CHAPTER 5
Some Extensional Many Valued Semantics
CHAPTER 5

PART 1: Some Three Valued Extensional Semantics
PART 2: Many Valued Extensional Semantics
CHAPTER 5

PART 1: Some Three Valued Extensional Semantics
First Many Valued Logics

**First many valued logic** (defined semantically only) was formulated by Łukasiewicz in 1920.

We present here some of the first 3-valued extensional semantics, historically called 3-valued logics.

They are named after their authors: Łukasiewicz, Kleene, Heyting, and Bochvar.

**We assume** that the language of all logics considered except of Bochvar logic is

\[ \mathcal{L} = \mathcal{L}\{\neg, \cup, \cap, \Rightarrow\} \]
3-Valued Semantics

We add one extra logical value ⊥ to the classical set of two values \{T, F\} to be able to express that the logical value of a statement \(A\) may now be not only true or false.

The third logical value denotes a notion of "unknown", "uncertain", "undefined", or even can express that "we don’t have a complete information about \(A\)", depending on the context and motivation for the logic. ⊥ is the most frequently used symbol for different concepts of unknown.
3 Valued Semantics Assumptions

All three valued logics considered here, when defined semantically, enlist a third logical value which we denote by \( \bot \), or \( m \) in case of Bochvar semantics.

We assume that the third value is intermediate between truth and falsity, i.e. the set of logical values is ordered and we have the following:

Assumption 1

\[
F < \bot < T, \quad \text{or} \quad F < m < T
\]

Assumption 2

In all of presented here semantics we take \( T \) as designated value, i.e. \( T \) is the value that defines the notion of satisfiability and tautology.
Many Valued Semantics Assumptions

The third value $\perp$ corresponds also to some notion of incomplete information, inconsistent information, or to a notion of being undefined, or unknown.

Historically all these semantics, and many others, were and still are called logics.

We also will use the name logic for them, instead saying each time "logic defined semantically", or "semantics for a given logic".
Many Valued Extensional Semantics

Reminder: we assumed that in all cases, except of Bochvar logic the language is

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

Formal definition of many valued extensional semantics follows the pattern of the classical case and consists of giving definitions of the following main components:

1. Logical Connectives
2. Truth Assignment
3. Satisfaction Relation, Model, Counter-Model
4. Tautology

We define all the steps in case of Łukasiewicz’ s semantics (logic) to establish a pattern and proper notation and leave in case of other logics as an exercise for the reader.
Łukasiewicz Logic Ł

Motivation
Łukasiewicz developed his semantics (called logic) to deal with future contingent statements. Contingent statements are not just neither true nor false but are indeterminate in some metaphysical sense. It is not only that we do not know their truth value but rather that they do not possess one.
The Language:

\[ L = L\{\neg, \land, \lor, \Rightarrow\} \]

Observe that the language is the same as in the classical case.
The set \( \mathcal{F} \) of formulas is defined in a standard way.
Ł Semantics: Connectives

Step 1 of Ł semantics definition
Remember that we assumed:  \( F < \bot < T \)

Ł Negation \( \neg \) is a function:

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

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

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

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

such that \( a \cap b = \text{min}\{a, b\} \)
Ł Semantics: Connectives

Remember that we assumed: \( F < \bot < T \)

Ł **Disjunction** \( \cup \) is a function:

\[
\cup : \{T, \bot, F\} \times \{T, \bot, F\} \rightarrow \{T, \bot, F\}
\]
such that \( a \cup b = \max\{a, b\} \)

Ł **Implication** \( \Rightarrow \) is a function:

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

\[
a \Rightarrow b = \begin{cases} 
\neg a \cup b & \text{if } a > b \\
T & \text{otherwise}
\end{cases}
\]
Ł Connectives Truth Tables

Ł Negation

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

Ł Conjunction

<table>
<thead>
<tr>
<th>∩</th>
<th>F</th>
<th>⊥</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>⊥</td>
<td>F</td>
<td>⊥</td>
<td>⊥</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>⊥</td>
<td>T</td>
</tr>
</tbody>
</table>
Ł Connectives Truth Tables

