Formal languages, grammars, and automata

Assignment 2, Wednesday, Nov. 19, 2014

Exercises with answers

1. Let $M = (Q, q_0, \delta, F)$ be the deterministic finite automaton (DFA) over $\Sigma = \{a, b, c\}$, where $Q = \{q_0, q_A, q_B, q_C\}$, $F = \{q_0\}$ and δ is given by the table

δ	a	b	c
q_0	q_A	q_B	q_C
q_A	q_0	q_C	q_B
q_B	q_C	q_0	q_A
q_C	q_B	q_A	q_0

(a) Make a state transition diagram for M.

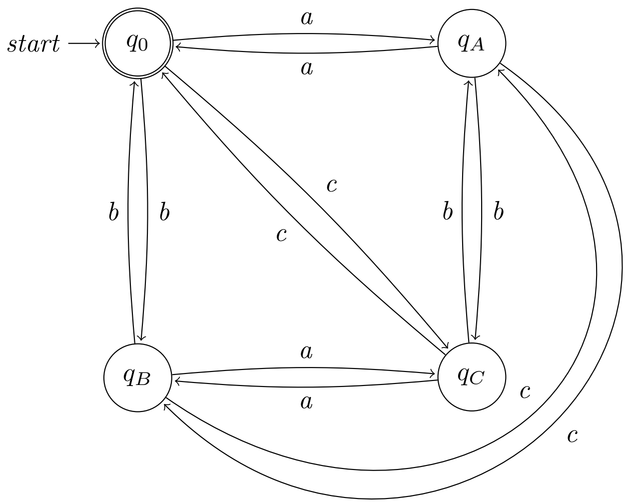
(b) Determine for the following words whether they belong to $\mathcal{L}(M)$: $abba, baab, bac, cac$.

(c) Is it the case that $\{a^n b^n | n \geq 0\} \subseteq \mathcal{L}(M)$? Give a proof or a counterexample.

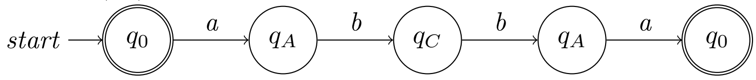
(d) Is it the case that $\{a^n c^n b^n | n \geq 0\} \subseteq \mathcal{L}(M)$? Give a proof or a counterexample.

Answer Info: ....

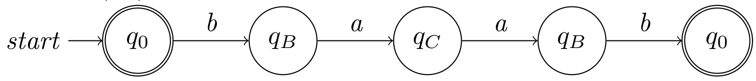
(a) The state transition diagram can be drawn as



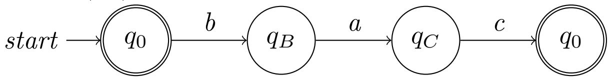
(b) $abba \in \mathcal{L}(M)$, *because the automaton ends in state* q_0*:*



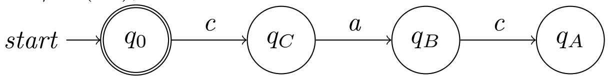
$baab \in \mathcal{L}(M)$, *because the automaton ends in state* q_0*:*



$bac \in \mathcal{L}(M)$, because the automaton ends in state q_0:



$cac \notin \mathcal{L}(M)$, because the automaton doesn't end in a final state:



(c) *It is not the case that $\{a^n b^n | n \geq 0\} \subseteq \mathcal{L}(M)$, because for $n = 1$ we have $a^1 b^1 = ab \notin \mathcal{L}(M)$. This is the case, because the automaton ends in state q_C, which is not accepting.*

(d) *It is the case that $\{a^n c^n b^n | n \geq 0\} \subseteq \mathcal{L}(M)$. We prove this by case distinction.*

First consider the case that n is even. Then, starting in q_0, after a^n the automaton is in state q_0. Then after c^n it is again in q_0, and then after b^n it is again in q_0, which is an accepting state.

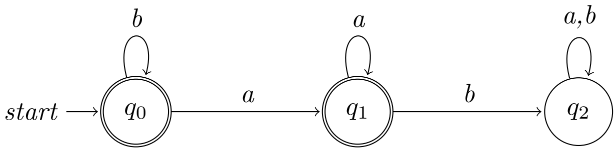
If n is odd, then starting in q_0, after a^n, the automaton is in state q_A. Then after c^n, it goes to state q_B, and after b^n back to state q_0, which is an accepting state.

End Answer Info ....

2. Let $\Sigma = \{a, b\}$. (Give DFAs by drawing their state diagrams.)

   (a) **(5 points)** Give a DFA that accepts $L_1 := \{w \in \Sigma^* \mid w \text{ does not contain } ab\}$, and prove that your answer is correct.

   Answer Info: ....



