Theory of Computation

KTU S5 CSE CS306

November 14, 2020

Contents

Chapter 1

Module 1

- FSM : Finite State Machine
- CFL : Context Free Language
- Turing Machine

FSM $\rightarrow$ CFL $\rightarrow$ Turing Machine

Language here is a **set of Strings**

1.1 FSM

- Symbol: `a b c 0 1 2 4...`
- Alphabet : Denoted by Σ is a collection of symbols
  - Eg : $\{a, b, c\}$ *or* $\{1, 2, 3\}$
- String : Sequence of Symbols eg: a,b,c... or aa,bb,cc,...
- Language : Set of Strings
  - Eg: $\Sigma = \{0, 1\}$
  - Set of all strings of Length 2: $\{00, 01, 10, 11\}$
  - Set of all strings of length 3: $\{000, 001, 010\}$
  - Set of all strings that begin with 0: $\{0, 001, 010\}$
  - Third example is an ∞ set

1.1.1 Powers of Σ

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

- Σ^0 = Set of all strings of length 0: $\Sigma^0 = \{\epsilon\}$ (Epsilon)
- ϵ denotes all strings of length 0
- Σ^1 = Set of all strings of length 1; $\Sigma^1 = \{0, 1\}$
- Σ^n = Set of all strings of length n

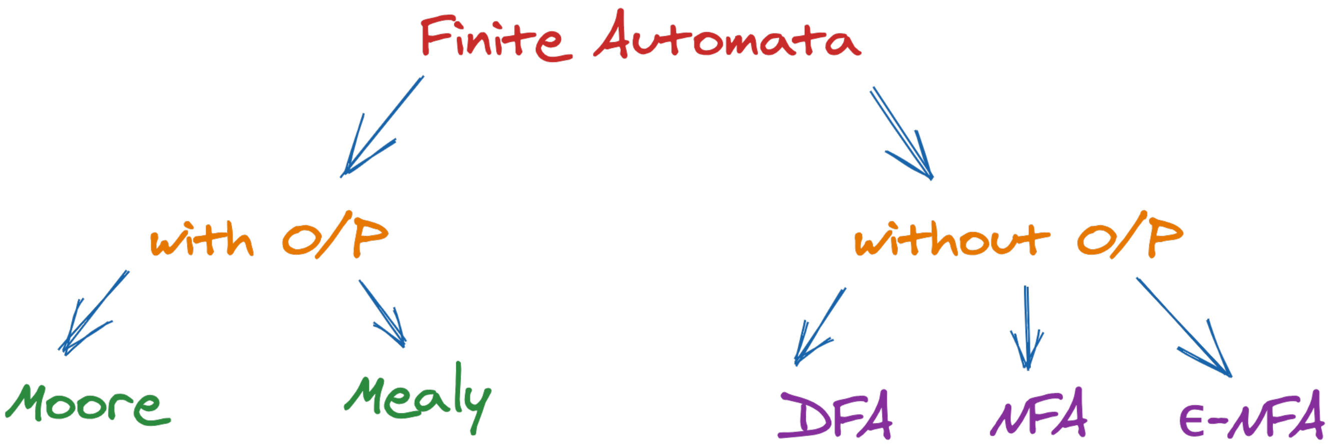
1.1.2 Cardinality

- No of Elements in a set
- Cardinality of $\Sigma^n = 2^n$

1.1.3 Σ^*

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup ......\Sigma^n$
=> $\{\epsilon\} \cup \{0, 1\} \cup .....$
=> Set of all possible strinfgs of all length over $\{0, 1\}$ -> ∞ set

