CHAPTER 3

Propositional Semantics: Classical and Many Valued

1 Formal Propositional Languages

We define here a general notion of a propositional language. We obtain, as specific cases, various languages for propositional classical logic as well as languages for many non-classical logics.

We assume that any propositional language contains a countably infinite set VAR of propositional variables. What distinguishes one propositional language from the other is the choice of its set CON of *propositional connectives.* We adopt a notation $\mathcal{L}_{CON}$ for a *propositional language* with the set CON of logical connectives. For example, the language $\mathcal{L}_{\{\neg\}}$ denotes a propositional language with only one connective $\neg$. The language $\mathcal{L}_{\{\neg,\Rightarrow\}}$ denotes that the language has only two connectives $\neg$ and $\Rightarrow$ adopted as propositional connectives. All propositional languages share the general way their sets of *formulas* are formed.

Theoretically one can use any symbols to denote propositional connectives. But there are some preferences, as connectives have a long history and intuitive meaning. The formal meaning, i.e. a *semantics* for a given language is discussed and defined in the next section.

Different semantics can share the same language. For example, the language $\mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow\}}$ is used as a propositional language for classical logic semantics, intuitionistic logic semantics, and some many valued logics semantics. It is also possible for several languages to share the same semantics. The classical propositional logic is the best example of such situation. We will prove in the section **??** that the languages:

$$\mathcal{L}_{\{\neg\Rightarrow\}}, \mathcal{L}_{\{\neg\cap\}}, \mathcal{L}_{\{\neg\cup\}}, \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow\}}, \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow,\Leftrightarrow\}},$$

and even two languages with only one binary propositional connectives, denoted usually by $\uparrow$ and $\downarrow$, respectively, i.e languages $\mathcal{L}_{\{\uparrow\}}, \mathcal{L}_{\{\downarrow\}}$ all share the same semantics characteristic for the classical propositional logic.

The connectives have well established symbols and names, even if their semantics can differ. We use names *negation, conjunction, disjunction*, *implication* and *equivalence (biconditional)* for $\neg, \cap, \cup, \Rightarrow, \Leftrightarrow$, respectively. The connective $\uparrow$ is called *alternative negation* and $A \uparrow B$ reads: *not both A and B*. The connective $\downarrow$ is called *joint negation* and $A \downarrow B$ reads: *neither A nor B*.

Other most common propositional connectives are probably modal connectives of *possibility* and *necessity*. Standard modal symbols are $\Box$ for *necessity* and $\Diamond$ for *possibility*. We will also use symbols $\mathbf{C}$ and $\mathbf{I}$ for modal connectives of possibility and necessity, respectively.

A formula $\mathbf{C}A$, or $\Diamond A$ reads: *it is possible that A* , *A is possible*, and a formula $\mathbf{I}A$, or $\Box A$ reads: *it is necessary that A*, *A is necessary.*

A motivation for notation $\mathbf{C}$ and $\mathbf{I}$ arises from topological semantics for modal S4 and S5 logics. $\mathbf{C}$ becomes equivalent to a set closure operation, and $\mathbf{I}$ becomes equivalent to a set interior operation.

The symbols $\Diamond$, $\mathbf{C}$ and $\Box$, $\mathbf{I}$ are not the only symbols used for modal connectives. Other symbols include N for necessity and P for possibility. There is also a variety of modal logics created by computer scientists, all with their set of symbols and motivations for their use and their semantics. The modal logics *extend* the classical logic and hence their language is for example $\mathcal{L}_{\{\Box,\Diamond,\neg,\cap,\cup,\Rightarrow\}}$.

Knowledge logics also extend the classical logic. Their languages add to the classical connectives a new *knowledge* connective, often denoted denoted by K. The formula KA reads: *it is known that A* , *A is known.* The language of a knowledge logic is for example $\mathcal{L}_{\{K,\ \neg,\cap,\cup,\Rightarrow\}}$.

Autoepistemic logics use a *believe* connective, often denoted by B. The formula BA reads: *it is believed that A*. They also extend the classical logic and hence their language is $\mathcal{L}_{\{B,\ \neg,\cap,\cup,\Rightarrow\}}$.

Temporal logics add temporal connectives to the set of classical propositional connectives. For example some of them use connectives (operators, as they are often called) F, P, G, and H to denote the following intuitive readings. FA reads *A is true at some future time*, PA reads *A was true at some past time*, GA reads *A will be true at all future times*, and HA reads *A has always been true in the past.* In order to take account of this variation of truth-values over time, some formal semantics were created, and many more will be created.

It is possible to create connectives with more then one or two arguments, but we allow here only one and two argument connectives, as logics which will be discussed here use only those two kind of connectives.

We adopt the following definition, common to all propositional languages considered in our propositional logics investigations.

Definition 1 (Propositional Language)

By a propositional language with a set CON of **propositional connectives** *we understand a pair*

$$\mathcal{L}_{CON} = (\mathcal{A}, \mathcal{F}). \tag{1}$$

$\mathcal{A}$ is a called **propositional alphabet** *and $\mathcal{F}$ is called a set of* **propositional formulas** *of the language $\mathcal{L}_{CON}$. The alphabet $\mathcal{A}$, the set CON of propositional connectives, and the set $\mathcal{F}$ of propositional formulas are defined as follows.*

1. Alphabet $\mathcal{A}$

The alphabet $\mathcal{A} = VAR \cup CON \cup PAR$, where VAR, CON, PAR are all disjoint sets and VAR, CON are non-empty sets. VAR is countably infinite and is called a set of **propositional variables**; we denote elements of VAR by $a, b, c, ...$ etc, (with indices if necessary).

CON is a finite set of **propositional connectives**, PAR is a set of **auxiliary symbols**. We assume that $PAR \neq \emptyset$ and contains two elements $(,)$ called parentheses, i.e. $PAR = \{(,)\}$. The set PAR may be empty, for example of a case of Polish notation, but we assume that it contains two parenthesis as to make the reading of formulas more natural and uniform.

2. Propositional connectives CON

We assume that the set CON is non empty and finite. We specify it for specific cases (specific logics). It is possible to consider languages with connectives which have more then one or two arguments, nevertheless we restrict ourselves to languages with one or two argument connectives only.
We assume that

$$CON = C_1 \cup C_2$$

where C_1 is a finite set (possibly empty) of **unary connectives**, C_2 is a finite set (possibly empty) of **binary connectives** of the language $\mathcal{L}_{CON}$.

2. Set $\mathcal{F}$ of formulas

The set $\mathcal{F}$ is built recursively from the elements of the alphabet $\mathcal{A}$, i.e. $\mathcal{F} \subseteq \mathcal{A}^*$, where $\mathcal{A}^*$ is the set of all finite sequences (strings) form from elements of $\mathcal{A}$ and is defined as follows.

The set $\mathcal{F}$ of all **formulas** of a propositional language $\mathcal{L}_{CON}$ is **the smallest set**, such that the following conditions hold:

(1) $VAR \subseteq \mathcal{F}$;

(2) if $A \in \mathcal{F}$, $\bigtriangledown \in C_1$ i.e $\bigtriangledown$ is an one argument connective, then $\bigtriangledown A \in \mathcal{F}$;

(3) if $A, B \in \mathcal{F}$, $\circ \in C_2$ i.e $\circ$ is a two argument connective, then
$(A \circ B) \in \mathcal{F}$.

The elements of the set $VAR \subseteq \mathcal{F}$ are called **atomic formulas**. The set $\mathcal{F}$ is also called a set of all **well formed formulas** (wff) of the language $\mathcal{L}_{CON}$.

The alphabet $\mathcal{A}$ is *countably infinite* and consequently the set $\mathcal{A}^*$ of all finite sequences of elements of $\mathcal{A}$ is also countably infinite. By definition, $\mathcal{F} \subseteq \mathcal{A}^*$, hence the set $\mathcal{F}$ is also countably infinite. We state as separate fact.

Fact 1 *For any propositional language $\mathcal{L}_{CON} = (\mathcal{A}, \mathcal{F})$, the set $\mathcal{F}$ of formulas is countably infinite. We hence consider here only* **infinitely countable languages**.

Observation 1

When defining a language $\mathcal{L}_{CON}$ we choose not only the propositional connectives but also the symbols denoting them.

For example, $\mathcal{L}_1 = \mathcal{L}_{\{\neg\}}$ and $\mathcal{L}_2 = \mathcal{L}_{\{\sim\}}$ are two different propositional languages both with negation as the only connective.

The choice of appropriate well established symbols for logical connectives depends on a personal preferences of books' authors and creators of different logics. One can find a variety of them in the literature. We presented some historical choices in the chapter 2.

Example 1

Let $\mathcal{L}_1 = \mathcal{L}_{\{\neg\}}$ and $\mathcal{L}_2 = \mathcal{L}_{\{\sim\}}$. The formulas of both languages $\mathcal{L}_1$, $\mathcal{L}_2$ are propositional variables or multiple negations of of a propositional variable.

The strings $a, \neg b, \neg\neg b, \neg\neg\neg a$ are well formed formulas of $\mathcal{L}_1$. The corresponding formulas of $\mathcal{L}_2$ are $a, \sim b, \sim\sim b, \sim\sim\sim a$.

Observe that the strings $(\neg a), \neg, \neg(\neg a), \neg(a), (\sim a), \neg, \sim (\sim a) \sim (a)$ are not well formed formulas of neither of the languages $\mathcal{L}_1, \mathcal{L}_2$.

We adopt the general definition of the set $\mathcal{F}$ of formulas of $\mathcal{L}_{CON}$ to for example the language $\mathcal{L}_{\{\sim\}}$ as follows.

Example 2

The set $\mathcal{F}$ of all **formulas** *of a propositional language $\mathcal{L}_{\{\sim\}}$ is the smallest set, such that the following conditions hold:*

(1) *$VAR \subseteq \mathcal{F}$ (atomic formulas);*

(2) *if $A \in \mathcal{F}$, then $\sim A \in \mathcal{F}$.*

Example 3

Consider now $\mathcal{L}_{CON}$ for the set of connectives $CON = \{\neg\} \cup \{\Rightarrow\}$, where $\neg \in C_1$ and $\Rightarrow \in C_2$. It means that we defined a language $\mathcal{L} = \mathcal{L}_{\{\neg, \Rightarrow\}}$.

By the initial recursive step we get for any $a \in VAR$, $a \in \mathcal{F}$. By the recursive step and its repetition we get for example that $\neg a \in \mathcal{F}$, $\neg\neg a \in \mathcal{F}$, $\neg\neg\neg a \in \mathcal{F}$, ... etc., i.e. get all formulas from the the example 1 language $\mathcal{L}_1$. But also we also get that $(a \Rightarrow a)$, $(a \Rightarrow b), \neg(a \Rightarrow b), (\neg a \Rightarrow b), \neg((a \Rightarrow a) \Rightarrow \neg(a \Rightarrow b))$.... etc. are all in $\mathcal{F}$ and infinitely many others.

Observe that $(\neg(a \Rightarrow b))), a \Rightarrow b, (a \Rightarrow)$ are not in $\mathcal{F}$.

Example 4

Consider $\mathcal{L} = \mathcal{L}_{CON}$ for $C_1 = \{\neg, P, N\}$, $C_2 = \{\Rightarrow\}$. If we understand P, N as a possibility and necessity connectives, the obtained language is called a modal language with only negation as non-modal connective.

The set of formulas $\mathcal{F}$ of $\mathcal{L}$ contains all formulas from example 3, but also formulas $Na, \neg Pa, P\neg a, (N\neg b \Rightarrow Pa), \neg P\neg a, ((N\neg b \Rightarrow Pa) \Rightarrow b)$, ... etc.

We adopt the general definition of the set $\mathcal{F}$ of formulas of $\mathcal{L}_{CON}$ to for example the *modal language* $\mathcal{L}_{\{\neg, P, N, \Rightarrow\}}$ as follows.

Example 5

The set $\mathcal{F}$ of all **formulas** *of a propositional language $\mathcal{L}_{\{\neg, P, N, \Rightarrow\}}$ is the smallest set, such that the following conditions hold:*

(1) $VAR \subseteq \mathcal{F}$ *(atomic formulas);*

(2) *if $A \in \mathcal{F}$, then $\neg A$, PA, NA $\in \mathcal{F}$;*

(3) *if $A, B \in \mathcal{F}$, then $(A \Rightarrow B) \in \mathcal{F}$.*

We introduce now formal definitions of basic syntactical notions of a main connective, a sub-formula of a given formula, and of a degree of a given formula.

Definition 2 (Main Connective)

Given a language $\mathcal{L}_{CON} = (\mathcal{A}, \mathcal{F})$.

For any connectives $\bigtriangledown \in C_1$ and $\circ \in C_2$,

$\bigtriangledown$ *is called* **a main connective** *of* $\bigtriangledown A \in \mathcal{F}$ *and*

$\circ$ *is* **a main connective** *of* $(B \circ C) \in \mathcal{F}$.

Observe that it follows directly from the definition of the set of formulas that for any formula $C \in \mathcal{F}$, exactly one of the following holds: C is atomic, or there is a unique formula A and a unique unary connective $\bigtriangledown \in C_1$, such that C is of the form $\bigtriangledown A$, or here are unique formulas A and B and a unique binary connective $\circ \in C_2$, such that C is $(A \circ B)$. We have hence proved the following.

Observation 2

For any formula $A \in \mathcal{F}$, A is atomic or has a unique main connective.

Example 6

The main connective of $(a \Rightarrow \neg Nb)$ is $\Rightarrow$. The main connective of $N(a \Rightarrow \neg b)$ is N. The main connective of $\neg(a \Rightarrow \neg b)$ is $\neg$ The main connective of $(\neg a \cup \neg(a \Rightarrow b))$ is $\cup$.

Definition 3

We define a notion of direct **a direct sub-formula** *as follows: 1. Atomic formulas have no direct sub-formulas. 2. A is a direct sub-formula of a formula $\bigtriangledown A$, where $\bigtriangledown$ is any unary connective. 3. A, B are direct sub-formulas of a formula $(A \circ B)$ where $\circ$ is any binary connective.*

Directly from the definition 3 we get the following.

Observation 3

For any formula A, A is atomic or has exactly one or two direct sub-formulas depending on its main connective being unary or binary, respectively.

Example 7

The formula $(\neg a \cup \neg(a \Rightarrow b))$ has exactly $\neg a$ and $\neg(a \Rightarrow b)$ as direct sub-formulas.

Definition 4

We define a notion of a **sub-formula** *of a given formula in two steps. 1. For any formulas A and B, A is a* **proper sub-formula** *of B if there is sequence of formulas, beginning with A, ending with B, and in which each term is a direct sub-formula of the next. 2. A* **sub-formula** *of a given formula A is any proper sub-formula of A, or A itself.*

The formula $(\neg a \cup \neg(a \Rightarrow b))$ has $\neg a$ and $\neg(a \Rightarrow b)$ as direct sub-formula. The formulas $\neg a$ and $\neg(a \Rightarrow b)$ have a and $(a \Rightarrow b)$ as their direct sub-formulas, respectively. The formulas $\neg a$, $\neg(a \Rightarrow b)$, a and $(a \Rightarrow b)$ are all proper sub-formulas of the formula $(\neg a \cup \neg(a \Rightarrow b))$ itself. Atomic formulas a and b are direct sub-formulas of $(a \Rightarrow b)$. Atomic formula b is a proper sub-formula of $\neg b$.

Example 8

The set of all sub-formulas of

$$(\neg a \cup \neg(a \Rightarrow b))$$

consists of $(\neg a \cup \neg(a \Rightarrow b))$, $\neg a$, $\neg(a \Rightarrow b)$, $(a \Rightarrow b)$, a and b.

Definition 5 (Degree of a formula)

By a degree of a formula we mean the number of occurrences of logical connectives in the formula.

The degree of $(\neg a \cup \neg(a \Rightarrow b))$ is 4. The degree of $\neg(a \Rightarrow b))$ is 2. The degree of $\neg a$ is 1. The degree of a is 0.

Note that the degree of any proper sub-formula of A must be one less than the degree of A. This is the central fact upon mathematical induction arguments are based. Proofs of properties formulas are usually carried by mathematical induction on their degrees.

Example 9

Given a formula $A : (\neg \mathbf{I}\neg a \Rightarrow (\neg \mathbf{C}a \cup (\mathbf{I}a \Rightarrow \neg \mathbf{I}b)))$.
1. The language to which A belongs is a modal language $\mathcal{L}_{\{\neg,\mathbf{C},\mathbf{C},\cup,\cap,\Rightarrow\}}$ *with the possibility connective* $\mathbf{C}$ *and necessity connective* $\mathbf{C}$*. Both of them are one argument connectives.*
2. The main connective of A is $\Rightarrow$*, the degree of A is 11.*
3. All sub-formulas of A of the degree 0 are the atomic formulas a, b*. All sub-formulas of A of the degree 1 are:* $\neg a$, $\mathbf{C}a$, $\mathbf{I}a$, $\mathbf{I}b$.

Languages with Propositional Constants

A propositional language $\mathcal{L}_{CON} = (\mathcal{A}, \mathcal{F})$ is called a language with propositional constants, when we distinguish certain constants, like symbol of truth T or falsehood F, or other symbols as elements of the alphabet. The propositional constants are zero-argument connectives. In this case the set CON of logical connectives contains a a finite, non empty set of *zero argument connectives* C_0, called **propositional constants**, i.e. we put

$$CON = C_0 \cup C_1 \cup C_2.$$

The definition of the set $\mathcal{F}$ of all **formulas** of the language $\mathcal{L}_{CON}$ contains now an additional recursive step and goes as follows.

The set $\mathcal{F}$ of all formulas of the language $\mathcal{L}_{CON}$ with propositional constants is the smallest set built from the signs of the alphabet $\mathcal{A}$, i.e. $\mathcal{F} \subseteq \mathcal{A}^*$, such that the following conditions hold:

(1) $VAR \subseteq \mathcal{F}$ (atomic formulas),

(2) $C_0 \subseteq \mathcal{F}$ (atomic formulas),

(3) if $A \in \mathcal{F}$, $\triangledown \in C_1$ i.e $\triangledown$ is an one argument connective, then $\triangledown A \in \mathcal{F}$,

(4) if $A, B \in \mathcal{F}$, $\circ \in C_2$ i.e $\circ$ is a two argument connective, then $(A \circ B) \in \mathcal{F}$.

Example 10

Let $\mathcal{L} = \mathcal{L}_{\{T,\neg,\cap\}}$, i.e. $C_0 = \{V\}$. Atomic formulas of $\mathcal{L}$ are all variables and the symbol T.

The language admits formulas that involve the symbol T like $T, \neg T$, $(T \cap a)$, $(\neg a \cap \neg T)$, $\neg(b \cap T)$, etc... We might interpret the symbol T as a symbol of truth (statement that is always true).

Here are some exercises and examples dealing with the formal definition of propositional languages, syntactical correctness, and their expressiveness.

Exercise 1

Given a language $\mathcal{L} = \mathcal{L}_{\{\neg,\ C,I,\cup,\cap,\Rightarrow\}}$ and the following set S.

$$S = \{\mathbf{C}\neg a \Rightarrow (a \cup b),\ (\mathbf{C}(\neg a \Rightarrow (a \cup b))),\ C\neg(a \Rightarrow (a \cup b))\}$$

Determine which of the elements of $\mathcal{S}$ are, and which are not well formed formulas of $\mathcal{L}$. If $A \in \mathcal{S}$ is not a correct formula write its corrected version. For each correct or corrected formula determine its main connective, its degree and write what it says in the natural language.

Solution

A_1 : $\mathbf{C}\neg a \Rightarrow (a \cup b)$ is not a well formed formula. The corrected formula is $(\mathbf{C}\neg a \Rightarrow (a \cup b))$. Its main connective is $\Rightarrow$ and the degree is 4. The corrected formula says: *If negation of a is possible, then we have a or b.*

Another corrected formula is $\mathbf{C}(\neg a \Rightarrow (a \cup b))$. Its main connective is $\mathbf{C}$, the degree is 4. The corrected formula says: *It is possible that not a implies a or b.*

A_2 : $(\mathbf{C}(\neg a \Rightarrow (a \cup b)))$ is not a well formed formula. The correct formula is $\mathbf{C}(\neg a \Rightarrow (a \cup b))$. The main connective is $\mathbf{C}$, the degree is 4. The formula $\mathbf{C}(\neg a \Rightarrow (a \cup b))$ says: *It is possible that not a implies a or b* .

3. The formula $\mathbf{C}\neg(a \Rightarrow (a \cup b))$ is a correct formula. The main connective is $\mathbf{C}$, the degree is 4. The formula says: *the negation of the fact that a implies a or b is possible.*

Exercise 2

Given a set S of formulas:

$$S = \{((a \Rightarrow \neg b) \Rightarrow \neg a), \Box(\neg\Diamond a \Rightarrow \neg a), (a \cup \neg(a \Rightarrow b))\}.$$

Define a formal language $\mathcal{L}_{CON}$ *to which to which all formulas in* S *belong, i.e. a language* **determined** *by the set* S*.*

Solution
Any propositional language $\mathcal{L}_{\mathcal{CON}}$ is determined by its set of connectives. The connectives appearing in the formulas of the set S are: $\Rightarrow, \neg b, \Box, \Diamond$ and $\cup$. Hence the required language is $\mathcal{L}_{\{\neg,\Box,\Diamond,\cup,\Rightarrow\}}$.

Exercise 3

Write down a set S_1 *all sub-formulas of the* $\Diamond((a \cup \neg a) \cap b)$*, a set* S_2 *all proper sub-formulas of* $\neg(a \Rightarrow (b \Rightarrow))$*.*

Solution
The set S_1 of all sub-formulas of $\Diamond((a \cup \neg a) \cap b)$ is

$$S_1 = \{\Diamond((a \cup \neg a) \cap b),\ ((a \cup \neg a) \cap b),\ (a \cup \neg a),\ \neg a,\ b,\ a\}$$

a, b are atomic sub-formulas, and $\Diamond((a \cup \neg a) \cap b)$ is not a proper sub-formula.

The set S_2 of all proper sub-formulas of $\neg(a \Rightarrow (b \Rightarrow c))$ is

$$S_2 = \{(a \Rightarrow (b \Rightarrow c)),\ (b \Rightarrow c), a, b, c\}.$$

Exercise 4

Write the following natural language statement **S***:*

"From the fact that it is possible that Anne is not a boy we deduce that it is not possible that Anne is not a boy or, if it is possible that Anne is not a boy, then it is not necessary that Anne is pretty."

in the following two ways.

1. *As a formula* $A_1 \in \mathcal{F}_1$ *of a language* $\mathcal{L}_{\{\neg,\ \Box,\ \Diamond,\ \cap,\ \cup,\ \Rightarrow\}}$.

2. *As a formula* $A_2 \in \mathcal{F}_2$ *of a language* $\mathcal{L}_{\{\neg,\ \cap,\ \cup,\ \Rightarrow\}}$.

Solution
1. We translate the statement **S** into a formula A_1 of the modal language $\mathcal{L}_{\{\neg,\ \Box,\ \Diamond,\ \cap,\ \cup,\ \Rightarrow\}}$ as follows.

Propositional variables are: a, b. The variable a denotes statement *Anne is a boy* and b denotes a statement *Anne is pretty*.

Propositional modal connectives are: $\Box$, $\Diamond$. The connective $\Diamond$ reads *it is possible that*, and $\Box$ reads *it is necessary that*.

Translation: the formula A_1 is $(\Diamond\neg a \;\Rightarrow\; (\neg\Diamond\neg a \cup (\Diamond\neg a \Rightarrow \neg\Box b)))$.

2. We translate our statement into a formula A_2 of the language $\mathcal{L}_{\{\neg,\ \cap,\ \cup,\ \Rightarrow\}}$ as follows.

Propositional variables are: a, b. The variable a denotes statement *it is possible that Anne is not a boy* and b denotes a statement *it is necessary that Anne is pretty.* **Translation:** the formula A_2 is $(a \Rightarrow (\neg a \cup (a \Rightarrow \neg b)))$.

Exercise 5

Write the following natural language statement **S***:*

"For all natural numbers $n \in N$ the following implication holds: if $n < 0$, then there is a natural number m, such that it is possible that $n + m < 0$, or it is not possible that there is a natural number m, such that $m > 0$"

in the following two ways.

1. *As a formula A_1 of a language $\mathcal{L}_{\{\neg,\ \cap,\ \cup,\ \Rightarrow\}}$.*

2. *As a formula A_2 of a language $\mathcal{L}_{\{\neg,\ \Box,\ \Diamond,\ \cap,\ \cup,\ \Rightarrow\}}$.*

Solution
1. We translate the statement **S** into a formula A_1 of the language $\mathcal{L}_{\{\neg,\ \cap,\ \cup,\ \Rightarrow\}}$ as follows.

Propositional variables are: a, b. The variable a denotes statement *For all natural numbers $n \in N$ the following implication holds: if $n < 0$, then there is a natural number m, such that it is possible that $n + m < 0$.* The variable b denotes statement *it is not possible that there is a natural number m, such that $m > 0$.* **Translation:** the formula A_1 is $(a \cup \neg b)$.

2. We translate the statement **S** into a formula A_2 of a language $\mathcal{L}_{\{\neg,\ \Box,\ \Diamond,\ \cap,\ \cup,\ \Rightarrow\}}$ as follows. Propositional variables are: a, b. The variable a denotes statement *For all natural numbers $n \in N$ the following implication holds: if $n < 0$, then there is a natural number m, such that it is possible that $n + m < 0$.* The variable b denotes statement *there is a natural number m, such that $m > 0$.* **Translation:** the formula A_2 is $(a \cup \neg\Diamond b)$.

