Exam 1 — CSU 390 Theory of Computation — Fall 2007

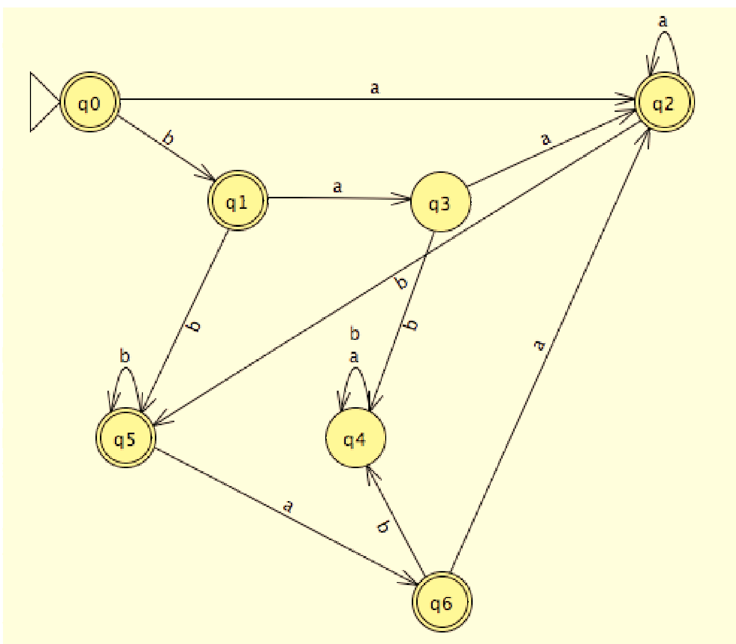
Solutions

Problem 1 [10 points]

Construct a state transition diagram for a DFA that recognizes the following language over the alphabet $\Sigma = \{a, b\}$:

$L_1 = \{w | w \neq ba$, and w does not contain $bab\}$

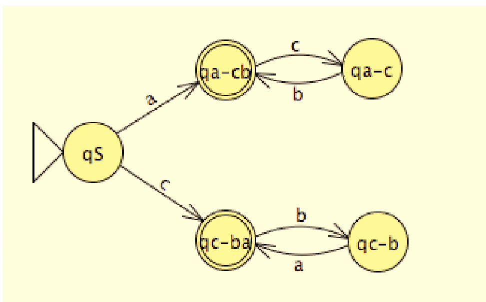
Solution:



Problem 2 [10 points]

Construct a state transition diagram for 5-state NFA that recognizes the language given by the regular expression $a(cb)^* \cup c(ba)^*$.

Solution:



Problem 3 [10 points]

Give a regular expression for the language L_2 over the alphabet $\Sigma = \{0, 1\}$:

$L_2 = \{w|w$ starts with 1, ends with 0, contains an even number of *substrings* 01$\}$

Solution:

$1(0+1+0+1+)*0$

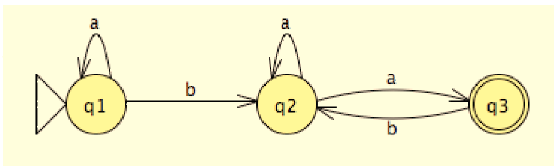
Note: I omitted the word *substrings* on the exam — it was shown on the blackboard. However, a large number of people misread the problem. Therefore, the solution:

$1(0101)+0$

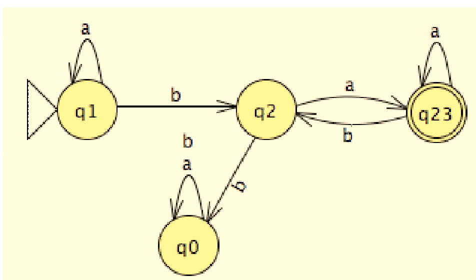
was considered *nearly correct* for 9 out of 10 points.

Problem 4 [10 points]

Convert the following NFA to an equivalent DFA using the technique shown in class (and in Example 1.41 of the Sipser text). In your final result, show only those states reachable from the start state.

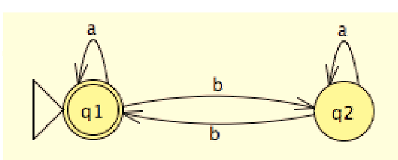


Solution:



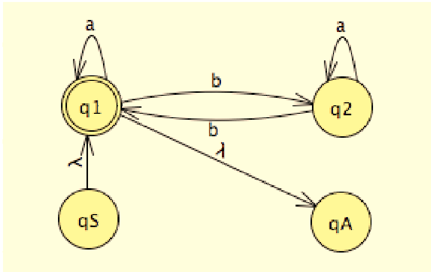
Problem 5 [10 points]

Convert the following DFA to an equivalent regular expression using the technique shown in class (and in Examples 1.66 and 1.68 of the Sipser text). *Eliminate states as follows:* First eliminate state 2, then state 1.