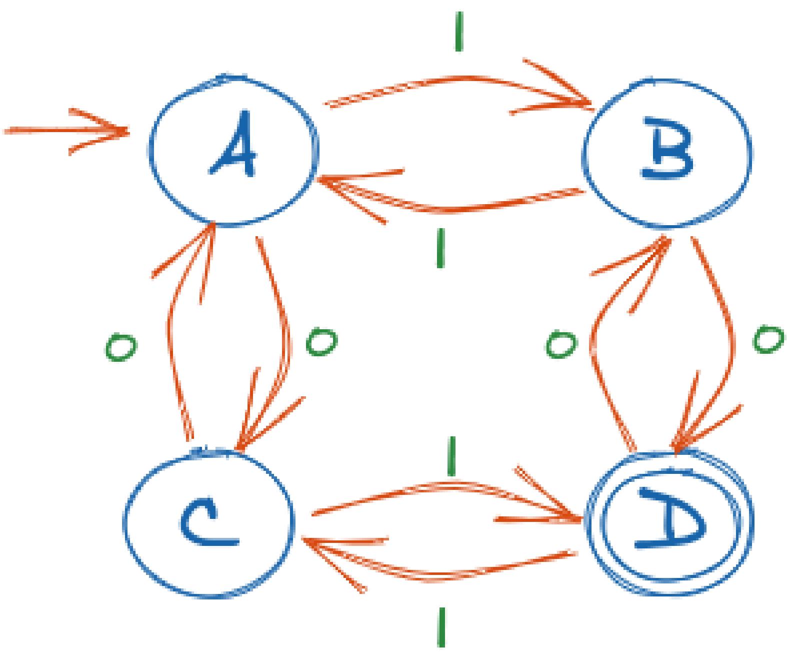
1.1.4 Finite Automata



FA-1

1.1.5 DFA - Deterministic Finite Automata

- Simplest Model fo Computation
- It has veru limited memory



DFA-1

- Circles are **States**
- Edges/ Arrows are **transistions**
- Labelling are **Inputs**
- Double Cirlce is Final State
- Arrow from *nowhere* is the Initial State

Every DFA can be represented using 5 tuples $\rightarrow (Q, \Sigma, q_0, F, \delta)$

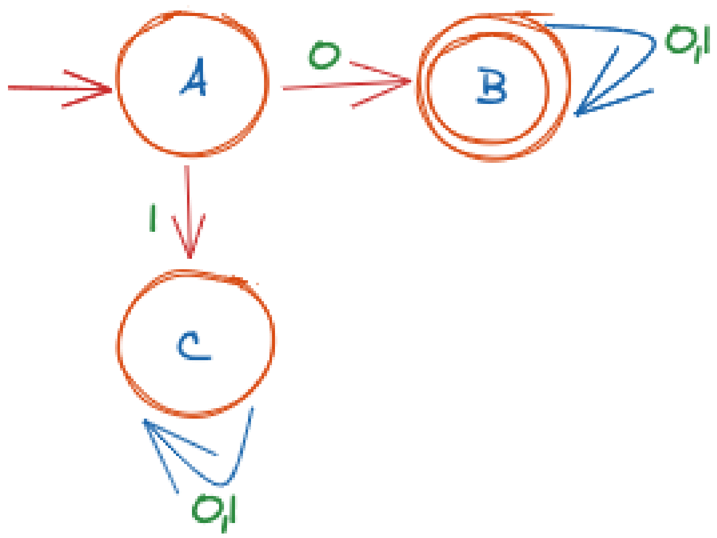
- Q : Set if al States
- Σ : Inputes
- q_0: Start state/ Initaial State
- F : Set of Final States
- δ : Transisition Function that maps from $Q \times \Sigma \rightarrow Q$

For the Above DFA, the values are - $Q = \{A, B, C, D\}$ - $\Sigma = \{0, 1\}$ - $q_0 = A$ - $F = \{D\}$ - $\delta =$

	0	1
A	C	B
B	D	A
C	A	D
D	B	C

Example Question: Let L1= Set of all strings that stras with 0 = {0,00,01,000,010,011,0000,....}. Design the DFA

Answer:

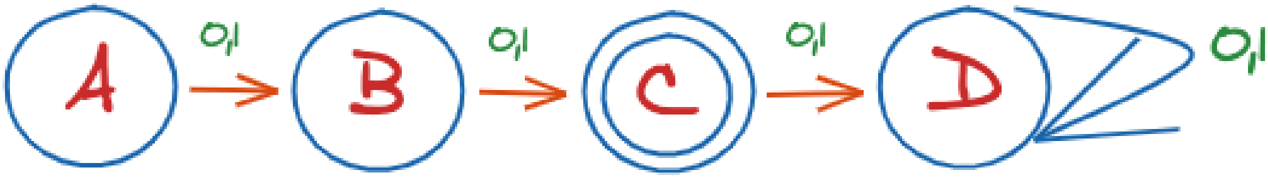


DFA-2

Here C is the Dead State of Trap State

Example Question: Construct a DFA that accepts sets of all strings over {0,1} of length 2.

