BITS Pilani
Pilani|Dubai|Goa|Hyderabad

Theory of Computation
CS F315

Vishal Gupta
Department of Computer Science and Information Systems
Birla Institute of Technology and Science
Pilani Campus, Pilani

Some Examples of Turing Machine

Let L1 = {“M” “w” | M is a Turing Machine that accepts string w}. Answer the following.

- Is L1 a Recursive language?
- Is L1 a Recursively Enumerable language?
- Is complement of L1 a Recursively Enumerable language? Why ?

Some Examples of Turing Machine

"Decidable languages are closed under UNION operation". TRUE/FALSE.

What is wrong with the following proof for above theorem?

Let T1 = (K1, ∑1, δ1, s1, {Y1, N1}) and T2 = (K2, ∑2, δ2, s2, {Y2, N2}) be two single tape Turing Machines that decide languages L1 and L2, respectively. Y1 and Y2 are accepting halting states. Similarly, N1 and N2 are non-accepting halting states. Assuming both T1 and T2 starts in initial configuration, we need to design a single tape Turing Machine T = (K, ∑, δ, s, H) that decides L = L1 U L2.

Is the following construction of T correct? Justify your answer briefly.

- T = (K, ∑, δ, s, H), where
- K = K1 U K2 U {qn} , ∑ = ∑1 U ∑2 , s = s1 , H = H1 U H2
- **∀x ∈ ∑1, ∀y ∈ ∑1 – {⊳},**
- δ = **δ1** U **δ2** U **{(N1, x)(qn, x)}** U **{(qn, y)(qn, ←)}** U **{(qn, ⊳)(s2, →)}**
- [**Note**: Transition of the form (q, x)(p, y) means in state q if the symbol read is x, write y in place of x and move to state p.]

Some Examples of Turing Machine

- Let L = {“M” | M is a Turing Machine which decides its respective language}. Give a high level description of Turing Machine U_{TM} which takes “M” $\in$ L (i.e. encoding of M) as input and outputs all halting states of M.
- Also, is $L(U_{TM})$ decidable?

Problem

Let G1 be a CFG and string $x \in L(G1)$. Also, let G2 be a CFG such that $L(G2) = L(G1) - \{x\}$. Prove that (by giving an algorithm) computing grammar G2 from grammar G1 is decidable.

Problem

- Let G1 be a CFG and string $x \in L(G1)$. Also, let G2 be a CFG such that $L(G2) = L(G1) – \{x\}$. Prove that (by giving an algorithm) computing grammar G2 from grammar G1 is decidable.
- Let L = {G | G is a CFG and G generates at least 51 strings}. Prove that L is decidable. You can give an algorithm here as a high-level description of corresponding TM.

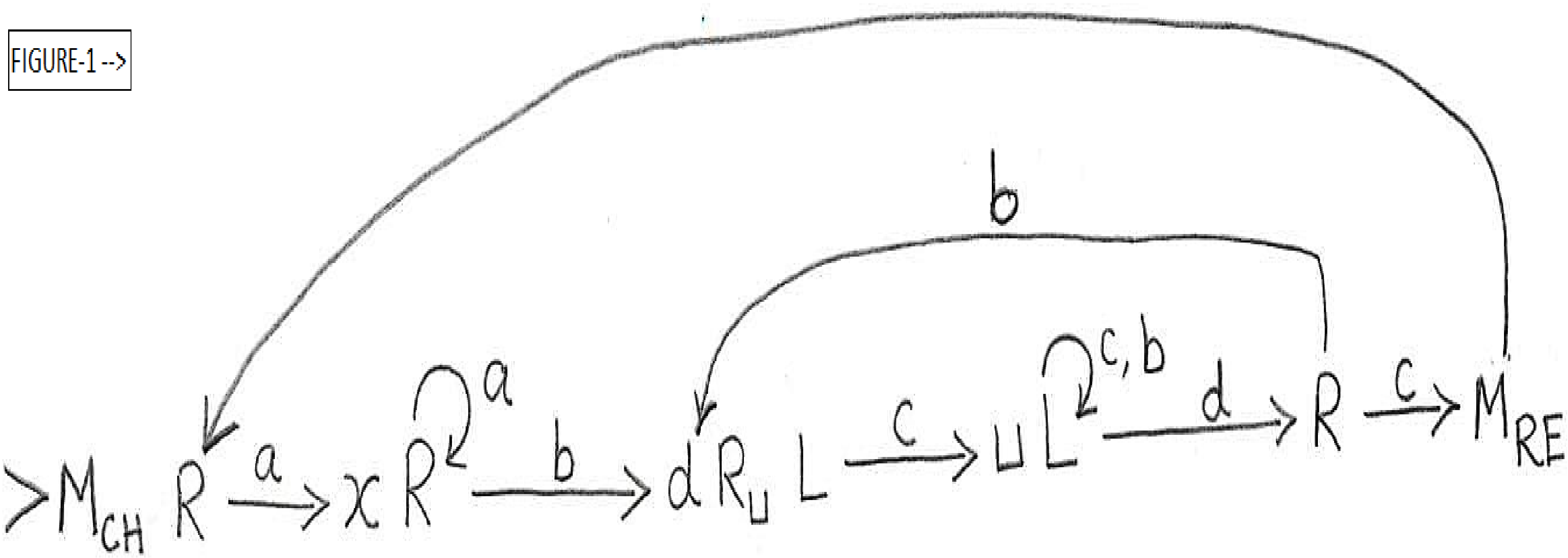
[**<u>Hint</u>**: Think about the outcome of the question just before this one.]

Some Examples of Turing Machine

- Let L = $\{a^i b^j c^k \mid i \times j = k \text{ and } i, j, k > 0\}$. Consider the single tape Turing Machine (FIGURE-1) to accept L. Its initial configuration is $\blacktriangleright \underline{\sqcup} w$ and final configuration is $\blacktriangleright \underline{\sqcup} w1$, where w1 is any string. It uses machine M_{CH} (which accepts if all a's are followed by all b's and all b's are followed by all c's; otherwise it rejects). Also it uses machine M_{RE} to restore all b's (understand yourself why it is required). Attempt the following:
- Make TM's for M_{CH} and M_{RE} so that they are meaningful in deciding language L by TM in FIGURE-1.
- Modify FIGURE-1 so that it semi-decides the language L.
- Finally, modify FIGURE-1 so that it decides the language L.

Some Examples of Turing Machine

FIGURE-1 -->



innovate achieve lead

BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Thank You