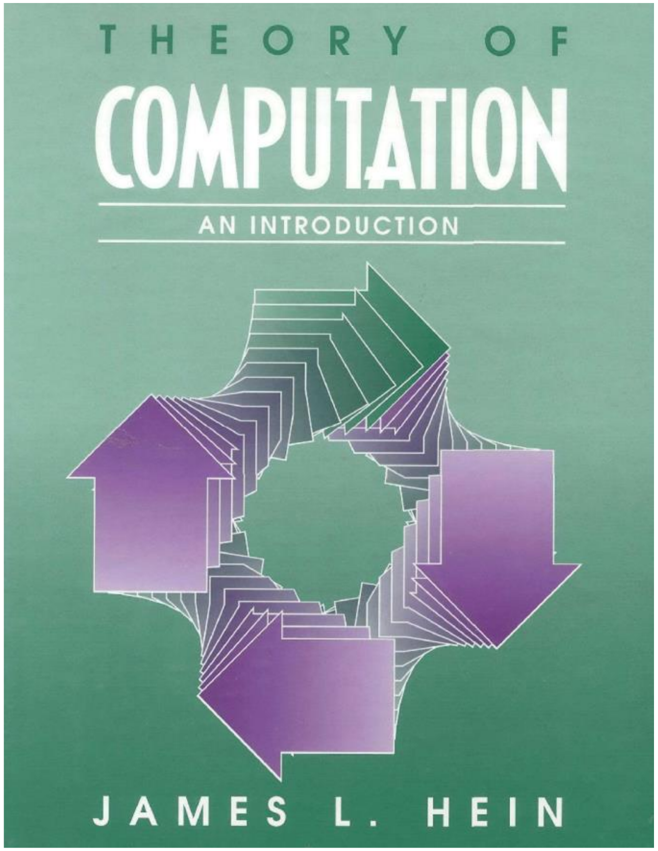
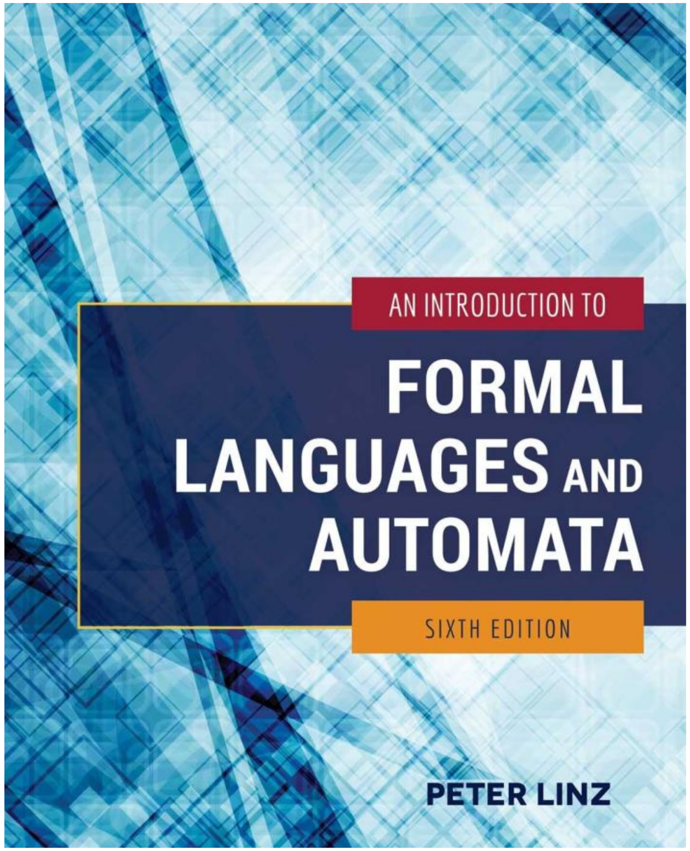
# Automata and Formal Languages

