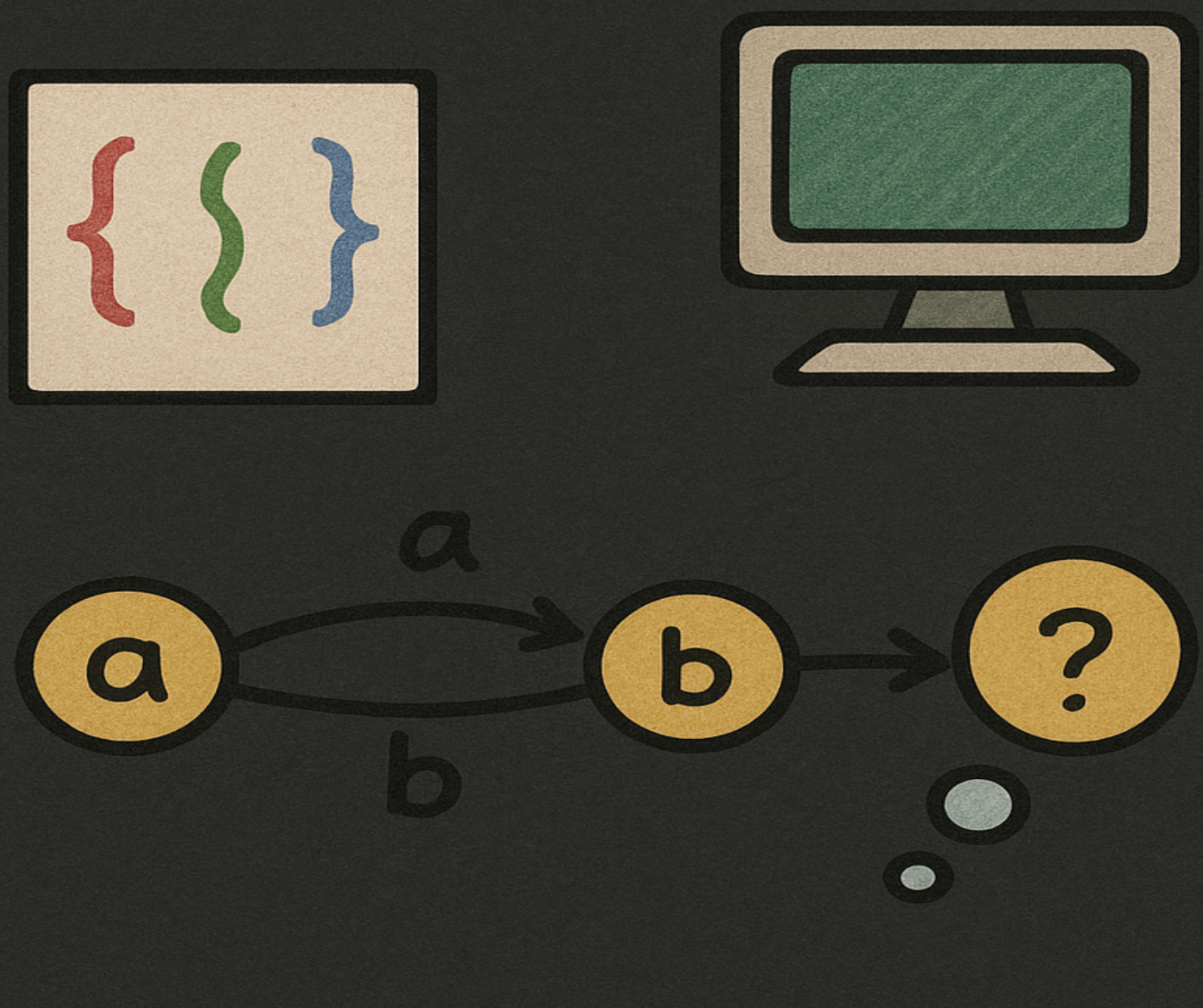
CSE 331

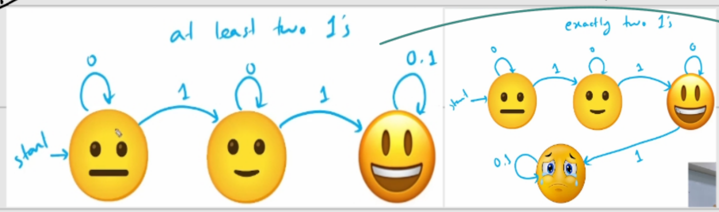
AUTOMATA AND COMPUTABILITY



[Farhan Feroz Sir YT]

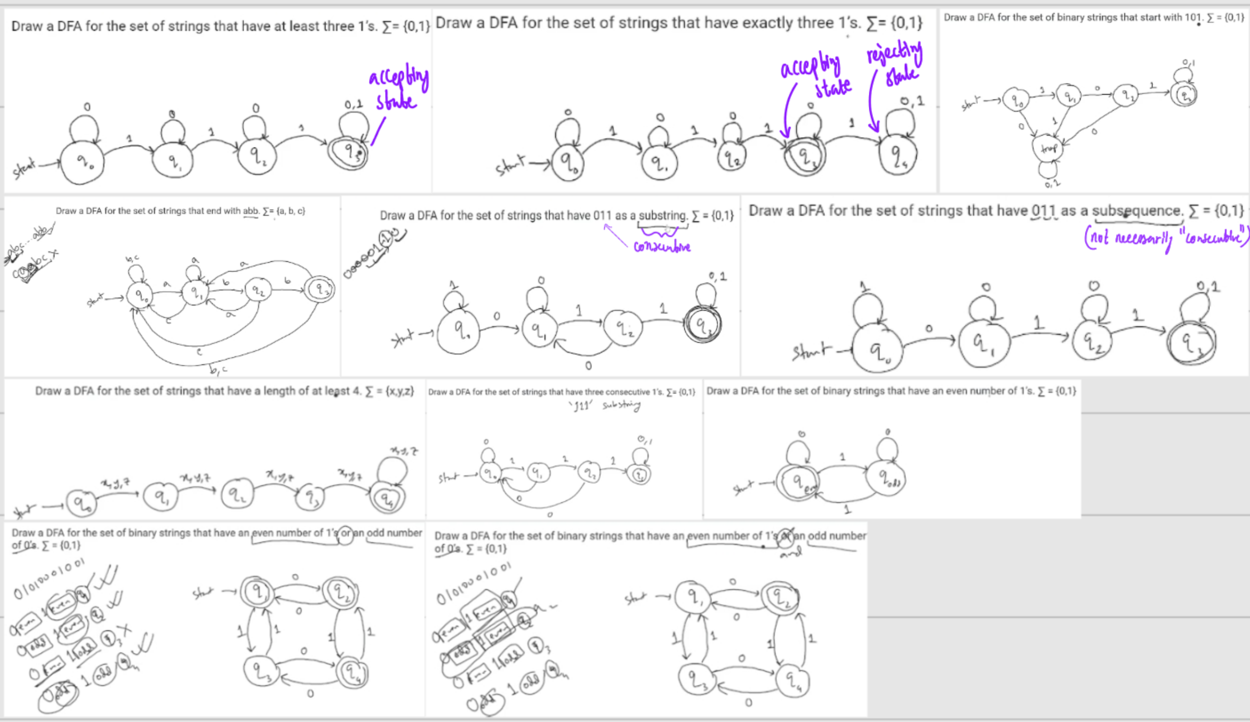
Introduction to DFA → Deterministic Finite Automata

21/02/2025
Friday



It is a language of a set of string containing at least two 1s

DFA | Example

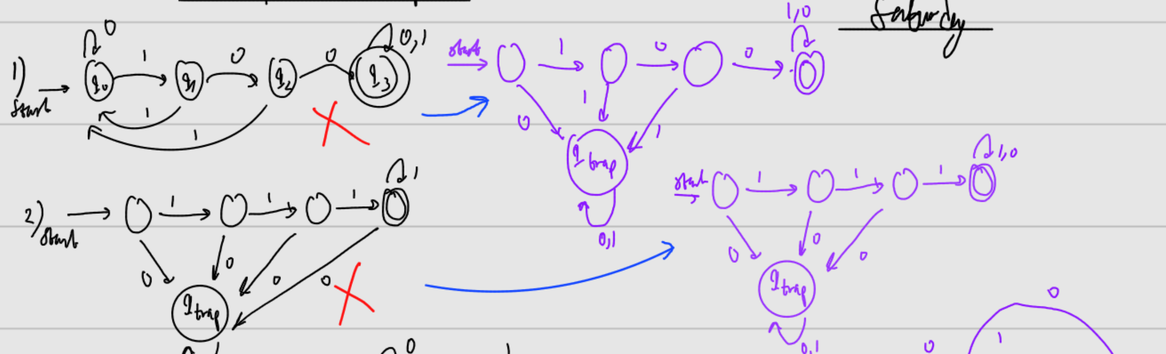


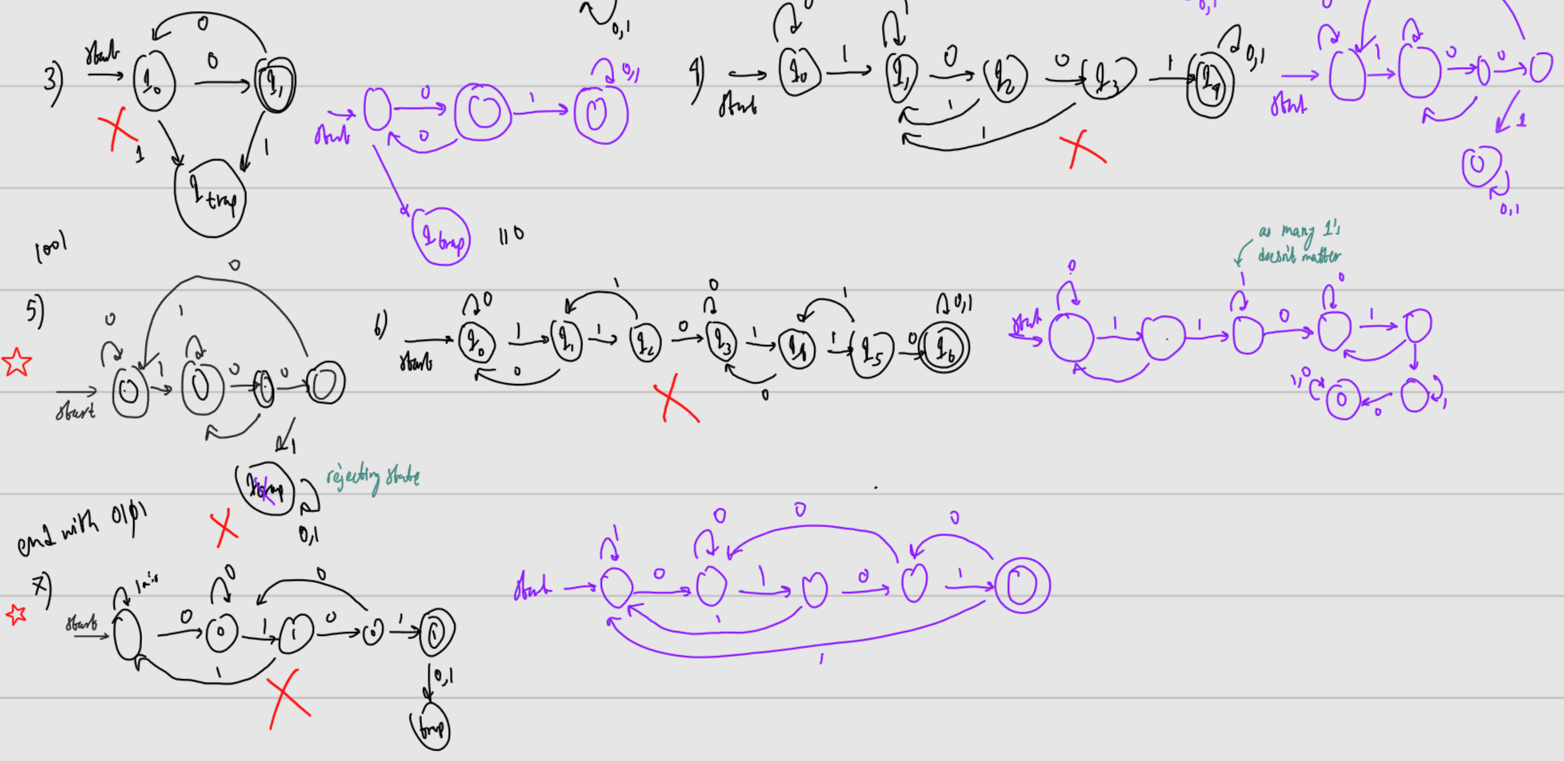
DFA | More Examples

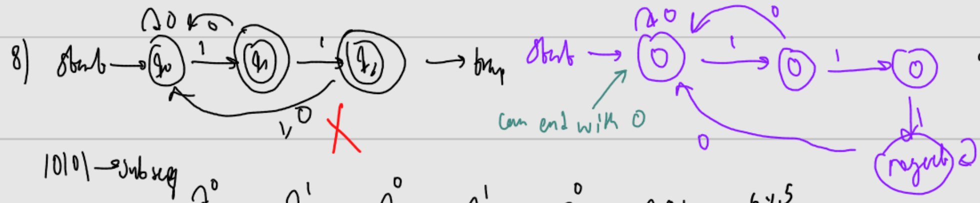
22/02/2025
Saturday

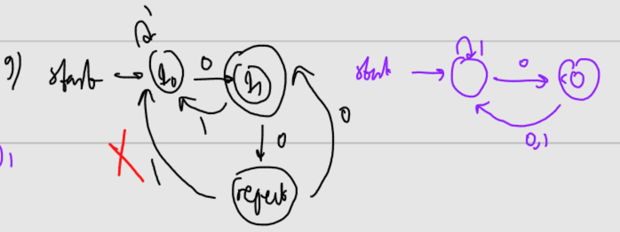
Draw DFAs for the following languages. Assume Σ = {0, 1}

1. Strings that start with 100.
2. Strings that start with at least three 1s.
3. Strings that start with odd number of 0s.
4. Strings that contain 1001 as a substring.
5. Strings that do not contain 1001 as a substring.
6. Strings that have at least two occurrences of 110.
7. Strings that end with 0101.
8. Strings that end with at most two 1s.
9. Strings that end with odd number of 0s.
10. Strings that have 10101 as a subsequence.
11. Strings that do not have 10101 as a subsequence.
12. Binary strings whose decimal equivalent is divisible by 6.
13. Binary strings which has a 0 at its third position.
14. Binary strings which has a 1 at its second last position.
15. Binary strings whose decimal equivalent is divisible by 8(You are not allowed to use more than 4 states).

