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

Test (14 September, 2000)

Time allowed = 90 minutes
Answer ALL questions

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) $1^*(01)^*0^*$
min length string 011

(ii) $(0^*|1^*)(0^*|1^*)(0^*|1^*)$
min length string 0101 or 1010

(iii) $0^*(100^*)^*1^*$
min length string 110

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

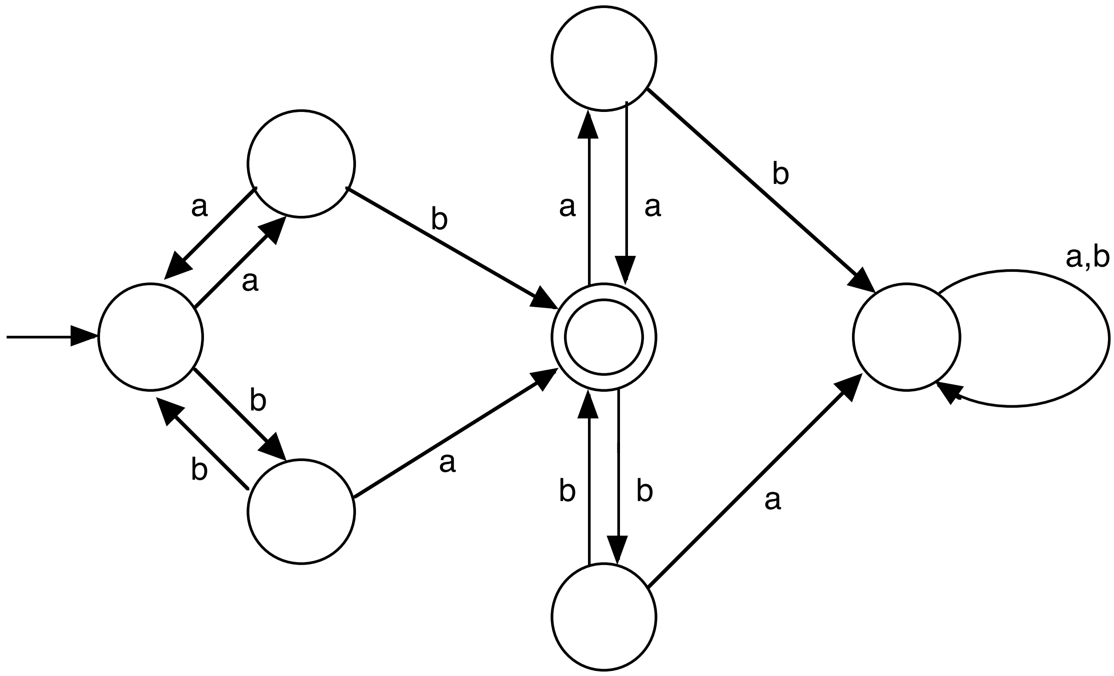
(i) The language of all strings that begin or end with 00 or 11.
$(00|11)(0|1)^* \mid (0|1)^*(00|11)$

(ii) The language of all strings that do not end with 01.
$\lambda \mid 1 \mid (0|1)^*(0|11)$

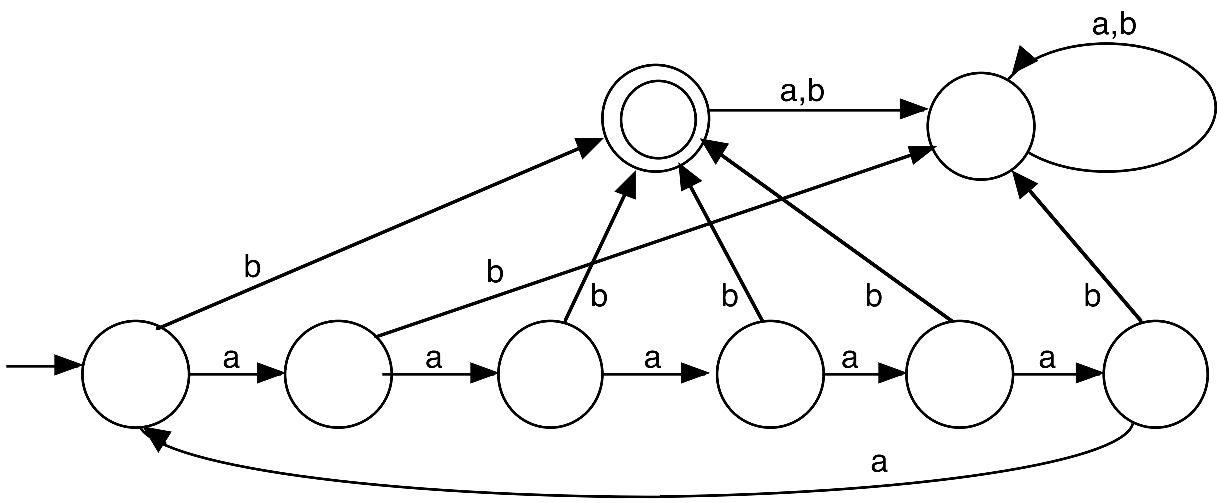
(iii) The language of all strings containing exactly one occurrence of the string 00. (The string 000 should be viewed as containing two occurrences of 00.)
$(01|1)^*00(10|1)^*$

