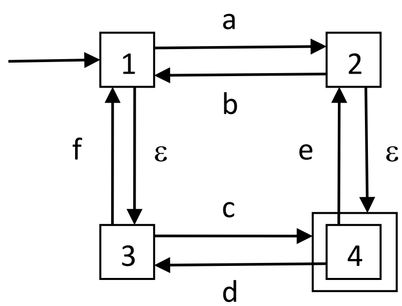
CS 475/575 Exam 1 Fall 2012 Solution

This exam has 115 possible points (includes 15 points extra credit).

1. Write all strings with length exactly 3 that are accepted by this finite-state machine: **[10 points]**



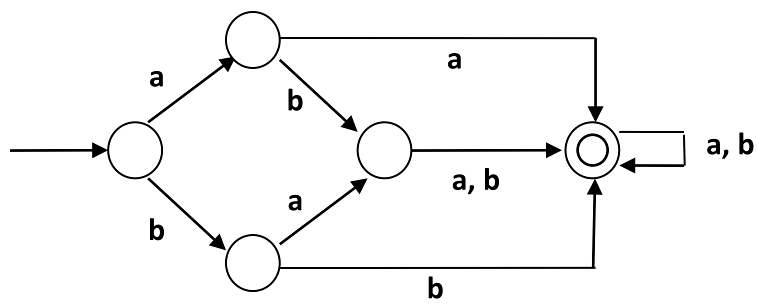
aba	**abc**	**adc**	**aee**	**cdc**
cee	**fae**	**fce**	**ffa**	**ffc**

2. For each pair of languages X and Y below, write a string that belongs to language X but that does not belong to language Y. If no such string exists, write "None". **[10 points]**

Language X	Language Y	String in X but not in Y
$(a \cup b \cup c)^*$	$a^* \cup b^* \cup c^*$	**ab**
$(abc)^*$	$a^* b^* c^*$	**abcabc**
$a^* b^* c^*$	$(abc)^*$	**aa**
$(b^* a)^* b^*$	$a^* (b a^*)^*$	**None**
$(b^+ a^+)^+ b^+$	$a^+ (b^+ a^+)^+$	**bab**
$(a^* \cup b^*) (c^* \cup d^*)$	$a^* c^* \cup b^* d^*$	**ad**
$(aa \cup ab \cup ba \cup bb)^*$	$a^* b^* \cup b^* a^*$	**abab**
$a^* b^* \cup b^* a^*$	$(aa \cup ab \cup ba \cup bb)^*$	**a**
$(a^* b)^*$	$(a \cup b)^* b$	**ε**
$(a^+ \cup b^+ \cup c^+)^*$	$(a^+ b^+ c^+)^*$	**a**

3. Let language L_1 be the set of strings over alphabet {a, b} that contain at least two a's or at least two b's.

   a. Draw a deterministic finite-state machine that accepts language L_1. **[10 points]** (Half credit for a non-deterministic machine.)

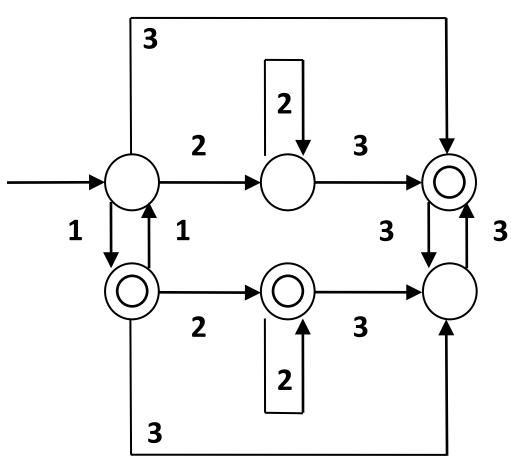


   b. Write a regular expression that generates language L_1. **[10 points]**

   (a∪b)* (ab*a ∪ ba*b) (a∪b)*

4. Let language L_2 be the set of strings over alphabet {1, 2, 3} such that the digits in the string are arranged in ascending order and the sum of all digits in the string is odd. For example, language L_2 includes strings 133 and 112223, but it does not include 123 or 3211.

   a. Draw a deterministic finite-state machine that accepts language L_2. **[10 points]** (Half credit for a non-deterministic machine.)

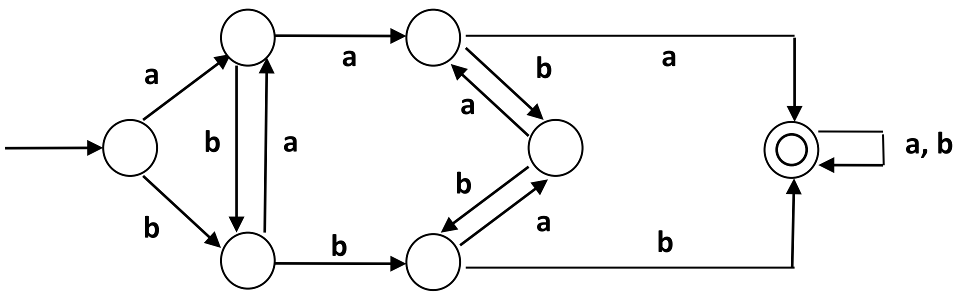


   b. Write a regular expression that generates language L_2. **[10 points]**

   1 (11)* 2* (33)* ∪ (11)* 2* 3 (33)*

5. Let language L_3 be the set of strings over alphabet {a, b} that contain some substring in which | (number of a's) – (number of b's) | = 3. For example, language L_3 includes string abbabba because it contains substring bbabb which has (number of a's) = 1 and (number of b's) = 4, and |1 – 4| = 3.