Lecture 01

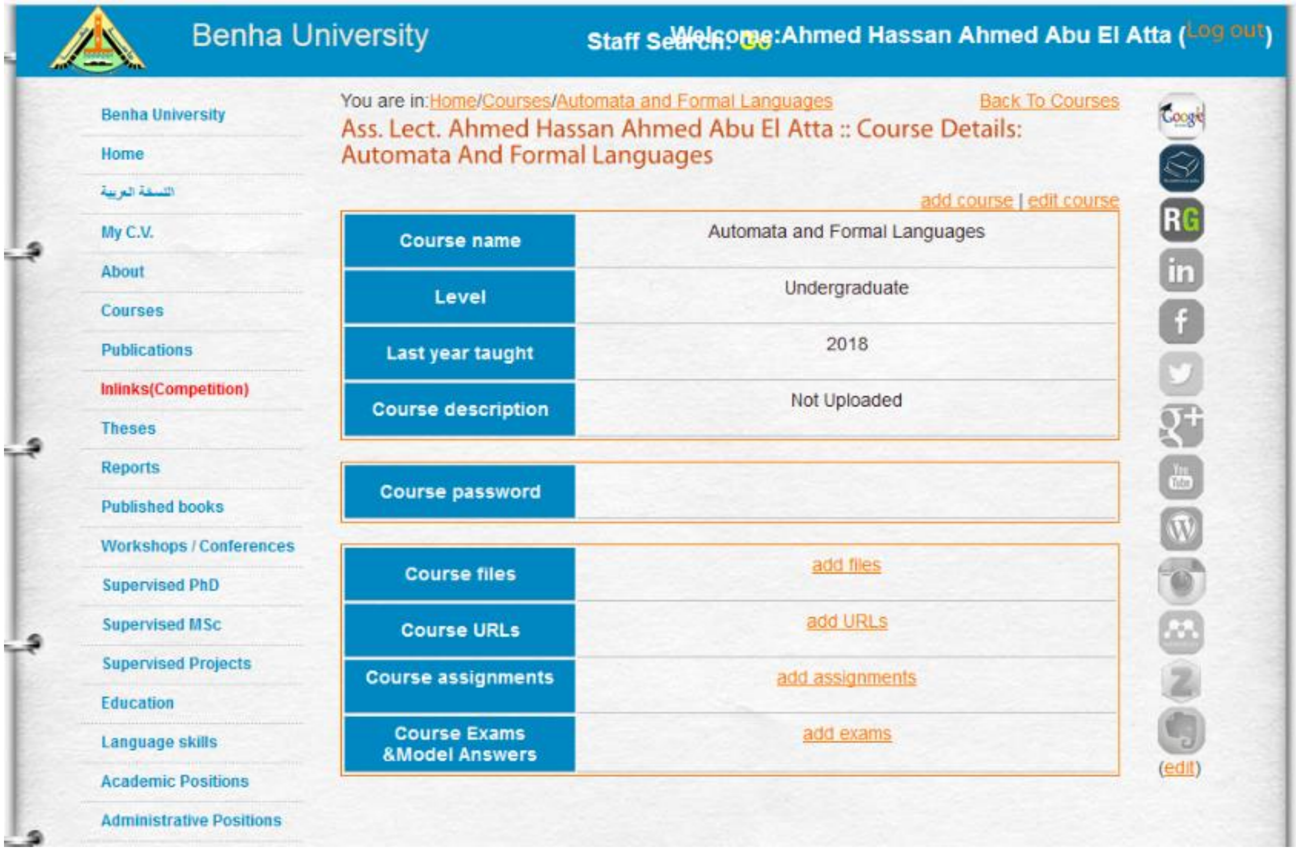
# Books


THEORY OF
COMPUTATION
AN INTRODUCTION
JAMES L. HEIN


AN INTRODUCTION TO
FORMAL
LANGUAGES AND
AUTOMATA
SIXTH EDITION
PETER LINZ

# PowerPoint

http://www.bu.edu.eg/staff/ahmedaboalatah14-courses/14767

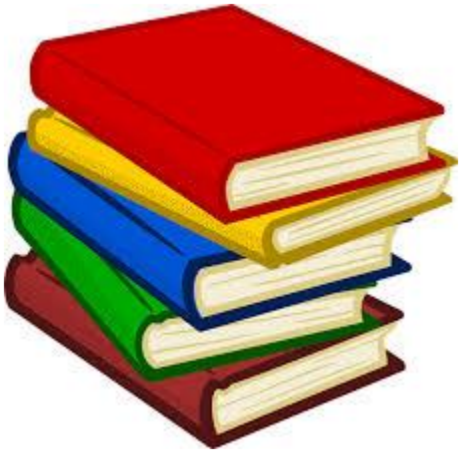


# Agenda

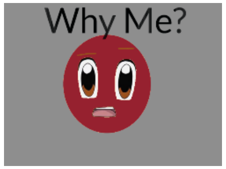
- Course Contents
- Why???
- A Simple Vending Machine
- Language
- Words
- Combining Languages
- Word Concatenations
- Properties

# Course Contents

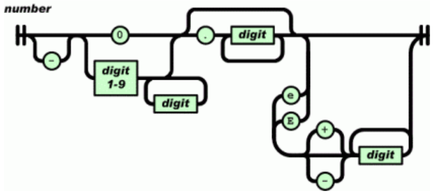
- Introduction to formal languages.
- Regular Languages and Finite Automata
  - Regular Languages
  - Regular Expressions
  - Finite Automata
