Problems Solved:

6	7	8	9	10

Name:

Matrikel-Nr.:

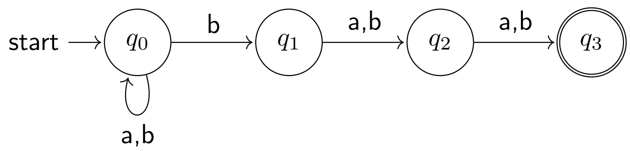
RNonDeterministic

Problem 6. Let L be the set of all strings $x \in \{a, b\}^*$ with $|x| \geq 3$ whose third symbol from the right is b. For example, $babaa$ and bbb are elements of L, but bb and $baba$ are not.

1. Construct the transition graph of a NFSM N such that $L(N) = L$. (4 states are sufficient.)
2. Construct the transition graph of a DFSM D such that $L(D) = L$. (8 states are sufficient.)

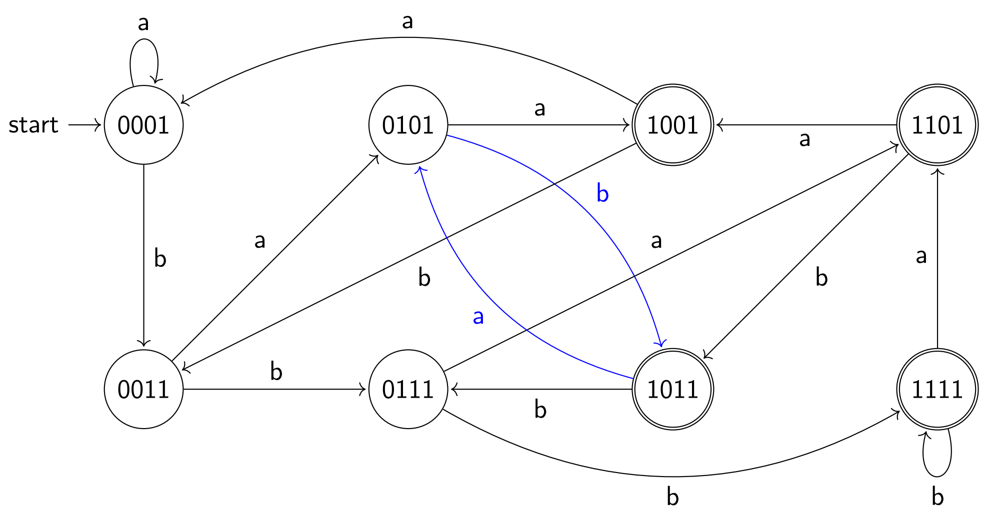
Solution of Problem 6:

1. It is clearly enough if N has just 4 states.



2. By naive consideration 16 states should be sufficient (subset construction), but for turning N into a deterministic automaton, it is easily seen that 8 states are sufficient, because q_0 must be contained in each of the (state-)subsets.

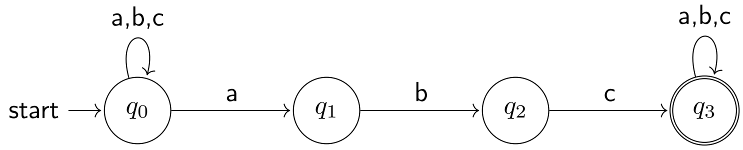
   The state name $rstu$ corresponds to the characteristic value of the state set, i.e., 0101 corresponds to $\{q_2, q_0\}$. Some arrows are blue for better visibility.



DEAnot-abc

Problem 7. Construct the transition graph of a deterministic finite state machine M over $\Sigma = \{a, b, c\}$ such that $L(M)$ consists of all words that do not contain the string *abc*.
Hint: Start by constructing a nondeterministic finite state machine N that recogizes the words that *do* contain the string *abc*. Proceed by converting your nondeterministic machine N to a deterministic machine D that accepts the same language. Now you are left with the task of coming up with a machine M whose language is precisely the complement of the language of D. This can be done by a small modification of D.