2 Extensional Semantics M

Given a propositional language $\mathcal{L}_{CON}$, the symbols for its connectives always have some intuitive meaning. A formal definition of the meaning of these symbols is called a semantics for the language $\mathcal{L}_{CON}$. A given language can have different semantics but we always define them in order to single out special formulas of the language, called tautologies, i.e. formulas of the language that is always true under the given semantics.

We introduced in Chapter 2 a notion of a classical propositional semantics, discussed its motivation and underlying assumptions. The fist assumption was that we consider only two logical values. The other one was that all classical propositional connectives are *extensional*. We have also observed that in everyday language there are expressions such as "I believe that", "it is possible that", " certainly", etc.... and they are represented by some propositional connectives which are not extensional. Non-extensional connectives do not play any role in mathematics and so are not discussed in classical logic and will be studied separately. The extensional connectives are defined intuitively as such that the logical value of the formulas form by means of these connectives and certain given formulas depends only on the logical value(s) of the given formulas.

We adopt a following formal definition of extensional connectives for a propositional language $\mathcal{L}_{CON}$ under a semantics **M** with the set LV of logical values.

Definition 6 (Extensional Connectives)

Let $\mathcal{L}_{CON}$ be such that $CON = C_1 \cup C_2$, where C_1, C_2 are the sets of unary and binary connectives, respectively. Let LV be a non-empty set of logical values. A connective $\bigtriangledown \in C_1$ or $\circ \in C_2$ is called **extensional** *if it is defined by a respective function*

$$\bigtriangledown : \; LV \longrightarrow LV \quad \text{or} \quad \circ : \; LV \times LV \longrightarrow LV.$$

A semantics **M** *for a language $\mathcal{L}_{CON}$ is called* **extensional** *provided all connectives in CON are extensional.*

A semantics with a set of m-logical values is called a m-valued semantics. The classical semantics is a special case of a 2-valued extensional semantics. Given a language, its different semantics define corresponding different logics. Classical semantics defines classical propositional logic with its set of classical propositional tautologies. Many of m- valued logics are defined by various extensional semantics with sets of logical values LV with more then 2 elements. The languages of many important logics like modal, multi-modal, knowledge, believe, temporal logics, contain connectives that are not extensional. Consequently they are defined by the non-extensional semantics. The intuitionistic logic is based on the same language as the classical one, its Kripke Models semantics is not extensional. Defining a semantics for a given propositional language means more then defining its propositional connectives. The *ultimate goal* of any semantics is to define the notion of its own *tautology.* In order to define which formulas of $\mathcal{L}_{CON}$ we want to to be tautologies under a given semantics **M** we assume that the set LV of logical values of **M** always has a distinguished logical value, often denoted by T (for truth). We also can distinguish, and often we do, another special value F representing "absolute" falsehood. We will use these symbols T, F. We may also use other symbols like 1,0 or others. The value T serves to define a notion of a tautology (as a formula always "true"). Extensional semantics share not only the similar pattern of defining their connectives (definition 6), but also the method of defining the notion of a *tautology.*

We hence define a general notion of an extensional semantics (definition 7) as sequence of steps leading to the definition of a tautology. Here are the steps.

Step1: we define all connectives of $\mathbf{M}$ as specified by definition 6.

Step 2: we define the main component of the definition of a tautology, namely a function v that assigns to any formula $A \in \mathcal{F}$ its logical value form VL. It is often called a truth assignment and we will use this name.

Step 3: given a truth assignment v and a formula $A \in \mathcal{F}$, we define what does it mean that v *satisfies* A, i.e. that v is a *model* for A under semantics $\mathbf{M}$.

Step 4: we define a notion of *tautology* as follows: A is a tautology under semantics $\mathbf{M}$ if and only if all truth assignments v *satisfy* A, i.e. all truth assignments v are *models* for A.

We use a notion of a model because it is an important, if not the most important notion of modern logic. It is usually defined in terms of the notion of satisfaction. In classical propositional logic these two notions are the same. The use of expressions *" v satisfies A"* and *"v is a model for A"* is interchangeable. This is also a case for the extensional semantics; in particular for some non-classical semantics, like m-valued semantics discussed in this chapter. The notions of satisfaction and model are not interchangeable for predicate languages semantics. We already discussed these notions in Chapter 2 and will define them in full formality in Chapter (predicate Logic) The use of the notion of a model also allows us to adopt and discuss the standard predicate logic definitions of consistency and independence for propositional case.

Given a language $\mathcal{L}_{CON}$ and non-empty set LV of logical values. We assume that the set LV has a special, distinguished logical value which serves to define a notion of tautology under the semantics $\mathbf{M}$. We denote this distinguished value as T. We define formally a general notion of an extensional semantics $\mathbf{M}$ for $\mathcal{L}_{CON}$ as follows.

Definition 7 (Extensional Semantics)

A formal definition of an extensional semantics $\mathbf{M}$ *for a given language* $\mathcal{L}_{CON}$ *consists of specifying the following steps defining its main components.*

Step 1*: we define a set LV of logical values and its distinguish value T, and define all connectives of* $\mathcal{L}_{CON}$ *to be extensional;*
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* $\mathbf{M}$*.*

What differs one semantics from the other is the choice of the set LV of logical values and definition of the the connectives of $\mathcal{L}_{CON}$, i.e. the components defined in the Step1. The definitions for the Steps 2 and 3, 4 are modification of the definitions established for the classical case and they are as follows.

Step 1: we follow the definition 6 to define the connectives of **M**.

Step 2 : we define a function called truth assignment and its extension in terms of the propositional connectives as defined in the Step 1. We us the term **M** truth assignment and **M** truth extention to stress that it is defined relatively to a given semantics **M**.

Definition 8 (M Truth Assignment)

Let LV be the set of logical values of **M** *and VAR the set of propositional variables of the language* $\mathcal{L}_{CON}$*. Any function* $v : VAR \longrightarrow LV$*, is called a truth assignment under semantics* **M***, for short* **M** *truth assignment.*

Definition 9 (M Truth Extension)

Given **M** *truth assignment*
$v : VAR \longrightarrow LV$*. We define its extension* v^* *to the set* $\mathcal{F}$ *of all formulas of* $\mathcal{L}_{CON}$ *as any function*

$$v^* : \mathcal{F} \longrightarrow LV,$$

such that the following conditions are satisfied.

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

$$v^*(a) = v(a);$$

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

$$v^*(\triangledown A) = \triangledown v^*(A), \quad v^*((A \circ B)) = \circ(v^*(A), v^*(B)).$$

We call the v^* *the* **M truth extension***.*

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 in the Step1.
We use names "**M** truth assignment", "**M** truth extension" to stress that we define them for the set of logical values of **M** and moreover, that the extension of v connects the formulas of the language with the connectives as defined by the semantics **M**.

Notation Remark For a given function f, we use a symbol f^* to denote its *extension* to a larger domain. Mathematician often use the same symbol f for both a function and its extension f^*.

Step 3: the notions of satisfaction and model are interchangeable in extensional semantics. They are not interchangeable in other propositional semantics and in semantics for predicate languages.We define them as follows.

Definition 10 (M Satisfaction, Model)

Given an **M** *truth assignment*

$v : VAR \longrightarrow \ LV$ *and its* **M** *truth extension* v^*. *Let* $T \in LV$ *be the distinguished logical value. We say that*
the truth assignment v **M satisfies** *a formula* A *if and only if* $v^*(A) = T$.
We write symbolically

$$v \models_{\mathbf{M}} A.$$

Any truth assignment v*, such that* $v \models_{\mathbf{M}} A$ *is called* **M model** *for* A.

Definition 11 (M Counter Model)

Given an **M** *truth assignment*

$v : VAR \longrightarrow \ LV$. *Let* $T \in LV$ *be the distinguished logical value. We say that* v **does not satisfy** *a formula* $A \in \mathcal{F}$ *if and only if* $v^*(A) \neq T$.

We denote it by

$$v \not\models_{\mathbf{M}} A.$$

A any v*, such that* $v \not\models_{\mathbf{M}} A$ *is called* **M counter model** *for* A.

Step 4: we define the notion of a tautology under semantics **M**, called **M tautology** as follows.

Definition 12 (M Tautology)

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

A *is* **M tautology** *if and only if* $v \models_{\mathbf{M}} A$*, for all truth assignments* v, $v : VAR \longrightarrow \ LV$. *We denote it as*

$$\models_{\mathbf{M}} A.$$

We also say that A *is* **M tautology** *if and only if all truth assignments* v *are* **M models** *for* A.

Observe that immediately from definition 11 we get the following equivalent form of the definition 12.

Definition 13

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

A *is a* **M tautology** *if and only if* $v^*(A) = T$*, for all truth assignments* v, $v : VAR \longrightarrow \ LV$.

We denote by **MT** the set of all tautologies under the semantic **M**, i.e.

$$\mathbf{MT} = \{A \in \mathcal{F} : \models_M A.\} \tag{2}$$

Obviously, when we develop a logic by defining its semantics we want the semantics to be such that the logic has a non empty set of its tautologies. We stress that fact by putting it in a form of the following definition.

Definition 14

Given a language $\mathcal{L}_{CON}$ and its extensional semantics **M** *(definition 7), we say that the semantics* **M** *is* **well defined** *if and only if its set* **MT** *of all tautologies (2) is non empty, i.e. when*

$$\mathbf{MT} \neq \emptyset \tag{3}$$

We follow the definitions and pattens established here first in section 3. We use them to define and discuss in details the classical propositional semantics. Definitions and short discussions of some of the many-valued semantics follow next in section 5. Many valued logics had their beginning in the work of Łukasiewicz (1920). He was the first to define a 3- valued extensional semantics for a language $\mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow\}}$ of classical logic, and called it *a three valued logic* for short. The other logics, now of historical value followed and we will discuss some of them. In particular we present a Heyting 3-valued semantics as an introduction to the definition and discussion of first ever semantics for the intuitionistic logic and some modal logics. It was proposed by J.C.C McKinsey and A. Tarski in 1946-48 in a form of cylindrical algebras, now called pseudo-boolean algebras, or Heyting algebras. The semantics in a form of abstract algebras are called algebraic models for logics. It became a separate field of modern logic. The algebraic models are generalization of the extensional semantics, hence the importance of this section. It can me treated as an introduction to *algebraic models for logics.* It will be discussed again in chapter**??**.

3 Classical Semantics

We follow now **Steps 1- 4** of the definition 7 of extensional semantics adopted to the case of the classical propositional logic.

The language is $\mathcal{L}_{\{\neg,\ \cup,\ \cap,\ \Rightarrow,\ \Leftrightarrow\}}$. The set LV of logical values is $\{T,\ F\}$. The letters T, F stand as symbols of truth and falsehood, respectively. We adopt T as the distinguished value. There are other notations for logical values, for example 0,1, but we will use T, F.

Step 1: Definition of connectives

Negation is a function $\neg : \{T, F\} \longrightarrow \{T, F\}$, such that $\neg(T) = F$, $\neg(F) = T$. We write it as $\neg T = F$, $\neg F = T$.

Notation: we write the name of a function (our connective) *between the arguments*, not in front as in function notation, i.e. we write for any binary connective $\circ$, $T \circ T = T$ instead of $\circ(T, T) = T$.

Conjunction is a function $\cap : \{T, F\} \times \{T, F\} \longrightarrow \{T, F\}$, such that $\cap(T, T) = T$, $\cap(T, F) = F$, $\cap(F, T) = F$, $\cap(F, F) = F$. We write it as $T \cap T = T$, $T \cap F = F$, $F \cap T = F$, $F \cap F = F$.

Disjunction is a function $\cup : \{T, F\} \times \{T, F\} \longrightarrow \{T, F\}$, such that $\cup(T, T) = T$, $\cup(T, F) = T$, $\cup(F, T) = T$, $\cup(F, F) = F$. We write it as $T \cup T = T$, $T \cup F = T$, $F \cup T = T$, $F \cup F = F$.

Implication is a function $\Rightarrow: \{T, F\} \times \{T, F\} \longrightarrow \{T, F\}$, such that $\Rightarrow(T, T) = T$, $\Rightarrow(T, F) = F$, $\Rightarrow(F, T) = T$, $\Rightarrow(F, F) = T$. We write it as $T \Rightarrow T = T$, $T \Rightarrow F = F$, $F \Rightarrow T = T$, $F \Rightarrow F = T$.

Equivalence is a function $\Leftrightarrow: \{T, F\} \times \{T, F\} \longrightarrow \{T, F\}$, such that $\Leftrightarrow(T, T) = T$, $\Leftrightarrow(T, F) = F$, $\Leftrightarrow(F, T) = F$, $\Leftrightarrow(T, T) = T$. We write it as $T \Leftrightarrow T = T$, $T \Leftrightarrow F = F$, $F \Leftrightarrow T = F$, $T \Leftrightarrow T = T$.

We write function defining the connectives in a standard form of tables defining operations in finite sets. We call these tables *truth tables definition* of propositional connectives, or *classical connectives truth tables* for short.

Classical Connectives Truth Tables

$\neg$	T	F
	F	T

$\cap$	T	F
T	T	F
F	F	F

$\cup$	T	F
T	T	T
F	T	F

$\Rightarrow$	T	F
T	T	F
F	T	T

$\Leftrightarrow$	T	F
T	T	F
F	F	T

As ultimate goal of our semantics is to define the notion of tautology, a formula that is always true, we assume that the set $\{T, F\}$ of our logical values is ordered and $F < T$, This makes the symbol T (for truth) the "greatest" logical value, what truth supposed to be. We now can write simple formulas defining the connectives (respective function) as follows.

Classical Connectives Formulas

$\neg : \{F, T\} \longrightarrow \{F, T\}$, such that $\neg F = T$, $\neg T = F$.

$\cap : \{F, T\} \times \{F, T\} \longrightarrow \{F,\ T\}$, such that for any $x, y \in \{F, T\}$, $\cap(x, y) = min\{x,\ y\}$. We write it as $x \cup y = min\{x,\ y\}$.

$\cup:$ $\{F, T\} \times \{F, T\} \longrightarrow \{F,\ T\}$, such that for any $x, y \in \{F, T\}$,
$\cup(x, y) = max\{x,\ y\}$. We write it as $x \cup y = max\{x,\ y\}$.

$\Rightarrow$: $\{F, T\} \times \{F, T\} \longrightarrow \{F,\ T\}$, such that for any $x, y \in \{F, T\}$,
$\Rightarrow (x, y) = \cup(\neg x, y)$. We write it as $x \Rightarrow y = \neg x \cup y$.

$\Leftrightarrow$: $\{F, T\} \times \{F, T\} \longrightarrow \{F,\ T\}$, such that for any $x, y \in \{F, T\}$,
$\Leftrightarrow (x, y) = \cup(\Rightarrow (x, y), \Rightarrow (y, x))$.
We write it as $x \Leftrightarrow y = (x \Rightarrow y) \cap (y \Rightarrow x)$.

Exercise 6

Prove that the above connectives formulas are correct, i.e. that they define the same classical connectives as defined in **Step 1.**.

Solution
This is a problem of proving equality of functions that are given the same names. We have to show that the use of the same names: $\neg$, $\cup$, $\cap$, $\Rightarrow$, $\Leftrightarrow$ for them is justified. The equality of functions is defined as follows.

Definition 15

Given two sets A, B and functions f, g, such that $f:\ A \longrightarrow B$ *and* $g: A \longrightarrow B$. *We say that the functions f, g are* **equal** *and write it* $f = g$ *if and only if f(x) = g(x) for all elements* $x \in A$.

The negation definition is the same in both cases. We prove that the two conjunctions and two disjunctions functions are the equal by comparing he Truth Tables build for both definitions. We verify now the correctness of the implication function formula. Consider two functions $\Rightarrow$: $\{T, F\} \times \{T, F\} \longrightarrow \{T, F\}$ and $h : \{T, F\} \times \{T, F\} \longrightarrow \{T, F\}$, where $\Rightarrow$ is the classical implication defined by Definition **??** and h is defined by the formula $h(x, y) = \cup(\neg x, y)$. Observe that we have already proved that functions $\cup$ and $\neg$ are equal in both cases. We prove that $\Rightarrow\ =\ h$ by evaluating that $\Rightarrow (x, y) = h(x, y) = \cup(\neg x, y)$, for all $(x, y) \in \{T, F\} \times \{T, F\}$ as as follows.
$T \Rightarrow T = T$ and $h(T, T) = \neg T \cup T = F \cup T = T$ yes.
$T \Rightarrow F = F$ and $h(T, F) = \neg T \cup F = F \cup F = F$ yes.
$F \Rightarrow F = T$ and $h(F, F) = \neg F \cup F = T \cup F = T$ yes.
$F \Rightarrow T = T$ and $h(F, T) = \neg F \cup T = T \cup T = T$ yes.
This proves the correctness of the implication formula $\Rightarrow (x, y) = \cup(\neg x, y)$. We write it as $x \Rightarrow y = \neg x \cup y$ and call it a formula defining implication in terms of disjunction and negation. We verify the correctness of the equivalence formula $\Leftrightarrow (x, y) = \cup(\Rightarrow (x, y), \Rightarrow (y, x))$ in a similar way.

Special Properties of Connectives

Observe that the formulas defining connectives of implication and equivalence are certain compositions of previously defined connectives. Classical semantics is a special one, its connectives have strong properties that often do not hold under other semantics, extensional or not. One of them is a property of *definability of connectives*, the other one is a *functional dependency.* These are basic properties one asks about any new semantics, and hence a logic, being created. We generalize these the notion of *functional dependency* of connectives under a given extensional semantics $\mathbf{M}$.

Definition 16 (Definability of Connectives)

Given a propositional language $\mathcal{L}_{CON}$ *and its extensional semantics* $\mathbf{M}$*. A connective* $\circ \in CON$ *is* **definable** *in terms of some connectives* $\circ_1, \circ_2, ...\circ_n \in CON$ *if and only if* $\circ$ *is a certain function composition of functions* $\circ_1, \circ_2, ...\circ_n$*, as they are defined by the semantics* $\mathbf{M}$*.*

We have just proved in Exercise 6 that the implication $\Rightarrow$ is definable in terms of $\cup$ and $\neg$ under classical semantics as it is a composition of $\cup$ and $\neg$ defined by the formula $\Rightarrow (x, y) = \cup(\neg x, y)$. The classical equivalence is definable in terms of $\Rightarrow$ and $\cap$ by the formula $\Leftrightarrow (x, y) = \cup(\Rightarrow (x, y), \Rightarrow (y, x))$.

Definition 17 (Functional Dependency)

Given a propositional language $\mathcal{L}_{CON}$ *and its extensional semantics* $\mathbf{M}$*. A property of defining the set of connectives CON in terms of its proper subset is called* **a functional dependency** *of connectives under the semantics* $\mathbf{M}$*.*

Proving the property of functional dependency under a given semantics $\mathbf{M}$ consists of identifying a proper subset CON_0 of the set CON of connectives, such that each connective $\circ \in CON - CON_0$ is definable (definition 16) in terms of connectives from CON_0. This is usually a difficult, and often impossible task for many semantic. We prove now that it holds in the classical case.

Theorem 1

The set of connectives of the language $\mathcal{L}_{\{\neg,\ \cup,\ \cap,\ \Rightarrow,\ \Leftrightarrow\}}$ *is functionally dependent under the classical semantics.*

Proof

Let's take a set $\{\neg,\ \cup\}$. We have already proved in Exercise 6 that the implication $\Rightarrow$ and is definable in terms of $\cup$ and $\neg$ by the formula $x \Rightarrow y = \neg x \cup y$. The conjunction is defined by easy verification, similar to the one in Exercise 6, by a formula $x \cap y = \neg(\neg x \cup \neg y)$. By Exercise 6, the equivalence formula is definable in terms of $\Rightarrow$ and $\cap$ by the formula $x \Leftrightarrow y = (x \Rightarrow y) \cap (y \Rightarrow x)$. The final formula for for the equivalence is $x \Leftrightarrow y = (\neg x \cup y) \cap (\neg y \cup x)$.

There are many ways to prove this theorem, it means there are many ways to choose a proper subset CON_0 of the set $\{\neg, \cup, \cap, \Rightarrow, \Leftrightarrow\}$ that defines all other connectives. Here are the choices.

Theorem 2 (Definability of Connectives)

All connectives of the language $\mathcal{L}_{\{\neg, \cup, \cap, \Rightarrow, \Leftrightarrow\}}$ *are definable in terms of* $\neg$ *and* $\circ$*, for any* $\circ \in \{\cup, \cap, \Rightarrow\}$.

Proof
We list all required definability formulas, including the formulas developed in the proof of Theorem 1. An easy verification of their correctness is left as an exercise.

1. Definability in terms of $\Rightarrow$ and $\neg$.
$x \cap y = \neg(x \Rightarrow \neg y)$, $x \cup y = \neg x \Rightarrow y$, $x \Leftrightarrow y = \neg((x \Rightarrow y) \Rightarrow \neg(y \Rightarrow x))$.

2. Definability in terms of $\cap$ and $\neg$.
$x \cup y = \neg(\neg x \cap \neg y)$, $x \Rightarrow y = \neg(x \cap \neg y)$, $x \Leftrightarrow y = \neg(x \cap \neg y) \cap \neg(y \cap \neg x)$.

3. Definability in terms of $\cup$ and $\neg$.
$x \Rightarrow y = \neg x \cup y$, $x \cap y = \neg(\neg x \cup \neg y)$ $x \Leftrightarrow y = (\neg x \cup y) \cap (\neg y \cup x)$.

There are two other important classical binary connectives denoted by $\uparrow$ and $\downarrow$. The connective $\uparrow$ was discovered in 1913 by H.M. Sheffer, who called it *alternative negation*. Now it is often called a *Sheffer's connective*. A formula $(A \uparrow B)$ reads: *not both A and B*. The connective $\downarrow$ was discovered in 1920 by J. Łukasiewicz and named *joint negation*. The formula $(A \downarrow B)$ reads: *neither A nor B*. They are defined as follows.

Alternative Negation is a function $\uparrow$: $\{T, F\} \times \{T, F\} \longrightarrow \{T, F\}$ such that
$T \uparrow T = F$, $T \uparrow F = T$, $F \uparrow T = T$, $F \uparrow F = T$.

Joint Negation is a function $\downarrow$: $\{T, F\} \times \{T, F\} \longrightarrow \{T, F\}$ such that
$T \downarrow T = F$, $T \downarrow F = F$, $F \downarrow T = F$, $F \downarrow F = T$.

Truth Tables for $\uparrow$ and $\downarrow$

$\uparrow$	T	F
T	F	T
F	T	T

$\downarrow$	T	F
T	F	F
F	F	T

We extend our language $\mathcal{L}_{\{\neg, \cup, \cap, \Rightarrow, \Leftrightarrow\}}$ by adding Sheffer and Łukasiewicz connectives to it. We obtain the language $\mathcal{L}_{\{\neg, \cup, \cap, \Rightarrow, \Leftrightarrow, \uparrow, \downarrow\}}$ that contains now all possible classical connectives.

Theorem 3

All connectives of a language $\mathcal{L}_{\{\neg, \cup, \cap, \Rightarrow, \Leftrightarrow, \uparrow, \downarrow\}}$ *are definable in terms of* $\uparrow$*, and also separately in terms of* $\downarrow$.

Proof
Definability formulas of $\neg$ and $\cap$ in terms of $\uparrow$ are the following.

$$\neg x = x \uparrow x, \quad x \cap y = (x \uparrow y) \uparrow (x \uparrow y) \tag{4}$$

Definability formulas for of the connectives $\{\cup,\ \Rightarrow,\ \Leftrightarrow\}$ in terms of $\uparrow$ follow directly from the formulas in the proof of Theorem 2 and the formulas (4). Observe that the $x \uparrow y = \neg(x \cup y)$. The definability of $x \downarrow y$ in terms of $x \uparrow y$ follows from (4) and definability of $\cup$ in terms $\uparrow$.
Definability formulas of $\neg$ and $\cup$ in terms of $\downarrow$ are, by simple verification, the following.

$$\neg x = x \downarrow x, \quad x \cup y = (x \downarrow y) \downarrow (x \downarrow y) \tag{5}$$

Definability formulas for of the connectives $\{\cap,\ \Rightarrow,\ \Leftrightarrow, \uparrow\}$ in terms of $\downarrow$ follow directly, as in the previous case, from the Theorem 2 and the formulas (5).

Functional dependency and definability of connectives as expressed in Theorems 2, 3 are very strong and characteristic properties of the classical semantics. They hold, for some connectives for some non-classical logics, never in others. For example, the necessity connective $\Box$ is definable in terms of the possibility connectives $\Diamond$ and negation $\neg$ in Modal S4 and S5 logics, but not in majority of others. The classical implication is definable in terms of negation and disjunction, but the intuitionistic implication is not. We defined and discussed these classical properties here as they have to be addressed and examined when one is building semantics for any of a non-classical logic.

Step 2: Truth Assignment, Truth Extension

We define now and examine the components in Step 2 of the definition 7. We start with the basic notion of the *truth assignment*. We adopt the extensional semantics **M** definition 8 to the classical case as follows.

Definition 18 (Truth Assignment)

Let VAR be the set of all propositional variables of the language $\mathcal{L}_{\{\neg,\ \cup,\ \cap,\ \Rightarrow,\ \Leftrightarrow\}}$.
A **truth assignment** *is any function* $v : VAR \longrightarrow \{T, F\}$.