Ł Disjunction

<table>
<thead>
<tr>
<th>$\lor$</th>
<th>F</th>
<th>$\bot$</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>$\bot$</td>
<td>T</td>
</tr>
<tr>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

Ł Implication

<table>
<thead>
<tr>
<th>$\Rightarrow$</th>
<th>F</th>
<th>$\bot$</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>$\bot$</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>$\bot$</td>
<td>$\bot$</td>
<td>$\bot$</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>$\bot$</td>
<td>T</td>
</tr>
</tbody>
</table>
Ł - Semantics: Truth Assignment

Step 2 of Ł -semantics definition

Definition

A truth assignment is now any function

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

Observe that the domain of truth assignment is the set of propositional variables, i.e. the truth assignment is defined only for atomic formulas.
Truth Assignment Extension $v^*$ to $\mathcal{F}$

**Definition**

Given a truth assignment $v : \text{VAR} \rightarrow \{T, \bot, F\}$

We define its extension $v^* : \mathcal{F} \rightarrow \{T, \bot, F\}$ by the induction on the degree of formulas as follows

(i) for any $a \in \text{VAR}$, $v^*(a) = v(a)$;

(ii) and for any $A, B \in \mathcal{F}$ we put

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

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

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

$v^*((A \Rightarrow B)) = v^*(A) \Rightarrow v^*(B)$
Ł Semantics: Satisfaction Relation

Step 3 of Ł-semantics definition

Definition

Let \( v : \text{VAR} \rightarrow \{ T, \perp, F \} \)

We say that a truth assignment \( v \) Ł-satisfies a formula \( A \in \mathcal{F} \) iff \( v^*(A) = T \)

Notation: \( v \models _Ł A \)

Definition

We say that a truth assignment \( v \) does not Ł-satisfy a formula \( A \in \mathcal{F} \) iff \( v^*(A) \neq T \)

Notation: \( v \not\models _Ł A \)
Ł - Semantics: Model, Counter Model

Ł - Model
Any truth assignment \( v \), \( v : \text{VAR} \rightarrow \{ F, \perp, T \} \) such that

\[ v \models_{Ł} A \]

is called a \( Ł \) - model for \( A \)

Ł - Counter Model
Any \( v \) such that

\[ v \not\models_{Ł} A \]

is called a \( Ł \) - counter model for the formula \( A \)
Ł - Semantics: Tautology

Step 4 of Ł-semantics definition

Definition
For any $A \in \mathcal{F}$,
$A$ is a Ł tautology if and only if $v^*(A) = T$ for all $v : \text{VAR} \rightarrow \{F, \bot, T\}$
We also say that
$A$ is a Ł tautology if and only if all truth assignments $v : \text{VAR} \rightarrow \{F, \bot, T\}$ are Ł models for $A$

Notation
$$\models_{Ł} A$$

Set of all Ł tautologies
$$\mathcal{LT} = \{A \in \mathcal{F} : \models_{Ł} A\}$$
Ł Tautologies

Let $LT$, $T$ denote the sets of all Ł tautologies and the classical tautologies, respectively.

Q1 Is the Ł logic (defined semantically!) really different from the classical logic?

It means are theirs sets of tautologies different?

Answer: YES, they are different sets.

Consider a classical tautology $\neg a \cup a$, i.e. we know that

$$\models (\neg a \cup a)$$

We will show that

$$\not\models_L (\neg a \cup a)$$
Classical and Ł Tautologies

Consider the formula \((\neg a \cup a)\)

Take a truth assignment \(v\) such that

\[v(a) = \bot\]

Evaluate

\[v^\star(\neg a \cup a) = v^\star(\neg a) \cup v^\star(a) = \neg v(a) \cup v(a)\]

\[= \neg \bot \cup \bot = \bot \cup \bot = \bot\]

This proves that \(v\) is a counter-model for \((\neg a \cup a)\)

\[\not \models_L (\neg a \cup a)\]

and we have a property:

\[\text{LT} \neq \text{T}\]
Classical and Ł Tautologies

Q2 Do the Ł logic and classical logic have something more in common besides the common language?
Do they share some tautologies?
Which is the relationship (if any) between their sets of tautologies LT and T?

