CSE 105: Introduction to the Theory of Computation

Professor Jeanne Ferrante, CSE

TA's: Michel Abdalla,Sam Hamilton, Sara Miner

Class home page:
http://www-cse.ucsd.edu/classes/fa99/cse105

What is the class about?

How will it be run?

Why should you take it?

Finite automata, Ch. 1.1

Reading asst.: Ch 0, 1

Introduction to the Theory of Computation

What's a computer? a computation?

Ans1: What current machines are, and what they can do.

***Changes over time. Doesn't give us* LIMITS.**

Ex. Can machines ever .....?

What problems are computers capable of solving?
(Computability)

How easy/hard is a problem?
(Complexity)

What resources are needed to solve a problem?
(Computational Models)

WHY SHOULD YOU TAKE THIS COURSE?**>##?

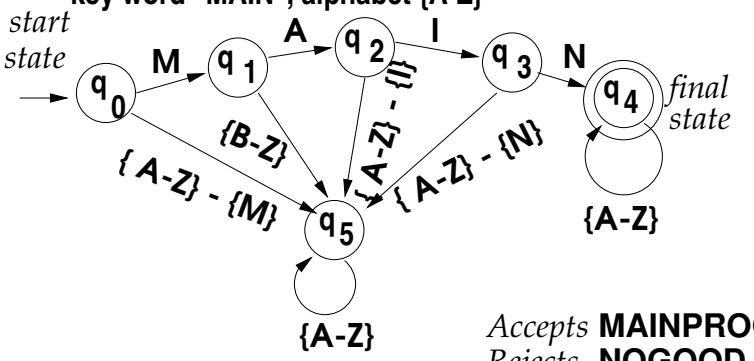
1. *It's fun stuff.*
2. **It will give you new insights into computers/ computation.**
3. ***It will give you knowledge* essential *to a computer scientist.***
4. **It will stretch your brain.**
5. ***Its essential for compilers (CSE 131A)***

   Regular expressions, finite automata, context-free grammars

A computational model: Finite Automata

A finite automaton models a computer with no separate memory.

Ex: State diagram of finite automaton that accepts key word "MAIN", alphabet {A-Z}



Accepts **MAINPROG**
Rejects **NOGOOD**

What are Finite Automata (FA) good for?

Recognize patterns in strings

e.g. keywords, constants in prog. languages

Ex. integers, reals, ...

***Used in editors like EMACS, tools like* Perl,**
a scripting language

Used in first phase of compilers, lexical analysis

string of symbols ⟶ **tokens of language**

const x = 5 → **const** | **x** | **=** | **5**

Control systems, household devices

Ex. control of refrigerator light, automatic door,...

But first, a break for some definitions....

Alphabet Σ ***:*** any finite set of symbols

Ex. {0,1}

String over Σ***:*** any finite sequence of symbols

Exs. 01 0 111

Empty string ε

Concatenation of strings: Given x and y, xy

$x\varepsilon = \varepsilon\ x = x$

Ex. x= dog, y= house, xy = doghouse

x^2 = dogdog x^3 = dogdogdog

Language over Σ ***:*** any set of strings over Σ

Ex. {ε, 0, 01, 10}

Ex. set of all strings over {0,1}

Formal Defintion of Finite Automaton

A *finite automaton M* is a 5-tuple (Q, Σ, δ, q_0, F)
where,

1. Q is a finite set of *states*
2. Σ is a finite set called the *alphabet*
3. δ : Q x Σ → Q the *transition function*
4. q_0 in Q, the *start state*
5. F $\subseteq$ Q, set of *accept (final) states*

The *language accepted by fa M is L(M).*
We say that M *accepts* L(M).

Previous ex:
Q = { q_0, q_1, q_2, q_3, q_4 , q_5 } Σ = {A, B, C, ...Z}

δ(q,a) = q_0 start F = {q_4 } L(M) =

7

Formal Definition of Computation on FA

M = (Q,Σ, δ, q_0, F)
w = w1 w2 w3 ... wn , a string over Σ

We say *M accepts w* if there is a sequence of states r0, r1, ..., rn in Q, such that

1. r0 = q_0 *(starts correctly)*
2. δ (ri, wi+1) = ri+1, for i = 0,...n-1 *(moves correctly)*
3. rn is in F *(ends in accept state)*

A language is *regular* if some fa accepts it.

8

Some examples of FA

1. Q = { q0, q1} Σ = {a,b} q0 start F = {q1}

   δ given by

	a	b
q0	q0	q1
q1	q1	q0

2. M = ({q1,q2}, {0,1}, , q1, {q2}) *(Ex. 1.2, p. 37)*

   δ given by

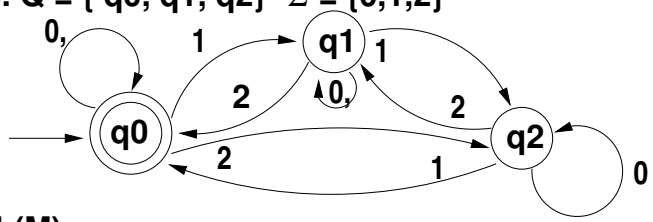
	0	1
q1	q1	q2
q2	q1	q2

State diagram? L(M) =

Change accept state of 2?

More examples of FA

3. Q = { q0, q1, q2} Σ = {0,1,2}



L(M) =

4. Construct a fa to accept the language over {a,b} which contains an even number of b's.

What info is needed about input string in order to accept/ reject?

Make states of fa to keep track of that info.

5. L = {w | w does not contain 3 b's as a substring}

Regular Operations on Languages

A, B languages

$A \cup B = \{ x \mid x \text{ in } A \text{ or } x \text{ in } B\}$ *(union)*

$A \circ B = \{xy \mid x \text{ in } A \text{ and } y \text{ in } B\}$ *(concatentation)*

$A^* = \{x1\ x2 \ldots xk \mid k \geq 0 \text{ and each } xi \text{ in } A\}$ *(star)*

Th: **If A1 and A2 are regular, then $A1 \cup A2$ is regular.**

Idea: A1 has fa M1 that accepts A1, A2 has M2, accepts A2.
We need fa A that accepts $A1 \cup A2$.
M is constructed from M1 and M2 by simulating both;
if either would accept, M should accept.
Can't do them one after the other; need to do simultaneously.
New state will be a pair of old states.

11

Proof: Regular languages are closed under union

Let M1 = $(Q1, \Sigma, \delta_1, q1, F1)$
Let M2 = $(Q2, \Sigma, \delta_2, q2, F2)$
Construct M = $(Q, \Sigma, \delta, q_0, F)$ to recognize
$A1 \cup A2$ as follows.

$Q = \{(r1, r2) \mid r1 \text{ in } Q1 \text{ and } r2 \text{ in } Q2\} = Q1 \times Q2$

(Cartesian product)

$\delta((r1,r2), a) = (\delta_1(r1, a), \delta_2(r2, a))$

$q_0 = (q_1, q_2)$

$F = \{(r1,r2) \mid r1 \text{ in } F1 \text{ or } r2 \text{ in } F2\}$

Qu: What if F = F1 X F2?
Another Qu: What about $A1 \circ A2$? $A1^*$?

12