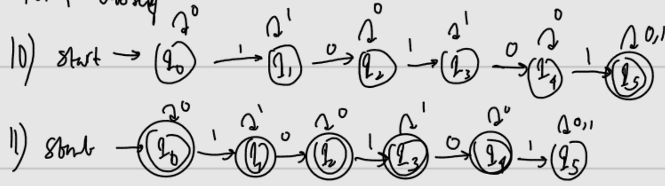


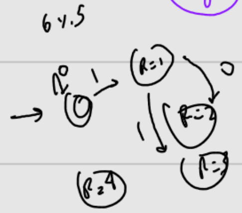


8) 

9) 

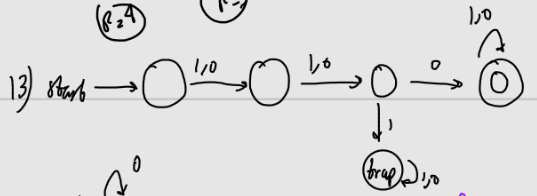
1010 → subseq

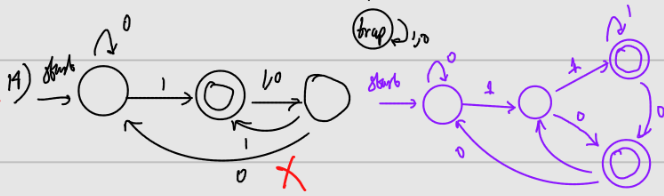
10) 11) 

6 % 5 

12) ☆

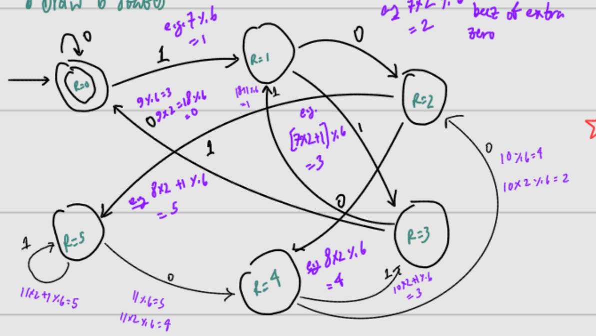
110 → 6,	1100 → 12,	1101
☆☆☆	adding zero doubles the decimal value	means (6×2)+1 adding one

13) 

☆ 14) 

[Binary string whose decimal equivalent is divisible by 6]

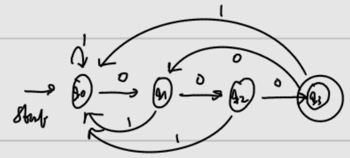
• Draw 6 states



☆ 15) [Decimal equivalent is divisible by 8 (max 4 states)]

110	1100	11000
divisible by 2	divisible by 4	divisible by 8

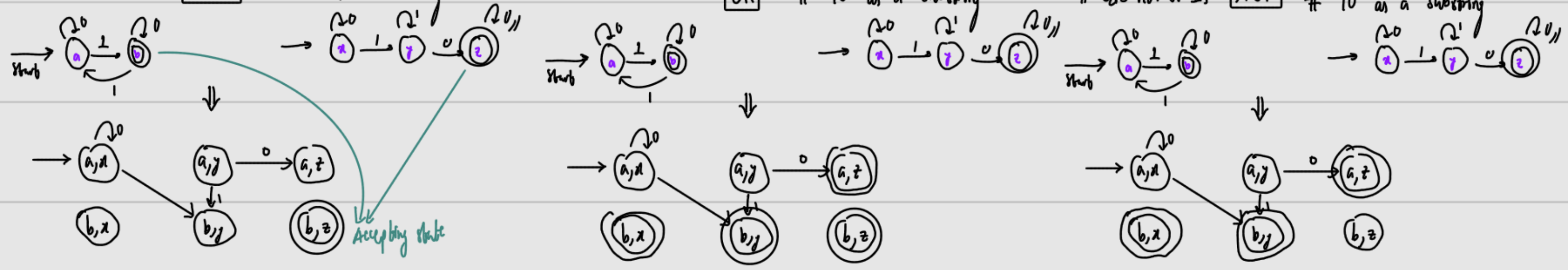
→ basically ends with "000"



07/03/2025
Friday

Cross Product → For multiple conditions

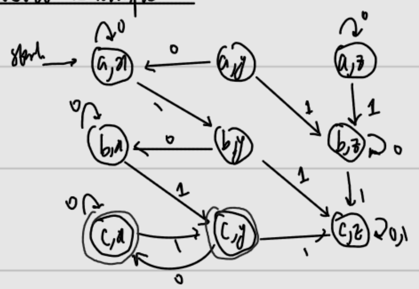
odd no. of 1's AND # "10" as a substring | # odd no. of 1's OR # "10" as a substring | # odd no. of 1's XOR # "10" as a substring



Cross Product Example

first do this then find no

at least two 1s but no consecutive 1s



Cross Product (Example of Implication)

If it has even no. of 0's, then it'll have odd no. of 1's
p q

P	q	P → q
0	0	1
0	1	1
1	0	0
1	1	1

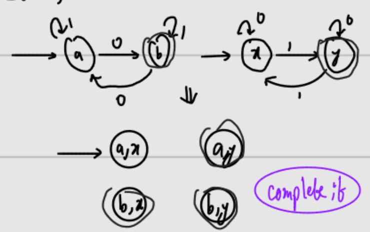
¬P v q
Logically equivalent

hypothesis T but Conclusion F
∴ P → q F

(Not same)
Parity of 1 & 0 are different

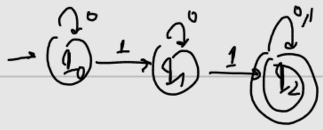
It has odd no. of 0s OR odd no. of 1's

It has odd no. of 0s   odd no. of 1s



Regular Expression (Basics)

08/03/2025
Saturday

→ A Language is a set of strings e.g. at least two 1s  — A machine → DFA

"Regular Languages": Languages for which it is possible to draw DFA.

Operations on numbers: e.g. $(5+3)\times 4^2$, Operations on Languages: "set"

$\boxed{\text{Regular Language} \xrightarrow{\text{Regular Operations}} \text{Regular expression}}$

Regular Operations:

1) Union/OR (|) — or "+"
2) Concatenation (·)
3) Kleene Closure/Star Operation (*)

e.g. $L_1 = \{0,1\}$, $L_2 = \{00,11\}$

$L_1 | L_2 = \{0,1,00,11\}$

$L_1 \cdot L_2 = \{000, 011, 100, 111\}$

$(0|1)\cdot(0|1) \rightarrow L_1 \cdot L_1 = \{00, 01, 10, 11\}$

e.g. $(0|1).1$
$= L = \{01, 11\}$

e.g. $0^* = 0$'s for any no. of times (Not "any no. of times")
$= \{\epsilon, 0, 00, 000, 0000, 00000 \ldots\}$ — no zero (empty)

regular expressions:
$01^* = \{0, 01, 011, 0111, \ldots\}$
$(01)^* = \{\epsilon, 01, 0101, 010101, \ldots\}$

$\Sigma^* = (0|1)^*$

$(0|1)^0 = \{\epsilon\}$

$(0|1)^1 = \{0,1\}$

$(0|1)\cdot(0|1) = \{00, 01, 10, 11\}$

$(0|1)\cdot(0|1)\cdot(0|1) = \{000, 001, 010, 011, 100, 101, 110, 111\}$

$\therefore \Sigma^* = (0|1)^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}$

☆ ↳ all possible binary strings

e.g. 1) Strings starting with "01"
$= 01\Sigma^*$ $= \{01, 010, 011, 0100, 0100\}$
↳ meaning anything after 01

2) strings ending with "01"
$= \Sigma^*01$

3) Strings having "100" as a substring
$= \Sigma^*100\Sigma^*$

4) Strings having "100" as a subsequence
$= \Sigma^*1\Sigma^*0\Sigma^*0\Sigma^*$
OR $0^*11^*01^*0\Sigma^*$ ☆ starting from the first "1"
OR $\Sigma^*10^*01^*01^*$ ← Ending with last "0"

5) Strings having even no. of 1's
• no. of 0's doesn't matter

$= $ eg $\underbrace{0^*1\,0^*1}_{\text{even}}\ \underbrace{0^*1\,0^*1}_{\text{Repeats}}\ 0^*1\,0^*1\,0^*1\,0^*1\,0^* = (0^*10^*1)^*0^*$

OR $0^*1\,0^*1\,0^*1\,0^*1\,0^*1\,0^*1\,0^*1\,0^*1\,0^* = 0^*(10^*10^*)^*$ — repeats

6) String having 0's at even positions.

_ 0 _ 0 _ 0 _ 0 ☆

$(1^*0)^*1^*$ [check]

$\boxed{0^+ \rightarrow \text{Positive Kleene Closure: At least one zero}}$

Regular Expressions Examples

1) Strings that end in three consecutive 1's
$= (0|1)^*111 \ / \ \Sigma^*111$

2) Strings that have at least one 1
$= \Sigma^*1\Sigma^*$ or $0^*1\Sigma^*$ or Σ^*10^* (1st '1', last '1')

3) Strings that have at most one 1
$= 0^*10^*$ ✗ $\quad 0^*10^* | 0^*$ meaning it can have no 1 $\quad$ OR $0^*(1|\epsilon)0^*$

4) Strings that contain the substring "101"
$= \Sigma^*101\Sigma^*$

5) Strings that do not have consecutive 1's
$= 0^*(10^+)^*(1|\epsilon)$ or $(0|10^+)^*(1|\epsilon)$

6) Strings that do not contain the substring "11" (same)
$= 0^*(10^+)^*(1|\epsilon)$ or $(0|10^+)^*(1|\epsilon)$

7) Strings that have neither consecutive 1s, nor consecutive 0s

If starts with 01

= eg 10101010.. $(10)^*(1|\epsilon) \mid (01)^*(0|\epsilon)$ or $(0|\epsilon)(10)^*(1|\epsilon)$ //

8) Strings that may have consecutive 1s, or consecutive 0s but not both.

or meaning not both

$= \underbrace{(0|10)^*(1|\epsilon)}_{\text{no consecutive 1}} \mid \underbrace{(1|01)^*(0|\epsilon)}_{\text{no consecutive 0}}$

no consecutive 1 but can have consecutive 0s

no consecutive 0 but can have consecutive 1s

9) String in which the no. of 0s is even

$= (1^*01^*01^*)^*$

10) Strings in which the no. of 0s is odd

$= (1^*01^*01^*)^*1^*01^*$

11) Strings in which the no. of 0s is divisible by four

$= (1|01^*01^*01^*0)^*$ ✗ two term

Regular Expression | Assignment_1 sol^n | Part-01

Q1: Write down regular expressions for each of the following languages. Assume $\Sigma = \{0, 1\}$ [6 marks]

- The language containing strings where 0s and 1s are alternate
- The language containing strings which starts and ends with same character.
- The language containing strings in which the number of 1s between every pair of consecutive 0s is odd.

1) (a) eg 010101.. or 1010...

1|ε 0|ε

$= 1? \mid (01)^* \mid 0? \,/\, 01(10)^* \mid 1$ //

(b) $0\Sigma^*0 \mid 1\Sigma^*1 \mid 0 \mid 1$

(c) eg 0111011110 . .

no 0s

$= 1^*(0\underbrace{(11)^*}_{\text{odd}}1)^*01^* \mid 1^*$

Regular Expression | Assignment_1 sol^n | Part-02

Q2: Write a regular expression for the language $L_1 \cap L_2$. [4 marks]

Let $\Sigma = \{0, 1\}$. Consider the following pair of languages over Σ.

$$L_1 = \{w : \text{the length of } w \text{ is divisible by three}\}$$

$$L_2 = \{w : \text{every second letter in } w \text{ is a } 0\}$$

2) $L_1 = (\Sigma\Sigma\Sigma)^*$ any character

$L_2 = (\Sigma 0)^*(\Sigma|\epsilon)$

$\Sigma 0 \Sigma$

$L_1 \cap L_2 = (\underbrace{\Sigma 0 \Sigma 0 \Sigma 0}_{6})^*(\underbrace{\Sigma 0 \Sigma}_{3} \mid \epsilon)$ //

$3 = 9 \rightarrow$ Not multiple of 6 but is of 3

Q3: [3 marks]

Write down a regular expression that generates the following language.

