Chapter 1

Introduction to Logic Programming

Contents

Course Information

Lecture	Saturday 10:10 - 11:50
Office Hours	Saturday 12-2 Tuesday 11-1
Grading	60% Final exam, 40% (Midterm+ Practical + Oral exam)
Assignments	5-8 assignment across course
Useful references	➢ Introduction to Logic Second Edition, Michael Genesereth ➢ Programming in Prolog. Fourth edition, Clocksin and Mellish (free e-book)
Course web Site	➢ https://www.coursesites.com/s/_log_CS ➢ course access code: cs2017
Useful Link	➢ http://logic.stanford.edu/intrologic

Course Site

COURSEsites

Tweet G+ Share 4K

Logic_programming_CS

This course is designed \tfor undergraduate\"s computer science students. The course formalizes students with the logic programming paradigm and its programming techniques.

Self-Enroll
in this course

Login
as an enrolled student

Instructor(s):

E F

Course Site

COURSEsites

Tweet G+ Share 4K

Logic_programming_CS

This course is designed \tfor undergraduate\"s computer science students. The course formalizes students with the logic programming paradigm and its programming techniques.

Self-Enroll
in this course

Please enter the course access code:

Save and Continue

Login
as an enrolled student

Instructor(s):

E F

Course Site

COURSEsites

Logic_programming_CS

Welcome!

This course is designed \tfor undergraduate\"s computer science students. The course formalizes students with the logic programming paradigm and its programming techniques.

Self-Enroll

To complete the self-enrollment process for this course, please choose one of the options below:

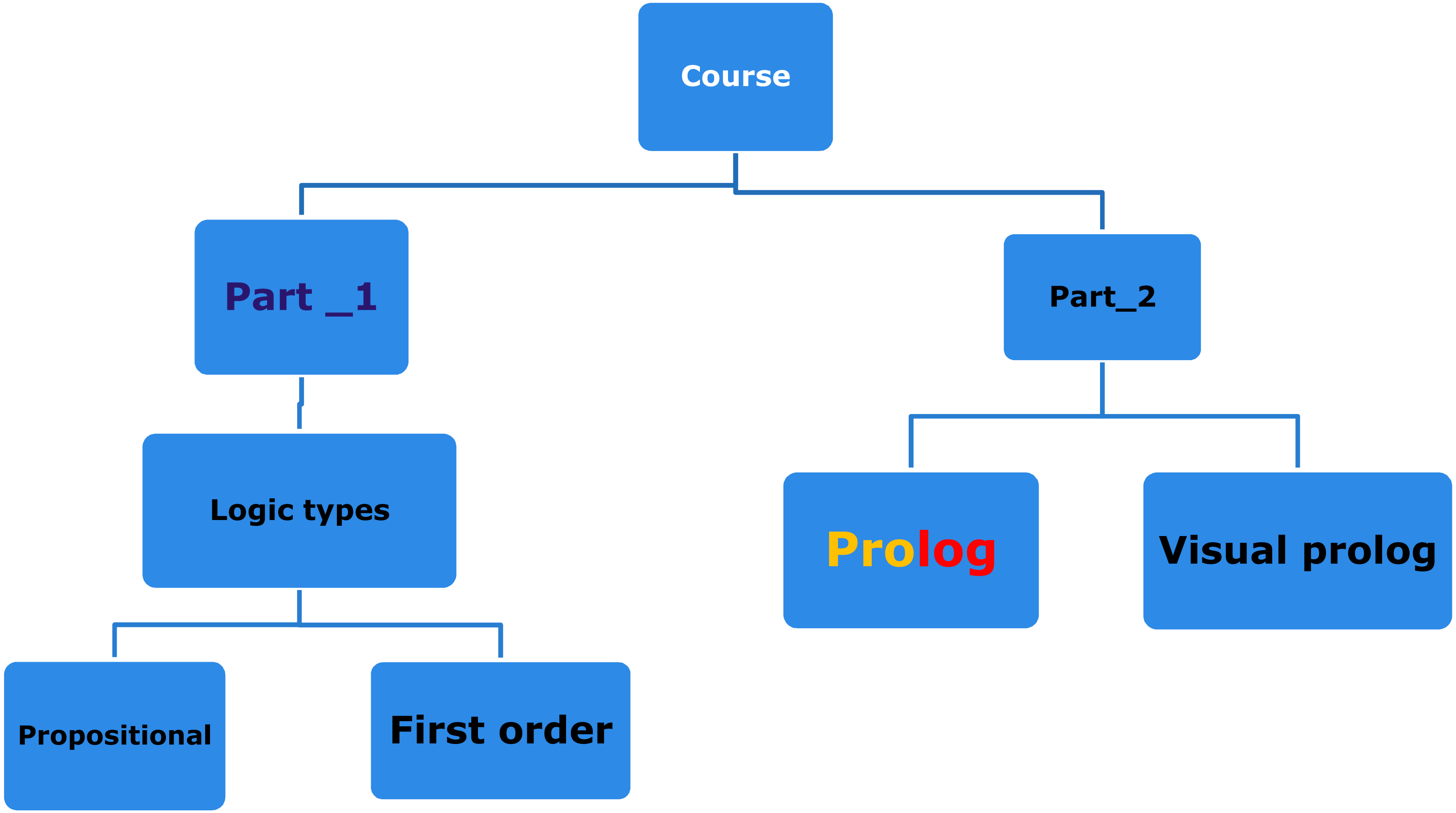
I Have a CourseSites Account

Use this option if you ***already have*** a CourseSites account. You will be asked to enter your login credentials, and then will be enrolled into the course.

I Need a CourseSites Account

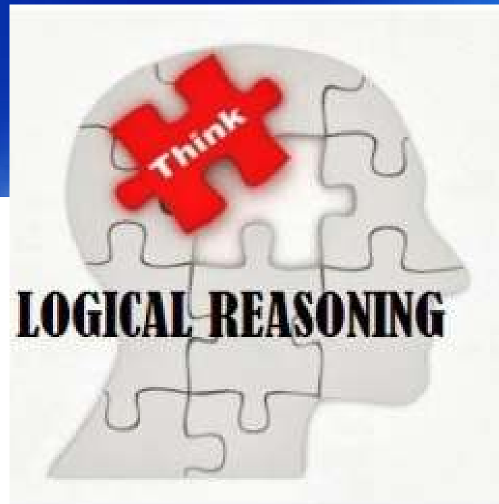
Use this option if you ***do not have*** a CourseSites account. You will be asked to register, and then will be enrolled into the course.

Course Description



- Syntax
- Semantic
- Inference (reasoning)

Human Logic



- Logic is the science of reasoning.

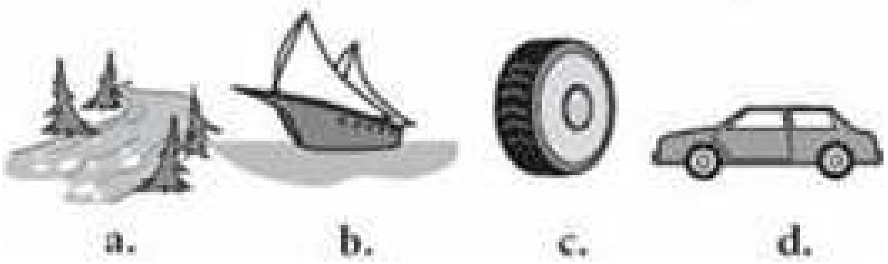
- Human brains are information processors, We acquire information about the world and use this information to form our behavior and actions.

- The strengths of human information processing is our ability to represent and manipulate logical information, not just simple facts but also more complex forms of information, such as negations, alternatives, constraints, and so forth.

Human Logic

Examples:

