Page No.:

Date:

Q6 theory

Q1) Post correspondence problem

⇒ ① Let A & B be two non-empty set of strings over Σ

② A & B are given as, $A = \{x_1, x_2, \cdots, x_n\}$ & $B = \{y_1, y_2, \cdots y_n\}$

③ We say that there is post correspondence between A & B, if there exist a sequence of one or more integers $i, j, k \ldots m$ such that the string $x_i, x_j, \ldots x_m$ is equal to $y_i, y_j, \ldots y_m$

④ Post correspondence problem with two lists:

$A = \{1, 10111, 10\}$

$B = \{111, 10, 0\}$

We will have to find a sequence using which, when elements of A and B are listed, will produce identical strings.

The required string is (2, 1, 1, 3)

i.e. $A_2 A_1 A_1 A_3 = B_2 B_1 B_1 B_3 = 101111110$

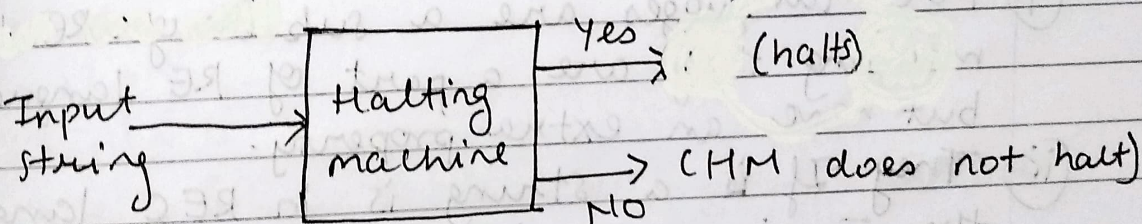
Q2) Halting problem

→ ① Input : A turing machine and an input string w.

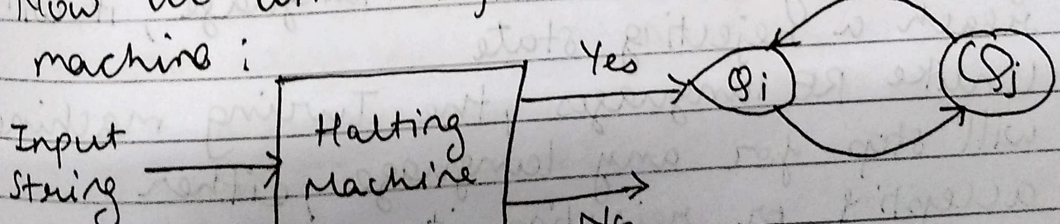
② Problem : Can we know for sure that if this machine will eventually stop (halt) when it starts processing the input ?

③ The answer must be either 'yes' (it will halt) or 'no' (it will go on forever)

④ Assumption : Let's assume that there's another machine called a 'Halting Machine', which can solve this question by saying 'yes' if the machine will halt and 'no' if it won't halt.



⑤ Now we will design an inverted halting machine :



(i) if HM says 'yes', the new machine will loop forever

(ii) if it loops forever, it should stop

⑥ This contradiction shows that the halting machine can't exist because it creates an impossible situation

⑦ So, the halting problem is undecidable.

Q3) RE and REC languages

=> (1) Recursively enumerable languages (RE)

(i) RE languages also called as type-0 languages are generated by type-0 grammar

(ii) They can be recognized by a Turing machine

(iii) If the string belongs to an RE language, the turing machine will eventually reach a final state.

(iv) If a string is not part of the RE language, the Turing machine may either enter a rejecting state or keep looping forever.

(2) Recursive languages (REC)

(i) REC languages are a subset of RE languages meaning they are a part of RE languages but have an extra property.

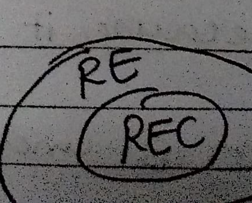
(ii) If a string is in REC language, the Turing machine will reach a final state

(iii) If a string is not in the language, it will reach a rejecting state

(iv) Unlike RE languages, the Turing machine will stop for any language, either accepting or rejecting it

(5) RE languages are called Turing recognizable languages while REC languages are called Turing decidable languages.

