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

Test (14 September, 2000)

Time allowed = 90 minutes
Answer ALL questions

There are 80 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.

1. Understanding regular expressions (15 points total).

1(a). (6 points) In each case, find a string of minimum length in $\{0,1\}^*$ **not** in the language corresponding to the given regular expression.

(i) $1^*(01)^*0^*$

(ii) $(0^*|1^*)(0^*|1^*)(0^*|1^*)$

(iii) $0^*(100^*)^*1^*$

1(b). (9 points) Find a regular expression corresponding to each of the following subsets of $\{0,1\}^*$.

(i) The language of all strings that begin or end with 00 or 11.

(ii) The language of all strings that do not end with 01.

(iii) The language of all strings containing exactly one occurrence of the string 00. (The string 000 should be viewed as containing two occurrences of 00.)

2. Designing DFA's (20 points total).

(a) **(10 points)** Find a DFA that recognises the language $L(\alpha)$ over the alphabet $\Sigma = \{a, b\}$ where α is the the regular expression $(aa|bb)^*(ab|ba)(aa|bb)^*$.

Make your DFA complete (i.e., no implicit *dead state*).

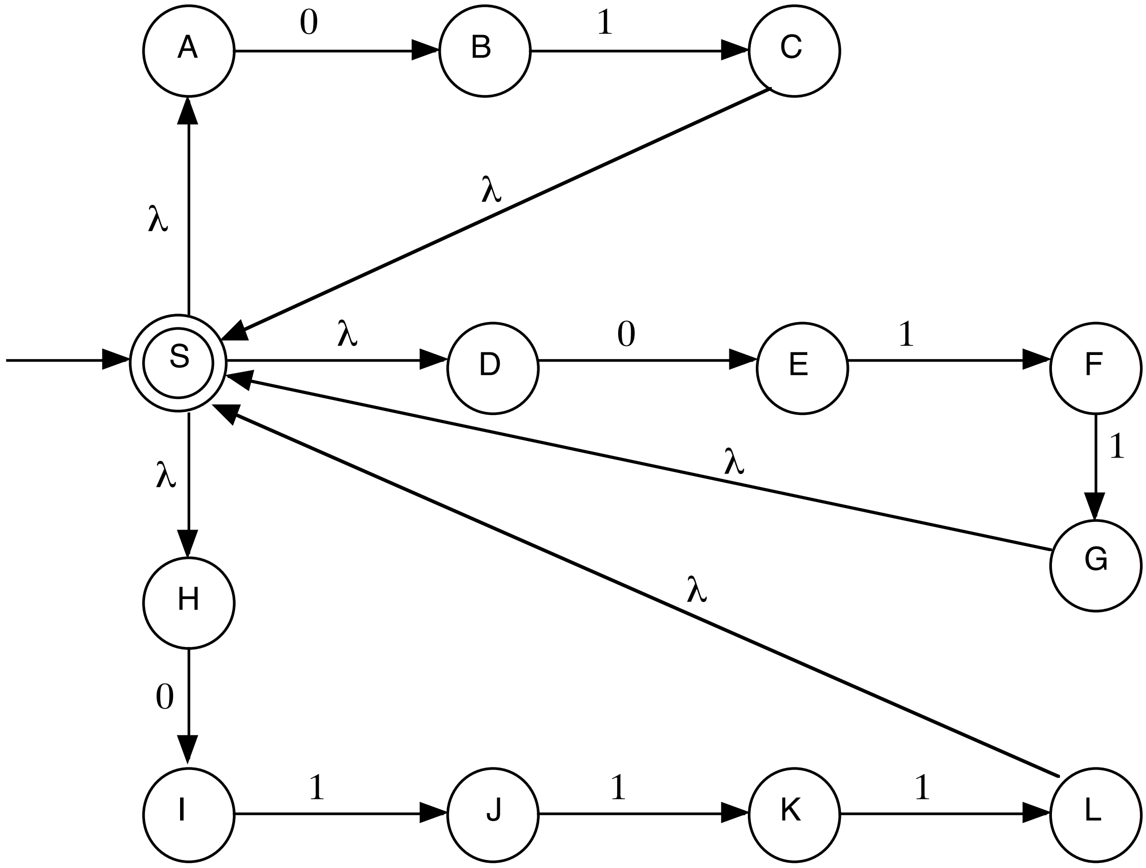
(b) **(10 points)** Find a DFA that recognizes the language $L(\alpha)$ over the alphabet $\Sigma = \{a, b\}$ where α is the the regular expression $(aaa)^*b|(aa)^*b$.

Make your DFA complete (i.e., no implicit *dead state*). Hint : The product construction may help you in designing this DFA.

3. Designing an NFA (5 points). Find an NFA, with only four states, that accepts the regular language over the alphabet $\Sigma = \{0, 1\}$ described by the following regular expression.

$(01|011|0111)^*$

4. Converting an NFA-λ to a DFA (10 points). Here is an NFA-λ automaton that also accepts $L((01|011|0111)^*)$.



Convert this automaton to an equivalent (complete) deterministic one using the extended version of the subset construction, where we calculate the λ-closure of the current state at each step. Show clearly which set of states corresponds to each state of the DFA. Omit inaccessible states.

It is easy to make mistakes, so show all your working in calculating the DFA.

5. Understanding the Myhill-Nerode Theorem (20 points total)

Consider the regular language L over the alphabet $\Sigma = \{a, b\}$ represented by the following regular expression

$a^*b^*|b^*a^*$

(a) **(8 points)** Draw the minimal (complete) DFA for L. Use the marking algorithm given in lectures to check your diagram. Show your working when applying the marking algorithm.

(b) **(6 points)** The Myhill-Nerode relation $\equiv_L$ is defined over Σ^* as follows:

$x \equiv_L y \overset{\text{def}}{\iff} \forall z \in \Sigma^*(xz \in L \iff yz \in L)$.

In other words, $x \equiv_L y$ iff x and y are indistinguishable with respect to L.

For each of the following pairs of strings x and y, determine if $x \equiv_L y$, where $L = L(a^*b^*|b^*a^*)$. Give justification for your answers.

(i) $x = aa$ and $y = aaa$

(ii) $x = aab$ and $y = ababa$

(iii) $x = \lambda$ and $y = ab$

(c) **(6 points)** Give a regular expression to describe each of the equivalence classes of $\equiv_L$. Hint: there is one equivalence class corresponding to each of the states in the minimal DFA for L.

6. Using the Myhill-Nerode Theorem (10 points)

Let L be the language over $\{0, 1\}^*$ defined as follows:

$L = \{0^n 1 0^{2n} \mid n \geq 0\}$

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