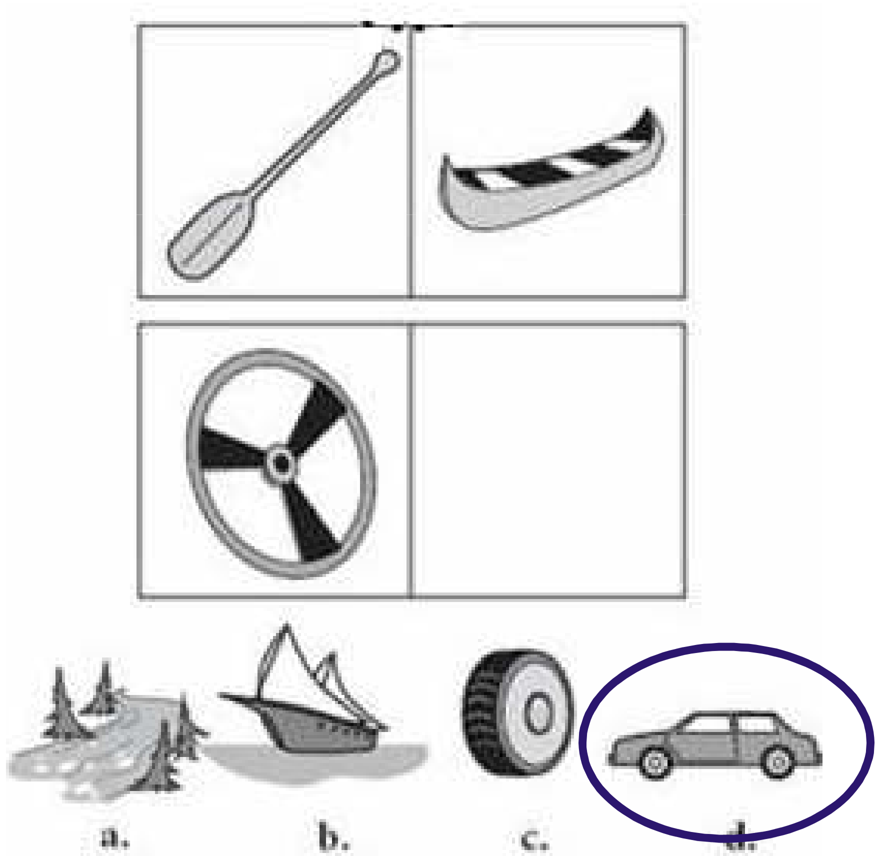
1) Objects relationships



Human Logic

Examples:

1) Objects relationships



Human Logic

<u>Examples</u>:

2) Logical Conclusions

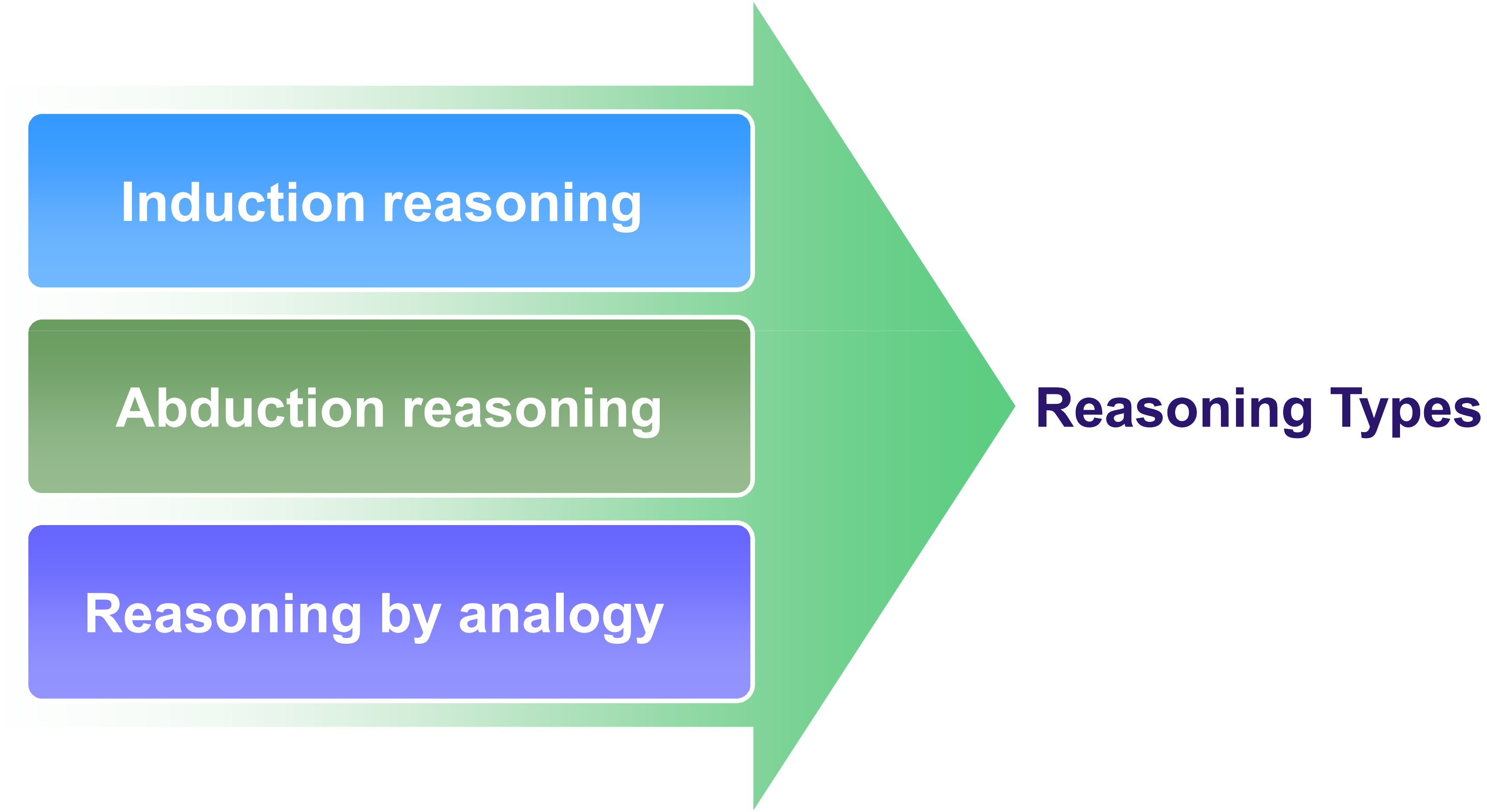
a)

