) if one of the β-rules applies, replace $U(l)$ with two descendent nodes labeled $(U(l) - \{\beta\}) \cup \{\beta_1\}$ and $(U(l) - \{\beta\}) \cup \{\beta_2\}$.

Definition

A tableau is said to be **completed** if its construction terminates; i.e. eventually, all branches end with leaves containing literals only. It is closed if all its leaves are closed; otherwise, we say that the tableau is open.

Theorem

The construction of a semantic tableau for a propositional formula always terminates.

- this construction can be extended to non-atomically closed tableaux: all leaves eventually contain a pair of complementary formulas $A, \neg A$.

2.7 Soundness and Completeness

Main Theorem: A completed semantic tableau for a formula A is closed if and only if A is unsatisfiable.

- Soundness: If a tableau is closed, then A is unsatisfiable.
- Completeness: If A is unsatisfiable, then any tableau for A is closed.

Corollary

A is a satisfiable formula if and only if any tableau for A is open.

Corollary

A is a valid formula (tautology) if and only if a tableau for $\neg A$ is closed.

Corollary

The method of semantic tableaux is a decision procedure for the validity of formulas in propositional logic.

[Stop at Example 2.54 in the textbook.]