cse371/mat371
LOGIC

Professor Anita Wasilewska
Chapter 3
Propositional Semantics: Classical and Many Valued

Many Valued Semantics:
Łukasiewicz, Heyting, Kleene, Bohvar
First Many Valued Logics

The study of many valued logics in general and 3-valued logics in particular has its beginning in the work of a Polish mathematician Jan Leopold Łukasiewicz in 1920. Łukasiewicz was the first to define a 3-valued semantics for the language $$L\{\neg, \land, \lor, \Rightarrow\}$$ of classical logic, and called it a logic for short.

He left the problem of finding a proper axiomatic proof system for it open.
First Many Valued Logics

The other 3-valued semantics presented here were also first called logics and this terminology is still widely used.

Nevertheless, as these logics were defined only semantically, i.e. defined only by providing a semantics for their languages we call them semantics (for logics to be developed), not logics.
Creating a Logic

Existence of a proper axiomatic proof system for a given semantics and proving its completeness is always a next open question to be answered (when it is possible).

A process of creating a logic (based on a given language) is three fold: we have to define semantics, create axiomatic proof system and prove completeness theorem that establishes a relationship between semantics and proof system.
First Many Valued Logics

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 semantics (logics) considered here except of Bochvar semantics is

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

All three valued semantics considered here enlist a third logical value which we denote by $\bot$, or $m$ in case of Bochvar semantics.

The third logical value denotes a notion of unknown, uncertain, undefined, or even the notion of we don’t have a complete information about depending on the context and motivation for the semantics (logic).

The symbol $\bot$ is the most frequently used for different concepts of unknown.
Many Valued Semantics

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 will also use the name **logic** for them, instead saying each time ”logic defined semantically”, or ”semantics for a given logic”
3 Valued Semantics Assumptions

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

Assumption 1

\[ F \prec \perp \prec T, \quad \text{and} \quad F \prec m \prec T \]

Assumption 2

We take \( T \) as designated value, i.e. \( T \) is the value that defines the notions of satisfiability and tautology
Many Valued Extensional Semantics

Formal definition of all many valued semantics presented here follows the definition of the extensional semantics $M$ in general, and the pattern presented in detail for the classical semantics in particular.

It consists of giving definitions of the following main components:

Step 1: given the language $L$ we define a set of logical values and its distinguish value $T$ and define all extensional logical connectives of $L$.

Step 2: we define notions of a truth assignment and its extension.

Step 3: we define notions of satisfaction, model, counter model.

Step 4: we define notions tautology under the semantics $M$. 
Łukasiewicz Semantics

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.
Semantics: Language

We define all the steps in case of Łukasiewicz semantics (logic) to establish a pattern and proper notation and leave adopting all steps to the case of other semantics as an exercise.

**Step 1** The language is \( \mathcal{L}\{\neg, \cap, \cup, \Rightarrow\} \).

Observe that the language is the same as in the classical semantics case.

The set \( \mathcal{F} \) of formulas is defined in a standard way.
Step 1 Connectives

We assumed: \( F < \bot < T \) and we define the connectives as follows

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

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

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

**Conjunction** \( \land \) is a function

\[
\land : \{T, \bot, F\} \times \{T, \bot, F\} \longrightarrow \{T, \bot, F\}
\]

such that for any \((x, y) \in \{T, \bot, F\} \times \{T, \bot, F\}\), we put

\[
x \land y = \text{min}\{x, y\}
\]
**L Semantics: Connectives**

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

\[
\cup : \{T, \bot, F\} \times \{T, \bot, F\} \rightarrow \{T, \bot, F\}
\]
such that for any \((a, b) \in \{T, \bot, F\} \times \{T, \bot, F\}\), we put

\[
x \cup y = \text{max}\{x, y\}
\]

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

\[
\Rightarrow : \{T, \bot, F\} \times \{T, \bot, F\} \rightarrow \{T, \bot, F\}
\]
such that for any \((x, y) \in \{T, \bot, F\} \times \{T, \bot, F\}\), we put

\[
x \Rightarrow y = \begin{cases} 
\neg x \cup y & \text{if } x > y \\
T & \text{otherwise}
\end{cases}
\]
**L Connectives Truth Tables**

