Automata and Formal Languages — Exercise Sheet 11

Exercise 11.1

Give Büchi automata (NBA) for the following ω-languages:

- $L_1 = \{w \in \{a,b\}^\omega : w$ contains infinitely many a's$\}$,
- $L_2 = \{w \in \{a,b\}^\omega : w$ contains finitely many b's$\}$,
- $L_3 = \{w \in \{a,b\}^\omega :$ each occurrence of a in w is (immediately) followed by a $b\}$.

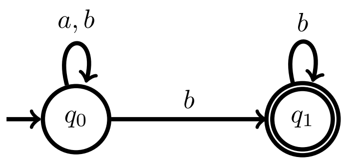
Intersect these automata and decide if the obtained automaton is the smallest Büchi automaton for $L_1 \cap L_2 \cap L_3$.

Exercise 11.2

Give algorithms that directly complement deterministic Muller and parity automata, without going through Büchi automata.

Exercise 11.3

(a) Consider the following Büchi automaton A over $\Sigma = \{a, b\}$:



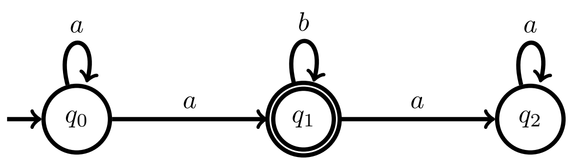
Draw $\mathrm{dag}(abab^\omega)$ and $\mathrm{dag}((ab)^\omega)$.

(b) Let r_w be the ranking of $\mathrm{dag}(w)$ defined by

$$r_w(q,i) = \begin{cases} 1 & \text{if } q = q_0 \text{ and } \langle q_0, i\rangle \text{ appears in } \mathrm{dag}(w), \\ 0 & \text{if } q = q_1 \text{ and } \langle q_1, i\rangle \text{ appears in } \mathrm{dag}(w), \\ \perp & \text{otherwise.} \end{cases}$$

Are r_{abab^ω} and $r_{(ab)^\omega}$ (over A) odd rankings?

(c) Consider the following Büchi automaton B over $\Sigma = \{a, b\}$:



Draw $\mathrm{dag}(a^\omega)$. Show that any odd ranking for this dag must contain a node of rank 3 or more.

(d) Consider again the automaton A from (a). Let w be an ω-word and r_w the ranking of $\mathrm{dag}(w)$ as defined in (b). Show that r_w is an odd ranking for $\mathrm{dag}(w)$ if and only if $w \notin L_\omega(A)$.

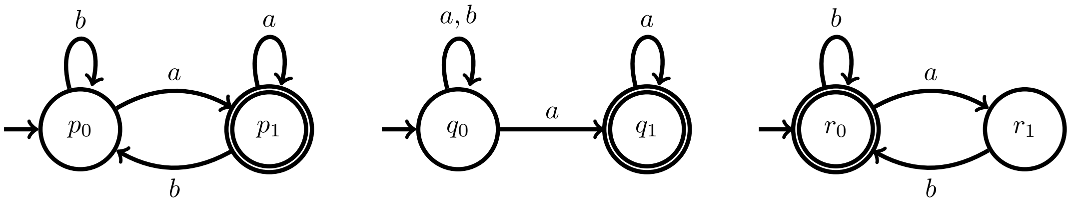
(e) Construct a Büchi automaton accepting $\overline{L_\omega(A)}$ using the construction seen in class.
Hint: by (d), it is sufficient to use $\{0, 1\}$ as ranks.

Exercise 11.4

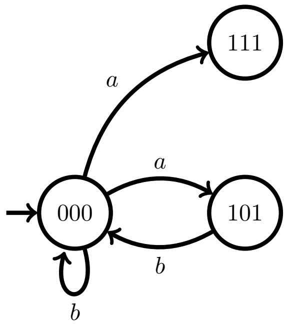
Show that for every DBA A with n states there is an NBA B with $2n$ states such that $B = \overline{A}$. Explain why your construction does not work for NBAs.

