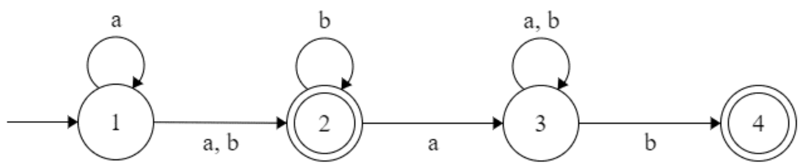
CMPS 130
Computational Models
Spring 2016
Midterm Exam 2 **Solutions**

1. (20 Points) Let M be the NFA on $\{a, b\}^*$ pictured below.



a. (5 Points) Write a regular expression corresponding to $L(M)$.

Solution: $a^*(a+b)b^*(\lambda + a(a+b)^*b)$

b. (15 Points) Using the subset construction, write the transition table and draw the transition diagram for a DFA accepting $L(M)$. Label the states in your DFA so as to make it clear how it was obtained from the subset construction.

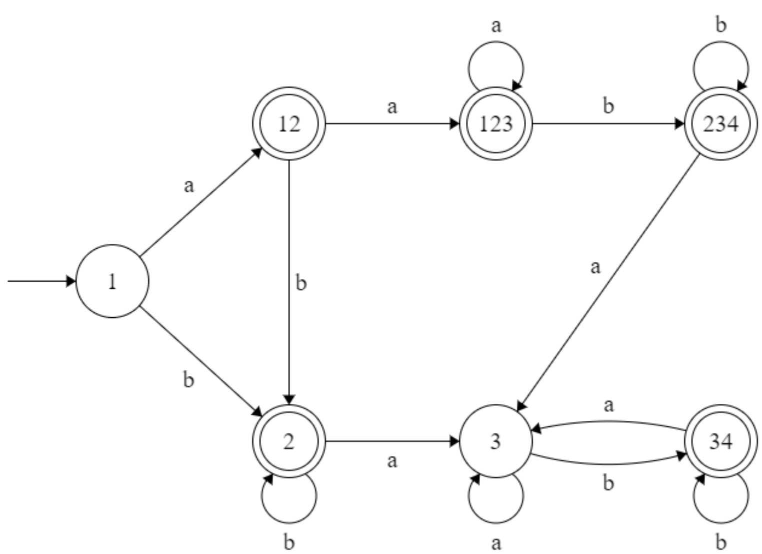
Solution: Initial states are in the first row, and accepting states are indicated by an asterisk *.

NFA

	a	**b**
1	{1, 2}	{2}
***2**	{3}	{2}
3	{3}	{3, 4}
***4**	∅	∅

DFA

	a	**b**
1	12	2
***2**	3	2
3	3	34
***12**	123	2
***34**	3	34
***123**	123	234
234	3	234



2. (20 Points) Let $M = (Q, \Sigma, q_0, A, \delta)$ be an NFA and let $S, T \subseteq Q$. .
   a. (10 Points) State the recursive definition of the λ-closure $\lambda(S \cup T)$.

      Definition:
      (1) $S \cup T \subseteq \lambda(S \cup T)$.
      (2) $q \in \lambda(S \cup T) \Rightarrow \delta(q, \lambda) \subseteq \lambda(S \cup T)$.

   b. (10 Points) Prove that $\lambda(S \cup T) \subseteq \lambda(S) \cup \lambda(T)$ using structural induction.

      Proof:
      We must show that every $q \in \lambda(S \cup T)$ has the property $q \in \lambda(S) \cup \lambda(T)$.

      I. Let $q \in S \cup T$. We have two cases:
         <u>case 1</u>: $q \in S$. Then $q \in \lambda(S)$ by the base definition of $\lambda(S)$.
         <u>case 2</u>: $q \in T$. Then $q \in \lambda(T)$ by the base definition of $\lambda(T)$.
         Therefore either $q \in \lambda(S)$ or $q \in \lambda(T)$, hence $q \in \lambda(S) \cup \lambda(T)$.

      II. Let $q \in \lambda(S \cup T)$ and assume $q \in \lambda(S) \cup \lambda(T)$. We must show that $\delta(q, \lambda) \subseteq \lambda(S) \cup \lambda(T)$. The induction hypothesis says either $q \in \lambda(S)$ or $q \in \lambda(T)$, so again we have two cases:
         <u>case 1</u>: $q \in \lambda(S)$. Then $\delta(q, \lambda) \subseteq \lambda(S)$ by the recursive definition of $\lambda(S)$.
         <u>case 2</u>: $q \in \lambda(T)$. Then $\delta(q, \lambda) \subseteq \lambda(T)$ by the recursive definition of $\lambda(T)$.
         Therefore either $\delta(q, \lambda) \subseteq \lambda(S)$ or $\delta(q, \lambda) \subseteq \lambda(T)$, and hence $\delta(q, \lambda) \subseteq \lambda(S) \cup \lambda(T)$.

3. (20 Points) Let Pal $= \{ x \in \{a, b\}^* \mid x = x^r \}$ be the set of Palindromes over $\{a, b\}$.
   a. (5 Points) State the pumping lemma.
      <u>Pumping Lemma</u>
      If $M = (Q, \Sigma, q_0, A, \delta)$ is a DFA with $|Q| = n$, the for every $x \in L(M)$ with $|x| \geq n$ there exists $u, v, w \in \Sigma^*$ such that $x = uvw$ and
      (1) $|uv| \leq n$
      (2) $|v| \geq 1$
      (3) $uv^iw \in L(M)$ for every $i \geq 0$.

