Krithikha Balamurugan

U19CS076
Tutorial - ~~13~~ 13

1. A language is recursive if there exists a TN that decides L.

A language is recursively enumerates if there exists a TM M that accepts L.

<u>Closure</u>

<u>Union</u>
If L1, L2 are 2 recursive languages their union L1 ∪ L2 will be recursive because if TM halts for L1, L2 It will also halt for L1 ∪ L2

Concatenation

If L1 and L2 are 2 recursive languages their concatenation L1.L2 will be recursive.

For example,

$L1 = \{a^n b^n c^n \mid n >= 0\}$
$L2 = \{d^m e^m f^m \mid m \geq 0\}$
$L3 = L1.L2$

. $\{a^n b^n c^n d^m e^m f^M \mid n >= 0 \;\&\; m \geq 0\}$ is also recursive.

Scanned with CamScanner

U19CS076

Kleen Closure

If L1 is recursive its Klene closure $L1^*$ will also be recursive

$$L1 = \{a^n b^n c^n \mid n >= 0\}$$
$$L1 = \{a^n b^n c^n \mid n >= 0\}^*$$ is also recursive

Instruction & complement

If L1, L2 are 2 recursive languages.

$$L1 = \{a^n b^n c^n d^m \mid n >= 0, m >= 0\}$$
$$L2 = \{a^n b^n c^n d^n \mid n >= 0, m >= 0\}$$
$$L3 = L1 \cap L2$$
$$= \{a^n b^n c^n d^m \mid n >= 0\}$$ will be recursive

Similarly complemented recursive language L1 which is $\Sigma^* - L1$, will also be recursive.

2. Explain NP simple and NP hard closses

NP complex problems are hardest problems in NP set

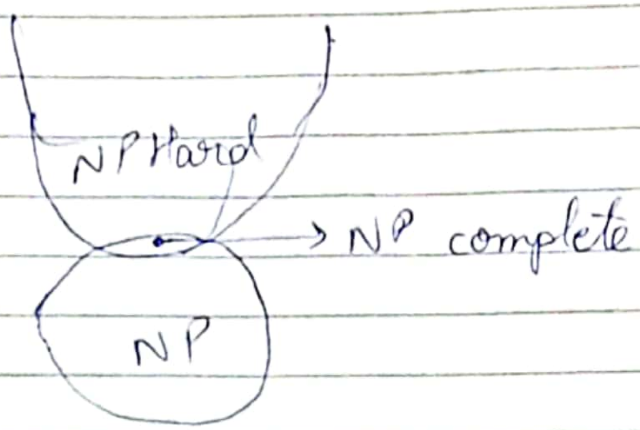
A decision problem L is NP.

- L is in NP
- Every NP problem in NP is redusible to L in polynomial time

A problem is NP hard it follows 2^{nd} property mentioned above doesnt need to follow

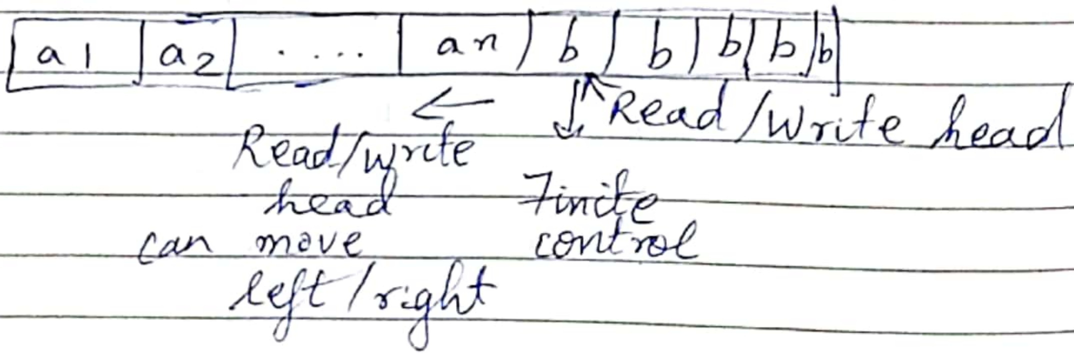
U19CS076

property 1.