Solution 11.1

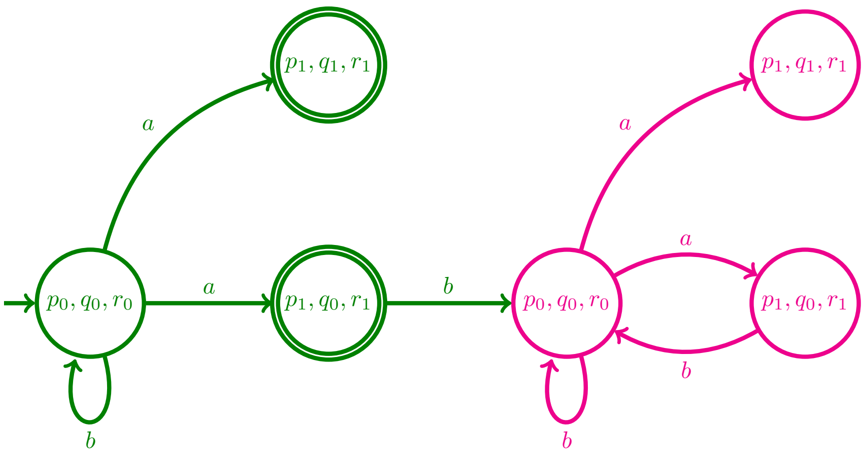
The following Büchi automata respectively accept L_1, L_2 and L_3:



Taking the intersection of these automata leads to the following generalized Büchi automaton with the acceptance condition $\{\{101, 111\}, \{111\}, \{000\}\}$, where 000 is the state p_0, q_0, r_0, 101 is the state p_1, q_0, r_1, and 111 is the state p_1, q_1, r_1.



If we convert this GBA to an NBA, we obtain the following Büchi automaton:



Note that the language of this automaton is the empty language. Therefore, the obtained automaton is surely not the smallest NBA accepting the empty language.

Solution 11.2

Let us consider the case of a deterministic Muller automaton A with acceptance condition $\mathcal{F} = \{F_0, \ldots, F_{m-1}\} \subseteq 2^Q$. Since every ω-word w has a single run ρ_w in A, we have $w \notin L_\omega(A)$ iff $\inf(\rho_w) \in 2^Q \setminus \mathcal{F}$. Thus, to complement A, we change its acceptance condition to $\mathcal{F}' = 2^Q \setminus \mathcal{F}$.

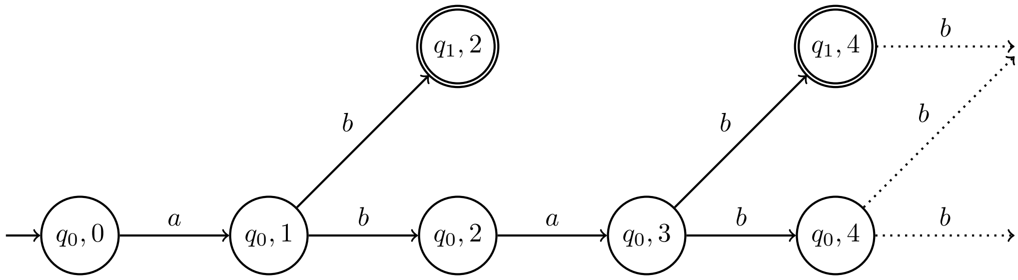
Let us consider the case of a deterministic parity automaton A with acceptance condition $F_1 \subseteq \cdots \subseteq F_{2n}$. Since every ω-word w has a single run ρ_w in A, we have

$$w \in L_\omega(A) \iff \min\{i : \inf(\rho_w) \cap F_i \neq \emptyset\} \text{ is even.}$$