$w \in L_1$ if and only if w does not contain ab. So if after an a a b is seen, then $w \notin L_1$, and otherwise $w \in L_1$. This corresponds to the DFA, because after an a we are in state q_1, and then after a b we are in the non-accepting state q_2.

Using invariants for the states, we note that we have the following a invariants. (That is: when we are in that state, the word w read so far satisfies the announced property.)

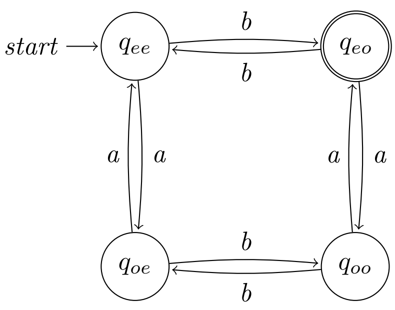
q_0	*w does not contain ab and the last symbol of w is not a*
q_1	*w does not contain ab and the last symbol of w is a*
q_2	*w has ab and the last symbol of w is not a*

It is a straightforward check that the invariants are preserved by the state transitions. We conclude that, after reading w we are in q_0 or q_1 if and only if w does not contain ab.

End Answer Info ....

   (b) **(5 points)** Give a DFA that accepts $L_2 := \{w \in \Sigma^* \mid |w|_a \text{ is even and } |w|_b \text{ is odd}\}$ and prove that your answer is correct.

   Answer Info: ....



In the DFA the first index of the state indicated whether the number of as is even or odd, and the second index indicates whether the number of bs is even or odd. Using invariants:

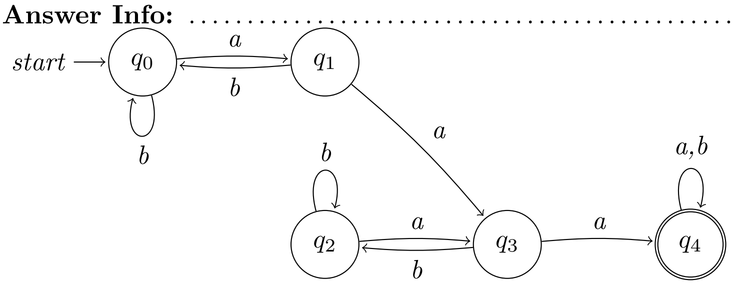
q_{ee}	$\|w\|_a$ *is even and* $\|w\|_b$ *is even*
q_{eo}	$\|w\|_a$ *is even and* $\|w\|_b$ *is odd*
q_{oe}	$\|w\|_a$ *is odd and* $\|w\|_b$ *is even*
q_{oo}	$\|w\|_a$ *is odd and* $\|w\|_b$ *is odd*

That the invariants are preserved is obvious. The state q_{eo} *is the only state where the invariant implies the required property.*

End Answer Info ....

(c) **(5 points)** Give a DFA that accepts $L_3 := \{w \in \Sigma^* \mid w \text{ contains } aa \text{ twice}\}$. (Beware of *aaa*.)

Answer Info: ....



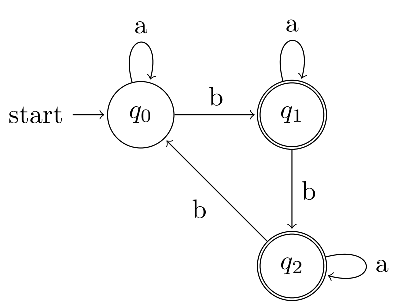
After the first aa, the DFA is in state q_3. *One more a means that we go to state* q_4, *and that the input contains aaa, and so contains aa twice. If instead a b follows, then we go to* q_2, *and only end in* q_4 *after another aa subword. When in the state* q_4 *the consumed prefix contains aa twice, and this will remain the case regardless of the rest of the input. Using invariants:*

q_0	*w does not contain aa and the last symbol is not a*
q_1	*w does not contain aa and the last symbol is a*
q_2	*w contains aa once and the last symbol is b*
q_3	*w contains aa once and the last symbol is a*
q_4	*w contains aa twice*

It is a straightforward check that the invariants are preserved by the state transitions. We conclude that, w contains aa twice if and only if, after reading w we are in q_4.

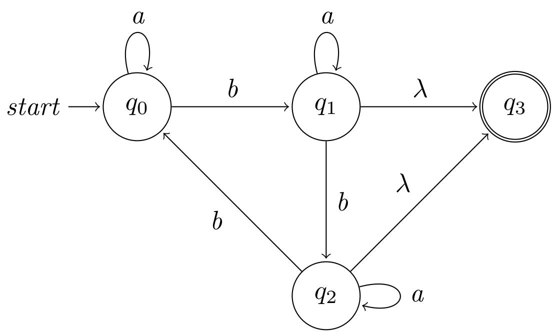
End Answer Info ....