$L = \{w \in \{0, 1, 2\}^* : \text{the last letter of } w \text{ appears at least twice in } w\}$

$= \Sigma^*0\Sigma^*0 \mid \Sigma^*1\Sigma^*1 \mid \Sigma^*2\Sigma^*2$ //

Q4: [3 marks]

Write down a regular expression that generates the following language.

$L = \{w \in \{0, 1, 2\}^* : w \text{ contains at least one 1 and one 0}\}$

$= \Sigma^*1\Sigma^*0\Sigma^* \mid \Sigma^*0\Sigma^*1\Sigma^*$

Q5: [4 marks]

Write down a regular expression that generates the following language.

$L = \{w \in \{0, 1\}^* : \text{exactly one occurrence of 00 appears in } w\}$

$= \Sigma^*00\Sigma^*$ ✗

$1^*(01^+)^*00(1^+0)^*1^*$ //

Regular Expression | Summer 2023 Mid Term Practice

09/03/2025

Sunday

- The language containing strings where 0s and 1s always appear in pairs. For example - 001100, 1100110011001100 etc.
- The language containing strings which starts and ends with diffrent characters.
- The language containing strings in which the number of 1s between every pair of consecutive 0s is even.
- The language containing strings having equal number of "01" and "10" substrings
- The language containing strings whose parity of 0 and 1 are different.
- The language containing strings where every 0 is followed by at least three 1's.

(a) $(00|11)^*$ ✓ $(00)^*|(11)^*$ ✗

(b) $0\Sigma^*1 \mid 1\Sigma^*0$

(c) $1^*(0(11)^*)^*1^*$

(d) $0\Sigma^*0 \mid 1\Sigma^*1 \mid 0 \mid 1$

(same as "starting & ending with same chr)

(e) $(\Sigma\Sigma)^*\Sigma$ $(\Sigma\Sigma)^*$

For diff parity For same parity

(f) $1^*(01111^*)^*$

Different parity	Same parity
0 even, 1 odd	0 even, 1 even
1 even, 0 odd	0 odd, 1 odd

Length is always odd

Length is always even

8) 0 followed by two or less 1's

$= \Sigma^*0(\epsilon|1|11)(\epsilon|0\Sigma^*)$

Regular Expressions | More Examples

L_1 The set of strings that end with 11.

$\Sigma^* 11$ $\quad (0|1)^* 11$

$\bar{L}_1$ The set of strings that do not end with 11.

$\Sigma^*(00|01|10) \,|\, (0|1|\epsilon)$

for length ≥ 2 $\quad$ for length ≤ 1

The set of strings that contain 01 as a substring.

$\Sigma^* 01 \Sigma^*$

The set of strings that do not contain 01 as a substring.

$1^* 0^*$

The set of strings having 0 at every odd position.

$(0(0|1))^*(0|\epsilon)$ $\quad 0\,(0|1)\,0\,(0|1)\,0 \ldots 0$ $\quad (00|01)^*(0|\epsilon)$

The set of strings having 0 at every third position.

$(\Sigma\Sigma 0)^*(0|1|\epsilon)(0|1|\epsilon)$ $\quad$ _ _ 0 _ _ 0 _ _

The set of strings having 0 at the third position.

$\Sigma^2 0 \Sigma^*$ $\quad (0|1)(0|1)0\Sigma^*$

The set of strings having 0 at the third last position.

$\Sigma^* 0 \Sigma^2$

Regular Expression | Problem Solving | Complement Operation of Language

CSE331 Quiz 2 Set A
Marks: 20 Time: 25 minutes
Semester: Fall 2024
Name: Section: ID:

Question 1: [20 Points]

Assume, $\Sigma = \{0, 1\}$

$L_1 = \{w \mid \text{the length of } w \text{ is even}\}$

$L_2 = \{w \mid w \text{ starts and ends with different characters}\}$

$L_3 = \{w \mid w \text{ have 0's at all odd positions}\}$

a) *Write a RegEx for the language L_1*

b) *Write a RegEx for the language L_2* "L_2 difference L_1"

c) *Write a RegEx for the language L_3*

d) *Write a RegEx for the language $L_2 \setminus L_1$*

e) *Write a RegEx for the language $L_3 \setminus L_2$*

(a) $(\Sigma\Sigma)^* \rightarrow L_1$

(b) $0\Sigma^*1 \mid 1\Sigma^*0 \rightarrow L_2$

(c) $\underline{0}\ \underline{\Sigma}\ \underline{0}\ \underline{\Sigma}$ _ . . $\quad L_3$

$(0\Sigma)^*(0|\epsilon)$ → or we can write 0?

(d) $L_2 \setminus L_1$ same as $L_2 \cap \bar{L}_1$

$\{w \mid w$ starts and ends with diff chr & w is odd$\}$

$= 0(\Sigma\Sigma)^*\Sigma 1 \mid 1(\Sigma\Sigma)^*\Sigma 0$ //

(e) $L_3 \setminus L_2$: 0s at all odd position & starts and ends with same chr

$= \underline{0}\ \_\ \underline{0}\ \_\ \underline{0}$ ∴ starts with 0

$\underbrace{(0\Sigma)^*0}_{\text{odd length}} \mid \underbrace{(0\Sigma)^*00}_{\text{even length}}$ // — ends with 0

OR $(0\Sigma)^*(0|00)$ //

1) Assume, $\Sigma = \{0, 1\}$.
Write the regular expressions for the following languages and their complements.

a. $L_1 = \{w \mid \text{the length of } w \text{ is divisible by } 4\}$
b. $L_2 = \{w \mid w \text{ has 0's at all even positions}\}$
c. $L_3 = \{w \mid w \text{ has 1's at all odd positions}\}$
d. $L_4 = \{w \mid w \text{ has a 1 at every third position}\}$

2) Assume, $\Sigma = \{0, 1, 2\}$.
Write the regular expressions for the following languages and their complements.

a. $L_5 = \{w \mid w \text{ has the same first and last character}\}$
b. $L_6 = \{w \mid w \text{ starts with a 0 but does not end with a 0}\}$

1)(a) $L_1: (\Sigma\Sigma\Sigma\Sigma)^*$, $\bar{L}_1 = (\Sigma\Sigma\Sigma\Sigma)^*(\Sigma|\Sigma^2|\Sigma^3)$ //

(b) $L_2: (\Sigma 0)^*\Sigma?$, $\bar{L}_2: (\Sigma\Sigma)^*\Sigma 1 \Sigma^*$ or $(\Sigma 0)^*\Sigma 1 \Sigma^*$ //

(c) $L_3: (1\Sigma)^*1?$, $\bar{L}_3: (\Sigma\Sigma)^*0\Sigma^*$ //

(d) $L_4: (\Sigma\Sigma 1)^*\Sigma?\Sigma?$, $\bar{L}_4: (\Sigma\Sigma\Sigma)^*\Sigma\Sigma 0\Sigma^*$

2)(a) $L_5: 0\Sigma^*0 \mid 1\Sigma^*1 \mid 2\Sigma^*2 \mid \Sigma$ → $0|1|2$ //

$\bar{L}_5: 0\Sigma^*(1|2) \mid 1\Sigma^*(0|2) \mid 2\Sigma^*(0|1)$ //

$L_6: 0\Sigma^*(1|2) \mid \cancel{1|2}$ $\quad 0\Sigma^*(1|2)$ //

$\bar{L}_6$: $\boxed{\neg(p \wedge q) \equiv \neg p \vee \neg q}$

$(1|2)\Sigma^* \mid \Sigma^*0$ //

3) Consider the following languages over $\Sigma = \{0, 1\}$.

$L_1 = \{w \mid w \text{ does not contain } 11\}$

$L_2 = \{w \mid \text{every 1 in } w \text{ is followed by at least one } 0\}$

$L_3 = \{w \mid \text{the number of times 1 appears in } w \text{ is even}\}$

Now solve the following problems.

a) Give a regular expression for the language L_1.

b) Your friend claims that $L_1 = L_2$. Prove her wrong by writing down a five-letter string in $L_1 \setminus L_2$. Recall that $L_1 \setminus L_2$ contains all strings in L_1 but not in L_2.

c) Give a regular expression for the language $L_1 \setminus L_2$.

d) Give a regular expression for the language L_3.

e) Give a regular expression for the language $L_2 \setminus L_3$.

(a) L_1: $\cancel{\Sigma^*1\Sigma^*1\Sigma^*}$ $\quad (0|10)^*1?$ or $0^*(10^+)^*1?$

(b) $L_2: 0^*(10^+)^*$

∴ five letter string in $L_1 \setminus L_2$: "00001"

OR "10101" $\in L_1$, $\notin L_2$

ϵ is missing

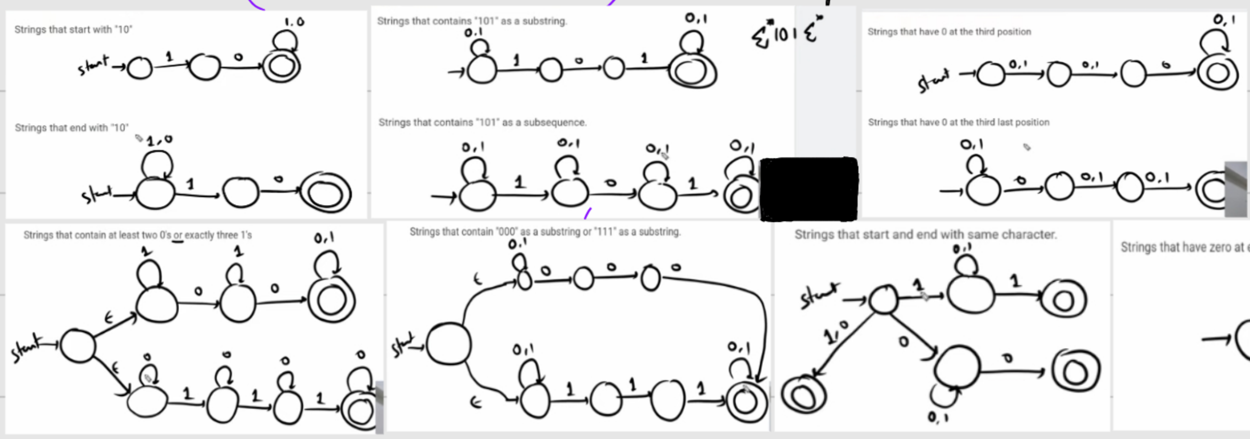
(c) $L_1 \setminus L_2$: $0^*(10^+)^*1$

$\boxed{0^* \text{ has } \epsilon \text{ in it}}$

(d) L_3: $(0|\epsilon)^*(10^*1)^*(0|\epsilon)^*$ $\quad (0^*10^*1)^*0^*$ //

(e) $L_2 \setminus L_3$: $\cancel{0^*(10^+)^*1\,1\,0^*}$ $\quad 0^*(10^+10^+)^*10^+$ //

(Non-Deterministic Finite Automata) NFA | Examples



Assignment_2 Solution

A) Draw NFAs for the following languages. Assume $\Sigma = \{0, 1\}$ [5

1. Strings that start with at least two 1's.
2. Strings that end with even number of 0's.
3. Strings that have even number of 0's or odd number of 1's.

B) Write regular expressions for the following languages. Assume $\Sigma = \{a, b, c, d, e, f\}$ [5

1. Strings that start with a vowel.
2. Strings that have at least 4 vowels.
3. Strings of length at least 5.
4. Strings that end with a vowel followed by two consonants.
5. Strings that do not have "bad" as a substring.
6. Strings that have "deaf" as a subsequence.