The function v defined above is called the truth assignment because it can be thought as an assignment to each variable (which represents a logical sentence) its logical value of T(ruth) of F(alse). Observe that the domain of the truth assignment is the set of propositional variables, i.e. the truth assignment is defined only for *atomic formulas*.

We now extend the truth assignment v from the set of *atomic formulas* to the set of all formulas $\mathcal{F}$ in order define formally the assignment of a logical value to *any formula* $A \in \mathcal{F}$.

The definition of the *truth extension* of the truth assignment v to the set $\mathcal{F}$ follows the definition 8 for the extensional semantics $\mathbf{M}$.

Definition 19 (Truth Extension) *Given the truth assignment* $v : VAR \longrightarrow \{T, F\}$. *We define its extension* v^* *to the set* $\mathcal{F}$ *of all formulas as any function* $v^* : \mathcal{F} \longrightarrow \{T, F\}$, *such that the following conditions are satisfied.*

(1) *for any* $a \in VAR$, $\quad v^*(a) = v(a)$;

(2) *for any* $A, B \in \mathcal{F}$,

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

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

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

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

$$v^*((A \Leftrightarrow B)) = \Leftrightarrow (v^*(A), v^*(B)).$$

The symbols on the *left-hand side* of the equations represent the connectives in their *natural language meaning.* The symbols on the *right-hand side* represent connectives in their *classical semantics* meaning defined by the classical connectives defined by the classical Truth Tables.

For binary connectives (two argument functions) we adopt a convention to write the symbol of the connective (name of the 2 argument function) between its arguments as we do in a case arithmetic operations. We use this standard notation and re-write the definition 19 as follows.

Definition 20 (Standard Notation) *Given the truth assignment* $v : VAR \longrightarrow \{T, F\}$. *We define its extension* v^* *to the set* $\mathcal{F}$ *of all formulas as any function* $v^* : \mathcal{F} \longrightarrow \{T, F\}$, *such that the following conditions are satisfied.*

(1) *for any* $a \in VAR$, $\quad v^*(a) = v(a)$;

(2) *for any* $A, B \in \mathcal{F}$,

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

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

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

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

$$v^*((A \Leftrightarrow B)) = v^*(A) \Leftrightarrow v^*(B).$$

Given a formula A: $((a \Rightarrow b) \cup \neg a))$ and a truth assignment v, such that $v(a) = T$, $v(b) = F$. We evaluate the logical value of the formula A using the standard notation definition 20 as follows.
$v^*(A) = v^*(((a \Rightarrow b) \cup \neg a))) = \cup(v^*((a \Rightarrow b), v^*(\neg a)) = \cup(\Rightarrow (v^*(a), v^*(b)), \neg v^*(a)))$
$= \cup(\Rightarrow (v(a), v(b)), \neg v(a))) = \cup(\Rightarrow (T, F), \neg T)) = \cup(F, F) = F$.

Observe that we did not specify $v(x)$ of any $x \in VAR - \{a, b\}$, as these values do not influence the computation of the logical value of the formula A. We say: "v such that" as we consider its values for the variables a and b only. Nevertheless, the domain of the truth assignment v is *always* is the set of all variables VAR and we have to *remember* that.

Given a formula A: $((a \Rightarrow b) \cup \neg a))$ and a truth assignment v, such that $v(a) = F$, $v(b) = F$. We use now the standard notation definition 20 to evaluate the logical value of the formula A. We write is as follows. $v^*(A) = v^*(((a \Rightarrow b) \cup \neg a))) = v^*((a \Rightarrow b)) \cup v^*(\neg a) = (v(a) \Rightarrow v(b)) \cup \neg v(a) = (F \Rightarrow F) \cup \neg F = T \cup T = T$.

Step 3: Satisfaction, Model, Counter-Model

We define now and examine the components in Step 3 of the definition 7. We adopt the extensional semantics **M** definitions 10, 11, and 12 to the classical case as follows.

Definition 21 (Satisfaction)

Let $v : VAR \longrightarrow \{T, F\}$.
We say that v **satisfies** *a formula $A \in \mathcal{F}$ if and only if $v^*(A) = T$. We denote it by $v \models A$.*

v **does not satisfy** *a formula $A \in \mathcal{F}$ if and only if $v^*(A) \neq T$. We denote it by $v \not\models A$.*

The relation $\models$ is often called a **satisfaction relation**. Observe, that in the classical semantics we have that $v^*(A) \neq T$ if and only if $v^*(A) = F$. In this case we say that v **falsifies** a formula A.

Exercise 7

Let A be a formula $((a \Rightarrow b) \cup \neg a))$ and v be a truth assignment $v : VAR \longrightarrow \{T, F\}$, such that $v(a) = T$, $v(b) = F$, and $v(x) = F$ for all $x \in VAR - \{a, b\}$. Show that $v \not\models ((a \Rightarrow b) \cup \neg a))$.

Proof We evaluate the logical value of the formula A as follows: $v^*(A) =$

$v^*((a \Rightarrow b) \cup \neg a)) = (v^*(a \Rightarrow b) \cup v^*(\neg a)) = ((v(a) \Rightarrow v(b)) \cup \neg v(a)) = ((T \Rightarrow F) \cup \neg T) = (F \cup F) = F$. It proves tha $v \not\models ((a \Rightarrow b) \cup \neg a))$ and we say that v falsifies the formula A.

As we remarked before, in practical cases we use a short-hand notation for while evaluating the logical value of a given formula. Here is a short procedure for any v and A. We use show it how it works for v and A from the exercise 7.

Short-hand Evaluation
Given any formula $A \in \mathcal{F}$ and any truth assignment $v : VAR \longrightarrow \{T, F\}$.
1. We write the value of v only for variables appearing in the formula in A.
In our case we write: $a = T$, $b = F$ for $v(a) = T, v(b) = F$.
2. Replace all variabes in A by their respective logical values.
In our case we replace a by T and b by F in the formula A $((a \Rightarrow b) \cup \neg a))$. We get an equation $((T \Rightarrow F) \cup \neg T)$.
3. Use use the connectives definition, in this case the definitionTTables to evaluate the logical value of the equation obtained in the step **2**.
In our case we evaluate $((T \Rightarrow F) \cup \neg T) = (F \cup F) = F$.
4 Write your answer in one of the forms: $v \models A$, $v \not\models A$ or "v satisfies A", " v falsifies A"
In our case v *falsifies* A and write $v \not\models ((a \Rightarrow b) \cup \neg a))$.

Example 11

Let A be a formula $((a \cap \neg b) \cup \neg c)$ and v be a truth assignment $v : VAR \longrightarrow \{T, F\}$, such that $v(a) = T$, $v(b) = F$, $v(c) = T$, and $v(x) = T$ for all $x \in VAR - \{a, b, c\}$. Using the **the short-hand** *notation we get $((T \cap \neg F) \cup \neg T) = ((T \cap T) \cup F) = (T \cup F) = T$. It proves that v satisfies the formula A and we write $v \models ((a \cap \neg b) \cup \neg c)$.*

Definition 22 (Model, Counter Model)

Given a formula $A \in \mathcal{F}$.

Any $v : VAR \longrightarrow \{T,\ F\}$, such that $v \models A$ is called a **model** *for A .*

Any v, such that $v \not\models A$ is called a **counter model** *for the formula A .*

The truth assignment from the Example 11 is a model for the formula $((a \cap \neg b) \cup \neg c)$ and the truth assignment from the Exercise 7 is a counter-model for the formula $((a \Rightarrow b) \cup \neg a))$.

Step 4: Classical Tautology Definition

There are two equivalent ways to define the notion of classical tautology. We will use them interchangeably. The first uses the notion of truth assignment and states the following.

Definition 23 (Tautology 1) *For any formula $A \in \mathcal{F}$,*
A is **tautology** *if and only if* $v^*(A) = T$ *for all* $v : VAR \longrightarrow \{T, F\}$.

The second uses the notion of satisfaction and model and the fact that in any extensional semantic the notions *" v satisfies A"* and *"v is a model for A"* are interchangeable. It is stated as follows.

Definition 24 (Tautology 2)

For any formula $A \in \mathcal{F}$,
A is **tautology** *if and only if* $v \models A$ *for all* $v : VAR \longrightarrow \{T, F\}$*, i.e. when all truth assignments are* **models** *for A.*

We write symbolically

$$\models A$$

for the statement "A is a **tautology**".

Remark 1

We use the symbol $\models A$ only for classical tautology. For all other extensional semantics $\mathbf{M}$ *we must use the symbol* $\models_{\mathbf{M}} A$ *and say " A is a tautology under a semantics* $\mathbf{M}$*, or to say in short "A is a* $\mathbf{M}$ *semantics tautology".*

We usually use the definition 24 to express that a formula in not a tautology, i.e. we say that a formula is not a tautology if it has a counter model. To stress it we put it in a form of a following definition.

Definition 25

For any formula $A \in \mathcal{F}$,
A is **not a tautology** *if and only if A has a* **counter model***;*
i.e. when there is a truth assignment $v : VAR \longrightarrow \{T, F\}$*, such that* $v \not\models A$.

We denote the statement "A **is not a tautology**" symbolically by

$$\not\models A.$$

A formula $A:\ ((a \Rightarrow b) \cup \neg a))$ is not a tautology ($\not\models\ ((a \Rightarrow b) \cup \neg a))$). A truth assignment $v : VAR \longrightarrow \{T, F\}$, such that $v(a) = T$, $v(b) = F$, and $v(x) = F$ for all $x \in VAR - \{a, b\}$ is a counter model for A, as we proved Exercise 7.

This ends the formal definition of classical semantics that follows the pattern for extensional semantics established in the definition 7.

3.1 Tautologies: Decidability and Verification Methods

There is a large number of basic and important tautologies listed and discussed in Chapter 2. We assume that the reader is familiar, or will familiarize with them when needed. We will refer to them and use them within our book. Chapter 2 also provides the motivation for classical approach to definition of tautologies as ways of describing correct rules of our mathematical reasoning. It also contains an informal definition of classical semantics and discusses a tautology verification method. We have just defined formally the classical semantics. Our goal now is to prove formally that the notion of classical tautology is decidable (Theorem 9) and to prove correctness of the tautology verification method presented in Chapter 2. Moreover we present here other basic tautology verification methods and prove their correctness.

We start now a natural question. How do we verify whether a given formula $A \in \mathcal{F}$ is or not is not a tautology. The answer seems to be very simple. By definition 23 we examine *all* truth assignments $v : VAR \longrightarrow \{T, F\}$. If they all evaluate in T, we proved that $\models A$. It at least one of them evaluate to F, we found a counter model and proved $\not\models A$. The verification process is decidable, if the we have a finite number of v to consider. So now all we have to do is to *count* how many truth assignments there are, i.e. how many there are functions that map the set VAR of propositional variables into the set $\{T, F\}$ of logical values. In order to do so we need to introduce some standard notations and some known facts. For a given set X, we denote by $|X|$ the *cardinality* of X. In a case of a finite set, it is called a number of elements of the set. We write $|X| = n$ to denote that X *has* n *elements*, for $n \in N$. We have a special names and notations to denote the *cardinalities* of infinite set. In particular we write $|X| = \aleph_0$ and say " cardinality of X is aleph zero," for any countably infinite set X. We write $|X| = \mathcal{C}$ and say " cardinality of X is continuum" for any uncountable set X that has the same cardinality as Real numbers.

Theorem 4 (Counting Functions)

For any sets X, Y, *there are* $|Y|^{|X|}$ *functions that map the set* X *into* Y.

In particular, when the set X *is countably infinite and the set* Y *is finite, then there are* $n^{\aleph_0} = \mathcal{C}$ *functions that map the set* X *into* Y.

In our case of counting the truth assignment $v : VAR \longrightarrow \{T, F\}$ we have that $|VAR| = \aleph_0$ and $|\{T, F\}| = 2$. We know that $2^{\aleph_0} = \mathcal{C}$ and hence we get directly from Theorem 4 the following.

Theorem 5 (Counting Truth Assignments)

There are uncountably many (exactly as many as real numbers) of all possible truth assignments v, *where* $v : VAR \longrightarrow \{T, F\}$.

Fortunately for us, we are going to prove now that in order to decide whether a given formula $A \in \mathcal{F}$ is, or is not a tautology it is enough consider only a *finite number* of special truth assignments, not the uncountably many of them as required by the tautology definition 23, i.e. we are going to prove the Tautology Decidability Theorem 9. In order to be able to do so we need to introduce some new notions and definitions.

Definition 26

For any $A \in \mathcal{F}$, let VAR_A be a set of all propositional variables appearing in A. Any function $v_A : VAR_A \longrightarrow \{T, F\}$, is called a **truth assignment restricted** *to A.*

Example 12

Let $A = ((a \Rightarrow \neg b) \cup \neg c)$. The set of variables appearing in A is $VAR_A = \{a, b, c\}$. The truth assignment **restricted** *to A is any function $v_A : \{a, b, c\} \longrightarrow \{T, F\}$.*

Definition 27

Given a formula $A \in \mathcal{F}$ and a set VAR_A of all propositional variables appearing in A. Any function $v_A : VAR_A \longrightarrow \{T, F\}$, such that $v \models A$ ($v \not\models A$) is called a **restricted model (counter model)** *for A.*

We use the following particular case of Theorem 4 to count, for any formula A, possible truth assignment restricted to A, i.e. all possible restricted models and counter models for A.

Theorem 6 (Counting Functions 1)

For any finite sets X and Y, if X has n elements and Y has m elements, then there are m^n possible functions that map X into Y.

We also can prove it independently by Mathematical Induction over m.

Given a formula $A \in \mathcal{F}$, the set VAR_A is always finite, and $|\{T, F\}| = 2$, so directly from Theorem 6 we get the following.

Theorem 7 (Counting Restricted Truth)

For any $A \in \mathcal{F}$, there are $2^{|VAR_A|}$ of possible truth assignments **restricted** *to A*

So there are $2^3 = 8$ possible truth assignment **restricted** to the formula $A = ((a \Rightarrow \neg b) \cup \neg c)$. We usually list them, and their value on the formula A in a form of an extended truth table below.

v_A	a	b	c	$v^*(A)$ *computation*	$v^*(A)$
v_1	T	T	T	$(T \Rightarrow T) \cup \neg T = T \cup F = T$	T
v_2	T	T	F	$(T \Rightarrow T) \cup \neg F = T \cup T = T$	T
v_3	T	F	F	$(T \Rightarrow F) \cup \neg F = F \cup T = T$	T
v_4	F	F	T	$(F \Rightarrow F) \cup \neg T = T \cup F = T$	T
v_5	F	T	T	$(F \Rightarrow T) \cup \neg T = T \cup F = T$	T
v_6	F	T	F	$(F \Rightarrow T) \cup \neg F = T \cup T = T$	T
v_7	T	F	T	$(T \Rightarrow F) \cup \neg T = F \cup F = F$	F
v_8	F	F	F	$(F \Rightarrow F) \cup \neg F = T \cup T = T$	T

(6)

$v_1,\ _2,\ v_3,\ v_4,\ v_5,\ v_6,\ v_8$ are **restricted models** for A and v_7 is a **restricted counter model** for A.

Now we are ready to prove the *correctness* of the well known truth tables tautology verification method. We formulate it as the follows.

Theorem 8 (Truth Tables)

For any formula $A \in \mathcal{F}$,
$\models A$ *if and only if* $v_A \models A$ *for all* $v_A : VAR_A \longrightarrow \{T, F\}$, *i.e.*
$\models A$ *if and only if all* $v_A : VAR_A \longrightarrow \{T, F\}$ *are restricted models for* A.

Proof Assume $\models A$. By definition 24 we have that $v \models A$ for all $v : VAR \longrightarrow \{T, F\}$, hence $v_A \models A$ for all $v_A : VAR_A \longrightarrow \{T, F\}$ as $VAR_A \subseteq VAR$.

Assume $v_A \models A$ for all $v_A : VAR_A \longrightarrow \{T, F\}$. Take any $v : VAR \longrightarrow \{T, F\}$, as $VAR_A \subseteq VAR$, any $v : VAR \longrightarrow \{T, F\}$ is an *extersion* of some v_A, i.e. $v(a) = v_A(a)$ for all $a \in VAR_A$. By Truth Extension Definition 19 we get that $v^*(A) = {v_A}^*(A) = T$ and $v \models A$. This ends the proof.

Directly from Theorem 7 and the above Theorem 8 we get the proof of the correctness and decidability of the Truth Tables Method, and hence the decidability of the notion of classical propositional tautology.

Theorem 9 (Tautology Decidability)

For any formula $A \in \mathcal{F}$, *one has to examine at most* 2^{VAR_A} *restricted truth assignments* $v_A :\ VAR_A \longrightarrow \{F, T\}$ *in order to decide whether*

$$\models A \quad or \quad \not\models A,$$

i.e. the notion of classical tautology is decidable.

We present now and prove correctness of some basic tautologies verification methods. We just proved (Theorem 9) the correctness of the truth table tautology verification method, so we we start with it.

Truth Table Method

The verification method, called a **truth table method** consists of examination, for any formula A, all possible truth assignments restricted to A. By theorem 7 we have to perform at most $2^{|VAR_A|}$ steps. If we find a truth assignment which evaluates A to F, we stop the process and give answer: $\not\models A$. Otherwise we continue. If all truth assignments evaluate A to T, we give answer: $\models A$.

We usually list all restricted truth assignments v_A in a form of a truth table similar to the table6, hence the name of the method.

Consider, for example, a formula A: $(a \Rightarrow (a \cup b))$. There are $2^2 = 4$ possible truth assignment restricted to A. We usually list them, and evaluate their value on the formula A in a form of an extended truth table as follows.

$$\begin{array}{c|cc|c|c} w & a & b & w^*(A)\ computation & w^*(A) \\ \hline w_1 & T & T & T \Rightarrow (T \cup T) = T \Rightarrow T = T & T \\ w_2 & T & F & T \Rightarrow (T \cup F) = T \Rightarrow T = T & T \\ w_3 & F & T & F \Rightarrow (F \cup T) = F \Rightarrow T = T & T \\ w_4 & F & F & F \Rightarrow (F \cup F) = F \Rightarrow F = T & T \end{array} \tag{7}$$

The table (7) shows that all $w : VAR_A \longrightarrow \{T, F\}$ are restricted models for A and hence by Theorem 9 we proved that $\models (a \Rightarrow (a \cup b))$ and $\mathbf{T} \neq \emptyset$.
Observe that the table (7) proves that the formula $\not\models \ ((a \Rightarrow \neg b) \cup \neg c)$.

Moreover we have proved that the condition (3) of the definition 14 is fulfilled and the classical semantics is well defined. We put it as a separate statement.

Fact 2

The classical semantics is **well defined**.

The complexity of the truth table methods grows exponentially. Impossible for humans to handle formulas with more then few variables, and cumbersome for computers for formulas with a great number of variables, In practice, if we need, we use often much shorter and more elegant tautology verification methods presented below.

Proof by Contradiction Method

In this method, in order to prove that $\models A$ we proceed as follows.

We assume that $\not\models A$. We work with this assumption. If we get a **contradiction**, we have proved that $\not\models A$ is impossible. We hence proved $\models A$. If we do not get a contradiction, it means that the assumption $\not\models A$ is true, i.e. we have proved that A is not a tautology.

Exercise 8

Follow the Proof by Contradiction Method and examine whether $\models (a \Rightarrow (a \cup b))$.

Solution
We use a short-hand notation.
Assume that $\not\models\ (a \Rightarrow (a \cup b))$. It means that $(a \Rightarrow (a \cup b)) = F$ for some truth assignment v. By definition of implication $\Rightarrow$ we have that
$(a \Rightarrow (a \cup b)) = F$ if and only if $a = T$ and $(a \cup b) = F$.

From $a = T$ and $(a \cup b) = F$ we get $(T \cup b) = F$. This is a **contradiction** with the definition of disjunction $\cup$. Hence we proved $\models\ (a \Rightarrow (a \cup b))$.

Exercise 9

Use the Proof by Contradiction Method to decide whether $\models\ ((a \cup b) \Rightarrow a)$.

Solution We do not use short-hand notation.

Assume that $\not\models\ ((a \cup b) \Rightarrow a)$. It means that there is $v : VAR \longrightarrow \{T, F\}$, such that $v^*(((a \cup b) \Rightarrow a)) = F$. We evaluate, $v^*(((a \cup b) \Rightarrow a))) = v^*((a \cup b)) \Rightarrow v(a)$ and we get that the truth assignment v is such that $v^*((a \cup b)) \Rightarrow v(a) = F$. By definition implication $\Rightarrow$ we have that $v^*((a \cup b)) \Rightarrow v(a) = F$ if and only if $v(a) \cup v(b) = T$ and $(a) = F$. From $(a) = F$ and $v(a) \cup v(b) = T$ we get that $F \cup v(b) = T$. This is **possible** for any $v : VAR \longrightarrow \{T, F\}$, such that $v(b) = T$. This proves that any truth assignment $v : VAR \longrightarrow \{T, F\}$, such that $(a) = F, v(b) = T$ is a counter model for $((a \cup b) \Rightarrow a)$, i.e. that $\not\models\ ((a \cup b) \Rightarrow a)$.

Substitution Method

We define and prove the correctness of a method, called Substitution Method that allows us to obtain new tautologies from formulas already proven to be tautologies.

We can use the same reasoning as we used in the solution to the Exercise 8 that proved $\models\ (a \Rightarrow (a \cup b))$ to prove that, for example the formulas

$$((((a \Rightarrow b) \cap \neg c) \Rightarrow ((((a \Rightarrow b) \cap \neg c) \cup \neg d)) \tag{8}$$

$$((a \Rightarrow b) \cap \neg c) \cup d) \cap \neg e) \Rightarrow (((a \Rightarrow b) \cap \neg c) \cup d) \cap \neg e) \cup ((a \Rightarrow \neg e))) \tag{9}$$

are also a tautologies.

Instead of repeating the same argument from Exercise 8 for a much more complicated formulas we make a simple observation that we can obtain (8), (9) from the formula $(a \Rightarrow (a \cup b))$ by a proper *substitutions* (replacements) of more complicated formulas for the variables a and b in $(a \Rightarrow (a \cup b))$. We use a notation $A(a, b) = (a \Rightarrow (a \cup b))$ to denote that $(a \Rightarrow (a \cup b))$ is a formula A with two variables a, b and we denote by

$$A(a/A_1,\ b/A_2)$$

a result of a substitution (replacement) of formula $A_1,\ A_2$ on a place of the variables a, b, respectively, everywhere where they appear in $A(a, b)$.
Theorem 10 we are going to prove states that *substitutions* lead always from a tautology to a tautology. In particular, making the following substitutions s1 and s2 in $A(a, b) = (a \Rightarrow (a \cup b))$ we get, that the respective formulas (8), (9) are tautologies.

By substitution s1: $A(a/((a \Rightarrow b) \cap \neg c),\ b/\neg d)$ we get that

$$\models ((((a \Rightarrow b) \cap \neg c) \Rightarrow ((((a \Rightarrow b) \cap \neg c) \cup \neg d)).$$

By substitution s2: $A(a/((a \Rightarrow b) \cap \neg c),\ b/((a \Rightarrow \neg e))$ we get that
$\models (((a \Rightarrow b) \cap \neg c) \cup d) \cap \neg e) \Rightarrow (((a \Rightarrow b) \cap \neg c) \cup d) \cap \neg e) \cup ((a \Rightarrow \neg e))).$

The theorem 10 describes validity of a method of constructing new tautologies from given tautologies. In order to formulate and prove it we first introduce needed notations.

Let $A \in \mathcal{F}$ be a formula and $VAR_A = \{a_1, a_2, ...a_n\}$ be the set of all propositional variables appearing in A. We will denote it by $A(a_1, a_2, ...a_n)$.

Given a formula $A(a_1, a_2, ...a_n)$, and $A_1, ...A_n$ be any formulas. We denote by

$$A(a_1/A_1, ..., a_n/A_n)$$

the result of simultaneous replacement (substitution) in $A(a_1, a_2, ...a_n)$ variables $a_1, a_2, ...a_n$ by formulas $A_1, ...A_n$, respectively.

Theorem 10

For any formulas $A(a_1, a_2, ...a_n),\ A_1,\ \dots\ ,\ A_n \in \mathcal{F}$,

If $\models A(a_1, a_2, ...a_n)$ *and* $B = A(a_1/A_1, ..., a_n/A_n)$, *then* $\models B$.

Proof. Let $B = A(a_1/A_1, ..., a_n/A_n)$. Let $b_1, b_2, ...b_m$ be all those propositional variables which occur in $A_1, ...A_n$. Given a truth assignment $v : VAR \longrightarrow \{T, F\}$, any values $v(b_1), v(b_2), ...v(b_m)$ defines the logical value of $A_1, ...A_n$, i.e. $v^*(A_1), ...v^*(A_n)$ and, in turn, $v^*(B)$.

Let $w : VAR \longrightarrow \{T, F\}$ be a truth assignment such that $w(a_1) = v^*(A_1), w(a_2) = v^*(A_2), ...w(a_n) = v^*(A_n)$. Obviously, $v^*(B) = w^*(A)$. Since A is a propositional tautology, $w^*(A) = T$, for all possible w, hence $v^*(B) = w^*(A) = T$ for all truth assignments w and B is also a tautology.

