LOGICS FOR COMPUTER SCIENCE: Classical and Non-Classical Springer 2019

Anita Wasilewska

Chapter 3
Propositional Semantics: Classical and Many Valued

CHAPTER 3 SLIDES

Chapter 3
Propositional Semantics: Classical and Many Valued

Chapter 3
Propositional Semantics: Classical and Many Valued

Slides Set 5

PART 7 Classical Tautologies and Logical Equivalences

PART 8 Definability of Connectives and Equivalence of Languages

Slides Set 6

PART 9 Many Valued Semantics: Łukasiewicz, Heyting, Kleene, and Bohvar

Slides Set 7

PART 10 **M** Tautologies, **M** Consistency, and **M** Equivalence of Languages

Chapter 3

Propositional Semantics: Classical and Many Valued

Slides Set 1

PART 1 Formal Propositional Languages: Introduction

Propositional Languages Introduction

We define now a general notion of a **propositional** language

We show how to obtain, as specific cases, various languages for propositional classical logic and some non-classical logics

We **assume** the following

All **propositional** languages contain an infinitely countable set of variables VAR, which elements are denoted by

$$a, b, c, ...$$

with indices, if necessary

All **propositional** languages share the general way their sets of formulas are formed

Propositional Languages

What **distinguishes** one propositional language from the other is the choice of its set of propositional connectives

We **adopt** a notation

$$\mathcal{L}_{CON}$$

where CON stands for the set of propositional connectives

We **use** a notation

$$\mathcal{L}$$

when the set of connectives is fixed

Propositional Languages

For **example**, the language

$$\mathcal{L}_{\{\neg\}}$$

denotes a propositional language with only one connective $\neg$

The language

$$\mathcal{L}_{\{\neg,\Rightarrow\}}$$

denotes that a language with two connectives $\neg$ and $\Rightarrow$ adopted as propositional connectives

Remember: formal languages deal with symbols only and are also called symbolic languages

General Principles

Symbols for connectives do have intuitive meaning

Semantics provides a **formal** meaning of the connectives and is defined separately

One language can have many semantics

Different logics can **share** the same language

For **example**, the language

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

is **used** as a propositional language of classical and intuitionistic logics, some many- valued logics, and we **extend** it to the language of many modal logics

General Principles

Several languages can share the same semantics

The **classical** propositional logic is the best example of such situation

Due to the functional dependency of **classical** logic connectives the languages:

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

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

are all equivalent under the classical **semantics**

We will define **formally** languages equivalency in the next chapter

General Principles

Propositional connectives have well established **names** and the way we read them, even if their **semantics** may differ

We use **names** negation, conjunction, disjunction and implication for $\neg$, $\cap$, $\cup$, $\Rightarrow$, 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

Some Non-Classical Propositional Connectives

Other most common propositional connectives are **modal** connectives of possibility and necessity

Modal connectives are **not** extensional

Standard modal symbols are:
$\Box$ for necessity and $\Diamond$ for possibility

We will also use symbols C and I for **modal** connectives of possibility and necessity, respectively.

The formula CA, or $\Diamond A$ reads: it is possible that A or A is possible

The formula I A, or $\Box A$ reads: it is necessary that A or A is necessary

Modal Propositional Connectives

Symbols C and I are used for their **topological** meaning in the algebraic semantics of standard **modal logics** S4 and S5

In **topology** C is a symbol for a set closure operation and CA means a closure of a set A

I is a symbol for a set interior operation and IA denotes an interior of the set A

Some More Non-Extensional Connectives

Modal logics extend the **classical** logic

Modal logics languages are for example

$$\mathcal{L}_{\{C, I, \neg, \cap, \cup, \Rightarrow\}} \quad \text{or} \quad \mathcal{L}_{\{\Box, \Diamond, \neg, \cap, \cup, \Rightarrow\}}$$

Knowledge logics also extend the **classical** logic by adding a new one argument knowledge connective

The knowledge connective is often denoted by K

A formula KA reads: it is known that A or A is known

A language of a **knowledge** logic is for example

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

Some More Non-Extensional Connectives

Autoepistemic logics extend **classical** logic by adding an one argument believe connective, often denoted by B

A formula BA reads: it is believed that A

A language of an **autoepistemic** logic is for example

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

Some More Non-Extensional Connectives

Temporal logics also extend **classical** logic by adding one argument temporal connectives

Some of temporal connectives are: F, P, G, H.

Their intuitive meanings are:

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,

HA reads A has always been true in the past

Propositional Connectives

It is **possible** to create and there are connectives with **more** then one or two arguments

We **consider** here only one or two argument connectives

Chapter 3
Propositional Semantics: Classical and Many Valued

PART 2 Propositional Languages: Definitions

Propositional Language

Definition

A **propositional language** is a pair

$$\mathcal{L} = (\mathcal{A}, \mathcal{F})$$

where $\mathcal{A}, \mathcal{F}$ are called an alphabet and a set of formulas, respectively

Definition

Alphabet is a set

$$\mathcal{A} = VAR \cup CON \cup PAR$$

VAR, CON, PAR are all disjoint sets of propositional **variables, connectives** and **parenthesis**, respectively

The sets VAR, CON are non-empty

Alphabet Components

Alphabet Components

VAR is a countably infinite set of propositional variables

We denote elements of VAR by

$$a, b, c, d, ...$$

with indices if necessary

$CON \neq \emptyset$ is a **finite** set of propositional connectives

We assume that the set CON of connectives is **non-empty**, i.e. that a propositional language always has at least one connective

Alphabet Components

Notation

We **denote** the language $\mathcal{L}$ with the set of connectives CON by

$$\mathcal{L}_{CON}$$

Observe that propositional languages **differ** only on a **choice** of the connectives, hence our notation.

Alphabet Components

PAR is a set of auxiliary symbols

This set may be empty; for example in case of parenthesis free Polish notation.

Assumptions

We **assume** that PAR contains only 2 parenthesis and

$$PAR = \{(,)\}$$

We also **assume** that the set CON of connectives contains **only** unary and binary connectives, i.e.

$$CON = C_1 \cup C_2$$

where C_1 is the set of all unary connectives, and C_2 is the set of all binary connectives

Definition

The set $\mathcal{F}$ of all **formulas** of a propositional language $\mathcal{L}_{CON}$ is build **recursively** from the elements of the alphabet $\mathcal{A}$ as follows.

$\mathcal{F} \subseteq \mathcal{A}^*$ and $\mathcal{F}$ is the **smallest** set for which the following conditions are satisfied

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

(2) If $A \in \mathcal{F}$, $\nabla \in C_1$, then $\nabla A \in \mathcal{F}$

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

By (1) propositional variables are formulas and they are called **atomic formulas**

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

Set of Formulas

Observe that the the alphabet $\mathcal{A}$ is countably infinite

Hence the set $\mathcal{A}^*$ of all finite sequences of elements of $\mathcal{A}$ is also countably infinite

By definition $\mathcal{F} \subseteq \mathcal{A}^*$ and hence we get that the set of all formulas $\mathcal{F}$ is also countably infinite

We state as separate fact

Fact

For any propositional language $\mathcal{L} = (\mathcal{A}, \mathcal{F})$, its sets of formulas $\mathcal{F}$ is always a **countably infinite** set

We hence consider here only infinitely countable languages

Main Connectives and Direct Sub-Formulas

∇ is called a main connective of the formula $\nabla A \in \mathcal{F}$

A is called its direct sub-formula of ∇A

$\circ$ is called a main connective of the formula $(A \circ B) \in \mathcal{F}$

A, B are called direct sub-formulas of $(A \circ B)$

Examples

E1 Main connective of $(a \Rightarrow \neg N b)$ is $\Rightarrow$

a, $\neg N b$ are direct sub-formulas

E2 Main connective of $N(a \Rightarrow \neg b)$ is N

$(a \Rightarrow \neg b)$ is the direct sub-formula

E3 Main connective of $\neg(a \Rightarrow \neg b)$ is $\neg$

$(a \Rightarrow \neg b)$ is the direct sub-formula

Sub-Formulas

We define a notion of a sub-formula in two steps:

Step 1

For any formulas A and B, the formula 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

Step 2

A sub-formula of a given formula A is any proper sub-formula of A, or A itself

Sub-Formulas

Example

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

has two direct sub-formulas: $\neg a, \quad \neg(a \Rightarrow b)$,

the direct sub-formulas of which are $a, \quad (a \Rightarrow b)$

The next direct sub-formulas are $a, \quad b$

End of the process

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

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

The set of all its sub-formulas is

$$S \cup \{(\neg a \cup \neg(a \Rightarrow b))\}$$

Formula Degree

Definition

A degree of a formula as a number of occurrences of logical connectives in the formula.

Example

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

Formula Degree

Observation

The **degree** of any proper sub-formula of A must be one less than the degree of A

This is the central fact upon which mathematical induction arguments are based

Proofs of properties of formulas are usually carried by mathematical induction on their **degrees**

Exercise

Exercise 1

Consider a language $\mathcal{L} = \mathcal{L}_{\{\neg,\ \Diamond,\ \Box,\ \cup,\ \cap,\ \Rightarrow\}}$ and a set

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

1. **Determine** which of the elements of S are, and which are not well formed **formulas** (wff) of $\mathcal{L}$

2. If a formula A is a well formed **formula**, i.e. $A \in \mathcal{F}$, determine its its **main** connective

3. If $A \notin \mathcal{F}$ write the corrected formula and then determine its **main** connective

Solution

The expression $\Diamond\neg a \Rightarrow (a \cup b)$ **is not** a well formed formula

The corrected formula is

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

The **main** connective is $\Rightarrow$

The formula says: "If negation of a is possible, then we have a or b "

Another corrected formula in is

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

The main connective is $\Diamond$

The formula says: " It is possible that not a implies a or b "

Exercise 1 Solution

The expression $(\Diamond(\neg a \Rightarrow (a \cup b)))$ **is not** a well formed formula

The correct formula is $\Diamond(\neg a \Rightarrow (a \cup b))$

The **main** connective is $\Diamond$

The formula says: " It is possible that not a implies a or b"

$\Diamond\neg(a \Rightarrow (a \cup b))$ is a well formed **formula**

The **main** connective is $\Diamond$

The formula says: " It is possible that it is not true that a implies a or b "

Exercise

Exercise 2

Given a formula:

$$\Diamond((a \cup \neg a) \cap b)$$

1. Determine its degree
2. Write down all its sub-formulas

Solution

The degree is 4

All its sub-formulas are:

$$\Diamond((a \cup \neg a) \cap b), \quad ((a \cup \neg a) \cap b),$$

$$(a \cup \neg a), \quad \neg a, \quad b, \quad a$$

Definition

Given a set S of formulas of a language $\mathcal{L}_{CON}$

Let $CS \subseteq CON$ be the set of **all connectives** that appear in formulas of S

A language $\mathcal{L}_{CS}$
is called the **language defined** by the set of formulas S

Example

Let S be a set
$S = \{((a \Rightarrow \neg b) \Rightarrow \neg a), \ \Box(\neg\Diamond a \Rightarrow \neg a)\}$
All connectives appearing in the formulas in S are:

$$\Rightarrow, \quad \neg, \quad \Box, \quad \Diamond$$

The **language defined** by the set S is

$$\mathcal{L}_{\{\neg,\ \Rightarrow,\ \Box,\ \Diamond\}}$$

Exercise

Exercise 3

Write the following natural language statement:

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

Exercise 3 Solution

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

Propositional Variables: a,b

a denotes statement: *Anne is a boy*,

b denotes a statement: *Anne is pretty*

Propositional Modal Connectives: $\Box, \Diamond$

$\Diamond$ denotes statement: *it is possible that*

$\Box$ denotes statement: *it is necessary that*

Translation 1: the formula A_1 is

$$(\Diamond\neg a \;\Rightarrow\; (\neg\Diamond\neg a \cup (\Diamond\neg a \Rightarrow \neg\Box b)))$$

Exercise 3 Solution

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

Propositional Variables: a,b

a denotes statement: *it is possible that Anne is not a boy*

b denotes a statement: *it is necessary that Anne is pretty*

Translation 2: the formula A_2 is

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

Exercise

Exercise 4

Write the following natural language 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 $n + m < 0$, 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