**Negation**

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

**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>
L Connectives Truth Tables

Disjunction

<table>
<thead>
<tr>
<th>$\cup$</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>
</tbody>
</table>

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>$\bot$</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>$\bot$</td>
<td>T</td>
</tr>
</tbody>
</table>
Semantics: Truth Assignment

Step 2  Truth assignment and its extension

Definition
A truth assignment is 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 $v^*$

**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

\[ v^*(\neg A) = \neg v^*(A); \]
\[ v^*((A \cap B)) = v^*(A) \cap v^*(B); \]
\[ v^*((A \cup B)) = v^*(A) \cup v^*(B); \]
\[ v^*((A \Rightarrow B)) = v^*(A) \Rightarrow v^*(B). \]
L Semantics: Satisfaction Relation

Step 3  Satisfaction, Model, Counter Model

Definition

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

We say that a truth assignment \( v \) \textbf{L satisfies} a formula \( A \in \mathcal{F} \) if and only if \( v^*(A) = T \)

Notation: \( v \models_L A \)

Definition

We say that a truth assignment \( v \) \textbf{does not L satisfy} a formula \( A \in \mathcal{F} \) if and only if \( v^*(A) \neq T \)

Notation: \( v \not\models_L A \)
L Semantics: Model, Counter Model

Model
Any truth assignment \( v : \text{VAR} \rightarrow \{F, \perp, T\} \) such that
\[
v \models_L A
\]
is called a L model for \( A \)

Counter Model
Any \( v \) such that
\[
v \not\models_L A
\]
is called a L counter model for the formula \( A \)
Step 4 Tautology

For any $A \in \mathcal{F}$,

$A$ is a **L tautology** if and only if $v^*(A) = T$ for all $v : \text{VAR} \rightarrow \{F, \bot, T\}$

We also say that

$A$ is a **L tautology** if and only if all truth assignments $v : \text{VAR} \rightarrow \{F, \bot, T\}$ are **L models** for $A$

**Notation**

$\models_L A$
We denote the set of all **L tautologies** by

\[
\mathbf{LT} = \{ A \in \mathcal{F} : \models_L A \}
\]

Let \( \mathbf{LT}, \ \mathbf{T} \) be the sets of all **L tautologies** and the **classical tautologies**, respectively.

**Q1** Is the \( \mathcal{L} \) logic (defined semantically!) really **different** from the **classical logic**?

It means are theirs sets of tautologies different?

**Answer:** **YES**, they are **different** sets.

We know that

\[
\models (\neg a \cup a)
\]

We will show that

\[
\not\models_L (\neg a \cup a)
\]
Classical and L Tautologies

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

Take a truth assignment \(v\) such that

\[ v(a) = \bot \]

Evaluate

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

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

This proves that \(v\) is a \textbf{counter-model} for \((\neg a \cup a)\), i.e.

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

and we proved

\[ LT \neq T \]
Classical and L Tautologies

Q2  Do the L and classical logics have something more in common besides the same language?
YES, they also share some tautologies

Q3  Is there relationship (if any) between their sets of tautologies LT and T?
YES, their sets of tautologies LT and T do have an interesting relationship
Classical and L Tautologies

Let’s restrict the functions defining L connectives (Truth Tables for L connectives) to the values $T$ and $F$.

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

We have hence proved that

$$\text{LT} \subset \text{T}$$
Exercise

Exercise

Use the fact that $v : VAR \rightarrow \{F, \bot, T\}$ is such that

$$v^*((a \cap b) \Rightarrow \neg b) = \bot$$

under L semantics to evaluate

$$v^*((b \Rightarrow \neg a) \Rightarrow (a \Rightarrow \neg b)) \cup (a \Rightarrow b))$$

Use shorthand notation.
**Exercise**

**Solution**

Observe that \(((a \cap b) \Rightarrow \neg b) = \bot\) in two cases

- **c1**: \((a \cap b) = \bot\) and \(\neg b = F\)
- **c12**: \((a \cap b) = T\) and \(\neg b = \bot\)

Consider **c1**

We have \(\neg b = F\), i.e. \(b = T\)

Hence \((a \cap T) = \bot\) if and only if \(a = \bot\)

We get that \(v\) is such that \(v(a) = \bot\) and \(v(b) = T\)
Exercise

We got from analyzing case \textbf{c1} that \( v \) is such that \( v(a) = \bot \) and \( v(b) = T \).