We have proved (Exercise 8) that the formula $D(a, b) = (a \Rightarrow (a \cup b))$ is a tautology. By the above Theorem 10 we get that $D(a/A,\ b/B) = ((A \cup B) \Rightarrow A)$ is a tautology. We hence get the following.

Fact 3

For any $A, B \in \mathcal{F}$, $\models ((A \cup B) \Rightarrow A)$.

Generalization Method

Now let's look at the task of finding whether the formulas (8), (9) are tautologies from yet another perspective. This time we observe that both of them are build in a similar way as a formula $(A \Rightarrow (A \cup B))$, for $A = ((a \Rightarrow b) \cap \neg c),\ B = \neg d$ in (8) and for $A = ((a \Rightarrow b) \cap \neg c),\ B = ((a \Rightarrow \neg e))$ in (9).

It means we represent, if it is possible, a given formula as a particular case of some much simpler *general formula*. Hence the name Generalization Method. We then use Proof by Contradiction Method or Substitution Method to examine whether the given formula is /is not a tautology.

In this case, we prove, for example Proof by Contradiction Method by that $\models (A \Rightarrow (A \cup B))$, for any formulas $A, B \in \mathcal{F}$ and get, as a particular cases for A, B that that both formulas (8), (9) are tautologies.

Let's assume that there are formulas $A, B \in \mathcal{F} \not\models (A \Rightarrow (A \cup B))$. This means that $(A \Rightarrow (A \cup B)) = F$ for some truth assignment v. This holds only when $A = T$ and $(A \cup B) = F$, i.e. $(T \cup B) = F$. This is a contradiction with the definition of $\cup$. So $\models (A \Rightarrow (A \cup B))$ for all $A, B \in \mathcal{F}$.

Exercise 10

Show that $v \models (\neg((a \cap \neg b) \Rightarrow ((c \Rightarrow (\neg f \cup d)) \cup e)) \Rightarrow ((a \cap \neg b) \cap (\neg(c \Rightarrow (\neg f \cup d)) \cap \neg e)))$, *for all* $v : VAR \longrightarrow \{T, F\}$.

Solution

Observe that we really have to prove that $\models (\neg((a \cap \neg b) \Rightarrow ((c \Rightarrow (\neg f \cup d)) \cup e)) \Rightarrow ((a \cap \neg b) \cap (\neg(c \Rightarrow (\neg f \cup d)) \cap \neg e)))$. We can hence use any of our tautology verification methods. In this case $VAR_A = \{a, b, c, d, e, f\}$, so there are $2^6 = 64$ restricted truth assignments to consider. Much too many to apply the Truth Table Method. Our formula is also far too complicated to guess a simple tautology from which we could obtain it by the Substitution Method.

The Proof by Contradiction Method is less complicated, but before we apply it let's look closer at the sub-formulas of our formula and patterns they form

inside the formula it, i.e. we try to apply the Generalization Method first.
Let's put $B = (a \cap \neg b),\ C = (c \Rightarrow (\neg f \cup d)),\ D = e$. We re-write our formula in a general form as $(\neg(B \Rightarrow (C \cup D)) \Rightarrow (B \cap (\neg C \cap \neg D)))$ and prove that for all $B, C, D \in \mathcal{F}$,

$$\models (\neg(B \Rightarrow (C \cup D)) \Rightarrow (B \cap (\neg C \cap \neg D)).$$

We use Proof by Contradiction Method, i.e. we assume that there are formulas $B, C, D \in \mathcal{F}$, such that

$$\not\models (\neg(B \Rightarrow (C \cup D)) \Rightarrow (B \cap (\neg C \cap \neg D)).$$

This means that there is a truth assignment v, such that (we use short-hand notation) $(\neg(B \Rightarrow (C \cup D)) \Rightarrow (B \cap (\neg C \cap \neg D))) = F$. By definition of implication it is possible if and only if $\neg(B \Rightarrow (C \cup D)) = T$ and $(B \cap (\neg C \cap \neg D)) = F$, i.e. if and only if
$(B \Rightarrow (C \cup D)) = F$ and $(B \cap (\neg C \cap \neg D)) = F$ Observe that $(B \Rightarrow (C \cup D)) = F$ if and only if $B = T,\ C = F,\ D = F$. We now evaluate the logical value of $(B \cap (\neg C \cap \neg D))$ for $B = T, C = F, D = F$, i.e. we compute $(B \cap (\neg C \cap \neg D)) = (T \cap (\neg F \cap \neg F)) = (T \cap (T \cap T)) = T$. This **contradicts** that we must have $(B \cap (\neg C \cap \neg D)) = F$. This proves that for all $B, C, D \in \mathcal{F}$

$$\models (\neg(B \Rightarrow (C \cup D)) \Rightarrow (B \cap (\neg C \cap \neg D))),$$

and hence is holds for our particular case, i..e.

$$\models (\neg((a \cup b) \Rightarrow ((c \Rightarrow d) \cup e)) \Rightarrow ((a \cup b) \cap (\neg(c \Rightarrow d) \cap \neg e)))$$

and that all truth assignments are models for $(\neg((a \cup b) \Rightarrow ((c \Rightarrow d) \cup e)) \Rightarrow ((a \cup b) \cap (\neg(c \Rightarrow d) \cap \neg e)))$.

Sets of Formulas; Tautologies and Contradictions

We distinguish now special sets of formulas and examine their properties. We define sets of all tautologies, contradictions, consistent sets, inconsistent sets and discuss a notion of independence of formulas from sets of formulas.

Definition 28 (Set of Tautologies)

We denote by $\mathbf{T}$ *the set of all tautologies, i.e. we put*

$$\mathbf{T} = \{A \in \mathcal{F} : \quad \models A.\}$$

We distinguish now another type of formulas, called *contradictions*.

Definition 29 (Contradiction)

A formula $A \in \mathcal{F}$ *is called a contradiction if it does not have a model.*

We write symbolically $=\mid A$ for the statement "A is a **contradiction**."

Directly from the Definition 29 we have that

$$=\mid A \quad \text{if and only if} \quad v \not\models A \quad \text{for all} \quad v: VAR \longrightarrow \{T, F\}.$$

Example 13

The following formulas are contradictions

$$(a \cap \neg a), \quad (a \cap \neg(a \cup b)), \quad \neg(a \Rightarrow a), \quad \neg(\neg(a \cap b) \cup b)).$$

Definition 30 (Set of Contradictions)

We denote by **C** *the set of all tautologies, i.e. we put*

$$\mathbf{C} = \{A \in \mathcal{F} : \quad =\mid A.\}$$

Following the proof of Theorem 10 we get similar theorem for contradictions, and hence a a proof of correctness of the Substitution Method of constructing new contradictions.

Theorem 11

For any formulas $A(a_1, a_2, ...a_n)$, $A_1, \ \ldots, \ A_n \in \mathcal{F}$,

If $A(a_1, a_2, ...a_n) \in \mathbf{C}$ *and* $B = A(a_1/A_1, ..., a_n/A_n)$, *then* $B \in \mathbf{C}$.

Directly from the Theorem 11 we get the following.

Example 14 *For any formulas* $A, B \in \mathcal{F}$, *the following formulas are contradictions*

$$(A \cap \neg A), \quad (A \cap \neg(A \cup B)), \quad \neg(A \Rightarrow A), \quad \neg(\neg(A \cap B) \cup B)).$$

Observe, that there are formulas which neither in **T** nor in **C**, for example $(a \cup b)$. Any truth assignment v, such that $v(a) = F, v(b) = F$ falsifies $(a \cup b)$ and it proves that it is not a tautology. Any truth assignment v, such that $v(a) = T, v(b) = T$ satisfies $(a \cup b)$, what proves that it is not a contradiction.

3.2 Sets of Formulas: Consistency and Independence

Next important notions for any logic are notions of consistency, inconsistency of the sets of formulas and the independence of a formula from the set of formulas. We adopt the following definitions.

Definition 31

A truth truth assignment $v : VAR \longrightarrow \{T, F\}$ *is* **model** *for the set* $\mathcal{G} \subseteq \mathcal{F}$ *of formulas if and only if* $v \models A$ *for all formulas* $A \in \mathcal{G}$*. We denote it by*

$$v \models \mathcal{G}.$$

The restriction $v_{\mathcal{G}}$ *of the* **model** v *to the domain* $VAR_{\mathcal{G}} = \bigcup_{A \in \mathcal{G}} VAR_A$ *is called a* **restricted model** *for* $\mathcal{G}$*.*

Exercise 11

Find a model and a restricted model for a set

$$\mathcal{G} = \{((a \cap b) \Rightarrow b),\ (a \cup b), \neg a\}.$$

Solution

Let v be a truth assignment $v : VAR \longrightarrow \{T, F\}$. By the defininition 31, $v \models \{((a \cap b) \Rightarrow b),\ (a \cup b), \neg a\}$ if and only if $v^*(((a \cap b) \Rightarrow b)) = T,\ v^*((a \cup b) = T)$, and $v^*(\neg a) = T$. Observe that $\models ((a \cap b) \Rightarrow b)$, so we have to find v, such that $v^*((a \cup b)) = T,\ v^*(\neg a) = T$. This holds if and only if $v(a) = F$ and $F \cup v(b) = T$, i.e. if and only if $v(a) = F$ and $v(b) = T$. This proves that any v such that $v(a) = F$ and $v(b) = T$ is a model for $\mathcal{G}$, and $\mathcal{G}$ has only one restricted model. We put it as a separate fact.

Fact 4

Given $\mathcal{G} = \{((a \cap b) \Rightarrow b),\ (a \cup b), \neg a\}$*, we have that* $VAR_{\mathcal{G}} = \bigcup_{A \in \mathcal{G}} VAR_A = \{a, b\}$ *and* $v_{\mathcal{G}} : \{a, b\} \longrightarrow \{T, F\}$*, such that* $v_{\mathcal{G}}(a) = F$ *and* $v_{\mathcal{G}}(b) = T$ *is a unique* **restricted model** *for* $\mathcal{G}$*.*

Observation 4

For some sets $\mathcal{G} \subseteq \mathcal{F}$ *,* $VAR_{\mathcal{G}}$ *can be infinite. For example, for* $\mathcal{G} = VAR$ *we have that* $VAR_{\mathcal{G}} = VAR$ *and the notions of model and restricted model are the same.*

Definition 32

A set $\mathcal{G} \subseteq \mathcal{F}$ *is called* ***consistent*** *if and only if there is* $v : VAR \longrightarrow \{T, F\}$*, such that* $v \models \mathcal{G}$*.*

Otherwise the set $\mathcal{G}$ *is called* ***inconsistent****.*

Plainly speaking, a set $\mathcal{G}$ is consistent if it has a model, and is inconsistent if it does not have a model.

Example 15

The set $\mathcal{G}_1 = \{((a \cap b) \Rightarrow b),\ (a \cup b), \neg a\}$ *is* **consistent** *as* $v : VAR \longrightarrow \{T, F\}$, *such that* $v(a) = F$ *and* $v(b) = T$ *is the model for* $\mathcal{G}_1$.

The set $\mathcal{G}_2 = VAR$ *is also* **consistent**, *as* $v : VAR \longrightarrow \{T, F\}$, *such that* $v(a) = T$, *for all* $a \in VAR$ *is a model for* $\mathcal{G}_2$.

Observe that $\mathcal{G}_1$ is a finite consistent set. $\mathcal{G}_2$ is an infinite consistent set. This and other examples justify the need of truth assignment domain being the set VAR of all propositional variables.

Example 16

The set $\mathcal{G}_1 = \{((a \cap b) \Rightarrow b),\ (a \cap \neg a), \neg a\}$ *is a finite* **inconsistent** *set as it contains a formula* $(a \cap \neg a) \in \mathbf{C}$.

The set $\mathcal{G}_2 = VAR \cup \{\neg a\}$ *for some* $a \in VAR$, *is an infinite* **inconsistent** *set as it contains a certain variable* a *and its negation* $\neg a$.

Of course the most obvious example of an infinite consistent set is the set $\mathbf{T}$ of all tautologies, and of an infinite inconsistent consistent set is the set $\mathbf{C}$ of all contradictions.

Definition 33

A formula $A \in \mathcal{F}$ *is called* ***independent*** *from a set* $\mathcal{G} \subseteq \mathcal{F}$ *if and only if the sets* $\mathcal{G} \cup \{A\}$ *and* $\mathcal{G} \cup \{\neg A\}$ *are both* **consistent**. *I.e. when there are truth assignments* $v_1,\ v_2$ *such that*

$$v_1 \models \mathcal{G} \cup \{A\} \quad \text{and} \quad v_2 \models \mathcal{G} \cup \{\neg A\}.$$

Exercise 12

Show that a formula $A = ((a \Rightarrow b) \cap c)$ *is* ***independent*** *from the set* $\mathcal{G} = \{((a \cap b) \Rightarrow b),\ (a \cup b), \neg a\}$.

Solution

We define two truth assignments $v_1,\ v_2 : VAR \longrightarrow \{T, F\}$ such that $v_1 \models \mathcal{G} \cup \{(a \Rightarrow b) \cap c)\}$ and $v_2 \models \mathcal{G} \cup \{\neg(a \Rightarrow b) \cap c)\}$ as follows. We have just proved (Exercise 11) that any $v : VAR \longrightarrow \{T, F\}$, such that $v(a) = F$, $v(b) = T$ is a model for $\mathcal{G}$. Take as v_1 any truth assignment such that $v_1(a) = v(a) = F$, $v_1(b) = v(b) = T$, $v_1(c) = T$. We evaluate $v_1{}^*(A) = v_1{}^*(((a \Rightarrow b) \cap c)) = (F \Rightarrow T) \cap T = T$. This proves that $v_1 \models \mathcal{G} \cup \{A\}$. Take as v_2 any truth assignment such that, $v_2(a) = v(a) = F$, $v_2(b) = v(b) = T$, $v_2(c) = F$. We evaluate $v_2{}^*(\neg A) = v_2{}^*(\neg(((a \Rightarrow b) \cap c)) = T \cap T = T$. This proves that $v_2 \models \mathcal{G} \cup \{\neg A\}$. It ends the proof that formula A is **independent** from $\mathcal{G}$.

Exercise 13

Show that a formula $A = (\neg a \cap b)$ *is* ***not independent*** *from* $\mathcal{G} = \{((a \cap b) \Rightarrow b),\ (a \cup b), \neg a\}$.

Solution We have to show that it is impossible to construct v_1, v_2 such that $v_1 \models \mathcal{G} \cup \{A\}$ and $v_2 \models \mathcal{G} \cup \{\neg A\}$. From Fact 4 $\mathcal{G}$ has a unique restricted model $v : \{a, b\} \longrightarrow \{T, F\}$, such that $v(a) = F$, and $v(b) = T$. and $\{a, b\} = VAR_A$. So we have to check now if it is possible $v \models A$ and $v \models \neg A$. We evaluate $v^*(A) = v^*((\neg a \cap b) = \neg v(a) \cap v(b) = \neg F \cap T = T \cap T = T$ and get $v \models A$. By definition $v^*(\neg A) = \neg v^*(A) = \neg T = F$ and $v \not\models \neg A$. This end the proof that the formula $A = (\neg a \cap b)$ is **not independent** from $\mathcal{G}$.

Exercise 14

Given a set $\mathcal{G} = \{a, (a \Rightarrow b)\}$.
Find a formula A *that is* **independent** *from* $\mathcal{G}$.

Solution
Observe that truth assignment v such that $v(a) = T, v(b) = T$ is the only restricted model for $\mathcal{G}$. So we have to come up with a formula A such that there are two different truth assignments, v_1, v_2 such that $v_1 \models \mathcal{G} \cup \{A\}$ and $v_2 \models \mathcal{G} \cup \{\neg A\}$. Let's think about as simple a formula as it could be, namely let's consider $A = c$, where c any propositional variable (atomic formula) different from a and b. $\mathcal{G} \cup \{A\} = \{a, (a \Rightarrow b), c\}$ and any truth assignment v_1, such that $v_1(a) = T,\ v_1(b) = T, v_1(c) = T$ is a model for $\mathcal{G} \cup \{c\}$. Likewise for $\mathcal{G} \cup \{\neg c\} = \{a, (a \Rightarrow b), \neg c\}$. Any v_2 such that $v_2(a) = T,\ v_2(b) = T, v_2(c) = F$ is a model for $\mathcal{G} \cup \{\neg c\}$. This proves that we have found the formula $A = c$ that is **independent** from $\mathcal{G}$.

Here is a simple generalization of the Exercise 14.

Exercise 15

Find an infinite number of formulas that are **independent** *from* $\mathcal{G} = \{((a \cap b) \Rightarrow b),\ (a \cup b), \neg a\}$.

Solution
First we have to find all $v : VAR \longrightarrow \{T, F\}$ such that $v \models \{((a \cap b) \Rightarrow b),\ (a \cup b), \neg a\}$, i.e such that (shorthand notation) $((a \cap b) \Rightarrow b) = T,\ (a \cup b) = T,\ \neg a = T$. Observe that $\models ((a \cap b) \Rightarrow b)$, so we have to consider only $(a \cup b) = T,\ \neg a = T$. This holds if and only if $a = F$ and $(F \cup b) = T$, i.e. if and only if $a = F$ and $b = T$. This proves that that $v_{\mathcal{G}}$ such that $v_{\mathcal{G}}(a) = F$ and $v_{\mathcal{G}}(b) = T$ is the only one restricted model for $\mathcal{G}$. All possible models for $\mathcal{G}$ must be extensions of $v_{\mathcal{G}}$. We define a countably infinite set of formulas (and

their negations) and corresponding extensions v of $v_{\mathcal{G}}$ (restricted to to the set of variables $\{a, b\}$) such that $v \models \mathcal{G}$ as follows.

Observe that all extensions of v of $v_{\mathcal{G}}$ have as domain the infinitely countable set $VAR = \{a_1,\ a_2,\ \dots, a_n \dots\}$. We take as the infinite set of formulas in which every formula is to be proved independent of $\mathcal{G}$ the set of *atomic formulas*

$$\mathcal{F}_0 = VAR - \{a,\ b\} = \{a_1,\ a_2,\ \dots, a_n \dots\} - \{a,\ b\}.$$

Let $c \in \mathcal{F}_0$. We define truth assignments $v_1,\ v_2 : VAR \longrightarrow \{T, F\}$ as follows

$v_1(a) = v(a) = F,\ v_1(b) = v(b) = T$, and $v_1(c) = T$ for all $c \in \mathcal{F}_0$.

$v_2(a) = v(a) = F,\ v_2(b) = v(b) = T$, and $v_2(c) = F$ for all $c \in \mathcal{F}_0$.

Obviously, $v_1 \models \mathcal{G} \cup \{c\}$ and $v_2 \models \mathcal{G} \cup \{\neg c\}$ for all $c \in \mathcal{F}_0$. What proves that the set $\mathcal{F}_0$ is a countably infinite set of formulas **independent** from $\mathcal{G} = \{((a \cap b) \Rightarrow b),\ (a \cup b), \neg a\}$.

4 Classical Tautologies and Equivalence of Languages

We first present here a set of most widely used classical propositional tautologies which we will use, in one form or other, in our investigations in future chapters. Another extended list of tautologies and their discussion is presented in Chapter **??**.

As the next step we define notions of a *logical equivalence* and an *equivalence of languages*. We prove that all of the languages

$$\mathcal{L}_{\{\neg \Rightarrow\}},\ \mathcal{L}_{\{\neg \cap\}},\ \mathcal{L}_{\{\neg \cup\}},\ \mathcal{L}_{\{\neg, \cap, \cup, \Rightarrow\}},\ \mathcal{L}_{\{\neg, \cap, \cup, \Rightarrow, \Leftrightarrow\}},\ \mathcal{L}_{\{\uparrow\}},\ \mathcal{L}_{\{\downarrow\}}$$

are equivalent under classical semantics and hence can be used (and are) as different languages for classical propositional logic.

We generalize these notions to the case of any extensional semantics **M** in the next section 5. We also discuss and examine there some particular many valued extensional semantics and properties of their languages.

Some Tautologies

For any $A, B \in \mathcal{F}$, the following formulas are tautologies.

Implication and Negation

$$(A \Rightarrow (B \Rightarrow A)),\quad ((A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))),$$

$$((\neg B \Rightarrow \neg A) \Rightarrow ((\neg B \Rightarrow A) \Rightarrow B)) \tag{10}$$

$$(A \Rightarrow A), \quad (B \Rightarrow \neg\neg B), \quad (\neg A \Rightarrow (A \Rightarrow B)), \quad (A \Rightarrow (\neg B \Rightarrow \neg(A \Rightarrow B))),$$
$$(\neg\neg B \Rightarrow B), \quad ((A \Rightarrow B) \Rightarrow ((\neg A \Rightarrow B) \Rightarrow B)), \quad ((\neg A \Rightarrow A) \Rightarrow A).$$

Disjunction, Conjunction

$$(A \Rightarrow (A \cup B)), \quad (B \Rightarrow (A \cup B)), \quad ((A \cap B) \Rightarrow A), \quad ((A \cap B) \Rightarrow A),$$
$$((A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow ((A \cup B) \Rightarrow C))),$$
$$(((A \cap B) \Rightarrow C) \Rightarrow (A \Rightarrow (B \Rightarrow C)),$$
$$(\neg(A \cap B) \Rightarrow (\neg A \cup \neg B)), \quad ((\neg A \cup \neg B) \Rightarrow \neg(A \cap B)), \tag{11}$$
$$((\neg A \cup B) \Rightarrow (A \Rightarrow B)), \quad ((A \Rightarrow B) \Rightarrow (\neg A \cup B)),$$
$$(A \cup \neg A).$$

Contraposition (1)

$$((A \Rightarrow B) \Leftrightarrow (\neg B \Rightarrow \neg A)), \quad ((B \Rightarrow A) \Leftrightarrow (\neg A \Rightarrow \neg B)). \tag{12}$$

Contraposition (2)

$$((\neg A \Rightarrow B) \Leftrightarrow (\neg B \Rightarrow A)), \quad ((A \Rightarrow \neg B) \Leftrightarrow (B \Rightarrow \neg A)). \tag{13}$$

Double Negation

$$(\neg\neg A \Leftrightarrow A), \tag{14}$$

Logical Equivalences

Logical equivalence is a very useful notion when we want to obtain new formulas or new tautologies, if needed, on a base of some already known in a way that guarantee preservation of the logical value of the initial formula. For any formulas A, B, we say that are *logically equivalent* if they always have the same logical value. We write it symbolically as $A \equiv B$. We have to remember that the symbol " $\equiv$" not a logical connective. It is a *metalanguage symbol* for saying "A, B are logically equivalent". This is a very useful symbol. It says that two formulas always have the same logical value, hence it can be used in the same way we use the equality symbol " =." Formally we define it as follows.

Definition 34 (Logical Equivalence)

For any $A,\ B \in \mathcal{F}$, we say that the formulas A and B are logically equivalent and denote it as $A \equiv B$

if and only if $v^*(A) = v^*(B)$, for all $v : VAR \rightarrow \{T, F\}$.

Observe that the following property follows directly from the definition 34.

Property 1

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

$$A \equiv B \quad \textit{if and only if} \quad \models (A \Leftrightarrow B)$$

For example we write the laws of contraposition (15), (16), and the law of double negation (17) as **logical equivalences** as follows.

E - Contraposition (1), (2)

$$(A \Rightarrow B) \equiv (\neg B \Rightarrow \neg A), \quad (B \Rightarrow A) \equiv (\neg A \Rightarrow \neg B). \tag{15}$$

E - Contraposition (2)

$$(\neg A \Rightarrow B) \equiv (\neg B \Rightarrow A), \quad (A \Rightarrow \neg B) \equiv (B \Rightarrow \neg A). \tag{16}$$

E - Double Negation

$$\neg\neg A \equiv A. \tag{17}$$

Logical equivalence is a very useful notion when we want to obtain new formulas, or tautologies, if needed, on a base of some already known in a way that guarantee preservation of the logical value of the initial formula.

For example, we easily obtain equivalences for laws of E -mContraposition (16) from equivalences for laws of E- Contraposition (15) and the E - Double Negation equivalence (17) as follows. $(\neg A \Rightarrow B) \equiv^{(15)} (\neg B \Rightarrow \neg\neg A) \equiv^{(17)} (\neg B \Rightarrow A)$. We also have that $(A \Rightarrow \neg B) \equiv^{(15)} (\neg\neg B \Rightarrow \neg A) \equiv^{(17)} (B \Rightarrow \neg A)$. This end the proof of E- Contraposition (16).

The correctness of the above procedure of proving new equivalences from the known ones is established by the following theorem.

Theorem 12 (Equivalence Substitution)

Let a formula B_1 be obtained from a formula A_1 by a substitution of a formula B for one or more occurrences of a sub-formula A of A_1, what we denote as

$$B_1 = A_1(A/B).$$

Then the following holds for any formulas $A,\ A_1,\ B,\ B_1 \in \mathcal{F}$.

$$\textit{If} \quad A \equiv B, \quad \textit{then} \quad A_1 \equiv B_1. \tag{18}$$

Proof

By the logical equivalence Definition 34 proving our theorem statement 18 is equivalent to proving that the implication

$$\text{If} \quad v^*(A) = v^*(B), \quad \text{then} \quad v^*(A_1) = v^*(B_1) \tag{19}$$

holds for all $v : VAR \rightarrow \{T, F\}$.

Consider a truth assignment v. If $v^*(A) \neq v^*(B)$, then the implication (19) is vacuously true. If $v^*(A) = v^*(B)$, then so $v^*(A_1) = v^*(B_1)$, since B_1 differs from A_1 only in containing B in some places where A_1 contains A and the implication (19) holds.