$A_1 \in \mathcal{F}_1$ of a language $\mathcal{L}_{\{\neg, \cap, \cup, \Rightarrow\}}$

2. As a formula

$A_2 \in \mathcal{F}_2$ of a language $\mathcal{L}_{\{\neg, \Box, \Diamond, \cap, \cup, \Rightarrow\}}$

Exercise 4 Solution

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

Propositional Variables: a, b

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* $n + m < 0$

b denotes a statement: *it is not possible that there is a natural number m, such that* $m > 0$

Translation: the formula A_1 is

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

Exercise 4 Solution

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

Propositional Variables: a, b

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 $n + m < 0$*

b denotes a statement: *there is a natural number m, such that $m > 0$*

Translation: the formula A_2 is

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

Exercise

Exercise 5

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

The following statement holds for all natural numbers $n \in N$:

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

$A_1 \in \mathcal{F}_1$ of a language $\mathcal{L}_{\{\neg, \cap, \cup, \Rightarrow\}}$

2. As a formula

$A_2 \in \mathcal{F}_2$ of a language $\mathcal{L}_{\{\neg, \Box, \Diamond, \cap, \cup, \Rightarrow\}}$

Exercise 5 Solution

Solution

Observe that the statement **S** is build as follows

$$\forall_{n \in N} A(n),$$

where A(n) represents the statement " 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 $m > 0$ "

From a **propositional** point of view the statement $\forall_{n \in N} A(n)$ can only be represented by a propositional variable

$$a$$

in a case of **both** propositional languages $\mathcal{L}_{\{\neg,\ \cap,\ \cup,\ \Rightarrow\}}$ and $\mathcal{L}_{\{\neg,\ \Box,\ \Diamond,\ \cap,\ \cup,\ \Rightarrow\}}$

Exercise

Exercise 6

Write the following natural language statement:

From the fact that each natural number is greater than zero we deduce that it is not possible that Anne is a boy or, if it is possible that Anne is not a boy, then it is necessary that it is not true that each natural number is greater than zero

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 is similar to the **Exercise 4**

Chapter 3
Propositional Semantics: Classical and Many Valued

CHAPTER 3 SLIDES

Slides Set 2

Chapter 3
Propositional Semantics: Classical and Many Valued

Slides Set 2

PART 3 Extensional Semantics **M**

Extensional Semantics **M** - Introduction

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 $\mathcal{L}_{CON}$ can have different semantics but we always **define** them in order to single out special formulas of the language, called **tautologies**

Tautologies are formulas of the language that are always true under a given **semantics**

Extensional Semantics **M** Introduction

We introduced in Chapter 2 an intuitive notion of a classical **semantics**, discussed its motivation and underlying assumptions

The classical **semantics** assumption is that it considers only two logical values. The other one is 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 that they are represented by some propositional connectives which are **not extensional**

Extensional Semantics **M** Introduction

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

Extensional Connectives Definition

We adopt a following **formal** definition of extensional connectives for a propositional language $\mathcal{L}_{CON}$

Definition

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 $\triangledown \in C_1$ or $\circ \in C_2$ is called **extensional** if it is defined by a respective function

$$\triangledown : LV \longrightarrow LV \quad \text{or} \quad \circ : LV \times LV \longrightarrow LV$$

A semantics **M** for a language $\mathcal{L}_{CON}$ is called **extensional** provided all connectives in CON are extensional and its notion of **tautology** is defined terms of the connectives and their logical values

A semantics with a set of m logical values is called a m-valued **extensional**

The **classical** semantics is a special case of a 2-valued **extensional** semantics

Classical semantics **defines** classical logic with its set of classical propositional **tautologies**

Many of logics are defined by various **extensional semantics** with sets of logical values LV with more then 2 elements

Extensional Semantics **M** Introduction

The languages of many important **logics** like modal, multi-modal, knowledge, believe, temporal, contain **connectives** that are **not extensional** because they are defined by **non-extensional** semantics

The intuitionistic logic is based on the **same** language as the classical one and its Kripke Models semantics is **not extensional**

Defining a **semantics** for a given language means **more** then defining connectives

The ultimate **goal** of any semantics is to define the notion of its own **tautology**

In order to **define** which formulas of a given

$$\mathcal{L}_{CON}$$

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 ”absolute” truth

We also can **distinguish**, and often we do, another special value F representing ”absolute” falsehood

We will use these symbols T, F for ”absolute” truth and falsehood

We may also use other symbols like 1, 0 or others

Extensional Semantics **M** Introduction

The ”absolute” truth 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 (extensional) connectives, but also the method of **defining** the notion of a **tautology**

We hence **define** a general notion of an extensional semantics as sequence of **steps** leading to the definition of a **tautology**

Extensional Semantics **M** Introduction

Here are the **steps** leading to the definition of a **tautology**

Step 1 We **define** all extensional connectives of **M**

Step 2 We **define** main component of the definition of a **tautology**, namely a **function** v that assigns to any formula $A \in \mathcal{F}$ its logical **value** from LV

The function v is often called a **truth assignment** and we will use this name

Extensional Semantics **M** Introduction

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. we define a notion saying that v is a **model** for A under semantics **M**

Step 4 We **define** a notion of tautology as follows

A is a **tautology** under semantics **M** if and only if **all** truth assignments v **satisfy** A

i.e. that **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**

The notion of a **model** is usually defined in terms of the notion of **satisfaction**

In **classical** propositional logic these notions are the same and the **use** of expressions

" v **satisfies** A" and "v is a **model** for A"

is **interchangeable**

This also is true for of any propositional **extensional semantics** and in particular it holds for m-valued semantics discussed later in this chapter

Extensional Semantics **M** Introduction

The notions of **satisfaction** and **model** are not interchangeable for **predicate** languages semantics

We already discussed intuitively the notion of **model** and **satisfaction** for predicate language in chapter 2

We will define them in **full formality** in chapter 8

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

Extensional Semantics **M** Formal Definition

Definition

Any formal definition of an **extensional semantics** **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, its distinguished value T, and define all connectives of $\mathcal{L}_{CON}$ to be **extensional**

Step 2 We define notion of a truth assignment and its extension

Step 3 We define notions of satisfaction, model, counter model

Step 4 We define notion of a **tautology** under the semantics **M**

Extensional Semantics M Formal Definition

What differs one semantics from the other is the **choice** of the set LV of logical values and **definition** of the connectives of $\mathcal{L}_{CON}$, that are defined in the first step below

Step 1 We adopt a following formal definition of **extensional** connectives of $\mathcal{L}_{CON}$

Definition

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 $\triangledown \in C_1$ or $\circ \in C_2$ is called **extensional** if it is defined by a respective function

$$\triangledown : \ LV \longrightarrow LV \quad \text{or} \quad \circ : \ LV \times LV \longrightarrow LV$$

M Truth Assignment Formal Definition

Step 2 We define a function called **truth assignment** and its **extension** in terms of the propositional connectives as defined in the **Step 1**

Definition

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**

We call it for short a **M truth assignment**

We use the term **M** truth assignment and **M** truth extension to stress that it is defined relatively to a given semantics **M**

M Truth Extension Formal Definition

Definition

Given a **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 $\nabla \in C_1$, $\circ \in C_2$, and for any formulas A, $B \in \mathcal{F}$,

$$v^*(\nabla A) = \nabla v^*(A) \quad \text{and} \quad v^*((A \circ B)) = \circ(v^*(A), v^*(B))$$

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

M Truth Extension Formal Definition

Remark

The symbols on the **left-hand side** of the equations

$$v^*(\triangledown A) = \triangledown v^*(A) \quad \text{and} \quad v^*((A \circ B)) = \circ(v^*(A), v^*(B))$$

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**

M Truth Extension Formal Definition

We use names " **M truth assignment**" and " **M truth extension**" to stress that we define them for the set of logical values of the semantics **M**

Notation Remark

For any function g, we use a symbol g^* to denote its **extension** to a larger domain

Mathematician often use the same symbol g for both a function g and its extension g^*

Satisfaction and Model

Step 3 The notions of **satisfaction** and **model** are interchangeable in **M** semantics and we define them as follows.

Definition

Given an **M** truth assignment $v : VAR \longrightarrow LV$ and its **M** truth extension v^*. Let $T \in LV$ be the distinguished logical truth 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 an **M model** for the formula A

Counter Model

Definition

Given an **M** truth assignment $v : VAR \longrightarrow LV$ and its **M** truth extension v^*. Let $T \in LV$ be the distinguished logical truth value

We say that the truth assignment v **M does not satisfy** a formula A if and only if $v^*(A) \neq T$

We write symbolically

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

Any truth assignment v, such that $v \not\models_{\mathbf{M}} A$ is called an **M counter model** for the formula A

M Tautology

Step 4 We define the notion of **M tautology** as follows

Definition

A formula A is an **M tautology** if and only if

$v \models_{\mathbf{M}} A$, for all truth assignments $v : VAR \longrightarrow LV$

We denote it as

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

We also say that

A is an **M tautology** if and only if all truth assignments $v : VAR \longrightarrow LV$ are **M models** for A

M Tautology

Observe that directly from definition of the **M model** we get the following equivalent form of the definition of **tautology**

Definition

A formula A is an **M tautology** if and only if

$v^*(A) = T$, for all truth assignments $v : VAR \longrightarrow LV$

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

$$\mathbf{MT} = \{A \in \mathcal{F} : \models_{\mathbf{M}} A\}$$

M Tautology

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 **express** it in a form of the following definition

Definition

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

The semantics **M** is **well defined** if and only if its set **MT** of all tautologies is non empty, i.e. when

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

Extensional Semantics **M**

As the next steps we use the **definitions** established here to define and discuss in details the following particular cases of the extensional semantics **M**

Sets 3, 4, 5: the classical semantics, tautologies, consistency, independence, equivalence of languages

Set 6: Some examples of many valued semantics

Set 7: **M** tautologies, **M** consistency, and **M** equivalence of languages

Extensional Semantics **M**

Many valued **semantics** have 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 3- valued logic, for short

The other logics defined by **extensional semantics** followed and we will discuss some of them

In particular we present Heyting's 3-valued **semantics** as an introduction to the discussion of **first** ever semantics for the intuitionistic logic and some modal logics

Challenge Exercise

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

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

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

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

4. Define formally your own extensional semantics **M** for your language $\mathcal{L}_{CON}$

Write carefully all **Steps 1- 4** of the definition of your **M**

Chapter 3
Propositional Semantics: Classical and Many Valued

Slides Set 3

PART 4 **Classical Semantics**

Semantics- General Principles

Given a propositional language $\mathcal{L} = \mathcal{L}_{CON}$

Symbols for connectives of $\mathcal{L}$ always have some intuitive meaning

Semantics provides a formal definition of the meaning of these symbols

It also provides a method of defining a notion of a **tautology**, i.e. of a formula of the language that is always true under the given semantics

Extensional Connectives

In **Chapter 2** we described the intuitive classical propositional semantics and its motivation and introduced the following notion of extensional connectives

Extensional connectives are the propositional connectives that have the following property:
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 also assumed that

All classical **propositional connectives**

$$\neg, \; \cup, \; \cap, \; \Rightarrow, \; \Leftrightarrow, \; \uparrow, \; \downarrow$$

are **extensional**

Non-Extensional Connectives

We have also observed the following

Remark

In everyday language there are expressions such as

"I believe that", "it is possible that", " certainly", etc....