3. **(5 points)** Consider the DFA M below and compute a regular expression e such that $\mathcal{L}(M) = \mathcal{L}(e)$, using the method that was shown at the lecture (and that is also in Section 4.2.2 of the lecture notes of Silva). Show the steps in your computation.

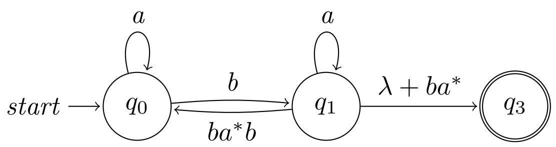


Answer Info: ....

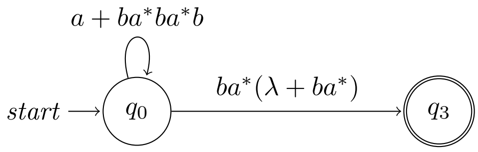
We first use a λ *transition to get a single final state:*



Now delete node q_2:



Now delete node q_1:

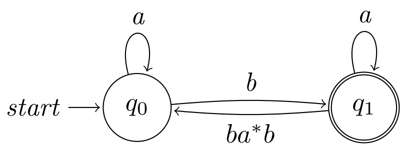


The regular expression for this automaton is

$$e' = (a + ba^*ba^*b)^*ba^*(\lambda + ba^*).$$

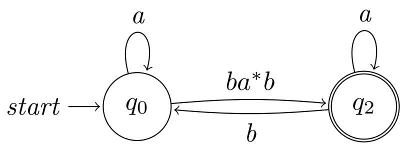
Alternatively, we can do the process twice, once for each final state.

First with only q_1 final. Then we delete node q_2 to get



The regular expression for this automaton is $a^*b(a + ba^*ba^*b)^*$.

Now with only q_2 final, delete node q_1, to get



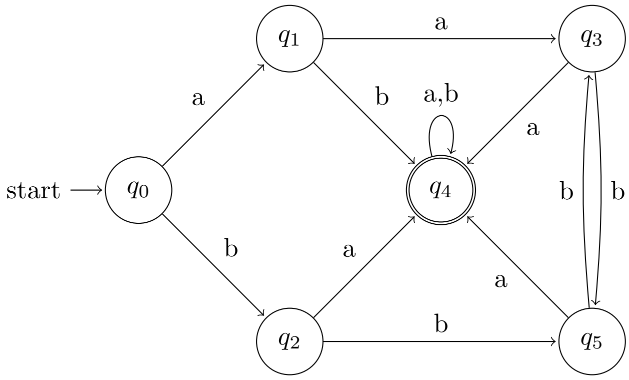
The regular expression for this automaton is $a^*ba^*b(a + ba^*ba^*b^*)$.

So another regular expression for the whole $\mathcal{L}(M)$ is

$$e = a^*b(a + ba^*ba^*b)^* + a^*ba^*b(a + ba^*ba^*b)^*.$$

End Answer Info . . . .

4. Let a DFA M over $\Sigma = \{a, b\}$, be given by



Show that the set $\{w \in \Sigma^* | w \text{ ends with } bb\} \cap \mathcal{L}(M)$ is regular by giving a regular expression for it.

Answer Info: ....

We first observe that, if $w \in \mathcal{L}(M)$ ends with bb, there are two possibilities

(a) $w = abb$

(b) *w is of the form vu, where v takes us from q_0 to q_4 and u is an arbitrary word that ends with bb.*

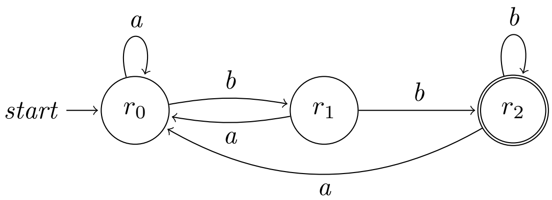
The first case is special. For the second case: $u \in \mathcal{L}((a+b)^*bb)$ and v has one of the following shapes: (i) ab, (ii) aaa (iii) ba, (iv) bba, (v) $aa(bb)^*a$, (vi) $aa(bb)^*ba$, (vii) $bb(bb)^*a$, (viii) $bb(bb)^*ba$.

