Chapter 4: Classical Propositional Semantics

Language:

\[ \mathcal{L}\{\neg, \cup, \cap, \Rightarrow\} \]

Classical Semantics assumptions:

TWO VALUES: there are only two logical values: truth (T) and false (F), and

EXTENSIONALITY: the logical value of a formula depends only on a main connective and logical values of its sub-formulas.

We define formally a classical semantics for \( \mathcal{L} \) in terms of two factors: classical truth tables and a truth assignment.
We summarize now here the chapter 2 tables for $\mathcal{L}\{\neg,\cup,\cap,\Rightarrow\}$ in one simplified table as follows.

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$\neg A$</th>
<th>$(A \cap B)$</th>
<th>$(A \cup B)$</th>
<th>$(A \Rightarrow B)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

Observe that The first row of the above table reads:

For any formulas $A, B$, if the logical value of $A = T$ and $B = T$, then logical values of $\neg A = T$, $(A \cap B) = T$, $(A \cup B) = T$ and $(A \Rightarrow B) = T$.

We read and write the other rows in a similar manner.
Our table indicates that the logical value of of propositional connectives depends only on the logical values of its factors; i.e. it is independent of the formulas \( A, B \).

**EXTENSIONAL CONNECTIVES**: The logical value of a given connective depend only of the logical values of its factors.

**We write** now the last table as the following equations.

\[
\begin{align*}
\neg T &= F, & \neg F &= T; \\
(T \cap T) &= T, & (T \cap F) &= F, & (F \cap T) &= F, & (F \cap F) &= T; \\
(T \cup T) &= T, & (T \cup F) &= T, & (F \cup T) &= T, & (F \cup F) &= F; \\
(T \Rightarrow T) &= T, & (T \Rightarrow F) &= F, & (F \Rightarrow T) &= T, & (F \Rightarrow F) &= T.
\end{align*}
\]
Observe now that the above equations describe a set of unary and binary operations (functions) defined on a set \( \{T, F\} \) and a set \( \{T, F\} \times \{T, F\} \), respectively.

**Negation** \( \neg \) is a function:

\[
\neg : \{T, F\} \rightarrow \{T, F\},
\]

such that \( \neg T = F, \neg F = T \).

**Conjunction** \( \cap \) is a function:

\[
\cap : \{T, F\} \times \{T, F\} \rightarrow \{T, F\},
\]

such that
\[
(T \cap T) = T, \quad (T \cap F) = F,
\]
\[
(F \cap T) = F, \quad (F \cap F) = F.
\]
**Dissjunction** $\cup$ is a function:

\[
\cup : \{T, F\} \times \{T, F\} \rightarrow \{T, F\},
\]

such that

\[
(T \cup T) = T, \quad (T \cup F) = T,
\]
\[
(F \cup T) = T, \quad (F \cup F) = F.
\]

**Implication** $\Rightarrow$ is a function:

\[
\Rightarrow : \{T, F\} \times \{T, F\} \rightarrow \{T, F\},
\]

such that

\[
(T \Rightarrow T) = T, \quad (T \Rightarrow F) = F,
\]
\[
(F \Rightarrow T) = T, \quad (F \Rightarrow F) = T.
\]

**Observe** that if we have have a language $\mathcal{L}\{\neg, \cup, \cap, \Rightarrow, \Leftrightarrow\}$ containing also the equivalence connective $\Leftrightarrow$ we define

\[
\Leftrightarrow : \{T, F\} \times \{T, F\} \rightarrow \{T, F\},
\]

as a function such that

\[
(T \Leftrightarrow T) = T, \quad (T \Leftrightarrow F) = F,
\]
\[
(F \Leftrightarrow T) = F, \quad (T \Leftrightarrow T) = T.
\]
We write these definitions of connectives as the following tables, usually called the classical truth tables.

<table>
<thead>
<tr>
<th>Negation</th>
<th>Disjunction</th>
</tr>
</thead>
<tbody>
<tr>
<td>¬</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>F</td>
</tr>
<tr>
<td></td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conjunction</th>
<th>Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>∩</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>F</td>
</tr>
<tr>
<td></td>
<td>F</td>
</tr>
<tr>
<td></td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Equivalence</th>
</tr>
</thead>
<tbody>
<tr>
<td>⇔</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
A truth assignment is any function

\[ v : VAR \rightarrow \{ T, F \}. \]

Observe that the truth assignment is defined only on variables (atomic formulas).