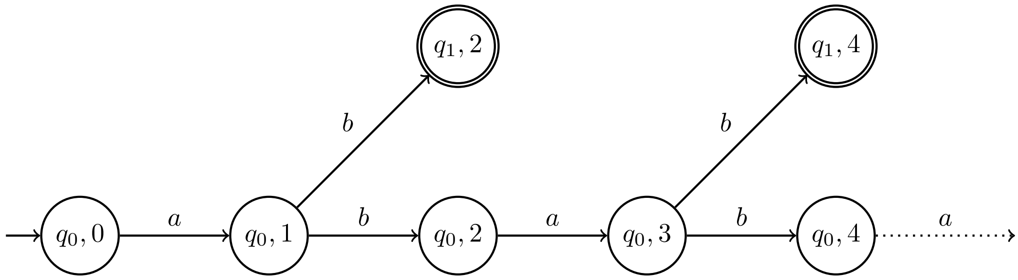
Thus, to complement A, it suffices to "swap the parity" of states. This can be achieved by adding a new dummy state $q_\perp$ to A and changing its acceptance condition to $\{q_\perp\} \subseteq (F_1 \cup \{q_\perp\}) \subseteq \cdots \subseteq (F_{2n} \cup \{q_\perp\})$, where the purpose of $q_\perp$ is to keep the chain of inclusion required by the definition.

Solution 11.3

(a) $\mathrm{dag}(abab^\omega)$:



$\mathrm{dag}((ab)^\omega)$:



(b)
- r *is not* an odd rank for $\mathrm{dag}(abab^\omega)$ since

$$\langle q_0, 0\rangle \xrightarrow{a} \langle q_0, 1\rangle \xrightarrow{b} \langle q_0, 2\rangle \xrightarrow{a} \langle q_0, 3\rangle \xrightarrow{b} \langle q_1, 4\rangle \xrightarrow{b} \langle q_1, 5\rangle \xrightarrow{b} \cdots$$

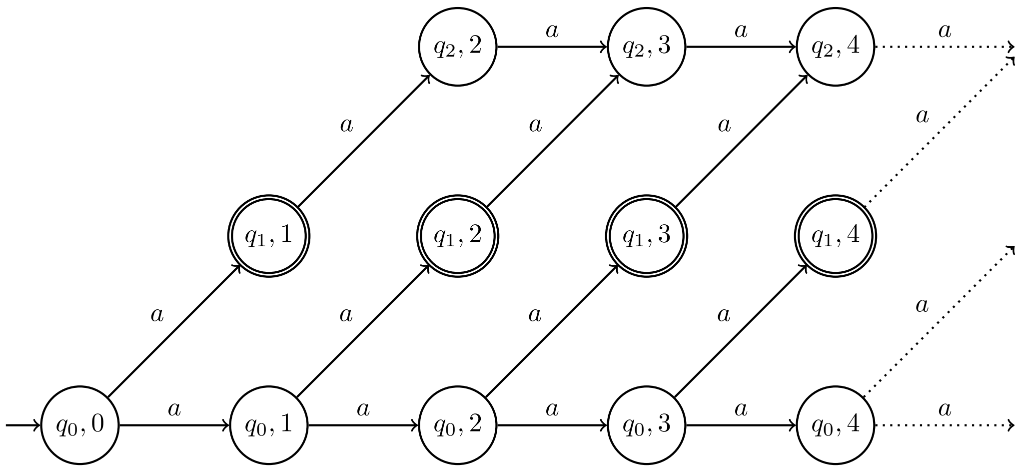
  is an infinite path of $\mathrm{dag}(abab^\omega)$ not visiting odd nodes infinitely often.

- r *is* an odd rank for $\mathrm{dag}((ab)^\omega)$ since it has a single infinite path:

$$\langle q_0, 0\rangle \xrightarrow{a} \langle q_0, 1\rangle \xrightarrow{b} \langle q_0, 2\rangle \xrightarrow{a} \langle q_0, 3\rangle \xrightarrow{b} \langle q_0, 4\rangle \xrightarrow{a} \langle q_0, 5\rangle \xrightarrow{b} \cdots$$

  which only visits odd nodes.

