Name: SRIKAR KALE
Email: srikar.kale@research.iiit.ac.in
Collaborators:

Ⓕ ①

DATE

Q1.1 For a turing machine, can the i/p alphabet Σ be equal to Γ? Γ is the tape alphabet.

Ans) No it cannot
Blank symbol 'B' cannot be in the i/p alphabet set.
Even temporary tape alphabets.

Q1.2 Can there be a T.M that recognizes a non trivial language (i.e. not the empty lang and not $\{0,1\}^*$) with a single state? why/why not?

Ans) No there cannot be such turing machines.
For the T.M to execute a non trivial language string it has to remember or store a symbol or keep track of other info in states. With only one state it is not possible.

Q1.3 Consider a turing machine with 3 tapes such that it can read, write and move the head on all the tapes simultaneously. Write the formal specification (7-tuple) for such a T.M

Ans $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject} \rangle$

$Q \rightarrow$ set of states
$\Sigma \rightarrow$ i/p alphabet
$\Gamma \rightarrow$ tape symbol
→ finite sets

$\delta : Q \times \Gamma^3 \rightarrow Q \times \Gamma^3 \times \{L, R, S\}^3$

left Right stop

$\delta(q_i, a_1, a_2, a_3) = (q_j, b_1, b_2, b_3, L, R, L)$

tape1 tape2 tape3 tape1 tape2 tape3 tape1 tape2 tape3

$q_0 \rightarrow$ start state

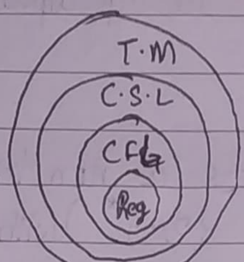
$q_{accept} \rightarrow$ acceptance state

$q_{reject} \rightarrow$ rejecting state $q_{accept} \neq q_{reject}$

$\in Q$

Q 1.4 Give an example of a language that can be recognized by a T.M but not a CFG.

Ans) The relation b/w different classes of languages is



for L must be outside the set of CFG and inside T.M set.

$L = \{a^n b^n c^n, n \geq 1\}$ is a CSL which is recognized by a T.M but not by a CFG

Q 1.5 Consider a variant of a Push down Automata where the stack is replaced by a FIFO queue. Does this change give it more power compared to usual push down automata in terms of the languages that can be recognized?

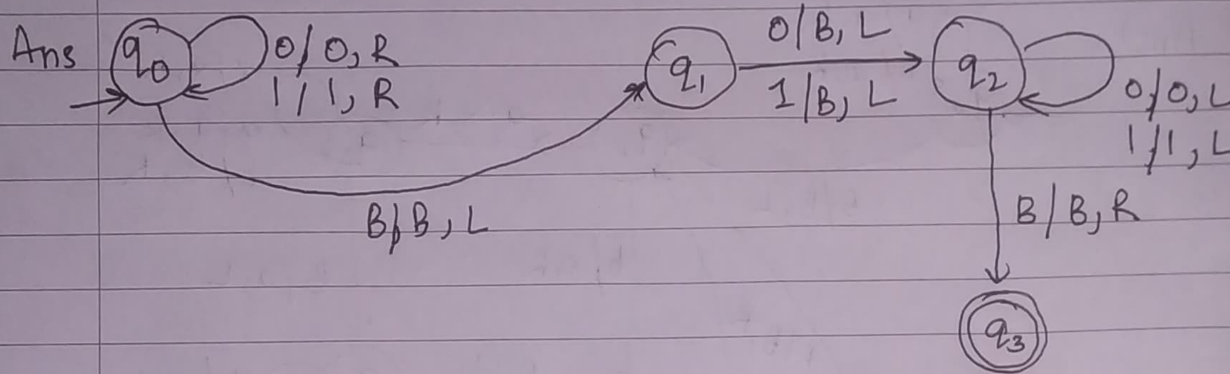
Ans Yes & No

By using a FIFO queue, we can non-deterministically solve the Language $L = \{ww \mid w \in \{0,1\}^*\}$ but at the same time we cannot solve the language $L = \{ww^R \mid w \in \{0,1\}^*\}$ which can be done non-deterministically using a PDA.