Answer
YES, they do share tautologies and
YES, they do have an interesting relationship
Classical and Ł Tautologies

Let’s restrict the Truth Tables for Ł connectives to the values $T$ and $F$ only.

Observe that by doing so we get the Truth Tables for classical connectives, i.e. the following holds for any $A \in \mathcal{F}$

If $v^*(A) = T$ for all $v : \text{VAR} \rightarrow \{F, \bot, T\}$,
then $v^*(A) = T$ for all $v : \text{VAR} \rightarrow \{F, T\}$

We have hence proved that

Fact

$$\text{LT} \subset T$$
Łukasiewicz Life, Works and Logics

Jan Leopold Łukasiewicz was born on 21 December 1878 in Lwow, historically a Polish city, at that time the capital of Austrian Galicia.

He died on 13 February 1956 in Ireland and is buried in Glasnevin Cemetery in Dublin, ”far from dear Lwow and Poland”, as his gravestone reads.

Here is a very good, interesting and extended entry in Stanford Encyclopedia of Philosophy about his life, influences, achievements, and logics:

Motivation
We model now a situation where the third logical value ⊥ intuitively represents the notion of "undecided", or "state of partial ignorance"
A sentence is assigned a value ⊥ just in case it is not known to be either true or false
For example imagine a detective trying to solve a murder. He may conjecture that Jones killed the victim. He cannot, at present, assign a truth value T or F to his conjecture, so we assign the value ⊥
But it is certainly either true of false and ⊥ represents our ignorance rather then total unknown.
The K-Language is the same in case of classical propositional and \( L \) logic, i.e.

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

We form the set \( \mathcal{F} \) of formulas in a standard way
K- Semantics: Connectives

Connectives \( \neg, \cup, \cap \) of \( K \) are defined as in \( \mathcal{L} \) semantics, i.e.

\[ \neg \bot = \bot, \quad \neg F = T, \quad \neg T = F \]

and for any \( (a, b) \in \{T, \bot, F\} \times \{T, \bot, F\} \), we put

\[ a \cup b = \max\{a, b\} \]

\[ a \cap b = \min\{a, b\} \]

Remember that we assumed: \( F < \bot < T \)
K- Semantics: Connectives

Implication
For any \((a, b) \in \{T, \bot, F\} \times \{T, \bot, F\}\) we put

\[ a \Rightarrow b = \neg a \cup b \]

Kleene’s 3-valued truth tables differ hence from Łukasiewicz’s truth tables only in a case of implication. This table is:

K-Implication

<table>
<thead>
<tr>
<th>(\Rightarrow)</th>
<th>F</th>
<th>(\bot)</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>(\bot)</td>
<td>T</td>
<td>(\bot)</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>(\bot)</td>
<td>T</td>
</tr>
</tbody>
</table>
K- Semantics: Tautologies

K Tautologies

\[ KT = \{ A \in \mathcal{F} : \models_K A \} \]

Relationship between \( \mathcal{L} \), \( K \), and classical logic.

\[ LT \neq KT, \quad KT \subset T \]

Proof of \( LT \neq KT \).

Obviously \( \models_L (a \Rightarrow a) \)

Take \( v \) such that \( v(a) = \bot \); we have that for \( K \) semantics
\[ v^*(a \Rightarrow a) = (v(a) \Rightarrow v(a)) = (\bot \Rightarrow \bot) = \bot \]

This proves that \( \not\models_K (a \Rightarrow a) \)

and

\[ LT \neq KT \]
K- Tautologies

The second sets of tautologies property

\[KT \subset T\]

follows directly from the fact that, as in the Ł case, if we restrict the K Truth Tables to the values \(T\) and \(F\) only, we get the Truth Tables for classical connectives.
Motivation and History
We call the $H$ logic also a Heyting logic because its connectives are defined as operations on the set $\{F, \bot, T\}$ in such a way that they form a 3-element pseudo-Boolean algebra which is also often called a 3-element Heyting algebra.