(c) $\mathrm{dag}(a^\omega)$:



Let r be an odd rank for $\mathrm{dag}(a^\omega)$. It exists since a^ω is not accepted by B. Since r is odd, all infinite paths must visit odd nodes infinitely often (i.o.). In particular the bottom infinite path of q_0 nodes must stabilize to nodes with odd rank.

Let us assume the nodes $\langle q_0, j\rangle$ have rank 1 for all $j \geq i$ for some $i \geq 0$. Consider the infinite path $\rho = \langle q_0, i\rangle \xrightarrow{a} \langle q_1, i+1\rangle \xrightarrow{a} \langle q_2, i+2\rangle \xrightarrow{a} \langle q_2, i+3\rangle \ldots$. Node $\langle q_1, i+1\rangle$ must have an even rank (since q_1 is accepting) smaller or equal to 1, so it has rank 0. This entails that $\langle q_2, k\rangle$ has rank 0 for all $k \geq i+2$. This contradicts r being an odd ranking because the path ρ is infinite yet does not visit odd nodes infinitely often.

Thus the bottom infinite path of q_0 nodes must stabilize to nodes with odd rank strictly bigger than 1, i.e., bigger or equal to 3.

(d) $\Rightarrow$) (By contraposition) Let $w \in L_\omega(B)$. We have $w = ub^\omega$ for some $u \in \{a,b\}^*$. This implies that

$$\langle q_0, 0\rangle \xrightarrow{u} \langle q_0, |u|\rangle \xrightarrow{b} \langle q_1, |u|+1\rangle \xrightarrow{b} \langle q_1, |u|+2\rangle \xrightarrow{b} \cdots$$

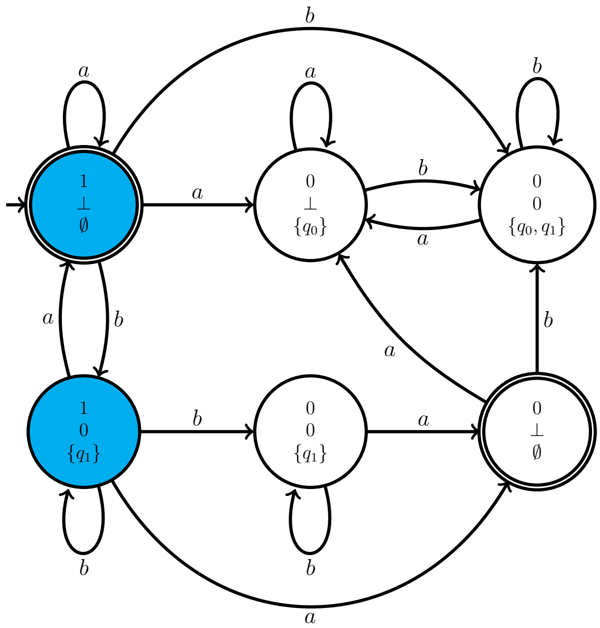
is an infinite path of $\mathrm{dag}(w)$. Since this path does not visit odd nodes infinitely often, R is not odd for $\mathrm{dag}(w)$.

$\Leftarrow$) Let $w \notin L_\omega(B)$. Suppose there exists an infinite path of $\mathrm{dag}(w)$ that does not visit odd nodes infinitely often. At some point, this path must only visit nodes of the form $\langle q_1, i\rangle$. Therefore, there exists $u \in \{a,b\}^*$ such that

$$\langle q_0, 0\rangle \xrightarrow{u} \langle q_1, |u|\rangle \xrightarrow{b} \langle q_1, |u|+1\rangle \xrightarrow{b} \langle q_1, |u|+2\rangle \xrightarrow{b} \cdots$$

This implies that $w = ub^\omega \in L_\omega(B)$ which is contradiction.

