January - 6th - 2025
Warren - Michigan.

Chapter I - Introduction. Set Theory

Set Theory

Outside Class

- Sed — Stream Editor (outside the class / LINUX command)
- awk — Al Aho
  Peter Weinberger
  Brian Kernighan } Extremely powerful command. Knowing awk will have an advantage.

Mastering Regular Expressions. O'Reilly By Jeffrey E. F. Friedl

Important Book about regular expressions.

From Here there is nothing course related only outside and can be important for us.

grep — Regular Expression Print

Chapter I - Sets

· x is an element of $S \Rightarrow x \in S$

y is not an element of $S \Rightarrow y \notin S$

Universal Set $\Rightarrow U$

· Everything is in universal set (number, symbols)

$$S = \{x, 3, 5, b, k\}$$

S_1 & S_2 are sets.

Union $S_1 \cup S_2 = \{x : x \in S_1 \text{ or } x \in S_2\}$ set expression

Intersection $S_1 \cap S_2 = \{x : x \in S_1 \text{ and } x \in S_2\}$

$S_1 - S_2 = \{x : x \in S_1 \text{ and } x \notin S_2\}$

$\bar{S} = \{x : x \in U, x \notin S\}$

ie. $S = \{1, 2, 3\}$ $U = \{\cancel{1, 2, 3}, 4, 5, a, b, c, A, B, C\}$

$\bar{S} = \{4, 5, a, b, c, A, B, C\}$

null set $\Rightarrow \emptyset$ (empty set)

$S \cup \emptyset = S = S - \emptyset$

$S \cap \emptyset = \emptyset$

$\overline{\emptyset} = U$

$\overline{\overline{S}} = S$

De Morgan's Law

- $\overline{S_1 \cup S_2} = \overline{S_1} \cap \overline{S_2}$
- $\overline{S_1 \cap S_2} = \overline{S_1} \cup \overline{S_2}$

Subset

$S_1 = \{1, 2, 3\}$ $S = \{1, 2, 3, 4\}$

$S_1 \subseteq S$ $\Rightarrow$ $S_1 = \{1, 2, 3\}$ $S = \{1, 2, 3, 4\}$

$S_1 \subset S$

No common Element

$S = \{2, 4\}$ $S_1 = \{5, 6, 7\}$ ⇒ Disjoin Set

$S_1 = \{2, 3\}$ $S_2 = \{2, 3, 5, 6\}$

$S_1 \times S_2 = \{(2,2), (2,3), (2,5), (2,6)$
$(3,2), (3,3), (3,5), (3,6)\}$

Functions → Relations.

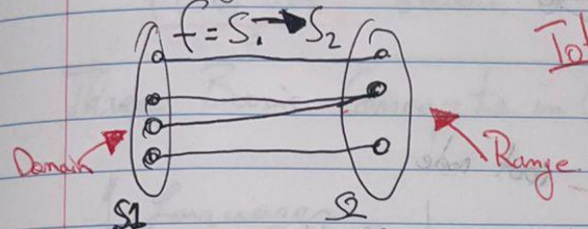
$f = S_1 \longrightarrow S_2$ (function is the rule)

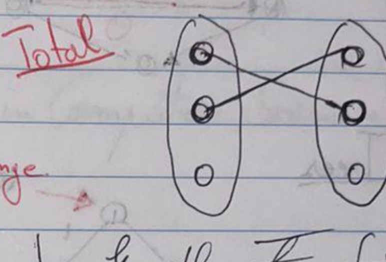
f = function
S_1 = Domain of f
S_2 = Range of f

Warren - Michigan
January 13th 2025

Chapter I: Intro Cont.

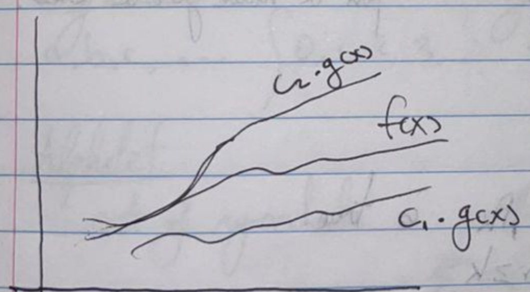
A function is a rule assign to elements of one set a unique elements of another set.





