Problem

Next

For strings w and t, write $w \doteq t$ if t and w have the same symbols in the same quantities, but possibly in a different order.

For any string w, define $SCRAMBLE(w) = \{t| \ t \doteq w\}$. For any language A, let $SCRAMBLE(A) = \{t| \ t \in SCRAMBLE(w) \text{ for some } w \in A\}$.

a. Show that if ∑ = {0,1}, then the *SCRAMBLE* of a regular language is context free.

b. What happens in part (a) if ∑ contains three or more symbols? Prove your answer.

Step-by-step solution

Step 1 of 4

For any two strings *w* and *t*, *SCRAMBLE* is defined as, $SCRAMBLE(w) = \{t \mid t \square w\}$. Consider a language A, $SCRAMBLE(A) = \{t \mid t \in SCRAMBLE(w) \text{ for some } w \in A\}$.

Comment

Step 2 of 4

a.

Consider the input alphabet $\Sigma = \{0,1\}$. Assume the language $L = (10)^*$ over the input alphabet $\Sigma = \{0,1\}$. The *SCRAMBLE*(*w*) can have all the word permutations of *w*. The states in FA for the language *L* must be rearranged to accept the *SCRAMBLE* of the language.

The intersection of *SCRAMBLE*(*L*) and a regular language need to be regular if *SCRAMBLE*(*L*) is regular. Consider a regular language $M = 1^*0^*$. The intersection of *SCRAMBLE*(*L*) and *M* is, $SCRAMBLE(L) \cap 1^*0^* = \{1^n0^n : n \geq 0\}$ which is context free language. Thus, the *SCRAMBLE*(*L*) cannot be a regular language.

Comment

Step 3 of 4

Now, using push down automata *P* to show that scramble $SCRAMBLE(A)$ of regular language *A* is context free. The Push down automata *P* contains set of states similar to DFA *D* which accepts *A*. Consider the input $w = w_1w_2w_3 \cdots w_n$, the push down automata *P* non-deterministically works to guess the permutation u of w in A regular language. When it guesses $w_1 = u_1$ then transition occurs by reading the input w_1 like *D*. When it guesses $w_1 \neq u_1$ then null transition occurs. The input symbol u_1 stored in stack for the future use.

When number of elements in the stack is greater than 1 then the machine works as follows:

If the input symbol is same as the top element of the stack, then it consumes both the symbol but does not make any transition. If the input symbol does not matches with the top element of the stack then it guesses that may be it is the next right element in *u* or guesses that current input symbol is wrong and make null transition while push the complement of the symbol in the stack. The symbol that is inserted in the stack will pay back as soon as possible. For language *A,* if *w* is a permutation for string *u* then there must be a path that lead to the accept state with empty state. This will be the acceptance conditions for PDA *P.*

Therefore, the SCRAMBLE of a regular language is context free.

Comment

Step 4 of 4

b.

Consider the input alphabet $\Sigma = \{0,1,2\}$. Assume the language $L = (210)^*$ over the input alphabet $\Sigma = \{0,1,2\}$.

The intersection of *SCRAMBLE*(*L*) and a regular language need to be regular if *SCRAMBLE*(*L*) is regular. Consider a regular language $M = 2^*1^*0^*$.

The intersection of *SCRAMBLE*(*L*) and *M* is, $SCRAMBLE\left(L\right) \cap 2^*1^*0^* = \{2^n1^n0^n : n \geq 0\}$ which is not a ntext free language.

Next

The pushdown automata has a single stack which can be used to compare the occurrences of two symbols, but for 3 or more symbols PDA cannot be drawn.

Therefore, the SCRAMBLE of a regular language with 3 or more symbols is not context free.

Comment