Varun Khadayate A016

FLAT
Assignment - I

Q1. Structural Induction is a proof methodology similar to mathematical Induction, only instead of working in the domain of positive integers (N) it works in the domain of such recursively defined structures.
The set of natural numbers N has a particular structure that allows us to define it using the following recursive definition:

- $0 \in N$
- if $n \in N$, then $n+1 \in N$
- N contains nothing else.

Comparable structures exist in many sets and allows us to define them recursively as follows:

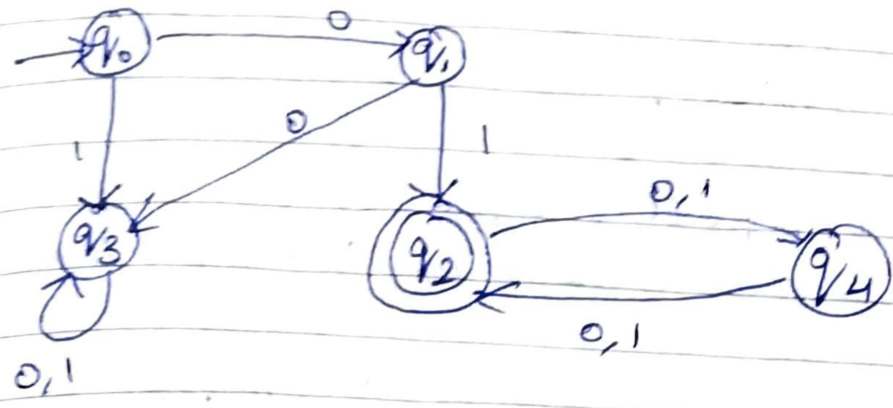
1. Base Case: Define the "smallest" or "simplest" object in the set.
2. Induction Step: Define the ways in which "larger" or "complex" objects in the set can be constructed out of "smaller" or "simpler" objects in the set.

Mutual Induction, is a technique for proving results or establising statements for natural numbers. This part illustrates the method through a variety of examples.
Mathematical Induction is a mathematical technique which is used to prove a statement, a formula or a theorem is true for every natural number.
The technique involve 2 steps to prove a statement as stated below:-

1. Base Step: It proves that a statement is true for initial value.
2. Inductive Step: It proves that if the statement is true for the n^{th} iteration, then its also true for $(n+1)^{th}$ iteration.

Varun K. A016

Q2



Q3. $R = \{(1,2), (2,3), (3,4), (5,4)\}$

Transitive Closure:

$R^+ = \{(1,2); (2,3); (3,4); (5,4); (1,3); (2,4); (1,4)\}$

$\because$ (1,2) and (2,3) are present in R^+, $\therefore$ (1,3) is added. Similar is the case for (2,4) and (1,4)

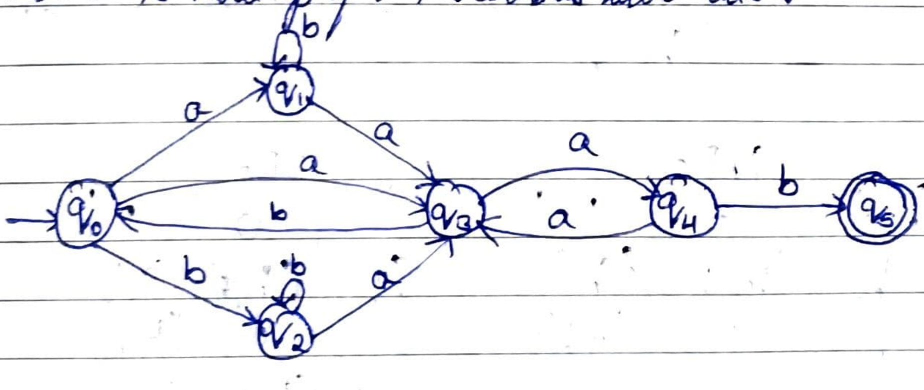
Symmetric Closure:

$R = \{(1,2); (2,3); (3,4); (5,4); (2,1); (3,2); (4,3); (4,5)\}$

$\because$ (1,2) is present in R, $\therefore$ (2,1) is added. Similar is the case for (3,2), (4,3), (4,5).

Hence Proved.

Q4. L: L = {x | x is made up of {a, b} and ends with 'aab'}



Varun K   A016

Q5. METHOD-I

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

$\delta = Q \times \Sigma \rightarrow 2^Q$

Eg: $M = [\{q_0, q_1\}, \{0,1\}, \delta, q_0; \{q_1\}]$

In this the above expression is in NFA and we have to convert it to DFA.