Growth Function

$$c_1 \cdot |g(x)| \le |f(x)| \le c_2 \cdot |g(x)|$$

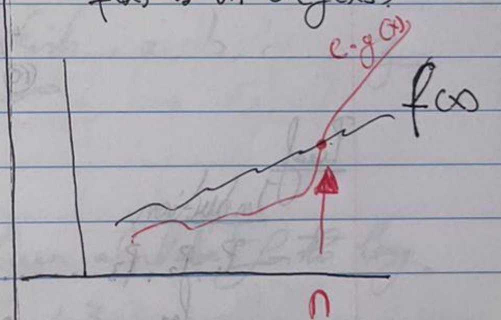


$f(x)$ is in $\Theta(g(x))$

Growth Functions

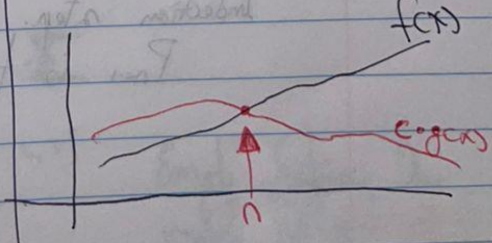
$f(x) \le c \cdot |g(x)|$

$f(x)$ is in $O(g(x))$

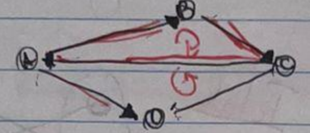


$f(x) \ge c \cdot |g(x)|$

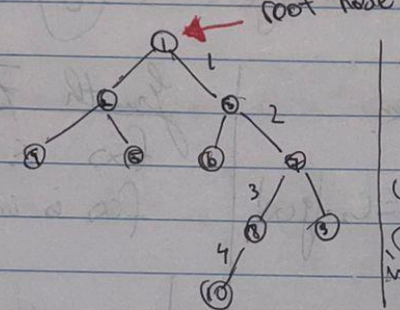
$f(x)$ is in $\Omega(g(x))$



Graphs (Vertices and Edges connected)



Trees



Height is 4

You can have as many childs in a node for a tree.

Proof

Induction.

$P_1, P_2, P_3 \ldots\ldots, P_k$ is true.

for any $n \geq k$

$\underbrace{P_1, P_2, P_3 \ldots P_n}_{\text{True.}}$

Induction step.

P_{n+1} is true

Proof Cont.

- Contradiction.

$\boxed{P \text{ is True.}}$ assume P is false,

Three Basic Concepts in Computer Science

1. Languages
2. Grammars.
3. Automata.

Language.

Symbols. English, a, b, c ... z

a, b, c, ... {0, 1, 2, 3, ... ∞

Alphabet

set of symbols. Our alphabet for the lang.

$\Sigma = \{a, b\}$

$\Sigma = \{0, 1, 2, 3\}$

String

Sequence of symbols from Σ

$\Sigma = \{a, b\}$ $\Sigma = \{0, 1, 2, 3\}$ Empty String λ

$v = aba$. $w = abaa$ $x = 2002$ $v = aba$ $|v| = 3$ → length of a string.

$|uv| = |u| + |v|$ $|\lambda| = 0$

~~Empty~~ String Cont.

Contentination of Strings

$w = aba$
$v = cda$
$wv = abacda$

Length of 2 Strings

$|wv| = |u| + |v|$

Reverse of a String

$w = a_1 a_2 a_3 \dots a_n$
$w^R = a_n, a_{n-1}, \dots a_3\, a_2\, a_1$
$v = abc$
$v^R = cba$

Prefix and Suffix of a string

$w = abbab$, prefix of $w = \{\lambda, a, ab, abb, abba, abbab\}$

Suffix $\{\lambda, b, ab, bab \dots\}$

$\Sigma = \{a, b\}$

Σ

Powers of Sigma

Because it contain λ

$\Sigma = \{a, b\}$ $\Sigma^0 = \{\lambda\}$ $|\Sigma^0| = 1 = 2^0$

$\Sigma^1 = \{a, b\}$ $|\Sigma^1| = 2 = 2^1$

$\Sigma^2 = \{ab, ba, aa, bb\}$ $|\Sigma^2| = 4 = 2^2$

$\Sigma^3 = \left\{\begin{matrix} aaa, aab, aba, abb, \\ baa, bab, bba, bbb \end{matrix}\right\}$ $|\Sigma^3| = 8 = 2^3$

Σ^n w/ length n $|\Sigma^n| = 2^n$

$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, \ldots\}$

(All possible length strings to that substring)

• Remember that it contains λ any combination of strings

$\Sigma^+ = \Sigma^* - \{\lambda\} = \{a, b, aa, \ldots\}$ (Excludes the λ)

$L = \{a, aa, aab\}$

- Power of Sigma.

$\Sigma = \{a, b\}$ $L = \{a^n b^n : n \geq 0\}$

$n=2$ $aabb \neq a^2 b^2$

$\{\lambda, ab, aabb, aaabbb \ldots\}$

$\Sigma = \{a, b\}$

$\bar{L} = \Sigma^* - L$

L prime — All elements excluding "L"

all possible

Our Language

$= \{\lambda, a, b, ab, ba, \ldots\}$
$- \{\lambda, ab, aabb, \ldots\}$
$\bar{L} = \{a, b, aabb, ba, \ldots\}$

$w \rightarrow$ Lower Case Omega

$L^R = \{w^R : w \in L\}$ Language Reverse.
$= \{\lambda, ba, bbaa, \ldots\}$

$L_1 L_2 = \{xy : x \in L_1, y \in L_2\}$

$L^n = L$ concatinated w/itself n times

$L^0 = \{\lambda\}$ $L^1 = L$ $L^2 = LL$ $L^3 = LLL$

$L^* = L^0 \cup L^1 \cup L^2 \ldots$ $L^+ = L^1 \cup L^2 \cup L^3 \ldots$

Grammars (G)

$G = (V, T, S, P)$

V = Finite set of variables
T = Finite set of Terminal Symbols
S = $S \in V$ = Start ~~Symbol~~. Variable (S is part of V)
P = Production rules

$G = (\{S\}, \{a, b\}, S, P)$.

P is $\quad S \to aSb \qquad \lambda$ ✓
$\quad S \to \lambda \qquad ab$ ✓

$S \to \lambda$ ✓ $\qquad aabb$ ✓

$S \to aSb \to ab$

$S \to aSb \to a\,aSb\,b \to aabb$

$L = \{a^n b^n : n \geq 0\}$

$G = (V, T, S, P)$ Derived to => Substitute any number of times to get the word

$L(G) = \{w \in T^* : S \overset{*}{\Rightarrow} w\}$ that is part of that lang.