Pseudo-Boolean algebras were invented and developed as the first ever semantics for the Intuitionistic Logic.
Motivation and History

The Intuitionistic Logic was defined by its inventor Brouwer and his school in 1900s as a proof system only. Heyting provided first axiomatization for the Intuitionistic Logic, so the pseudo-Boolean algebras are often also called Heyting algebras in his honor. The pseudo-Boolean algebras semantics was discovered some 35 years later by McKinsey and Tarski in 1942 for Intuitionistic propositional logic only. It took yet another 15 years to extend it to predicate Intuitionistic logic by Rasiowa, Mostowski in 1957.
A formula $A$ is an **Intuitionistic tautology** if and only if it is true in all **pseudo-Boolean algebras**. Hence, if $A$ is an **Intuitionistic tautology**, it is also a tautology under the **3-valued Heyting semantics**. If $A$ is not a 3-valued Heyting tautology, then it is not an Intuitionistic tautology. It means that our 3-valued **Heyting semantics** is a good candidate for a **counter model** for the formulas that might not be Intuitionistic tautologies.
H Logic and Intuitionistic Logic

Denote by $\text{IT}$, $\text{HT}$ the sets of all tautologies of the Intuitionistic logic and Heyting 3-valued logic, respectively.

We have that

$$\text{IT} \subset \text{HT}$$

We conclude that for any formula $A$,

$$\text{If } \not\models_H A \text{ then } \not\models_I A$$

It means that if we show that a formula $A$ has a Heying 3-valued counter-[\text{model}], then we have proved that \text{it is not} an intuitionistic tautology.
Kripke Models

The other type of semantics for the Intuitionistic Logic were defined by Kripke in 1964. They are called Kripke Models. Kripke Models were proved to be equivalent to the pseudo-Boolean algebras models in case of the Intuitionistic Logic. Kripke Models are very general and serve as a general method of defining not extensional semantics for various classes of logics. That includes semantics for hundreds of Modal, Knowledge Logic and different logics developed and being developed by computer scientists.
H Semantics

The Language:

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

Logical connectives: \(\cup\) and \(\cap\) are the same as in the case of \(\mathcal{L}\) and \(\mathcal{K}\) semantics, i.e.

for any \((a, b) \in \{T, \bot, F\} \times \{T, \bot, F\}\) we put

\[ a \cup b = \max\{a, b\}, \quad a \cap b = \min\{a, b\} \]

Remember that we assumed: \(F < \bot < T\)
Heyting Semantics

Implication

For any \((a, b) \in \{T, \bot, F\} \times \{T, \bot, F\}\) we put

\[
a \Rightarrow b = \begin{cases} 
T & \text{if } a \leq b \\
b & \text{otherwise}
\end{cases}
\]

Negation

\[\neg a = a \Rightarrow F.\]
H Truth Tables

H Implication

$$\Rightarrow \begin{array}{c|ccc}
F & \bot & T \\
F & T & T & T \\
\bot & F & T & T \\
T & F & \bot & T \\
\end{array}$$

H Negation

$$\neg \begin{array}{c|ccc}
F & \bot & T \\
T & F & F \\
\end{array}$$
Sets of Tautologies Relationships

Notation: \( HT, T, LT, KT \) denote the set of all tautologies of the \( H \), classical, \( \mathcal{L} \), and \( K \) logic, respectively.