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 $(aa|bb)^*(ab|ba)(aa|bb)^*$.

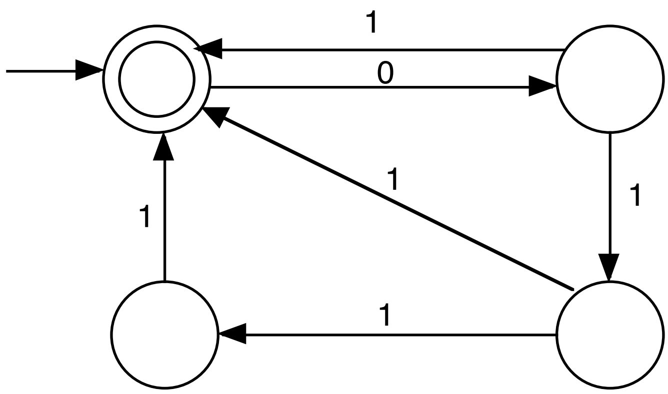


(b) **(10 points)** Find a DFA that recognizes the language $L(\alpha)$ over the alphabet $\Sigma = \{a, b\}$ where α is the the regular expression $(aaa)^*b|(aa)^*b$.

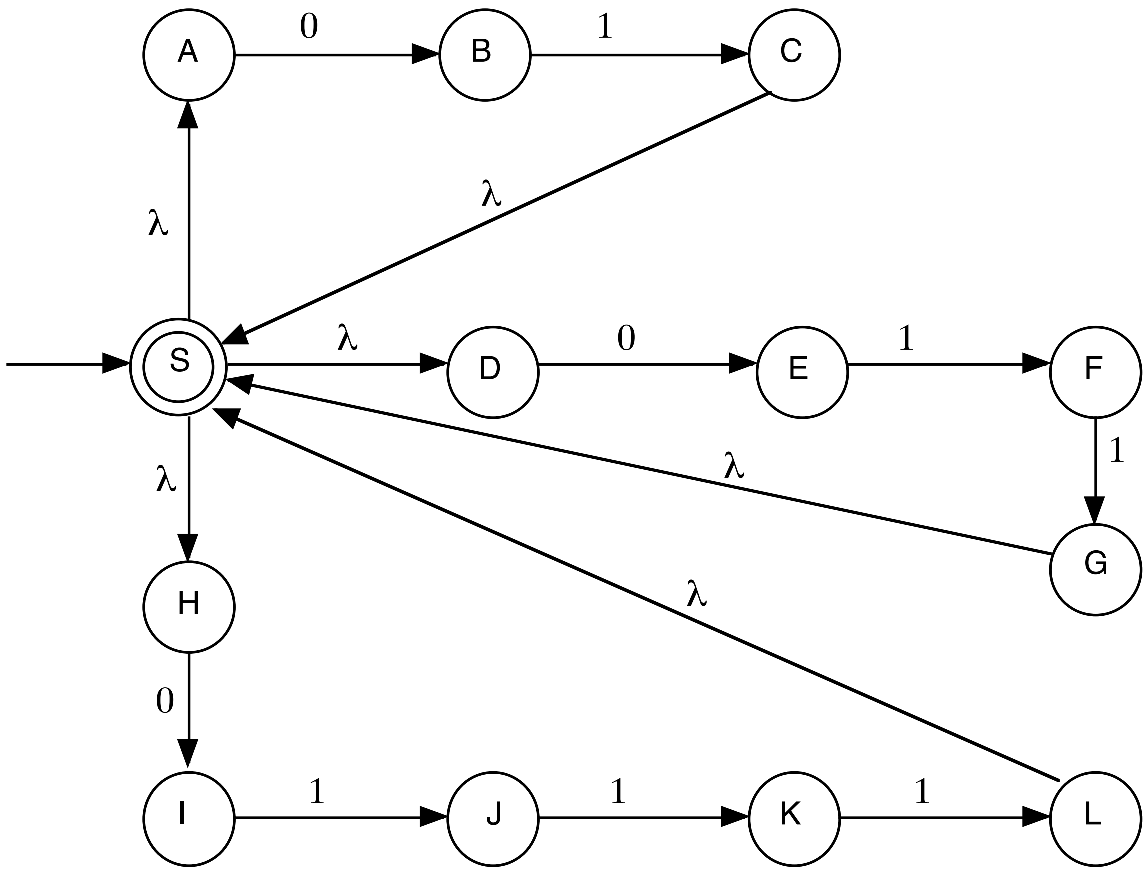


3. Designing an NFA (5 points). Find an NFA, with only four states, that accepts the regular language over the alphabet $\Sigma = \{0, 1\}$ described by the following regular expression.

(01|011|0111)*

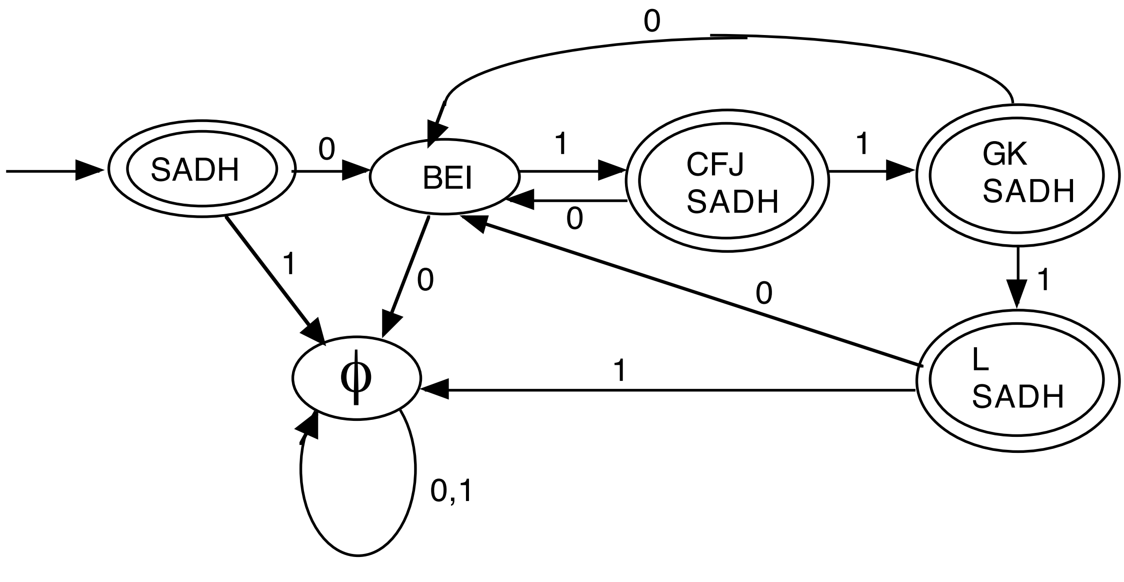


4. Converting an NFA-λ to a DFA (10 points). Here is an NFA-λ automaton that also accepts $L((01|011|0111)^*)$.