Grammar Cont.

aaaaabbbbb

$S \to aSb$
$\to a\,aSb\,b$
$\to aa\,aSb\,bb$
$\to aaa\,aSb\,bbb$
$\to aaaa\,aSb\,bbbb$
$\to aaaaa\,bbbbb$

S derives that word

$L = \{a^n b^{n+1} : n \geq 0\}$

$\{b, abb, aabbb, aaabbbb \ldots\}$

$G = (\{S, A\}, \{a, b\}, S, P)$

$S \to Ab$
$A \to aAb$
$A \to \lambda$

$S \to Ab \to aAbb \to abb$

$S \to Ab \to aAbb \to aaAbbb \to aabbb$

January 27th - 2025
Detroit - Michigan

Grammar Cont.

$L = \{a^n b^{n+1} : n \geq 0\}$

b
abb
aabbb
aaabbbb

Variables

$S \to Ab$

$A \to aAb$

$A \to \lambda$

$S \to Ab \to b$

$S \to \underline{Ab} \to \underline{aAb}b \to abb.$

$S \to \overline{Ab} \to \underline{aAb}b \to aaAbbb \to aabbb$

Any string generated by $a^n b^{n+1}$ grammar is part of λ

$S \to aAb \mid \lambda$ aaAbb ?

$A \to aAb \mid \lambda$

$S \to \lambda$

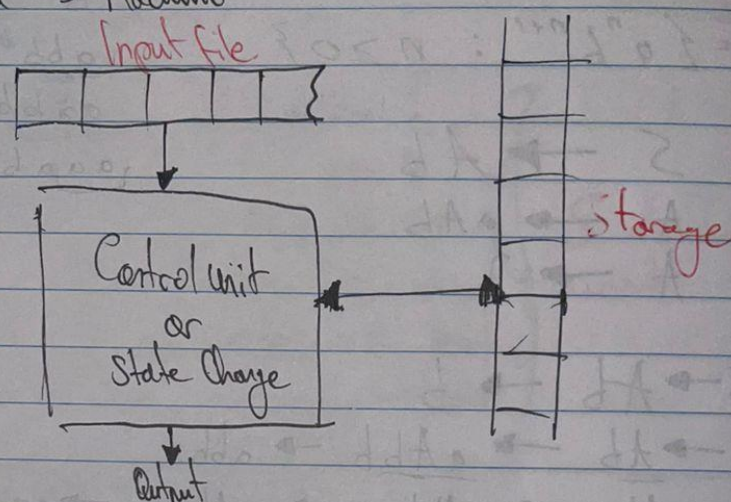
$S \to aAb \to ab$

$S \to a\underline{aAb}b \to aabb$

NULL, ab, aabb, aaabbb, . . .

$L(G) = \{a^n b^n : n \geq 0\}$

Autamata => Machine



- Accept or Reject are the only outputs produced by the ma

- Alphabet

$\Sigma = \{a, b\}$, Find Grammar

① All strings with exactly two a's.

$S \to AaAaA$ or $B \to bB \mid \lambda$

$A \to bA \mid \lambda$

all strings with at least two a's.

Not efficient

$(A \rightarrow AaA)$

$S \rightarrow AaAaA$

Non terminating variables $\leftarrow$ $A \rightarrow aA \mid bA \mid \lambda$ $\rightarrow$ Terminating variables.

$S \rightarrow AaAaA \rightarrow AaaA \rightarrow AaabA$

$\rightarrow Aaab \rightarrow bAaab$

$\rightarrow baAaab$

$\rightarrow babAaab$

$\rightarrow babaab$

Automata

the machine all strings with even numbers of b's

$S \rightarrow SbSbS \mid A \mid \lambda$

$A \rightarrow aA \mid \lambda$

bbbbaaa.

$S \rightarrow SbSbS$

$\rightarrow SbSbA \rightarrow SbSbaA \rightarrow SbSbaaA$

$\rightarrow SbSbaaaA \rightarrow SbSbaaa\lambda \rightarrow \lambda bSbaaa\lambda$

$\rightarrow \lambda bSbSbSbaaa\lambda \rightarrow \cancel{\lambda}b\cancel{\lambda}b\cancel{\lambda}b\cancel{\lambda}baaa\cancel{\lambda}$