Example 17

Let $A_1 = (C \cup D)$ and $B = \neg\neg C$. By E - Double Negation equivalence (17) we have that $\neg\neg C \equiv C$. Let $B_1 = A_1(C/B) = A_1(C/\neg\neg C) = (\neg\neg C \cup D)$. By the Equivalence Substitution Theorem 12

$$(C \cup D) \equiv (\neg\neg C \cup D).$$

Equivalence of Languages

The next set of equivalences, or corresponding tautologies, correspond the notion of *definability of connectives* discussed in section 3. For example, a tautology

$$\models ((A \Rightarrow B) \Leftrightarrow (\neg A \cup B))$$

makes it possible, via Property 1, to define implication in terms of disjunction and negation. We state it in a form of logical equivalence and call it as follows.

Definability of Implication in terms of negation and disjunction:

$$(A \Rightarrow B) \equiv (\neg A \cup B) \tag{20}$$

Observation 5 *The direct proof of this and other Definability of Connectives Equivalences presented here follow from the definability formulas developed in the the proof of the Definability of Connectives Theorem 2, hence the names.*

We are using the logical equivalence notion, instead of the tautology notion, as it makes the manipulation of formulas much easier.

The equivalence 20 allows us, by the force of Theorem 12 to replace any formula of the form $(A \Rightarrow B)$ placed anywhere in another formula by a formula $(\neg A \cup B)$ while preserving their logical equivalence. Hence we can use the equivalence (20) to transform a given formula containing implication into an logically equivalent formula that does contain implication (but contains negation and disjunction).

We usually use the equation 20 to transform any formula A of language containing implication into a formula B of language containing disjunction and negation and not containing implication at all, such that $A \equiv B$.

Example 18

Let $A = ((C \Rightarrow \neg B) \Rightarrow (B \cup C))$.

We use equality (20) to transform A into a logically equivalent formula not containing $\Rightarrow$ *as follows.*

$((C \Rightarrow \neg B) \Rightarrow (B \cup C)) \equiv (\neg(C \Rightarrow \neg B) \cup (B \cup C))) \equiv (\neg(\neg C \cup \neg B) \cup (B \cup C)))$.

It means that for example that we can, by the Theorem 12 transform any formula A of the language $\mathcal{L}_1 = \mathcal{L}_{\{\neg,\cap,\Rightarrow\}}$ into a logically formula B of the language $\mathcal{L}_2 = \mathcal{L}_{\{\neg,\cap,\cup\}}$. In general, we say that we can transform a language $\mathcal{L}_1$ into a logically equivalent language $\mathcal{L}_2$ if the following condition **C1** holds.

C1: for any formula A of $\mathcal{L}_1$, there is a formula B of $\mathcal{L}_2$, such that $A \equiv B$.

Example 19

Let $A = (\neg A \cup (\neg A \cup \neg B))$. *We also can use, in this case, the equivalence 20 as follows.*

$(\neg A \cup (\neg A \cup \neg B)) \equiv (\neg A \cup (A \Rightarrow \neg B)) \equiv (A \Rightarrow (A \Rightarrow \neg B))$.

It means we eliminated disjunction from A by replacing it by logically equivalent formula containing implication only.

Observe, that we can't always use the equivalence (20) to eliminate any disjunction. For example, we can't use it for a formula $A = ((a \cup b) \cap \neg a)$.

In order to be able to transform any formula of a language containing disjunction (and some other connectives) into a language with negation and implication (and some other connectives), but without disjunction we need the following logical equivalence.

Definability of Disjunction in terms of negation and implication:

$$(A \cup B) \equiv (\neg A \Rightarrow B) \tag{21}$$

Example 20

Consider a formula $A = (a \cup b) \cap \neg a)$.

We use equality (21) to transform A into its logically equivalent formula not containing $\cup$ *as follows:* $((a \cup b) \cap \neg a) \equiv ((\neg a \Rightarrow b) \cap \neg a)$.

In general, we use the equality 21 and Theorem 12 to transform any formula C of the language $\mathcal{L}_2 = \mathcal{L}_{\{\neg,\cap,\cup\}}$ into a logically equivalent formula D of the language $\mathcal{L}_1 = \mathcal{L}_{\{\neg,\cap,\Rightarrow\}}$. In general, the following condition hols.

C2: for any formula C of $\mathcal{L}_2$, there is a formula D of $\mathcal{L}_1$, such that $C \equiv D$.

The languages $\mathcal{L}_1$ and $\mathcal{L}_2$ for which we the conditions **C1, C2** hold are logically equivalent and denote it by $\mathcal{L}_1 \equiv \mathcal{L}_2$.

We put it in a general, formal definition as follows.

Definition 35 (Equivalence of Languages)

Given two languages:
$\mathcal{L}_1 = \mathcal{L}_{CON_1}$ *and* $\mathcal{L}_2 = \mathcal{L}_{CON_2}$*, for* $CON_1 \neq CON_2$.
We say that they are **logically equivalent** *and denote it as* $\mathcal{L}_1 \equiv \mathcal{L}_2$ *if and only if the following conditions* **C1, C2** *hold.*

C1 *For every formula* A *of* $\mathcal{L}_1$*, there is a formula* B *of* $\mathcal{L}_2$*, such that* $A \equiv B$,

C2 *For every formula* C *of* $\mathcal{L}_2$*, there is a formula* D *of* $\mathcal{L}_1$*, such that* $C \equiv D$.

Example 21

To prove the logical equivalence $\mathcal{L}_{\{\neg,\cup\}} \equiv \mathcal{L}_{\{\neg,\Rightarrow\}}$ *we need two definability equivalences (20) and (21), and the Theorem 12.*

Exercise 16

To prove the logical equivalence $\mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow\}} \equiv \mathcal{L}_{\{\neg,\cap,\cup\}}$ *we needed only the definability equivalence (20).*

Solution
The equivalence (20) proves, by Theorem 12 that for any formula A of $\mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow\}}$ there is B of $\mathcal{L}_{\{\neg,\cap,\cup\}}$ that equivalent to A, i.e. condition **C1** holds. Any formula A of language $\mathcal{L}_{\{\neg,\cap,\cup\}}$ is also a formula of $\mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow\}}$ and of course $A \equiv A$, so both conditions **C1** and **C2** of definition 35 are satisfied.

Exercise 17

Show that $\mathcal{L}_{\{\neg,\cap\}} \equiv \mathcal{L}_{\{\neg,\Rightarrow\}}$.

Solution
The equivalence of languages holds by Theorem 12, Observation 5, and the following two logical equalities. **Definability of Conjunction** in terms of implication and negation and **Definability of Implication** in terms of conjunction and negation:

$$(A \cap B) \equiv \neg(A \Rightarrow \neg B) \tag{22}$$

$$(A \Rightarrow B) \equiv \neg(A \cap \neg B). \tag{23}$$

Exercise 18

Show that $\mathcal{L}_{\{\neg,\cap\}} \equiv \mathcal{L}_{\{\neg,\cup\}}$.

Solution
Similarly, it is true by Theorem 12, Observation 5, and the following two logical

equalities. Definability of disjunction in terms of negation and conjunction and definability of conjunction in terms of negation and disjunction:

$$(A \cup B) \equiv \neg(\neg A \cap \neg B) \tag{24}$$

$$(A \cap B) \equiv \neg(\neg A \cup \neg B). \tag{25}$$

Theorem 12, Observation 5, and definability of equivalence in terms of implication and conjunction equality

$$(A \Leftrightarrow B) \equiv ((A \Rightarrow B) \cap (B \Rightarrow A)). \tag{26}$$

prove that, for example, $\mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow\}} \equiv \mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow,\Leftrightarrow\}}$.

Exercise 19

Show that $\mathcal{L}_{\{\neg,\cap\}} \equiv \mathcal{L}_{\{\uparrow\}}$ *and* $\mathcal{L}_{\{\neg,\cap\}} \equiv \mathcal{L}_{\{\uparrow\}}$

Proof
We use the proof of Theorem 3 to prove the following definability equivalences of $\neg$ and $\cap$ in terms of $\uparrow$:

$$\neg A \equiv (A \uparrow A), \quad (A \cap B) \equiv (A \uparrow B) \uparrow (A \uparrow B) \tag{27}$$

and definability equivalences of $\neg$ and $\cup$ in terms of $\downarrow$:

$$\neg A \equiv (A \downarrow A), \quad (A \cup B) \equiv (A \downarrow B) \downarrow (A \downarrow B). \tag{28}$$

This proves the condition **C1** of definition 35.

The definability equivalences for fulfillment of the condition **C2** are:

$$(A \uparrow B) = \neg(A \cup B) \quad \text{and} \quad (A \uparrow B) = \neg(A \cup B) \tag{29}$$

Here are some more frequently used, important logical equivalences.

Idempotent

$$(A \cap A) \equiv A, \qquad (A \cup A) \equiv A,$$

Associativity

$$((A \cap B) \cap C) \equiv (A \cap (B \cap C)), \quad ((A \cup B) \cup C) \equiv (A \cup (B \cup C)),$$

Commutativity

$$(A \cap B) \equiv (B \cap A), \qquad (A \cup B) \equiv (B \cup A),$$

Distributivity

$$(A \cap (B \cup C)) \equiv ((A \cap B) \cup (A \cap C)), \quad (A \cup (B \cap C)) \equiv ((A \cup B) \cap (A \cup C)),$$

De Morgan Laws

$$\neg(A \cup B) \equiv (\neg A \cap \neg B), \quad \neg(A \cap B) \equiv (\neg A \cup \neg B).$$

Negation of Implication

$$\neg(A \Rightarrow B) \equiv (A \cap \neg B), \tag{30}$$

De Morgan laws are named after A. De Morgan (1806 - 1871), an English logician, who discovered analogous laws for the algebra of sets. They stated that for any sets A,B the complement of their union is the same as the intersection of their complements, and vice versa, the complement of the intersection of two sets is equal to the union of their complements. The laws of the propositional calculus were formulated later, but they are usually also called De Morgan Laws.

Observe that De Morgan Laws tell us how to negate disjunction and conjunction, so the laws stating how to negate other connectives follows them.

Consider a tautology A: $\models ((\neg(A \Rightarrow B) \Rightarrow \neg A) \Rightarrow (A \Rightarrow B))$.

We know by (20) that $(A \Rightarrow B) \equiv (\neg A \cup B)$. By Theorem 12, if we replace $(A \Rightarrow B)$ by $(\neg A \cup B)$ in A, the logical value of A will remain the same and $((\neg(A \Rightarrow B) \Rightarrow \neg A) \Rightarrow (A \Rightarrow B)) \equiv ((\neg(\neg A \cup B) \Rightarrow \neg A) \Rightarrow (\neg A \cup B))$. Now we use de Morgan Laws and Double Negation Laws and by Theorem 12 we get $((\neg(A \Rightarrow B) \Rightarrow \neg A) \Rightarrow (A \Rightarrow B)) \equiv ((\neg(\neg A \cup B) \Rightarrow \neg A) \Rightarrow (\neg A \cup B)) \equiv (((\neg\neg A \cap \neg B) \Rightarrow \neg A) \Rightarrow (\neg A \cup B)) \equiv (((A \cap \neg B) \Rightarrow \neg A) \Rightarrow (\neg A \cup B))$.

This proves that

$$\models (((A \cap \neg B) \Rightarrow \neg A) \Rightarrow (\neg A \cup B)).$$

Exercise 20

Prove using proper logical equivalences that

(i) $\neg(A \Leftrightarrow B) \equiv ((A \cap \neg B) \cup (\neg A \cap B))$,

(ii) $((B \cap \neg C) \Rightarrow (\neg A \cup B)) \equiv ((B \Rightarrow C) \cup (A \Rightarrow B))$.

Solution (i)
$\neg(A \Leftrightarrow B) \equiv^{(26)} \neg((A \Rightarrow B) \cap (B \Rightarrow A)) \equiv^{de\ Morgan} (\neg(A \Rightarrow B) \cup \neg(B \Rightarrow A)) \equiv^{(30)} ((A \cap \neg B) \cup (B \cap \neg A)) \equiv^{commut} ((A \cap \neg B) \cup (\neg A \cap B))$.

Solution (ii)
$((B \cap \neg C) \Rightarrow (\neg A \cup B)) \equiv^{(21)} (\neg(B \cap \neg C) \cup (\neg A \cup B)) \equiv^{de\ Morgan} ((\neg B \cup \neg\neg C) \cup (\neg A \cup B)) \equiv^{(17)} ((\neg B \cup C) \cup (\neg A \cup B)) \equiv^{(21)} ((B \Rightarrow C) \cup (A \Rightarrow B))$.

5 Many Valued Semantics: Łukasiewicz, Heyting, Kleene, Bohvar

Many valued logics in general and 3-valued logics in particular is an old object of study which has its beginning in the work of a Polish mathematician Jan Leopold Łukasiewicz in 1920. He was the first to define a 3 - valued semantics for the language $\mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow\}}$ of classical logic, and called it a *three valued logic* for short. He left the problem of finding a proper axiomatic proof system for it (i.e. the one that is complete with respect to his semantics) open. The same happened to all other 3 - valued semantics presented here. They were also first called *3 valued logics* and this terminology is still widely used. Nevertheless, as these logics were defined only semantically, i.e. defined by providing a semantics for their languages we call them just semantics (for logics to be developed), not logics. 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) always is three fold: we define semantics, create an an axiomatic proof system and prove a completeness theorem that established a relationship between a given semantics and proof system.

The first of many valued logics invented were first presented in a semantical form only for other components to be developed later. We can think about the process of their creation as inverse to the creation of Classical Logic, Modal Logics, the Intuitionistic Logic which existed as axiomatic systems longtime before invention of their formal semantics.

Formal definition of all many valued extensional semantics **M** for the language $\mathcal{L}$ we present and discuss here follows the extensional semantics Definition 7 in general and the pattern of presented in detail for the classical case (Section 3) in particular. It consists of giving definitions of the following main components:

Step 1: given the language $\mathcal{L}$ we define a set of logical values and its distinguish value T, and define all logical connectives of $\mathcal{L}$
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**.

We present here some of the historically first 3-valued extensional semantics, called also 3-valued logics. They are named after their authors: *Łukasiewicz, Heyting, Kleene*, and *Bochvar*.

The 3-valued semantics we define here enlist a third logical value, besides classical T, F. We denote this third value by $\perp$, or m in case of Bochvar semantics. We also assume that the third value is intermediate between truth and falsity, i.e. that $F < \perp < T$ and $F < m < T$.

There has been many of proposals relating both to the intuitive interpretation of this third value $\perp$. If T is the only designated value, the third value $\perp$ corresponds to some notion of *incomplete information*, like *undefined* or *unknown* and is often denoted by the symbol U or I. If, on the other hand, $\perp$ corresponds to *inconsistent information*, i.e. its meaning is something like *known to be both true and false* then corresponding semantics takes both T and the third logical value $\perp$ as designated. In general, the third logical value denotes a notion of "unknown", "uncertain", "undefined", or even can express that "we don't have a complete information", depending on the context and motivation for the logic we plan to develop. 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.*

Łukasiewicz Semantics L

Motivation

Łukasiewicz developed his semantics (called logic) to deal with future contingent statements. According to him, such 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. Intuitively, $\perp$ signifies that the statement cannot be assigned the value true of false; it is not simply that we do not have sufficient information to decide the truth value but rather the statement does not have one.

We define all the steps of the Definition7 in case of Łukasiewicz' s semantics to establish a pattern and proper notation. We leave the detailed steps of other semantics as an exercise for the reader.

Step 1: L Connectives

The language of the semantics **L** is $\mathcal{L}_{\{\neg,\ \cup,\ \cap,\ \Rightarrow,\ \Leftrightarrow\}}$. The set LV of logical values is $\{T,\ \perp, F\}$. T is the distinguished value. We assume that the set of log0cal values is ordered, i.e. that

$$F < \perp < T.$$

L Negation is a function $\neg :\ \{T, \perp, F\} \longrightarrow \{T, \perp, F\}$ such that

$$\neg \perp = \perp, \quad \neg T = F, \quad \neg F = T.$$

L Conjunction is a function $\cap :\ \{T, \perp, F\} \times \{T, \perp, F\} \longrightarrow \{T, \perp, F\}$ such that for any $(x, y) \in \{T, \perp, F\} \times \{T, \perp, F\}$, we put

$$x \cap y = min\{x,\ x\}.$$

L Disjunction is a function $\cup: \{T, \bot, F\} \times \{T, \bot, F\} \longrightarrow \{T, \bot, F\}$, such that for any $(a, b) \in \{T, \bot, F\} \times \{T, \bot, F\}$, we put

$$x \cup y = max\{x,\ y\}$$

L Implication is a function $\Rightarrow$: $\{T, \bot, F\} \times \{T, \bot, F\} \longrightarrow \{T, \bot, F\}$ such that for any $(x, y) \in \{T, \bot, F\} \times \{T, \bot, F\}$, we put

$$x \Rightarrow y = \begin{cases} \neg x \cup y & \text{if } x > y \\ T & \text{otherwise} \end{cases} \tag{31}$$

We write function defining the connectives in a standard form of tables defining operations in finite sets. We call these tables *truth tables definition* of propositional connectives, or *L connectives truth tables* for short.

L Connectives Truth Tables

$\neg$	F	$\bot$	T
	T	$\bot$	F

$\cap$	F	$\bot$	T
F	F	F	F
$\bot$	F	$\bot$	$\bot$
T	F	$\bot$	T

$\cup$	F	$\bot$	T
F	F	$\bot$	T
$\bot$	$\bot$	$\bot$	T
T	T	T	T

$\Rightarrow$	F	$\bot$	T
F	T	T	T
$\bot$	$\bot$	T	T
T	F	$\bot$	T

Step 2: Truth Assignment, Truth Extension

A **truth assignment** is now any function $v : VAR \longrightarrow \{F,\ \bot,\ T\}$. We define its extension to the set $\mathcal{F}$ of all formulas as any function $v^* : \mathcal{F} \longrightarrow \{T, F\}$, such that the following conditions are satisfied.

(1) for any $a \in VAR$, $v^*(a) = v(a)$;

(2) for any $A, B \in \mathcal{F}$,

$$v^*(\neg A) = \neg v^*(A);$$
$$v^*((A \cap B)) = v^*(A) \cap v^*(B);$$
$$v^*((A \cup B)) = v^*(A) \cup v^*(B);$$
$$v^*((A \Rightarrow B)) = v^*(A) \Rightarrow v^*(B).$$

Step 3: Satisfaction, Model, Counter-Model

We say that a truth assignment $v : VAR \longrightarrow \{F, \perp, T\}$ **L** satisfies a formula $A \in \mathcal{F}$ if and only if $v^*(A) = T$. We denote it by $v \models_{\mathbf{L}} A$.

Any truth assignment v, $v : VAR \longrightarrow \{F, \perp, T\}$ such that $v \models_{\mathbf{L}} A$ is called a **L** model for A.

We say that a truth assignment v does not **L** satisfy a formula $A \in \mathcal{F}$ and denote it by $v \not\models_{\mathbf{L}} A$, if and only if $v^*(A) \neq T$.

Any truth assignment v, $v : VAR \longrightarrow \{F, \perp, T\}$ such that $v \not\models_{\mathbf{L}} A$ is called a **L** counter- model for A.

Step 4: L Tautology

We define, for any $A \in \mathcal{F}$, A is a **L** tautology if and only if $v^*(A) = T$ for all $v : VAR \longrightarrow \{F, \perp, T\}$. We also say that A is a **L** tautology if and only if all truth assignments $v : VAR \longrightarrow \{F, \perp, T\}$ are **L** models for A. We write the statement " A is a **L** tautology" symbolically as

$$\models_{\mathbf{L}} A.$$

As a next step we define, as we did in the case of classical semantics the notions of restricted truth assignment and restricted models, (Definitions 26, 27) i.e. we have the following.
Any function $v_A : VAR_A \longrightarrow \{F, \perp, T\}$, such that $v_A \models_{\mathbf{L}} A$ ($v_A \not\models_{\mathbf{L}} A$) is called a restricted **L** model (**L** counter model) for A, where VAR_A is the set of all propositional variables appearing in A. We call the function v_A, a truth assignment restricted to A, or restricted truth assignment for short.

We prove, in the same way we proved Theorem8 in Section 3, the following theorem that justifies the correctness of the truth tables **L** tautologies verification method.

Theorem 13 (L Truth Tables)

For any formula $A \in \mathcal{F}$,
$\models_{\mathbf{L}} A$ *if and only if* $v_A \models_{\mathbf{L}} A$ *for all* $v_A : VAR_A \longrightarrow \{T, \perp, F\}$, *i.e.*
$\models_{\mathbf{L}} A$ *if and only if all* v_A *are restricted models for* A.

Directly from Theorem 13 we get that the notion of **L** propositional tautology is decidable, i.e. that the following holds.

Theorem 14 (Decidability)

For any formula $A \in \mathcal{F}$, one has examine at most 3^{VAR_A} truth assignments $v_A : VAR_A \longrightarrow \{F, \perp, T\}$ in order to decide whether $\models_{\mathbf{L}} A$, or $\not\models_{\mathbf{L}} A$, i.e. the notion of **L** *tautology is decidable.*

We denote by **LT** the set of all **L** tautologies, i.e. we have that

$$\mathbf{LT} = \{A \in \mathcal{F} : \models_{\mathbf{L}} A\}. \tag{32}$$

We just proved (Theorem 14) the correctness of the truth table tautology verification method for **L** semantics stated as follows.

L Truth Table Method

The verification method, called a **truth table method** consists of examination, for any formula A, all possible truth assignments **restricted** to A. By Theorem 13 we have to perform at most $3^{|VAR_A|}$ steps. If we find a truth assignment which **does not** evaluate A to T, i.e. evaluates A to F, or to $\perp$, we stop the process and give answer: $\not\models_{\mathbf{L}} A$. Otherwise we continue. If all truth assignments evaluate A to T, we give answer: $\models_{\mathbf{L}} A$.

Consider, for example, a formula A: $(a \Rightarrow a)$. There are $3^1 = 3$ possible restricted truth assignment $v : \{a\} \longrightarrow \{F, \perp, T\}$. We list them, and evaluate their value on the formula A in a form of an extended truth table as follows.

v	a	$v^*(A)$ *computation*	$v^*(A)$
v_1	T	$T \Rightarrow T = T$	T
v_2	$\perp$	$\perp \Rightarrow \perp = T$	T
v_3	F	$F \Rightarrow F = T$	T

This proves that the classical tautology $(a \Rightarrow a)$ is also a **L** tautology, i.e.

$$\models (a \Rightarrow a) \text{ and } \models_{\mathbf{L}} (a \Rightarrow a). \tag{33}$$

Moreover (33) proves that the condition (3) of the definition 14 is fulfilled and the **L** semantics is well defined. We put it as a separate fact.

Fact 5

The Lukasiewicz semantics **L** *is* **well defined***.*

As a next step we can adopt all other classical tautology verification methods from Section 3. It is a quite straightforward adaptation and we leave it a san exercise. Moreover it works for all of many valued semantics presented here, as does the Decidability Theorem 14.

When defining and developing a new logic the first question one asks is how it relates and compares with the classical case, it means with the classical logic. In case of new semantics (logics defined semantically) we describe this relationship in terms of respective sets of tautologies.

Let **LT**, **T** denote the sets of all **L** and classical tautologies, respectively.

Theorem 15

The following relationship holds between classical and **L** *tautologies:*

$$\mathbf{LT} \neq \mathbf{T} \text{ and } \mathbf{LT} \subset \mathbf{T}. \tag{34}$$

Proof
Consider a formula $(\neg a \cup a)$. It is obviously a classical tautology. Take any truth assignment $v : VAR \longrightarrow \{F, \bot, T\}$ such that $v(a) = \bot$. By definition we have that $v^*(\neg a \cup a) = v^*(\neg a) \cup v^*(a) = \neg v(a) \cup v(a) = \neg \bot \cup \bot = \bot \cup \bot = \bot$. This proves that v is a **L** counter-model for $(\neg a \cup a)$ and hence $\not\models_{\mathbf{L}} \ (\neg a \cup a)$. This proves $\mathbf{LT} \neq \mathbf{T}$.

Observe now that if we restrict the values of functions defining **L** connectives to the values T and F only, we get the functions defining the classical connectives. It is directly visible when we compare the **L** and classical connectives truth tables. This means that if $v^*(A) = T$ for all $v : VAR \longrightarrow \{F, \bot, T\}$, then $v^*(A) = T$ for any $v : VAR \longrightarrow \{F, T\}$ and for any $A \in \mathcal{F}$, i.e. $\mathbf{LT} \subset \mathbf{T}$.

Exercise 21

Use the fact that $v : VAR \longrightarrow \{F, \bot, T\}$ *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.*

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. $b = T$, and hence $(a \cap T) = \bot$ if and only if $a = \bot$. We get that v is such that $v(a) = \bot$ and $v(b) = T$. We evaluate (in short hand notation) $v^*(((b \Rightarrow \neg a) \Rightarrow (a \Rightarrow \neg b)) \cup (a \Rightarrow b)) = (((T \Rightarrow \neg \bot) \Rightarrow (\bot \Rightarrow \neg T)) \cup (\bot \Rightarrow T)) = ((\bot \Rightarrow \bot) \cup T) = T$.