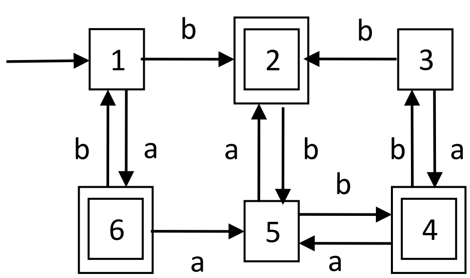
   a. Draw a deterministic finite-state machine that accepts language L_3. **[10 points]** (Half credit for a non-deterministic machine.)



   b. Write a regular expression that generates language L_3. **[10 points]**

   (a∪b)* (aa (ba)* a ∪ bb (ab)* b) (a∪b)*

6. Given the deterministic finite-state machine shown below, complete the table of distinguishabilities, and specify which states are equivalent. **[10 points]**



2	**X**	–	–	–	–
3	**O**	**X**	–	–	–
4	**X**	**X**	**X**	–	–
5	**X**	**X**	**X**	**X**	–
6	**X**	**X**	**X**	**O**	**X**
	1	2	3	4	5

Equivalent states are {1, 3} and {4, 6}.

7. Let $L_4 = \{ a^m b^n \mid m > n \text{ or } n \geq 2m \}$ and use the pumping theorem to show that L_4 is not regular. **[15 points]**

First complete this statement of the pumping theorem for regular languages:
For every regular language L, there exists some constant p such that
for every string s with $\underline{\mathbf{s \in L}}$ and $\underline{\mathbf{|s| \geq p}}$.
it is possible to write s = uvw such that $\underline{\mathbf{|uv| \leq p}}$, $\underline{\mathbf{|v| \geq 1}}$, and
for every integer $i \geq 0$, $\underline{\mathbf{u v^i w \in L}}$.

Next apply the pumping theorem to show that $L_4 = \{ a^m b^n \mid n < m \text{ or } n \geq 2m \}$ is not regular:
Choose string s = $\underline{\mathbf{a^p b^{2p}}}$.
Write u = $\underline{\mathbf{a^j}}$, v = $\underline{\mathbf{a^k}}$, and w = $\underline{\mathbf{a^{p-j-k} b^{2p}}}$.
Choose value i = $\underline{\mathbf{2}}$.

Finally show the contradiction:

$\mathbf{u v^2 w = a^j a^{2k} a^{p-j-k} b^{2p} = a^{p+k} b^{2p}}$.
But $\mathbf{|uv| \leq p \Rightarrow k \leq p \Rightarrow}$ **NOT** $\mathbf{p+k > 2p}$.
And $\mathbf{|v| \geq 1 \Rightarrow k \geq 1 \Rightarrow}$ **NOT** $\mathbf{2p \geq 2(p+k)}$.
Therefore $\mathbf{u v^2 w = a^{p+k} b^{2p} \notin L_4}$.

8. The following "theorem" is obviously false, so its "proof" by mathematical induction must contain an error. Identify the precise location in the "proof" that is incorrect, and explain why it is incorrect. **[10 points]**

"Theorem": Let Σ denote any ordered alphabet (such as ASCII). Also let $S = a_1 a_2 \ldots a_{n-1} a_n$ denote any string with length $n \geq 1$ over alphabet Σ. Then the characters of S are always arranged in ascending sorted order, that is, $a_1 \leq a_2 \leq \ldots \leq a_{n-1} \leq a_n$.

"Proof": By mathematical induction on the length n.
Basis: When n = 1, the string $S = a_1$ is arranged in ascending order.
Inductive hypothesis: Assume for some n = k that every string $a_1 a_2 \ldots a_{k-1} a_k$ appears in ascending order $a_1 \leq a_2 \leq \ldots \leq a_{k-1} \leq a_k$.
Inductive step: Consider n = k+1, and choose arbitrary string $S = a_1 a_2 \ldots a_k a_{k+1}$. This string S has prefix $S' = a_1 a_2 \ldots a_{k-1} a_k$ and suffix $S'' = a_2 a_3 \ldots a_k a_{k+1}$. Both S' and S'' have length k, so by the inductive hypothesis both S' and S'' are arranged in ascending order. Therefore $a_1 \leq a_2 \leq \ldots \leq a_{k-1} \leq a_k$ and $a_2 \leq a_3 \leq \ldots \leq a_k \leq a_{k+1}$. Putting these inequalities together, we obtain the desired result that $a_1 \leq a_2 \leq \ldots \leq a_k \leq a_{k+1}$, so the string S of length n = k+1 is arranged in ascending order. Since string S was chosen arbitrarily, this result holds for every string S of length n = k+1.

The error in the "proof" occurs during the inductive step only when k = 1 and n = k+1 = 2. In this case $S = a_1 a_2$, prefix $S' = a_1$, and suffix $S'' = a_2$. Here it does not follow that $a_1 \leq a_2$.

(Note: for all k ≥ 2, the "proof" is mechanically correct. For example, when k = 2 and n = 3, $S = a_1 a_2 a_3$, prefix $S' = a_1 a_2$, and suffix $S'' = a_2 a_3$. If $a_1 \leq a_2$ and $a_2 \leq a_3$ then $a_1 \leq a_2 \leq a_3$.)