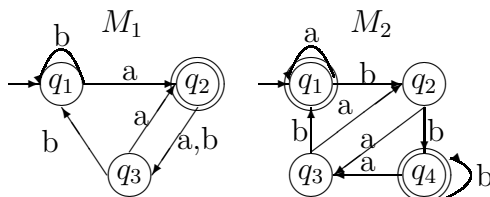


Assignment 2: Automata theory (240 points)

You are required to solve the following problems. To compete for all 200 points available you need to (1) solve all problems, (2) write your solution neatly and clearly, and (3) type your solutions on printable files.

- (10 points) The following are the state diagrams of two DFAs, M_1 and M_2 .



You are required to answer the following questions:

- (1 points) What is the start state of each automaton?
- (1 points) What are the set of accept states of each automaton?
- (3 points) What are the sequence of states each machine go through on input $aabb$?
- (3 points) Does each machine accept the string $aabb$?
- (2 points) Does each machine accept the string ϵ ?

Solution sketch: obvious

- (20 points) Give the formal description of the machines M_1 and M_2 .

Solution sketch: obvious

- (10 points) The formal description of a DFA M is $(\{q_1, q_2, q_3, q_4, q_5\}, \{u, d\}, \delta, q_3, \{q_3\})$ where δ is given by the following table: Give the state transition diagram of this

	u	d
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_2	q_4
q_4	q_3	q_5
q_5	q_4	q_5

automaton.

Solution sketch: obvious

4. (20 points) Each of the following languages is the intersection of two simpler languages. In each part, construct DFAs for simpler languages, then combine them using the construction of automata intersection to give the state diagram of DFA for the language given. In all parts $\Sigma = \{a, b\}$.

Example: if the language is $L = \{w \mid w \text{ has exactly three a's \& at least two b's}\}$ then the following construction solves the problem:

The two simpler languages are

$L' = \{w \mid w \text{ has exactly three a's}\}$ and

$L'' = \{w \mid w \text{ has at least two b's}\}$

The state diagrams of L' , L'' are in Figure 1 and their intersection is Figure 2

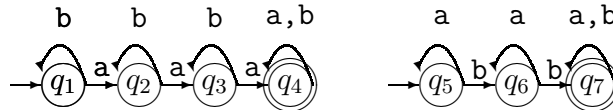


Figure 1: Transition diagrams of given languages

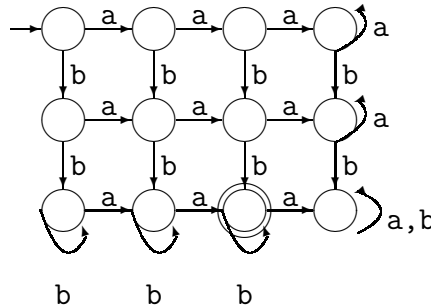


Figure 2: Transition diagram of intersection

The languages you are required to work on this problem are:

$L_1 = \{w \mid w \text{ has an even number of a's and one or two b's}\}$

$L_2 = \{w \mid w \text{ has an even number of a's \& each a is followed by at least one b}\}$.

Solution sketch: similar to the example provided above

5. (20 points) Each of the following languages is the complement of a simpler language. In each part, construct a DFA for the simpler language and then use it to give the state diagram of a DFA for the language given. In all parts $\Sigma = \{a, b\}$.

Example: if $L = \{w \mid w \text{ does not contain the substring ab}\}$ then it is the complement of the language $L' = \{w \mid w \text{ contains ab}\}$. The DFA that recognize the simpler language is the left DFA in the Figure 3 and the DFA that recognizes the complement is the right DFA in the Figure 3.

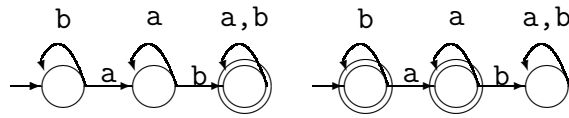


Figure 3:

The languages on which you are asked to work are:

$$L_1 = \{w \mid w \text{ does not contain the substring } \text{baba} \}$$

$$L_2 = \{w \mid w \text{ contains neither the substring } \text{ab} \text{ nor } \text{ba} \}$$

$$L_3 = \{w \mid w \text{ is any string not in } \text{a}^*\text{b}^* \}$$

$$L_4 = \{w \mid w \text{ is any string except } \text{a} \text{ and } \text{b} \}.$$

Solution sketch: similar to the example treated above

6. (10 points) Give the state diagrams of the DFAs recognizing the following languages over the alphabet $\{0, 1\}$:

$$L_1 = \{w \mid w \text{ contains the substring } 0101, \text{ i.e., } w = x0101y \text{ for some } x, y \in \{0, 1\}^*\}$$

$$L_2 = \{w \mid \text{the length of } w \text{ is at most } 5 \}$$

Solution sketch: obvious.

7. (10 points) Give the state diagrams of the NFAs recognizing the following languages over the alphabet $\{0, 1\}$:

$$L_1 = \{w \mid w \text{ ends with } 00 \text{ with three states} \}$$

$$L_2 = 0^*1^*0^+ \text{ with three states.}$$

Solution sketch: obvious.