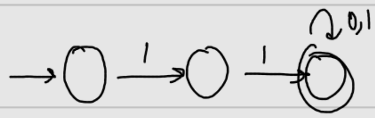
C) Convert the following regular expression into NFAs. Assume $\Sigma = \{a, b, c\}$ [5

$$(a^*b^* + (ac + b^*c)b)^*$$

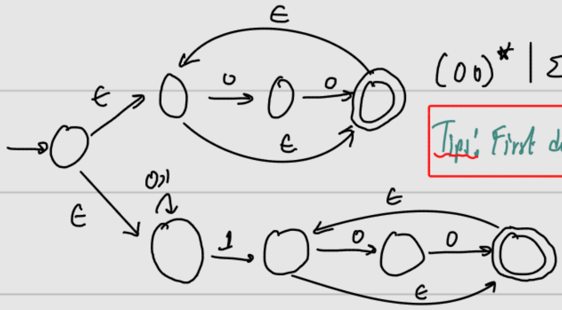
D) Convert the following regular expression into NFAs. Assume $\Sigma = \{a, b, c\}$ [5

$$ab^*c \cup ((ab \cup bc)^* \cup (a \cup b)(a \cup b)^*)\, bc \cup a(bc)^*$$

(A) 1)



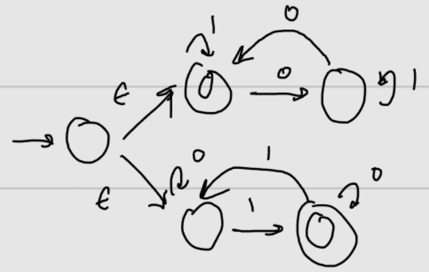
2) ☆



$(00)^* \mid \Sigma^* 1 (00)^*$

Tip: First do RE, then do NFA

3)



(B) 1) $(a|e)\Sigma^*$ //

2) $\Sigma^*(a|e)\Sigma^*(a|e)\Sigma^*(a|e)\Sigma^*(a|e)\Sigma^*$ //

3) $\Sigma^5 \Sigma^*$ //

4) $\Sigma^*(a|e)(b|c|d|f)(b|c|d|f)$ //

or $\gamma(\text{vowels}) = \{a, e\}$, $\delta(\text{consonants}) = \{b, c, d, f\}$

$\therefore \Sigma^* \gamma \delta^2$ //

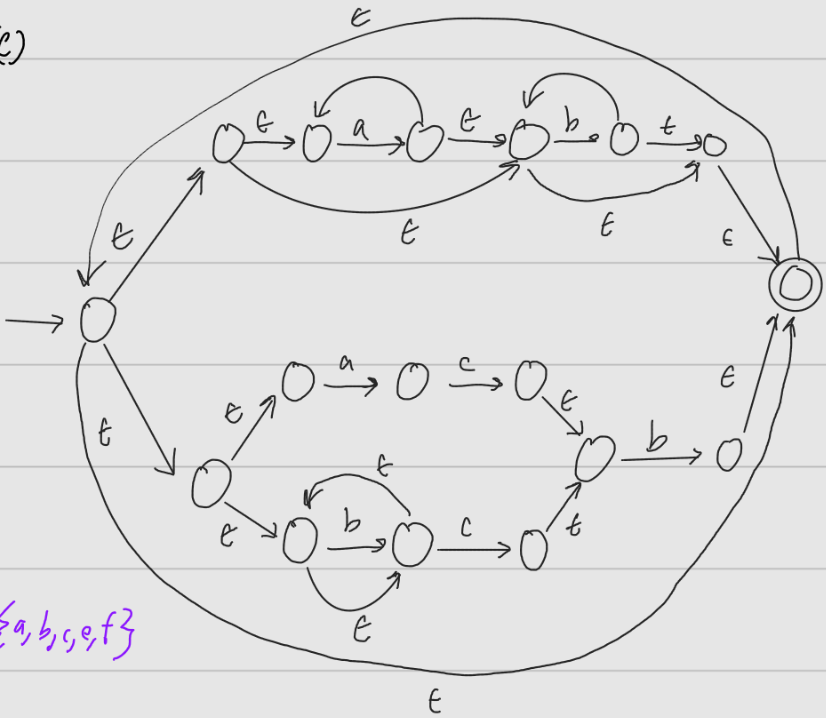
5) ☆ $\Sigma^* b \Sigma^* a \Sigma^* d \Sigma^*$ ✗

$x = \{a, c, d, e, f\}$, $y = \{c, d, e, f, b\}$, $z = \{a, b, c, e, f\}$

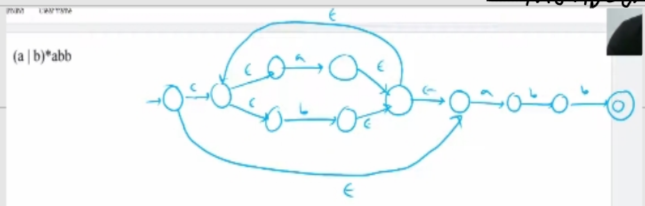
$(x \mid by \mid baz)^*$ //

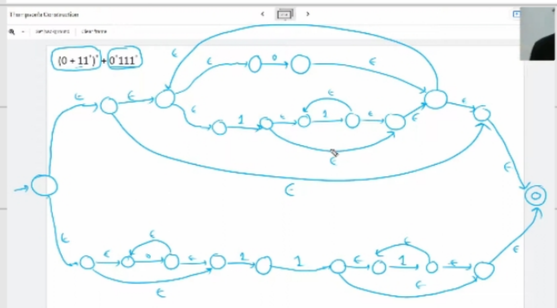
6) $\Sigma^* d \Sigma^* e \Sigma^* a \Sigma^* f \Sigma^*$ //

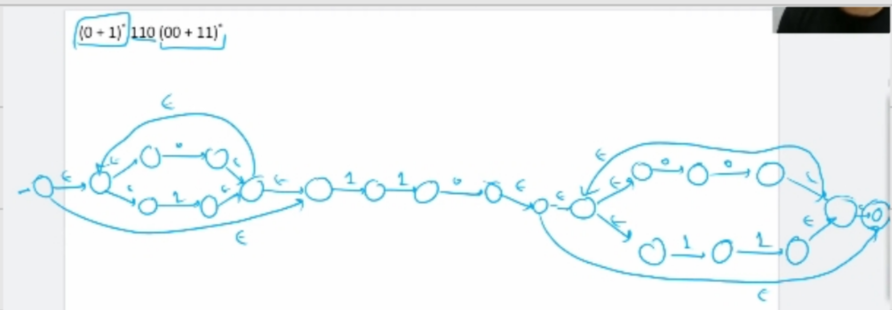
(C)



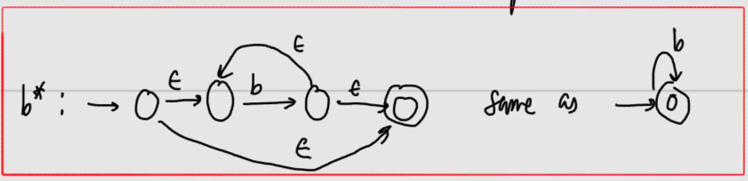
Thompson's Construction | RE to NFA



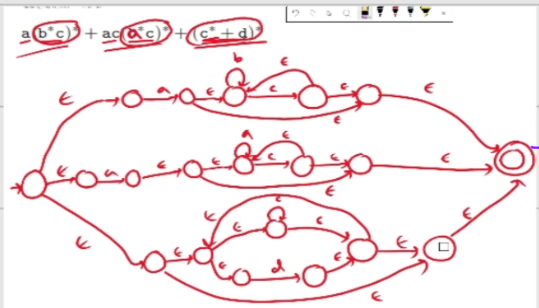




Thompson's Construction | Optimization Tips



- Use pencil
- Start drawing from the middle of the page



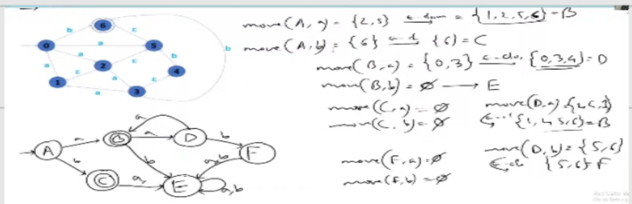
→ Watch the recording

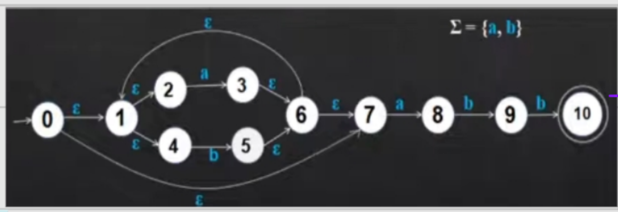
Subset Construction | NFA to DFA

In NFA is 0

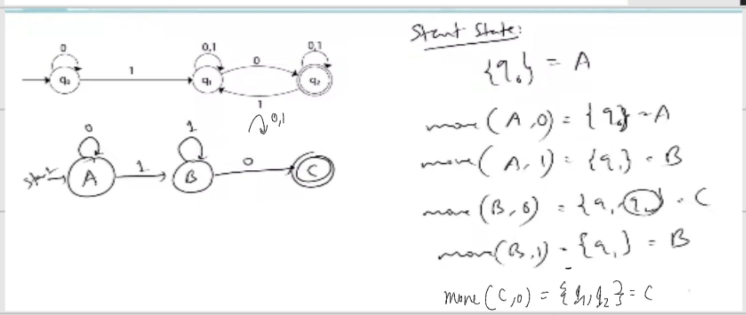
$\Sigma = \{a, b\}$ subset Construction

start state: ϵ-closure $(0) = \{0\} = A$





→ Sol^n in Box



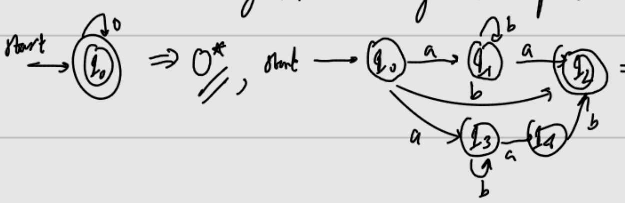
move $(C, 0) = \{q_1, q_2\} = C$

move $(C, 1) = \{q_1, q_2\} = C$

DFA to RegEx

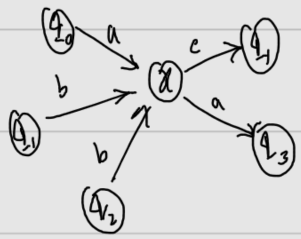
State Elimination | How to think

→ DFA will be given. Then RegEx to express the language of DFA



⇒ RegEx: $ab^*a \mid b \mid ab^*ab$

"State Elimination"



Incoming ↘ ↙ outgoing

Problem: $3 \times 2 = 6$

Compensation

$q_0 \to q_1$: a c

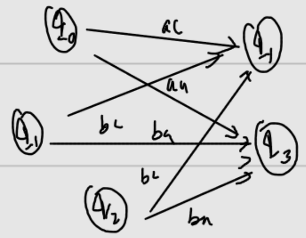
$q_0 \to q_3$: a a

$q_1 \to q_1$: b c

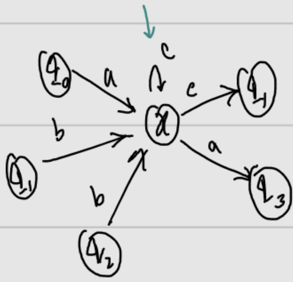
$q_1 \to q_3$: b a

$q_2 \to q_1$: b c

$q_2 \to q_3$: b a



self transition



Problem: $3 \times 2 = 6$

counts remain same

Compensation ← due to self transition

$q_0 \to q_1$: ac^*c

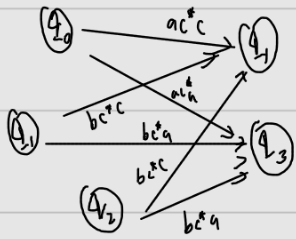
$q_0 \to q_3$: ac^*a

$q_1 \to q_1$: bc^*c

$q_1 \to q_3$: bc^*a

$q_2 \to q_1$: bc^*c

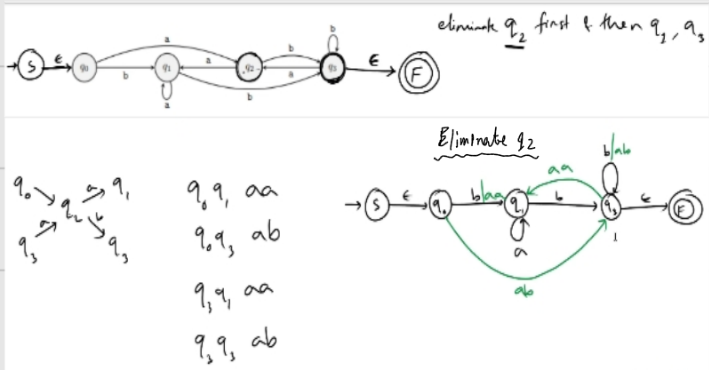
$q_2 \to q_3$: bc^*a

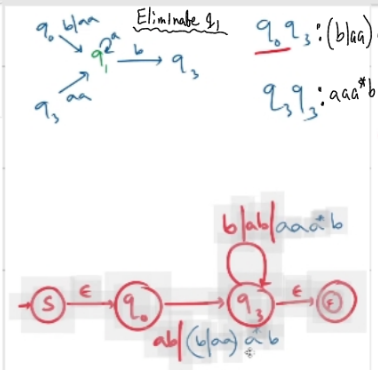


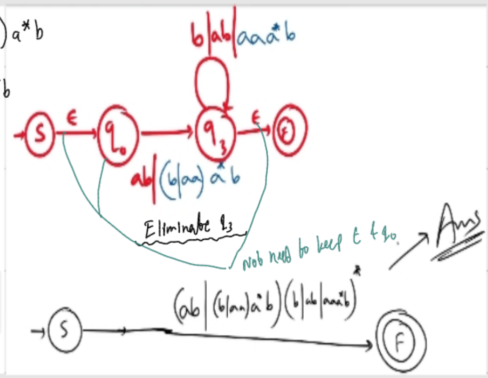
State Elimination | Example (Step by Step)

Sequence will be given in the question that which state to eliminate first

- Make a new starting & accepting state with ε transition — meaning no transition
- For multiple accepting case, ε transition must be from all the accepting states.
- If there are two self loops, merge them into one, separated by 'OR' (|)

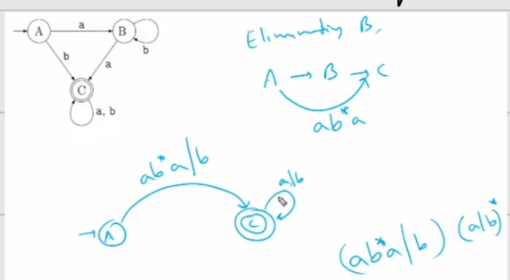


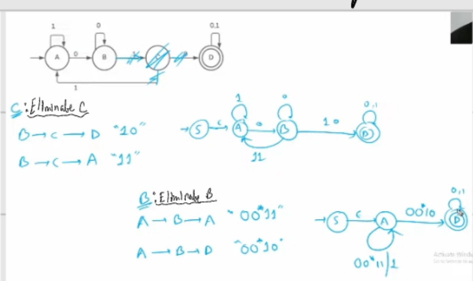




State Elimination Method | DFA to Regular Expression

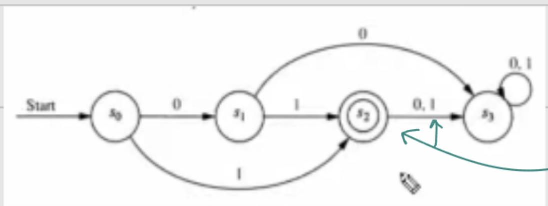
*** If there's no incoming arrow on a state then it's starting state And no outgoing on a state then it's Accepting State. [So, no need to create starting & Accepting state]





