cse541
LOGIC for COMPUTER SCIENCE

Professor Anita Wasilewska

LECTURE 3d

CHAPTER 3
Classical Tautologies and Logical Equivalences

PART 1: Classical Tautologies

PART2: Classical Logical Equivalence of Formulas

PART3: Classical Logical Equivalence of Languages

PART 4: Semantics **M** Logical Equivalence of Formulas

Semantics **M** Logical Equivalence Languages

CHAPTER 3
Classical Tautologies and Logical Equivalences

We present and **discuss** here a set of most widely used classical tautologies and logical equivalences

We introduce a notion of equivalence of propositional languages under classical and under other semantics

We also discuss the relationship between definability of connectives the equivalences of languages in classical and non-classical semantics

Classical Tautologies

PART 1: Classical Tautologies

Classical Tautologies

We assume that **all formulas** considered here belong to the language

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

Here is a list of some of the most known classical **notions** and **tautologies**

Modus Ponens known to the Stoics (3rd century B.C)

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

Detachment

$$\models ((A \cap (A \Leftrightarrow B)) \Rightarrow B)$$

$$\models ((B \cap (A \Leftrightarrow B)) \Rightarrow A)$$

Stoics, 3rd century B.C.

Hypothetical Syllogism

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

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

$$\models ((B \Rightarrow C) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))).$$

Modus Tollendo Ponens

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

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

12 to 19 Century

Duns Scotus 12/13 century

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

Clavius 16th century

$$\models ((\neg A \Rightarrow A) \Rightarrow A)$$

Frege 1879

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

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

Frege gave the the first formulation of the classical propositional logic as a formalized axiomatic system

CLASSICAL TAUTOLOGIES

YOU HAVE A VERY EXTENSIVE LIST OF CLASSICAL TAUTOLOGIES in CHAPTER 2 and in CHAPTER 3

Read them, memorize and use them to solve Hmk Problems listed in the BOOK and in published tests and quizzes

We will use them freely in the future Chapters assuming that you remember them

PART 2: Logical Equivalences

Logical Equivalence Definition

Logical equivalence:

For any formulas A, B, we **say** that are logically equivalent if and only if they always have the same logical value

Notation: we write symbolically $A \equiv B$ to denote that A, B are logically equivalent

Symbolic Definition

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

Logical Equivalence Property

The following property follows directly from the definition

Property

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

Remember

$\equiv$ **is not** a logical connective

$\equiv$ is just a metalanguage **symbol** for **saying** that the formulas A, B are logically equivalent

Some of Logical Equivalence Laws

Laws of contraposition

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

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

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

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

Law of Double Negation

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

Exercise: Prove validity of all of them

CLASSICAL LOGICAL EQUIVALENCES

YOU HAVE A VERY EXTENSIVE LIST OF CLASSICAL LOGICAL EQUIVALENCES in CHAPTER 3

Read them, memorize them and use to solve Hmk Problems listed in the BOOK and problems on your TESTS

We will use them freely in the future Chapters assuming that you remember them

Use of Logical Equivalence

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

Use of Logical Equivalence

Example

We easily obtain **new** Law of Contraposition from **the one** we already **have** and from already known Law of Double Negation as follows

$(\neg A \Rightarrow B) \equiv (\neg B \Rightarrow \neg\neg A) \equiv (\neg B \Rightarrow A)$, i.e. we proved that

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

$(A \Rightarrow \neg B) \equiv (\neg\neg B \Rightarrow \neg A) \equiv (B \Rightarrow \neg A)$, i.e. we proved that

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

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

Substitution Theorem

Let B_1 be obtained from A_1 by **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.

