Name:

1	10	
2 a	5	
b	10	
3	10	
4 a	15	
b	15	
c	15	
d	15	
5 a	5	
b	15	
c	15	
6	15	
7	15	
8 a	5	
b	5	
c	5	
Total	175	

CS 341 First Midterm Exam Practice

1. Use extra paper to determine your solutions then neatly transcribe them (including intermediate steps) onto these sheets.
2. It's possible that you won't be able to finish. Read through the whole exam once and start working on the problems you're sure you know how to do. Come back to the harder ones as you have time.

(1) Consider the following problem: Given a database *D* and a query *Q*, what result is returned when *Q* is executed against *D*?

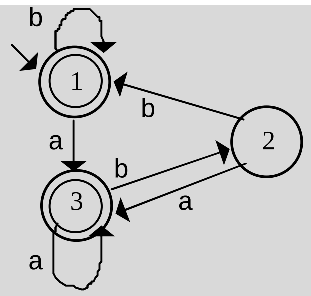
L = {<*D*, *Q*, *R*> : *R* is the result of executing *Q* against *D*}.

(2) Let $L = \{w \in \{a, b\}^* : w$ does not end in ab$\}$

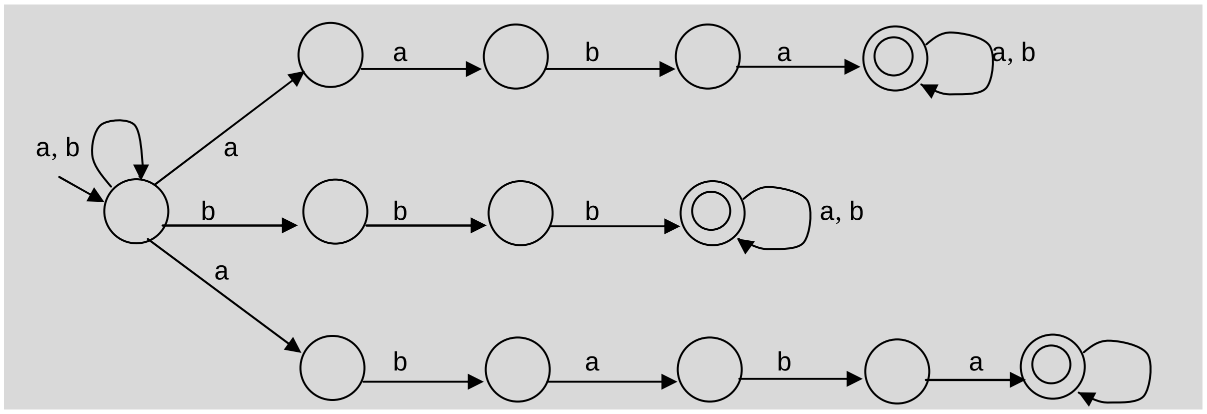
(a) Show a regular expression that generates *L*.

$\varepsilon \cup b \cup (a \cup b)^* (a \cup bb)$

(b) Show an FSM that accepts *L*.



(3) Show a (possibly nondeterministic) FSM that accepts $\{w \in \{\mathtt{a}, \mathtt{b}\}^* : w$ contains at least one instance of `aaba`, `bbb` or `ababa`$\}$.



(4) For each of the following languages L, state whether it is regular or not and prove your answer.

(a) $\{x\#y$: $x, y \in \{\texttt{0}, \texttt{1}\}^*$, when viewed as binary numbers, $x+y = 3y\}$. Example: `1000#100` $\in L$.

Not regular, which we show using the Pumping Theorem. We must start by choosing a string that is in fact in L. Let $w = \texttt{100}^k\texttt{\#10}^k$. Then $w \in L$ since x ($\texttt{100}^k$) is equal to $2y$ (where y is $\texttt{10}^k$). We must consider three cases for where y can fall:

$y = \texttt{1}$ Pump out. Arithmetic is wrong. The left side is 0 but right side isn't.
$y = \texttt{10*}$ Pump out. Arithmetic is wrong. "
$y = \texttt{0}^p$ Pump out. Arithmetic is wrong. Decreased left side but not right. So, in particular, it is no longer the case that $x > y$ (required since $y \neq 0$).