- **Either Ali or Ahmed will drive the car.**
- **Ali will not drive the car.**
- **<u>Therefore</u>, Ahmed will drive the car.**

b)

- **All men are mortal.**
- **John is a man.**
- **<u>Therefore</u>, John is mortal.**

Types of Reasoning



Types of Reasoning

Induction Reasoning:

Reasoning from particular to general.

Example:

- I have seen 1000 black cars in my city.
- I have never seen a car that is not black.
- Therefore, every car in my city is black.

Types of Reasoning

Abduction Reasoning:

Reasoning from effects to causes.

Example:

- If there is no fuel, the car will not start.
- The car will not start.
- Therefore, there is no fuel.

Types of Reasoning

Reasoning by analogy:

Reasoning based on similarity of two situations.

<u>Example:</u>

- The flow in a pipe is proportional to its diameter.
- Wires are like pipes.
- <u>Therefore</u>, the current in a wire is proportional to its diameter.

Formal Logic

- **Formal Logic is a formal version of human logic.**

- **Formal Logic eliminates natural Languages difficulties using formal language for encoding information.**

EXAMPLE :

- Ahmed is three times as old as Ali.
- Ahmed's age and Ali's age add up to twelve.
- How old are Ahmed and Ali ?

Formalization

- $x - 3y = 0$
- $x + y = 12$

$$x = 9$$
$$y = 3$$

Formal Logic

EXAMPLE :

- Ahmed like reading and swimming .
- If Ahmed like some thing, then Ali like it.
- Does Ali like swimming ?

Formalization

Likes(Ahmed, reading)
Likes(Ahmed, swimming)
Likes(Ahmed, X) → likes (Ali, X)

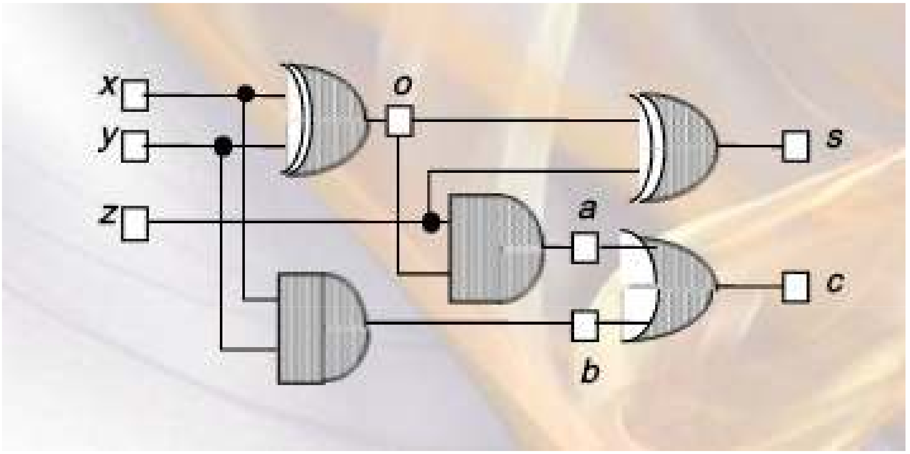
Formal Logic

- Given the syntax and semantics of this formal language, we can give a precise definition for the notion of logical conclusion. Moreover, we can establish precise reasoning rules that produce logical conclusions.

