cse541
LOGIC FOR COMPUTER SCIENCE

Professor Anita Wasilewska

Spring 2015

LECTURE 6

CHAPTER 6

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 6
Classical Tautologies and Logical Equivalences

PART 1: Classical Tautologies

Classical Tautologies

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

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

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

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

Sufficient and Necessary

Sufficient: Given an implication $(A \Rightarrow B)$,

A is called a sufficient condition for B to hold.

Necessary : Given an implication $(A \Rightarrow B)$,

B is called a necessary condition for A to hold.

Implication Names

Simple $(A \Rightarrow B)$ is called a simple implication.

Converse $(B \Rightarrow A)$ is called a converse implication to $(A \Rightarrow B)$.

Opposite $(\neg B \Rightarrow \neg A)$ is called an opposite implication to $(A \Rightarrow B)$.

Contrary $(\neg A \Rightarrow \neg B)$ is called a contrary implication to $(A \Rightarrow B)$.

Laws of contraposition

Here they are:

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

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

The laws of contraposition make it possible to replace, in any deductive argument, a sentence of the form

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

and conversely.

Necessary and sufficient

We read $(A \Leftrightarrow B)$ as

B is necessary and sufficient for A

because of the following tautology

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

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

Apagogic Proofs

Apagogic Proofs: means proofs by reductio ad absurdum

Reductio ad absurdum: to prove A to be true,

we assume $\neg A$

If we get a contradiction, it means that we have proved A to be true

Correctness of this reasoning is guarantee by the following tautology

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

CLASSICAL TAUTOLOGIES

YOU HAVE A VERY EXTENSIVE LIST OF CLASSICAL TAUTOLOGIES in CHAPTER 6

Read them, memorize and use 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 they always have the same logical value

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

Symbolic Definition

$$A \equiv B \quad \text{iff} \quad v^*(A) = v^*(B) \quad \text{for all} \quad v : VAR \to \{T, F\}$$

Directly from the definition we have that

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

Remember that $\equiv$ is **not a logical connective**,

it is just a metalanguage symbol for saying " 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 6

Read them, memorize and use 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

Use of Logical Equivalence

Logical equivalence is a very useful notion when we want to obtain new formulas, or tautologies, if needed, on a base of some already known in a way that guarantee preservation of the logical value of the initial formula.
For example, we easily obtain new **Laws of Contraposition** from the one we have and the **Law of Double Negation** as follows

$$(\neg A \Rightarrow B) \equiv (\neg B \Rightarrow \neg\neg A) \equiv (\neg B \Rightarrow A), \quad \text{i.e.}$$

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

$$(A \Rightarrow \neg B) \equiv (\neg\neg B \Rightarrow \neg A) \equiv (B \Rightarrow \neg A), \quad \text{i.e.}$$

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

Substitution Theorem

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

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

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 **Theorem** that

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

Example 2

Example 2: 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** equivalence that concerns the implication:

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

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

Example 2

We transform (via our Theorem) a formula

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

into its **logically equivalent** form not containing $\Rightarrow$ as follows

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

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

We get

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

PART 3: Definability of Connectives

Equivalence of Languages

Definability of Connectives

Chapter 6 contains a large set of equivalences, or corresponding tautologies that deal with the definability of connectives in classical semantics.

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

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

Definability of Connectives

For example, a classical tautology

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

The **proof** of this tautology follows directly from **definability of implication** in terms of disjunction and negation in classical semantics
We state it in a form of a logical equivalence and call it by the same name as in semantic case, i.e. we have the following

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

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

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

Definability of Connectives

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 we 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

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**

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 use the **definability of implication** equivalence
$(A \Rightarrow B) \equiv (\neg A \cup B)$ to **eliminate disjunction** as follows

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

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

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 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 formula 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: 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 $C \equiv D$

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 $C \equiv D$

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** 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

1. Prove that

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

Solution

True due to the **Substitution Theorem** and two definability of connectives equivalences:

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

Exercise 1

2. Transform a formula $A = \neg(\neg(\neg a \cap \neg b) \cap a)$ of $\mathcal{L}_{\{\cap,\neg\}}$ into a logically equivalent formula B of $\mathcal{L}_{\{\cup,\neg\}}$

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

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

Exercise 2

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 2

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

Definition

For any formulas A, B, 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**

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

Symbolic Definition

$$A \equiv_{\mathbf{M}} B \quad \text{iff} \quad v^*(A) = v^*(B) \quad \text{for all} \quad v : VAR \rightarrow V$$

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**"

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 $C \equiv_{\mathbf{M}} D$