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

General Definition of Extensional Connectives

We will adopt a following **general** definitions of extensional connectives and extensional semantics introduced in **Lecture 2** to the case of **classical semantics**, so we repeat it here

Definition

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 $\triangledown \in C_1$ or $\circ \in C_2$ is called **extensional** if it is defined by a respective function

$$\triangledown : LV \longrightarrow LV \quad \text{or} \quad \circ : LV \times LV \longrightarrow LV$$

General Extensional Semantics Formal Definition

Definition

Any formal definition of an **extensional semantics** **M** consists of specifying the following steps

Step 1 We define a set LV of logical values, its distinguished value T, and define all connectives of $\mathcal{L}_{CON}$ to be **extensional**

Step 2 We define notion of a truth assignment and its extension

Step 3 We define notions of satisfaction, model, counter model

Step 4 We define notion of a **tautology** under the semantics **M**

Classical Semantics

We adopt **Steps 1- 4** of the definition of extensional semantics to the case of the classical propositional logic as follows

Step 1 We define the language, set of logical values, and define all connectives of the language to be **extensional**

The **language** is

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

The set of logical values is

$$LV = \{T,\ F\}$$

The letters T, F stand as symbols of **truth** and —bf falsehood, respectively

We adopt T as the distinguished value

Classical Connectives

Definition of connectives

Negation $\neg$ is a **function:**

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

such that

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

Notation

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

Classical Connectives

Conjunction $\cap$ is a **function:**

$$\cap : \{T, F\} \times \{T, F\} \longrightarrow \{T, F\}$$

such that

$$\cap(T,T) = T, \quad \cap(T,F) = F, \quad \cap(F,T) = F, \quad \cap(F,F) = F$$

We write it as

$$T \cap T = T, \quad T \cap F = F, \quad F \cap T = F, \quad F \cap F = F$$

Classical Connectives

Disjunction $\cup$ is a **function:**

$$\cup : \quad \{T, F\} \times \{T, F\} \longrightarrow \{T, F\}$$

such that

$$\cup(T, T) = T, \quad \cup(T, F) = T, \quad \cup(F, T) = T, \quad \cup(F, F) = F$$

We write it as

$$T \cup T = T, \quad T \cup F = T, \quad F \cup T = T, \quad F \cup F = F$$

Classical Connectives

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

$$\Rightarrow: \quad \{T, F\} \times \{T, F\} \longrightarrow \{T, F\}$$

such that

$$\Rightarrow (T, T) = T, \quad \Rightarrow (T, F) = F, \quad \Rightarrow (F, T) = T, \quad \Rightarrow (F, F) = T$$

We write it as

$$T \Rightarrow T = T, \quad T \Rightarrow F = F, \quad F \Rightarrow T = T, \quad F \Rightarrow F = T$$

Classical Connectives

Equivalence $\Leftrightarrow$ is a **function:**

$$\Leftrightarrow: \quad \{T, F\} \times \{T, F\} \longrightarrow \{T, F\}$$

such that

$$\Leftrightarrow (T, T) = T, \quad \Leftrightarrow (T, F) = F, \quad \Leftrightarrow (F, T) = F, \quad \Leftrightarrow (T, T) = T$$

We write it as

$$T \Leftrightarrow T = T, \quad T \Leftrightarrow F = F, \quad F \Leftrightarrow T = F, \quad T \Leftrightarrow T = T$$

Classical Connectives Truth Tables

We write the **functions** defining connectives in a form of tables, usually called the classical truth tables

Negation

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

$\neg$	T	F
	F	T

Conjunction

$$T \cap T = T, \quad T \cap F = F, \quad F \cap T = F, \quad F \cap F = F$$

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

Classical Connectives Truth Tables

Disjunction

$$T \cup T = T, \quad T \cup F = T, \quad F \cup T = T, \quad F \cup F = F$$

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

Implication

$$T \Rightarrow T = T, \quad T \Rightarrow F = F, \quad F \Rightarrow T = T, \quad F \Rightarrow F = T$$

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

Classical Connectives Truth Tables

Equivalence

$$T \Leftrightarrow T = T, \; T \Leftrightarrow F = F, \; F \Leftrightarrow T = F, \; F \Leftrightarrow F = T$$

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

This ends the **Step1** of the classical semantics definition

Classical Connectives

Special Properties

Classical semantics is a **special** one. Classical connectives have some **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 new logic being created

Definability of Connectives

We adopt the following definition

Definition

A connective $\circ \in CON$ is **definable** in terms of some connectives $\circ_1, \circ_2, ...\circ_n \in CON$ iff $\circ$ is a **certain function composition** of functions $\circ_1, \circ_2, ...\circ_n$

Example

Classical implication $\Rightarrow$ is **definable** in terms of $\cup$ and $\neg$ because $\Rightarrow$ can be defined as a **composition** of functions $\neg$ and $\cup$

More precisely, a function $h : \{T, F\} \times \{T, F\} \longrightarrow \{T, F\}$ defined by a formula

$$h(x, y) = \cup(\neg x, y)$$

is a **composition of functions** $\neg$ and $\cup$ and we **prove** that the implication function $\Rightarrow$ is equal with h

Short Review: Equality of Functions

Definition

Given two sets A, B and functions f, g such that

$$f:\ A\ \longrightarrow B \quad \text{and} \quad g:\ A\ \longrightarrow B$$

We say that the functions f, g are **equal** and write is as f = g iff f(x) = g(x) for all elements $x \in A$

Example: Consider functions

$$\Rightarrow:\ \{T,F\} \times \{T,F\} \longrightarrow \{T,F\} \quad \text{and} \quad h : \{T,F\} \times \{T,F\} \longrightarrow \{T,F\}$$

where $\Rightarrow$ is classical implication and h is defined by the formula $h(x,y) = \cup(\neg x, y)$

We **prove** that $\Rightarrow\, = \ h$ by evaluating that
$\Rightarrow (x,y) = h(x,y) = \cup(\neg x, y)$, for all $(x,y) \in \{T,F\} \times \{T,F\}$

Definability of Classical Implication

We re-write formula $\Rightarrow (x, y) = \cup(\neg x, y)$ in our adopted notation as

$$x \Rightarrow y = \neg x \cup y \quad \text{for all} \quad (x, y) \in \{T, F\} \times \{T, F\}$$

and call it a **formula defining** $\Rightarrow$ in terms of $\cup$ and $\neg$

We verify correctness of the **definition** as follows

$T \Rightarrow T = T$ and $\neg T \cup T = F \cup T = T$ **yes**

$T \Rightarrow F = F$ and $\neg T \cup F = F \cup F = F$ **yes**

$F \Rightarrow F = T$ and $\neg F \cup F = T \cup F = T$ **yes**

$F \Rightarrow T = T$ and $\neg F \cup T = T \cup T = T$ **yes**

Definability of Connectives

Exercise 1

Find formulas defining $\cap, \Leftrightarrow$ in terms of $\cup$ and $\neg$

Exercise 2

Find formulas defining

$\Rightarrow, \cup, \Leftrightarrow$ in terms of $\cap$ and $\neg$

Exercise 3

Find formulas defining $\cap, \cup, \Leftrightarrow$ in terms of $\Rightarrow$ and $\neg$

Exercise 4

Find a formula defining $\cup$ in terms of $\Rightarrow$ alone

Two More Classical Connectives

Sheffer Alternative Negation $\uparrow$

$$\uparrow: \quad \{T, F\} \times \{T, F\} \longrightarrow \{T, F\}$$

such that

$$T \uparrow T = F, \quad T \uparrow F = T, \quad F \uparrow T = T, \quad F \uparrow F = T$$

Łukasiewicz Joint Negation $\downarrow$

$$\downarrow: \quad \{T, F\} \times \{T, F\} \longrightarrow \{T, F\}$$

such that

$$T \downarrow T = F, \quad T \downarrow F = F, \quad F \downarrow T = F, \quad F \downarrow F = T$$

Definability of Connectives

Exercise 5

Show that the **Sheffer Alternative Negation** $\uparrow$ defines all classical connectives $\neg$, $\Rightarrow$, $\cup$, $\cap$, $\Leftrightarrow$

Exercise 6

Show that **Łukasiewicz Joint Negation** $\downarrow$ defines all classical connectives $\neg$, $\Rightarrow$, $\cup$, $\cap$, $\Leftrightarrow$

Exercise 7

Show that the two binary connectives: $\downarrow$ and $\uparrow$ **suffice**, each of them separately, to define **all** classical connectives, whether unary or binary

Functional Dependency of Connectives

Definition

Given a propositional language the set CON and its extensional semantics **M**. A property of **defining** the set CON in terms of its proper subset is called a **functional dependency** of connectives under **M**

Proving the property of **functional dependency** consists of identifying a proper subset CON_0 of the set CON, such that each connective $\circ \in CON - CON_0$ is **definable** in terms of connectives from CON_0

Functional Dependency of Connectives

Proving **functional dependency** of a the set CON of a given language under a given semantics **M** is usually a difficult, and often **impossible** task for many semantic

Functional dependency holds in the classical case and we express it as follows

Theorem

The set of connectives of the languages

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

is **functionally dependent** under the classical semantics.

The proof follows from **Exercises 1 - 7**

Semantics Definition: Truth Assignment

Step 2 We define the next components of the classical **semantics** in terms of the propositional connectives as defined in the **Step 1** and a function called truth assignment

Definition

A **truth assignment** is any function

$$v : VAR \longrightarrow \{T, F\}$$

Observe that the domain of truth assignment is the set of propositional variables, i.e. the truth assignment is defined only for atomic formulas

Truth Assignment Extension

We **extend** now the truth assignment v to the set $\mathcal{F}$ of all formulas

We do so in order to **define** formally the logical value for any formula $A \in \mathcal{F}$

The definition of the **extension** of the truth assignment v to the set $\mathcal{F}$ follows the same pattern for the all extensional connectives, i.e. for all extensional semantics

Truth Assignment Extension v^* to $\mathcal{F}$

Definition

Given the truth assignment

$$v : VAR \longrightarrow \{T, F\}$$

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

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

such that the following conditions are satisfied

(i) for any $a \in VAR$

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

Truth Assignment Extension v^* to $\mathcal{F}$

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

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

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

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

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

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

The symbols on the **left-hand side** of the equations represent connectives in their natural language meaning and

the symbols on the **right-hand side** represent connectives in their semantical meaning given by the classical truth tables

Extension v^* Definition Revisited

Notation

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

The **condition (ii)** of the definition of the extension v^* can be hence **written** as follows

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

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

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

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

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

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

We will use this notation for the rest of the book

Truth Assignment Extension Example

Consider a formula

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

and a truth assignment v such that

$$v(a) = T, \quad v(b) = F$$

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 $v^*(A)$ of the formula A

We say: "v such that" - as we consider its values for the set $\{a, b\} \subseteq VAR$

Nevertheless, the **domain** of v is the set VAR of all variables and we have to **remember** that

Truth Assignment Extension Example

Given a formula A: $((a \Rightarrow b) \cup \neg a))$ and a truth assignment v such that v(a) = T, v(b) = F

We calculate the logical value of the formula A as follows:

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

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

$\cup(\Rightarrow(T, F), \neg T)) = \cup(F, F) = F$

We can also calculate it as follows:

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

$(v(a) \Rightarrow v(b)) \cup \neg v(a) = (T \Rightarrow F) \cup \neg T = F \cup F = F$

We write it in a **short-hand** notation as

$(T \Rightarrow F) \cup \neg T = F \cup F = F$

Semantics: Satisfaction Relation

Step 3 We define notions of satisfaction, model, counter model

Definition Let $v : VAR \longrightarrow \{T, F\}$ be a truth assignment

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

Notation: $v \models A$

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

Notation: $v \not\models A$

The relation $\models$ is called a **satisfaction relation**

Semantics: Satisfaction Relation

Observe that $v^*(A) \neq T$ is is equivalent to the fact that $v^*(A) = F$ **only** in 2-valued semantics and so we also write

$$v \not\models A \quad \text{if and only if} \quad v^*(A) = F$$

Definition

We say that v **falsifies** A if and only if $v^*(A) = F$

Remark

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

$v \not\models A$ if and only if v **falsifies** the formula A

Examples

Example 1 : Let $A = ((a \Rightarrow b) \cup \neg a))$ and $v : VAR \longrightarrow \{T, F\}$ be such that $v(a) = T, v(b) = F$

We calculate $v^*(A)$ using a **short hand** notation as follows

$$(T \Rightarrow F) \cup \neg T = F \cup F = F$$

By definitiom

$$v \not\models ((a \Rightarrow b) \cup \neg a))$$

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

Examples

Example 2

Let $A = ((a \cap \neg b) \cup \neg c)$ and $v : VAR \longrightarrow \{T, F\}$ be such that $v(a) = T,\ v(b) = F,\ v(c) = T$