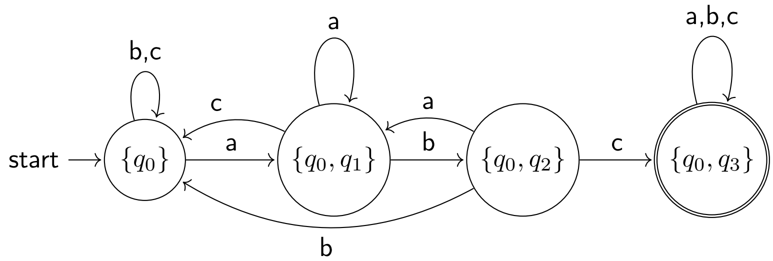
Solution of Problem 7:



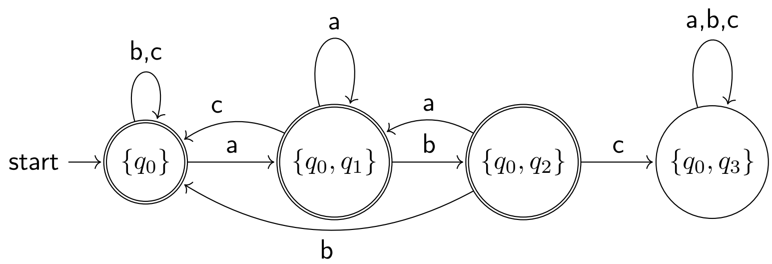
Problem: Complementing F to get the complement of the language does not work for nondeterministic machines as the one above.

Therefore we first construct a DFSM D that accepts abc and then construct its complement.

The automaton below can be obtained by applying the subset construction to the one above.



We finally construct the complement.

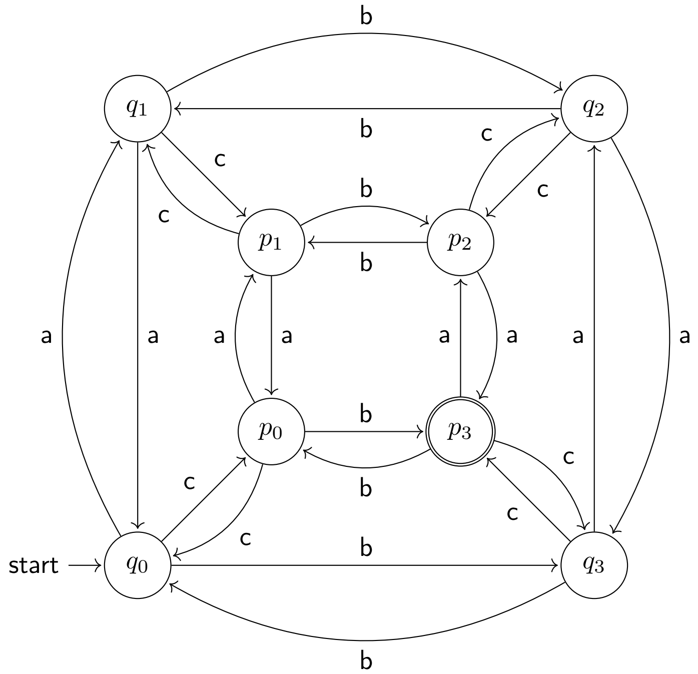


DEA-Even-odd-odd

Problem 8. Construct the transition graph of a deterministic finite state machine $D = (Q, \Sigma, \delta, S, F)$ with alphabet $\Sigma = \{a, b, c\}$, such that the words of $L(D)$ contain an even number of a's, an odd number of b's, and an odd number of c's. For example, *aabccc*, *cacbac*, *acabaabb* are from $L(D)$ and *babc*, *ccabab*, *caacbaabba* are not from $L(D)$.

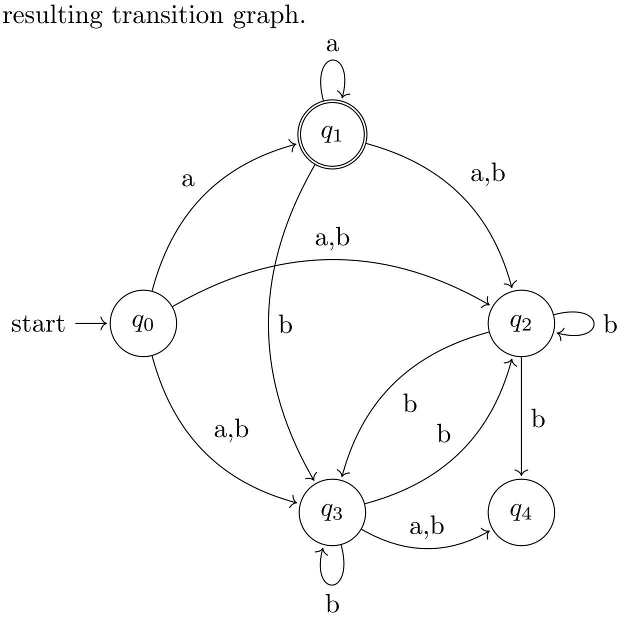
Solution of Problem 8:

The graph has a form of a cube with two floors each of which has four states.