In this, Q [Finite set of states] are $\{q_0, q_1\}$

Σ [Finite input alphabet] are $\{0, 1\}$

δ [STF that maps $Q \times \Sigma \rightarrow Q$, here its $Q \times \Sigma \rightarrow 2^Q$] are $q_1 \delta$

q_0 [Initial state of FA] are q_0

F [Set of final states] are $\{q_1\}$.

Now, lets write it in resultant DFA as:

$M' = \{Q', \Sigma, \delta', [q_0], F'\}$

$2^Q = \{\phi, [q_0], [q_1], [q_0, q_1]\}$.

Power Set of Q

$Q' = \{[q_0], [q_1], [q_0, q_1]\}$

Here, ϕ is excluded cause it donot denote any state in the NFA.

$\delta'([q_0], 0) = [q_0, q_1]$ $\delta'([q_0], 1) = [q_1]$

$\delta'([q_1], 0) = \phi$ $\delta'([q_1], 1) = [q_0, q_1]$

$\delta'([q_0, q_1], 0) = [\delta(q_0, 0) \cup \delta(q_1, 0)] = [\{q_0, q_1\} \cup \phi] = [q_0, q_1]$

$\delta'([q_0, q_1], 1) = [\delta(q_0, 1) \cup \delta(q_1, 1)] = [\{q_1\} \cup \{q_0, q_1\}] = [\{q_0, q_1\}]$

$= [q_0, q_1]$

NFA

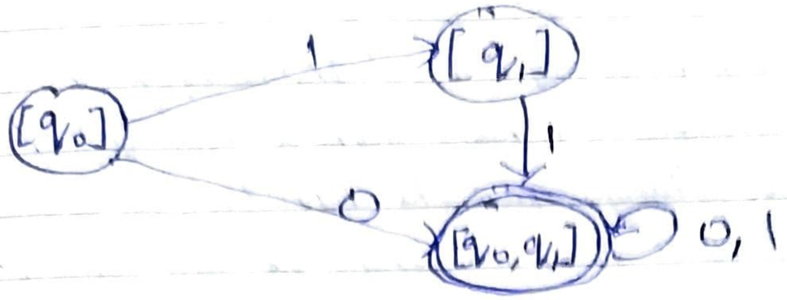
Q \ Σ	0	1
q_0	$\{q_0, q_1\}$	$\{q_1\}$
q_1	ϕ	$\{q_0, q_1\}$

DFA

Q' \ Σ	0	1
$[q_0]$	$[q_0, q_1]$	$[q_1]$
$[q_1]$	ϕ	$[q_0, q_1]$
$[q_0, q_1]$	$[q_0, q_1]$	$[q_0, q_1]$

Varun K   A016

DFA Trans. Graph.



METHOD - ~~III~~ II

In this method instead of considering the set $Q' = 2^Q$ and then removing the states that are not required. ~~This~~ The effort required for minimization is hence lesser compared to the ~~to~~ previous method.

Eg: NFA: $M = [\{q_0, q_1\}, \{0,1\}, \delta, q_0, \{q_1\}]$

NFA Table [STF]

Q \ Σ	0	1
q_0	$\{q_0, q_1\}$	$\{q_1\}$
q_1	ϕ	$\{q_0, q_1\}$

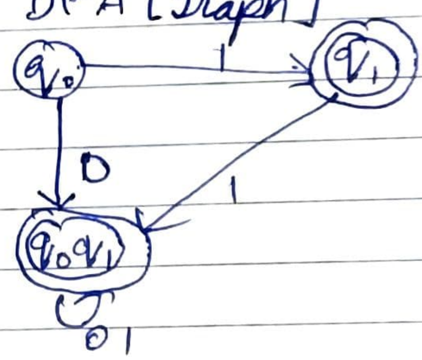
$$\delta'(q_0 q_1, 0) = \delta(q_0, 0) \cup \delta(q_1, 0)$$
$$= \{q_0, q_1\} \cup \phi$$
$$= q_0 q_1$$

$$\delta'(q_0, q_1, 1) = \delta(q_0, 1) \cup (q_1, 1)$$
$$= \{q_1\} \cup \{q_0, q_1\}$$
$$= q_0 q_1$$

In this method we consider only those states that are required.

NFA [Graph]

DFA [Graph]



VARUN K A016

METHOD-III [MyHill Nerode Theorem]

A language L is regular if and only if and only if the equivalence R_L has a finite number of equivalence classes of strings and the number of states in the smallest DFA recognizing L is equal to the number of equivalence classes in R_L.