We define its extension \( v^* \) to the set \( \mathcal{F} \) of all formulas of \( \mathcal{L} \) as follows.

\[ v^* : \mathcal{F} \rightarrow \{ T, F \} \]

is such that

(i) for any \( a \in VAR \),

\[ v^*(a) = v(a); \]
(ii) and for any $A, B \in \mathcal{F}$,

$$v^*(\neg A) = \neg v^*(A);$$

$$v^*(A \cap B) = v^*(A) \cap v^*(B);$$

$$v^*(A \cup B) = v^*(A) \cup v^*(B);$$

$$v^*(A \Rightarrow B) = v^*(A) \Rightarrow v^*(B),$$

$$v^*(A \iff B) = v^*(A) \iff v^*(B),$$

where the symbols on the left-hand side of the equations represent connectives in their natural language meaning and the symbols on the right-hand side represent connectives in their logical meaning given by the classical truth tables.
Example

Consider a formula

\[ ((a \Rightarrow b) \cup \neg a) \]

a truth assignment \( v \) such that

\[ v(a) = T, v(b) = F. \]

We calculate the logical value of the formula \( A \) as follows:

\[
v^*(A) = v^*((a \Rightarrow b) \cup \neg a) = v^*(a \Rightarrow b) \cup v^*(\neg a) = (v(a) \Rightarrow v(b)) \cup \neg v(a) = (T \Rightarrow F) \cup \neg T = F \cup F = F.
\]

Observe that we did not need (and usually we don’t) to specify the \( v(x) \) of any \( x \in \text{VAR} - \{a, b\} \), as these values do not influence the computation of the logical value \( v^*(A) \).
**SATISFACTION** relation

**Definition:** Let $v : VAR \rightarrow \{T, F\}$. We say that

$v$ satisfies a formula $A \in \mathcal{F}$ iff $v^*(A) = T$

**Notation:** $v \models A$.

**Definition:** We sat that

$v$ does not satisfy a formula $A \in \mathcal{F}$ iff $v^*(A) \neq T$.

**Notation:** $v \not\models A$.

**REMARK** In our classical semantics we have that

$v \not\models A$ iff $v^*(A) = F$ and we say that $v$ falsifies the formula $A$. 
**OBSERVE** $v^*(A) \neq T$ is equivalent to the fact that $v^*(A) = F$ ONLY in 2-valued logic!

This is why we adopt the following

**Definition:** For any $v$,

$v$ does not satisfy a formula $A \in \mathcal{F}$ iff $v^*(A) \neq T$
Example

\[ A = ((a \Rightarrow b) \cup \neg a) \]

\[ v : VAR \longrightarrow \{ T, F \} \]

such that \( v(a) = T, v(b) = F \).

Calculation of \( v^*(A) \) using the short hand notation:

\[ (T \Rightarrow F) \cup \neg T = F \cup F = F. \]

\[ v \not\models ((a \Rightarrow b) \cup \neg a)). \]

Observe that we did not need (and usually we don’t) to specify the \( v(x) \) of any \( x \in VAR - \{a, b\} \), as these values do not influence the computation of the logical value \( v^*(A) \).
Example

\[ A = ((a \cap \neg b) \cup \neg c) \]

\[ v : VAR \rightarrow \{T, F\} \]

such that \( v(a) = T, v(b) = F, v(c) = T \).

**Calculation** in a short hand notation:

\[ (T \cap \neg F) \cup \neg T = (T \cap T) \cup F = T \cup F = T. \]

\[ v \models ((a \cap \neg b) \cup \neg c). \]
Formula: \[ A = ((a \cap \neg b) \cup \neg c). \]

Consider now \( v_1 : VAR \rightarrow \{T, F\} \) such that
\[
\begin{align*}
  v_1(a) &= T, \\
v_1(b) &= F, \\
v_1(c) &= T, \\
v_1(x) &= F, & \text{for all } x \in VAR - \{a, b, c\},
\end{align*}
\]