We calculate $v^*(A)$ using a **short hand** notation as follows

$$(T \cap \neg F) \cup \neg T = (T \cap T) \cup F = T \cup F = T$$

By definition

$$v \models ((a \cap \neg b) \cup \neg c)$$

Examples

Example 3

Let $A = ((a \cap \neg b) \cup \neg c)$

Consider now $v_1 : VAR \longrightarrow \{T, F\}$ such that
$v_1(a) = T,\ v_1(b) = F,\ v_1(c) = T$ and
$v_1(x) = F,$ for all $x \in VAR - \{a, b, c\}$

Observe that

$v(a) = v_1(a),\ \ v(b) = v_1(b),\ \ v(c) = v_1(c)$

Hence we get

$$v_1 \models ((a \cap \neg b) \cup \neg c)$$

Examples

Example 4

Let $A = ((a \cap \neg b) \cup \neg c)$

Consider now $v_2 : VAR \longrightarrow \{T, F\}$ such that

$v_2(a) = T, v_2(b) = F, v_2(c) = T, v_2(d) = T$ and

$v_1(x) = F,$ for all $x \in VAR - \{a, b, c, d\}$

Observe that

$v(a) = v_2(a), v(b) = v_2(b), v(c) = v_2(c)$

Hence we get

$$v_2 \models ((a \cap \neg b) \cup \neg c)$$

Semantics: Model, Counter-Model

Definition:

Given a formula $A \in \mathcal{F}$ and $v : VAR \longrightarrow \{T, F\}$

Any v 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 A

Observe that all truth assignments v, v_1, v_2 from our **Examples 2, 3, 4** are **models** for the same formula A

Semantics: Tautology

Step 4 Classical tautology definition

Definition 1

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

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

The second definition uses the notion of satisfaction and model and the fact that in any extensional semantic these notions interchangeable

Definition 2

A is a **tautology** if and only if **any** $v : VAR \longrightarrow \{T, F\}$, $v \models A$, i.e. **any** v is a **model** for A

We write symbolically

$$\models A$$

for the statement "A is a tautology"

Semantics: not a tautology

Definition 1

A is **not a tautology** if and only if there is v, such that $v^*(A) \neq T$

Definition 2

A is **not a tautology** if and only if A has a **counter-model**

Notation

We write $\not\models A$ to denote the statement "A is not a tautology"

This ends the **formal definition** of the classical propositional **semantics** that follows the pattern for extensional semantics established in **Lecture 2**

Chapter 3
Propositional Semantics: Classical and Many Valued

Slides Set 4

PART 5 **Tautologies:** Decidability and Verification Methods

PART 6 **Sets of Formulas:** Consistency and Independence

Chapter 3
Propositional Semantics: Classical and Many Valued

Slides Set 4

PART 5 **Tautologies:** Decidability and Verification Methods

Classical Tautologies

There is a large number of basic and important propositional **tautologies** listed and discussed in **Chapter 2**

We **assume** that at this point everybody is familiar, or will familiarize with them if needed

Chapter 2 provides the motivation for **classical approach** to definition of **tautologies** as ways of describing correct rules of our mathematical reasoning

Chapter 2 also contains an **informal** definition of classical semantics and discusses some tautology verification methods

Classical Tautologies

Here is the formal definition of classical **tautology**

Definition

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

A is a **tautology** if and only if $v^*(A) = T$, for **all** truth assignments $v : VAR \longrightarrow \{T, F\}$. We denote it as

$$\models A$$

Our goal now is to **prove** that the notion of classical tautology is **decidable** and to prove correctness of the **tautology verification** method presented in Chapter 2

Moreover we present here other tautology **verification methods** and prove their correctness

Decidability and Verification

We start now a natural **question**:

How do we verify whether a given formula $A \in \mathcal{F}$ **is** or **is not** a tautology?

The **answer** seems to be very simple

By tautology definition we have to examine all truth assignments $v : VAR \longrightarrow \{T, F\}$

If they all evaluate to T, we **proved** that $\models A$

If at least one evaluates to F, we found a **counter model** and proved $\not\models A$

The verification process is **decidable**, if the we have only a **finite** number of v to consider

Decidability and Verification

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 any $n \in N$

Cardinality of Sets

We have special names and notations for the **cardinalities** of infinite sets

In particular we write

$$|X| = \aleph_0$$

and say " **cardinality** of X is **aleph zero**," for any countably infinite set X, i.e. the set that has the **same** cardinality as natural numbers

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

Counting Functions

Counting Functions Theorem 1

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

Counting Truth Assignments

In our case of counting the truth assignments

$$v : VAR \longrightarrow \{T, F\}$$

we have that $|VAR| = \aleph_0$ and $|\{T, F\}| = 2$

We know that $2^{\aleph_0} = C$ and hence we get directly from Counting Functions **Theorem 1** the following

Truth Assignments Theorem

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

Restricted Truth Assignments

To address and to answer these questions **formally** we first introduce some notations and definitions

Notation For any formula A, we denote by

$$VAR_A$$

a set of all variables that **appear** in A

Definition

Given $v : VAR \longrightarrow \{T, F\}$, any function

$$v_A : VAR_A \longrightarrow \{T, F\}$$

such that $v(a) = v_A(a)$ for all $a \in VAR_A$ is called a **restriction** of v to the formula A

Restricted Model

Restricted Model Theorem

For any formula A, any v, and its **restriction** v_A

$$v \models A \quad \text{ij and only if} \quad v_A \models A$$

Definition: Given a formula $A \in \mathcal{F}$, any function

$$w : \quad VAR_A \longrightarrow \{T, F\}$$

is called a truth assignment **restricted** to A

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

Any function

$$w : \quad VAR_A \longrightarrow \{T, F\} \quad \text{such that} \quad w^*(A) = T$$

is called a **restricted model** for A

Example

Example

$$A = ((a \cap \neg b) \cup \neg c)$$

$$VAR_A = \{a, b, c\}$$

Truth assignment **restricted** to A is any function:

$$w : \quad \{a, b, c\} \longrightarrow \{T, F\}.$$

We use the following theorem to count all possible truth assignment **restricted** to A

Counting Functions

Counting Functions Theorem 2

For any **finite** sets A and B,
if the set A has n elements and B has m elements, then
there are m^n **possible functions** that map A into B

Proof by Mathematical Induction over m

Example

There are $2^3 = 8$ truth assignments w **restricted** to

$$A = ((a \Rightarrow \neg b) \cup \neg c)$$

Counting Functions

Counting Restricted Truth

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

$$2^{|VAR_A|}$$

possible truth assignments **restricted** to A

Example

Let $A = ((a \cap \neg b) \cup \neg c)$

All w **restricted** to A are listed in the table below

w	a	b	c	$w^*(A)$ computation	$w^*(A)$
w_1	T	T	T	$(T \Rightarrow T) \cup \neg T = T \cup F = T$	T
w_2	T	T	F	$(T \Rightarrow T) \cup \neg F = T \cup T = T$	T
w_3	T	F	F	$(T \Rightarrow F) \cup \neg F = F \cup T = T$	T
w_4	F	F	T	$(F \Rightarrow F) \cup \neg T = T \cup F = T$	T
w_5	F	T	T	$(F \Rightarrow T) \cup \neg T = T \cup F = T$	T
w_6	F	T	F	$(F \Rightarrow T) \cup \neg F = T \cup T = T$	T
w_7	T	F	T	$(T \Rightarrow F) \cup \neg T = F \cup F = F$	F
w_8	F	F	F	$(F \Rightarrow F) \cup \neg F = T \cup T = T$	T

$w_1, w_2, w_3, w_4 w_5, w_6, w_8$ are restricted models for A

w_7 is a restricted counter- model for A

Restrictions and Extensions

Given a formula A and $w: VAR_A \longrightarrow \{T, F\}$

Extension Definition

Any function v, such that $v: VAR \longrightarrow \{T, F\}$ and $v(a) = w(a)$, for all $a \in VAR_A$ is called an **extension** of w to the set VAR of all propositional variables

Extension Fact

For any formula A, any w **restricted** to A, and any of its **extensions** v

$$w \models A \quad \text{if and only if} \quad v \models A$$

Tautology Theorem

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

$\models A$ if and only if $v_A \models A$ for all $v_A : VAR_A \longrightarrow \{T, F\}$

Proof Assume $\models A$

By tautology definition $v \models A$ 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 **extension** of some v_A, i.e. $v(a) = v_A(a)$ for all $a \in VAR_A$. By the **extension definition** we get that $v^*(A) = {v_A}^*(A) = T$ and $v \models A$

Tautology Decidability

Directly from **Tautology Theorem** we get the proof of decidability of the notion of classical propositional tautology

Decidability Theorem

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

$$\models A \quad \text{or} \quad \not\models A,$$

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

We present now some tautologies verification methods

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

If we **find** a truth assignment which evaluates A to F,

we **stop** and give **answer**: $\not\models A$

Otherwise we continue

If **all** truth assignments evaluate A to T,

we give we **stop** and **answer**: $\models A$

We usually **list all** restricted truth assignments v_A in a form of a **truth table**, hence the name of the method

Truth Table Method Example

Consider a formula A:

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

We write the Truth Table:

w	a	b	$w^*(A)$ computation	$w^*(A)$
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

We evaluated that for all w **restricted** to A, i.e. all functions $w : VAR_A \longrightarrow \{T, F\}$, $w \models A$

This proves

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

Tautology Verification

Imagine now that A has for example 200 variables.

To find whether A is a tautology by using the **Truth Table Method** one would have to evaluate 200 variables long expressions - not to mention that one would have to list 2^{200} **restricted** truth assignments

We use now and later in case of many valued semantics a more elegant and faster method called **Proof by Contradiction Method**

Proof by Contradiction Method

in order to verify whether $\models A$ one works backwards trying to find a truth assignment v which makes a formula A **false**

If we **find one**, it means that A **is not** a tautology

if we prove that it is **impossible**, i.e. we got a **contradiction** it means that the formula A is a tautology

Example

Let $A = (a \Rightarrow (a \cup b))$

Step 1: Assume that $\not\models A$, i.e. we write in a shorthand notion $A = F$

Step 2: We use shorthand notation to analyze **Strep 1**

$(a \Rightarrow (a \cup b)) = F$ if and only if $a = T$ and $(a \cup b) = F$

Step 3: Analyze **Step 2**

$a = T$ and $(a \cup b) = F$, i.e. $(T \cup b) = F$

This is **impossible** by the definition of $\cup$

We got a **contradiction**, hence

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

Substitution Example

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

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

The following formulas are also **tautologies**

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

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

because they are particular cases - **substitutions** - of $(a \Rightarrow (a \cup b))$

Substitution Method

This method allows us to obtain new tautologies from formulas already proven to be tautologies.