Relationships:

\[
HT \not\subset T \not\subset LT \not\subset KT,
\]

\[
HT \subset T
\]

Proof of \( HT \not\subset T \)

For the formula \( (\neg a \cup a) \) we have:

\[
\models (\neg a \cup a) \quad \text{and} \quad \not\models_H (\neg a \cup a)
\]
Sets of Tautologies Relationships

**Proof** of \( HT \neq KT \)

Take any truth assignment \( v \), such that \( v(a) = \perp \)

We get

\[ \models_H (A \Rightarrow A) \]

but

\[ \nvdash_K (A \Rightarrow A) \]
Sets of Tautologies Relationships

**Proof of HT ≠ LT**

Take now a variable assignment \( v \) such that \( v(a) = v(b) = \bot \)

It proves that

\[ \not\models_K (\neg (a \cap b) \Rightarrow (\neg a \cup \neg b)) \]

but we verify that

\[ \models_L (\neg (a \cap b) \Rightarrow (\neg a \cup \neg b)) \]
Sets of Tautologies Relationships

Proof of $\text{HT} \subset \text{T}$

Observe now that if we restrict the truth tables for $\text{H}$ to connectives $\text{T}$ and $\text{F}$ only, we get the truth tables for classical connectives. All together we have proved that the classical logic extends all of our three-valued logics $\text{L}$, $\text{K}$ and $\text{H}$, i.e.

$$\text{LT} \subset \text{T}, \quad \text{KT} \subset \text{T}, \quad \text{HT} \subset \text{T}$$
Bochvar 3-valued logic B

Motivation
Consider a semantic paradox given by a sentence: this sentence is false.
If it is true it must be false,
if it is false it must be true.
According to Bochvar, such sentences are neither true of false but rather paradoxical or meaningless.
Bochvar 3-valued logic B

Bochvar’s semantics follows the principle that the third logical value, denoted now by \( m \) (for mining less) is in some sense ”infectious”;
if one component of the formula is assigned the value \( m \) then the formula is also assigned the value \( m \).
Bochvar also adds an one assertion operator \( S \) that asserts the logical value of \( T \) and \( F \), i.e.

\[
SF = F, \quad SF = F
\]

and it asserts that meaningfulness \( m \) is false, i.e

\[
Sm = F
\]
B Language

Language: we add a new one argument connective $S$ and get

$$L_B = L_{\{\neg, S, \Rightarrow, \cup, \cap\}}$$

We denote by $F_B$ the set of all formulas of the language $L_B$ and by $F$ the set of formulas of the language $L_{\{\neg, \Rightarrow, \cup, \cap\}}$ common to the classical and all 3 valued logics considered till now.

Observe that directly from the definition we have that

$$F \subset F_B$$

The formula $SA$ reads "assert A"
### B Logical Connectives

#### B Negation

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

#### B Conjunction

<table>
<thead>
<tr>
<th>∩</th>
<th>F</th>
<th>m</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>m</td>
<td>F</td>
</tr>
<tr>
<td>m</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>m</td>
<td>T</td>
</tr>
</tbody>
</table>
# B Semantics

## B Disjunction

<table>
<thead>
<tr>
<th>$\lor$</th>
<th>F</th>
<th>m</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>m</td>
<td>T</td>
</tr>
<tr>
<td>m</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>m</td>
<td>T</td>
</tr>
</tbody>
</table>

## B Implication

<table>
<thead>
<tr>
<th>$\Rightarrow$</th>
<th>F</th>
<th>m</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>T</td>
<td>m</td>
<td>T</td>
</tr>
<tr>
<td>m</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>m</td>
<td>T</td>
</tr>
</tbody>
</table>

## B Assertion

<table>
<thead>
<tr>
<th>$S$</th>
<th>F</th>
<th>m</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

B Tautologies

\[ \text{BT} = \{ A \in \mathcal{F}_B : \models_B A \} \]

Let \( A \) be a formula that do not contain the assertion operator \( S \), i.e. the formula \( A \in \mathcal{F} \) of the language \( \mathcal{L}_{\{\neg, \Rightarrow, \cup, \cap\}} \).

Observe that any \( v \), such that \( v(a) = m \) for at least one variable in the formula \( A \in \mathcal{F} \) is a counter-model for that formula.