(b) Let $\Sigma = \{\texttt{a}, \texttt{b}\}$. $L = \{w \in \Sigma^*$: (w contains the substring `ab`) $\rightarrow$ (w contains the substring `ba`)$\}$

Regular. It helps to rewrite L as:

$L = \{\text{w} \in \Sigma^*$: $\neg$(w contains the substring `ab`) $\vee$ (w contains the substring `ba`)$\}$

$L =$ $\texttt{b*a*}$ $\cup$ $(\texttt{a} \cup \texttt{b})^*\ \texttt{ba}\ (\texttt{a} \cup \texttt{b})^*$

You can also do this with an FSM.

(c) $\{w = xyzy : x, y, z \in \{\texttt{0}, \texttt{1}\}^+\}$.

Regular. The key to why this is so is to observe that we can let y be just a single character. Then the rest of w can generated by x and z. So any string w in $\{\texttt{0}, \texttt{1}\}^+$ is in L iff:

- the last letter of w occurs in at least one other place in the string,
- that place is not the next to the last character,
- nor is it the first character, and
- w contains least 4 letters.

Either the last character is `0` or `1`. So:

$L = ((\texttt{0} \cup \texttt{1})^+\ \texttt{0}\ (\texttt{0} \cup \texttt{1})^+\ \texttt{0}) \cup ((\texttt{0} \cup \texttt{1})^+\ \texttt{1}\ (\texttt{0} \cup \texttt{1})^+\ \texttt{1})$.

(d) $\{w = st : s \in \{\texttt{a}, \texttt{b}\}^*$ and $t \in \{\texttt{b}, \texttt{c}\}^*$ and $\#_{\texttt{b}}(s) = 2\cdot\#_{\texttt{a}}(s)$ and $\#_{\texttt{c}}(t) = 3\cdot\#_{\texttt{b}}(t)\}$.

Not regular, which we show by pumping. Let $w = \texttt{b}^{2k}\texttt{a}^k\texttt{c}^{3k}\texttt{b}^k$. y must occur in the first `b` region. It is $\texttt{b}^p$, for some nonzero p. Note that, when we pump, the boundary between the s and t regions cannot move because there can be no `a`'s in s or `c`'s in t. Let $q = 0$ (i.e., pump out). The resulting string is $\texttt{b}^{2k-p}\texttt{a}^k\texttt{c}^{3k}\texttt{b}^k$. The s region is $\texttt{b}^{2k-p}\texttt{a}^k$. It doesn't have twice as many `b`'s as `a`'s. So this string is not in L.

(5) Recall that *maxstring*(L) = $\{w: w \in L \text{ and } \forall z \in \Sigma^* (z \neq \varepsilon \rightarrow wz \notin L)\}$.

(a) What is *maxstring*(L_1L_2), where $L_1 = \{w \in$ {a, b}* : contains exactly one a} and L_2 = {a}?

L_1L_2

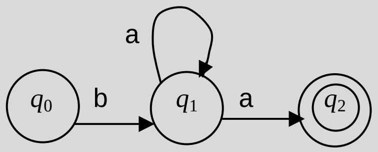
(b) Prove that the regular languages are closed under *maxstring*.

The proof is by construction. If L is regular, then it is accepted by some DFSA $M = (K, \Sigma, \Delta, s, A)$. We construct a new DFSM $M^* = (K^*, \Sigma^*, \Delta^*, s^*, A^*)$, such that $L(M^*)$ = *maxstring*(L). The idea is that M^* will operate exactly as M would have except that A^* will include only states that are accepting states in M and from which there exists no path of at least one character to any accepting state (back to itself or to any other). So an algorithm to construct M^* is:

```
1.  Initially, let $M^* = M$.
/* Check each accepting state in $M$ to see whether there are paths from it to some accepting state.
2.  For each state $q$ in $A$ do:
    2.1. Follow all paths out of $q$ for $|K|$ steps or until the path reaches an element of $A$ or some state
         it has already visited.
    2.2. If the path reached an element of $A$, then $q$ is not an element of $A^*$.
    2.3. If the path ended without reaching an element of $A$, then $q$ is an element of $A^*$.
```

Comments on this algorithm:

1. Why do we need to start with a deterministic machine? Suppose L is ba*a. *maxstring*(L) = {}. But suppose that M were:



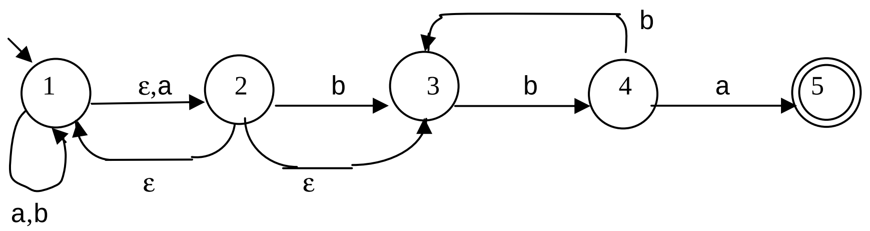
If we executed our algorithm with this machine, we would accept ba*a rather than {}.

2. Your initial thought may be that the accepting states of M^* should be simply the accepting states of M that have no transitions out of them. But there could be transitions that themselves lead no where, in which case they don't affect the computation of *maxstring*. So we must start by finding exactly those accepting states of M such that there is no continuation (other than ε) that leads again to an accepting state.

(c) If *maxstring*(L) is regular, must L also be regular? Prove your answer.

No. Consider $\text{Prime}_a = \{a^n : n \text{ is prime}\}$. Prime_a is not regular. But *maxstring*(Prime_a) = $\varnothing$, which is regular.

(6) Consider the following NDFSM *M*. Use *ndfsmtodfsm* to construct an equivalent DFSM. Begin by showing the value of *eps*(*q*) for each state *q*:



eps(1) = {1, 2, 3}
eps(2) = {2, 3, 1}
eps(3) = {3}
eps(4) = {4}
eps(5) = {5}

{1, 2, 3}, a, {1, 2, 3}
, b, {1, 2, 3, 4}
{1, 2, 3, 4}, a, {1, 2, 3, 5}
, b, {1, 2, 3, 4}
{1, 2, 3, 5}, a, {1, 2, 3}
, b, {1, 2, 3, 4}

There is one accepting state, {1, 2, 3, 5}.

(7) Define a decision procedure to answer the following question. You may use as subroutines all the procedures that we have discussed in class. Let $\Sigma = \{a, b\}$ and let α and β be regular expressions. Is the following sentence true:

$$(L(\beta) = a^*) \vee (\forall w\ (w \in \{a, b\}^* \wedge |w| \text{ even}) \rightarrow w \in L(\alpha))$$

1. From β, build FSM M_1. Make it deterministic. Minimize it, producing M_2.
2. Build M_a, the simple one-state machine that accepts a*.
3. If M_2 and M_a are identical except for state names then return true. Else continue.

/* Observe that the second condition says that L_E, the language of even length strings of a's and b's, is a subset of $L(\alpha)$. This is equivalent to saying that $L_E - L(\alpha)$ is empty. So:
4. From α, build FSM M_3.
5. Build M_E that accepts exactly L_E, the language of even length strings of a's and b's.
6. Build M_D that accepts $L(M_E) - L(M_3)$.
7. See if $L(M_D)$ is empty. If it is, return *True*. Else return *False*.

(8) Prove or disprove each of the following statements:

(a) It is possible that the intersection of an infinite number of regular languages is not regular.

True. Let $x_1, x_2, x_3, \ldots$ be the sequence 0, 1, 4, 6, 8, 9, … of nonprime, nonnegative integers. Let a^{x_i} be a string of x_i a's. Let L_i be the language $a^* - \{a^{x_i}\}$.

Now consider L = the infinite intersection of the sequence of languages $L_1, L_2, \ldots$ Note that $L = \{a^p$, where p is prime$\}$. We have proved that L is not regular.

(b) Every subset of a regular language is regular.

False.
Let $L = \mathtt{a}^*$, which is regular.
Let $L' = \mathtt{a}^p$, where p is prime. L' is not regular, but it is a subset of L.

(c) Let $L_4 = L_1L_2L_3$. If L_1 and L_2 are regular and L_3 is not regular, it is possible that L_4 is regular.

True. Example:

Let $L_1 = \{\varepsilon\}$.	L_1 is regular.
Let $L_2 = \mathtt{a}^*$.	L_2 is regular.
Let $L_3 = \mathtt{a}^p$, where p is prime.	L_3 is not regular.
$L_4 = \mathtt{a}^k$, where $k \geq 2$	L_4 is regular, because it is defined by $\mathtt{aaa}^*$.