  - Regular Grammar
  - Pumping Lemma (Regular Languages)
- Context-Free Languages and Pushdown Automata
  - Context-Free Languages
  - Pushdown Automata
  - Context-Free Grammars



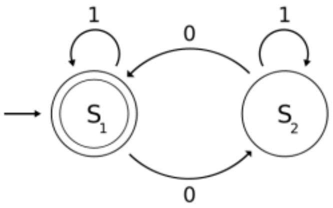
# Why?????????



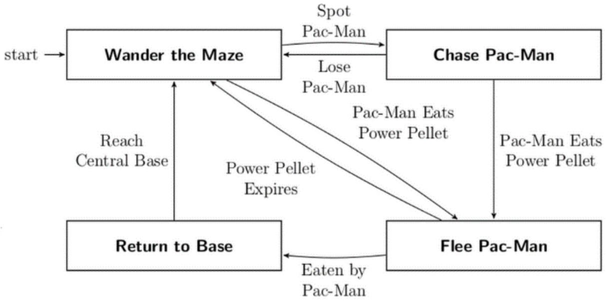

In compilers ,interpreters , ……



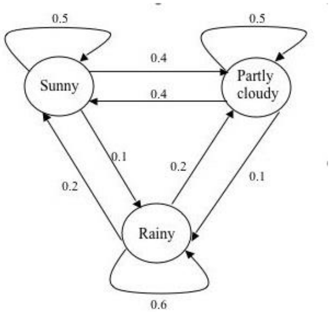

natural language processing




a model of computation




artificial intelligence




in probability

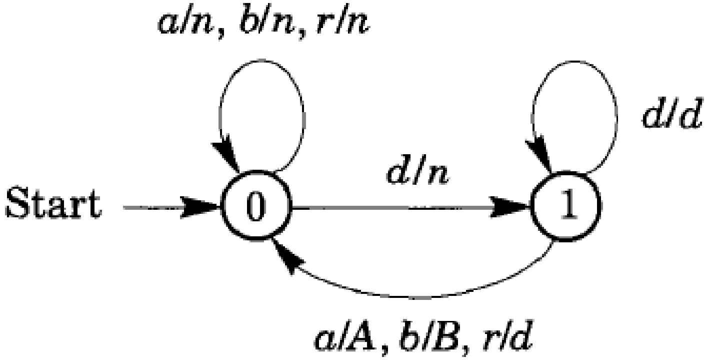
………

# A Simple Vending Machine

Suppose we have a simple vending machine that allows the user to pick from two 10-cent items A and B. (To simplify things, the slot will accept only dimes.)