3. ~~Expl~~ One tap turing machine

A turing machine consist of a tape of infinite length on which reads and writes operation can be performed.
The tape consists of infinite cells on which each cell either contains input symbol.



A TM is expressed as T. Tuple ($Q, T, B, \Sigma, \delta, q_0, F$)

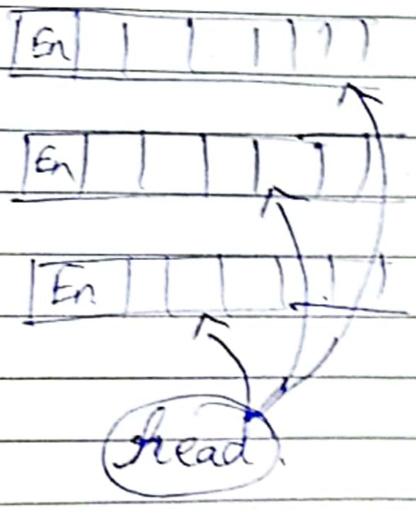
Q : Finite set of state
T : Tape of alphabet
B : Blank symbol
Σ : input ~~sy~~ alphabet
δ : $Q \to T \to Q \to T \times \{L, R\}$
q_0 : Initial state
f : Final states set

U19CS076

Multitape turing machine

Multitape turing machines have multiple tapes where each tapes accesses with a seperate head
Each head can move independently on other heads.

Initially the input on tape 1 and others are blank. At ~~fi~~ first the first tape is occupied by input and other tapes are kept blank.



A multitape turing machine can describe as 6 tuple

Q : Finite set of sets
X : Tape alphabet
B : Blank symbol
δ : retain on status & symbol.
q_0 : Initial state
F : Set of final state

U19CS076

Q4. A turing machine is said to be universal turing machine if accept
→ The input data accept
→ An alogorithm for computing

A general purpose TM will be called a universal TM if it is powerful enough to simulate behaviour of any digital computer

A universal TM can simulate behaviour of arbitary TM

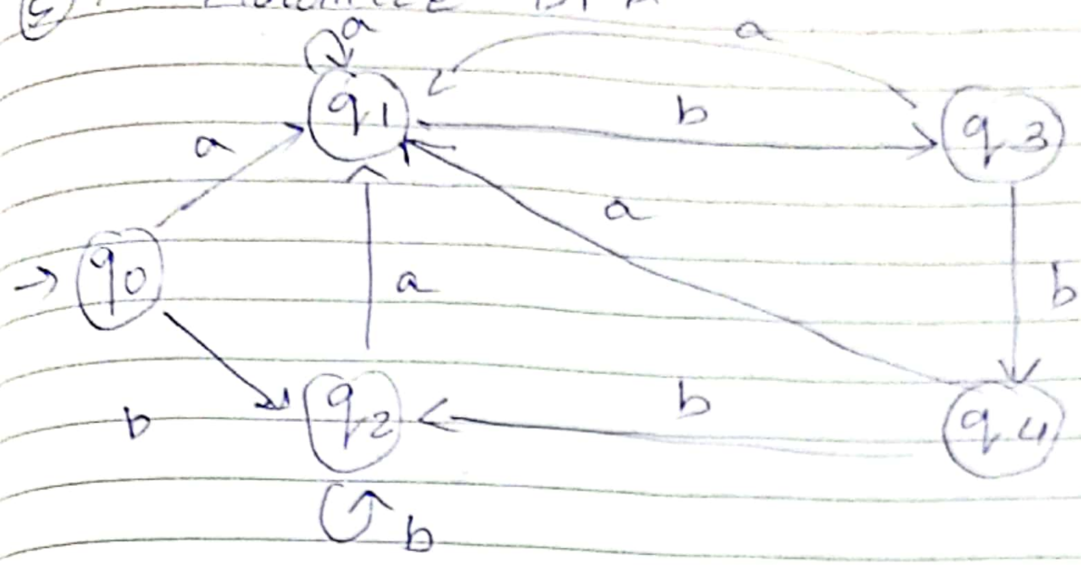
The model of universal TM is considered to be theoretical breakthrough that led to the concept at stored programm

By modyfing our basic model al a TM we can design a universal Turing machine

We modify our basic model by
→ Increase the number of read/wrile hed
→ Increase number of dimensions of input tape
→ Adding a special purpose memory.