Observe: \( v(a) = v_1(a), v(b) = v_1(b), v(c) = v_1(c) \), so we get
\[
v_1 \models ((a \cap \neg b) \cup \neg c).\]
Consider $v_2 : VAR \rightarrow \{T, F\}$ such that
\[ v_2(a) = T, v_2(b) = F, v_2(c) = T, v_2(d) = T, \]
and
\[ v_2(x) = F, \quad \text{for all } x \in VAR - \{a, b, c, d\}, \]

Observe: $v(a) = v_2(a), v(b) = v_2(b), v(c) = v_2(c)$, so we get
\[ v_2 \models ((a \cap \neg b) \cup \neg c). \]
We are going to prove that there are as many of such truth assignments as real numbers! but they are all the same as the first \( v \) with respect to the formula \( A \).

When we ask a question: "How many truth assignments satisfy/fasify a formula \( A \)?" we mean to find all assignment that are different on the formula \( A \), not just different on a set \( \text{VAR} \) of all variables, as all of our \( v_1, v_2 \)’s were.

To address and to answer this question formally we first introduce some notations and definitions.
Notation: for any formula $A$, we denote by
\[
VAR_A
\]
a set of all variables that appear in $A$.

Definition: Given a formula $A \in \mathcal{F}$, any function
\[
w : VAR_A \longrightarrow \{T, F\}
\]
is called a truth assignment restricted to $A$. 
Example

\[ A = ((a \cap \neg b) \cup \neg c) \]

\[ \text{VAR}_A = \{a, b, c\} \]

Truth assignment restricted to \( A \) is any function:

\[ w : \{a, b, c\} \rightarrow \{T, F\} \].

We use the following theorem to count all possible truth assignment restricted to \( A \).

Counting Functions Theorem (1) For any finite sets \( A \) and \( A \), if \( A \) has \( n \) elements and \( B \) has \( m \) elements, then there are \( m^n \) possible functions that map \( A \) into \( B \).

There are \( 2^3 = 8 \) truth assignment restricted to \( A = ((a \Rightarrow \neg b) \cup \neg c) \).
**General case** For any $A$ there are

$$2^{|VAR_A|}$$

possible truth assignments $w$ restricted to $A$. 
All \( w \) restricted to \( A \) are listed in the table below.

\[
A = ((a \cap \neg b) \cup \neg c)
\]

<table>
<thead>
<tr>
<th>( w )</th>
<th>( a )</th>
<th>( b )</th>
<th>( c )</th>
<th>( w^*(A) ) computation</th>
<th>( w^*(A) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( w_1 )</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>(( T \Rightarrow T )) \cup \neg T \equiv T \cup F \equiv T</td>
<td>T</td>
</tr>
<tr>
<td>( w_2 )</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>(( T \Rightarrow T )) \cup \neg F \equiv T \cup T \equiv T</td>
<td>T</td>
</tr>
<tr>
<td>( w_3 )</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>(( T \Rightarrow F )) \cup \neg F \equiv F \cup T \equiv T</td>
<td>T</td>
</tr>
<tr>
<td>( w_4 )</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>(( F \Rightarrow F )) \cup \neg T \equiv T \cup F \equiv T</td>
<td>T</td>
</tr>
<tr>
<td>( w_5 )</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>(( F \Rightarrow T )) \cup \neg T \equiv T \cup F \equiv T</td>
<td>T</td>
</tr>
<tr>
<td>( w_6 )</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>(( F \Rightarrow T )) \cup \neg F \equiv T \cup T \equiv T</td>
<td>T</td>
</tr>
<tr>
<td>( w_7 )</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>(( T \Rightarrow F )) \cup \neg T \equiv F \cup F \equiv F</td>
<td>F</td>
</tr>
<tr>
<td>( w_8 )</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>(( F \Rightarrow F )) \cup \neg F \equiv T \cup T \equiv T</td>
<td>T</td>
</tr>
</tbody>
</table>

**Model** for \( A \) is a \( v \) such that

\[
 v \models A.
\]

\( w_1, w_2, w_3, w_4, w_5, w_6, w_8 \) are **models** for \( A \).