Example

We can obtain the formula

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

from a formula $(a \Rightarrow (a \cup b))$ by a proper **substitutions** (replacements) of more complicated formulas for the variables a and b in a formula $(a \Rightarrow (a \cup b))$

Substitution Method

We write

$$A(a, b) = (a \Rightarrow (a \cup b))$$

to **denote** that $(a \Rightarrow (a \cup b))$ is a formula A with two variables a and b

We denote by

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

a result of a **substitution** of formulas A_1, A_2 on a place of the variables a and b , everywhere where they appear in the formula $A(a, b)$

Substitution Example

Example

Given a formula $A(a,b) = (a \Rightarrow (a \cup b))$

Making a substitution **s1**

$$A(a/((a \Rightarrow b) \cap \neg c),\ \ b/\neg d)$$

we get a formula

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

Substitution Example

Making a substitution **s2**

$$A(a/((a \Rightarrow b) \cap \neg c),\ \ b/((a \Rightarrow \neg e))$$

we get a formula

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

We know $\models (a \Rightarrow (a \cup b))$

By correctness (to be proved) of the **Substitution Method** we know that also both formulas obtained by substitutions **s1** and **s2** are also tautologies

Substitution Correctness

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

Substitution Method correctness is established by the following Theorem

Correctness Theorem

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

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

Proof of Substitution Correctness

Correctness Theorem

For any formulas $A, \ A_1, ... A_n \in \mathcal{F}$,

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 $B = A(a_1/A_1, ..., a_n/A_n)$ and let $b_1, b_2, ... b_m$ be all propositional variables which occur in $A_1, ... A_n$

Given a truth assignment $v : VAR \longrightarrow \{T, F\}$, the values $v(b_1), v(b_2), ... v(b_m)$ define $v^*(A_1), ... v^*(A_n)$ and, in turn define $v^*(A(a_1/A_1, ..., a_n/A_n))$

Proof of Substitution Method Correctness

Let now $w : VAR \longrightarrow \{T, F\}$ 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 $\models A$ and $w^*(A) = T$, for all possible w, hence $v^*(B) = w^*(A) = T$ for all truth assignments w and we have $\models B$

Constructing New Tautologies

Observation

The **Correctness Theorem** establishes validity of use of the **Substitution Method** as a method of constructing new tautologies from given tautologies

Example

We know that $\models (a \cup \neg a)$ and $A(a)$ is $(a \cup \neg a)$

Making a substitution

$$A(a/((a \Rightarrow b) \cap \neg c)$$

we get a **new tautology**

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

Generalization Method

Generalization Method consists of representing, if it is **possible**, a given formula A as a **particular case** of some much simpler and more **general** formula B

We then can use any other verification method to examine whether the representation B of the given formula A **is** or **is not** a tautology

Generalization Method

Example

Given a formula

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

We **represent** it as a simple and more general formula

$$(A \Rightarrow (A \cup B))$$

for $A = ((a \Rightarrow b) \cap \neg c)$ and $B = \neg d$

We then prove using, for example, **Proof by Contradiction Method** that

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

Set of all Tautologies

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

Definition

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

$$=\mid A$$

Directly from the definition we have that

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

Set of all Contradictions

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

Examples

Tautology $(A \Rightarrow (B \Rightarrow A))$

Contradiction $(A \cap \neg A)$

Neither $(a \cup \neg b)$

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

Any v such that $v(a) = T$ is a **model** for $(a \cup \neg b)$, so it is **not a contradiction**

Any v such that $v(a) = F, \ v(b) = T$ is a **counter-model** for $(a \cup \neg b)$ so $\not\models (a \cup \neg b)$

Simple Properties

Theorem 1 For any formula $A \in \mathcal{F}$ the following conditions are equivalent.

(1) $A \in \mathbf{T}$

(2) $\neg A \in \mathbf{C}$

(3) For all v, $v \models A$

Theorem 2 For any formula $A \in \mathcal{F}$ the following conditions are equivalent.

(1) $A \in \mathbf{C}$

(2) $\neg A \in \mathbf{T}$

(6) For all v, $v \not\models A$

Chapter 3
Propositional Semantics: Classical and Many Valued

Slides Set 4

PART 6 **Sets of Formulas:** Consistency and Independence

Models for Sets of Formulas

Consider $\mathcal{L} = \mathcal{L}_{CON}$ and let $\mathcal{S} \neq \emptyset$ be any non empty set of formulas of $\mathcal{L}$, i.e.

$$\mathcal{S} \subseteq \mathcal{F}$$

We adopt the following definition.

Definition

A truth truth assignment $v : VAR \longrightarrow \{T, F\}$

is a **model for the set** $\mathcal{S}$ of formulas if and only if

$$v \models A \quad \text{for } \textbf{all} \text{ formulas } A \in \mathcal{S}$$

We write

$$v \models \mathcal{S}$$

to denote that v is a **model** for the set $\mathcal{S}$ of formulas

Counter- Models for Sets of Formulas

Similarly, we define a notion of a **counter-model**

Definition

A truth assignment $v : VAR \longrightarrow \{T, F\}$

is a **counter-model** for the set $\mathcal{S} \neq \emptyset$ of formulas

if and only if

$v \not\models A$ for **some** formula $A \in \mathcal{S}$

We write

$$v \not\models \mathcal{S}$$

to denote that v is a **counter- model** for the set $\mathcal{S}$ of formulas

Restricted Model for Sets of Formulas

Remark that the set $\mathcal{S}$ can be **finite**, or **infinite**

In a case when $\mathcal{S}$ is a **finite** subset of formulas we define, as before, a notion of restricted model and restricted counter-model

Definition

Let $\mathcal{S}$ be a **finite** subset of formulas and $v \models \mathcal{S}$

Any restriction of the model v to the domain

$$VAR_{\mathcal{S}} = \bigcup_{A \in \mathcal{S}} VAR_A$$

is called a **restricted model** for $\mathcal{S}$

Restricted Counter - Model for Sets of Formulas

Definition

Any restriction of a **counter-model** v of a set $\mathcal{S} \neq \emptyset$ of formulas to the domain

$$VAR_{\mathcal{S}} = \bigcup_{A \in \mathcal{S}} VAR_A$$

is called a **restricted counter-model** for $\mathcal{S}$

Example

Example

Let $\mathcal{L} = \mathcal{L}_{\{\neg,\cap\}}$ and let

$$\mathcal{S} = \{a,\ (a \cap \neg b),\ c,\ \neg b\}$$

We have $VAR_{\mathcal{S}} = \{a, b, c\}$ and atruth assignment
$v : VAR_{\mathcal{S}} \rightarrow \{T, F\}$ such that
$v(a) = T,\ \ v(c) = T,\ \ v(b) = F$
is a **restricted model** for $\mathcal{S}$

A truth assignment $v : VAR_{\mathcal{S}} \rightarrow \{T, F\}$ such that $v(a) = F$
is a **restricted counter-model** for $\mathcal{S}$

Models for Infinite Sets

The set $\mathcal{S}$ from the previous **example** was a finite set

Some natural questions arise:

Q1 Give an example of an infinite set $\mathcal{S}$ that **has a model**

Q2 Give an example of an infinite set $\mathcal{S}$ that **does not** have **model**

Here are simple, natural examples

Q1 Example

Consider set **T** of all **tautologies**

It is a countably infinite set and by definition of a tautology any v is a **model** for **T**, i.e. $v \models \mathbf{T}$

Models for Infinite Sets

Q2 Give an example of an infinite set $\mathcal{S}$ that **does not** have **model**

Q2 Example

Consider set **C** of all **contradictions**

It is a countably infinite set and

for any v, $v \not\models$ **C** by definition of a contradiction, i.e. any any v is a **counter-model** for **C**

Models for Infinite Sets

Here are some more a bit more difficult natural questions

Q3 Give an example of an infinite set $\mathcal{S}$, such that $\mathcal{S} \neq \mathbf{T}$ and $\mathcal{S}$ **has a model**

Q4 Give an example of an infinite set $\mathcal{S}$, such that $\mathcal{S} \cap \mathbf{T} = \emptyset$ and $\mathcal{S}$ **has a model**

Q5 Give an example of an infinite set $\mathcal{S}$, such that $\mathcal{S} \neq \mathbf{C}$ and $\mathcal{S}$ **does not** have a **model**

Q6 Give an example of an infinite set $\mathcal{S}$, such that $\mathcal{S} \neq \mathbf{C}$ and $\mathcal{S}$ **has a counter model**

Q7 Give an example of an infinite set $\mathcal{S}$, such that $\mathcal{S} \cap \mathbf{C} = \emptyset$ and $\mathcal{S}$ **has a counter model**

Consistent Sets of Formulas

Definition

A set $\mathcal{G} \subseteq \mathcal{F}$ of formulas is called **consistent** if and only if $\mathcal{G}$ **has a model**, i.e. we say hat

$\mathcal{G} \subseteq \mathcal{F}$ is **consistent** if and only if **there is** v such that $v \models \mathcal{G}$

Otherwise $\mathcal{G}$ is called **inconsistent**

More Questions

Here are some more of natural questions

Q8 Give an example of an infinite set $\mathcal{S}$, such that $\mathcal{S} \neq \mathbf{T}$ and $\mathcal{S}$ is **consistent**

Q9 Give an example of an infinite set $\mathcal{S}$, such that $\mathcal{S} \cap \mathbf{T} = \emptyset$ and $\mathcal{S}$ is **consistent**

Q10 Give an example of an infinite set $\mathcal{S}$, such that $\mathcal{S} \neq \mathbf{C}$ and $\mathcal{S}$ is **inconsistent**

Q11 Give an example of an infinite set $\mathcal{S}$, such that $\mathcal{S} \cap \mathbf{C} = \emptyset$ and $\mathcal{S}$ is **inconsistent**

Independent Statements

Definition

A formula A is called **independent** from a set $\mathcal{G} \subseteq \mathcal{F}$ if and only if **there are** truth assignments v_1, v_2 such that

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

i.e. we say that a formula A is **independent** if and only if

$$\mathcal{G} \cup \{A\} \quad \text{and} \quad \mathcal{G} \cup \{\neg A\} \quad \text{are } \textbf{consistent}$$

Example

Example

Given a set

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

Show that $\mathcal{G}$ is **consistent**

Solution

We have to find $v : VAR \longrightarrow \{T, F\}$ such that

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

It means that we need to **find** v such that

$$v^*((a \cap b) \Rightarrow b) = T, \quad v^*(a \cup b) = T, \quad v^*(\neg a) = T$$

Consistent: Example

To prove hat $\mathcal{G}$ is **consistent** we have to consider the following case

1. Formula $((a \cap b) \Rightarrow b)$ is a tautology, i.e. $v^*((a \cap b) \Rightarrow b) = T$ for any v and we do not need to consider it anymore.

2. Formula $\neg a = T$ (we use shorthand notation) if and only if $a = F$ so we get that v must be such that $v(a) = F$

3. We want $(a \cup b) = T$ but v is such that $v(a) = F$ so $(a \cup b) = F \cup b = T)$ if and only if $b = T$

This **means** that for any $v : VAR \longrightarrow \{T, F\}$ such that $v(a) = F, \ v(b) = T$

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

and we **proved** that $\mathcal{G}$ is **consistent**

Independent: Example

Example

Show that a formula $A = ((a \Rightarrow b) \cap c)$ is **independent** of

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

Solution

We construct $v_1,\ v_2 : VAR \longrightarrow \{T, F\}$ such that

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

We have just proved that any $v : VAR \longrightarrow \{T, F\}$ such that $v(a) = F,\ \ v(b) = T$ is a **model** for $\mathcal{G}$

Independent: Example

Take as v_1 any truth assignment such that

$v_1(a) = v(a) = F, \quad v_1(b) = v(b) = T, \quad v_1(c) = T$

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

$v_2(a) = v(a) = F, \quad v_2(b) = v(b) = T, \quad v_2(c) = F$

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 A is **independent** of $\mathcal{G}$

Not Independent: Example

Example

Show that a formula $A = (\neg a \cap b)$ is **not independent** of

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

Solution

We have to show that **it is impossible** to construct v_1, v_2 such that

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

Observe that we have just proved that any v such that $v(a) = F$, and $v(b) = T$ is the only model restricted to the set of variables $\{a, b\}$ for $\mathcal{G}$ and $\{a, b\} = VAR_A$

So we have to check now if it is **possible** $v \models A$ and $v \models \neg A$

Not Independent: Example

We have to evaluate $v^*(A)$ and $v^*(\neg A)$ for

$v(a) = F$, and $v(b) = T$

$v^*(A) = v^*((\neg a \cap b) = \neg v(a) \cap v(b) = \neg F \cap T = T \cap T = T$

and so $v \models A$

$v^*(\neg A) = \neg v^*(A) = \neg T = F$

and so $v \not\models \neg A$

This end the proof that A is **not independent** of $\mathcal{G}$

Example

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

Observe that 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 and v_2, and

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

Let's consider $A = c$, then $\mathcal{G} \cup \{A\} = \{a, (a \Rightarrow b), c\}$

A truth assignment v_1, such that $v_1(a) = T$, $v_1(b) = T$ and $v_1(c) = T$ is a **model** for $\mathcal{G} \cup \{A\}$

Likewise for $\mathcal{G} \cup \{\neg A\} = \{a, (a \Rightarrow b), \neg c\}$

Any v_2, such that $v_2(a) = T$, $v_2(b) = T$ and $v_2(c) = F$ is a **model** for $\mathcal{G} \cup \{\neg A\}$ and so the formula A is **independent**