$\rightarrow$ bbbbaaa

KALI

$L = \{a^{n+3} b^n : n \geq 2\}$ aaaaabb

aaaaaaabbb

$S \rightarrow aaaaaAbb$

$A \rightarrow aAb \mid \lambda$

$a^3 b^{3-3} = a^{0+3} b^0$

$L = \{a^n b^{n-3} : n \geq 3\}$ aaa

aaaab

$L = \{a^{m+3} b^m : m \geq 0\}$ aaaaabb

$S = aaaA$

$A = aAb \mid \lambda$

$\Sigma = \{a\}$

$L = \{w : |w| \bmod 3 = 2\}$ $|w| \bmod 3$

$S = aaaS \mid aa$ aa $2 \bmod 3 = 2$

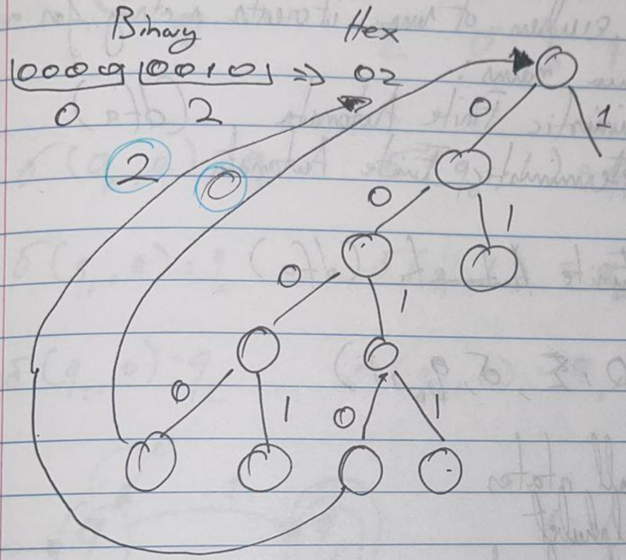
$S \rightarrow aa$ aaaaa $5 \bmod 3 = 2$

$S \rightarrow aaaS \rightarrow aaaaa$ aaaaaaaa $8 \bmod 3 = 2$

$S \rightarrow aaaS \rightarrow aaaaaaS \rightarrow aaaaaaaa$

$\boxed{\begin{array}{l} S = aaaA \mid A \\ A \rightarrow aaaA \mid aa \end{array}}$ Redundant

Transducer



Chapter II: Finite Automata

- Is a machine, when it runs it creates a string for a language
- Split into two mains:
  - Deterministic Finite Automata (dfa)
  - Non-Deterministic Finite Automata

- Deterministic Finite Automata (dfa)

$$m = (Q, \Sigma, \delta, q_0, F)$$

$Q \rightarrow$ set of all states

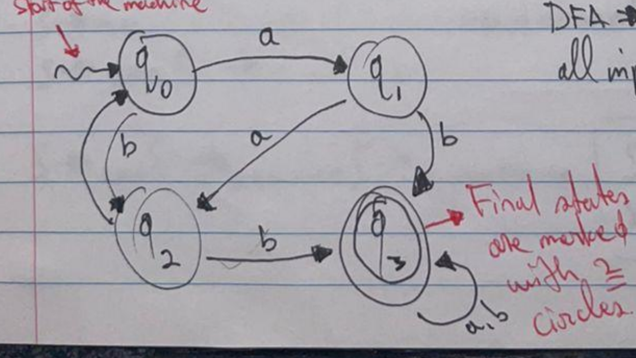
$\Sigma \rightarrow$ Input alphabet

$\delta \rightarrow$ transition function.

$$\delta : Q \times \Sigma \rightarrow Q$$

$q_0 \rightarrow \in Q$ starting state.

$F \rightarrow F \subseteq Q \rightarrow$ set of Final states



DFA → every node must handle all inputs.

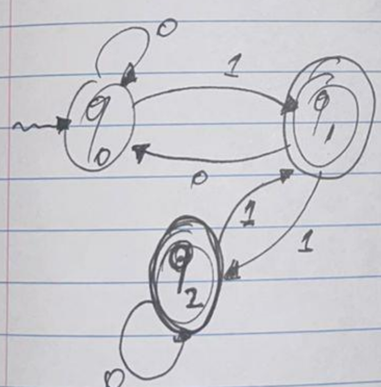
$m = (\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_1\})$

$Q \quad \Sigma \quad \delta \quad q_0 \quad F$

$\delta(q_0, 0) = q_0 \qquad \delta(q_0, 1) = q_1$

$\delta(q_1, 0) = q_0 \qquad \delta(q_1, 1) = q_2$

$\delta(q_2, 0) = q_2 \qquad \delta(q_2, 1) = q_1$



Final State

1 ✓

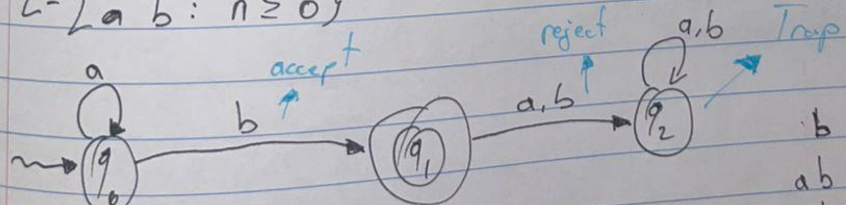
101 ✓

00101 ✓

100 X

1100 X

$L = \{a^n b : n \geq 0\}$



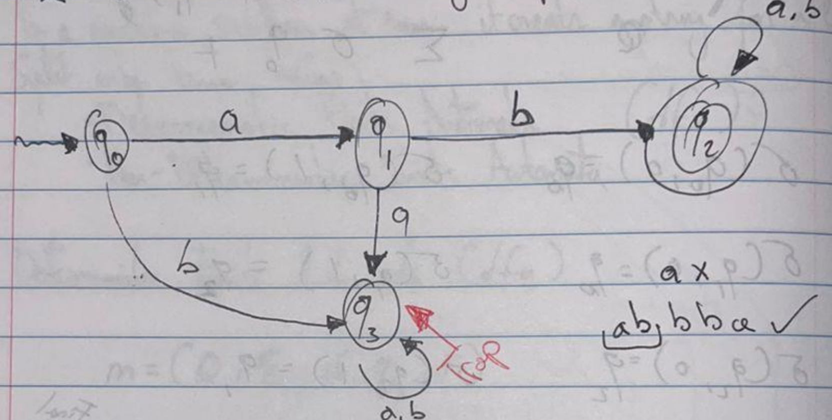
b

ab

aab

aaab

$\Sigma = \{a,b\}$ dfa starting w/ prefix ab



a ✗

ab, bba ✓

January 30th 2026
Detroit Michigan.