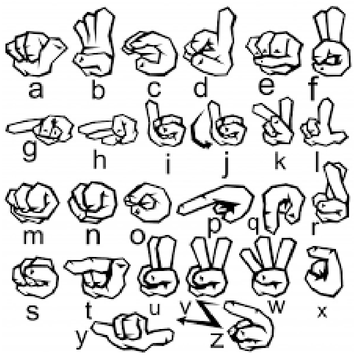

# A Simple Vending Machine

- There are four inputs to the machine:
  - d (dime),
  - a (select item A),
  - b (select item B),
  - and r (return coins).
- The outputs will be
  - n (do nothing),
  - A (vend item A),
  - B (vend item B),
  - and d (dime).



# Language

- A *language* is a set of strings.
- An alphabet is a set of elements (letters).
- If A is an alphabet, then a *language* over A is a collection of strings whose components come from A.
- Recall that A* denotes the set of all strings over A.
- So A* is the biggest possible language over A,
- and every other language over A is a subset of A*.

# Words

A **word** over an alphabet is a finite sequence of letters.

$A = \{a\}$

$W_1 = aaa$

$A = \{a,b\}$

$W_1 = aaa$

$W_2 = aba$

# Examples

$A = \{a\}$

$L_1 = \emptyset$

$L_2 = \{\in\}$

$L_3 = \{\in, a, aa, aaa, aaaa, \ldots\}$

$A = \{a, b\}$

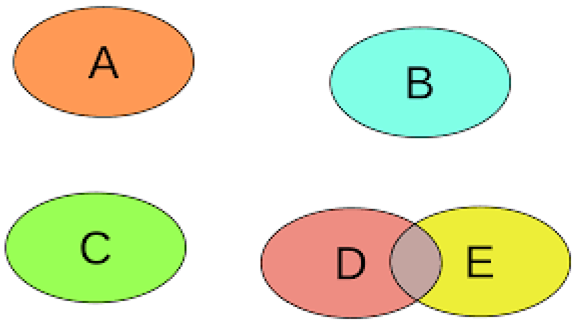
$L_1 = \{a, aa, aaa, \ldots\}$

$L_2 = \{aa, ab, ba, bb\}$

$L_3 = \{a, aa, ab, aaa, aab, aba, abb, \ldots\ldots.\}$

# Combining Languages

- Since languages are sets of strings, they can be combined by the usual set operations of:
  - union,
  - intersection,
  - difference,
  - and complement



# Concatenations

- Another important way to combine two languages ***L*** and ***M*** is to form:
  - the set of all concatenations of strings in ***L*** with strings in ***M***.
- This new language is called the product of ***L*** and ***M*** and is denoted by ***L•M*** .
- $L \cdot M = \{s \cdot t | s \in L \; and \; t \in M\}$

# Word Concatenations

W = bb

U = ab

V = b

W.U = bbab

U.W = abbb

W.W = bbbb

W.V = bbb

V.W = bbb

Is W.U = U.W for any two words W and U?

# Example

- L = {ab, ac)
- and M = (a, bc, abc)

L .M = (aba, abbc, ababc, aca, acbc, acabc}.

# Properties

L.{A) = {A}.L =*L.*

***L.ø = ø.**L**= ø.***

$L^0 = \{\Lambda\},$

$L^n = L \cdot L^{n-1} \quad \text{if } n > 0.$

$L^* = L^0 \cup L^1 \cup L^2 \cup \ldots .$

$L^+ = L^1 \cup L^2 \cup L^3 \cup \ldots .$

# Properties of Closure

Let $L$ and $M$ be languages over the alphabet $A$. Then

a) $\{\Lambda\}^* = \varnothing^* = \{\Lambda\}$.

b) $L^* = L^* \cdot L^* = (L^*)^*$.

c) $\Lambda \in L$ if and only if $L^+ = L^*$.

d) $(L^* \cdot M^*)^* = (L^* \cup M^*)^* = (L \cup M)^*$.

e) $L \cdot (M \cdot L)^* = (L \cdot M)^* \cdot L$.