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

Proof in the book - but write it as an exercise- and then check with the book

Example 1

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

and let $C = \neg\neg C$

We get

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

By **Double Negation** Law

$$\neg\neg C \equiv C$$

So we get by **Substitution Theorem** that

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

Example 2

Example 2

We want to transform any formula with implication into a **logically equivalent** formula without implication

We use in this type of problems one of the **Definability of Connectives** Equivalences that concerns the implication, for example we use

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

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

Example 2

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

PART 3: Definability of Connectives and Equivalences
Equivalence of Languages

Definability of Connectives Equivalences

Chapter 3 contains a large set of **logical equivalences**, or corresponding **tautologies** that deal with the definability of connectives in classical semantics

Remember they the **logical equivalences** corresponding to the definability of connectives property is very strongly connected with the **classical semantics**

We leave it as an excellent **EXERCISE** to verify which of them (in any) holds in which of our different non-classical semantics

Definability of Connectives Equivalences

Definability of Implication in terms of negation and disjunction **equivalence**

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

is defined by a **a classical tautology**

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

We use the notion of **logical equivalence** instead of the **tautology** notion, as it makes the manipulation of formulas via **Substitution Theorem** much easier

Definability of Connectives Equivalences

Here is the

Definability of Implication in terms of negation and disjunction **equivalence**

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

The **proof** of this **logical equivalence,** and hence the corresponding **tautology** follows directly from **definability of implication connective** i n terms of disjunction and negation connectives already proved for classical semantics, hence the **same name**

Proofs of Definability of Connectives Equivalences

We present here the **proof** of **Definability of Implication** in terms of negation and

disjunction **equivalence**

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

as an **example** of a **pattern** to follow while conducting

the **proofs** of Definability of Connectives Equivalences for

other connectives

Proofs of Definability of Connectives Equivalences

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

By definition of logical equivalence we have that

$(A \Rightarrow B) \equiv (\neg A \cup B)$ holds if and only if

$v^*(A \Rightarrow B) = v^*(\neg A \cup B)$ for all $v : VAR \rightarrow \{T, F\}$

Observe that, by definition of v^* we have that

$v^*(A \Rightarrow B) = v^*(A) \Rightarrow v^*(B) = \neg v^*(A) \cup v^*(B)$ where

$v^*(A), v^*(B) \in \{T, F\}$ and $\Rightarrow, \neg, \cup$ are functions defined by the classical semantics

We have proved (definability of classical connectives) that for any $x, y \in \{T, F\}$ we have that $x \Rightarrow y = \neg x \cup y$

hence $v^*(A \Rightarrow B) = v^*(\neg A \cup B)$ for all $v : VAR \rightarrow \{T, F\}$

what **ends** the proof

Definability of Connectives Equivalences

Definability of Implication **equivalence** allows us, by the force of **Substitution Theorem** to replace any formula of the form $(A \Rightarrow B)$ placed anywhere in **another** 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 contain implication but contains negation and disjunction only

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 colorred logically equivalent to the formulas of the language $\mathcal{L}_1$

We write it as the following **condition C1**

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

Example 2

Let now A be a formula

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

We can use here the **definability of implication** equivalence

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

to **eliminate** disjunction as follows

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

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

Example 2

Observe that we **can't always** use the equivalence

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

to **eliminate** disjunction

For example, **we can't** use it for a formula

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

Nevertheless we **can eliminate** disjunction from it,

but we need a **different** equivalence

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 need the following logical equivalence

Definability of Disjunction in terms of negation and implication

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

Example 3

Consider a formula

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

We use the equivalence

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

to transform $(A \cup B) \cap \neg A)$ into its **logically equivalent** form **not** containing $\cup$ but containing $\Rightarrow$ as follows.