**Counter- Model** for \( A \) is a \( v \) such that

\[
 v \not\models A.
\]

\( w_7 \) is a **counter- model** for \( A \).
Tautology:

A is a tautology iff any \( v \) is a model for \( A \), i.e.

\[
\forall v (v \models A).
\]

Not a tautology:

A is not a tautology iff there is \( v : VAR \rightarrow \{T, F\} \), such that \( v \) is a counter-model for \( A \), i.e.

\[
\exists v (v \not\models A).
\]

Tautology Notation \( \models A \)

Example

\( \not\models ((a \cap \neg b) \cup \neg c) \)

because the truth assignment \( w_7 \) is a counter-model for \( A \).
Tautology Verification

**Truth Table Method:** list and evaluate all possible truth assignments restricted to $A$.

**Example:** $(a \Rightarrow (a \cup b))$.

<table>
<thead>
<tr>
<th>$v$</th>
<th>$a$</th>
<th>$b$</th>
<th>$v^*(A)$ computation</th>
<th>$v^*(A)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$</td>
<td>T</td>
<td>T</td>
<td>$(T \Rightarrow (T \cup T)) = (T \Rightarrow T) = T$</td>
<td>T</td>
</tr>
<tr>
<td>$v_2$</td>
<td>T</td>
<td>F</td>
<td>$(T \Rightarrow (T \cup F)) = (T \Rightarrow T) = T$</td>
<td>T</td>
</tr>
<tr>
<td>$v_3$</td>
<td>F</td>
<td>T</td>
<td>$(F \Rightarrow (F \cup T)) = (F \Rightarrow T) = T$</td>
<td>T</td>
</tr>
<tr>
<td>$v_4$</td>
<td>F</td>
<td>F</td>
<td>$(F \Rightarrow (F \cup F)) = (F \Rightarrow F) = T$</td>
<td>T</td>
</tr>
</tbody>
</table>

for all $v : VAR \rightarrow \{T, F\}$, $v \models A$, i.e.

\[ \models (a \Rightarrow (a \cup b)). \]
Proof by Contradiction Method

One works backwards, trying to find a truth assignment \( v \) which makes a formula \( A \) false.

If we find one, it means that \( A \) is not a tautology,

if we prove that it is impossible,

it means that the formula is a tautology.

Example \( A = (a \Rightarrow (a \cup b)) \)

Step 1 Assume that \( \not\models A \), i.e. \( A = F \).
Step 2  Analyze Strep 1:

\[(a \Rightarrow (a \cup b)) = F \iff a = T \text{ and } a \cup b = F.\]

Step 3  Analyze Step 2:

\[a = T \text{ and } a \cup b = F, \text{ i.e. } T \cup b = F.\]

This is impossible by the definition of \(\cup\).

Conclusion:

\[\models (a \Rightarrow (a \cup b)).\]

Observe that exactly the same reasoning proves that for any formulas \(A, B \in \mathcal{F},\)

\[\models (A \Rightarrow (A \cup B)).\]
Observe that the following formulas are tautologies

\[ (((a \Rightarrow b) \cap \neg c) \Rightarrow (((a \Rightarrow b) \cap \neg c) \cup \neg d)), \]

\[ (((a \Rightarrow b) \cap \neg C') \cup d) \cap \neg e) \Rightarrow \]

\[ (((a \Rightarrow b) \cap \neg C') \cup d) \cap \neg e) \cup ((a \Rightarrow \neg e)) \]

because they are of the form

\[ (A \Rightarrow (A \cup B)). \]
Tautologies, Contradictions

\[ T = \{ A \in \mathcal{F} : \models A \}, \]
\[ C = \{ A \in \mathcal{F} : \forall v (v \not\models A) \}. \]
Theorem 1 For any formula $A \in \mathcal{F}$ the following conditions are equivalent.

(1) $A$ is a tautology

(2) $A \in T$

(3) $\neg A$ is a contradiction

(4) $\neg A \in C$

(5) $\forall v \ (v^*(A) = T)$

(6) $\forall v \ (v \models A)$

(7) Every $v$ is a model for $A$
**Theorem 2** For any formula $A \in \mathcal{F}$ the following conditions are equivalent.

1. $A$ is a contradiction
2. $A \in C$
3. $\neg A$ is a tautology
4. $\neg A \in T$
5. $\forall v \ (v^*(A) = F)$
6. $\forall v \ (v \not\models A)$
7. $A$ does not have a model.