U19CS076

5. Minimize DFA



Step 1 → Given DFA attributes no dead state inaccessible states

Step 2 → Draw a state transition table

	a	b
→ q_0	q_1	q_2
q_1	q_1	q_3
q_2	q_1	q_2
q_3	q_1	* q_4
* q_4	q_1	q_2

Step 3

Now using equivalence theorem

$P_0 = \{q_0, q_1, q_2, q_3\} \{q_4\}$

$P_1 = \{q_0, q_1, q_2\} \{q_3\} \{q_4\}$

$P_2 = \{q_0, q_2\} \{q_1\} \{q_3\} \{q_4\}$

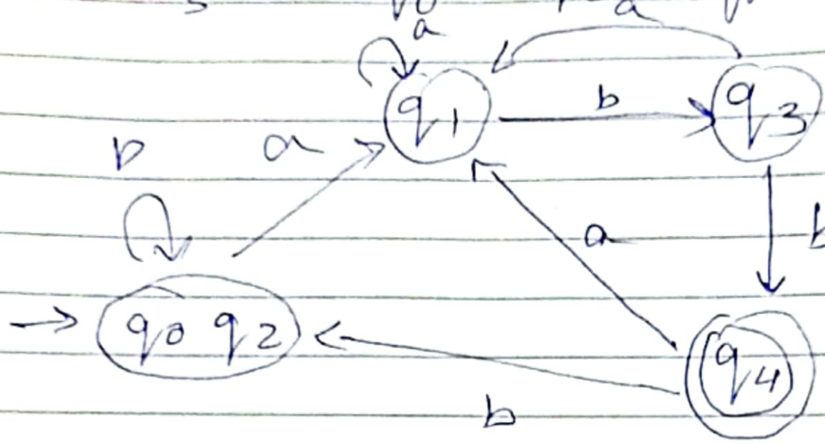
Scanned with CamScanner

U19CS076

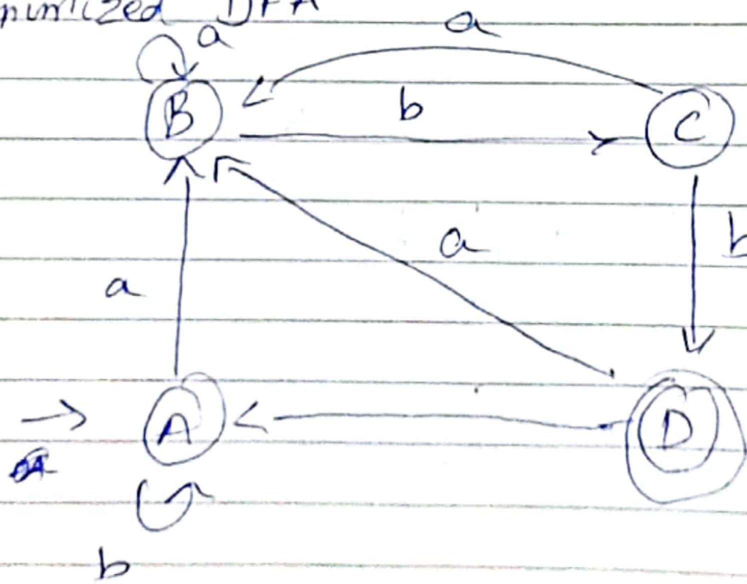
$P_3 = \{q_0, q_2\} \{q_1\} \{q_3\} \{q_4\}$