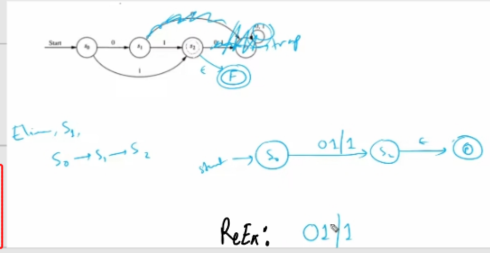
Eliminate A

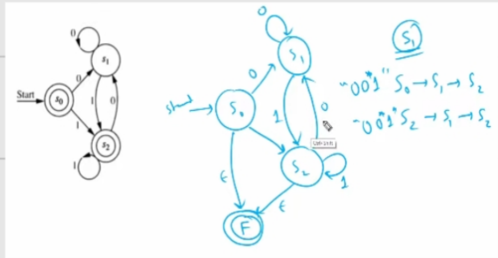
$(00^*11|1)^* 00^*10(0|1)^*$

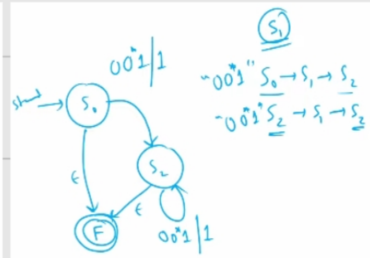


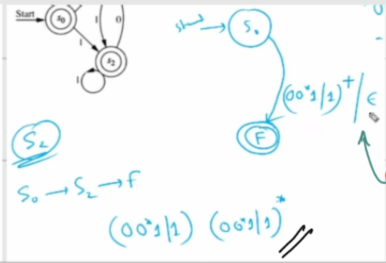
Trap state has no outgoing arrow and It is Not an accepting state.

Accepting state has outgoing arrow. So we need to create a new accepting state

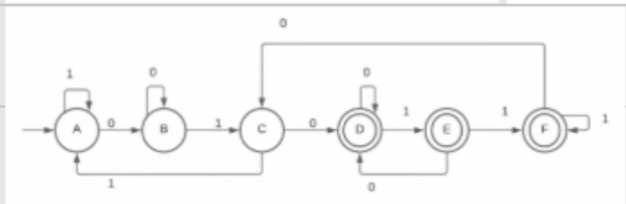








here ε is must



Quiz 1 Sol^n | DFA and RE

Quiz 1 Section 1
Total marks: 10
Duration: 30 minutes

Student ID: ______________

You have to use the designated spaces for your answers. No extra pages will be provided.

Problem 1: Regular Languages and DFAs (10 points)

Let $\Sigma = \{0, 1\}$.

$$L_1 = \{w \in \Sigma^* : w \text{ starts with odd number of 1's}\}$$
$$L_2 = \{w \in \Sigma^* : w \text{ starts and ends with same character}\}$$

(a) Write down all strings in L_2 which are of length 3. (2 points)

(b) Give the state diagram for a DFA that recognizes L_1. (3 points)

(c) Give the state diagram for a DFA that recognizes L_2. (3 points)

(d) Give the state diagram for a DFA that recognizes $L_1 \cap L_2$. (2 points)

Quiz 1 Section 2
Total marks: 10
Duration: 30 minutes

Student ID: ______________

You have to use the designated spaces for your answers. No extra pages will be provided.

Problem 1: Regular Languages and DFAs (10 points)

Let $\Sigma = \{0, 1\}$.

$$L_1 = \{w \in \Sigma^* : w \text{ starts with at least three 0's}\}$$
$$L_2 = \{w \in \Sigma^* : w \text{ starts and ends with different character}\}$$

(a) Write down all strings in L_2 which are of length 3. (2 points)

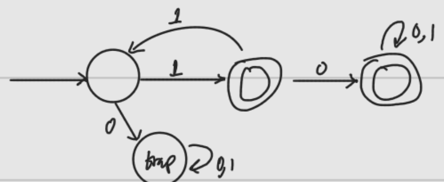
(b) Give the state diagram for a DFA that recognizes L_1. (3 points)

(c) Give the state diagram for a DFA that recognizes L_2. (3 points)

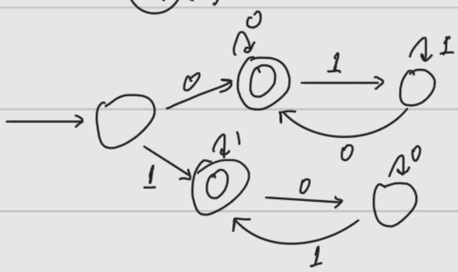
(d) Give the state diagram for a DFA that recognizes $L_1 \cap L_2$. (2 points)

(a) 000, 010, 101, 111

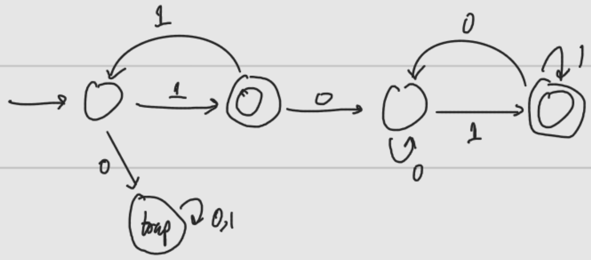
(b)



(c)

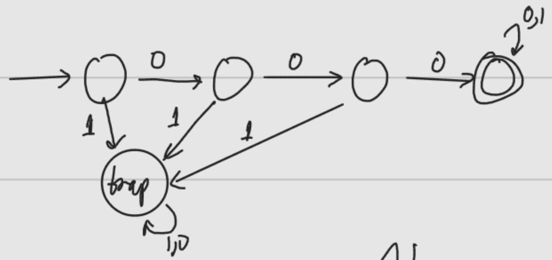


(d) $L_1 \cap L_2$: starts with odd no. of 1's and ends with 1

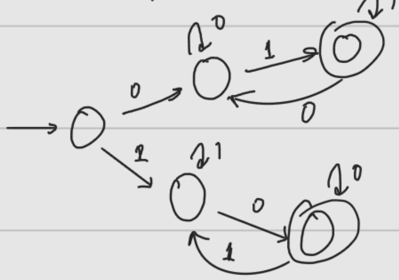


(a) 001, 011, 100, 110

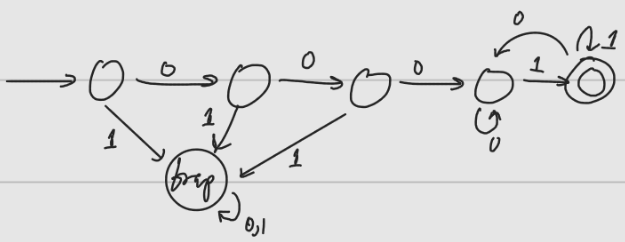
(b)



(c)



(d) $L_1 \cap L_2$: starts with 3 0's and ends with 1



Spring 2022 Mid Term Question Solved

Total marks: 40
Duration: 70 minutes CSE33

Solve problems 1 through 4. Problem 5 is optional.

Problem 1: Regular Expressions (10 points)
Write down regular expressions for each of the following languages. Assume that $\Sigma = \{0, 1\}$.

(a) The language containing strings where 0s and 1s alternate. (3 points)

(b) The language containing strings in which the number of 1s is divisible by 4. (3 points)

(c) The language containing strings in which the number of 0s between every pair of consecutive 1s is even. (4 points)

Problem 2: Constructing a DFA (10 points)
Consider the following language.

$$L = \{w \in \{0,1\}^* : w = 0^m 1^n \text{ where } m \text{ and } n \text{ are either both even or both odd}\}$$

(a) Write down the strings $w \in L$ such that the length of w is six. (2 points)

(b) Consider the following pair of languages.

$$L_1 = \{w \in \{0,1\}^* : w = 0^m 1^n \text{ where } m \text{ and } n \text{ are both even}\},$$
$$L_2 = \{w \in \{0,1\}^* : w = 0^m 1^n \text{ where } m \text{ and } n \text{ are both odd}\}.$$

Notice that $L = L_1 \cup L_2$. So, one way of designing a DFA for L would be to construct DFA for L_1 and L_2 and combine them using the "cross-product" construction shown in class.

Construct a DFA for L_1. (5 points)

(c) If you were to construct a DFA for L using the method described in (b), how many states would it have? Your answer should only be a number. (1 point)

(d) However, there is a DFA for L using at most seven states. Find that DFA. (2 points)

Problem 3 : Thompson's Construction Method (10 points)
Convert the following regular expression into an ϵ-NFA using Thompson's construction method.

$$ab^*c \cup ((ab \cup bc)^* \cup (a \cup b)(a \cup b)^*)bc \cup a(bc)^*$$

1) (a) $(1|\epsilon)^*(01)^*(0|\epsilon)$

OR $1?(01)^*0?$

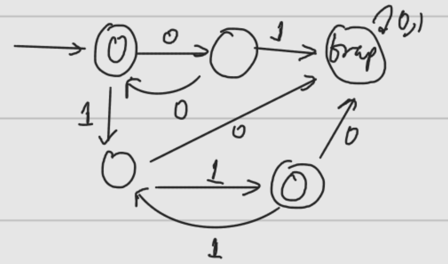
OR $0?(10)^*1?$

(b) $0^*(0^*10^*10^*10^*10^*)^*$

(c) $0^*(1(00)^*)^*(1|\epsilon)0^*$

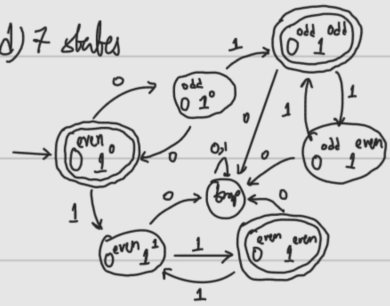
2) (a) 000000, 000001, 000011, 000111, 011111, 001111, 111111

(b)



(c) $5 \times 5 = 25$

(d) 7 states



Summer 2022 Mid Term Question Solved

Midterm Exam
Total marks: 40
Duration: 70 minutes CSE331

There are a total of four problems. You have to solve them all.

Problem 1: Finite Automata and the Regular Operations (10 points)

Let $\Sigma = \{0, 1, \#\}$. Consider the following two languages.

$$L_1 = \{w \in \Sigma^* : w \text{ does not contain } \# \text{ and the number of 0s in } w \text{ is not a multiple of 3}\}$$
$$L_2 = \{w \in \Sigma^* : \text{the substring between any two successive occurrences of } \#\text{s in } w \text{ is in } L_1\}$$

Now solve the following problems.

(a) Write down a string $w \in L_2$ such that the length of w is ten. (1 point)

(b) Give the state diagram for a DFA that recognizes L_1. (4 points)

(c) Give the state diagram for a DFA that recognizes L_2. (3 points)

(d) If you use the "cross product" construction shown in class to obtain a DFA for $L_1 \cap L_2$, how many states will it have? (1 point)

(e) Give an upper bound on the number of states in the smallest DFA that recognizes $L_1 \cap L_2$. (1 point)

Problem 2: Regular Expressions (10 points)

Mike and Willy recently learned how to write regular expressions. Mike wrote the regular expression 10^*1^* for a language L_1 on the board and Willy wrote the regular expression 1^*01^* for another language L_2 below that.

(a) Write down a string that is present in the language L_1 but not in the language L_2. (2 points)

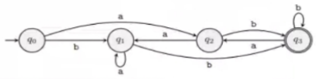
(b) Write down a string that is not present in the language L_1 but present in the language L_2. (2 points)

(c) Write down a string that is neither present in the language L_1 nor in the language L_2. (2 points)

(d) Mike and Willy asked their friend Dustin to write a regular expression for the language $L_1 \cap L_2$. Dustin came up with $1^*0^*1^*$. Is Dustin's regular expression correct? If you think it's not correct, then write down a correct regular expression for $L_1 \cap L_2$. (4 points)

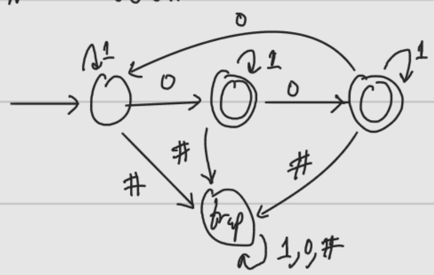
Problem 3: Converting DFAs to Regular Expressions (10 points)

Convert the following DFA into an equivalent regular expression using the state elimination method. First eliminate q_1, then q_2 and so on. You must show work.