So we have that

\[ \mathcal{T} \cap \text{BT} = \emptyset \]

For a formula \( A \in \mathcal{F}_B \) to be a B tautology, it must contain the connective \( S \).
CHAPTER 5

PART 2: Many Valued Extensional Semantics M
Many Valued Extensional Semantics $M$

Here is a straightforward generalization of classical and 3 valued semantics presented here to a semantics $M$ defined for any propositional language.

The semantics $M$ defined here is extensional and is defined for a non-empty set of $V$ of logical values of any cardinality. We only assume that the set $V$ of logical values of $M$ always has a special, distinguished logical value which serves to define a notion of tautology.

We denote this distinguished value as $T$. 
Many Valued Extensional Semantics $\mathbf{M}$

Given a propositional language $L_{CON}$ for the set $CON = C_1 \cup C_2$, where $C_1$ is the set of all unary connectives, and $C_2$ is the set of all binary connectives.

Formal definition of many valued extensional semantics $\mathbf{M}$ for the language $L_{CON}$ follows the pattern of the classical and 3-valued cases and consists of giving definitions of the following main components:

1. Logical Connectives under semantics $\mathbf{M}$
2. Truth Assignment for $\mathbf{M}$
3. Satisfaction Relation, Model, Counter-Model under semantics $\mathbf{M}$
4. Tautology under semantics $\mathbf{M}$
Definition of $\mathbf{M}$ - Extensional Connectives

Given a propositional language $\mathcal{L}_{CON}$ for the set $CON = C_1 \cup C_2$, where $C_1$ is the set of all unary connectives, and $C_2$ is the set of all binary connectives. Let $V$ be a non-empty set of logical values adopted by the semantics $\mathbf{M}$. We adopt now a following formal definition of $\mathbf{M}$ - extensional connectives.

Definition

Connectives $\triangledown \in C_1, \circ \in C_2$ are called $\mathbf{M}$ -extensional iff their semantics $\mathbf{M}$ is defined by respective functions:

$$\triangledown : V \rightarrow V \quad \text{and} \quad \circ : V \times V \rightarrow V$$
Definability of Connectives under a semantics $M$

Given a propositional language $\mathcal{L}_{CON}$ and its extensional semantics $M$

We adopt the following definition

Definition

A connective $\circ \in CON$ is **definable** in terms of some connectives $\circ_1, \circ_2, ... \circ_n \in CON$ for $n \geq 1$ under the semantics $M$ if and only if the connective $\circ$ is a certain function composition of functions $\circ_1, \circ_2, ... \circ_n$ as they are defined by the semantics $M$

Example

Classical implication $\Rightarrow$ is **definable** in terms of $\cup$ and $\neg$ under classical semantics because under this semantics $\Rightarrow$ is a composition of functions $\neg$ and $\cup$ defined as follows

For all $(a, b) \in \{T, F\} \times \{T, F\}$,

$$a \Rightarrow b = \neg a \cup b$$
Definability of Connectives

Exercise 1
Verify which (if any) of our 3 valued semantics $L, K, H$ are definable in terms of $\cup$ and $\neg$ by the classical case composition formula.

Exercise 2
Verify which (if any) of our 3 valued semantics $L, K, H$ are definable in terms of $\Rightarrow$ and $\neg$ by the classical case composition formula.

Exercise 3
Verify which of our 3 valued semantics $L, K, H$ is definable in terms of $\Rightarrow$ alone.
M Semantics: Truth Assignment

M Semantics Assumption
We assume that the set $V$ of logical values of M always has a special, distinguished logical value which serves to define a notion of tautology under the semantics $M$.
We denote this distinguished value as $T$.

Step 2
We define M semantics, as in previous cases, in terms of the propositional connectives as defined in the Step 1 and a function called M truth assignment.