- Example:
  - Propositional logic
  - First Order Logic
  - Higher order Logic

Logic Applications

❖ Hardware Engineering:



$$o \Leftrightarrow (x \wedge \neg y) \vee (\neg x \wedge y)$$

$$a \Leftrightarrow z \wedge o$$

$$b \Leftrightarrow x \wedge y$$

$$s \Leftrightarrow (o \wedge \neg z) \vee (\neg o \wedge z)$$

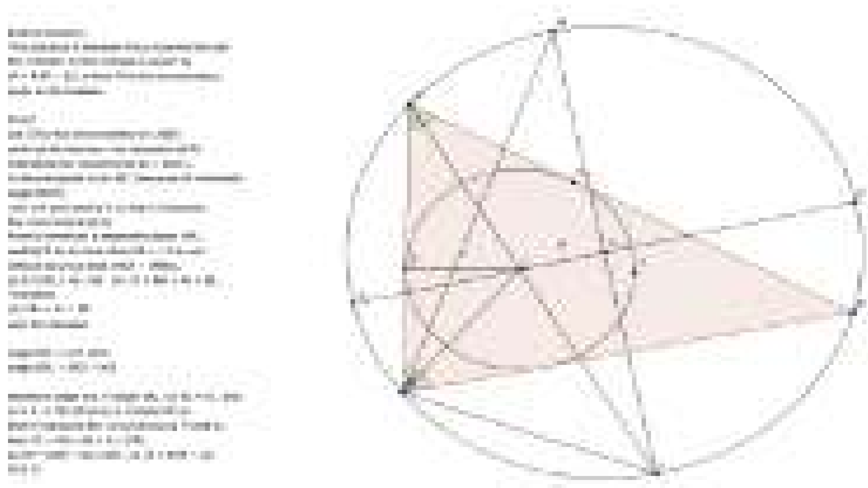
$$c \Leftrightarrow a \vee b$$

- **Simulations**
- **Configuration and simplification**
- **Diagnosis**
- **Testing**

Logic Applications

❖ Mathematics:

- Automated reasoning programs could be used to check mathematical proofs and in some cases, could be used to produce proofs or portions of proofs.



Logic Applications

❖ **Data Bases:**

parent

art	bob
art	bea
bea	coe

$parent(art, bob)$

$parent(art, bea)$

$parent(bob, coe)$

Constraints

$$\neg parent(x, x)$$

$$parent(x, y) \Rightarrow \neg parent(y, x)$$

Definitions

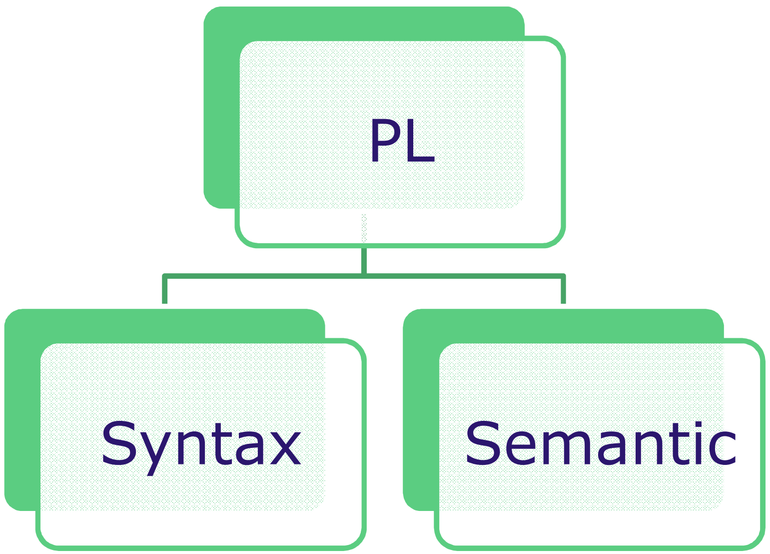
$$parent(x, y) \wedge parent(y, z) \Rightarrow grandparent(x, z)$$

Propositional logic (PL)