Challenge Problem

Challenge Problem

Find an infinite number of formulas that are **independent** of a set

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

Challenge Problem Solution

This my solution - there are many others- this one seemed to me the **most simple**

Solution

We just proved that any v such that $v(a) = F, \ \ v(b) = T$ is **the only** model restricted to the set of variables $\{a, b\}$ and so all other possible models for $\mathcal{G}$ must be **extensions** of v

Challenge Problem Solution

We **define** a countably infinite set of formulas (and their negations) and corresponding **extensions** of v (restricted to to the set of variables $\{a, b\}$) such that $v \models \mathcal{G}$ as follows

Observe that **all extensions** of v restricted to to the set of variables $\{a, b\}$ have as domain the infinitely countable set

$$VAR = \{a_1,\ a_2,\ \ldots, a_n \ldots\}$$

We take as an infinite set of formulas in which every formula **independent** of $\mathcal{G}$ the set of **atomic formulas**

$$\mathcal{F}_0 = \{a_1,\ a_2,\ \ldots, a_n \ldots\} - \{a,\ b\}$$

Challenge Problem Solution

Let $c \in \mathcal{F}_0 = \{a_1,\ a_2,\ \ldots, a_n.\ldots\} - \{a,\ b\}$

We define truth assignments $v_1,\ v_2 : VAR \longrightarrow \{T, F\}$ such that

$$v_1 \models \mathcal{G} \cup \{c\} \text{ and } v_2 \models \mathcal{G} \cup \{\neg c\}$$

as follows

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

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

Chapter 3
Propositional Semantics: Classical and Many Valued

Slides Set 5

PART 7 Classical Tautologies and Logical Equivalences

PART 8 Definability of Connectives and Equivalence of Languages

Chapter 3
Propositional Semantics: Classical and Many Valued

Slides Set 5

PART 7 Classical Tautologies and Logical Equivalences

Classical Tautologies and Equivalence of Languages

We present here as a **first** step a set of most widely used classical tautologies. We will **use them**, in one form or other, in our investigations in future chapters

An extended list of **tautologies** is presented in Chapter 2

As the **second** 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

Classical Tautologies

Some Tautologies

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

Implication and Negation

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

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

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

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

Classical Tautologies

Disjunction, Conjunction

$$(A \Rightarrow (A \cup B)), \ \ (B \Rightarrow (A \cup B)), \ \ ((A \cap B) \Rightarrow A),$$

$$((A \cap B) \Rightarrow A), \ \ ((A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow ((A \cup B) \Rightarrow C))),$$

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

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

$$((\neg A \cup B) \Rightarrow (A \Rightarrow B)), \ \ ((A \Rightarrow B) \Rightarrow (\neg A \cup B)),$$

$$(A \cup \neg A)$$

Classical Tautologies

Contraposition (1)

$$((A \Rightarrow B) \Leftrightarrow (\neg B \Rightarrow \neg A)), \ \ ((B \Rightarrow A) \Leftrightarrow (\neg A \Rightarrow \neg B))$$

Contraposition (2)

$$((\neg A \Rightarrow B) \Leftrightarrow (\neg B \Rightarrow A)), \ \ ((A \Rightarrow \neg B) \Leftrightarrow (B \Rightarrow \neg A))$$

Double Negation

$$(\neg\neg A \Leftrightarrow A)$$

Logical Equivalences

Logical equivalence is a very useful notion to use 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

We say that two formulas formulas A, B 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 $\equiv$ **is not** a logical connective. It is a metalanguage symbol for saying " A, B are **logically equivalent**"

Logical Equivalences

$\equiv$ is a very useful **symbol**. It says that two formulas always have the **same** logical value, hence can be used in the same way we use the **equality** symbol $=$. Formally we define it as follows.

Definition

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

$$A \equiv B \quad \text{if and only if} \quad v^*(A) = v^*(B) \quad \text{for all} \quad v : VAR \longrightarrow \{T, F\}$$

The following property follows directly from the definition

Property

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

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

Logical Equivalences

We, for **example** write the laws of contraposition, and the laws of double negation as **logical equivalences** as follows

E - Contraposition (1)

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

E - Contraposition (2)

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

E - Double Negation

$$\neg\neg A \equiv A$$

Use of Logical Equivalence

We use **logical equivalences** to obtain new Laws from some already known (proved). For **example**, we obtain new Law of **Contraposition** from the **E - Contraposition (1)** Law and the **E - Double Negation** Law as follows

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

We proved a new Law of **Contraposition (1):**

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

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

We proved another new Law of **Contraposition (2):**

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

Substitution Theorem

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

Substitution Theorem

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

$$\text{If} \quad A \equiv B, \quad \text{then} \quad A_1 \equiv B_1$$

Use of Substitution Theorem

Example

Let A_1 be a formula $(C \cup D)$, i.e.

$$A_1 = (C \cup D)$$

and let $B = \neg\neg C, \quad A = C$

We get

$$B_1 = A_1(C/B) = A_1(C/\neg\neg C) = (\neg\neg C \cup D)$$

By **Double Negation** Law

$$\neg\neg C \equiv C \quad \text{i.e.} \quad A \equiv B$$

So we get by **Substitution Theorem** that

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

Use of Substitution Theorem

Exercise

Transform formula a

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

into its logically equivalent formula without implication

Hint: use the the **Substitution Theorem** and the already known **Definability of Connectives** equivalence

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

Remark that it is not the only one equivalence we can use.

Use of Substitution Theorem

We transform via the **Substitution Theorem** a formula

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

into its **logically equivalent** formula as follows

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

$\equiv \neg(\neg C \cup \neg B) \cup (B \cup C))$ and we get that

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

Observe that if the formulas B, C contain $\Rightarrow$ as logical connective we can continue this process until we obtain a logically equivalent formula not containing $\Rightarrow$ at all

Chapter 3
Propositional Semantics: Classical and Many Valued

Slides Set 5

PART 8 Definability of Connectives and Equivalence of Languages

Definability of Connectives Equivalences

The next set of **equivalences** correspond the notion of definability of connectives discussed earlier in the chapter

For **example**, a tautology

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

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

Definability of Implication in terms of negation and disjunction

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

Definability of Connectives Equivalences

Observation

The direct proof of Definability of Connectives equivalences presented here follow directly from the definability formulas developed earlier in the chapter in the the proof of the **Definability of Connectives Theorem**, hence the names

We use the notion of **logical equivalence** instead of the **tautology** notion because it makes the manipulation of formulas much easier

Definability of Connectives Equivalences

Example

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

We use the **Definability of Implication** equivalence to transform A into a **logically equivalent** formula not containing $\Rightarrow$ as follows

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

and hence

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

Definability of Connectives Equivalences

Definability of Implication equivalence

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

allows us, via the **Substitution Theorem**, replace any sub-formula of the form $(A \Rightarrow B)$ of any formula by a formula

$$(\neg A \cup B)$$

Hence it allows us to recursively transform a given formula containing implication into an **logically equivalent** formula that does not contain implication but contains negation and disjunction instead

Equivalence of Languages

The **Substitution Theorem** and the equivalence

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

let us **transform a language** that contains implication **into a language** that does not contain the implication, but contains negation and disjunction instead

Observe that we use this equivalence **recursively**, i.e. if the formulas A, B contain $\Rightarrow$ as logical connective we continue this process until we obtain a logically equivalent formula not containing $\Rightarrow$ at all

Equivalence of Languages

Example

The language $\mathcal{L}_1 = \mathcal{L}_{\{\neg, \cap, \Rightarrow\}}$ becomes a language $\mathcal{L}_2 = \mathcal{L}_{\{\neg, \cap, \cup\}}$ such that all its formulas are **logically equivalent** to the formulas of the language $\mathcal{L}_1$

We write it as the following condition

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

Connectives Elimination

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 use the following logical equivalence

Definability of Disjunction in terms of negation and implication

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

Example

Consider a formula $C = ((A \cup B) \cap \neg A)$

We transform C into its **logically equivalent** form not containing $\cup$ but containing $\Rightarrow$ as follows

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

The **Definability of Disjunction** equivalence allows us transform for example a language

$$\mathcal{L}_2 = \mathcal{L}_{\{\neg,\ \cap,\ \cup\}}$$

into a language

$$\mathcal{L}_1 = \mathcal{L}_{\{\neg,\cap,\Rightarrow\}}$$

with all its formulas being **logically equivalent**

Equivalence of Languages

We write it as the following condition **C2** similar to the condition **C1**

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 the conditions **C1, C2** hold are called **logically equivalent**.

We denote it by

$$\mathcal{L}_1 \equiv \mathcal{L}_2.$$

A general, formal definition goes as follows.

Equivalence of Languages Definition

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**, i.e.

$$\mathcal{L}_1 \equiv \mathcal{L}_2$$

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 $A \equiv B$

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

Equivalence of Languages

Example

To prove the logical equivalence

$$\mathcal{L}_{\{\neg,\cup\}} \equiv \mathcal{L}_{\{\neg,\Rightarrow\}}$$

we need the following logical equivalences

Definability of Implication in terms of disjunction and negation

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

Definability of Disjunction in terms of implication and negation

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

and the **Substitution Theorem**

Equivalence of Languages

Example

To prove the logical equivalence of the languages

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

we need only the **definability of implication** equivalence

It proves, by **Substitution Theorem** that

for any formula A of $\mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow\}}$ **there is** a formula B of $\mathcal{L}_{\{\neg,\cap,\cup\}}$ such that $A \equiv B$ and the condition **C1** holds

Observe that any formula A of language $\mathcal{L}_{\{\neg,\cap,\cup\}}$ is also a formula of the language $\mathcal{L}_{\{\neg,\cap,\cup,\Rightarrow\}}$ and of course $A \equiv A$ so the condition **C2** also holds

Example

The logical equivalences:

Definability of Conjunction in terms of implication and negation

$$(A \cap B) \equiv \neg(A \Rightarrow \neg B)$$

and **Definability of Implication** in terms of conjunction and negation

$$(A \Rightarrow B) \equiv \neg(A \cap \neg B)$$

and the **Substitution Theorem** prove that

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

Equivalence of Languages

Exercise

Prove that

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

Solution

The equivalence holds due to the **Substitution Theorem** and two following **Definability of Connectives** equivalences:

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

They transform recursively any formula from $\mathcal{L}_{\{\cap,\neg\}}$ into a formula of $\mathcal{L}_{\{\cup,\neg\}}$ and vice-versa, respectively

Logical Equivalences

Here are some more frequently used logical equivalences

Idempotent

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

Associativity

$$((A \cap B) \cap C) \equiv (A \cap (B \cap C))$$

$$((A \cup B) \cup C) \equiv (A \cup (B \cup C))$$

Commutativity

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

Logical Equivalences

Here are some more frequently used logical equivalences

Distributivity

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

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

De Morgan Laws

$$\neg(A \cup B) \equiv (\neg A \cap \neg B)$$

$$\neg(A \cap B) \equiv (\neg A \cup \neg B)$$

Negation of Implication

$$\neg(A \Rightarrow B) \equiv (A \cap \neg B)$$

Equivalence of Languages

Exercise

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 $\mathcal{L}_{\{\cup,\neg\}}$

Solution

$$\neg(\neg(\neg a \cap \neg b) \cap a)$$

$$\equiv \neg(\neg\neg(\neg\neg a \cup \neg\neg b) \cap a)$$

$$\equiv \neg((a \cup b) \cap a)$$

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

The formula B of $\mathcal{L}_{\{\cup,\neg\}}$ equivalent to A is

$$B = \neg(\neg(a \cup b) \cup \neg a)$$

Equivalence of Languages

Exercise

Prove by transformation, using proper logical equivalences that

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

Solution

$$\neg(A \Leftrightarrow B)$$

$$\equiv^{def} \neg((A \Rightarrow B) \cap (B \Rightarrow A))$$

$$\equiv^{de\ Morgan} (\neg(A \Rightarrow B) \cup \neg(B \Rightarrow A))$$

$$\equiv^{neg\ impl} ((A \cap \neg B) \cup (B \cap \neg A))$$

$$\equiv^{commut} ((A \cap \neg B) \cup (\neg A \cap B))$$

Exercise

Prove by transformation, using proper logical equivalences that

$$((B \cap \neg C) \Rightarrow (\neg A \cup B))$$

$$\equiv ((B \Rightarrow C) \cup (A \Rightarrow B))$$

Solution

$$((B \cap \neg C) \Rightarrow (\neg A \cup B))$$

$$\equiv^{impl} (\neg(B \cap \neg C) \cup (\neg A \cup B))$$

$$\equiv^{de\ Morgan} ((\neg B \cup \neg\neg C) \cup (\neg A \cup B))$$

$$\equiv^{neg} ((\neg B \cup C) \cup (\neg A \cup B))$$

$$\equiv^{impl} ((B \Rightarrow C) \cup (A \Rightarrow B))$$

Chapter 3
Propositional Semantics: Classical and Many Valued

Slides Set 6

PART 9 **Many Valued Semantics:** Łukasiewicz, Heyting, Kleene, and Bohvar

First Many Valued Logics

The study of many valued logics in general and 3-valued logics in particular has its beginning in the work of a Polish mathematician **Jan Leopold Łukasiewicz** in 1920