$P_2 = P_3$ so stop

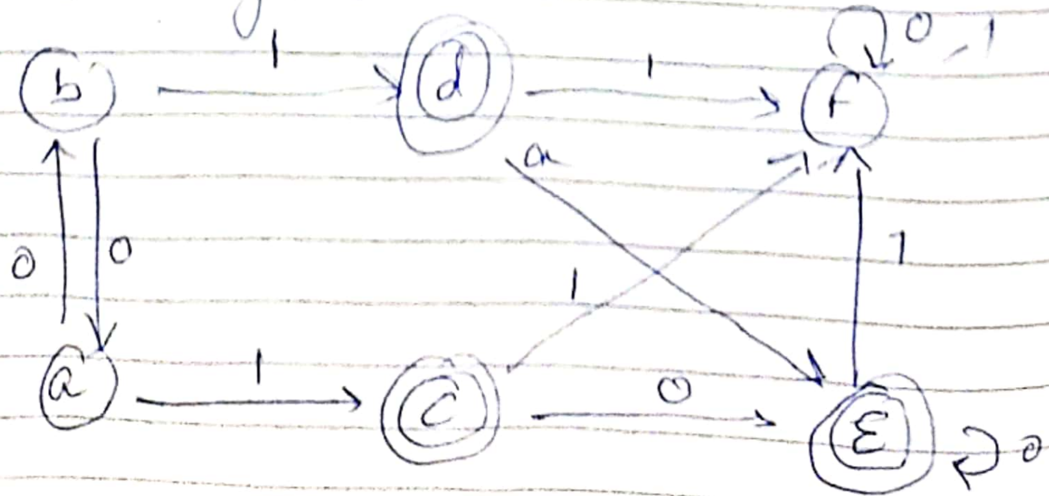
From $P_3 \Rightarrow q_0$ & q_2 equivalent, merged



Minimized DFA



(6). Minimize given DFA



U19CS076

P_0 will have two set of states. One set contain c, e, d which are final states.

So, $P_0 = \{\{c, e, d\}, \{a, b, f\}\}$

→ To calculate P_1, we check set of P_0 partition.

1) c & e are not distinguishable, c & d are also not, e & d are also not.

So $\{c, e, d\}$ not partitioned

2) So for set $\{a, b, f\}$
⇒ a, b, f not distinguishable
→ But a & f are

So set $\{a, b, f\} \rightarrow \{a, b\}, \{f\}$.

$P_1 = \{\{c, e, d\}, \{a, b\}, \{f\}\}$

3) Now for set P_2
we observe set $\{c, e, d\}$ not partitiones

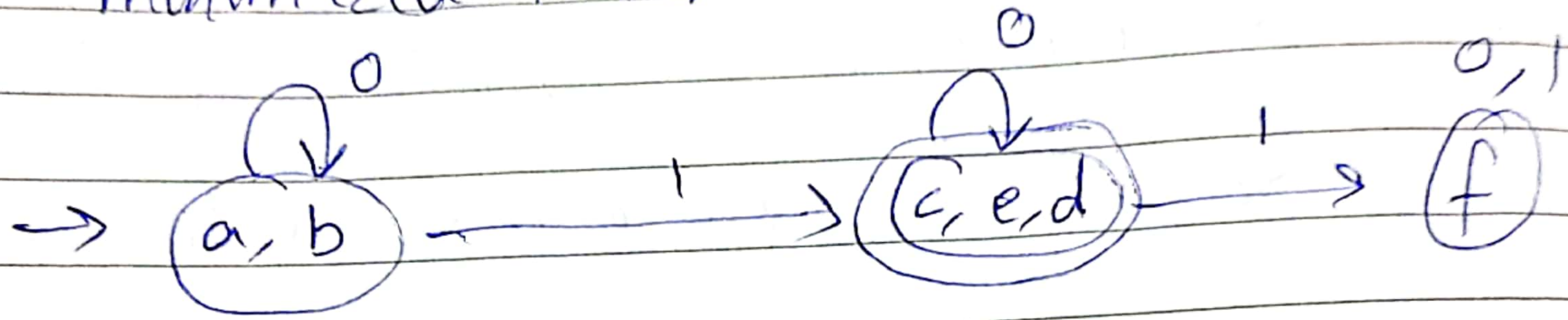
Set $\{a, b\}$ will not partition as they are also not distinguishable & $\{f\}$ same.

$P_2 = \{\{c, e, d\}, \{a, b\}, \{f\}\}$.

So this is final partition.
Partition P_2 means states c, e, d are merged into one.

U19CS076

Minimized DFA



i.e,