(4) Relationship between RE and REC languages



Page No.:

Date:

Q4) Pumping Lemma for Regular Language

=> (1) Pumping lemma gives necessary conditions for a language to be regular.

(2) It doesn't give sufficient conditions for a language to be regular

(3) Pumping lemma is used to prove that the language is not regular.

(4) Let M be a regular language and $M = \{Q, \Sigma, \delta, q_0, f\}$ be a finite automata with n states

(5) Languages will be accepted by M where $w \in L$ & $|w| \geq n$ then w can be written as xyz

where

(i) $|y| > 0$

(ii) $|xy| \leq n$

(iii) $xy^iz \in L$, $\forall i \geq 0$ and y^i denotes that y is repeated or pumped items.

(6) Steps:

(i) Assume language is regular

(ii) Choose string $w \in L$ / $|w| \geq n$

(iii) w is written as xyz with $|y| \geq 0$, $|xy| \leq n$

(iv) Now select $xy^iz \notin L$, this will contradict the assumption.

(v) Hence the language is not regular

Page No.:
Date:

Q5) Variations of turing machine.

⇒ (1) Multiple track turing machine
(i) It has multiple tracks on its tape and writes on each track one at a time
(2) Two-way infinite tape turing machine
(i) Its tape extends infinitely in both directions, allowing it to move left or right without limits.
(3) Multi-tape Turing machine.
(i) It has multiple tapes which is used by the same single head.
(4) Multiple taple multi head turing machine
(i) It has multiple tape, each with its own head for reading and writing.
(5) Multi-dimensional tape turing machine
(i) It has a multi-dimensional tape where the head can move left, right, up or down.
(6) Multi-head turing machine
(i) It has multiple heads on single tape, allowing it to read or write at different places at the same time.
(7) Non-deterministic turing machine
(i) It has a single, one-way infinite tape but can make multiple choices at each step.

Q6) Rice's theorem

⇒ ① Rice's theorem says that any non-trivial property of the language that a Turing machine recognizes is undecidable

② Key terms

(i) Property, P: A property is a feature or characteristic of a language recognized by a turing machine.

(ii) Non-trivial property: A property is non-trivial if some languages have it and others don't.

③ If a property P is non-trivial and recognized by Turing machine M, then it is impossible to decide in general if M has that property or not:

④ If a turing machine, M1, recognizes a certain language, it might share that languages with another Turing machine, M2, or it might not. Deciding this for all cases is undecideable.

⑤ There might be two machines, M1 and M2, where M1 recognizes the language and M2 does not, showing that the property is non-trivial.

Q7) Arden's theorem

⇒ ① It is used for writing regular expression from state transition diagram

② Steps:

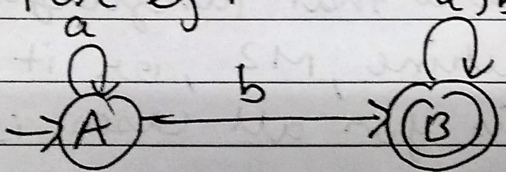
(i) Write state equation by looking at incoming transitions

(ii) Add ϵ to start state equation

(iii) Perform any combination of substitution, rearrangement and apply Arden's theorem until we are able to obtain final state in terms of alphabets.

③ Let P & Q be any 2 expressions and if P is ϵ-free then the following equation $R = Q + RP$ has a unique solution $R = QP^*$.

④ For eg:



$A = \epsilon + Aa$

comparing with $R = Q + RP$,

$A = \epsilon a^* = a^*$ ..... $(R = QP^*)$

$B = Ab + B(a+b)$

$B = a^*b + B(a+b)$

comparing with $R = Q + RP$

$B = a^*b(a+b)^*$ ..... $(R = QP^*)$