Homework 14

4396,baseline

December 18, 2019

Problem 1

Let $L_1, L_2, \ldots, L_k$ be a collection of languages over alphabet Σ such that:

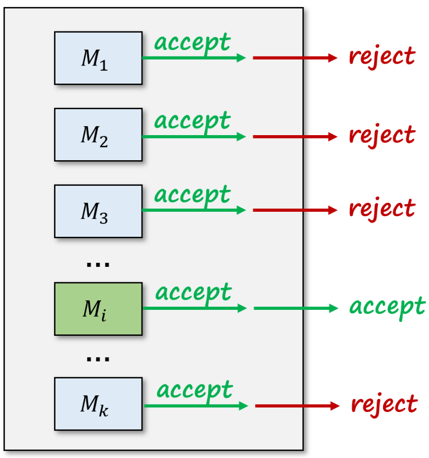
1. For all $i \neq j, L_i \cap L_j = \varnothing$;i.e., no string is in two of languages.
2. $L_1 \cup L_2 \cup \cdots \cup L_k = \Sigma^*$;i.e, every string is in one of the languages.
3. Each of the languages L_i, for $i = 1, 2, \ldots, k$ is recursively enumerable

Prove that each of the languages is recursive.

Solution.

Since L_i is recursively enumerable, so there exist Turing machines M_i for $L_i(i = 1, 2, ...k)$ such that if $L(M_i) = L_i$.

For any L_i, we can build a decider D based on $M_1, M_2...M_k$. We simultaneously simulate $M_1, M_2, ...M_k$. For any $w \in \Sigma*$, if $M_j(j \neq i)$ accepts w, then D rejects w; if M_i accepts w, then D accepts w.



Since $L_1 \cup L_2 \cup \cdots \cup L_k = \Sigma^*$ and $i \neq j, L_i \cap L_j = \varnothing$, for any $w \in \Sigma*$, there's one and only one L_j such that $w \in L_j$, it means there's always a certain M_j accepts w. So D always halts. And $L(D) = L(M_i) = L_i$, so D is a decider for L_i, or L_i is recursive.

Q.E.D

Problem 2

Let L_1, L_2 be two recursive languages, give an informal, but clear, construction to show that the concatenation of two languages $L_3 = \{w_1 w_2 | w_1 \in L_1 \land w_2 \in L_2\}$ is also recursive.

Solution.

For an input string w of length n, we can generate $n+1$ string pairs from w, and donate the ith pair as $p_i = \langle w[0:i], w[i:n] \rangle$, where $w[i:j]$ means the string from ith symbol(included) to jth symbol(excluded) of w.

As L_1 and L_2 are recursive, then there exists decider D_1, D_2 such that $L(D_1) = L_1$ and $L(D_2) = L_2$. Then We can build a decider D_3 for L_3 like follows: First we generate $n+1$ pairs $p_0, p_1 ... p_n$ from the input w of length n. Then we feed $p_i[0], p_i[1]$ to D_1, D_2 from $i = 0$ to $i = n$ respectively. If D_1, D_2 accept certain p_i, then we accept w; otherwise, we'll reject w.

Notice that if $w \in L_3$, then there exist at least one pair p_i such that $p_i[0] \in L_1$ and $p_i[1] \in L_2$, so D_3 will always accept w. If $w \notin L_3$, no pair can be accepted by D_1, D_2 simultaneously, then D_3 rejects w.

D_3 is a decider for L_3, so L_3 is recursive.

Problem 3

Encoding the TM to binary string(You may need to reassign the states):

	0	1	B
$\to q_0$	$(q_0, 1, R)$	$(q_0, 0, R)$	(q_1, B, L)
q_1	$(q_2, 0, L)$	-	-
$*q_2$	-	-	-

Solution.

$q_0 \to q_1, q_1 \to q_3, X_1 = 0, X_2 = 1, X_3 = B, D_1 = L, D_2 = R$

	X_1	X_2	X_3
$\to q_1$	(q_1, X_2, D_2)	(q_1, X_1, D_2)	(q_3, X_3, D_1)
$*q_2$	-	-	-
q_3	(q_2, X_1, D_1)	-	-

Encode the transition function as:

$\delta(q_1, X_1) = (q_1, X_2, D_2) \to 01010100100$

$\delta(q_1, X_2) = (q_1, X_1, D_2) \to 01001010100$

$\delta(q_1, X_3) = (q_3, X_3, D_1) \to 010001000100010$

$\delta(q_3, X_1) = (q_2, X_1, D_1) \to 000101001010$

So the code for this TM is:

0101010010011010010101001101000100010001011000101001010