Formal Languages and Automata Theory
16CS52
Open Book Assignment 2

USN	Question
2GI14CS038	Show that if L is regular, so is L^R .
2GI14CS040	Let L be any language. Define *even*(w) as the string obtained by extracting from w the letters in even-numbered positions; that is, if $w = a_1a_2a_3a_4 \ldots$, then *even*(w)= $a_2a_4 \ldots$. Corresponding to this, we can define a language *even*(L) = { *even*(w): $w \in L$}. Prove that if L is regular, so is *even*(L).
2GI14CS067	From a language L we create a new language *chop2*(L) by removing the two leftmost symbols of every string in L. Specifically, *chop2*(L) = {w: $vw \in L$, with $\|v\|= 2$}. Show that if L is regular, then *chop2*(L) is also regular.
2GI14CS133	Find minimal DFA for the following languages and prove that the result is minimal: $L = \{ a^nb^m : n \geq 2, m \geq 1 \}$
2GI15CS048	Find minimal DFA for the following languages and prove that the result is minimal: $L = \{ a^nb : n \geq 0 \} \cup \{ b^na : n \geq 1 \}$
2GI15CS058	Find minimal DFA for the following languages and prove that the result is minimal: $L = \{ a^n : n \geq 0 , n \neq 3 \}$
2GI15CS117	Find minimal DFA for the following languages and prove that the result is minimal: $L = \{ a^n : n \neq 2 \text{ and } n \neq 4 \}$
2GI15CS186	Find minimal DFA for the following languages and prove that the result is minimal: $L = \{ a^n : n \bmod 3 = 0 \} \cup \{ a^n : n \bmod 5 = 1 \}$
2GI17CS400	Show that if L is a nonempty language such that any w in L has length at least n, then any DFA accepting L must have at least $n + 1$ states.
2GI17CS401	Prove or disprove the following conjecture. If $M = (Q, \Sigma, \delta, q_0 ,F)$ is a minimal DFA for a regular language L, then $\widehat{M} = (Q, \Sigma, \delta, q_0 ,Q - F)$ is a minimal DFA for $\overline{L}$
2GI17CS402	Show that indistinguishability is an equivalence relation but that distinguishability is not.

2GI17CS403	Prove the following: If the states q_a and q_b are indistinguishable, and if q_a and q_c are distinguishable, then q_b and q_c must be distinguishable.
2GI17CS404	Show that given a regular language L, its minimal DFA is unique within a simple relabeling of the states.
2GI17CS405	What languages do the expressions $(\emptyset^*)^*$and $a\emptyset$ denote?
2GI17CS406	For the case of a regular expression r that does not involve λ or $\emptyset$, give a set of necessary and sufficient conditions that r must satisfy if $L(r)$ is to be infinite.
2GI17CS407	In some applications, such as programs that check spelling, we may not need an exact match of the pattern, only an approximate one. Once the notion of an approximate match has been made precise, automata theory can be applied to construct approximate pattern matchers. As an illustration of this, consider patterns derived from the original ones by insertion of one symbol. Let L be a regular language on Σ and define $insert(L) = \{ uav : a \in \Sigma , uv \in L \}$. In effect, $insert(L)$ contains all the words created from L by inserting a spurious symbol anywhere in a word. Given an NFA for L, show how one can construct an NFA for $insert(L)$.
2GI17CS408	In some applications, such as programs that check spelling, we may not need an exact match of the pattern, only an approximate one. Once the notion of an approximate match has been made precise, automata theory can be applied to construct approximate pattern matchers. As an illustration of this, consider patterns derived from the original ones by insertion of one symbol. Let L be a regular language on Σ and define $insert(L) = \{ uav : a \in \Sigma , uv \in L \}$. In effect, $insert(L)$ contains all the words created from L by inserting a spurious symbol anywhere in a word. How you might write a pattern-recognition program for $insert(L)$, using as input a regular expression for L.
2GI17CS409	Suggest a construction by which a left-linear grammar can be obtained from an NFA directly.
2GI17CS410	Show that for every regular language not containing λ there exists a right-linear grammar whose productions are restricted to the forms $A \to aB$, or $A \to a$, where $A, B \in V$, and $a \in T$.
2GI17CS411	Show that any regular grammar G for which $L(G) \neq \emptyset$ must have at least one production of the form $A \to x$, where $A \in V$ and $x \in T^*$.

