
Problem 3.16. Show that the collection of Turing-recognizable languages is closed under the operation of

Part b. concatenation.

For any two Turing-recognizable languages L_1 and L_2, let M_1 and M_2 be the **TM**s that recognize them. For a string $w = w_1w_2w_3 \cdots w_n$, define split $W = (P,\ S)$, where $P, S \in \Sigma^*$, and $w = PS$. In other words, the strings P and S are prefix and suffix of w, such that their concatenation is equivalent to w. We construct a **TM** $M^{'}$ that recognizes the concatenation of L_1 and L_2:

"On input w:

1. Calculate the list of all possible splits $W_1 = (P_1, S_1), W_2 = (P_2, S_2), \cdots W_n = (P_n, S_n)$ of w.
2. Let n be the number of splits. Repeat the following for each $i = 1, 2, 3, \cdots$.
3. Run M_1 on P_j, and M_2 on S_j alternately step by step for i steps, where $1 \leq j \leq n$.
4. If in any computation, both M_1 and M_2 accept, *accept*. "

Part c. star.

For any Turing-recognizable language L, let M be the **TM** that recognizes it. For any non-empty string $w = w_1w_2w_3 \cdots w_n$, define partition $P = (p_1,\ p_2,\ \cdots,\ p_k)$, where $1 \leq k \leq n$, each $p_i \in \Sigma^*$, and $w = p_1p_2 \cdots p_k$. We construct a **TM** $M^{'}$ that recognizes L^*:

"On input w:

1. If $w = \varepsilon$, *accept*.
2. For non-empty w, generate list of all possible partitions $P_1,\ P_2,\ P_3, \cdots$.
3. Let n be the number of partitions. Repeat the following for each $i = 1, 2, 3, \cdots$.
4. Run M on each $p \in P_j$ for i steps, where $1 \leq j \leq n$.
5. If in any computation, M accepts all $p \in P_j$, *accept*."

Part d. intersection.

For any two Turing-recognizable languages L_1 and L_2, let M_1 and M_2 be the **TM**s that recognize them. We construct a **TM** $M^{'}$ that recognizes the intersection of L_1 and L_2:

"On input string w:

1. Run M_1 and M_2 alternately on w step by step. If both accept, *accept.* If either one of M_1 or M_2 halts by rejecting, *reject.*"

If both M_1 and M_2 accepts w, $M^{'}$ accepts w because the accepting TM arrives to its accepting state after a finite number of steps. Note that if M_1 and M_2 reject and both of them do so by looping, then $M^{'}$ will loop.

Part e. homomorphism.

For any Turing-recognizable language L, and a homomorphism[1] f, let M be the **TM** that recognizes L. We construct a **TM** $M^{'}$ that recognizes $f(L)$:

"On input string w:

1. Run an enumerator E that enumerates L[2].
2. For each string $s = s_1 s_2 \cdots s_n$ generated by E, if $w = f(s_1) \cdot f(s_2) \cdots f(s_n)$, *accept.*"

[1]Problem 1.66, definition of homomorphism.
[2]Theorem 3.21