8. (20 points) Use the construction employed to prove that class of regular languages is closed under the union operation to give the state diagram of the NFA recognizing the union of the languages $L_1 \cup L_2$ and $L_3 \cup L_4$ where: $L_1 = \{w \mid w \text{ begins with } 1 \text{ and ends with a } 0 \}$

$$L_2 = \{w \mid w \text{ contains at least three } 1\text{s} \}$$

$$L_3 = \{w \mid w = x0101y, x, y \in \{0, 1\}^*\}$$

$$L_4 = \{w \mid w \text{ does not contain the substring } 110 \}.$$

Solution sketch: reproduce the construction of *union* of regular languages for the concrete languages required above.

9. (20 points) Use the construction employed to prove that class of regular languages is closed under the concatenation operation to give the state diagram of the NFA recognizing the concatenation of the languages L_1 and L_2 where: $L_1 = \{w \mid \text{the length of } w \text{ is at most } 5\}$

$$L_2 = \{w \mid \text{every odd position of } w \text{ is a } 1 \}.$$

Solution sketch: reproduce the construction of regular language concatenation for the concrete languages L_1 and L_2 required above.

10. (20 points) Use the construction employed to prove that class of regular languages is closed under the star operation to give the state diagram of the NFA recognizing the star of the languages L_1 , L_2 , and L_3 where: $L_1 = \{w|w \text{ contains at least two 0s and at most one 1}\}$

$$L_2 = \emptyset$$

$$L_3 = \{w|w \text{ contains at least three 1s}\}.$$

Solution sketch: reproduce the construction of star operation applied on regular languages for the concrete languages L_1 , L_2 , L_3 specified above.

11. (20 points) Prove that every NFA can be converted to an equivalent NSA that has a single accept state.

Solution sketch: Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA. Construct the NFA N' with a single accept state that accept the same language as N . Intuitively N' is exactly like N except it has ϵ -transitions from the states corresponding to accept states of N to the new accept state q_{accept} . The state q_{accept} has no emerging transitions. Formally, $N' = (Q \cup \{q_{accept}\}, \Sigma, \delta', q_0, \{q_{accept}\})$ where for each $q \in Q$ and $a \in \Sigma$ we have:

$$\delta'(q, a) = \begin{cases} \delta(q, a) & \text{if } a \neq \epsilon \text{ or } q \notin F; \\ \delta(q, a) \cup \{q_{accept}\} & \text{if } a = \epsilon \text{ and } q \in F; \\ \emptyset & \text{for each } a \in \Sigma_\epsilon \text{ and } q = q_{accept}. \end{cases}$$

12. (20 points) Let $D = \{w|w \text{ contains an even number of a's and an odd number of bs and does not contain the substring } ab\}$. Give a NFA with five state that recognizes D .

Solution sketch:

- The condition w does not contain ab implies that w cannot start with a .
- Also, the same condition implies that once a substring of a-s occur in w this cannot be followed by a substring of b-s.

The NFA is in Figure 4

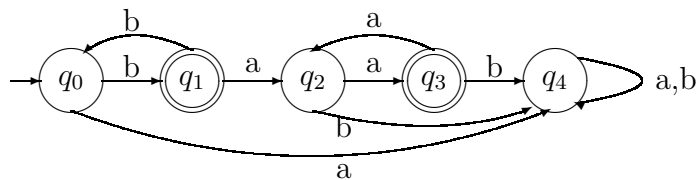


Figure 4:

13. (20 points) Show that, if M is a DFA that recognizes a language B , swapping the accept and non-accept states in M yields a new DFA M' that recognizes the complement of B . Conclude that the class of regular languages is closed under complement operation.

Solution sketch: suppose that M' accepts a string x , i.e., if we run M' of x we end in an accept state of M' . Because M and M' have swapped accept/non-accept states, if we run M on x we would end in a non-accept state. Therefore $x \notin B$. Similarly, if x is not accepted by M' , it would be accepted by M . Hence, M' accepts exactly those strings that are not accepted by M . Therefore M' recognizes the complement of B .

Conclusion: since B could be any arbitrary regular language, our construction shows how to build an automaton that recognize the complement of a regular language B . Hence, the complement of a regular language is regular. Therefore the class of regular language is closed under complement operation.

14. (20 points) Show by an example that if M is an NFA that recognizes a language C , swapping the accept and non-accept states in M does not necessarily yield a new NFA that recognize the complement of C . Is the class of languages recognized by NFAs closer under complement? Explain your answer.

Solution sketch: consider the NFA in Figure 5: The string aa is accepted

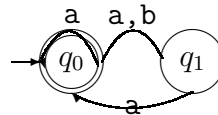


Figure 5:

by this automaton. If we swap the accept and non-accept states the string aa is still accepted. This shows that swapping the accept and non-accept states of an NFA does not necessarily yield an new NFA that recognize the complement of the language recognized by the original NFA. But the class of languages recognized by NFA is closed under complement. of languages recognized by DFA which we know is closed under the complement from the problem above.

Due date: 30 September 2009