Assignment I: Basic Grammar.

1. Find grammar for $\Sigma = \{a, b\}$ that generate the sets of all strings with at least three a's.

$$S \rightarrow AaAaAa$$
$$A \rightarrow aA \mid bA \mid \lambda$$

2. Let $\Sigma = \{a, b\}$. For each of the following languages, find a grammar that generates it.

$L = \{a^{3n} b^{2n} : n \geq 2\}$

$n=2$ $a^6 b^4 \Rightarrow aaaaaabbbb$
$n=3$ $a^9 b^6 \Rightarrow aaaaaaaaabbbbbb$
$n=4$ $a^{12} b^8 \Rightarrow aaaaaaaaaaaabbbbbbbb$

$$S \rightarrow \cancel{A}\ aaaaaaAbbbb.$$
$$A \rightarrow \cancel{aaAb \mid \lambda}\ aaaAbb \mid \lambda$$

(3) 2.2 $L = \{a^{n+3} b^n : n \geq 2\}$

$n=2$ aaaaabb.
$n=3$ aaaaaabbb
$n=4$ aaaaaaabbbb

$$S \rightarrow aaaaaAbb$$
$$A \rightarrow aAb \mid \lambda$$

$S \to aA \mid aaA$
$A \to aaaA \mid \lambda$

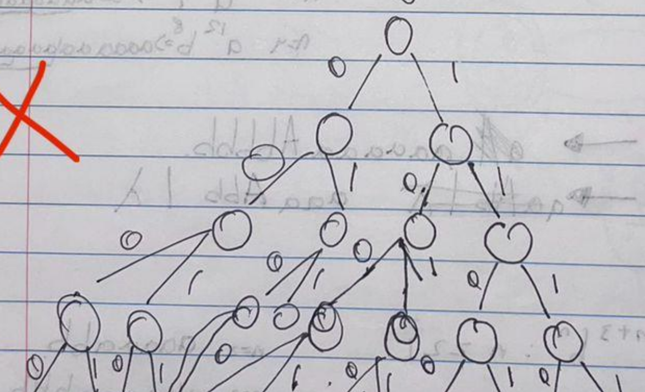
Correct Version

4. Find grammars for the following languages on $\Sigma = \{a\}$

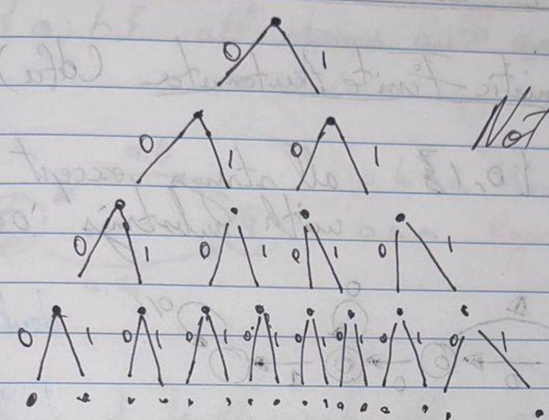
$L = \{w : |w| \bmod 3 > 0\}$ mod3 > 0

$S \to$ ~~aS | aa | aaa~~ $aA \mid aaA$ a mod 3 > 0
~~$A \to aaaA \mid \lambda$~~ aa mod 3 > 0
~~$S \to a \mid aa \mid aA$~~ aaa mod 3 > 0
~~$A \to aA \mid aaaA \mid a$~~ 6 mod 3 = 0?

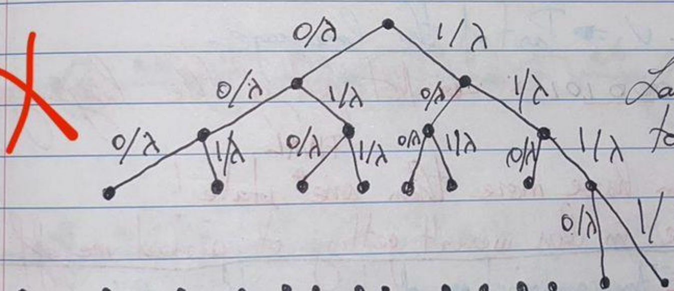
5. Design a transducer to convert a binary string into hexadecimal.
For example, the bit string 1111 1100 1001 1010 = FC9A