We evaluate \( v^*(((b \Rightarrow \neg a) \Rightarrow (a \Rightarrow \neg b)) \cup (a \Rightarrow b)) = (((T \Rightarrow \neg \bot) \Rightarrow (\bot \Rightarrow \neg T)) \cup (\bot \Rightarrow T)) = ((\bot \Rightarrow \bot) \cup T) = T \)

Consider \textbf{c2}

We have \( \neg b = \bot \), i.e. \( b = \bot \) and \( (a \cap \bot) = T \), what is impossible

Hence \( v \) from case \textbf{c1} is the only one and \( v^*(((b \Rightarrow \neg a) \Rightarrow (a \Rightarrow \neg b)) \cup (a \Rightarrow b)) = T \)
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.

Heyting Semantics $\textbf{H}$

Motivation and History

We discuss here the Heyting semantics $\textbf{H}$ because of its connection with intuitionistic logic.

The $\textbf{H}$ connectives are defined as operations on the set $\{F, \bot, T\}$ in such a way that they form a 3-element pseudo-Boolean algebra.

Pseudo-Boolean algebras were created by McKinsey and Tarski in 1948 to provide semantics for the intuitionistic logic.

Pseudo-Boolean algebras are often called Heyting algebras.
Motivation and History

The intuitionistic logic, was defined by its inventor Brouwer and his school in 1900s as a proof system only.

Heyting provided its first axiomatization which everybody accepted.

McKinsey and Tarski proved in 1942 the completeness of the Heyting axiomatization with respect to their pseudo Boolean algebras semantics.

The pseudo boolean algebras are also called Heyting algebras in his honor and so is our semantics $H$. 
Motivation and History

A formula $A$ is an intuitionistic tautology if and only if it is true in all pseudo boolean algebras.

We prove that the operations defined by $H$ connectives form a 3-element pseudo boolean algebra. 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 the 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 (semantics), respectively.

We have that

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

We conclude that for any formula $A$,

If $\not\models_\text{H} A$ then $\not\models_\text{I} A$

It means that if we show that a formula $A$ has an H counter model, then we have proved that $A$ 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.

The Kripke models were later proved to be equivalent to the pseudo boolean algebras models in case of the intuitionistic logic.

Kripke models also provide a general method of defining semantics for many classes of logics.

That includes semantics for various modal logics and new logics developed and being developed by computer scientists.
H Semantics

Language

$L\{\neg, \Rightarrow, \cup, \cap\}$

Connectives

$\cup$ and $\cap$ are the same as in the case of Ł semantics, i.e. for any $(x, y) \in \{T, \perp, F\} \times \{T, \perp, F\}$ we put

$$x \cup y = \max\{x, y\}, \quad x \cap y = \min\{x, y\}$$

where $F < \perp < T$
Implication

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

such that for any \((x, y) \in \{ T, \bot, F \} \times \{ T, \bot, F \}\) we put

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

Negation

\[ \neg x = x \Rightarrow F \]
# Truth Tables

## Implication

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

## Negation

<table>
<thead>
<tr>
<th></th>
<th>F</th>
<th>⊥</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>⊥</td>
<td>F</td>
<td>⊥</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
Sets of Tautologies Relationships

HT, T, LT denote the set of all tautologies of the H, classical, and L semantics, respectively.

Relationships

\[ HT \neq T \neq LT \]

\[ HT \subset T \]

Proof of \( HT \neq T \)

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

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

**Proof of** $\mathbf{HT} \neq \mathbf{LT}$

Take a truth assignment $\nu$ such that

$$\nu(a) = \nu(b) = \bot$$

We verify that

$$\not\models_H \left( \neg(a \cap b) \Rightarrow (\neg a \cup \neg b) \right)$$

and

$$\models_L \left( \neg(a \cap b) \Rightarrow (\neg a \cup \neg b) \right)$$
Proof of $HT \subset T$

Observe that if we restrict the truth tables for $H$ connectives to logical values $T$ and $F$ only we get the truth tables for the classical connectives, i.e. and the following holds for any formula $A$

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

All together we have proved that the classical semantics extends both $L$ and $H$ semantics, i.e.