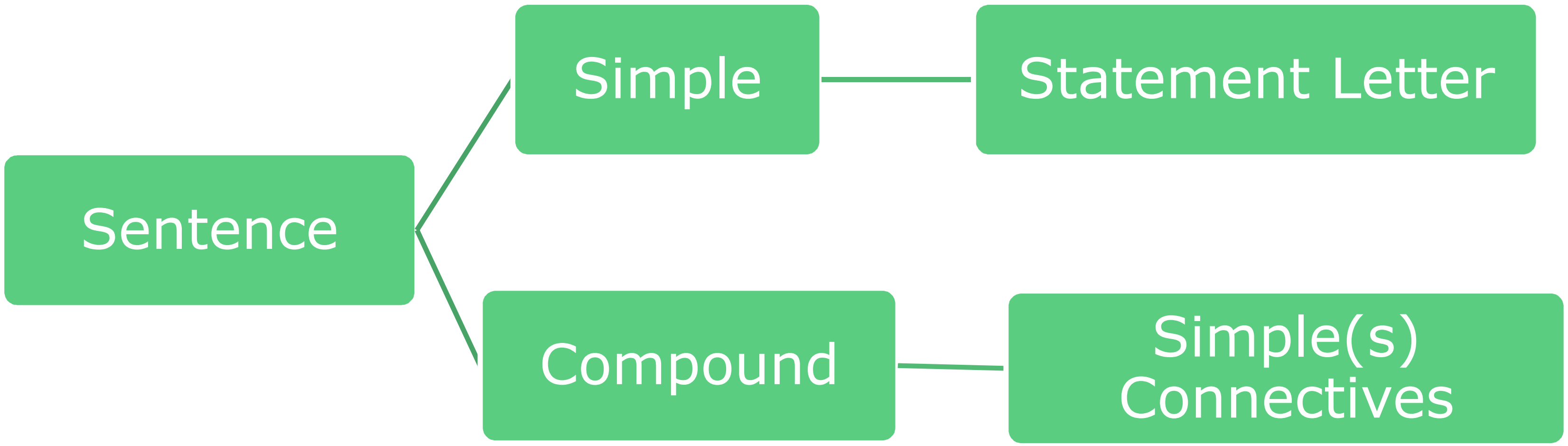
- Proposition is a declarative sentence that is either true or false
- The following are propositional statements:
  - It is raining
  - 5 is a prime number
  - 2+2 = 5
- The following are **not** propositional statements:
  - Are you hungry?
  - Shut the door!
  - X>8

Propositional logic (PL)

- PL is a logical model which is designed to represent propositional sentences as well as the logical relationships of combining or altering sentences.
- PL also known as **sentential logic**



PL Syntax



Statement letter

- A statement letter is any lower case letter written with or without a numerical subscript
- According to this definition: ‘p', ‘q', ‘p2', ‘p_2', and ‘p14' are examples of statement letters

PL Syntax

❖ connective

- A connective or operator of PL is any of the signs:

Symbol	Purpose
'¬'	not
'&'	and
'v'	or
'→'	Implication (If then)
'↔'	Equivalence (If and only if)

5 SIGNS

❖ Note

- Some textbooks use:

'~ ' instead of '¬'

'^' instead of '&'

PL Syntax

❖ ***PL syntax rules:***

1. Any statement letter is a well-formed formula.
2. If α is a well-formed formula, then so is (α).
3. If α is a well-formed formula, then so is $\neg\alpha$.
4. If α and β are well-formed formulas, then:

   $\alpha \,\&\, \beta$

   $\alpha \vee \beta$

   $\alpha \rightarrow \beta$

   $\alpha \leftrightarrow \beta$

   are well formed formulas.
5. Nothing else is a wff

PL Syntax

Check the syntax for the sentence: **$\neg p \wedge (q \rightarrow s)$** using PL syntax rules.

1. The statement letters p, q, s are wffs (by rule 1)
2. Since p is a wff, then $\neg p$ is a wff (by rule 3)
3. Since q and s are wffs, then $q \rightarrow s$ is a wff (by rule 4)
4. Since $q \rightarrow s$ are wffs, then $(q \rightarrow s)$ is a wff (by rule 2)
5. Since $\neg p$ is a wff and $(q \rightarrow s)$ is a wff, then $\neg p \wedge (q \rightarrow s)$ is a wff (by rule 4)

PL Syntax

Check the syntax for the following sentences

Sentence	PL_Check
p &¬p	wff
pqr	not wff
¬p ∨	not wff
¬p ∨ ¬p	wff
¬(q ∨ r) ¬q → ¬¬p	not Wff
(p & q) ∨ (p ¬& q)	not Wff

PL Syntax

Formalize the following sentences into PL:

- If a person is cool or funny, then he is popular

$$c \vee f \rightarrow p$$

- person is popular if and only if he is either cool or funny.

$$p \leftrightarrow c \vee f$$

- There is no one who is both cool and funny

$$\neg(c \,\&\, f)$$

Practical Assignment

Write a computer program which check the syntax structure for any given PL sentence.

[you are free to use **any** programming language...]


Chapter 1
Thank You !