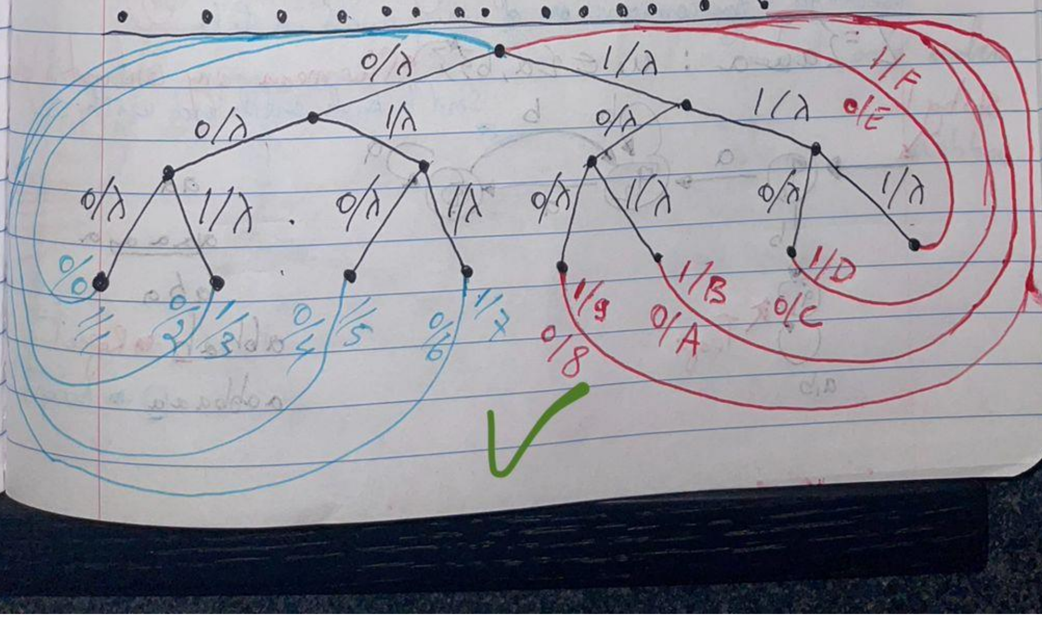
X



Not quite right



Last bit need to shown the result and 1/λ not return "λ"

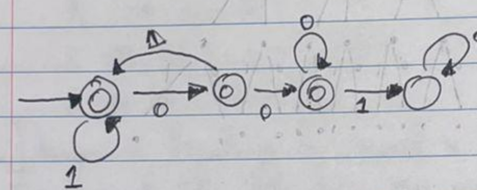


February 3rd 2025
Warren - Michigan.

Chapter II - Session II

- Deterministic Finite Automata (dfa)

$\Sigma = \{0,1\}$ all strings except strings with substrings '001'



double circle = Final state

011100 ✓ → Part of the language

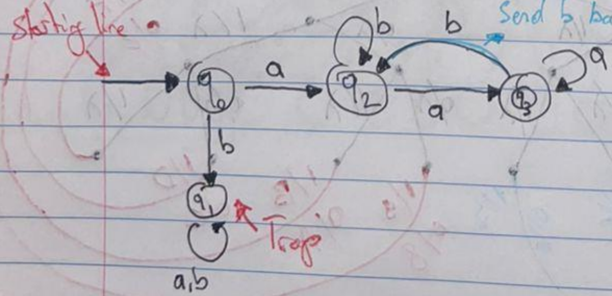
010000000101 × → Not part of the language

* You can have more than one FINAL state!
(People in class weren't getting it, pissed me off!!!)

little omega ⇒ Any combination of string with a & b

$L = \{awa : w \in \{a,b\}^*\}$ → Star means any combination of a and b

Send b back until end with a



aa

aaaaa

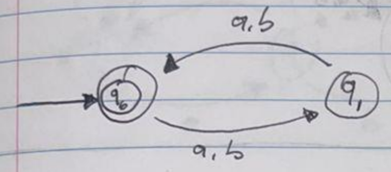
aba

abbab → Reject

aabbaaa → Accept

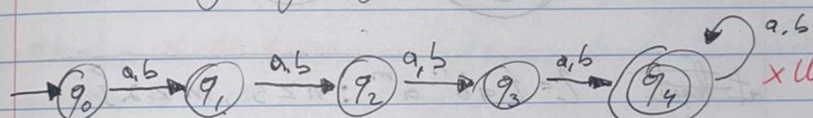
$\Sigma = \{a, b\}$ all strings with even length.

0
2
4
5
6.

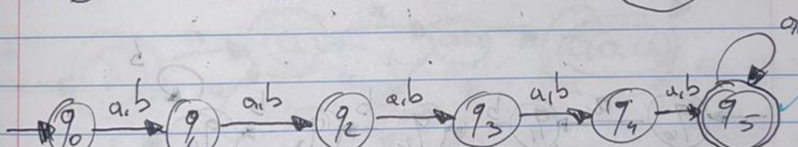


abba Accept
aaa Reject

All strings of length greater than 5 $\Sigma = \{a, b\}$



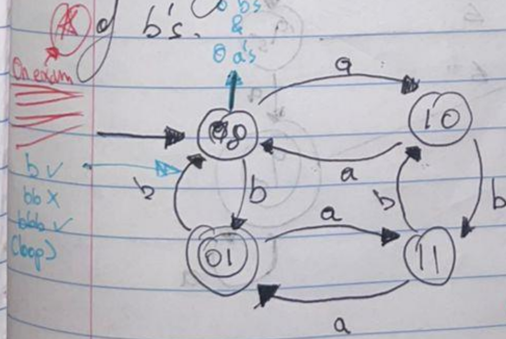
× Wrong



✓ Correct

all string with even number of a's and odd number of b's.