$LT \subset T$ and $HT \subset T$
Kleene Semantics K

Motivation

Kleene’s semantics was originally conceived to accommodate undecided mathematical statements.

It models a situation where the third logical value $\perp$ intuitively represents the notion of ”undecided”, or ”state of partial ignorance”.

A sentence is assigned a value $\perp$ just in case it is not known to be either true or false.
Kleene Semantics $K$

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 $\bot$.

But it is certainly either true or false and hence $\bot$ represents our ignorance rather than total unknown.
Kleene Semantics $K$

**Language**
We adopt the same language as in a case of classical, Łukasiewicz’s $L$, and Heyting $H$ semantics, i.e.

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

**Connectives**
We assume, as before, that $F < \bot < T$

The connectives $\neg, \cup, \cap$ of $K$ are defined as in $L, H$ semantics, i.e.

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

and for any $(x, y) \in \{T, \bot, F\} \times \{T, \bot, F\}$ we put

\[ x \cup y = \text{max}\{x, y\} \]

\[ x \cap y = \text{min}\{x, y\} \]
K Semantics: Connectives

K Implication
Kleene’s implication differ from L and H semantics. The K implication is defined by the same formula as the classical, i.e. for any \((x, y) \in \{T, \bot, F\} \times \{T, \bot, F\}\):

\[ x \Rightarrow y = \neg x \cup b \]

The connectives truth tables for the K negation, disjunction and conjunction are the same as the tables for L, H.

K implication table is

<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

Set of all K tautologies is

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

Relationship between \( \mathcal{L} \), \( \mathcal{H} \), \( \mathcal{K} \), and classical semantics is

\[ LT \neq KT, \quad HT \neq KT, \quad \text{and} \quad KT \subset T \]

Proof Obviously \( \models_L (a \Rightarrow a) \) and \( \models (a \Rightarrow a) \) We take \( v \) such that \( v(a) = \bot \) and evaluate in 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 hence

\[ LT \neq KT \quad \text{and} \quad LT \neq KT \]
K Tautologies

The third property

\[ KT \subset T \]

follows directly from the fact that, as in the \( L, H \) case, if we restrict the \( K \) connectives definitions functions to the values \( T \) and \( F \) only we get the functions defining the classical connectives

All together we have proved that the classical semantics extends all three \( L, H \) and \( K \) semantics, i.e.

\[ LT \subset T, \ HT \subset T, \ \text{and} \ K \subset T \]
Verification and Decidability

The following theorem justifies the correctness of the truth table method of tautology verification for \( L, H, K \) semantics.

**Theorem 1**

For any formula \( A \) of \( \mathcal{L}_{\{\neg, \Rightarrow, \cup, \cap\}} \), for any \( M \in \{L, H, K\} \)

\[
\models_M A \quad \text{if and only if} \quad v_A \models_M A
\]

for all \( v_A : \text{VAR}_A \rightarrow \{T, \bot, F\} \)

We also say that

\[
\models_M A \quad \text{if and only if} \quad \text{all } v_A \text{ are restricted } M \text{ models for } A,
\]

and \( M \in \{L, H, K\} \)
L, H, K Decidability

The following theorem proves the decidability of the tautology verification procedure for L, H, K semantics.

**Theorem 2**

For any formula $A$ of $L_{\{\neg, \Rightarrow, \cup, \cap\}}$, one has to examine at most $3^{\text{VAR}_A}$ truth assignments $v_A : \text{VAR}_A \rightarrow \{F, \perp, T\}$ in order to decide whether

$$\models_M A \quad \text{or} \quad \not\models_M A$$

i.e. the notion of $M$ tautology is decidable for any semantics $M \in \{L, H, K\}$

**Proofs of Theorems 1, 2** are carried in the same way as in case of classical semantics and are left as an exercise.
Exercise
We know that formulas

\[ ((a \land b) \Rightarrow a) , \ (a \Rightarrow (a \lor b)) , \ (a \Rightarrow (b \Rightarrow a)) \]

are classical tautologies.

Show that none of them is K tautology.

Solution
Consider any \( v \) such that \( v(a) = v(b) = \bot \).