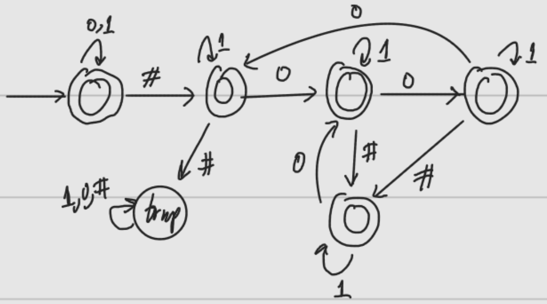


1) (a) #00000000#

(b)



(c)



(d) $4 \times 6 = 24$ states

(e) $L_1 \cap L_2 = L_1$ $\therefore$ 4 states

2) (a) $L_1 \rightarrow 10^*1^*$, $L_2 \rightarrow 1^*01^*$

$\therefore$ 100111

(b) 1110111

(c) 10101010

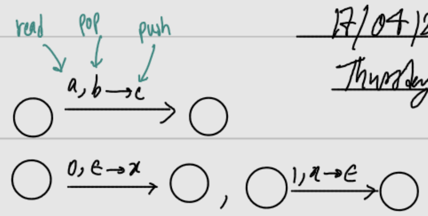
(d) Dustin's RE is wrong

correct one is 101^*

Finals

PDA – Introduction

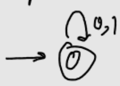
17/04/2025
Thursday



Stack   pop
         push

We can't draw infinite length DFA
e.g. $0^n 1^n$ here it requires infinite no. of counts

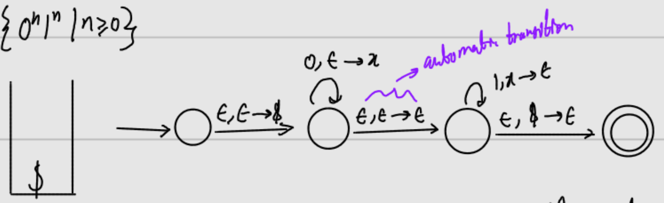
DFA can accept infinite length DFA but
e.g. Σ^* →



(finite no. of counts)

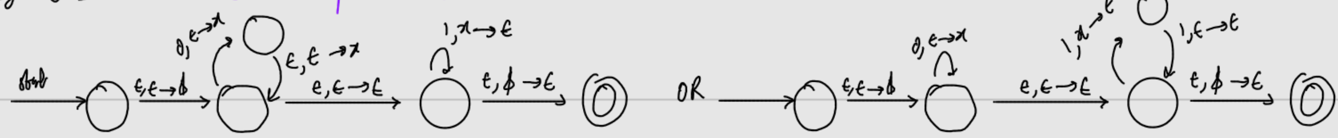
Put a dollar for (reference) stack for ready to use

Q) $\{0^n 1^n \mid n \geq 0\}$

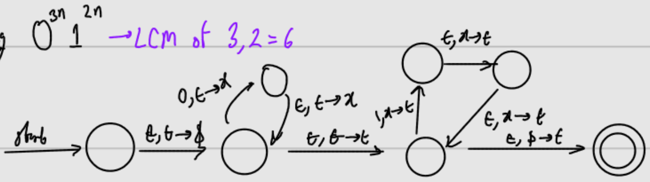


How to use the Stack

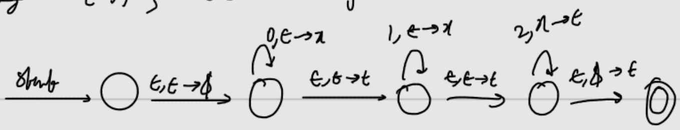
eg $0^n 1^{2n}$ For one zero we put two x



eg $0^{3n} 1^{2n}$ → LCM of 3, 2 = 6



eg $\Sigma = \{0, 1, 2\}$ $0^i 1^j 2^k$ $i + j = k$ eg $0^i 1^j 2^k$ $j = k + i$



Two Stack PushDown Automata (out of Syllabus)

eg $0^i 1^j 2^k$ $i = j = k$

PDA Examples

Palindrome

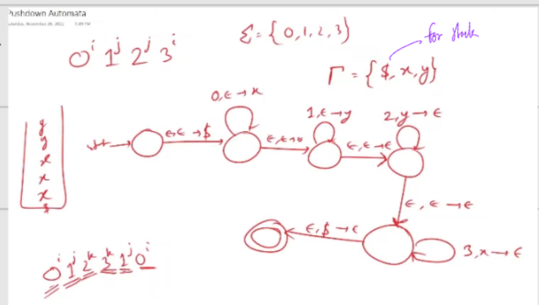
eg. $0^i 1^j 2^j 3^i$, $\Sigma = \{0, 1, 2, 3\}$

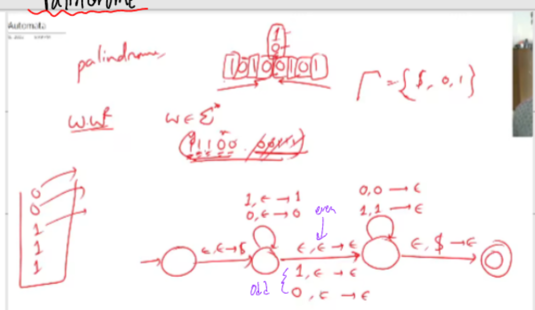
Possible due to nature of stack

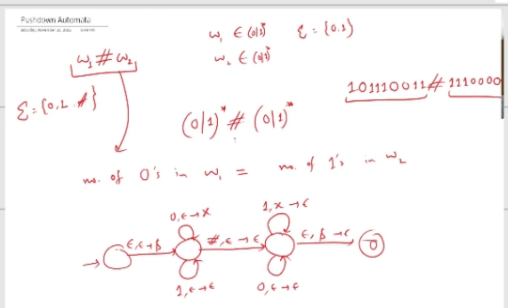
But not possible for $0^i 2^j 1^i 3^j$

For this we need two stacks or turing machine

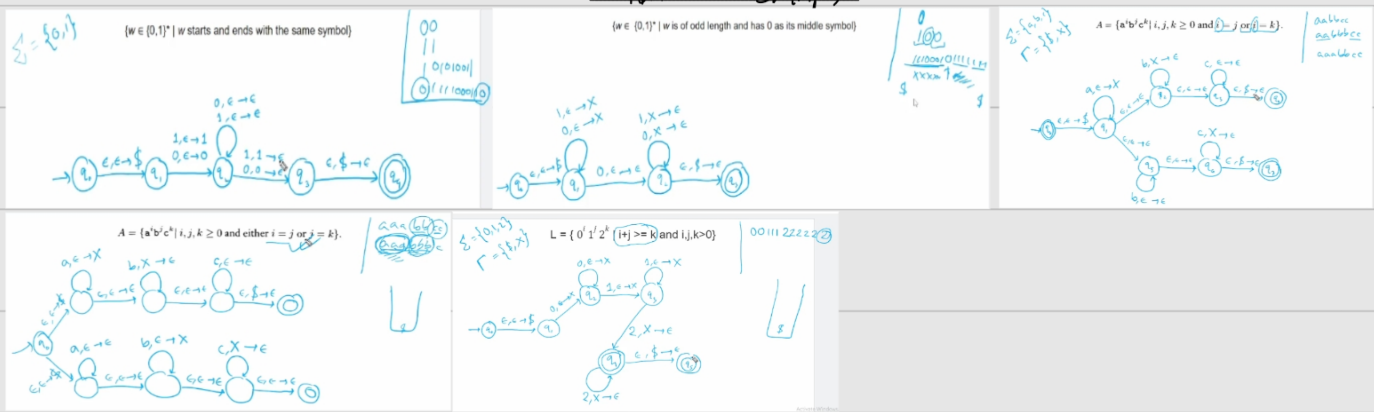
Ans







PDA More Examples



Assignment-3 Solution Video

1) Draw PDAs for the following language. Assume $\Sigma = \{0, 1\}$

$$L = \{ww^R : w \in \{0,1\}^*\}$$

Here, w^R means the reverse of the string w.

2) Draw PDAs for the following language. Assume $\Sigma = \{0, 1\}$

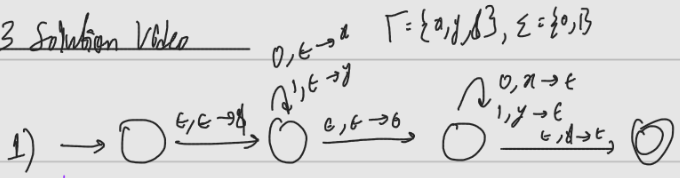
$L_1 = \{w \in \Sigma^* : w \text{ contains odd number of 0's}\}$
$L_2 = \{w \in \Sigma^* : w \text{ contains even number of 1's}\}$

As it's a RL, no need for PDA, DFA is enough

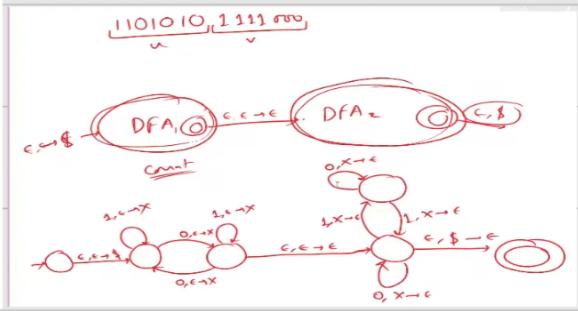
$L = \{w \in \{0,1\}^* : w = uv, \text{ where } u \in L_1, v \in L_2, \text{ and } |u| = |v|\}$

Describe what your automata is doing in a few sentences. equal length

$\Gamma = \{0, \gamma, \$\}, \Sigma = \{0, 1\}$



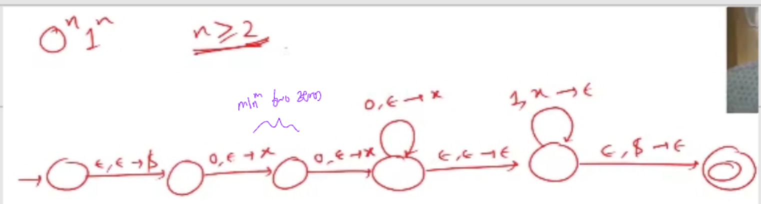
2) Well we stack to check length of u & v are same or not

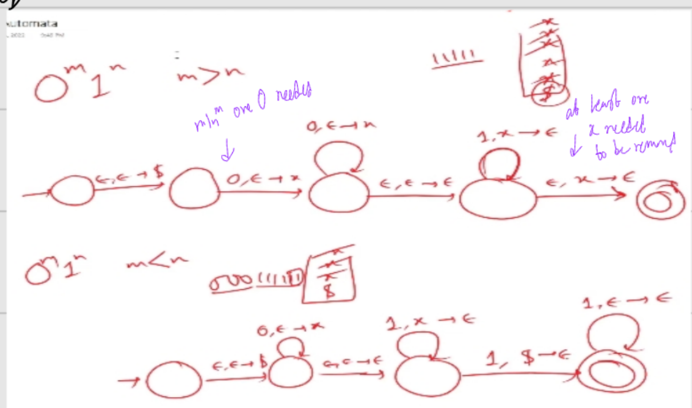


Draw PDAs for the following language. Assume $\Sigma = \{a, b, c, d, e, f\}$ [5x3 = 15 marks]

1. $a^n b^m c^k d^k e^m f^n$; where n, m, k ≥0. [3 marks] Assume $\Sigma = \{a, b, c, d, e, f\}$
2. $a^n b^m c^m d^k e^k f^n$; n≥0, m≥1, k≥2. [3 marks] Assume $\Sigma = \{a, b, c, d, e, f\}$
3. $a^m b^n c^p$; $n > m + p$ [3 marks] Assume $\Sigma = \{a, b, c\}$
4. $a^m b^n$; m>2n and m,n≥1 [3 marks] Assume $\Sigma = \{a, b\}$
5. $a^n b a^m b a^{n+m}$; m,n ≥1 [3 mark] Assume $\Sigma = \{a, b\}$

Cases of Inequalities





CFG | Introduction

(Context Free Grammar)

Not same

0^n1^n

Context free grammar of

$S \to 0S1$
$S \to \epsilon$

$0^n1^n = \{01, 0011, \ldots$

0 0000111111

S
$\Rightarrow 0S1$
$\Rightarrow 00S11$
$\Rightarrow 000S111$
000111

0^n1^n

0^n1^{2n}

000 11 11 11

$S \to 0S1$
$S \to \epsilon$

$S \to 0S11$
$S \to \epsilon$

$0^{3n}1^{5n}$

$S \to 000S11111$
$S \to \epsilon$

CFG | Derivation of string in a Grammar

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

$A \to 0A1 \mid B1$
$B \to 00$

derivations

A
$\Rightarrow B1$
$\Rightarrow 001$

A
$\Rightarrow 0A1$
$\Rightarrow 00A11$
$\Rightarrow 00B111$
$\Rightarrow 000111$

4 tuple (V, T, P, S)

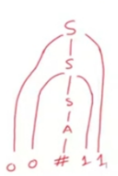
non-terminals/variables, terminals, start symbol

CFG | Left most Derivation and Parse Tree

eg. $S \to 0S1$
$S \to A$
$A \to \#$

$V = \{S, A\}$

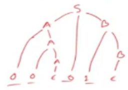
$T = \{0, 1, \#\}$



S
$\Rightarrow 0S1$
$\Rightarrow 00S11$
$\Rightarrow 00A11$
$\Rightarrow 00\#11$

$S \to A0B$
$A \to 0A \mid \epsilon$
$B \to 1B \mid \epsilon$

leftmost derivation

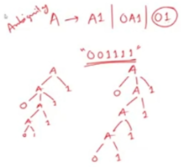


Same string can be derived different way

CFG | Ambiguity of a Grammar

$A \to A1 \mid 0A1 \mid 01$

"001111"



A
$\Rightarrow A1$
$\Rightarrow 0A11$
$\Rightarrow 0A111$
$\Rightarrow$ "001111"

A
$\Rightarrow 0A1$
$\Rightarrow 0A11$
$\Rightarrow 0A111$
$\Rightarrow$ "001111"