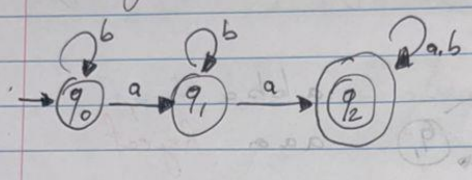
On exam



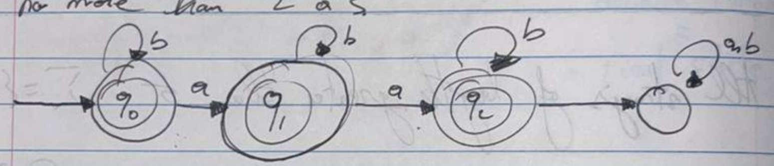
abbba abbba
abaaa
abbbaba
aba
abba

☆ Important (Exam question)

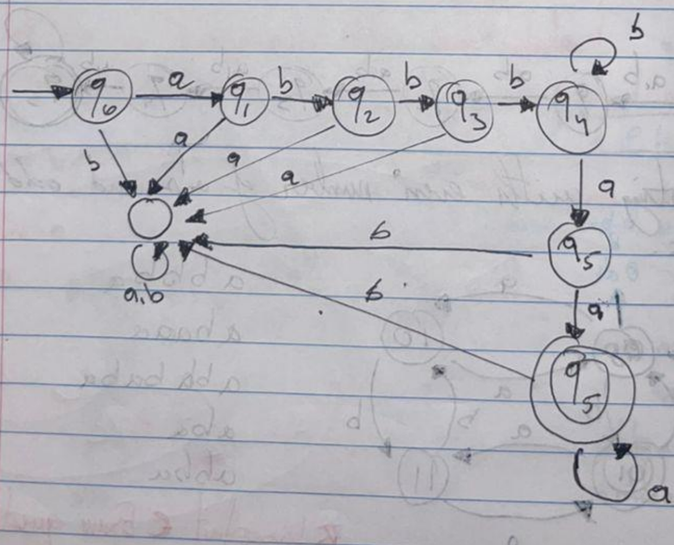
all strings with atleast 2 a's. AaAaA



no more than 2 a's



~~at least~~ $L = \{ab^n a^m : n \geq 3, m \geq 2\}$



EXAM

$L =$

$L =$

aba

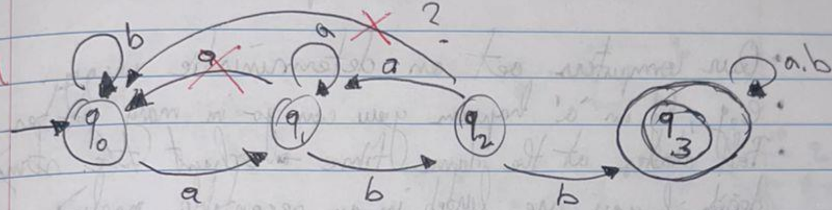
aaa

It is acceptable because lambda is not summing

Assume w's are empty for easy implementation

(A) ★ EXAM

$L = \{ w_1 \, abb \, w_2 : w_1 \in \{a,b\}^*, w_2 \in \{a,b\}^* \}$



$\Sigma = \{a, b\}$

$L = \{ w : w$ contain no run less than $3 \}$

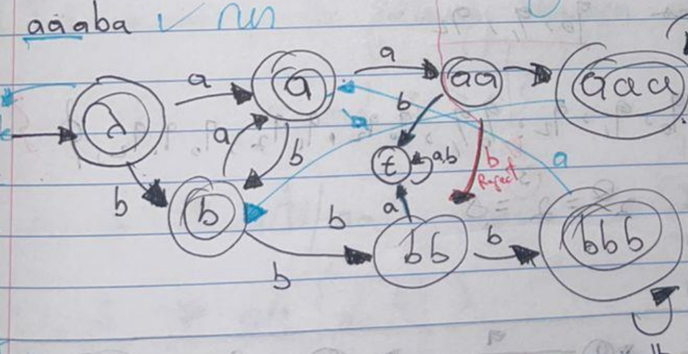
Wrong because there is a run

run

ababa × no run   (Running is at least (2))

aaaba ✓ run

It is acceptable because lambda is not running



Automata

- Non deterministic Finite Accepter (nfa)

- Our computers act on deterministic way.
- e.g., if an "a" happens you can go in many states
- Both states at the same time → exhaust the string and decide if you are landed in an acceptable machine.

Internal state, Alphabet, Transition, Initial, Final state (dfa)

$m = (Q, \Sigma, \delta, q_0, F)$

In nfa ⇒ $\delta = Q \times (\Sigma \cup \{\lambda\}) \rightarrow 2^Q$

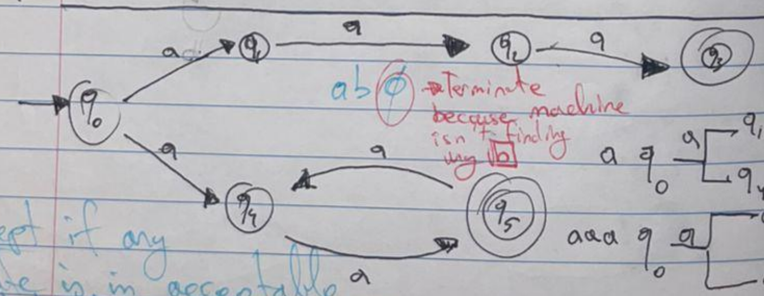
3 states q_0, q_1, q_2

$q_0 \rightarrow q_0, q_1, q_2, q_0 q_1, q_0 q_2, q_1 q_2, q_0 q_1 q_2, \phi$

$\phi \rightarrow$ zero

$2^Q = 2^3 = 8$

Final state = NULL state



How to check them?