$L = \{ww \mid w \in \{0,1\}^*\}$ cannot be done using a PDA
$L = \{ww^R \mid w \in \{0,1\}^*\}$ cannot be done using a FIFO Q

2.1 Q) Construct a single tape T.M that given a no as i/p computes its quotient with 2. Assume the i/p present on the tape in binary

Ans



The T.M is defined as:

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_3, B \rangle$$

$Q = \{q_0, q_1, q_2, q_3\}$

$\Sigma = \{0, 1\}$ i/p alphabet

$\Gamma = \{0, 1, B\}$ tape alphabet

→ finite sets

$\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$

left right

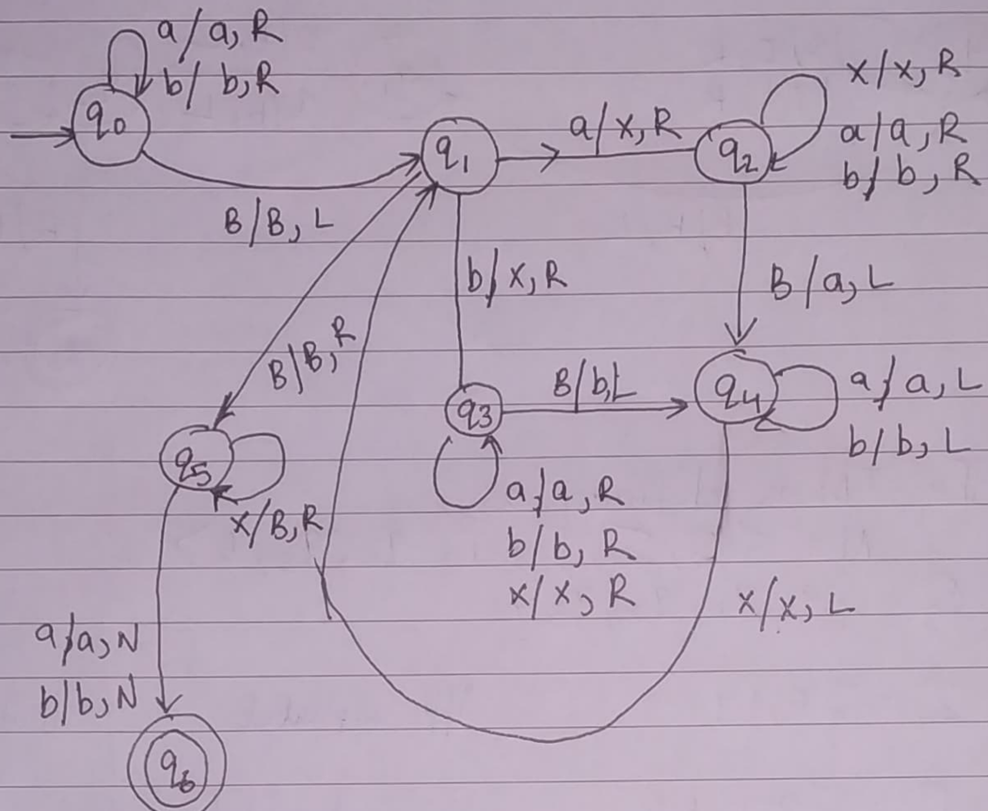
q_0 : start state

q_3 : acceptance state

B : Blank symbol.

2.1 Q) Construct a single tape T.M that reverses its input. Assume the i/p is present on the tape is binary. (eg. produces "0010111" from "1110100")

Ans)



The turing machine is defined as

$M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_6, B \rangle$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$

$\Sigma = \{a, b\}^*$ binary i/p

$\Gamma = \{a, b, x, B\}$ tape symbols

→ finite sets

$\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R, N\}$

Left Right Nothing

q_0 : start state

q_6 : Accept state

B : Blank symbol

DATE

Q 2·2) Consider a T·M with infinite 2D tape. The head can now move not only left & right but also up & down. The i/p is usually written to the right of the head position.

a) Write a detailed formal specification for this T.M. How will the transition function change? What is a configuration?

Ans) $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject} \rangle$

Q : set of states

Σ : i/p alphabets not containing Blank symbols 'B'

T : tape alphabet 'B' $\in$ T , $\Sigma \subset T$

δ : transition function

$$\delta : Q \times \Gamma \longrightarrow Q \times T \times \{ \underset{\text{left}}{L}, \underset{\text{right}}{R}, \underset{\text{up}}{U}, \underset{\text{down}}{D}, \underset{\text{stop}}{S} \}$$

q_0 : start state
q_{accept} : acceptance state
q_{reject} : rejection state

$\} \in Q$, $q_{accept} \neq q_{reject}$

Transition fn has to be modified a little bit in such a way that the head can move up & down also in the i/p tape.