Łukasiewicz was the first to **define** a 3 - valued **semantics** for the language

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

of classical logic, and called it a **logic** for short

He left the problem of **finding** a proper axiomatic proof system for it **open**

First Many Valued Logics

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

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

Creating a Logic

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

A process of creating a **logic** (based on a given language) is **three fold:** we have to

define semantics,

create axiomatic proof system and

prove completeness theorem that establishes a **relationship** between semantics and proof system

First Many Valued Logics

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

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

We assume that the **language** of all **semantics** (logics) considered here except of Bochvar semantics is

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

3-Valued Semantics

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

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

The symbol $\perp$ is the most frequently used for different concepts of **unknown**

Many Valued Semantics

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

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

We will also use the name **logic** for them, instead saying each time " logic defined semantically", or "semantics for a given logic"

3 Valued Semantics Assumptions

We **assume** that the third logical value is **intermediate** between truth and falsity, i.e.

the set of logical values is **ordered** and we have the following

Assumption 1

$$F < \perp < T, \quad \text{and} \quad F < m < T$$

Assumption 2

We take T as designated value, i.e. T is the value that **defines** the notions of satisfiability and tautology

Many Valued Extensional Semantics

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

It consists of giving **definitions** of the following main components:

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

Łukasiewicz Semantics **L**

Motivation

Łukasiewicz developed his semantics (called logic) to deal with future **contingent** statements

Contingent statements are not just neither **true** nor **false** but are indeterminate in some metaphysical sense

It is not only that we do not know their truth value but rather that they do not possess one

L Semantics: Language

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

Step 1 The **language** is $\mathcal{L}_{\{\neg, \cap, \cup, \Rightarrow\}}$

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

The set $\mathcal{F}$ of **formulas** is defined in a standard way

Step 1 Connectives

We assumed: $F < \perp < T$ and we define the connectoves as follows

Negation $\neg$ is a **function**

$$\neg : \{T, \perp, F\} \longrightarrow \{T, \perp, F\}$$

such that $\neg \perp = \perp, \quad \neg T = F, \quad \neg F = T$

Conjunction $\cap$ 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

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

L Semantics: Connectives

Disjunction $\cup$ is a **function**

$$\cup : \{T, \perp, F\} \times \{T, \perp, F\} \longrightarrow \{T, \perp, F\}$$

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

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

Implication $\Rightarrow$ 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

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

L Connectives Truth Tables

Negation

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

Conjunction

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

L Connectives Truth Tables

Disjunction

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

Implication

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

L Semantics: Truth Assignment

Step 2 Truth assignment and its extension

Definition

A **truth assignment** is any function

$$v : VAR \longrightarrow \{F, \perp, T\}$$

Observe that the domain of truth assignment is the set of propositional variables, i.e. the truth assignment is defined only for **atomic formulas**

Truth Assignment Extension v^*

Definition

Given a truth assignment $v : VAR \longrightarrow \{T, \perp, F\}$

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

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

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

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

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

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

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

L Semantics: Satisfaction Relation

Step 3 Satisfaction, Model, Counter Model

Definition

Let $v : VAR \longrightarrow \{T, \perp F\}$

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

Notation: $v \models_L A$

Definition

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

Notation: $v \not\models_L A$

L Semantics: Model, Counter Model

Model

Any truth assignment $v : VAR \longrightarrow \{F, \perp, T\}$ such that

$$v \models_L A$$

is called a **L model** for A

Counter Model

Any v such that

$$v \not\models_L A$$

is called a **L counter model** for the formula A

Step 4 Tautology

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

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

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

Notation

$$\models_L A$$

L Tautologies

We denote the set of all **L tautologies** by

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

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

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

It means are theirs sets of tautologies different?

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

We know that

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

We will show that

$$\not\models_L (\neg a \cup a)$$

Classical and **L** Tautologies

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

Take a truth assignment v such that

$$v(a) = \perp$$

Evaluate

$v^*(\neg a \cup a) = v^*(\neg a) \cup v^*(a) = \neg v(a) \cup v(a)$

$= \neg \perp \cup \perp = \perp \cup \perp = \perp$

This proves that v is a **counter-model** for $(\neg a \cup a)$, i.e.

$$\not\models_L \ (\neg a \cup a)$$

and we proved

$$\mathbf{LT} \neq \mathbf{T}$$

Classical and **L** Tautologies

Q2 Do the **L** and classical **logics** have something more in common besides the same language?

YES, they also share some tautologies

Q3 Is there relationship (if any) between their sets of **tautologies** **LT** and **T**?

YES, their sets of **tautologies** **LT** and **T** do have an interesting relationship

Classical and **L** Tautologies

Let's restrict the functions defining **L** connectives (Truth Tables for **L** connectives) to the values T and F

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

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

We have hence **proved** that

$$\mathbf{LT} \subset \mathbf{T}$$

Exercise

Exercise

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

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

under **L** semantics **to evaluate**

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

Use **shorthand** notation.

Exercise

Solution

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

c1: $(a \cap b) =\perp$ and $\neg b = F$

c12: $(a \cap b) = T$ and $\neg b =\perp$

Consider **c1**

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

Hence $(a \cap T) =\perp$ if and only if $a =\perp$

We get that v is such that $v(a) =\perp$ and $v(b) = T$

Exercise

We got from analyzing case **c1** that v is such that $v(a) = \perp$ and $v(b) = T$

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

Consider **c2**

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

Hence v from case **c1** is the only one and

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

Łukasiewicz Life, Works and Logics

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

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

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

http://plato.stanford.edu/entries/lukasiewicz/index.html

Heyting Semantics **H**

Motivation and History

We discuss here the Heyting semantics **H** because of its connection with intuitionistic logic

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

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

Pseudo-Boolean algebras are often called Heyting algebras

Motivation and History

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

Heyting provided provided its **first axiomatization** which everybody accepted

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

The **pseudo boolean** algebras are **also** called Heyting algebras in his honor and so is our semantics **H**

Motivation and History

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

We prove that the operations defined by **H** connectives form a 3-element **pseudo boolean** algebra

Hence, if *A* is an intuitionistic tautology, it is also a tautology under the 3- valued Heyting semantics

If A **is not** a 3- valued Heyting tautology, then it **is not** an intuitionistic tautology

It means that the 3-valued Heyting semantics is a good candidate for a **counter model** for the formulas that **might not** be intuitionistic tautologies

H Logic and Intuitionistic Logic

Denote by **IT**, **HT** the sets of all **tautologies** of the intuitionistic logic and Heyting 3-valued logic (semantics), respectively .

We have that

$$\mathbf{IT} \subset \mathbf{HT}$$

We conclude that for any formula A,

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

It means that if we show that a formula A has an **H counter model**, then we have proved that A it **is not** an intuitionistic tautology

Kripke Models

The other type of **semantics** for the intuitionistic logic were defined by Kripke in 1964

They are called **Kripke models**

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

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

That includes **semantics** for various modal logics and new logics developed and being developed by computer scientists

H Semantics

Language

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

Connectives

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

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

where $F < \perp < T$

H Semantics

Implication

$$\Rightarrow: \quad \{T, \bot, F\} \times \{T, \bot, F\} \longrightarrow \{T, \bot, F\}$$

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

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

Negation

$$\neg x = x \Rightarrow F$$

H Truth Tables

Implication

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

Negation

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

Sets of Tautologies Relationships

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

Relationships

$$\mathbf{HT} \neq \mathbf{T} \neq \mathbf{LT}$$

$$\mathbf{HT} \subset \mathbf{T}$$

Proof of $\mathbf{HT} \neq \mathbf{T}$

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

$$\models (\neg a \cup a) \text{ and } \not\models_{\mathbf{H}} (\neg a \cup a)$$

Sets of Tautologies Relationships

Proof of **HT $\neq$ LT**

Take a truth assignment v such that

$$v(a) = v(b) = \perp$$

We verify that

$$\not\models_{\mathbf{H}} (\neg(a \cap b) \Rightarrow (\neg a \cup \neg b))$$

and

$$\models_{\mathbf{L}} (\neg(a \cap b) \Rightarrow (\neg a \cup \neg b))$$

Sets of Tautologies Relationships

Proof of **HT** $\subset$ **T**

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

If $v^*(A) = T$ for all $v: VAR \longrightarrow \{F, \perp, T\}$,

then $v^*(A) = T$ for all $v: VAR \longrightarrow \{F, T\}$

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

LT $\subset$ **T** and **HT** $\subset$ **T**

Kleene Semantics **K**

Motivation

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

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

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

Kleene Semantics **K**

For **example** imagine a detective trying to solve a **murder**

He may **conjecture** that Jones killed the **victim**

He cannot, at present, **assign** a truth value T or F to his conjecture, so we **assign** the value $\perp$

But it is certainly either true or false and hence $\perp$ represents our **ignorance** rather then total **unknown**

Kleene Semantics **K**

Language

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

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

Connectives

We assume, as before, that $F < \perp < T$

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

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

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

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

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

K Semantics: Connectives

K Implication

Kleene's implication **differ** from **L** and **H** semantics

The **K** implication is defined by the same formula as the classical, i.e. for any $(x, y) \in \{T, \perp, F\} \times \{T, \perp, F\}$

$$x \Rightarrow y = \neg x \cup y$$

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

K implication table is

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

K Semantics: Tautologies

Set of all **K** tautologies is

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

Relationship between **Ł, H, K**, and classical semantics is

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

Proof Obviously $\models_L (a \Rightarrow a)$ and $\models (a \Rightarrow a)$ We take v such that $v(a) = \perp$ and 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

$$\mathbf{LT} \neq \mathbf{KT} \quad \text{and} \quad \mathbf{LT} \neq \mathbf{KT}$$

K Tautologies

The third property

$$\mathbf{KT} \subset \mathbf{T}$$

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

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

$$\mathbf{LT} \subset \mathbf{T}, \quad \mathbf{HT} \subset \mathbf{T}, \quad \text{and} \quad \mathbf{K} \subset \mathbf{T}$$

L, H, K Decidability

Verification and Decidability

The following theorem justifies the correctness of the **truth table** method of **tautology verification** for for **L, H, K** semantics

Theorem 1

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

$$\models_{\mathbf{M}} A \quad \text{if and only if} \quad v_A \models_{\mathbf{M}} A$$

$$\text{for all} \quad v_A : VAR_A \longrightarrow \{T, \perp, F\}$$

We also say that

$\models_{\mathbf{M}} A$ if and only if all v_A are **restricted M** models for A, and $\mathbf{M} \in \{\mathbf{L}, \mathbf{H}, \mathbf{K}\}$

L, H, K Decidability

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

Theorem 2

For any formula A of $\mathcal{L}_{\{\neg,\Rightarrow,\cup,\cap\}}$, one has to **examine** at most 3^{VAR_A} truth assignments $v_A : VAR_A \longrightarrow \{F, \perp, T\}$ in order to **decide** whether

$$\models_{\mathbf{M}} A \quad \text{or} \quad \not\models_{\mathbf{M}} A$$

i.e. the notion of **M** tautology is **decidable**

for any semantics $\mathbf{M} \in \{\mathbf{L}, \mathbf{H}, \mathbf{K}\}$

Proofs of **Theorems 1, 2** are carried in the same way as in case of classical semantics and are left as an exercise

K Tautologies Revisited

Exercise

We know that formulas

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

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)

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

K Tautologies Revisited

$v^*((a \Rightarrow (a \cup b))) = \perp \Rightarrow (\perp \cup \perp) = \perp \Rightarrow \perp = \perp$ and

$v^*((a \Rightarrow (b \Rightarrow a))) = (\perp \Rightarrow (\perp \Rightarrow \perp) = \perp \Rightarrow \perp = \perp$

This proves that any v such that

$$v(a) = v(b) = \perp$$

is a **counter model** for all of them

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

$$v(a) = \perp \quad \text{for all} \quad a \in VAR$$

is a **counter model** for **any formula** A of $\mathcal{L}_{\{\neg, \Rightarrow, \cup, \cap\}}$

K Tautologies Revisited

We proved the following

Theorem

For any formula A of $\mathcal{L}_{\{\neg,\Rightarrow,\cup,\cap\}}$, $\not\models_{\mathbf{K}} A$

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

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

Observe that the **Theorem** does not invalidate relationships

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

between **Ł**, **H**, **K**, and classical semantics

They become now perfectly true statements

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

K Tautologies Revisited

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

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

Definition

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

We say that the semantics **M** is **well defined** if and only if its set **MT** of all tautologies is non empty, i.e. when

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

K Tautologies Revisited

The semantics **K** is an example of a **correctly** and **carefully** defined semantics that **is not** well defined in terms of the above definition

Obviously the semantics **L** and **H** are well defined

We write is as a following separate fact

K Tautologies Revisited

Fact

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

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

This is the main reason, beside its historical value, why it is included here

Bochvar Semantics **B**

Motivation

Consider a **semantic paradox** given by a sentence:

this sentence is false.