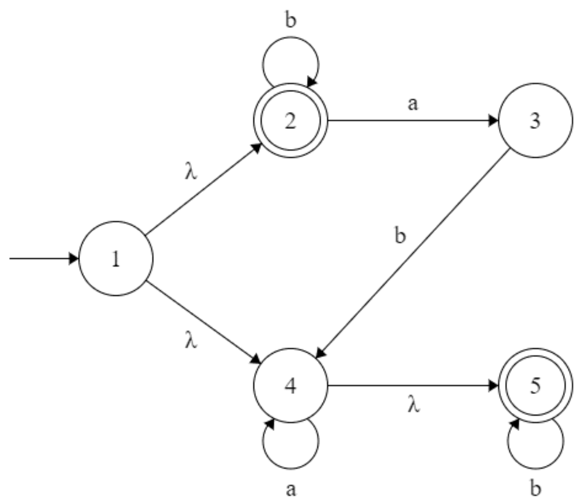
   b. (5 Points) State Kleene's Theorem.
      <u>Kleene's Theorem</u>
      A language is regular if and only if it is accepted by some DFA.

   c. (10 Points) Use (a) and (b) to prove that Pal is *not* a regular language.
      Proof:
      First we show that Pal is not accepted by any DFA. Assume, to get a contradiction, that Pal is accepted by a DFA with n states. Let $x = a^n b a^n$. Then clearly $x \in Pal$ and $|x| \geq n$. The Pumping Lemma gives us $u, v, w \in \{a, b\}^*$ such that $x = uvw$ and (1), (2) and (3) above hold.

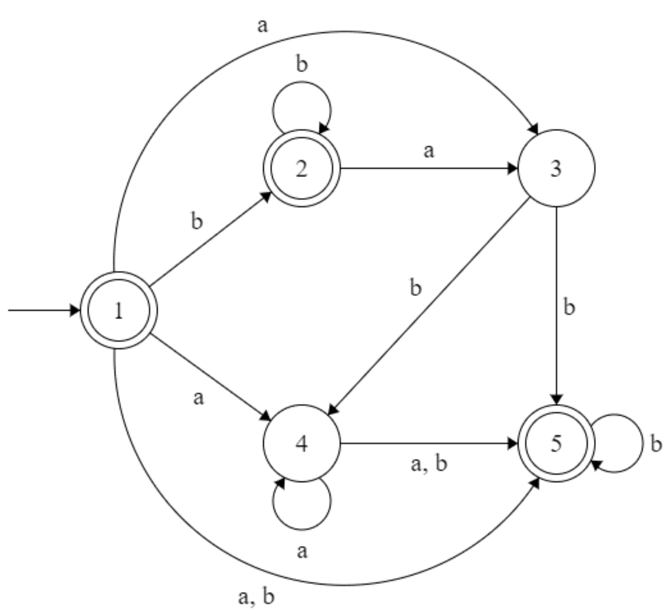
      By (1), u and v contain only a's. By (2) $v = a^k$ for some $k \geq 1$. By (3) $uv^2w \in$ Pal. But $uv^2w = a^{n+k}ba^n \notin$ Pal since $n \neq n + k$. This contradiction shows that our assumption was false, and therfore Pal is not accepted by any DFA.

      It now follows from Kleene's Theorem that Pal cannot be regular.

4. (20 Points) Consider the following NFA pictured below. Use the algorithm described in class to draw another NFA with no λ-transitions that accepts the same language.



Solution:
Since λ is accepted by the above NFA, $A_1 = A \cup \{1\} = \{1, 2, 5\}$.



5. (20 Points) Let Σ be a finite alphabet, $L \subseteq \Sigma^*$ and $x, y \in \Sigma^*$.

a. (5 Points) Write the definition of L-indistinguishability relation: xI_Ly.

Definition: xI_Ly if and only if $\forall z$: $xz \in L \leftrightarrow yz \in L$.

b. (5 Points) Write the definition of the set: L/x.

Definition: $L/x = \{ z \in \Sigma^* \mid xz \in L \}$.

c. (5 Points) Prove that xI_Ly if and only if $L/x = L/y$.

Proof:

xI_Ly	iff	$\forall z$: $xz \in L \leftrightarrow yz \in L$	(definition of I_L)
	iff	$\forall z$: $z \in L/x \leftrightarrow z \in L/y$	(definition of L/x)
	iff	$L/x = L/y$	(definition of set equality)

d. (5 Points) Prove that I_L is an equivalence relation on Σ^*.

Proof:
This follows directly from part (c) and the fact that $=$ is an equivalence relation.

<u>Reflexive</u>: xI_Lx, since $L/x = L/x$.

<u>Symmetric</u>: $xI_Ly \Rightarrow yI_Lx$, since $(L/x = L/y) \Rightarrow (L/y = L/x)$.

<u>Transitive</u>: $xI_Ly \wedge yI_Lz \Rightarrow xI_Lz$, since $(L/x = L/y) \wedge (L/y = L/z) \Rightarrow (L/x = L/z)$.