Configuration:

As T.M computes through the i/p changes occur in the current state, current tape contents, and the current head location. A setting of these 3 is called the configuration of the T.M.

A configuration C_i yeilds configuration C_j if the T.M can legally go from C_i to C_j in one step.

A T.M accepts i/p w if a sequence of configurations $C_1, C_2, \ldots C_k$ exists where

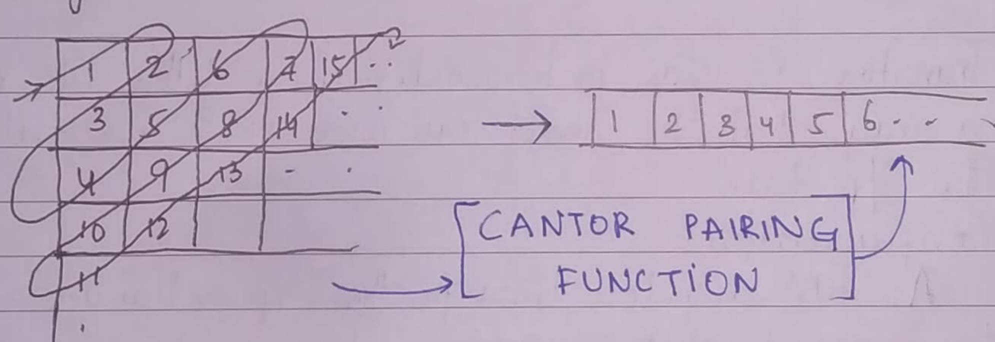
a) C_1 is the start of configuration of M on i/p w
b) Each C_i yeilds C_{i+1}
c) C_k is accepting configuration.

Q 2.2) Does the set of languages recognized by such a 2D T.M differ from that of the 3-tape T.M mentioned in Q 1.3. If so, give an example of a language that is recognized by one but not the other. If not, why? Would there be a tradeoff in such a case?

Ans) No, both T.M recognize the same set of languages. We know (from a Theorem) that any multitape T.M has an equivalent single tape T.M. So the 3-tape T.M has an equivalent single tape T.M.

Incase of a T.M with 2-D i/p tape, there is T.M with one-dimensional tape that is equally powerful and the former can be simulated by the latter.

To simulate a 2D tape with a 1-D tape, we map the squares of the 2D tape onto the 1-D tape Diagonally



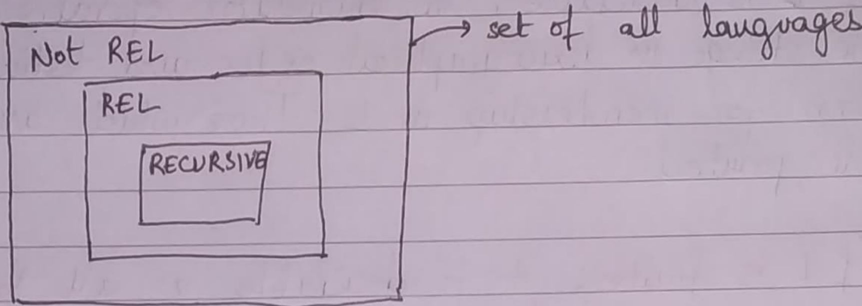
Yes, there would be a tradeoff in this case. The tradeoff is in terms of both space and running time. A single movement in the 2-D tape may take several operations to simulate in the 1-D tape T.M. Similary in the 3-tape T.M, a single operation may take several operations to simulate in 1-D T.M

Ⓕ ⑦

DATE

Q 2·3) What is a language that is recursively enumerable but not recursive?

Ans) The relationship between recursive language and recursively enumerable language is:



Recursive languages : logic exists for both valid & invalid strings
RE Languages : logic exists for valid strings
RE but not recursive languages: logic exists for only valid strings.
Not REL : logic doesnt exist for valid strings.

