School of Mathematical and Computing Sciences
COMP 202 : Formal Methods of Computer Science

Test (19 August, 1999)

Time allowed = 90 minutes
Answer ALL questions

There are 100 points possible. When a problem involves several steps in applying a procedure, it will be to your advantage to show your work so that proper partial credit can be assigned in the event of a minor error in applying the procedure. (Problems 3 and 4 are of this sort.)

1. Using and understanding regular expressions. (15 points total)

1(a). (5 points) Consider the regular expression

$$r = ((a + \lambda)a^*b)^*a$$

For each of the following words, determine whether the word belongs to the regular language described by r.

(1) the word λ
(2) the word a
(3) the word b^3a
(4) the word $baba$
(5) the word $abab$

1(b). (10 points) Give a regular expression that describes the language L (over the alphabet $\Sigma = \{a, b\}$) consisting of all words that *do not* begin with aab. That is,

$$L = \{x \in \{a, b\}^* : x \neq aaby,\ y \in \Sigma^*\}$$

Examples of words that belong to L include: $\lambda, abab, bb$ and $baab$. Examples of words that do not belong to L include: $aab, aabab, aabbb$ and $aabbbab$.

2. Designing a DFA (20 points). Find a DFA that recognizes the language L over the alphabet $\Sigma = \{a, b\}$ consisting of all words that avoid having three consecutive a's:

$$L = \{x \in \Sigma^* : x \neq uaaav,\ u, v \in \Sigma^*\}$$

Examples of words that belong to L include: $aaba, aabbaa, bbb$ and $abab$. Examples of words that do not belong to L include: $aaa, bbabaaab, bbaaab$ and $babaaabbbb$. Make every state of your DFA explicit (i.e., no implicit *dead state*). Hint: your DFA should have 4 states.

3. Translating a regular expression into an NFA (20 points). Give a diagram of an NFA that accepts the regular language described by the regular expression in 1(a).

4. Converting a DFA to an NFA (20 points). Find a DFA that accepts the same language as the NFA described by the following information:

(1) The set of states of the NFA is $Q = \{1, 2, 3, 4, 5\}$.
(2) The start state of the NFA is 1.
(3) The set F of accept states of the NFA is $F = \{1\}$.
(4) The transition relation is given by the following table:

	λ	a	b
1	$\{2\}$	$\{5\}$	$\{1\}$
2	$\{3\}$	$\{4\}$	$\emptyset$
3	$\emptyset$	$\emptyset$	$\{1\}$
4	$\{5\}$	$\emptyset$	$\{3\}$
5	$\emptyset$	$\{2\}$	$\{4\}$

It is easy to make mistakes, so show your work in calculating the DFA using the subset construction. If time allows, it is a good idea to test your construction. (Hint: a correct solution has 6 states.)

5. Understanding and using the Myhill-Nerode Theorem to demonstrate nonregularity. (15 points total)

5(a) (6 points) Let L be the language over the alphabet $\Sigma = \{a, b\}$ consisting of all words x for which the number of a's in x is equal to the number of b's in x:

$$L = \{x \in \Sigma^* : n_a(x) = n_b(x)\}$$

Examples of words that belong to L include: $abba, abbaab, \lambda$ and $bbabaaba$. Examples of words that do not belong to L include: bba, aba, aaa and $ababa$. For each of the following pairs of words x and y, determine if xR_Ly.
(1) $x = aa$ and $y = aaa$
(2) $x = aab$ and $y = ababa$
(3) $x = \lambda$ and $y = ab$

5(b) (9 points) Let L for this problem denote the language over the alphabet $\Sigma = \{a, b, c\}$ consisting of all words x such that the number of a's in x is equal to the sum of the number of b's in x and the number of c's in x:

$$L = \{x \in \Sigma^* : n_a(x) = n_b(x) + n_c(x)\}$$

Give a proof using the Myhill-Nerode Theorem that L is not regular.

6. Using the Myhill-Nerode Theorem to demonstrate regularity (10 points). In this problem we consider the alphabet to be $\Sigma = \{a, b\}$. The language L that we are interested in consists of all words x such that the first three letters of x are the same as the last three letters of x. (If the length of x is less than three, for example $x = ba$, then we define first3(x) and $last3(x)$ by using a special symbol B to denote a blank position. Thus first3$(ba) = baB$ and last3$(ba) = Bba$ and first3$(\lambda) = BBB =$ last3(λ). Examples of words that belong to L include: $\lambda, aaa, bbb, aabbaab$ and $ababa$. Examples of words that do not belong to L include: $abbaab, bababbb, baab$ and $abaab$.

In order to prove that this language is regular we argue as follows. Define the relation R on Σ^* by xRy if and only if first3$(x) =$ first3(y) and last3$(x) =$ last3(y). It is easy to argue that this is an equivalence relation. In order to apply the Myhill-Nerode Theorem to conclude that L is regular, we have to argue two more things:
(1) That R is a right congruence, that is, for any $x, y, z \in \Sigma^*$, if xRy then $xzRyz$.
(2) That L is a union of equivalence classes of R, that is, if xRy and $x \in L$ then $y \in L$.

Give arguments for these two points. (Hint: both arguments are easy and short if you understand what is required.)