We evaluate (in short hand notation)
\[ v^*(((a \land b) \Rightarrow a) = (\bot \land \bot) \Rightarrow \bot = \bot \Rightarrow \bot = \bot \]
K Tautologies Revisited

\[ v^*((a \Rightarrow (a \cup b))) = \bot \Rightarrow (\bot \cup \bot) = \bot \Rightarrow \bot = \bot \quad \text{and} \]
\[ v^*((a \Rightarrow (b \Rightarrow a))) = (\bot \Rightarrow (\bot \Rightarrow \bot) = \bot \Rightarrow \bot = \bot \]

This proves that any \( v \) such that

\[ v(a) = v(b) = \bot \]

is a counter model for all of them.

We generalize this example and prove (by induction over the degree of a formula) that a truth assignment \( v \) such that

\[ v(a) = \bot \quad \text{for all} \quad a \in \text{VAR} \]

is a counter model for any formula \( A \) of \( L\{\neg, \Rightarrow, \cup, \cap\} \).
We proved the following

**Theorem**

For any formula $A$ of $\mathcal{L}\{\neg,\Rightarrow,\cup,\cap\}$, $\not\models_K A$

In particular, the set of all **K tautologies** is empty, i.e.

$$KT = \emptyset$$

Observe that the theorem does not invalidate relationships

$$LT \neq KT, \quad HT \neq KT, \quad \text{and} \quad KT \subset T$$

between $\mathcal{L}$, $H$, $K$, and classical semantics

They become now perfectly true statements

$$LT \neq \emptyset, \quad T \neq \emptyset, \quad \text{and} \quad \emptyset \subset T$$
K Tautologies Revisited

When we develop a new logic by defining its semantics we must make sure for the semantics to be such that it has a non empty set of its tautologies

This is why we adopted (Set 2) the following definition

Definition
Given a language $\mathcal{L}_{CON}$ and its semantics $\mathcal{M}$
We say that the semantics $\mathcal{M}$ is well defined if and only if its set $\mathcal{MT}$ of all tautologies is non empty, i.e. when

$\mathcal{MT} \neq \emptyset$
The semantics $K$ is an example of a correctly and carefully defined semantics that is not well defined in terms of the above definition.

Obviously the semantics $L$ and $H$ are well defined.

We write is as a following separate fact.
K Tautologies Revisited

Fact

The semantics $L$ and $H$ are **well defined**, but the Kleene semantics $K$ is not.

*K* semantics also provides a justification for a need of introducing a **distinction** between correctly and well defined semantics.

This is the main **reason**, beside its **historical value**, why it is included here.
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’s semantics follows the principle that the third logical value, denoted now by $m$ (for miningless) 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.

$$SF = F, \quad ST = T$$

$S$ also **asserts** that meaningfulness $m$ is false, i.e

$$Sm = F$$
**B Semantics: Language**

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

$$\mathcal{L}_B = \mathcal{L}_{\{\neg, S, \Rightarrow, \cup, \cap\}}$$

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

Observe that directly from the definition we have that

$$\mathcal{F} \subset \mathcal{F}_B$$

The formula $SA$ reads "assert $A$"
**B Semantics: Connectives**

### Negation

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

### Conjunction

<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>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: Connectives

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>

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 Semantics: Connectives, Tautology**

**Assertion**

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

For all other steps of definition of B semantics we follow the standard established for the M semantics, as we did in all previous cases.

In particular the set of all B tautologies is

\[ \text{BT} = \{ A \in F : \models_B A \} \]
B Semantics: Tautology

We get by easy evaluation that

\[ \models_B (Sa \cup \neg Sa) \]

This proves that \( BT \neq \emptyset \), what means that

B semantics is **well defined**
Observe that not all formulas containing the connective $S$ are $B$ tautologies, for example we have that

$$\not\models_B (a \cup \neg Sa), \not\models_B (Sa \cup \neg a), \not\models_B (Sa \cup S\neg a)$$

as any truth assignment $v$ such that $v(a) = m$ is a counter model for all of them, because

$$m \cup x = m \text{ for all } x \in \{F, m, T\} \text{ and } Sm \cup S\neg m = F \cup Sm = F \cup F = F$$
B Semantics: Tautology

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

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, i.e.

$$T \cap BT = \emptyset$$

Observation

A formula $A \in \mathcal{F}_B$ to be considered to be a B tautology must contain the connective $S$ in front of each variable appearing in $A$