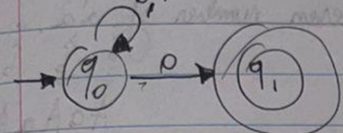
ab ϕ terminate machine because isn't finding any $\boxed{b}$

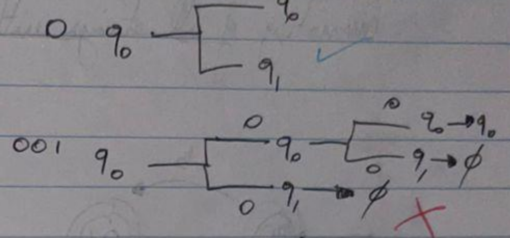
a $q_0 \xrightarrow{a} q_1, q_4$ ✗

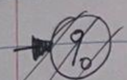
aaa $q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \xrightarrow{a} q_3$; $q_4 \rightarrow q_5 \rightarrow$ ✓

Accept if any state is in acceptable state!

$L = \{$set of all strings end with '0' → zero$\}$







-Redoing Problem IV foran A01: Basic grammar

IV. Find grammars for the following languages on $\Sigma = \{a\}$

$L = \{w : |w| \bmod 3 > 0\}$

$S \rightarrow aA \mid aaA$

$A \rightarrow aA \mid aaA \mid \lambda$

$0 \% 3 = 0$ ✗
$1 \% 3 = 1$
$2 \% 3 = 2$
$3 \% 3 = 0$ ✗
$4 \% 3 = 1$
$5 \% 3 = 2$
$6 \% 3 = 0$ ✗

a
aa
aaa

February 4th 2025
Detroit Michigan.

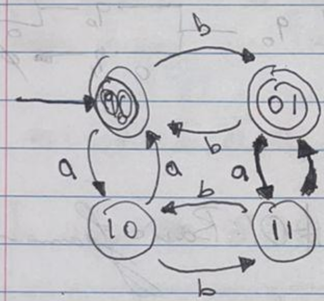
Assignment II : Construct dfa

Construct deterministic finite acceptor (dfa) for the following

I. For $\Sigma = \{a, b\}$, construct dfa's that accept the sets consisting of all strings with an even number of a's.

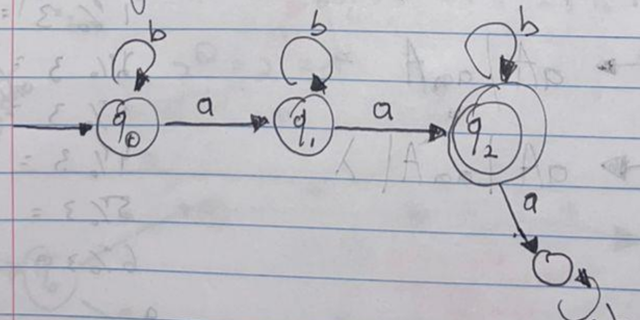
b b b
AaAaA

aabaababa



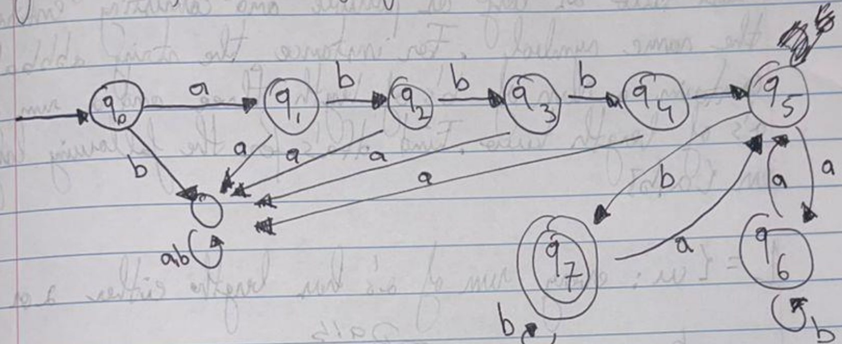
II. For $\Sigma = \{a, b\}$, construct dfa's that accept the sets consisting of all strings with at least one b and exactly two a's.

b b b
AaAaA



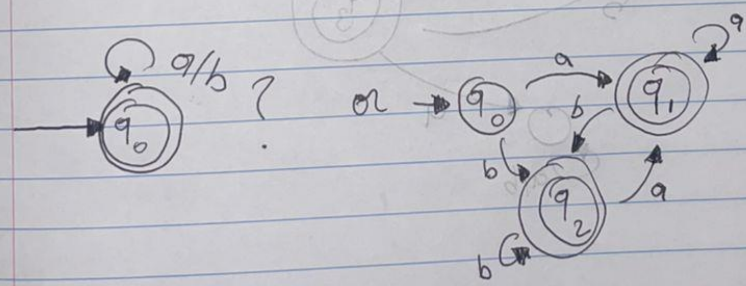
CEIT

III. Give dfa's for the languages $L = \{ab^4wb^2 : w \in \{a,b\}^*\}$



IV. Find dfa's for the following languages on $\Sigma = \{a, b\}$

$L = \{w : |w| \bmod 3 \neq 0\}$



V. A run in a string is a substring of a length at least two as long as possible and consisting entirely of the same symbol. For instance, the string abbbab contains a run of b's of length three and a run of a's of length two. Find dfa's for the following languages on {a,b}

L = {w : every run of a's has length either 2 or three}