Answer: $\Sigma = \{0, 1\}$ and $L = \{00, 01, 10, 11\}$



RegularLanguage - a regular language is said to be a regular language $\iff$ a finite state machhine recognizes it - Not Regular - When requires memory - Not recognized by FSM

Memory of FSM is very Limited and canno count strings

1.1.6 Operations on Regular Languages

Union : $A \cup B = \{x \mid x \in A \ or \ x \in B\}$
Concatenation: $A \circ B = \{xy \mid x \in A \ and \ y \in B\}$
Star: $A* = \{x_1, x_2, x_3 \dots x_k \mid k \geq 0 \ and \ each \ x_i \in A\}$

Eg: $A = \{pq, r\}, B = \{t, uv\}$
Ans:
- $A \cup B = \{pq, r, t, uv\}$ - $A \circ B = \{pqt, pquv, rt, ruv\}$ - $A* = \{\epsilon, pq, r, pqr, rpq \dots\} = \infty \ set$

Theorem 1: The class of Reglular Lanugauges is closed under $Union(\cup)$
Theorem 2: The class of RL is closed unde $Concatenation(\circ)$

1.2 NFA - Non Deterministic Finite Automata

- There could be multiple Next states
- The next state could be chosen at random
- All the next states may be chosen in Parallel

1.2.1 NFA - Formal Definition

- NFA are defined using
  - Q : Set if al States
  - Σ : Inputs
  - q_0: Start state/ Initaial State
  - F : Set of Final States
  - δ : Transisition Function that maps from $Q \times \Sigma \to 2^Q$

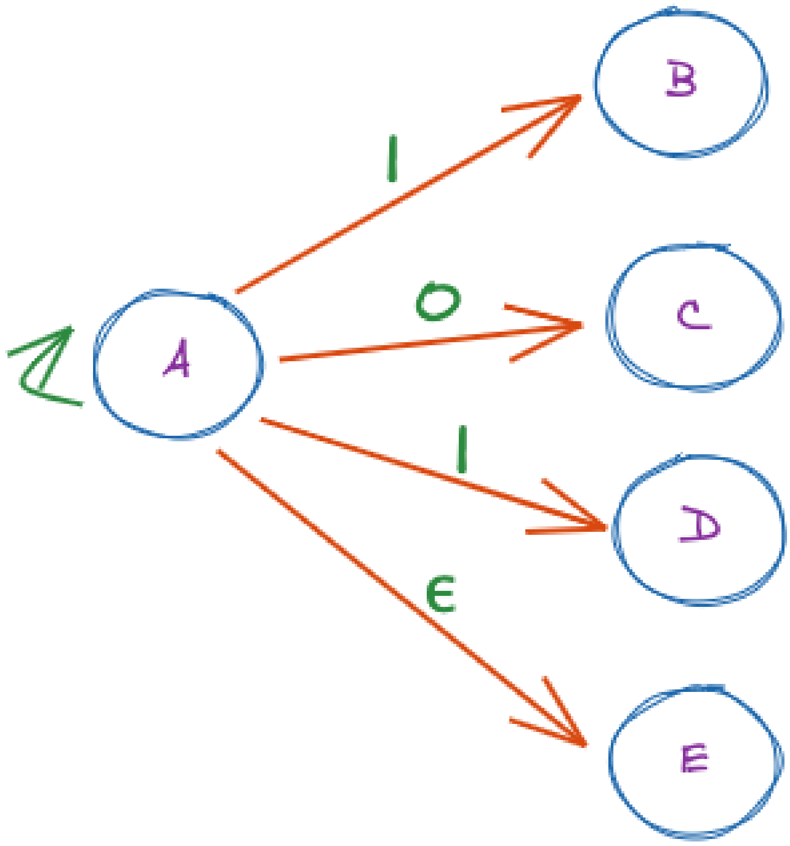
L={Set of all strings that end with 0}

If there is any way to run the machine that **ends in any set of states** out of which **atleast one state is a final state**, then the NFA accepts