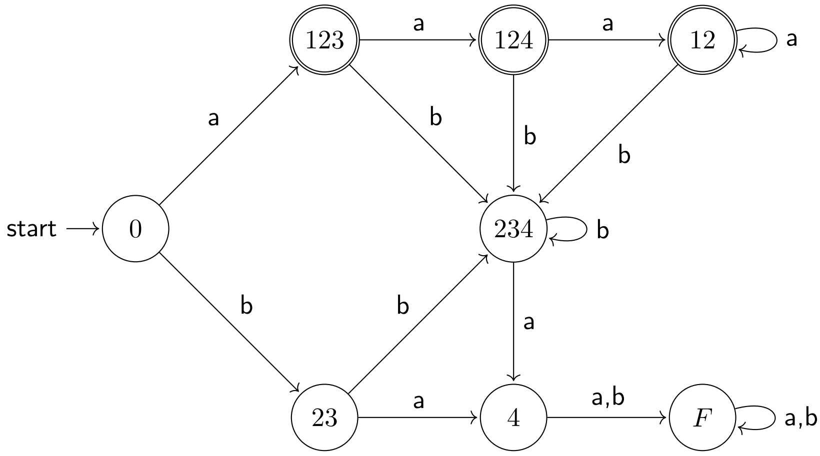


NFA2DFA

Problem 9. Convert the following NFSM to DFSM. It suffices to give the resulting transition graph.

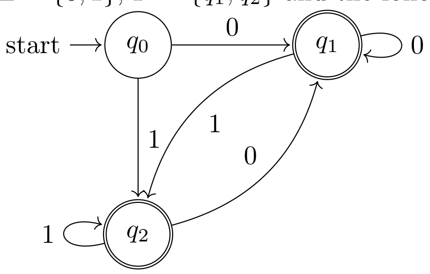


Solution of Problem 9:



DFSMmin

Problem 10. Let the DFSM $M = (Q, \Sigma, \delta, q_0, F)$ be given by $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{0, 1\}$, $F = \{q_1, q_2\}$ and the following transition function $\delta : Q \times \Sigma \to Q$:

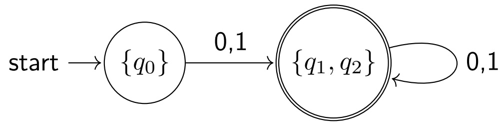


Construct a minimal DFSM D such that $L(M) = L(D)$ using Algorithm Minimize. (cf. Section 2.3 *Minimization of Finite State Machines*) Provide its transition graph as well.

Solution of Problem 10:

We execute $\textsc{Minimize}(Q, \Sigma, \delta, q_0, F)$.

1. There are no redundant states.
2. $Q' = \textsc{Partition}(Q, \Sigma, \delta, q_0, F)$
   (a) $P := \{\{q_1, q_2\}, \{q_0\}\}$
   (b) $S := P$
   (c) $P := \emptyset$
   (d) $p := \{q_1, q_2\}$
   (e) $P := P \cup \left\{[s]_p^S \,\middle|\, s \in p\right\} = \{\{q_1, q_2\}\}$
   (f) $[q_1]_p^S = p$
   (g) $[q_2]_p^S = p$
   (h) $p := \{q_0\}$
   (i) $P := P \cup \left\{[s]_p^S \,\middle|\, s \in p\right\} = \{\{q_1, q_2\}\} \cup \{\{q_0\}\}$
   (j) $[q_1]_{\{q_0\}}^S = p = \{q_0\}$
3. $Q' = \{\{q_1, q_2\}, \{q_0\}\}$


start
0,1
$\{q_0\}$
$\{q_1, q_2\}$
0,1