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.

$\Lambda\{S\} = \{S, A, D, H\}$
$\Delta(\{S, A, D, H\}, 0) = \{B, E, I\}$
$\Delta(\{S, A, D, H\}, 1) = \emptyset$
$\Lambda\{B, E, I\} = \{B, E, I\}$
$\Delta(\{B, E, I\}, 0) = \emptyset$
$\Delta(\{B, E, I\}, 1) = \{C, F, J\}$
$\Lambda\{C, F, J\} = \{C, F, J, S, A, D, H\}$
$\Delta(\{C, F, J, S, A, D, H\}, 0) = \{B, E, I\}$
$\Delta(\{C, F, J, S, A, D, H\}, 1) = \{G, K\}$
$\Lambda\{G, K\} = \{G, K, S, A, D, H\}$
$\Delta(\{G, K, S, A, D, H\}, 0) = \{B, E, I\}$
$\Delta(\{G, K, S, A, D, H\}, 1) = \{L\}$
$\Lambda\{L\} = \{L, S, A, D, H\}$
$\Delta(\{L, S, A, D, H\}, 0) = \{B, E, I\}$
$\Delta(\{L, S, A, D, H\}, 1) = \emptyset$
$\Delta(\emptyset, 0) = \emptyset$
$\Delta(\emptyset, 1) = \emptyset$