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

Equivalence of Languages

The equivalence

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

allows us to **transform** a language $\mathcal{L}_2 = \mathcal{L}_{\{\neg,\, \cap,\, \cup\}}$ into a language $\mathcal{L}_1 = \mathcal{L}_{\{\neg, \cap, \Rightarrow\}}$ with all their formulas being **logically equivalent**

Equivalence of Languages

We write this property as the following condition **C2** similar to the already adopted condition

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 $C \equiv D$

We say that the languages $\mathcal{L}_1$ and $\mathcal{L}_2$ for which the conditions **C1, C2** hold are logically equivalent and we adopt the following definition

Equivalence of Languages Definition

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 $C \equiv D$

Example 4

To prove the logical equivalence of the languages

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

we need **two definability equivalences**:

implication in terms of disjunction and negation

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

and disjunction in terms of implication negation,

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

and the **Substitution Theorem**

Example 5

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** in terms of disjunction and negation 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 6

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

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

Exercise 1

Exercise 1 Prove that

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

Solution

Equivalence holds due to the **Substitution Theorem** and two 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

Exercise 2

Use the **Definability of Conjunction** in terms of disjunction and negation equivalence to 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

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

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

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

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

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

Exercise 3

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 4

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

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

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

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

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

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

PART 4

Semantics **M** Logical Equivalence of Formulas

Semantics **M** Logical Equivalence Languages

M - Logical Equivalence of Formulas

Given an extensional semantics **M** defined for a propositional language $\mathcal{L}_{CON}$ and let $V \neq \emptyset$ be its set set of logical values

We say that any two formulas A, B of the language $\mathcal{L}_{CON}$ are **M -logically equivalent** if and only if they always have the same logical value assigned by the semantics **M**

Notation

we write symbolically $A \equiv_{\mathbf{M}} B$ to denote that the formulas A, B are **M -logically equivalent**

M - Logical Equivalence of Formulas

Definition

For any formulas A, B,

$A \equiv_{\mathbf{M}} B$ if amd only if $v^*(A) = v^*(B)$ for all $v : VAR \to V$

Remember

$\equiv_{\mathbf{M}}$ **is not** a logical connective

$\equiv_{\mathbf{M}}$ is just a metalanguage **symbol** for saying

" Formulas A, B are **M** -**logically equivalent** "

M - Logical Equivalence of Languages

Given two languages: $\mathcal{L}_1 = \mathcal{L}_{CON_1}$ and $\mathcal{L}_2 = \mathcal{L}_{CON_2}$, for $CON_1 \neq CON_2$

We say that they are **M- logically equivalent**, i.e.

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

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 5

Prove that in classical semantics

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

Solution

Observe that the condition **C1** holds because any formula of $\mathcal{L}_{\{\neg,\Rightarrow\}}$ is also a formula of $\mathcal{L}_{\{\neg,\Rightarrow,\cup\}}$

Condition **C2** holds due to the following definability of connectives equivalence

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

and the **Substitution Theorem**

Exercise 6

Prove that the equivalence defining $\cup$ in terms of negation and implication in classical logic **does not hold** under **L** semantics, i.e. that

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

but nevertheless

$$\mathcal{L}_{\{\neg,\Rightarrow\}} \equiv_{\mathbf{L}} \mathcal{L}_{\{\neg,\Rightarrow,\cup\}}$$

Exercise 6

Observe that the equivalence

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

defining $\cup$ in terms of $\neg$ and $\Rightarrow$ seems a valuable candidate for **L** semantics as definability as the definition of all **L** connectives restricted to the logical values T, F is the same as in the classical case

Unfortunately it is **not a good one** for **L** semantics, as any v such that $v^*(A) = v^*(B) = \perp$ is **counter- model**

But it **does not prove** that a different definability equivalence does not **exist**!

Exercise 6

We prove

$$\mathcal{L}_{\{\neg,\Rightarrow\}} \equiv_{\mathbf{L}} \mathcal{L}_{\{\neg,\Rightarrow,\cup\}}$$

as follows

Condition **C2** holds because the definability of connectives equivalence

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

Check it by verification as an exercise

C1 holds because any formula of $\mathcal{L}_{\{\neg,\Rightarrow\}}$ is a formula of $\mathcal{L}_{\{\neg,\Rightarrow,\cup\}}$

Observe that the equivalence $(A \cup B) \equiv (A \Rightarrow B) \Rightarrow B)$

provides also an alternative proof of **C2** in classical case