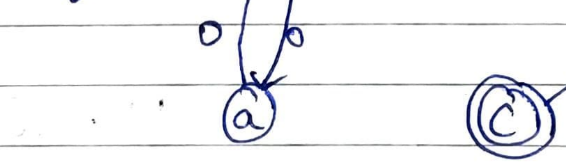
For a language L, defined over an alphabet Σ, L partitions Σ^* into distinct classes. If L generates finite number of classes then L is regular.

Eg: Let the language L of all strings, ending with b defined over $\Sigma = \{a, b\}$

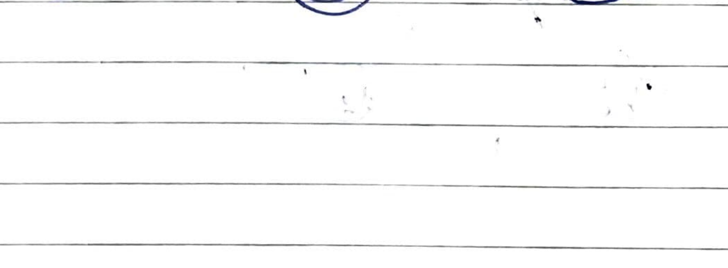
Here, since the ~~set~~ classes are finite, L is a regular language. And, the classes can ~~be~~ be defined as

C1: set of all strings ending in a

C2: set of all strings ending in b



~~DFA~~ Min Eg:



Varun K A016

S-1 Draw ~~all~~ a table for all pair of state Q_i and Q_J

Q_i \ Q_J	a	b	c	d	e	f
a						
b						
c	✓	✓				
d	✓	✓				
e	✓	✓				
f	✓	✓	✓	✓	✓	

S-2 Mark all pairs where $Q_i \in F$ and $Q_J \notin F$

S-3 If there is an unmarked pairs $[Q_i, Q_J]$ and $\delta(Q_i, A_i)$, $\delta(Q_i, A_J)$ is marked, then mark $[Q_i, Q_J]$ .... repeat the step until we cannot mark anymore

[a, b]

$\delta(a,1) = c$ $\delta(a,0) = b$

$\delta(b,1) = d$ $\delta(b,0) = a$

∵ c, d are not mark in previous table so, ∵ a, b are not mark in prev Table

So No Mark So, No Mark

[d, c]

$\delta(d,1) = f$ $\delta(d,0) = e$

$\delta(c,1) = f$ $\delta(c,0) = e$

No Mark No Mark

[e, c]

$\delta(e,1) = f$ $\delta(e,0) = e$

$\delta(c,1) = f$ $\delta(e,0) = e$

No Mark No Mark.

[e, d]

$\delta(e,1) = f$ $\delta(e,0) = e$

$\delta(d,1) = f$ $\delta(d,0) = e$

No Mark No Mark

[f, a]

$\delta(f,1) = f$ $\delta(f,0) = f$

$\delta(a,1) = c$ $\delta(a,0) = b$

Mark No Mark

[f, b]

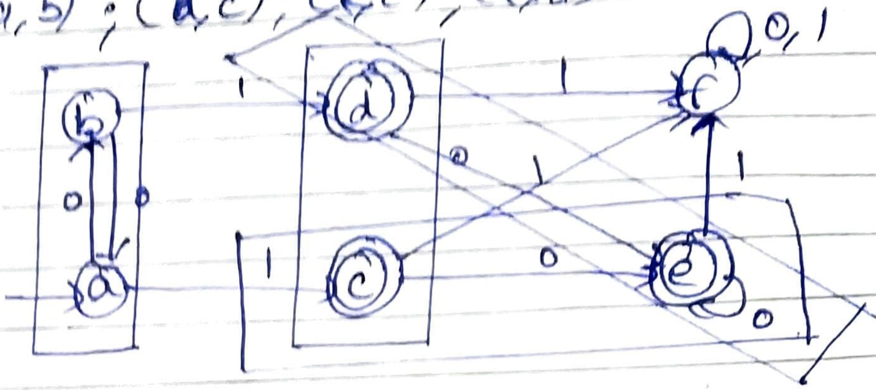
$\delta(f,1) = f$ $\delta(f,0) = f$

$\delta(B,1) = b$ $\delta(b,0) = a$

Mark No need since repeated.

Varun K   A016

S-4 Combine all the unmarked pairs (Q_i, Q_J) and make them single state in reduce DFA
(a, b) ; (d, c) ; (e, c) ; (e, d)



Reduced DFA