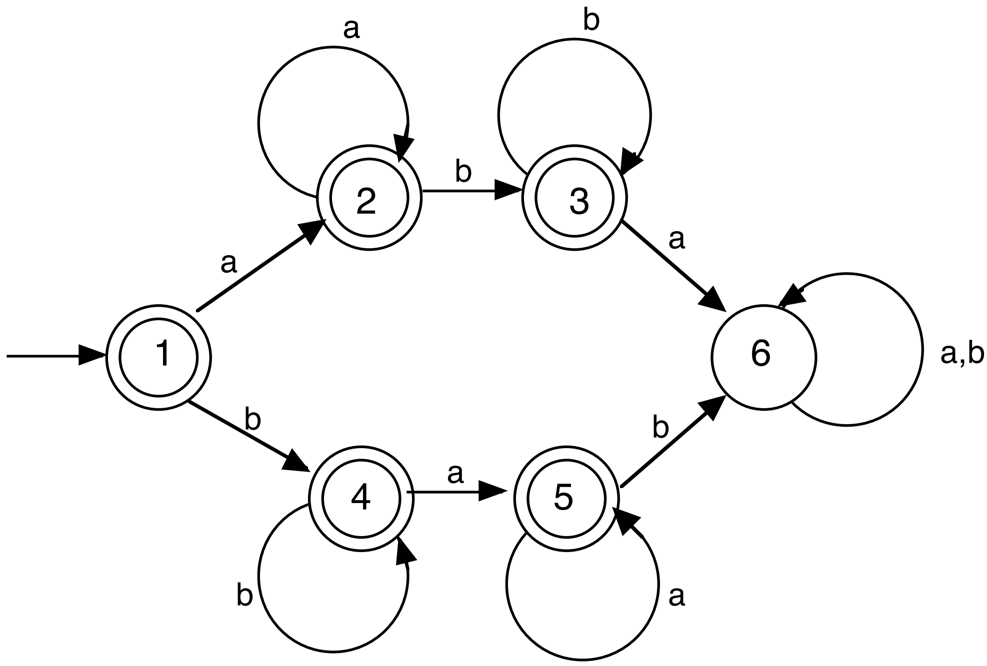


5. Understanding the Myhill-Nerode Theorem (20 points total)

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.



Marking algorithm:

1st round: mark pairs where one of the pair is final, the other non-final.

Mark $(1, 6), (2, 6), (3, 6), (4, 6), (5, 6)$.

2nd round: mark pairs where an already-marked pair can be reached from the current pair under some input symbol.

Mark $(1, 3), (1, 5), (2, 3), (2, 5), (3, 4), (3, 5), (4, 5)$.

3rd round: mark pairs where an already-marked pair can be reached from the current pair under some input symbol.

Mark $(1, 2), (1, 4), (2, 4)$.

All pairs are marked, so no states can be collapsed together. The DFA given is the unique (up to isomorphism) DFA for L.

(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 $y = aaa$
x and y are indistinguishable with respect to L: choose $z \in a^*b^*$, then $xz \in L$ and $yz \in L$, choose $z \in \Sigma^*$ such that $z \notin a^*b^*$, then $xz \notin L$ and $yz \notin L$.

(ii) $x = aab$ and $y = ababa$ x and y are distinguishable with respect to L: choose $z = \lambda$, then $xz \in L$, $yz \notin L$.

(iii) $x = \lambda$ and $y = ab$
x and y are distinguishable with respect to L: choose $z = a$, then $xz \in L$, $yz \notin L$.

Note that an argument that covers all $z \in \Sigma^*$ is required for (i), while the existence of a single suitable z is all that is required for (ii) and (iii).

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

$[\lambda]$
$[aa^*]$
$[bb^*]$
$[aa^*bb^*]$
$[bb^*aa^*]$
$[(aa^*bb^*a|bb^*aa^*b)(a|b)^*]$

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.

Choose an infinite set of strings from Σ^*, say $\{0^n \mid n > 0\}$.

Now show that these strings are all pairwise distinguishable with respect to L, and, thus, must be in different equivalence classes under $\equiv_L$. Therefore, $\equiv_L$ does not have finite index, so L cannot be regular.

To show that strings in $\{0^n \mid n > 0\}$ are all pairwise distinguishable with respect to L:

Choose two distinct strings 0^i and 0^j, $i \neq j$. Choose $z = 10^{2i}$, $z \in \Sigma^*$. Now, $xz \in L$ but $yz \notin L$.