Definition
M truth assignment is any function

$$v : VAR \rightarrow V$$

where $V$ is the set of logical values of $M$. 
**Truth Assignment Extension** $v^*$ to $\mathcal{F}$

**Definition**

Given the **$M$** truth assignment

\[ v : \text{VAR} \rightarrow V \]

We define its **$M$ extension** $v^*$ to the set $\mathcal{F}$ of all formulas of $\mathcal{L}$ as any function

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

such that the following conditions are satisfied

(i) for any $a \in \text{VAR}$

\[ v^*(a) = v(a); \]
M Truth Assignment Extension $\nu^*$ to $\mathcal{F}$

(ii) For any connectives $\nabla \in C_1$, $\circ \in C_2$ and for any formulas $A, B \in \mathcal{F}$ we put

$$\nu^*(\nabla A) = \nabla \nu^*(A)$$

$$\nu^*((A \circ B)) = \circ(\nu^*(A), \nu^*(B))$$

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 semantical meaning as defined by the semantics $M$. 
Step 3

Definition: Let \( v : \text{VAR} \rightarrow V \)

Let \( T \in V \) be the distinguished logical value.

We say that

\[ v \models_M A \iff v^*(A) = T \]

Notation: \( v \models_M A \)

Definition: We say that

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

Notation: \( v \not\models_M A \)

The relation \( \models_M \) is called a satisfaction relation under semantics \( M \), or \( x \) \( M \) satisfaction relation for short.
**M Semantics: Model, Counter-Model**

**Definition:**
Given a formula $A \in \mathcal{F}$ and $v : \text{VAR} \rightarrow V$

Any $v$ such that $v \models_M A$ is called a **M model** for $A$

Any $v$ such that $v \not\models_M A$ is called a **M counter model** for $A$
Step 4

Definition:
For any formula \( A \in \mathcal{F} \)

\( A \) is a **M tautology** iff \( v^*(A) = T \), for all \( v : \text{VAR} \rightarrow V \)

i.e. we have that

\( A \) is a **M tautology** iff any \( v : \text{VAR} \rightarrow V \) is a **M model** for \( A \)

Notation

We write symbolically \( \models_M A \) for the statement "\( A \) is a **M** tautology"
Semantics: not a tautology

Definition

\( A \) is not a M tautology \iff \text{there is } v, \text{ such that } v^*(A) \neq T \)

i.e. we have that

\( A \) is not a M tautology \iff A \) has a M counter-model

Notation

We write \( \not \models_M A \) to denote the statement "A is not M tautology"
Challenge Exercise

1. **Define** your own propositional language $L_{CON}$ that contains also **different connectives** that the standard connectives $\neg, \cup, \cap, \Rightarrow$

   Your language $L_{CON}$ **does not need** to include all (if any!) of the standard connectives $\neg, \cup, \cap, \Rightarrow$

2. **Describe** intuitive meaning of the new connectives of your language

3. Give some **motivation** for your own semantic

4. **Define** formally your own extensional semantics $M$ for your language $L_{CON}$ - it means write carefully all **Steps 1-4** of the definition of your $M$
Chapter 5
Some Simple Review Problems
Exercise 1

**Reminder:** we define $H$ semantics operations $\cup$ and $\cap$ as follows.

For any $(a, b) \in \{T, \bot, F\} \times \{T, \bot, F\}$ we put

$$a \cup b = \max\{a, b\}, \quad a \cap b = \min\{a, b\}$$

**Implication:**

$$a \Rightarrow b = \begin{cases} T & \text{if } a \leq b \\ b & \text{otherwise} \end{cases}$$

**Negation:**

$$\neg a = a \Rightarrow F.$$
Exercise 1

Question  We know that

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

is such that

\[ v^*((a \cap b) \Rightarrow (a \Rightarrow c)) = \perp \]

under H semantics.

Evaluate

\[ v^*((((b \Rightarrow a) \Rightarrow (a \Rightarrow \neg c)) \cup (a \Rightarrow b))) \]
Exercise 1

Solution
$v^*((a \cap b) \Rightarrow (a \Rightarrow c)) = \bot$ under $H$ semantics
if and only if $(a \cap b) = T$ and $(a \Rightarrow c) = \bot$
if and only if $a = T, b = T$ and $(T \Rightarrow c) = \bot$
if and only if $c = \bot$.