Solution:

Initial GNFA:



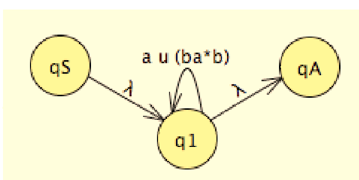
$qS : \epsilon \longrightarrow q2 : a \longrightarrow q2 : \emptyset \longrightarrow qA$
$qS : \emptyset \longrightarrow qA$
get: $qS : \emptyset \longrightarrow qA$

$qS : \epsilon \longrightarrow q2 : a \longrightarrow q2 : b \longrightarrow q1$
$qS : \epsilon \longrightarrow q1$
get: $qS : \epsilon \longrightarrow q1$

$q1 : b \longrightarrow q2 : a \longrightarrow q2 : b \longrightarrow q1$
$q1 : a \longrightarrow q1$
get: $q1 : a \cup (ba^*b) \longrightarrow q1$

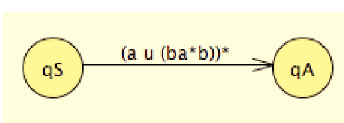
$q1 : b \longrightarrow q2 : a \longrightarrow q2 : b\emptyset \longrightarrow qA$
$q1 : a \longrightarrow qA$
get: $q1 : \epsilon \longrightarrow qA$

After state 2 has been eliminated:



$qS : \epsilon \longrightarrow q1 : a \cup (ba^*b) \longrightarrow q1 : \epsilon \longrightarrow qA$
$qS : \emptyset \longrightarrow qA$
get: $qS : (a \cup (ba^*b))^* \longrightarrow qA$

After state 1 has been eliminated:



The corresponding regular expression is: $(ba^*b))^*$

Problem 6 [30 points — 5, 5, 10, 10]

The *min* of a language L is defined as

$min(L) = \{w \in L |$ there is no $u \in L, v \in \Sigma^+$ such that $w = uv\}$

1. Define $min(L_3)$ for the language L_3 defined by aba^*b.

   Solution:

   $min(L_3)$ is defined by aba^*b, i.e. $min(L_3) = L_3$.

2. Define $min(L_4)$ for the language L_4 defined by $a(ba)^+$.

   Solution:

   $min(L_4) = \{aba\}$

3. Consider $min(L_5)$ for the language L_5 defined by $b^*(aa)^+b^*$.

   - Show all strings of length ≤ 5 five in the language $min(L_5)$.

     Solution:
     $aa, baa, bbaa, bbbaa$

   - Define the language $min(L_5)$.

     Solution:
     L is defined by b^*aa

4. Prove that regular languages are closed under the operation *min*.

   Hint: Construct the DFA for the language $min(L)$ from the DFA for the language L.

   Solution:

   Construct an $DFA = (Q, \Sigma, \delta, q_0, F)$ to represent the language L. Construct a new $DFA_{min} = Q \cup q_{sink}, \Sigma, \delta_{min}, F)$ to represent the language $min(L)$ with δ_{min} defined as follows:

   - $\delta_{min}(q_i, q_j) = \delta(q_i, q_j)$ for all $q_i \notin F$.
   - $\delta_{min}(q_i, q_j) = \delta(q_i, q_{sink})$ for all $q_i \in F$ and $q_j \in Q$.
   - For every $a \in \Sigma$ add the transitions $\delta_{min}(q_{sink}, q_{sink}) = a$.

That means every transition from an accept state is replaced with a new transition with the same label that leads to a new *sink* state q_{sink}. To complete the DFA we add transitions with the labels for all letters of alphabet emanating from q_{sink} lead to q_{sink}.

The resulting DFA_{min} accepts every word w in L that traverses only through non-accept states except for the last transition that leads to an accept state.

If there was a $u \in L, v \in \Sigma^+$ such that $w = uv\}$, then the last transition in u ends in one of the original accept states. However, in our DFA_{min} there are no transitions from the acccept states that lead later to an accept state, and so there is no $v \in \Sigma^+$ that would result in a word $uv \in L_{min}$. *QED*

Alternate Solution:

Regular languages are closed under complement and under concatenation. The language $L \circ \Sigma^+$ is regular. L_{min} = complement of $L \circ \Sigma^+$, and so it also must be regular. *QED*

Problem 7 [20 points — 5, 15]

1. Consider the following language over the alphabet $\Sigma = \{0, 1\}$: $L_6 = \{www | w \text{ starts with } 0\}$

   Asked to determine whether this language is regular, Amos Mart argues as follows: Language $L = \{w | w \text{ starts with } 0\}$ is regular. Regular languages are closed over concatenation, and because $L_6 = L \circ L \circ L$, L_6 is also regular.

   This argument is not correct. Explain what is wrong with it.

   Solution:

   $L \circ L \circ L = \{w_1 w_2 w_3 | w_i \in L\}$ does not say anything about $w_1 = w_2 = w_3$, so the statement that $L_6 = L \circ L \circ L$ is false.

2. Prove that the language L_6 is not regular.

Solution:

We use the pumping lemma. For any p we select the string $s = www$ where $w = 01^p$, i.e. $s = 01^p01^p01^p$.

The $xy = 01^k$ for some $1 \leq k < p$ and then we can only choose $x = 01^k$ where $0 \leq k < p-1$ and either $y = 1^m$ or $x = \epsilon$ and $y = 01^k$.

In the first case $xy^nz = 01^{m \times n}1^{p-k}01^p01^p$ where $m \times n + (p-k) > p$ and so the string s cannot be pumped this way.

In the second case $xy^nz = (01^k)^n1^{p-k}01^p01^p$ which is definitely not a string in L_6 — and so the string s cannot be pumped this way either.

That means the string $w = 01^p$ cannot be pumped and the language cannot be a regular language.