Consider **c2**. We have $\neg b = \bot$, i.e. $b = \bot$, and hence $(a \cap \bot) = T$ what is impossible, hence v from case **c1** is the only one, and $v^*(((b \Rightarrow \neg a) \Rightarrow (a \Rightarrow \neg b)) \cup (a \Rightarrow b)) = T$.

$\mathbf{L}_4$ Semantics

We define the semantics $\mathbf{L}_4$ as follows. The language is $\mathcal{L} = \mathcal{L}_{\{\neg,\Rightarrow,\cup,\cap\}}$. The logical connectives $\neg, \Rightarrow, \cup, \cap$ of $\mathbf{L}_4$ as the following operations in the set $\{F, \bot_1, \bot_2, T\}$, where $\{F < \bot_1 < \bot_2 < T\}$.

$\mathbf{L}_4$ **Negation** is a function such that $\neg\bot_1 = \bot_1,\ \neg\bot_2 = \bot_2,\ \neg F = T,\ \neg T = F.$

$\mathbf{L}_4$ **Conjunction** is a function such that for any $x, y \in \{F, \bot_1, \bot_2, T\},\ x \cap x = min\{x, y\}$.

$\mathbf{L}_4$ **Disjunction** is a function such that for any $x, y \in \{F, \bot_1, \bot_2, T\},\ x \cup y = min\{x, y\}$.

$\mathbf{L}_4$ **Implication** is a function such that for any $x, y \in \{F, \bot_1, \bot_2, T\}$,

$$x \Rightarrow y = \begin{cases} \neg x \cup y & \text{if } x > y \\ T & \text{otherwise} \end{cases} \tag{35}$$

Exercise 22

Here are 3 simple problems.

1. Write down $\mathbf{L}_4$ *Connectives Truth Tables.*

2. Give an example of a $\mathbf{L}_4$ *tautology.*

3. We know that the formula $((a \Rightarrow b) \Rightarrow (\neg a \cup b))$ *is a classical tautology, i.e.* $\models ((a \Rightarrow b) \Rightarrow (\neg a \cup b))$. *Verify whether* $\models_{Ł_4} ((a \Rightarrow b) \Rightarrow (\neg a \cup b))$.

Solution 1.
Here are $Ł_4$ Connectives Truth Tables.

$\neg$	F	$\bot_1$	$\bot_2$	T
	T	$\bot_1$	$\bot_2$	F

$\cap$	F	$\bot_1$	$\bot_2$	T
F	F	F	F	F
$\bot_1$	F	$\bot_1$	$\bot_1$	$\bot_1$
$\bot_2$	F	$\bot_1$	$\bot_2$	$\bot_2$
T	F	$\bot_1$	$\bot_2$	T

$\cup$	F	$\bot_1$	$\bot_2$	T
F	F	$\bot_1$	$\bot_2$	T
$\bot_1$	$\bot_1$	$\bot_1$	$\bot_2$	T
$\bot_2$	$\bot_2$	$\bot_2$	$\bot_2$	T
T	T	T	T	T

$\Rightarrow$	F	$\bot_1$	$\bot_2$	T
F	T	T	T	T
$\bot_1$	$\bot_1$	T	T	T
$\bot_2$	$\bot_2$	$\bot_2$	T	T
T	F	$\bot_1$	$\bot_2$	T

Solution 2.
Observe that by definition of $\mathbf{L}_4$ implication we get $x \Rightarrow x = T$ for all $x \in \{F, \perp_1, \perp_2, T\}$. Hence $v^*((a \Rightarrow a)) = v(a) \Rightarrow v(a) = T$ for all v, what proves $\models_{\mathbf{L}_4} (a \Rightarrow a)$.

Solution 3.
We use the Proof by Contradiction Method (section 3) to verify whether $\models_{\mathbf{L}_4}((a \Rightarrow b) \Rightarrow (\neg a \cup b))$. Observe that it applied to any situation, as its correctness is based on our classical reasoning. Assume that $\not\models_{\mathbf{L}_4} ((a \Rightarrow b) \Rightarrow (\neg a \cup b))$. Let $v : VAR \longrightarrow \{F, \perp_1, \perp_2, T\}$, such that $v^*(((a \Rightarrow b) \Rightarrow (\neg a \cup b))) \neq T$. Observe that in $\mathbf{L}_4$ semantics, for any formula $A \in \mathcal{F}$, $v^*(A) \neq T$ gives us three possibilities $v^*(A) = F$, $v^*(A) = \perp_1$, or $v^*(A) = \perp_2$ to consider (as opposed to one case in classical case). It is a lot of work, but still less then listing and evaluating $4^2 = 16$ possibilities of all restricted truth assignment. Moreover, our formula is a classical tautology, hence we know that it evaluates in T for all combinations of T and F. A good strategy is to examine first some possibilities of evaluating variables a, b for combination of $\perp_1, \perp_2$ with hope of finding a counter model. So let's v be a truth assignment such that $v(a) = v(b) = \perp_1$. We evaluate $v^*((a \Rightarrow b) \Rightarrow (\neg a \cup b)) = ((\perp_1 \Rightarrow \perp_1) \Rightarrow (\neg \perp_1 \cup \perp_1)) = (T \Rightarrow (\perp_1 \cup \perp_1)) = (T \Rightarrow \perp_1) = \perp_1$. This proves that v is a counter-model for our formula. Observe that the v serves also as a $\mathbf{L}$ counter model for A when we put $\perp_1 = \perp$ and so we get

$$\models ((a \Rightarrow b) \Rightarrow (\neg a \cup b)), \ \not\models_{\mathbf{L}_4} ((a \Rightarrow b) \Rightarrow (\neg a \cup b)), \not\models_{\mathbf{L}} \ ((a \Rightarrow b) \Rightarrow (\neg a \cup b))$$

Obviously, any v such that $v(a) = v(b) = \perp_2$ is also a counter model for A, as $v^*((a \Rightarrow b) \Rightarrow (\neg a \cup b)) = ((\perp_2 \Rightarrow \perp_2) \Rightarrow (\neg \perp_2 \cup \perp_2)) = (T \Rightarrow (\perp_2 \cup \perp_2)) = (T \Rightarrow \perp_2) = \perp_2$. We leave it as an exercise to find all possible counter models for A.

Heyting Semantics H

Motivation
We discuss here the semantics **H** because of its connection with intuitionistic logic. The **H** connectives are such that they represent operations in a certain 3 element algebra, historically called a 3 element *pseudo-boolean* algebra. Pseudo-boolean algebras were created by McKinsey and Tarski in 1948 to provide semantics for intuitionistic logic. The intuitionistic logic, the most important rival to the classical one was defined and developed 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 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**.

We say, that formula A is an intutionistic tautology if and only if it is valid in *all* *pseudo-boolean (Heying) algebras*. The pseudo boolean algebras are defined in a very general and mathematically sophisticated way. Their universe (it means the set of logical values) can be any non empty set. Their operations that correspond to $\mathcal{L} = \mathcal{L}_{\{\neg, \Rightarrow, \cup, \cap\}}$ connectives must fulfill a set of special properties (axioms). But we can prove that the operations defined by **H** connectives form a 3-element pseudo boolean algebra with the universe $U = \{F, \perp, T\}$. Hence, if A is an intuitionistic tautology, then in it is also valid (tautologically true) for the **H** semantics, i.e. all intuitionistic propositional tautologies are also the **H** semantics tautologies. It means that our **H** is a good candidate for finding counter models for the formulas that might not be intuitionistic tautologies.

The other type of models, called Kripke models were defined by Kripke in 1964 and were proved later to be equivalent to the pseudo-boolean models. They are very general and serve as a method of defining *not extensional* semantics for various classes of logics. That includes semantics for a great number of modal, knowledge, belief logics, and many new logics developed and being developed by computer scientists.

H Connectives

We adopt the same language as in case of classical and Łukasiewicz's **L** semantics, i.e. $\mathcal{L} = \mathcal{L}_{\{\neg, \Rightarrow, \cup, \cap\}}$. We assume, as before, that $\{F < \perp < T\}$.

The connectives $\neg$, $\cup$, $\cap$ of **H** are defined as in **L** semantics. They are functions defined by formulas $x \cup y = max\{x, y\}$, $x \cap y = min\{x, y\}$, for any $x, y \in \{F, \perp, T\}$.

The definition of implication and negation for **H** semantics differs **L** semantics and we define them as follows.

H Implication is a function $\Rightarrow$: $\{T, \perp, F\} \times \{T, \perp, F\} \longrightarrow \{T, \perp, F\}$ such that for any $(x, y) \in \{T, \perp, F\} \times \{T, \perp, F\}$, we put

$$x \Rightarrow y = \begin{cases} T & \text{if } x \leq y \\ y & \text{otherwise} \end{cases} \tag{36}$$

H negation is a function $\neg :$ $\{F, \perp, T\} \longrightarrow \{F, \perp, T\}$, such that

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

The truth tables for **H** disjunction and conjunction are hence the same as corresponding **L** tables and the truth tables for **H** implication and negation are as follows.

$\Rightarrow$	F	$\perp$	T
F	T	T	T
$\perp$	F	T	T
T	F	$\perp$	T

$\neg$	F	$\perp$	T
	T	F	F

For **Steps 2 - 4** of the definition 7 we adopt definitions established for **L** semantics. For example, we define the notion of **H** tautology as follows.

Definition 36 (H Tautology)

For any formula $A \in \mathcal{F}$,
*A is a **H** tautology if and only if $v^*(A) = T$, for all $v : VAR \longrightarrow \{F, \perp, T\}$, i.e. $v \models_{\mathbf{H}} A$ for all v. We write*

$$\models_{\mathbf{H}} A$$

*to denote that a formula A is an **H** tautology.*

We leave it as an exercise to the reader to prove, in the same way as in case of classical semantics (section3) the following theorems that justify the truth table method of verification and the decidability theorem for **K**.

Theorem 16 (H Truth Tables)

For any formula $A \in \mathcal{F}$,
$\models_{\mathbf{H}} A$ if and only if $v_A \models_{\mathbf{H}} A$ for all $v_A : VAR_A \longrightarrow \{T, \perp, F\}$, i.e.
$\models_{\mathbf{H}} A$ if and only if all v_A are restricted models for A.

Theorem 17 (H Decidability)

*For any formula $A \in \mathcal{F}$, one has examine at most 3^{VAR_A} truth assignments $v_A : VAR_A \longrightarrow \{F, \perp, T\}$ in order to decide whether $\models_{\mathbf{H}} A$, or $\not\models_{\mathbf{H}} A$, i.e. the notion of **H** tautology is decidable.*

We denote by **HT** the set of all **H** tautologies, i.e.

$$\mathbf{HT} = \{A \in \mathcal{F} : \models_{\mathbf{H}} A\}.$$

The following fact establishes relationship between classical and **H** tautologies.

Theorem 18

*Let **HT**, **LT**, **T** denote the sets of all **H**, **L**, and classical tautologies, respectively. Then the following relationship holds.*

$$\mathbf{HT} \neq \mathbf{LT}, \quad \mathbf{HT} \neq \mathbf{T}, \quad and \quad \mathbf{HT} \subset \mathbf{T}. \tag{37}$$

Proof
A formula $(\neg a \cup a)$ a classical tautology and not an **H** tautology. Take any truth assignment $v : VAR \longrightarrow \{F,\ \perp,\ T\}$ such that $v(a) =\perp$. We evaluate is $v^*((\neg a \cup a) = \neg \perp \cup \perp = F \cup \perp = \perp$ This proves that $(\neg a \cup a) \notin \mathbf{HT}$ and hence $\mathbf{HT} \neq \mathbf{T}$. Directly from the definition of **H** connectives we get that if we restrict the values of the functions defining them T and F only, we get the functions defining the classical connectives. Hence for any formula $A \in \mathbf{TH}$ we have that $A \in \mathbf{TH}$ and $\mathbf{LT} \subset \mathbf{T}$. A formula $(\neg\neg a \Rightarrow a)$ is a **L** tautology and not an **H** tautology by easy evaluation as presented in example 23 and (38). This proves $\mathbf{HT} \neq \mathbf{LT}$.

Exercise 23

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 $v^*(((b \Rightarrow a) \Rightarrow (a \Rightarrow \neg c)) \cup (a \Rightarrow b))$. *You can use a short hand notation.*

Solution
By definition of **H** connectives we have that for any v, $v^*((a \cap b) \Rightarrow (a \Rightarrow c)) =\perp$ if and only if $a \cap b) = T$ and $(a \Rightarrow c) =\perp$ if and only if $a = T, b = T$ and $(T \Rightarrow c) =\perp$ if and only if $c =\perp$. Hence $v^*((a \cap b) \Rightarrow (a \Rightarrow c)) =\perp$ if and only if $a = T,\ b = T, c =\perp$. We evaluate $v^*(((b \Rightarrow a) \Rightarrow (a \Rightarrow \neg c)) \cup (a \Rightarrow b)) = (((T \Rightarrow T) \Rightarrow (T \Rightarrow \neg \perp)) \cup (T \Rightarrow T)) = ((T \Rightarrow (T \Rightarrow F)) \cup T) = T$.

Exercise 24

We know that the following formulas are basic classical tautologies

$$\models (a \cup \neg a), \qquad \models (\neg\neg a \Rightarrow a), \qquad \models ((a \Rightarrow b) \Rightarrow (\neg a \cup b)). \tag{38}$$

Use the **H** *semantics to prove that none of them is intuitionostic tautology.*

Solution Any $v : VAR \longrightarrow \{F,\ \perp,\ T\}$ such that $v(a) = v(b) =\perp$ is an **H** counter model for all of the formulas. We evaluate (in shorthand notation) it as follows. $\perp \cup \neg \perp = \perp \cup F = \perp \neq T$, $\neg\neg \perp \Rightarrow \perp = \neg F \Rightarrow \perp = T \Rightarrow \perp = \perp \neq T$,

$(\perp \Rightarrow \perp) \Rightarrow (\neg \perp \cup \perp) = T \Rightarrow (\neg \perp \cup \perp) = T \Rightarrow (F \cup \perp) = T \Rightarrow \perp = \perp \neq T$. We hence proved by the fact *"if a given formula A is not the* **H** *semantics tautology, it is not intuitionistic tautology"* that none of classical tautologies (38) is neither intuitionostic nor **H** tautology.

The **H** semantics can serve as a tool of proving that some formulas are not intutionistic tautologies, but it is not a universal one

Example 22

We know that the classical tautology $(\neg(a \cap b) \Rightarrow (\neg a \cup \neg b))$ *is not intuitionistic tautology, but nevertheless* $\models_{\mathbf{H}} (\neg(a \cap b) \Rightarrow (\neg a \cup \neg b))$.

Proof
We use the Proof by Contradiction Method (section **??**) and shorthand notation. Assume that $\not\models_{\mathbf{H}} (\neg(a \cap b) \Rightarrow (\neg a \cup \neg b))$. Let $v : VAR \longrightarrow \{F, \ \bot, \ T\}$ such that $v^*((\neg(a \cap b) \Rightarrow (\neg a \cup \neg b))) \neq T$. We have to consider two cases: **c1** $v^*((\neg(a \cap b) \Rightarrow (\neg a \cup \neg b))) =\bot$ and **c2** $v^*((\neg(a \cap b) \Rightarrow (\neg a \cup \neg b))) = F$. If we get a contradiction in *both cases* we have proved $\models_{\mathbf{H}} (\neg(a \cap b) \Rightarrow (\neg a \cup \neg b))$.

Consider case **c1**. By definition of $\Rightarrow$ we have that $v^*((\neg(a \cap b) \Rightarrow (\neg a \cup \neg b))) =\bot$ if and only if $\neg(a \cap b) = T$ and $\neg a \cup \neg b =\bot$ if and only if $a \cap b = F$ and $\neg a \cup \neg b =\bot$. Let's look $\neg a \cup \neg b =\bot$. This is possible in 3 cases. 1. $\neg a =\bot$ and $\neg b =\bot$. Contradiction with the definition of $\bot$ as $\neg x \neq\bot$ for all $x \in \{F, \bot, T\}$. 2. $\neg a =\bot$ and $\neg b = F$. Contradiction with the definition of $\bot$. 3. $\neg a = F$ and $\neg b =\bot$. Contradiction with the definition of $\bot$. This proves that case **c1** always leads to contradiction.

Consider case **c2**. By definition of $\Rightarrow$ we have that $v^*((\neg(a \cap b) \Rightarrow (\neg a \cup \neg b))) = F$ if and only if 1. $\neg(a \cap b) =\bot$, $\neg a \cup \neg b = F$. Contradiction. 2. $\neg(a \cap b) = T$ and $\neg a \cup \neg b = F$ if and only if $a \cap b = F$ and $\neg a \cup \neg b = F$. Observe that $a \cap b = F$ in 3 cases. Two involve only T, F and we get a contradiction as in classical case (our formula is classical tautology). We have hence to consider only the cases when $a =\bot, b = F$ and $a = F, b =\bot$. They both lead to the contradiction with $\neg a \cup \neg b = F$. This proves that case **c2** always leads to contradiction and it ends the proof.

We can of course also use the Truth Tables Method that involves listing and evaluating all of $2^3 = B$ restricted truth assignments.

Kleene Semantics K

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

Motivation

In Kleene's semantics the third logical value $\bot$, intuitively, represents *undecided*. Its purpose is to signal a state of partial ignorance. A sentence a is assigned a value $\bot$ just in case it is not *known* to be either true of 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 $\perp$, but it is certainly either true of false and $\perp$ represents our ignorance rather then total unknown.

K Connectives

We adopt the same language as in a case of classical, Łukasiewicz's **L**, and Heyting **H** semantics, i.e. $\mathcal{L} = \mathcal{L}_{\{\neg, \Rightarrow, \cup, \cap\}}$.

We assume, as before, that $\{F < \perp < T\}$. The connectives $\neg$, $\cup$, $\cap$ of **K** are defined as in **L, H** semantics. They are functions defined by formulas $x \cup y = max\{x, y\}$, $x \cap y = min\{x, y\}$, for any $x, y \in \{F, \perp, T\}$, and

$$\neg \perp = \perp, \quad \neg F = T, \quad \neg T = F.$$

The **K** implication is defined by the same formula as the classical, i.e.

$$x \Rightarrow y = \neg x \cup y. \tag{39}$$

for any $x,\ y \in \{F, \perp, T\}$.

The connectives truth tables for the **K** negation, disjunction and conjunction are the same as the corresponding tables for **L, H** and **K** implication table is as follows.

$\Rightarrow$	F	$\perp$	T
F	T	T	T
$\perp$	$\perp$	$\perp$	T
T	F	$\perp$	T

For **Steps 2 - 4** of the definition of **K** semantics we follow the general **M** semantics definition 7, or adopt its particular case of **L** semantics definition. For example, we define the notion of **K** tautology as follows.

Definition 37 (K Tautology)

For any formula $A \in \mathcal{F}$,
*A is a **K** tautology if and only if $v^*(A) = T$, for all truth assignments $v : VAR \longrightarrow \{F, \perp, T\}$, i.e. $v \models_{\mathbf{K}} A$ for all v.*

We write

$$\models_{\mathbf{K}} A$$

to denote that A is a **K** tautology. We prove, in the same way as in case of **L** semantics the following theorems that justify truth table method of verification and decidability theorem for **K**.

Theorem 19 (K Truth Tables)

For any formula $A \in \mathcal{F}$,
$\models_{\mathbf{K}} A$ *if and only if* $v_A \models_{\mathbf{K}} A$ *for all* $v_A : VAR_A \longrightarrow \{T, \perp, F\}$, *i.e.*
$\models_{\mathbf{K}} A$ *if and only if all* v_A *are restricted models for* A.

Directly from Theorem 19 we get that the notion of **K** propositional tautology is decidable, i.e. that the following holds.

Theorem 20 (K Decidability)

For any formula $A \in \mathcal{F}$, *one has examine at most* 3^{VAR_A} *truth assignments* $v_A : VAR_A \longrightarrow \{F, \perp, T\}$ *in order to decide whether* $\models_{\mathbf{L}} A$, *or* $\not\models_{\mathbf{K}} A$, *i.e. the notion of* **K** *tautology is decidable.*

We write

$$\mathbf{KT} = \{A \in \mathcal{F} : \models_K A\}$$

to denote the set of all **K** tautologies. The following establishes relationship between **L**, **K**, and classical tautologies.

Theorem 21

Let **LT**, **T**, **KT** *denote the sets of all* **L**, *classical, and* **K** *tautologies, respectively. Then the following relationship holds.*

$$\mathbf{LT} \neq \mathbf{KT}, \quad \mathbf{KT} \neq \mathbf{T}, \text{ and } \mathbf{KT} \subset \mathbf{T}. \tag{40}$$

Proof
Obviously $\models (a \Rightarrow a)$ and also by (33) $\models_{\mathbf{L}} (a \Rightarrow a)$. Consider now any v such that $v(a) = \perp$. We evaluate in **K** semantics $v^*(a \Rightarrow a) = v(a) \Rightarrow v(a) = \perp \Rightarrow \perp = \perp$. This proves that $\not\models_{\mathbf{K}} (a \Rightarrow a)$ and hence the first two relationships in (40) hold. The third one follows directly from the the fact that, as in the **L** case, if we restrict the functions defining **K** connectives to the values T and F only, we get the functions defining the classical connectives.

Exercise 25

We know that formulas $((a \cap b) \Rightarrow a), (a \Rightarrow (a \cup b)), (a \Rightarrow (b \Rightarrow a))$ *are classical tautologies. Show that none of them is* **K** *tautology.*

Solution Consider any v such that $v(a) = v(b) = \perp$. We evaluate (in short hand notation) $v^*(((a \cap b) \Rightarrow a) = (\perp \cap \perp) \Rightarrow \perp = \perp \Rightarrow \perp = \perp \Rightarrow \perp \perp$, $v^*((a \Rightarrow (a \cup b))) = \perp \Rightarrow (\perp \cup \perp) = \perp \Rightarrow \perp = \perp$, and $v^*((a \Rightarrow (b \Rightarrow a))) = (\perp \Rightarrow (\perp \Rightarrow a \perp) = \perp \Rightarrow \perp = \perp$. This proves that v such that $v(a) = v(b) = \perp$ is a counter model for all of them.

We generalize this example and prove that in fact a similar truth assignment can serve as a counter model for not only any classical tautology, but also for any formula A of $\mathcal{L}_{\{\neg,\Rightarrow,\cup,\cap\}}$.

Theorem 22

For any formula $A \in \mathcal{F}$, $\not\models_{\mathbf{K}} A$, i.e. the set of all **K** *tautologies is empty. We write it as*

$$\mathbf{KT} = \emptyset.$$

Proof
We show that a truth assignment $v : VAR \longrightarrow \{F, \perp, T\}$, such that $v(a) = \perp$ for all $a \in VAR$ is a counter model for any $A \in \mathcal{F}$. We carry the proof the by mathematical induction over the degree $d(A)$ of the formula A.

Base Case: n=1 i.e. $d(A) = 1$. In this case we have that $A = \neg a$ for any $a \in VAR$, or $A = (a \circ b)$ for $a, b \in VAR, \circ \in \{\cup, \cap, \Rightarrow\}$.

We evaluate: $v^*(A) = v^*(\neg a) = \neg v^*(a) = \neg \perp = \perp$, $v^*(a \circ b) = v^*(a) \circ v^*(b) = \perp \circ \perp = \perp$. This proves that the *Base Case* holds.

Inductive assumption: $v^*(B) = \perp$ for all B such that $d(B) = k$ and $1 \leq k < n$.
Inductive thesis: $v^*(A) = \perp$ for any A such that $d(A) = n$.

Let A be such that $d(A) = n$. We have two cases to consider.

Case 1. $A = \neg B$, so $d(B) = n - 1 < n$. By inductive assumption $v^*(B) = \perp$. Hence $v^*(A) = v^*(\neg B) = \neg v^*(B) = \neg \perp = \perp$ and inductive thesis holds.

Case 2. $A = (B \circ C)$ for $B, C \in \mathcal{F}, \circ \in \{\cup, \cap, \Rightarrow\}$ (and $d(A) = n$). Let $d(B) = k_1, d(C) = k_2$. Hence $d(A) = d(B \circ C) = k_1 + k_2 + 1 = n$. We get that $k_1 + k_2 = n - 1 < n$. From $k_1 + k_2 < n$ we get that $k_1 < n$ and $k_2 < n$. Hence by inductive assumption $v^*(B) = \perp$ and $v^*(C) = \perp$. We evaluate: $v^*(A) = v^*(B \circ C) = v^*(B) \circ v^*(C) = \perp \circ \perp = \perp$. This ends the proof.

Observe that the theorem 22 does not invalidate relationships (40). They become now perfectly true statements

$$\mathbf{LT} \neq \emptyset, \quad \mathbf{T} \neq \emptyset, \text{ and } \emptyset \subset \mathbf{T}.$$

But when we develop a logic by defining its semantics we must make sure for semantics to be such that the logic has a non empty set of its tautologies. The semantics **K** an example of a correctly and carefully) defined semantics that is not well definedin terms of the definition 14. We write is as separate fact.

Fact 6

The Kleene semantics **K** *is* **not well defined***.*

K semantics also provides a justification for a need of introducing a distinction between correctly and well defined semantics. This is the main reason why it is included here.

Bochvar semantics B

Bochvar's 3-valued logic was directly inspired by considerations relating to semantic paradoxes. Here is the motivation for definition of its semantics.