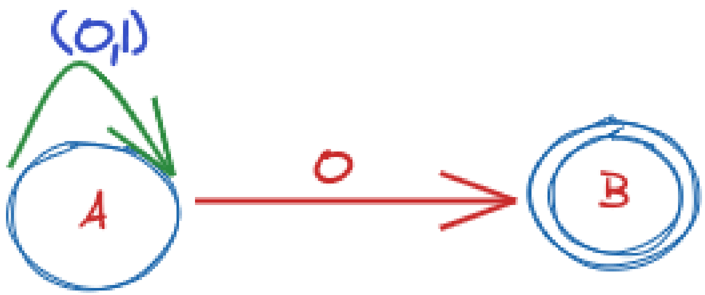
For examples, check this video

1.3 Conversion NFA to DFA

Every DFA is an NFA, but not vice-versa. But there is an equivalent DFA for every NFA



NFA-1

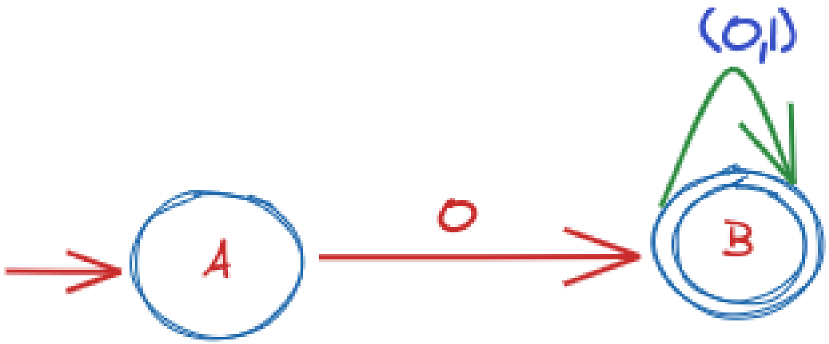


NFA-2

Things to keep in mind - ϕ in NFA is the Dead State in DFA - NFA's ϕ should be replaced by another state in DFA

Example 1: Convert to DFA, `L={Set of all strings over(0,1) that starts with 0}`

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



NFA(NFA to DFA)

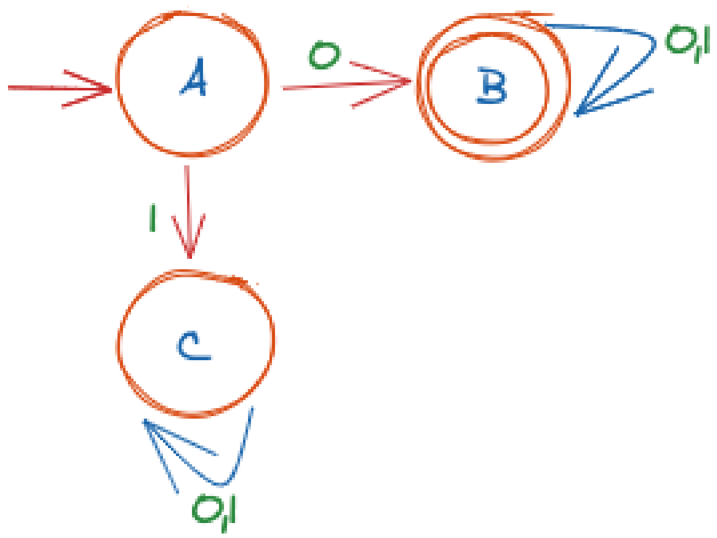
Transition Table:

	0	1
A	B	ϕ
B	B	B

While converting this to DFA, we have to account the ϕ as it should be converted to a **dead state**. This gives us the Transition Table,

	0	1
A	B	C
B	B	B
C	C	C

Which gives us the DFA, where `C` is the dead state



DFA-2