"01" A $\Rightarrow 01$

- Same string derived from more than one way
- Thus more than one parse tree

CFG – Regular Expressions

OR, Concatenation, Kleene Closure

0^n1^n

$S \to 0S1 \mid \epsilon$

1^n0^n

$S \to 1S0 \mid \epsilon$

000111 1100

$m_1 \mid m_2$

$S \to S_aS_b$
$S_a \to 0S_a1 \mid \epsilon$
$S_b \to 1S_b0 \mid \epsilon$

S $\Rightarrow S_aS_b$

000111 1100

OR, Concatenation, Kleene Closure

0^n1^n

$S \to 0S1 \mid \epsilon$

1^n0^n

$S \to 1S0 \mid \epsilon$

$0^* = \{\epsilon, 0, 00, 000, \ldots\}$

$(0100)^* = \{\epsilon, 0100, 01000100, \ldots\}$

$A \to 0A \mid \epsilon$

$B \to CB \mid \epsilon$
$C \to 0100$

$(0 \mid 10)^* \mid 10^*$

$S \to A \mid B$
$B \to 1C$
$C \to 0C \mid \epsilon$
$A \to DA \mid \epsilon$
$D \to 0 \mid 10$

CFG – Examples

$0^n1 2^n$

$S \to 0S2 \mid 1$

0^n1^{n+1}

0^n1^n1

0^n1^n2

$A \to S2$
$S \to 0S1 \mid \epsilon$

$0^n1^m2^k$ $k = m+n$

$0^n1^m2^m2^n$

$S \to 0S2 \mid X$
$X \to 1X2 \mid \epsilon$

$0^n1^m2^k$ $n = m+k$

$0^n1^{m+k}2^k$

$0^n1^n1^k2^k$

$S \to AB$
$A \to 0A1 \mid \epsilon$
$B \to 1B2 \mid \epsilon$

0^m1^n $m > n$

$S \to 0S1 \mid A$
$A \to 0A \mid 0$

CFG – More Examples

{w | w starts and ends with same symbol} $\Sigma = \{a, b\}$

$0\Sigma^*0 \mid 1\Sigma^*1$

$S \to 0A0 \mid 1A1$
$A \to 0A \mid 1A \mid \epsilon$

{w | w contains at least three 1s}

$S \to A1A1A1A$
$A \to 0A \mid 1A \mid \epsilon$

$S \to B1B1B1A$
$A \to 0A \mid 1A \mid \epsilon$
$B \to 0B \mid \epsilon$

{w | the length of w is odd}

$S \to AB$
$A \to 0 \mid 1$
$B \to AAB \mid \epsilon$

{w | the length of w is odd and its middle symbol is 0}

$S \to 0S0 \mid 1S1 \mid 0S1 \mid 1S0 \mid 0$

{w | w = w^R, i.e. w is a palindrome}

$S \to 0S0 \mid 1S1 \mid \epsilon \mid 1 \mid 0$

CFG | Equal Number of 0s and 1s

L: $w \in \Sigma^*$, $\Sigma = \{0,1\}$

#0's in w = #1's in w

$S \to 0S1S \mid 1S0S \mid \epsilon$

$S \to S0S1 \mid S1S0 \mid \epsilon$

#0's in w = #1's in w

$S \to 0S1 \mid 1S0 \mid SS \mid \epsilon$

001110

S ⇒ SS ⇒ 0S1S ⇒ 0011S ⇒ 0011 1S0 ⇒ 001110

$S \to 0S1 \mid 1S0 \mid SS \mid \epsilon$

$S \to S0S1 \mid S1S0 \mid \epsilon$

$S \to 0AS \mid 1BS \mid \epsilon$

$A \to 1 \mid 0AA$

$B \to 0 \mid 1BB$

00011011

S ⇒ 0AS ⇒ 00AAS ⇒ 000AAAS ⇒ 0001AAS ⇒ 00011AS ⇒ 000110AAS ⇒ 00011011

CFG | Examples from Previous Final

Palindrom

$S \to 0S0 \mid 1S1 \mid \epsilon \mid 0 \mid 1$

$0^i 1^j 2^k$, $i=j$ or $j \neq k$

$S \to A \mid B$

$A \to PQ$

$Q \to 2Q \mid \epsilon$

$P \to 0P1 \mid \epsilon$

$B \to WX$

$W \to 0W \mid \epsilon$

$X \to Y \mid Z$

$Y \to 1Y2 \mid Y'$

$Y' \to 2Y' \mid \epsilon$

$Z \to 1Z2 \mid Z'$

$Z' \to 1Z' \mid \epsilon$

L: $w_1 \# w_2$

no. of 0's in w_1 = no. of 1's in w_2

$S \to 1S \mid S0 \mid 0S1 \mid \#$

CFG | Quiz Questions Solved

Question 1: [15 Points]

a) Give a context-free grammar for the following language. (5 points) $A = \{w \in \{0,1\}^* : w \text{ contains odd number of 1s}\}$

b) Give a context-free grammar for the following language. (5 points) $W = \{w \in \{0,1\}^* : w \text{ starts and ends with same characters}\}$

c) Give a context-free grammar for the following language. (5 points) $L = \{w \in \{0,1\}^* : w = u0^{2i}v1^{3i} \text{ where } u \in B, v \in A \text{ and } i \geq 0\}$

b) $0\Sigma^*0 \mid 1\Sigma^*1 \mid 0 \mid 1$

$W \to 0P0 \mid 1P1 \mid 0 \mid 1$

$P \to QP \mid \epsilon$

$Q \to 0 \mid 1$

$(0^*10^*1)^*0^*10^*$

$A \to BCJC$

$C \to 0C \mid \epsilon$

$B \to DB \mid \epsilon$

$D \to C1C1$

c) $u\,0^{2i}1^{3i}\,v$

$S \to WXA$

$X \to 00X111 \mid \epsilon$

$W \to 0P0 \mid 1P1 \mid 0 \mid 1$

$P \to QP \mid \epsilon$

$Q \to 0 \mid 1$

CNF | Introduction

Chomsky Normal Form — Out of Syllabus

*** CNF is a special form of CFG

Only two types of production rule

$A \to BC$ — two variables

$A \to a$ — one terminal

e.g. $A \to aa$: $\boxed{A \to PP,\ P \to a}$

eg $A \to ABC$: $\boxed{A \to PC,\ P \to AB}$

e.g. $A \to XB$, $X \to a$

Rules:

1) Eliminate unreachable variables
2) Add a new start variable
3) Eliminate ϵ-production
4) Unit production elimination
5) Cleaning up the grammar

eg

(2) $S' \to S$

$S \to ASB \mid a$

$A \to aAS \mid a \mid \epsilon$

$B \to SbS \mid A \mid bb$

(1) ~~$C \to XY$~~

~~$X \to ab$~~

(3)

$S' \to S$

$S \to ASB \mid a \mid SB$

$A \to aAS \mid a \mid aS$

$B \to SbS \mid A \mid bb \mid \epsilon$

$S' \to S$

$S \to ASB \mid a \mid SB \mid AS \mid S$

$A \to aAS \mid a \mid aS$

$B \to SbS \mid A \mid bb$

(4)

$S' \to S$

$S \to ASB \mid a \mid SB \mid AS \mid \cancel{S}$

$A \to aAS \mid a \mid aS$

$B \to SbS \mid A \mid bb$

$S' \to ASB \mid a \mid SB \mid AS$

$S \to ASB \mid a \mid SB \mid AS$

$A \to aAS \mid a \mid aS$

$B \to SbS \mid bb \mid aAS \mid a \mid aS$

(5) First handle the non-single terminal

$S' \to ASB \mid a \mid SB \mid AS$

$S \to ASB \mid a \mid SB \mid AS$

$A \to XAS \mid a \mid XS$

$B \to SYS \mid YY \mid XAS \mid a \mid XS$

$X \to a$

$Y \to b$

$S' \to PB \mid a \mid SB \mid AS$

$P \to AS$

$S \to PB \mid a \mid SB \mid AS$

$A \to QS \mid a \mid XS$

$Q \to XA$

$B \to SR \mid YY \mid XP \mid a \mid XS$

$R \to YS$

$X \to a$

$Y \to b$

CNF | Examples

Out of Syllabus

e.g.

$S \to ASA \mid aB$
$A \to B \mid \epsilon$
$B \to b \mid \epsilon$

① $S' \to S$
$S \to ASA \mid aB \mid a$
② $A \to B \mid S \mid \epsilon$
$B \to b$

③ $S' \to S$
$S \to ASA \mid aB \mid a \mid SA \mid AS \mid S$
$A \to B \mid S$
$B \to b$

④ $S' \to ASA \mid aB \mid a \mid SA \mid AS$
$S \to ASA \mid aB \mid a \mid SA \mid AS$
$A \to b \mid ASA \mid aB \mid a \mid SA \mid AS$
$B \to b$

⑤ $S' \to ASA \mid XB \mid a \mid SA \mid AS$
$S \to ASA \mid XB \mid a \mid SA \mid AS$
$A \to b \mid ASA \mid XB \mid a \mid SA \mid AS$
$B \to b$
$X \to a$
$Y \to b$

$S' \to PA \mid XB \mid a \mid SA \mid AS$
$S \to PA \mid XB \mid a \mid SA \mid AS$
$A \to b \mid PA \mid XB \mid a \mid SA \mid AS$
$B \to b$
$X \to a$
~~$Y \to b$~~
$P \to AS$

CNF | Final Spring 2023 Question Solved

Out of Syllabus

Problem 5 (CO4): Chomsky Normal Form (10 points)

Answer the following questions.

(a) **List** the productions that violate the conditions of the Chomsky Normal Form (CNF) in the following grammar. (5 points)

$S \to aXbY \mid X$
$X \to a \mid ba \mid YY$
$Y \to c \mid \varepsilon$

(b) **Write** down the additional rules that need to be added to the following grammar if the production $Y \to \varepsilon$ is removed. (3 points)

$S \to aXbY \mid X$
$X \to a \mid ba \mid YY$
$Y \to c \mid \varepsilon$

(c) **Write** down the additional rules that need to be added to the following grammar if all the unit productions are removed. (2 points)

$S \to aAbB \mid A$
$A \to a \mid b \mid B$
$B \to c \mid d \mid A$

(a) $S \to aXbY$
$S \to X$
$X \to ba$
$Y \to \epsilon$

(b) $S \to aXb$
$X \to Y \mid \epsilon$

or

$S \to aYb \mid \epsilon \mid ab$
$X \to Y$

ϵ is allowed in starting variable

(c) $S \to aAbB \mid A$
$A \to a \mid b \mid B$
$B \to c \mid d \mid A$

$\longrightarrow$ $S \to aAbB \mid a \mid b \mid B$
$A \to a \mid b \mid B$
$B \to c \mid d \mid A$

$\longrightarrow$ $S \to aAbB \mid a \mid b \mid c \mid d \mid A$
$A \to a \mid b \mid c \mid d \mid A$
$B \to c \mid d \mid a \mid b$

no need

no need

Additional rules

CYK Algorithm

Out of Syllabus

(Watch the recording again for better understanding)

$\longrightarrow$ CYK algorithm returns a boolean output if an input string w is part of an input grammar G

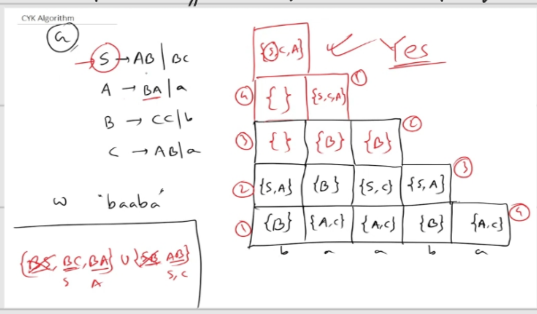
e.g. G

$S \to AB \mid BC$
$A \to BC \mid a$
$B \to CC \mid b$
$C \to AB \mid a$

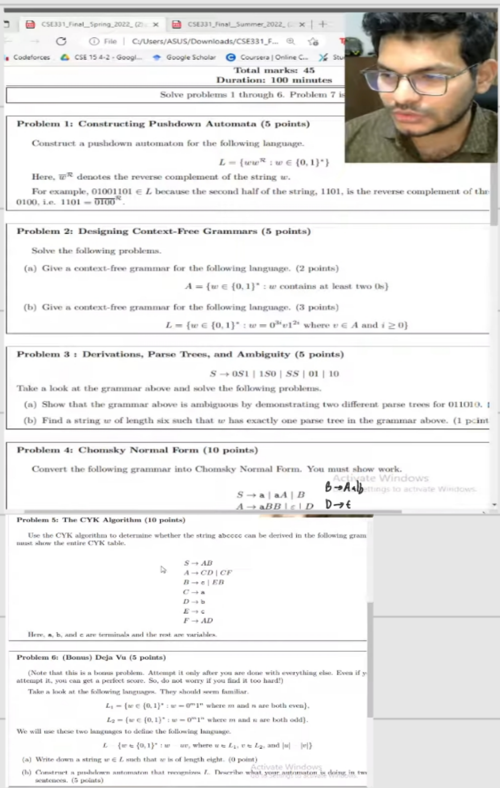
It's in CNF form

w: "baaba"

{B}	{A,C}	{A,C}	{B}	{A,C}
b	a	a	b	a



Spring 2022 Final Question Solved



1) $\longrightarrow \bigcirc \xrightarrow{\epsilon,\epsilon\to\$} \bigcirc \xrightarrow{\epsilon,\epsilon\to\epsilon} \bigcirc \xrightarrow{\epsilon,\$\to\epsilon} \circledcirc$