2GI17CS412	Let $G_1 = (V_1, \Sigma, S_1, P_1)$ be right-linear and $G_2 = (V_2, \Sigma, S_2, P_2)$ be a left-linear grammar, and assume that V_1 and V_2 are disjoint. Consider the linear grammar $G = (\{S\} \cup V_1 \cup V_2, \Sigma, S, P)$, where S is not in $V_1 \cup V_2$ and $P = \{ S \to S_1 \mid S_2 \} \cup P_1 \cup P_2$. Show that $L(G)$ is regular.
2GI17CS413	Show that the family of regular languages is closed under finite union and intersection.
2GI17CS414	The symmetric difference of two sets S_1 and S_2 is defined as $S_1 \ominus S_2 = \{ x : x \in S_1$ or $x \in S_2$, but x is not in both S_1 and $S_2 \}$. Show that the family of regular languages is closed under symmetric difference.
2GI17CS415	The *nor* of two languages is $nor(L_1, L_2) = \{ w : w \notin L_1$ and $w \notin L_2 \}$. Show that the family of regular languages is closed under the *nor* operation.
2GI17CS416	The c*or* of two languages is c*or*$(L_1, L_2) = \{ w : w \in \overline{L_1}$ or $w \in \overline{L_2} \}$. Show that the family of regular languages is closed under the *cor* operation.
2GI17CS417	Show that $L_1 = L_1L_2 / L_2$ is not true for all languages L_1 and L_2.
2GI17CS418	Suppose we know that $L_1 \cup L_2$ is regular and that L_1 is finite. Can we conclude from this that L_2 is regular?
2GI17CS419	If L is a regular language, prove that $L_1 = \{ uv : u \in L, \lvert v \rvert = 2 \}$ is also regular.
2GI17CS420	If L is a regular language, prove that the language $\{ uv : u \in L, v \in L^R \}$ is also regular.
2GI17CS421	The left quotient of a language L_1 with respect to L_2 is defined as $L_2 \backslash L_1 = \{ y : x \in L_2, xy \in L_1 \}$. Show that the family of regular languages is closed under the left quotient with a regular language.
2GI17CS422	Show that if the statement "If L_1 is regular and $L_1 \cup L_2$ is also regular, then L_2 must be regular" were true for all L_1 and L_2, then all languages would be regular.
2GI17CS423	The *tail* of a language is defined as the set of all suffixes of its strings, that is, $tail(L) = \{ y : xy \in L$ for some $x \in \Sigma^* \}$. Show that if L is regular, so is $tail(L)$.

2GI17CS424	The *head* of a language is the set of all prefixes of its strings, that is, $head(L) = \{ x : xy \in L \text{ for some } y \in \Sigma^* \}$. Show that the family of regular languages is closed under this operation.
2GI17CS425	Define an operation *third* on strings and languages as $\text{third}(a_1a_2a_3a_4a_5a_6 \ldots) = a_3a_6 \ldots$ with the appropriate extension of this definition to languages. Prove the closure of the family of regular languages under this operation.
2GI17CS426	For a string $a_1a_2...a_n$ define the operation shift as $shift(a_1a_2...a_n) = a_2...a_na_1$, From this, we can define the operation on a language as $shift(L) = \{ v : v = shift(w) , \text{ for some } w \in L \}$. Show that regularity is preserved under the shift operation.
2GI17CS427	Define $exchange(a_1a_2...a_{n-1}a_n) = a_na_2...a_{n-1}a_1$, and $exchange(L) = \{ v : \text{v} = \text{exchange(w) for some } w \in L \}$. Show that the family of regular languages is closed under exchange.
2GI17CS428	The *shuffle* of two languages L_1 and L_2 is defined as $shuffle(L_1 , L_2) = \{ w_1v_1w_2v_2...w_mv_m : w_1w_2...w_m \in L_1 , v_1v_2...v_m \in L_2$, for all $w_i , v_i \in \Sigma^* \}$. Show that the family of regular languages is closed under the shuffle operation.
2GI17CS429	Define an operation *minus5* on a language *L* as the set of all strings of *L* with the fifth symbol from the left removed (strings of length less than five are left unchanged). Show that the family of regular languages is closed under the *minus5* operation.
2GI17CS430	Define the operation *leftside* on *L* by $leftside(L) = \{ w : ww_R \in L \}$. Is the family of regular languages closed under this operation?
2GI17CS431	The *min* of a language *L* is defined as $min(L) = \{ w \in L : \text{there is no } u \in L , v \in \Sigma+ , \text{ such that } w = uv \}$. Show that the family of regular languages is closed under the *min* operation.
2GI17CS432	Let G_1 and G_2 be two regular grammars. Show how one can derive regular grammars for the languages $L(G_1) \cup L(G_2)$, $L(G_1) . L(G_2)$ and $L(G_1)^*$ from G_1 and G_2 .
2GI17CS433	Show that there exists an algorithm to determine whether or not $w \in L_1 - L_2$, for any given *w* and any regular languages L_1 and L_2 .
2GI17CS434	Show that there exists an algorithm for determining if $L_1 \subseteq L_2$, for any regular languages L_1 and L_2 .

2GI17CS435	Show that there exists an algorithm for determining if $\lambda \in L$, for any regular language L .
2GI17CS436	Show that for any regular L_1 and L_2 there is an algorithm to determine whether or not $L_1 = L_1 / L_2$.
2GI17CS437	A language is said to be a *palindrome* language if $L = L^R$. Find an algorithm for determining if a given regular language is a *palindrome* language.
2GI17CS438	Exhibit an algorithm for determining whether or not a regular language L contains any string w such that $w^R \in L$.
2GI17CS439	Exhibit an algorithm that, given any three regular languages , L, L_1 , L_2 , determines whether or not $L = L_1L_2$
2GI17CS440	Exhibit an algorithm that, given any regular language L , determines whether or not $L = L^*$.
2GI17CS441	Let L be any regular language on $\Sigma = \{ a , b \}$. Show that an algorithm exists for determining if L contains any strings of even length.
2GI17CS442	Find an algorithm for determining whether a regular language L contains an infinite number of even-length strings.
2GI17CS443	Describe an algorithm which, when given a regular grammar G, can tell us whether or not $L(G) = \Sigma^*$.