We write e for $(a+b)^*bb$. Then the regular expression is

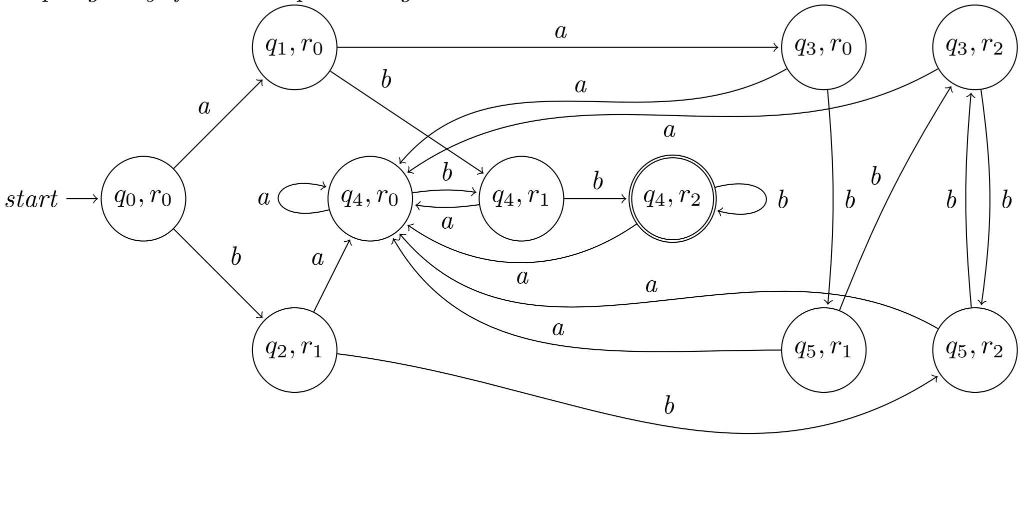
$$abb + (ab + aaa + ba + bba + aa(bb)^*a + aa(bb)^*ba + bb(bb)^*a + bb(bb)^*ba)e$$

This can be abbreviated to $abb + (ab + aaa + ba + bba + aab^*a + bbb^*a)e$.

A DFA accepting the language $\{w \in \Sigma^* | w \text{ ends with } bb\}$ is

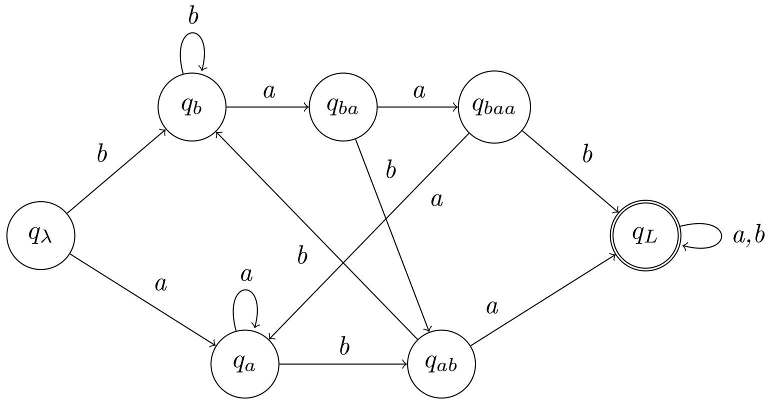


The intersection of the languages is given by running the two automata in parallel, and accepting only if both accept. This gives the automaton:



Note that in the DFA M, nodes q_3 and q_5 could be merged, because they have the same transitions and acceptance. And in the combined DFA, all states q_3, r_* and q_5, r_* could also be merged.
End Answer Info . . . .

5. (a) [More challenging question] Give a deterministic finite state automaton (DFA) with 7 states that accepts $\mathcal{L}((a+b)^*(aba+baab)(a+b)^*)$.
Answer Info: . . . .



End Answer Info . . . .

(b) [Challenge question] Let M be some finite state automaton that accepts the language $\mathcal{L}(M)$. Show that

$$L' := \{w | w \text{ is a prefix of some word } v \in \mathcal{L}(M)\}$$

can also be accepted by some finite state automaton.
(NB: w is a *prefix* of v if $wu = v$ for some word u. Phrased differently: w is a *beginpart* of v.)
Answer Info: . . . .
Suppose $M = (Q, \delta, q_0, F)$. Let F' be the set of states q from which there is a word that is accepted by M when starting from q. I.e.

$$F' = \{q \in Q \mid \exists u \in \mathcal{L}(Q, \delta, q, F)\}.$$

Phrased differently, F' is the set of states from which some state in F is reachable.

Now define $M' = (Q, \delta, q_0, F')$, so M' is the same as M, but now all these states of F' are accepting. We claim that $L' = \mathcal{L}(M')$.

Proof: Suppose that $w \in L'$. By definition that means that there is a word u such that $wu \in \mathcal{L}(M)$. After consuming w, the automata M and M' will both be in some state q. Continuing with u, the automata end in a state in F, because $wv \in \mathcal{L}(M)$. So $u \in \mathcal{L}(Q, \delta, q, F)$, hence $q \in F'$ and so $w \in \mathcal{L}(M')$.

Conversely, suppose that $w \in \mathcal{L}(M')$. That means that after consuming w, the automaton ends in a state $q \in F'$. By definition of F', there is a word $u \in \mathcal{L}(Q, \delta, q, F)$. This means that $wu \in \mathcal{L}(M)$.
End Answer Info . . . .