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. There have been many proposals relating to how one may deal with semantic paradoxes. Bohvar's proposal adopts a strategy of a change of logic. According to Bochvar, such sentences are neither true of false but rather *paradoxical* or *meaningless*. The semantics follows the principle that the third logical value, denoted now by m is in some sense "infectious"; if one one component of the formula is assigned the value m then the formula is also assigned the value m.

Bohvar also adds an one argument *assertion operator* S that asserts the logical value of T and F, i.e. $SF = F$, $ST = T$ and it asserts that meaningfulness is false, i.e $Sm = F$.

Language $\mathcal{L}_{\mathbf{B}}$

The language of **B** semantics differs from all previous languages in that it contains an extra one argument assertion connective S added to the usual set $\{\neg, \Rightarrow, \cup, \cap\}$ of the language $\mathcal{L} = \mathcal{L}_{\{\neg,S,\Rightarrow,\cup,\cap\}}$ of all previous semantics.

$$\mathcal{L}_{\mathbf{B}} = \mathcal{L}_{\{\neg,S,\Rightarrow,\cup,\cap\}}. \tag{41}$$

The set LV of logical values is $\{T,\ m,\ F\}$. T is the distinguished value.

B Connectives

We define the connectives of $\mathcal{L}_{\mathbf{B}}$ the functions defined in the set $\{F, mT\}$ by the following truth tables.

B Connectives Truth Tables

$\neg$	F	m	T
	T	m	F

$\cap$	F	m	T
F	F	m	F
m	m	m	m
T	F	m	T

$\cup$	F	m	T
F	F	m	T
m	m	m	m
T	T	m	T

$\Rightarrow$	F	m	T
F	T	m	T
m	m	m	m
T	F	m	T

S	F	m	T
	F	F	T

For all other steps of definition of **B** semantics we follow the standard way established for extensional **M** semantics, we did in all previous cases. In particular we define the notion of **K** tautology as follows.

Definition 38

A formula A of $\mathcal{L}_{\mathbf{B}}$ is a **B** *tautology if and only if $v^*(A) = T$, for all $v : VAR \longrightarrow \{F, m, T\}$, i.e. if all variable assignments v are* **B** *models for A.*

We write

$$\models_{\mathbf{B}} A$$

to denote that A is an **B** tautology.

We, prove, in the same way as for all previous logics semantics, the following theorems that justify the truth table method of verification and decidability for **B** tautologies.

Theorem 23 (B Truth Tables)

For any formula A of $\mathcal{L}_{\mathbf{B}}$,
$\models_{\mathbf{B}} A$ *if and only if* $v_A \models_{\mathbf{B}} A$ *for all* $v_A : VAR_A \longrightarrow \{F, m, T\}$.

Theorem 24 (Decidability)

For any formula A of $\mathcal{L}_{\mathbf{B}}$, one has examine at most 3^{VAR_A} truth assignments $v : VAR_A \longrightarrow \{F, m, T\}$ in order to decide whether $\models_{\mathbf{B}} A$, or $\not\models_{\mathbf{B}} A$, i.e. the notion of **B** *tautology is decidable.*

Let denote by $\mathcal{F}_{\mathbf{B}}$ the set of formulas of the language $\mathcal{L}_{\mathbf{B}}$ and by **BT** the set of all **B** tautologies:

$$\mathbf{BT} = \{A \in \mathcal{F}_{\mathbf{B}} : \models_{\mathbf{B}} A\}.$$

Which formulas (if any) are the **B** tautologies is more complicated to determine then in the case previous semantics because we have the following Fact 7.

Fact 7

For any formula $A \in \mathcal{F}_{\mathbf{B}}$ which do not contain a connective S, i.e. for any formula A of the language $\mathcal{L}_{\{\neg, \Rightarrow, \cup, \cap\}}$, $\not\models_{\mathbf{B}} A$.

Proof We show that a truth assignment $v : VAR \longrightarrow \{F, m, T\}$, such that $v(a) = m$ for all $a \in VAR$ is a counter model for any $A \in \mathcal{F}$. The proof the by mathematical induction over the degree $d(A)$ of the formula A is similar to the proof of Theorem 22 and is left to the reader as an exercise.

By the Fact 7 for a formula to be considered to be a **B** tautology, it must contain the connective S. We get by easy evaluation that $\models_{\mathbf{B}} (Sa \cup \neg Sa)$. This proves that $\mathbf{BT} \neq \emptyset$ and the **B** semantics is well defined by definition 14. Of course not all formulas containing the connective S are **B** tautologies, for example

$$\not\models_{\mathbf{B}} (a \cup \neg Sa), \;\; \not\models_{\mathbf{B}} (Sa \cup \neg a), \;\; \not\models_{\mathbf{B}} (Sa \cup S\neg a),$$

as any truth assignment v, such that $v(a) = m$ is a counter model for all of them, because $m \cup x = m$ for all $x \in \{F, m, T\}$ and $Sm \cup S\neg m = F \cup Sm = F \cup F = F$.

6 M Tautologies, M Consistency, and M Equivalence of Languages

The classical truth tables verification method a and classical decidability theorem hold in a proper form in all of **L. H, K** and **B** semantics as it was discussed separately for each of them. We didn't discuss other classical tautologies verification methods of substitution and generalization. We do it now in a general and unifying way in a case of an extensional **M** semantics.

Given an extensional semantics **M** defined for a propositional language $\mathcal{L}_{CON}$ with the set $\mathcal{F}$ of formulas and a finite, non empty set LV of logical values.

We introduce, as we did in classical case a notion of a restricted model (definition 26) and prove, in a similar way as we proved theorem8 the following theorem that justifies the correctness of the **M** truth tables tautologies verification method.

Theorem 25 (M Truth Tables)

For any formula $A \in \mathcal{F}$,
$\models_{\mathbf{M}} A$ if and only if $v_A \models_{\mathbf{M}} A$ for all $v_A : VAR_A \longrightarrow LV$, i.e.
$\models_{\mathbf{M}} A$ if and only if all v_A are restricted models for A.

M Truth Table Method

A verification method, called a **M** truth table method consists of examination, as in the classical case, for any formula A, all possible **M** truth assignments restricted to A. By theorem 25 we have to perform at most $|LV|^{|VAR_A|}$ steps. If we find a truth assignment which evaluates A to a value different then T, we stop the process and give answer: $\not\models_{\mathbf{M}} A$. Otherwise we continue. If all **M** truth assignments restricted to A evaluate A to T, we give answer: $\models_{\mathbf{M}} A$.

Example 23

Consider a formula $(\neg\neg a \Rightarrow a)$ *and* **H** *semantics. We evaluate*

v	a	$v^*(A)$ *computation*	$v^*(A)$
v_1	T	$\neg\neg T \Rightarrow T = \neg F \Rightarrow T = F \Rightarrow T = T$	T
v_2	$\perp$	$\neg\neg \perp \Rightarrow \perp = \neg F \Rightarrow \perp = T \Rightarrow \perp = \perp$	$\perp$

It proves that $\not\models_{\mathbf{H}} (\neg\neg a \Rightarrow a)$.

Example 24

Consider a formula $(\neg\neg a \Rightarrow a)$ *and* **L** *semantics. We evaluate*

v	a	$v^*(A)$ *computation*	$v^*(A)$
v_1	T	$\neg\neg T \Rightarrow T = \neg F \Rightarrow T = F \Rightarrow T = T$	T
v_2	$\perp$	$\neg\neg \perp \Rightarrow \perp = \neg \perp \Rightarrow \perp = \perp \Rightarrow \perp = T$	T
v_3	F	$neg \neg F \Rightarrow F = \neg T \Rightarrow F = F \Rightarrow F = T$	T

It proves that $\models_{\mathbf{L}} (\neg\neg a \Rightarrow a)$.

We also proved that the set **HT** of all **H** tautologies is different from the set set **LT** of all **L** tautologies, i.e.

$$\mathbf{LT} \neq \mathbf{HT} \tag{42}$$

Directly from Theorem 25 and the above we get that the notion of **M** propositional tautology is decidable, i.e. that the following holds.

Theorem 26 (M Decidability)

For any formula $A \in \mathcal{F}$, *one has examine at most* $|LV|^{VAR_A}$ *truth assignments* $v_A : VAR_A \longrightarrow LV$ *in order to decide whether* $\models_{\mathbf{M}} A$, *or* $\not\models_{\mathbf{M}} A$, *i.e. the notion of* **M** *tautology is decidable.*

M Proof by Contradiction Method

In this method, in order to prove that $\models_{\mathbf{M}} A$ we proceed as follows.

We assume that $\not\models A$. We work with this assumption. If we get a **contradiction**, we have proved that $\not\models_{\mathbf{M}} A$ is impossible. We hence proved $\models_{\mathbf{M}} A$. If we do not get a contradiction, it means that the assumption $\not\models_{\mathbf{M}} A$ is true, i.e. we have proved that A is not A **M** tautology.

Observe that correctness of his method is based on a correctness of classical reasoning. Its correctness is based on the *Reductio ad Absurdum* classical tautology $\models ((\neg A \Rightarrow (B \cap \neg B)) \Rightarrow A)$. The contradiction to be obtained follows from the properties of the **M** semantics under consideration.

Substitution Method

The Substitution Method allows us to obtain, as in a case of classical semantics new **M** tautologies from formulas already proven to be **M** tautologies. The theorem 27 and its proof is a straightforward modification of the classical proof (theorem 27) and we leave it as an exercise to the reader. It assesses the validity of the substitution method. In order to formulate and prove it we first remind of the reader of needed notations.

Let $A \in \mathcal{F}$ be a formula and $VAR_A = \{a_1, a_2, ...a_n\}$ be the set of all propositional variables appearing in A. We will denote it by $A(a_1, a_2, ...a_n)$. Given a formula $A(a_1, a_2, ...a_n)$, and $A_1, ...A_n$ be any formulas. We denote by

$$A(a_1/A_1, ..., a_n/A_n)$$

the result of simultaneous replacement (substitution) in $A(a_1, a_2, ...a_n)$ variables $a_1, a_2, ...a_n$ by formulas $A_1, ...A_n$, respectively.

Theorem 27

For any formulas $A(a_1, a_2, ...a_n),\ A_1,\ \dots\ ,\ A_n \in \mathcal{F}$,

If $\models_{\mathbf{M}} A(a_1, a_2, ...a_n)$ *and* $B = A(a_1/A_1, ..., a_n/A_n)$, *then* $\models_{\mathbf{M}} B$.

We have proved (exercise 24) that the formula $D(a) = (\neg\neg a \Rightarrow a)$ is **L** tautology. By the above theorem 27 we get that $D(a/A) = (\neg\neg A \Rightarrow A)$ is also **L** tautology for any formula $A \in \mathcal{F}$. We hence get the following.

Fact 8

For any $A \in \mathcal{F}$, $\models_{\mathbf{L}} (\neg\neg A \Rightarrow A)$.

M Generalization Method

In this method we represent, if it is possible, a given formula as a particular case of some simpler *general formula*. Hence the name Generalization Method. We then use other methods to examine the simpler formula thus obtained.

Exercise 26

Prove that

$\models_{\mathbf{L}} (\neg\neg(\neg((a \cap \neg b) \Rightarrow ((c \Rightarrow (\neg f \cup d)) \cup e)) \Rightarrow ((a \cap \neg b) \cap (\neg(c \Rightarrow (\neg f \cup d)) \cap \neg e))) \Rightarrow (\neg((a \cap \neg b) \Rightarrow ((c \Rightarrow (\neg f \cup d)) \cup e)) \Rightarrow ((a \cap \neg b) \cap (\neg(c \Rightarrow (\neg f \cup d)) \cap \neg e))))$.

Solution

Observe that our formula is a particular case of a more general formula $(\neg\neg A \Rightarrow A)$ for $A = (\neg((a \cap \neg b) \Rightarrow ((c \Rightarrow (\neg f \cup d)) \cup e)) \Rightarrow ((a \cap \neg b) \cap (\neg(c \Rightarrow (\neg f \cup d)) \cap \neg e)))$ and by fact 8 our formula is proved to be **L** tautology.

One of the most important notions for any logic are notions of consistency and inconsistency. We introduced and discussed them in case of classical semantics in section 3. We formulate them now for any **M** extensional semantics and examine them in cases of **L** and **H** semantics.

Consider $\mathcal{L}_{CON}$ and let $\mathcal{S} \neq \emptyset$ be any non empty set of formulas of $\mathcal{L}_{CON}$. Let **M** be an extensional semantics for $\mathcal{L}_{CON}$. We adopt the following definitions.

Definition 39

A truth truth assignment $v : VAR \longrightarrow LV$ *is* **M model** *for the set* $\mathcal{G}$ *of formulas if and only if* $v \models_{\mathbf{M}} A$ *for all formulas* $A \in \mathcal{G}$. *We denote it by* $v \models \mathcal{G}$.

Definition 40

A set $\mathcal{G} \subseteq \mathcal{F}$ *is called* ***M consistent*** *if and only if there is* $v : VAR \longrightarrow LV$, *such that* $v \models \mathcal{G}$.

Otherwise the set $\mathcal{G}$ *is called* ***M inconsistent***.

Observe that in this case the inconsistency definition is stated as follows.

Definition 41

A set $\mathcal{G} \subseteq \mathcal{F}$ *is called* ***M inconsistent*** *if and only if for all* $v : VAR \longrightarrow LV$, $v^*(A) \neq T$ *holds for all formulas* $A \in \mathcal{G}$.

Plainly speaking, a set $\mathcal{G}$ is consistent , if it has a model, and is inconsistent if it does not have a model under a semantic **M**.

Exercise 27

Prove that the set

$$\mathcal{G} = \{((a \cap b) \Rightarrow b),\ (a \cup b), \neg a\}$$

is **L**, **H**, *and* **K** *consistent.*

Solution Let v be a truth assignment $v : VAR \longrightarrow \{T, \perp, F\}$. By the definition 39, $v \models \{((a \cap b) \Rightarrow b),\ (a \cup b), \neg a\}$ if and only if $v^*(((a \cap b) \Rightarrow b)) = T,\ v^*((a \cup b) = T)$, and $v^*(\neg a) = T$. Observe that $\models ((a \cap b) \Rightarrow b)$, so we have to find v, such that $v^*((a \cup b)) = T,\ v^*(\neg a) = T$. This holds if and only if $v(a) = F$ and $F \cup v(b) = T$, i.e. if and only if $v(a) = F$ and $v(b) = T$. Observe that the semantics **L**, **H**, and **K** are all defined in such a way that if we restrict the functions defining their connectives to the values T and F only, we get the functions defining the classical connectives. This proves that any v such that $v(a) = F$ and $v(b) = T$ is a **L**, **H**, and **K** model for $\mathcal{G}$.

The same argument prove the following general fact.

Fact 9

For any non empty set $\mathcal{G}$ of formulas of a language $\mathcal{L}_{\{\neg,\Rightarrow,\cup,\cap\}}$ the following holds. If $\mathcal{G}$ is consistent under classical semantics, then it is **L**, **H**, *and* **K** *consistent.*

Exercise 28

Give an example of an infinite set $\mathcal{G}$ of formulas of a language $\mathcal{L}_{\mathbf{B}} = \mathcal{L}_{\{\neg,S,\Rightarrow,\cup,\cap\}}$ that is **L**, **H**, **K** *and* **B** *consistent.*

Solution

Observe that for the set $\mathcal{G}$ to be considered to be **L**, **H**, **K** consistent its formulas must belong to the language sub language $\mathcal{L}_{\{\neg,\Rightarrow,\cup,\cap\}}$ of the language $\mathcal{L}_{\mathbf{B}}$. So for example let's consider a set

$$\mathcal{G} = \{(a \cup \neg b) : \quad a, b \in VAR\}.$$

$\mathcal{G}$ is infinite since the set VAR is infinite. Let $v : VAR \longrightarrow \{F, m, T\}$ be such that $v(a) = T, v(b) = T$, we have $v^*(a \cup b) = v(a) \cup v(b) = T \cup T$ by the **L**, **H**, **K** and **B** definition of $\cup$. This proves that $\mathcal{G}$ is **L**, **H**, **K** and **B** consistent.

Exercise 29

Prove that the set

$$\mathcal{G} = \{(a \cap \neg a) : \quad a \in VAR\}$$

is **L**, **H, K**, *and* **B** *inconsistent..*

Solution

We know that the set $\mathcal{G}$ is classically inconsistent, i.e. $v^*((a \cap \neg a)) \neq T$ for all $v : VAR \longrightarrow \{F, T\}$ under classical semantics. It also holds for We have to show that it also holds for **L**, **H**, **K** and **B** semantics when we restrict the functions defining their connectives to the values T and F only. In order to prove inconsistency under **L**, **H, K**, semantics we have to show that $v^*((a \cap \neg a)) \neq T$ for all $v : VAR \longrightarrow \{F, \perp, T\}$ under the respective semantics, i.e. we have to evaluate additional case $v(a) = \perp$ in all of them. Observe that negation $\neg$ is defined in all of them as $\neg \perp = \perp$, and $v^*((a \cap \neg a)) = \perp \cap \neg \perp = \perp \cap \perp = \perp \neq T$. This proves that $\mathcal{G}$ is **L**, **H**, and **K** inconsistent. The case of **B** semantics is similar, except that now we consider all $v : VAR \longrightarrow \{F, m, T\}$ and the additional case is $v(a) = m$. By definition $\neg m = m$ and $v^*((a \cap \neg a)) = m \cap m = m \neq T$.

The examples of **B** consistent, or inconsistent sets $\mathcal{G}$ in exercise 28 and exercise 29 were restricted to formulas from $\mathcal{L}_{\mathbf{B}} = \mathcal{L}_{\{\neg, S, \Rightarrow, \cup, \cap\}}$ that did not include the connective S. In this sense they were not characteristic to the semantics **B**. We pose hence a natural question whether such examples exist.

Exercise 30

*Give an example of sets $\mathcal{G}_1, \mathcal{G}_2$ containing some formulas that include the S connective of the language $\mathcal{L}_{\mathbf{B}} = \mathcal{L}_{\{\neg, S, \Rightarrow, \cup, \cap\}}$ such that $\mathcal{G}_1$ is **B** consistent and $\mathcal{G}_2$ is **B** inconsistent*

Solution
There are many such sets $\mathcal{G}$, here are just two simple examples.

$$\mathcal{G}_1 = \{(Sa \cup S\neg a), \quad (a \Rightarrow \neg b), \quad S\neg(a \Rightarrow b), \quad (b \Rightarrow Sa)\}$$

$$\mathcal{G}_2 = \{Sa, \quad (a \Rightarrow b), \quad (\neg b \cup, \quad S\neg a\}.$$

Any $v : VAR \longrightarrow \{F, m, T\}$, such that $v(a) = T, v(b) = F$ is a **b** model for $\mathcal{G}_1$, i.e. $\mathcal{G}_1$ is consistent. Assume now that there is $v : VAR \longrightarrow \{F, m, T\}$, such that $v \models_{\mathbf{B}} \mathcal{G}_2$. In particular $v^*(Sa) = T$. This is possible if and only if $v(a) = T$, then $v^*(S\neg a) = SF = F$. This contradicts $v \models_{\mathbf{B}} \mathcal{G}_2$. Hence $\mathcal{G}_2$ is **B** inconsistent.

We introduce, as we did in classical case a notion of a contradiction as follows.

Definition 42

*Let **M** be an extensional semantics for $\mathcal{L}_{CON}$. We say that a formula A is a **M** contradiction if it doesn't have a **M** model.*

Example 25

*A formula $(Sa \cap S\neg a)$ of $\mathcal{L}_{\mathbf{B}} = \mathcal{L}_{\{\neg, S, \Rightarrow, \cup, \cap\}}$ is a **B** contradiction.*

Proof
Assume that there is v, such that $v \models (Sa \cap S\neg a)$, i.e. $v^*((Sa \cap S\neg a)) = T$ if

and only if (shorthand notation) $Sa = T$ and $S\neg a = T$. But $Sa = T$ if and only of $a = T$. In this case $S\neg T = SF = F \neq T$. This contradiction proves that such v does not exist, i.e. that for all v, $v \not\models (Sa \cap S\neg a)$.

This also justifies the following.

Example 26 *The set $\mathcal{G} = \{(Sa \cap S\neg a) : \quad a \in VAR\}$ is an countably infinite* **B** *inconsistent set.*

Here is a simple problem asking to create your own, specific **M** semantics fulfilling certain specifications. This semantics is different from all of previous semantics defined and examined. We also ask to examine some of its properties, including **M** consistency and **M** inconsistency. We provide an example two different semantics. We encourage the reader to come up with his/hers own and to write down formally its full definition according to definition 7 as it was done in the case of **L** semantics.

Review Problem

Part 1. Write the following natural language statement:

One likes to play bridge, or from the fact that the weather is good we conclude the following: one does not like to play bridge or one likes not to play bridge

as a formula of 2 different languages

1. Formula $A_1 \in \mathcal{F}_1$ of a language $\mathcal{L}_{\{\neg,\ L,\ \cup,\ \Rightarrow\}}$, where LA represents statement "one likes A", "A is liked".

2. Formula $A_2 \in \mathcal{F}_2$ of a language $\mathcal{L}_{\{\neg,\ \cup,\ \Rightarrow\}}$.

Part 2. Define formally, following all steps of the defnition 7, a 3 valued extensional semantics **LK** for the language $\mathcal{L}_{\{\neg,\ \mathbf{L},\ \cup,\ \Rightarrow\}}$ under the following assumptions.

s1 We assume that the third value is denoted by $\perp$ is *intermediate* between designated value T and F, i.e. that $F < \perp < T$.

s2 We model a situation in which one "likes" only truth, represented by T; i.e. in which

$$LT = T, \quad L\perp = F, \quad LF = F.$$

s3 The connectives $\neg,\ \cup,\ \Rightarrow$ can be defined as one wishes, but they have to be defined in such a way to make sure that always "one likes A or does not like A", i.e. it must be assured that $\models_{\mathbf{LK}} \ (LA \cup \neg LA)$.

Part 3.

1. Verify whether the formulas A_1 and A_2 from the **Part 1.** have a model/ counter model under your semantics **LK**. You can use shorthand notation.

2. Verify whether the following set G is **LK** consistent. You can use **shorthand notation**.

$$\mathcal{G} = \{La, \quad (a \cup \neg Lb), \quad (a \Rightarrow b), \; b \; \}.$$

3. Give an example on an infinite, **LK** consistent set of formulas of the language $\mathcal{L}_{\{\neg,\ L,\ \cap,\ \cup,\ \Rightarrow\}}$. Some formulas must contain the connective L.

Review Problem Solutions

Part 1 Solution

1. We translate the statement into a formula $A_1 \in \mathcal{F}_1$ of a language $\mathcal{L}_{\{\neg,\ \mathbf{L},\ \cap,\ \cup,\ \Rightarrow\}}$ as follows.

Propositional variables: a, b, where a denotes statement: *play bridge*, b denotes a statement: *the weather is good.*

$$A_1 = (\mathbf{L}a \cup (b \Rightarrow (\neg \mathbf{I}a \cup \mathbf{L}\neg a))).$$

2. We translate our statement into a formula $A_2 \in \mathcal{F}_2$ of a language $\mathcal{L}_{\{\neg,\ \cup,\ \Rightarrow\}}$ as follows.

Propositional Variables: a, b, c, where a denotes statement: *One likes to play bridge*, b denotes a statement: *the weather is good*, and c denotes a statement: *one likes not to play bridge.*

$$A_2 = (a \cup (b \Rightarrow (\neg a \cup c))).$$

Part 2 Solution 1

Here is a simple **LK** semantics. We define the logical connectives by writing functions defining connectives in form of the truth tables and skipping other points of the definition 7. We leave it to the reader as an exercise to write down a full definition according to the definition 7.

LK Semantics 1

L	F	$\perp$	T
	F	F	T

$\neg$	F	$\perp$	T
	T	F	F

$\cap$	F	$\perp$	T
F	F	F	F
$\perp$	F	$\perp$	$\perp$
T	F	$\perp$	T

$\cup$	F	$\perp$	T
F	F	$\perp$	T
$\perp$	$\perp$	T	T
T	T	T	T

$\Rightarrow$	F	$\perp$	T
F	T	T	T
$\perp$	T	$\perp$	T
T	F	F	T

We verify whether the condition **s3** is satisfied, i.e. whether $\models_{\mathbf{LK}} \ (LA \cup \neg LA)$ by simple evaluation. Let $v : VAR \longrightarrow \{F, \perp, T\}$ be any truth assignment. For any formula A, $v^*(A) \in \{F, \perp, T\}$ and $LF \cup \neg LF = LF \cup \neg LF = F \cup \neg F \cup T = T$, $\ L \perp \cup \neg L \perp = F \cup \neg F = F \cup T = T$, $\ LT \cup \neg LT = T \cup \neg T = F \cup T = T$.

Part 2 Solution 2

Here is another simple **LK** semantics. Writing, yet again, a full definition is left to the reader as an exercise.

LK Semantics 2

The logical connectives are the following funcions in the set $\{F, \perp, T\}$, where $\{F < \perp < T\}$. We define $\neg F = T$, $\neg \perp = T$, $\neg T = F$ and, as by **s2**, $LT = T$, $L \perp = F$, $LF = F$. We define, for any $x, y \in \{F, \perp, T\}$

$$x \cap y = min\{x,\ y\},\ x \cup y = T,\ x \Rightarrow y = T \text{ if } x \leq y,\ x \Rightarrow y = F \text{ if } x > y.$$

From the above definition we can see the **LK** satisfies the requirement **s3** that especially $\models_{\mathbf{LK}} (LA \cup \neg LA)$ since for any truth assignment v, no matter what values $v^*(LA)$ and $v^*(\neg LA)$ are, the combination of them by $\cup$ will always be T.