We want a language $L \in REL$ & $L \notin$ Recursive lang

$$L = \{ TM \mid TM \text{ accepts strings of length } 2019 \}$$

logic exists for all strings $\in L$, no logic for all strings $\notin L$

Q 2·3) What is a language that is recursive but not recursively enumerable?

Ans) By looking at the diagram above, we can say that no such languages exist.

Q 2·3) Read about enumerators from pg 180 of "Introduction to the theory of Computation, 3rd edition by micheal sipser". Read & understand the proof of Theorem 3·21 on the following page

Ans) I certify that I fully read the section on enumerators and the proof of Theorem 3·21.

DATE

Q 2.3) Using your understanding of enumerators and theorem 3.21 from the previous question, show that a language is recursive if and only if there is an enumerator that enumerates of the lang in a length increasing fashion.

Ans) If ~~A~~ is decidable, the enumerator operates by generating the strings in lexicographical order and testing each in turn for membership in L. Those which are in L are printed.

If L is finite, it is decidable as all finite languages are decidable (F.A).

If L is infinite, then on recieving an input x the T.M will enumerate all strings of L until some string y. y comes after x lexicographically. This will happen eventually as L is infinite. If x appeared in the enumeration already then accept otherwise it will never appear, so we can reject.

Let us assume L as a list of natural nos. If L is decidable then we can list all numbers in L in increasing order and test them by asking
Is $0 \in L$, $1 \in L$, ...

Note: If L is finite, then a no $m \geq \{\text{max no of } L\}$ might never appear.

$$L = \{2, 7, 10\}, \quad m = 11.$$

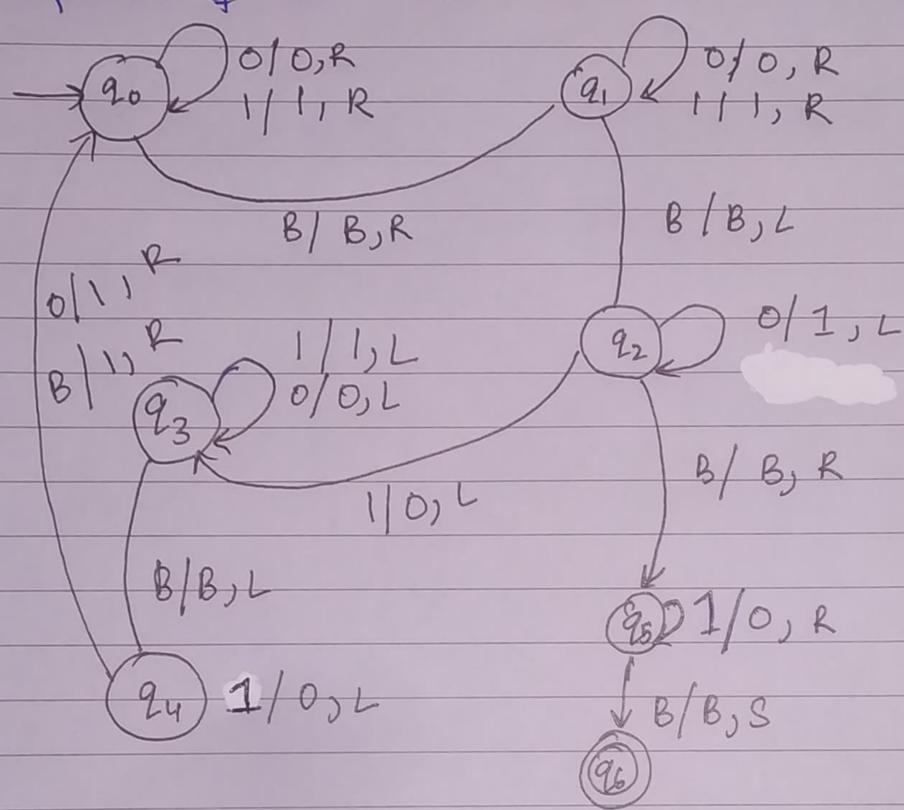
The T.M might wait forever for a larger number to appear.

Given an enumerator T.M for L, we cannot know if L is finite or infinite.

Q 3 Bonus problem:

Construct a single tape T.M that adds 2 numbers written in binary (Assume that the nos are seperated by a special symbol "+" that belongs to the external alphabet of the T.M).

Ans



Logic:

Second number is used as a counter
decrement second number by 1
increments first number by 1