If it is true it must be false,

if it is false it must be true.

According to Bochvar, such sentences are neither true of false but rather **paradoxical** or **meaningless**

B Semantics

Bochvar's semantics follows the principle that the third logical value, denoted now by m (for miningless) is in some sense "infectious";

if one component of the formula is **assigned** the value m then the formula is also **assigned** the value m

Bochvar also adds an one **assertion** operator S that **asserts** the logical value of T and F , i.e.

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

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

$$Sm = F$$

B Semantics: Language

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

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

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

Observe that directly from the definition we have that

$$\mathcal{F} \subset \mathcal{F}_{\mathcal{B}}$$

The formula SA **reads** "assert A"

B Semantics: Connectives

Negation

$\neg$	F	m	T
	T	m	F

Conjunction

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

B Semantics: Connectives

Disjunction

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

Implication

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

B Semantics: Connectives, Tautology

Assertion

S	F	m	T
	F	F	T

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

In particular the set of all **B** **tautologies** is

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

B Semantics: Tautology

We get by easy evaluation that

$$\models_{\mathbf{B}} (Sa \cup \neg Sa)$$

This proves that **BT** $\neq \emptyset$, what means that

B semantics is **well defined**

B Semantics: Tautology

Observe that **not all** formulas containing the connective S are **B** **tautologies**, for example we have that

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

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

$Sm \cup S\neg m = F \cup Sm = F \cup F = F$

B Semantics: Tautology

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

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

$$\mathbf{T} \cap \mathbf{BT} = \emptyset$$

Observation

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

Chapter 3
Propositional Semantics: Classical and Many Valued

Slides Set 7

PART 10 **M** Tautologies, **M** Consistency, and **M** Equivalence of Languages

M Tautologies Verification Methods

The classical **truth tables** verification method and classical **decidability theorem** hold in a proper form in all of **L. H, K** and **B** semantics

We didn't discuss other classical tautologies verification methods of **substitution** and **generalization**

We do it now in a **general** and **unifying** way for a special case of an extensional semantics **M**

Namely, we **assume** now that the set LV of logical values of the semantics **M** is **finite**

M Tautologies Verification Methods

We introduce, as we did in classical and other cases, a notion of a **restricted model** v_A and prove the following theorems

Truth TablesTheorem

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

$\models_{\mathbf{M}} A$ if and only if all v_A are restricted models for A

M Decidability Theorem

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

$$\models_{\mathbf{M}} A \quad \text{or} \quad \not\models_{\mathbf{M}} A$$

i.e. the notion of **M** tautology is **decidable**

M Truth Table Method

A tautology verification method, called a **M** truth table method consists of examination, for any formula A, **all** possible **M** truth assignments **restricted** to A

By **M Decidability Theorem** we have to perform at most $|LV|^{|VAR_A|}$ steps

If we **find** a restricted 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

Example

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

Example

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

$$\models_{\mathbf{L}} (\neg\neg a \Rightarrow a)$$

M Proof by Contradiction Method

M Proof by Contradiction Method

In this method, in order to prove that $\models_{\mathbf{M}} A$ we **assume** that $\not\models_{\mathbf{M}} A$

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** **M** tautology

M Proof by Contradiction Method

Observe that correctness of the **M Proof by Contradiction** method is based on the classical reasoning

Its correctness, in turn, is based on the **Reductio ad Absurdum** classical tautology

$$((\neg A \Rightarrow (B \cap \neg B)) \Rightarrow A)$$

The **contradiction** to be obtained **depends** on 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 following theorem establishes its correctness and its proof is a straightforward modification of the classical one

Theorem

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

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

M Generalization Method

In this method we represent, if it is possible, a given formula A as a particular instance of some **simpler** and more **general formula** B

We then use other verification methods to examine the simpler formula B thus obtained

Remark

Observe that Proof by Contradiction, Substitution and Generalization Methods are valid for any extensional semantics **M** while the **M** Truth Table Method is valid only for semantics **M** with **finite** the set LV of logical values

M Substitution Method

Example

In order to prove

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

we **observe** that that our formula is a particular case of a more general formula

$$(\neg\neg A \Rightarrow A)$$

for $A = (\neg((a \cap \neg b) \Rightarrow ((c \Rightarrow (\neg f \cup d)) \cup e)) \Rightarrow ((a \cap \neg b) \cap (\neg(c \Rightarrow (\neg f \cup d)) \cap \neg e)))$

As the next step we observe (or easily prove) that

$$\models_{\mathbf{L}} (\neg\neg A \Rightarrow A)$$

M Consistency

One of the most important **notion** in mathematics and hence even in propositional logic is the notion of **consistency** and **inconsistency**

We **formulate** them now for the general case of extensional semantics **M** and examine them particular cases of **L** and **H** semantics

Definition

A truth truth assignment $v : VAR \longrightarrow LV$ is a **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 $v \models_{\mathbf{M}} \mathcal{G}$

M Consistency

Definition

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

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

Observe that the definition of inconsistency can be stated as follows

Definition

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

M Consistency Exercise

Example

The set

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

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

Proof

Consider a truth assignment $v : VAR \longrightarrow \{T, \perp, F\}$. By the definition of **M consistency** v must be such that

$v^*(((a \cap b) \Rightarrow b)) = T,\ v^*((a \cup b) = T)$, and $v^*(\neg a) = T$

We want to prove that such v exists

Observe that $((a \cap b) \Rightarrow b)$ is classical tautology, so let's try to find $v : VAR \longrightarrow \{T, F\}$ such that
$v^*((a \cup b)) = T,\ v^*(\neg a) = T$

This holds when $v(a) = F$ and hence $F \cup v(b) = T$

This gives us $v(a) = F$ and $v(b) = T$

M Consistency Exercise

We proved that the connectives of **L**, **H**, and **K** semantics when restricted to the values T and F become **classical** connectives

Hence 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

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

M Consistency Exercise

Exercise

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 **sub language** $\mathcal{L}_{\{\neg, \Rightarrow, \cup, \cap\}}$ of the language $\mathcal{L}_{\mathbf{B}}$

M Consistency Exercise

Let's take, for example a set

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

$\mathcal{G}$ is **infinite** since the set VAR of propositional variables is infinite

Consider any of the truth assignments

$$v : VAR \longrightarrow \{F, m, T\} \quad \text{or} \quad v : VAR \longrightarrow \{F, \perp, T\}$$

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

We have that

$$v^*(a \cup b) = v(a) \cup v(b) = T \cup T = T$$

in all semantics **L**, **H**, **K**, **B**

This proves that $\mathcal{G}$ is **L**, **H**, **K** and **B consistent**

M Consistency Exercise

Exercise

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

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

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

M Consistency Exercise

Consider

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

and any truth assignment

$$v : VAR \longrightarrow \{F, m, T\}$$

such that $v(a) = T, v(b) = F$ (short hand notation)

We evaluate

$(ST \cup S\neg T) = T \cup T = T,\ \ (T \Rightarrow \neg F) = T,\ \ S\neg(T \Rightarrow F) = S\neg F = T,\ \ (F \Rightarrow ST) = F \Rightarrow T = T$

This proves that v is a **B** model for $\mathcal{G}_1$, and $\mathcal{G}_1$ is **consistent**

M Consistency Exercise

Consider now

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

Assume that there is

$$v : VAR \ \longrightarrow \ \{F, m, T\}$$

such that $v \models_{\mathbf{B}} \mathcal{G}_2$

In particular $v^*(Sa) = T$

By definition of **B** connectives this is possible if and only if $v(a) = T$

Then $v^*(S\neg a) = SF = F$

This contradicts the assumption $v \models_{\mathbf{B}} \mathcal{G}_2$

Hence $\mathcal{G}_2$ is **B** inconsistent

M Independence

Definition

Given a language $\mathcal{L}_{\{\neg, \Rightarrow, \cup, \cap\}}$

A formula $A \in \mathcal{F}$ is called **M independent** from a set $\mathcal{G} \subseteq \mathcal{F}$ if and only if the sets

$$\mathcal{G} \cup \{A\} \quad \text{and} \quad \mathcal{G} \cup \{\neg A\}$$

are both **M consistent**

I.e. when there are truth assignments v_1, v_2 such that

$$v_1 \models_{\mathbf{M}} \mathcal{G} \cup \{A\} \quad \text{and} \quad v_2 \models_{\mathbf{M}} \mathcal{G} \cup \{\neg A\}.$$

M Independence Exercises

Given a set

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

1. Find a formula A that is **L independent** from a set $\mathcal{G}$

2. Find a formula A that is **H independent** from a set $\mathcal{G}$

3. Find an infinite number of that are **L independent** from a set $\mathcal{G}$

4. Find an infinite number of that are **H independent** from a set $\mathcal{G}$

M Logical Equivalence and **M** Equivalence of Languages

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 notions of **logical equivalence** and **equivalence of languages** to the semantics **M**

M Logical Equivalence

Definition

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

$$v^*(A) = v^*(B) \quad \text{for all } v : VAR \rightarrow LV$$

We write

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

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

M Logical Equivalence

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

M Logical Equivalence

Exercise

The classical logical equivalence

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

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

We have to show $v^*((a \cup b)) \neq v^*((\neg a \Rightarrow b))$

for some $v : VAR \longrightarrow \{F, \perp, T\}$

Observe that $v^*((a \cup b)) = v^*((\neg a \Rightarrow b))$ for all $v : VAR \longrightarrow \{F, T\}$

So we have to check only truth assignments that involve $\perp$

Let v be 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

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

$\mathbf{M}$ Equivalence of Languages

We extend now, in a natural way, the classical notion equivalence of languages

Definition

Given two languages

$$\mathcal{L}_1 = \mathcal{L}_{CON_1} \quad \text{and} \quad \mathcal{L}_2 = \mathcal{L}_{CON_2} \quad \text{for} \quad CON_1 \neq CON_2$$

We say that $\mathcal{L}_1$ and $\mathcal{L}_2$ 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

Exercise

Prove that

$$\mathcal{L}_{\{\neg,\Rightarrow\}} \equiv_{\mathbf{L}} \mathcal{L}_{\{\neg,\Rightarrow,\cup\}}$$

Solution

Condition **C1** holds because any formula of language $\mathcal{L}_{\{\neg,\Rightarrow\}}$ is also a formula of $\mathcal{L}_{\{\neg,\Rightarrow,\cup\}}$

Condition **C2** holds because the equivalence

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