Part 3

1. Verify whether the formulas A_1 and A_2 from the **Part 1.** have a model/ counter model under your semantics **LK**. You can use shorthand notation.

Solution 1
A model for $A_1 = (La \cup (b \Rightarrow (\neg La \cup L\neg a)))$ under **LK** semantics 1 is any v, such that $v(a) = T$. By easy evaluation, A_1 does not have no counter model, i.e. $\models_{\mathbf{LK}} A_1$. Also any v, such that $v(a) = T$ is a model for A_1 as we have $v^*(A_2) = T \cup v^*((b \Rightarrow (\neg a \cup c))) = T$ by definition of $\cup$.

Solution 2 The main connective of A_1 and A_2 is $\cup$. By definition of $\cup$ in **LK** semantics 2, $x \cup y = T$ for all $x, y \in \{F, \perp, T\}$, and hence any v is a model for both A_1 and A_2, i.e. they are both tautologies under **LK** semantics 2.

Part 3

2. Verify whether the following set G is **LK** consistent. You can use **shorthand notation**.

$$\mathcal{G} = \{La, \quad (a \cup \neg Lb), \quad (a \Rightarrow b), \ b \ \}.$$

Solution 1]
$\mathcal{G}$ is **LK** consistent under semantics 1 because any v, such that $v(a) = T, \ v(b) = T$ is a **LK** model for $\mathcal{G}$ under semantics 1 by straightforward evaluation.

Solution 2
Consider any v, such that $v(a) = v(b) = T$. We evaluate: $v^*(La) = LT = T, \ v^*((a \cup \neg Lb)) = T \cup F = T, \quad v^*(a \Rightarrow b)) = T \Rightarrow T = T$. This proves $v \models_{\mathbf{LK}} \mathcal{G}$, i.e. $\mathcal{G}$ is consistent.

Part 3

3. Give an example on an infinite, **LK** consistent set of formulas of the language $\mathcal{L}_{\{\neg, \ L, \ \cap, \ \cup, \ \Rightarrow\}}$. Some formulas must contain the connective L.

Solution
The infinite set $\mathcal{G} = \{La : \quad a \in VAR\}$ is consistent under both **LK** semantics, as any v, such that $v(a) = T$ we get $v^*(La) = LT = T$ by **s2**.

The infinite set $\mathcal{G} = \{(La \cup (b \cap L\neg c)) : \quad a, \ b, c \in VAR\}$ is consistent under the semantics 2 by its definition of $\cup$. Any v, such that $v(a) = T$ is its model.

M Equivalence of Formulas

Given an extensional semantics **M** defined for a propositional language $\mathcal{L}_{CON}$ with the set $\mathcal{F}$ of formulas and a set $LV \neq \emptyset$ of logical values. We extend now the classical notion of logical equivalence introduced in section 4 to the extensional semantics **M**.

Definition 43

For any formulas $A, B \in \mathcal{F}$*, we say that* A, B *are* **M** *logically equivalent if and only if they always have the same logical value assigned by the semantics* **M***, i.e. when* $v^*(A) = v^*(B)$ *for all* $v : VAR \ \rightarrow \ LV$*. We write*

$$A \equiv_{\mathbf{M}} B$$

to denote that A, B *are* **M- logically equivalent**.

Remember that $\equiv_{\mathbf{M}}$ is not a logical connective. It is just a **metalanguage symbol** for saying "formulas A, B are logically equivalent under the semantics **M**". We use symbol $\equiv$ for classical logical equivalence only.

Exercise 31

The classical logical equivalence $(A \cup B) \equiv (\neg A \Rightarrow A)$ *holds for all formulas* A*,* B *and is defining* $\cup$ *in terms of negation and implication. Show that it does not hold under* **L** *semantics, i.e. that there are formulas* A*,* B*, such that*

$$(A \cup B) \not\equiv_{\mathbf{L}} (\neg A \Rightarrow B)$$

Solution
Consider a case when A = a and B = b. By definition 43 we have to show $v^*((a \cup b)) \neq v^*((\neg a \Rightarrow b))$ for some $v : VAR \rightarrow \{F, \perp, T\}$. Observe that $v^*((a \cup b)) = v^*((\neg a \Rightarrow b))$ for all $v : VAR \rightarrow \{F, T\}$. So we have to check only truth assignments that involve $\perp$. Let v be any v such that $v(a) = v(b) = \perp$. We evaluate $v^*((a \cup b) = \perp \cup \perp = \perp$ and $v^*((\neg a \Rightarrow b)) = \neg \perp \Rightarrow \perp = F \Rightarrow \perp = T$. This proves that $(a \cup b) \not\equiv_{\mathbf{L}} (\neg a \Rightarrow b)$.. and hence we have proved $(A \cup B) \not\equiv_{\mathbf{L}} (\neg A \Rightarrow B)$.

We proved that the classical equivalence defining disjunction in terms of negation and implication can't be used for the same goal in **L** semantics. It does not mean that we can't define **L** disjunction in terms of **L** implication. In fact, we prove by simple evaluation that the following holds.

Fact 10

The **L** *disjunction is definable in terms of* **L** *implication only, i.e. for any formulas* $A, B \in \mathcal{F}$

$$(A \cup B) \equiv_{\mathbf{L}} ((A \Rightarrow B) \Rightarrow B).$$

The classical equivalence substitution theorem 12 extends to any semantics **M** as follows.

Theorem 28 (M Equivalence)

Let a formula B_1 *be obtained from a formula* A_1 *by a substitution of a formula* B *for one or more occurrences of a sub-formula* A *of* A_1*, what we denote as*

$$B_1 = A_1(A/B).$$

Then the following holds for any formulas $A,\ A_1,\ B,\ B_1 \in \mathcal{F}$*.*

$$\textit{If} \quad A \equiv_{\mathbf{M}} B, \quad \textit{then} \quad A_1 \equiv_{\mathbf{M}} B_1.$$

We leave the proof to the reader as an exercise.

Example 27

Let $A_1 = (a \Rightarrow (\neg a \cup b))$ *and consider a sub formula* $A = (\neg a \cup b)$ *of* A_1. *By Fact 10,* $(\neg a \cup b) \equiv_{\mathbf{L}} ((\neg a \Rightarrow b) \Rightarrow b)$. *Take* $B = ((\neg a \Rightarrow b) \Rightarrow b)$ *and let* $B_1 = A_1(A/B) = A_1((\neg a \cup b)/((\neg \Rightarrow b) \Rightarrow b)) = (a \Rightarrow ((\neg a \Rightarrow b) \Rightarrow b))$. *By the M Equivalence Theorem 28*

$$(a \Rightarrow (\neg a \cup b)) \equiv_{\mathbf{L}} (a \Rightarrow ((\neg \Rightarrow b) \Rightarrow b)).$$

M Equivalence of Languages

We extend now, in a natural way, the classical notion equivalence of languages introduced and examined in section 4.

Definition 44

Given two languages: $\mathcal{L}_1 = \mathcal{L}_{CON_1}$ *and* $\mathcal{L}_2 = \mathcal{L}_{CON_2}$, *for* $CON_1 \neq CON_2$. *We say that they are* **M** *logically equivalent and denote it by*

$$\mathcal{L}_1 \equiv_{\mathbf{M}} \mathcal{L}_2$$

if and only if the following conditions **C1, C2** *hold.*

C1: *for any formula* A *of* $\mathcal{L}_1$, *there is a formula* B *of* $\mathcal{L}_2$, *such that* $A \equiv_{\mathbf{M}} B$,

C2: *for any formula* C *of* $\mathcal{L}_2$, *there is a formula* D *of* $\mathcal{L}_1$, *such that* $C \equiv_{\mathbf{M}} D$.

Exercise 32

Prove that

$$\mathcal{L}_{\{\neg,\Rightarrow\}} \equiv_{\mathbf{L}} \mathcal{L}_{\{\neg,\Rightarrow,\cup\}}$$

Solution
Condition **C1** holds because any formula of $\mathcal{L}_{\{\neg,\Rightarrow\}}$ is a formula of $\mathcal{L}_{\{\neg,\Rightarrow,\cup\}}$. Condition **C2** holds because the Fact 10 equivalence $(A \cup B) \equiv_{\mathbf{L}} ((A \Rightarrow B) \Rightarrow B)$ and the Theorem 28.

7 Homework Problems

Formal Propositional Languages

For the problems below do the following.

(i) Determine which of the formulas is, and which is not a well formed formula. Determine a formal language of $\mathcal{L}$ to which the formula or set of formulas belong.

(ii) If a formula is correct, write what its *main connective* is. If it is not correct, write the corrected formula and then write its *main connective*. If there is more then one way to correct the formula, write all possible corrected formulas.

(iii) If a formula is correct, write what it says. If it is not correct, write the corrected formula and then write what it says.

(iv) For each of correct formula determine its degree and write down its all sub-formulas of the degree 0 and 1.

Problems

1. $((a \uparrow b) \uparrow (a \uparrow b) \uparrow a)$
2. $(a \Rightarrow \neg b) \Rightarrow \neg a$
3. $\Diamond(a \Rightarrow \neg b) \cup a, \quad \Diamond(a \Rightarrow (\neg b \cup a), \quad \Diamond a \Rightarrow \neg b \cup a$
4. $(\Box\neg\Diamond a \Rightarrow \neg a), \quad \Box(\neg\Diamond a \Rightarrow \neg a), \quad \Box\neg\Diamond(a \Rightarrow \neg a)$
5. $((a \cup \neg K\neg a)), \quad KK(b \Rightarrow \neg a), \quad \neg K(a \cup \neg a)$
6. $(B(a \cap b) \Rightarrow Ka), \quad B((a \cap b) \Rightarrow Ka)$
7. $G(a \Rightarrow b) \Rightarrow Ga \Rightarrow Gb), \quad a \Rightarrow HFa, \quad FFa \Rightarrow Fa$
8. $(a \Rightarrow ((\neg b \Rightarrow (\neg a \cup c)) \Rightarrow \neg a))$
9. $\Diamond((a \cap \neg a) \Rightarrow (a \cap b))$
10. $\Box\neg\Diamond(a \Rightarrow \neg a)$
11. $\Diamond(\Diamond a \Rightarrow (\neg b \cup \Diamond a))$
12. $(\neg(a \cap b) \cup a)$
13. Write the natural language statement:

    From the fact that it is not necessary that an elephant is not a bird we deduce that:
    it is not possible that an elephant is a bird or, if it is possible that an elephant is a bird, then it is not necessary that a bird flies.

    in the following two ways.

    1. As a formula $A_1 \in \mathcal{F}_1$ of a language $\mathcal{L}_{\{\neg,\mathbf{C},\mathbf{I},\cap,\cup,\Rightarrow\}}$.

    2. As a formula $A_2 \in \mathcal{F}_2$ of a language $\mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow\}}$.

14. Write the natural language statement

    If it is not believed that quiz is easy or quiz is not easy, then from the fact that $2 + 2 = 5$ *we deduce that it is believed that quiz is easy.*

    in the following two ways.

    1. As a formula A_1 of a language $\mathcal{L}_1 = \mathcal{L}_{\{\neg, \mathbf{B}, \cap, \cup, \Rightarrow\}}$, where $\mathbf{B}$ is a believe connective. Statement $\mathbf{B}$A says: *It is believed that* A.

    2. As a formula A_2 of a language $\mathcal{L}_2 = \mathcal{L}_{\{\neg, \cap, \cup, \Rightarrow\}}$.

Formal Classical Semantics

1. Find and prove definability formula defining implication in terms of conjunction and negation.
2. Find and prove definability formula defining conjunction in terms of disjunction and negation.
3. Find and prove definability formula defining conjunction in terms of implication and negation.
4. Prove that $\cup$ can be defined in terms of $\Rightarrow$ alone.
5. Find and prove definability formula defining $\Rightarrow$ in terms of $\uparrow$.
6. Find definability formula defining $\Rightarrow$ in terms of $\downarrow$.
7. Define $\cap$ in terms of $\Rightarrow$ and $\neg$.
8. Find definability formula defining $\cap$ in terms of $\downarrow$ alone.
9. Given a formula A: $(((a \cap b) \cup \neg c) \Rightarrow b)$. Evaluate (do not use shorthand notation) $v^*(A)$ for truth assignments $v : VAR \longrightarrow \{T,\ F\}$ such that

    (i)$v(a) = T,\ v(b) = F,\ v(c) = F,\ v(x) = T$ for all $x \in VAR - \{a, b, c\}$,

    (ii)$v(a) = F,\ v(b) = T,\ v(c) = T,\ v(x) = F$ for all $x \in VAR - \{a, b, c\}$.
10. Given a formula A: $(((a \Rightarrow \neg b) \cup b) \Rightarrow a)$. Evaluate (use shorthand notation) $v^*(A)$ for all truth assignments restricted to A.
11. Given a formula A: $(((a \downarrow \neg b) \cup b) \uparrow a)$. Evaluate (do not use shorthand notation) $v^*(A)$ for truth assignments $v : VAR \longrightarrow \{T,\ F\}$ such that

    (i) v(a)=T, v(b)=F, v(c) =F for all $c \in VAR - \{a, b\}$,

    (ii) v(a)=F, v(b)=T, v(c) =T for all $c \in VAR - \{a, b\}$.

    (iii) List all restricted models and counter-models for A.

Write the following natural language statement *From the fact that it is possible that* $2+2 \neq 4$ *we deduce that it is not possible that* $2+2 \neq 4$ *or, if it is possible that* $2+2 \neq 4$*, then it is not necessary that you go to school.* as a formula . $A \in \mathcal{F}$ of a language $\mathcal{L} = \mathcal{L}_{\{\neg, \cap, \cup, \Rightarrow\}}$.

(i) Find a restricted model v for the formula A.

(ii) Find 3 models w of A such that $v^*(A) = w^*(A)$ the for v from (i). How many of such models exist?

(iii) Find all models, counter-models (restricted) for A. Use shorthand notation.

(iv) Is $A \in \mathbf{C}$?, is $A_2 \in \mathbf{T}$? Justify your answers.

12. Given $v : VAR \longrightarrow \{T, F\}$ such that $v^*((\neg a \cup b) \Rightarrow (a \Rightarrow \neg c)) = F$. Evaluate: $v^*(((b \Rightarrow a) \Rightarrow (a \Rightarrow \neg c)) \cup (a \Rightarrow b))$.

13. Show that all of the truth assignments v_1, v_2, v_3 defined below are **models** for the formula $A : ((a \cap \neg b) \cup \neg c)$.

    $v_1 : VAR \longrightarrow \{T, F\}$, is such that $v_1(a) = T,\ v_1(b) = F,\ v_1(c) = T$, and $v_1(x) = F$, for all $x \in VAR - \{a, b, c\}$;
    $v_2 : VAR \longrightarrow \{T, F\}$ is such that $v_2(a) = T,\ v_2(b) = F,\ v_2(c) = T$, $v_2(d) = T$, and $v_2(x) = F$ for all $x \in VAR - \{a, b, c, d\}$;
    $v_3 : VAR \longrightarrow \{T, F\}$ is such that $v_3(a) = T,\ v_3(b) = F,\ v_3(c) = T$, $v_3(d) = T,\ v_3(e) = T$, and $v_3(x) = F$, for all $x \in VAR - \{a, b, c, d, e\}$.

14. Prove that for any formula $A \in \mathcal{F}$, if A has a model (counter- model), then it has uncountably many models (counter-models). More precisely, as many as there are real numbers. *Hint* Use the Counting Functions Theorem 4.

15. Use Generalization Method to determine whether

    $\models (\neg((a \cup b) \Rightarrow ((c \Rightarrow d) \cup e)) \Rightarrow ((a \cup b) \cap (\neg(c \Rightarrow d) \cap \neg e)))$.

16. Prove $\models (\neg((a \cup b) \Rightarrow (c \Rightarrow d)) \Rightarrow (\neg((a \cup b) \Rightarrow (c \Rightarrow d)) \Rightarrow (\neg e \cap a)))$.

17. Use Proof by Contradiction Method to determine whether

    $\models (((A \Rightarrow (B \Rightarrow C)) \cap (A \Rightarrow B)) \Rightarrow (A \Rightarrow C))$.

18. Use Truth Table and Substitution Methods to prove $\models (\neg\neg A \Leftrightarrow A)$.

19. Use Truth Table and Substitution Methods to prove to prove the *Reductio ad Absurdum* tautology $((\neg A \Rightarrow (B \cap \neg B)) \Rightarrow A)$.

20. Use Proof by Contradiction Method to prove the *Exportation and Importation* tautology $(((A \cap B) \Rightarrow C) \Leftrightarrow (A \Rightarrow (B \Rightarrow C)))$.

21. For the formulas listed below determine whether they are tautologies or not. If a formula is not a tautology list its counter-model (restricted). Use shorthand notation.

(i) $A_1 = (\neg(a \Rightarrow (b \cap \neg c)) \Rightarrow (a \cap \neg(b \cap \neg c)))$

(ii) $A_2 = ((a \cap \neg b) \Rightarrow ((c \cap \neg d) \Rightarrow (a \cap \neg b)))$

(iii) $A_3 = (\neg(A \cap \neg B) \cup (A \cap \neg B))$

22. Find all models and a counter-model restricted to $\mathcal{G}$ (if exist) for the following sets $\mathcal{G}$ of formulas. Use shorthand notation.

    (i) $\mathcal{S}_1 = \{a, (a \cap \neg b), (\neg a \Rightarrow (a \cup b))\}$

    (ii) $\mathcal{S}_2 = \{(a \Rightarrow b), (c \cap \neg a), b\}$

    (iii) $\mathcal{S}_3 = \{a, (a \cap \neg b), \neg a, c\}$

23. Give an example of an infinite set $\mathcal{G} \subseteq \mathcal{F}$, such that $\mathcal{G} \neq \mathbf{T}$ and $\mathcal{G}$ has a model, i.e. is consistent.

24. Give an example of an infinite consistent set $\mathcal{G} \subseteq \mathcal{F}$, such that $\mathcal{G} \cap \mathbf{T} = \emptyset$.

25. Give an example of an infinite set $\mathcal{G} \subseteq \mathcal{F}$, such that $\mathcal{G} \neq \mathbf{C}$ and $\mathcal{G}$ does not have a model, i.e.is inconsistent.

26. Give an example of an infinite set $\mathcal{G} \subseteq \mathcal{F}$, such that $\mathcal{G} \cap \mathbf{C} = \emptyset$.

27. Find an infinite number of formulas that are independent from a set $\mathcal{G} = \{(a \Rightarrow (a \cup b)),\ (a \cup b),\ \neg b,\ (c \Rightarrow b)\}$. Use shorthand notation.

28. Given an infinite set $\mathcal{G} = \{(a \cup \neg a) :\ a \in VAR\}$. Find 3 formulas $A \in \mathcal{F}$ that are independent from $\mathcal{G}$.

29. Give an example of an infinite set $\mathcal{G}$ and an infinite set of formulas independent from it.

Equivalence of Languages

1. Prove that $\mathcal{L}_{\{\cap,\neg\}} \equiv \mathcal{L}_{\{\cup,\neg\}}$.

2. Transform a formula $A = \neg(\neg(\neg a \cap \neg b) \cap a)$ of $\mathcal{L}_{\{\cap,\neg\}}$ into a logically equivalent formula B of $\mathcal{L}_{\{\cup,\neg\}}$.

3. Transform a formula $A = (((\neg a \cup \neg b) \cup a) \cup (a \cup \neg c))$ of $\mathcal{L}_{\{\cup,\neg\}}$ into a formula B of $\mathcal{L}_{\{\cap,\neg\}}$, such that $A \equiv B$.

4. Prove, using proper logical equivalences (list them at each step) that

    (i) $\neg(A \Leftrightarrow B) \equiv ((A \cap \neg B) \cup (\neg A \cap B))$.

    (ii) $((B \cap \neg C) \Rightarrow (\neg A \cup B)) \equiv ((B \Rightarrow C) \cup (A \Rightarrow B))$.

5. Prove that $\mathcal{L}_{\{\neg,\cap\}} \equiv \mathcal{L}_{\{\neg,\Rightarrow\}}$.

6. Prove by using proper logical equivalences that

    (i) $\neg(\neg A \cup \neg(B \Rightarrow \neg C)) \equiv (A \cap \neg(B \cap C))$,

    (ii) $(\neg A \cap (\neg A \cup B)) \equiv (\neg A \cup (\neg A \cap B))$.

7. Prove that $\mathcal{L}_{\{\cap,\cup.\neg\}} \equiv \mathcal{L}_{\{\Rightarrow,\neg\}}$.
8. Prove that $\mathcal{L}_{\{\cap,\cup,\Rightarrow,\neg\}} \equiv \mathcal{L}_{\{\cup,\neg\}}$.
9. (i) Transform a formula $A = (((a \cup \neg b) \Rightarrow a) \cap (\neg a \Rightarrow \neg b))$ of $\mathcal{L}_{\{\cap,\cup,\Rightarrow,\neg\}}$ into a logically equivalent formula B of $\mathcal{L}_{\{\cup,\neg\}}$.

   (ii) Find all B of $\mathcal{L}_{\{\cup,\neg\}}$, such that $B \equiv A$, for A from (i).
10. (i) Transform a formula $A = (((\neg a \cup \neg b) \cup a) \cup (a \cup \neg c))$ of $\mathcal{L}_{\{\cup,\neg\}}$ into a formula B of $\mathcal{L}_{\{\cap,\cup,\Rightarrow,\neg\}}$, such that $A \equiv B$.

    (ii) Find all B of $\mathcal{L}_{\{\cap,\cup,\Rightarrow,\neg\}}$, such that $B \equiv A$, for A from (i)
11. Prove taht $\mathcal{L}_{\{\cap,\cup,\Rightarrow,\neg\}} \equiv \mathcal{L}_{\{\uparrow\}}$.
12. Prove that $\mathcal{L}_{\{\cap,\cup,\Rightarrow,\neg\}} \equiv \mathcal{L}_{\{\ \downarrow\}}$.
13. Prove that $\mathcal{L}_{\{\uparrow\}} = \mathcal{L}_{\{\ \downarrow\}}$.

Many Valued Semantics

1. In all 3-valued semantics presented here we chose the language without the equivalence connective "$\Leftrightarrow$". Extend t **L**, $\mathbf{L}_4$ semantics to a language containing the equivalence connective. Prove that your semantics is well defined as by definition 14.
2. Extend **H**, **K**, semantics to a language containing the equivalence connective. Are your semantics well defined as by definition 14?
3. Extend **B**, semantics to a language containing the equivalence connective. Are your semantics well defined as by definition 14?
4. Let $v : VAR \longrightarrow \{F, \perp, T\}$ be any v, such that $v^*((a \cup b) \Rightarrow (a \Rightarrow c)) = \perp$ under H semantics. Evaluate $v^*(((b \Rightarrow a) \Rightarrow (a \Rightarrow \neg c)) \cup (a \Rightarrow b))$.
5. Verify which of the classical tautologies (10) are, and which are not **L** tautologies.
6. Verify which of the classical tautologies (11) are, and which are not **L** tautologies.
7. Give an example of 3 formulas
8. For each of 3-valued logic semantics presented in this chapter, find 5 classical tautologies that are tautologies of that logic.
9. Examine the notion of *definability of connectives* as defined in section 3, definition 16 for **L** semantics. semantics.
10. Examine the notion of *definability of connectives* as defined in section 3, definition 16 for **H** semantics. semantics.

11. Given a set $\mathcal{G} = \{((a \cap b) \Rightarrow b),\ (a \cup b), a\}$. Verify whether $\mathcal{G}$ is consistent under **H** semantics.

12. Given a set $\mathcal{G} = \{((a \cap b) \Rightarrow b),\ (a \cup b), a\}$. Verify whether $\mathcal{G}$ is consistent under **L** semantics.

13. Given a language $\mathcal{L}_{\{\neg,\Rightarrow,\cup,\cap\}}$. We define: A formula $A \in \mathcal{F}$ is called **M independent** from a set $\mathcal{G} \subseteq \mathcal{F}$ if and only if the sets $\mathcal{G} \cup \{A\}$ and $\mathcal{G} \cup \{\neg A\}$ are both **M** consistent. I.e. when there are truth assignments $v_1,\ v_2$ such that $v_1 \models_{\mathbf{M}} \mathcal{G} \cup \{A\}$ and $v_2 \models_{\mathbf{M}} \mathcal{G} \cup \{\neg A\}$.

    Given a set $\mathcal{G} = \{((a \cap b) \Rightarrow b),\ (a \cup b), a\}$.

    (i) Find a formula A that is **L independent** from a set $\mathcal{G}$.

    (ii) Find a formula A that is **H independent** from a set $\mathcal{G}$.

    (iii) Find an infinite number of that are **L independent** from a set $\mathcal{G}$.

    (iv) Find an infinite number of that are **H independent** from a set $\mathcal{G}$.

14. By exercise 31 the classical logical equivalence $(A \cup B) \equiv (\neg A \Rightarrow A)$ does not hold under **L** semantics, i.e. that there are formulas A, B, such that $(A \cup B) \not\equiv_{\mathbf{L}} (\neg A \Rightarrow B)$. Show 3 formulas A,B such that it does hold for **L** semantics, i.e. such that are formulas A, B, such that $(A \cup B) \equiv_{\mathbf{L}} (\neg A \Rightarrow B)$.