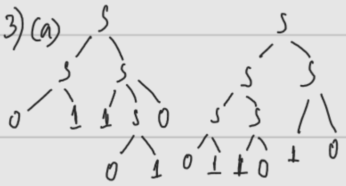
$1,\epsilon\to 1$; $0,\epsilon\to 0$ (loop on second state)
$1,0\to\epsilon$; $0,1\to\epsilon$ (loop on third state)

2)(a) RE: $1^*01^*0\Sigma^*$

$S \to P0P0Y$
$P \to 1P \mid \epsilon$
$Y \to 0Y \mid 1Y \mid \epsilon$

(b) $w = 0^{3i}v1^{2i}$ $v \in A$

$S' \to 000S'11$
$S' \to S$
$S \to P0P0Y$
$P \to 1P \mid \epsilon$
$Y \to 0Y \mid 1Y \mid \epsilon$

3)(a) 

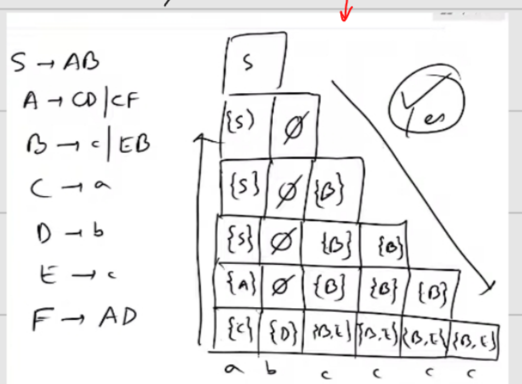
(b) 111000

4) $S \to a \mid aA \mid B$
$A \to aBB \mid \epsilon$
$B \to Aa \mid b$

$S \to a \mid aA \mid Aab$
$A \to aBB$
$B \to Aa \mid bk$
$X \to a$

$S \to a \mid XA \mid AX \mid b$
$A \to XY$
$B \to AX \mid b \mid a$
$X \to a$
$Y \to BB$

Out of Syllabus

5) 

Summer 2022 Final Question Solved

CSE331_Final_Spring_2022_(2) | CSE331_Final_Summer_2022_
File | C:/Users/ASUS/Downloads/CSE331_F...
Codeforces | CSE 15 4-2 - Googl... | Google Scholar | Coursera | Online C...

Final Exam
Total marks: 50
Duration: 95 minutes

There are a total of six problems. You have to solve

Problem 1 (CO2): Models Recognizing CFLs (10 points)

Let $\Sigma = \{0, 1\}$. Consider the following language.

$$L = \{w \in \Sigma^* : w = 0^n 1^n \text{ where } n \text{ is odd}\}$$

(a) **Construct** a context-free grammar that generates L. (5 points)

(b) **Design** a pushdown automaton that recognizes L. (5 points)

Problem 2 (CO3): Nonregular Languages (10 points)

Let $\Sigma = \{0, 1\}$. Consider the following language.

$$L = \{w \in \Sigma^* : w = 0^n 1^m 0^k \text{ where } n = m \text{ or } m \neq k\}$$

(a) Use the pumping lemma to **demonstrate** that L is not regular. (8 points)

(b) **Find** a string $w \in L$ such that there exist $x, y, z \in \Sigma^*$ with $y \neq \varepsilon$ such that $w = xyz$ and $xy^iz \in L$ fo
Does this contradict the pumping lemma? (2 points)

Problem 3 (CO2): Derivations, Parse Trees and Ambiguity (10 points)

Take a look at the grammar below and solve the following problems.

$$A \to A1 \mid 0A1 \mid 01$$

(a) **Give** a leftmost derivation for the string 001111. (3 points)

(b) **Sketch** the parse tree corresponding to the derivation you gave in (a). (2 points)

(c) **Demonstrate** that there are two more parse trees (apart from the one you already found in (b)) fo
string. (4 points)

(d) **Find** a string w of length six such that w has exactly one parse tree in the grammar above. (1 poin

Problem 4 (CO2): Chomsky Normal Form (10 points)

Convert the following grammar into Chomsky Normal Form. You must show work. Here a, b, c are
and the rest are variables.

$$S \to bXaY \mid ZXb$$
$$X \to aY \mid bY \mid Y$$
$$Y \to X \mid c \mid \varepsilon$$
$$Z \to ZaX$$

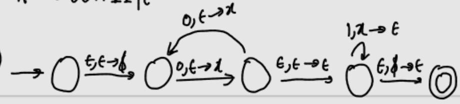
Problem 5 (CO4): The CYK Algorithm (10 points)

Apply the CYK algorithm to determine whether the string abcaa can be derived in the following gran
must show the entire CYK table. Here a, b, c are terminals and the rest are variables.

$$S \to CA$$
$$A \to AA \mid AD \mid a$$
$$B \to AB \mid BC \mid b$$
$$C \to CA \mid BC \mid c$$
$$D \to a$$

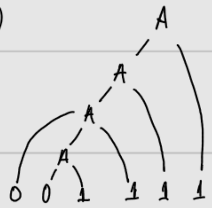
1) (a) $S \to 0A1$
$A \to 00A11 \mid \epsilon$

(b)

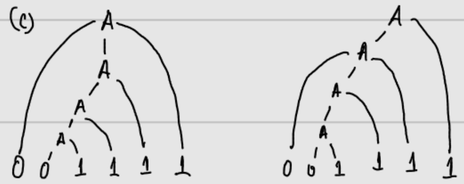


3) (a) A
$\Rightarrow A1$
$\Rightarrow A11$
$\Rightarrow 0A111$
$\Rightarrow 001111$ //

(b)



(c)



(d) 011111 //

4) $S \to bXaY \mid ZXb \mid bXa \mid baY \mid Zb \mid ba$
$X \to aY \mid bY \mid a \mid b \mid c$
$Y \to c \mid aY \mid bY \mid a \mid b$
$Z \to ZaX \mid Za$
$A \to a$
$B \to b$

Out of Syllabus

$S \to CD \mid ZP \mid BQ \mid BD \mid ZB \mid BA$
$X \to AY \mid BY \mid a \mid b \mid c$
$Y \to c \mid AY \mid BY \mid a \mid b$
$Z \to RX \mid ZA$
$A \to a$
$B \to b$
$C \to BX$
$D \to AY$
$P \to XB$
$Q \to XA$
$R \to ZA$ //

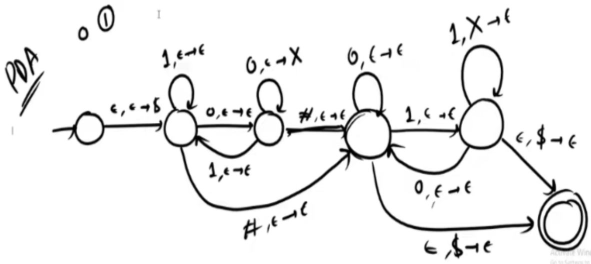
PDA and CFG Assignment Solutions

Language 2: Assume $\Sigma = \{0, 1, \#\}$
$L_2 = \{w_1\#w_2 \mid$ number of '00' substrings in w_1 is equal to the number of '11' substrings in $w_2\}$

00100①# 0100100011 0000

0 ①

PDA



(a) **Language 1:** Assume $\Sigma = \{0, 1, \#\}$
$L_2 = \{w_1\#w_2 \mid$ length of w_1 is double of the length of $w_2\}$

(b) **Language 2:** Assume $\Sigma = \{0, 1, 2, 3\}$
$L_1 = \{0^i 1^j 2^k 3^m$ where $j = k + 1$ and $i = m - 2\}$

(c) **Language 3:** Assume $\Sigma = \{0, 1, 2, 3, 4\}$
$L_1 = \{0^i 1^j 2^k 3^m 4^n$ where $j = i$ and $m = n$ and $k > 0\}$

(a) $S \to AASA \mid \#$

$A \to 0 \mid 1$

Language 1: Assume $\Sigma = \{0, 1, \#\}$
$L_2 = \{w_1\#w_2 \mid$ length of w_1 is double of the length of $w_2\}$

Language 2: Assume $\Sigma = \{0, 1, 2, 3\}$
$L_1 = \{0^i 1^j 2^k 3^m$ where $j = k + \underline{1}$ and $i = m - 2\}$

$i+2 = m$

Language 3: Assume $\Sigma = \{0, 1, 2, 3, 4\}$
$L_1 = \{0^i 1^j 2^k 3^m 4^n$ where $j = i$ and $m = n$ and $k > 0\}$

$0^i 1^j 2^k 3^m$

$0^i \boxed{1^{k+1} 2^k} 3^{i+2}$

(b) $S \to 0S3$

$S \to P33$

$P \to 1P2$

$P \to 1$

Language 1: Assume $\Sigma = \{0, 1, \#\}$
$L_2 = \{w_1\#w_2 \mid$ length of w_1 is double of the length of $w_2\}$

Language 2: Assume $\Sigma = \{0, 1, 2, 3\}$
$L_1 = \{0^i 1^j 2^k 3^m$ where $j = k + \underline{1}$ and $i = m - 2\}$

$i+2 = m$

Language 3: Assume $\Sigma = \{0, 1, 2, 3, 4\}$
$L_1 = \{0^i 1^j 2^k 3^m 4^n$ where $j = i$ and $m = n$ and $k > 0\}$

$0^i 1^j 2^k 3^m 4^n$
A B C

(c) $S \to ABC$

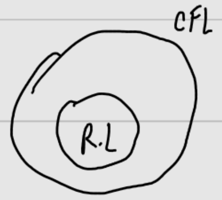
$A \to 0A1 \mid \epsilon$

$C \to 3C4 \mid \epsilon$

$B \to 2B \mid \epsilon$

Pumping Lemma (Concept)

20/05/2025
Wednesday



If L:
↳ is regular, draw DFA
↳ is not regular, pumping Lemma

Pumping Lemma ←

Ⓛ p ≤ $|s|$ all strings of L

xyz

☆ such that:

(i) $xy^iz \in L$ for $i \in \mathbb{N}$

(ii) $|y| > 0$

(iii) $|xy| \leq p$

*** For RL we do DFA, RE. For non-RL we do CFG

Pumping Lemma (Practice Problem)

0^n1^n is not regular

Let, the pumping length be p. 0^p1^p

$000\ldots0011\ldots111 = xyz$

case 1: If y consists of only 0's. Then xy^2z will have more 0's.

case 2: If y consists of only 1's. Then xy^2z will have more 1's.

case 3: If y consists of both 0's and 1's, then format will get ruined

∴ L is not regular.

$L = \{w \mid w$ is not a pallindrome$\}$

Difficult to prove that L is not regular. But easier to prove that $\bar{L}$ is not regular as

$\bar{L}$ is not regular.

$\bar{L} = L$ is also not regular

$\bar{L} = \{w \mid w$ is a pallindrome$\}$

$|xy| \leq p$

$\rightarrow p$ 0^p10^p

$\underbrace{000\ldots0}_{y}\underbrace{0}_{y}1\underbrace{00}_{y}\ldots000$

If a language is regular, its compliment is also regular

Turing Machine (Concept)

Turing Machine

There are languages outside CFL that we can't represent using CFG or PDA.

eg: $\{w\#w^R\}$ (Palindrome) we could do PDA but not $w\#w$

• It can freely move left & right

Transitions

Pointer at x on the tape x changes to y and pointer moves right.

$A \xrightarrow{x \to y, R} B$

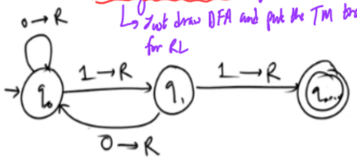
reading x without changing it and moving L

$A \xrightarrow{x \to L} B$

Turing Machine for RL and CFL

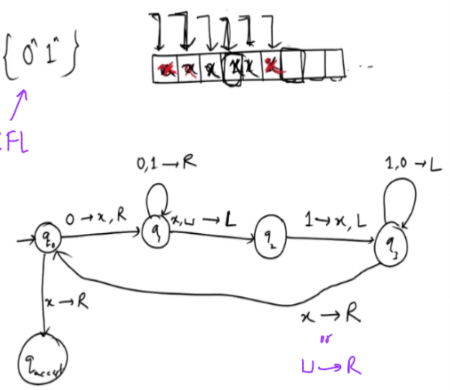
$L_1 = \{w \mid w$ contains "11" as a substring$\}$

Turing Machine State diagram
↳ Just draw DFA and put the TM transitions for RL



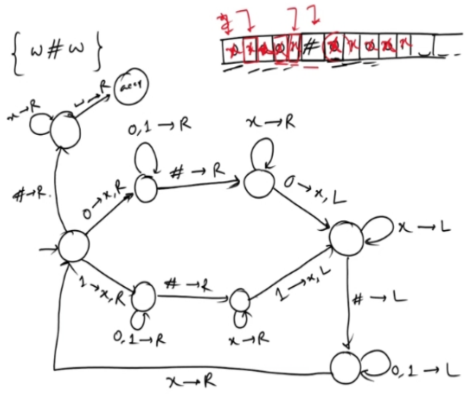
$L = \{0^n1^n\}$

CFL



Turing Machine (The Real Deal)

$L = \{w\#w\}$



Turing Decidable and Recognizable Language

DFA → RL
PDA → CFG
TM → Accept / Reject / loop

halts on all inputs
↓
Turing Decidable Language / Recursive Language

If TM accepts, we call Turing Recognizable language / Recursively enumerable Language (REL)

THE

END