I.e. we have that $v^*((a \cap b) \Rightarrow (a \Rightarrow c)) = \bot$
if and only if $a = T, b = T, c = \bot$
Exercise 1

Now we can evaluate
\[ v^*(((b \Rightarrow a) \Rightarrow (a \Rightarrow \neg c)) \cup (a \Rightarrow b)) \]
as follows
\[ v^*(((b \Rightarrow a) \Rightarrow (a \Rightarrow \neg c)) \cup (a \Rightarrow b)) \]
\[ = (((T \Rightarrow T) \Rightarrow (T \Rightarrow \neg \bot)) \cup (T \Rightarrow T)) \]
\[ = ((T \Rightarrow (T \Rightarrow F)) \cup T) \]
\[ = T \]
Exercise 2

We define a 4 valued $L_4$ logic semantics as follows. The language is $\mathcal{L}_{\{\neg, \Rightarrow, \cup, \cap\}}$.

We define the logical connectives $\neg, \Rightarrow, \cup, \cap$ as the following operations in the set

$\{F, \perp_1, \perp_2, T\}$, where $F < \perp_1 < \perp_2 < T$

**Negation**

$\neg : \{F, \perp_1, \perp_2, T\} \longrightarrow \{F, \perp_1, \perp_2, T\}$,

such that

$\neg \perp_1 = \perp_1, \neg \perp_2 = \perp_2, \neg F = T, \neg T = F$
Exercise 2

**Conjunction**

\[ \cap : \{F, \bot_1, \bot_2, T\} \times \{F, \bot_1, \bot_2, T\} \rightarrow \{F, \bot_1, \bot_2, , T\} \]

such that for any \((a, b) \in \{T, \bot_1, \bot_2, F\} \times \{T, \bot_1, \bot_2, F\}\) we put

\[ a \cap b = \text{min}\{a, b\} \]

**Disjunction**

\[ \cup : \{F, \bot_1, \bot_2, T\} \times \{F, \bot_1, \bot_2, T\} \rightarrow \{F, \bot_1, \bot_2, T\} \]

such that for any \((a, b) \in \{T, \bot_1, \bot_2, F\} \times \{T, \bot_1, \bot_2, F\}\) we put

\[ a \cup b = \text{max}\{a, b\} \]
Exercise 2

Implication

⇒ : \{F, \bot_1, \bot_2, T\} \times \{F, \bot_1, \bot_2, T\} \rightarrow \{F, \bot_1, \bot_2, T\}

such that for any \((a, b) \in \{T, \bot_1, \bot_2, F\} \times \{T, \bot_1, \bot_2, F\}\) we put

\[ a \Rightarrow b = \begin{cases} 
\neg a \cup b & \text{if } a > b \\
T & \text{otherwise}
\end{cases} \]

Verify whether

\[ \vdash_4 ((a \Rightarrow b) \Rightarrow (\neg a \cup b)) \]
Solution

Let \( v \) be a truth assignment such that \( v(a) = v(b) = \bot_1 \)

We evaluate

\[
\nu^*( (a \Rightarrow b) \Rightarrow (\neg a \cup b) ) = ((\bot_1 \Rightarrow \bot_1) \Rightarrow (\neg \bot_1 \cup \bot_1))
\]

\[
= (T \Rightarrow (\bot_1 \cup \bot_1)) = (T \Rightarrow \bot_1) = \bot_1.
\]

This proves that \( v \) is a counter-model for our formula and

\[
\not\models_4 ((a \Rightarrow b) \Rightarrow (\neg a \cup b))
\]
Exercise 2

Observe that a \( v \) such that
\[ v(a) = v(b) = \bot_2 \]
is also a counter model.

We evaluate (in shorthand notation)
\[ v^*((a \Rightarrow b) \Rightarrow (\neg a \cup b)) = ((\bot_2 \Rightarrow \bot_2) \Rightarrow (\neg \bot_2 \cup \bot_2)) = (T \Rightarrow (\bot_2 \cup \bot_2)) = (T \Rightarrow \bot_2) = \bot_2 \]