(e) Recall: we construct an NBA whose runs on an ω-word w are all the valid rankings of $dag(w)$. The automaton accepts a ranking R iff every infinite path of R visits nodes of odd rank i.o. By (d), for every $w \in \{a,b\}^\omega$, if $\mathrm{dag}(w)$ has an odd ranking, then it has one ranging over 0 and 1. Therefore, it suffices to execute *CompNBA* with rankings ranging over 0 and 1. We obtain the following Büchi automaton, for which some intuition is given below:



General explanation: Any ranking r of $dag(w)$ can be decomposed into a sequence $lr_1, lr_2, \ldots$ such that $lr_i(q) = r(< q, i >)$, the level i of rank r. Recall that in this automaton, the transitions $\begin{bmatrix} lr(q_0) \\ lr(q_1) \end{bmatrix} \xrightarrow{a} \begin{bmatrix} lr'(q_0) \\ lr'(q_1) \end{bmatrix}$ represent the possible next level for ranks r such that $lr(q) = r(< q, i >)$ and $lr'(q) = r(< q, i+1 >)$ for $q = q_0, q_1$.

The additional set of states in the automaton represents the set of states that "owe" a visit to a state of odd rank. Formally, the transitions are the triples $[lr, O] \xrightarrow{a} [lr', O']$ such that $lr \xrightarrow{a} lr'$ and $O' = \{q' \in \delta(O, a) | lr'(q') \textit{ is even}\}$ if $O \neq \emptyset$, and $O' = \{q' \in Q | lr'(q') \textit{ is even}\}$ if $O = \emptyset$.

Finally the accepting states of the automaton are those with no "owing" states, which represent the *breakpoints* i.e. a moment where we are sure that all runs on w have seen an odd rank since the last breakpoint.

Specific to this example: The states of this automaton are triples (x, y, S), where x is the rank of q_0, y is the rank of q_1, and $S \subseteq \{q_0, q_1\}$ is the set of "owing" states, that is, those that owe a visit to a state with an odd rank (since the last breakpoint). Our hint suggests that x and y can be either 0 or 1, or $\perp$ if the state is not present. Without hint we would have to consider all possibilities, that is, $x, y \in \{0, 1, 2, 3, 4\} \cup \{\perp\}$.

The initial state has x set to be maximal possible, that is, 1 (because of the hint, otherwise 4), as q_0 is the initial state in the original automaton A. As q_1 is not initial in A, it is not initially present, and thus y is set to $\perp$. No state is owing a visit to an odd-rank-state, since we have only one present state q_0 with an odd rank 1. Thus the third component is $\emptyset$.

Transitions are created following the general explanation from above. For example, there are two transitions from the initial state with letter b, that is $[1, \perp, \emptyset] \xrightarrow{b} [0, 0, \{q_0, q_1\}]$ and $[1, \perp, \emptyset] \xrightarrow{b} [1, 0, \{q_1\}]$. This is because by reading the letter b from q_0 with rank 1 (q_1 is not present in $[1, \perp, \emptyset]$), (i) we can reach q_0 and assign any rank not higher than the previous rank of q_0, that is, either 0 or 1, and (ii) we can reach q_1 which will have to have an even rank since it is the accepting state of A, and in our case the only option is 0. If we assign rank 0 to q_0, then both states q_0 and q_1 will have an even rank, so both of them will be the "owing states", so we will reach the state $[0, 0, \{q_0, q_1\}]$. If we assign rank 1 to q_0, the only "owing state" will be q_1, so we will reach the state $[1, 0, \{q_1\}]$.

The recipe for calculating S' in $[x, y, S] \xrightarrow{c} [x', y', S']$ is this: If $S = \emptyset$ then S' is the set of those states that have an even rank (after the transition); for example, in $[1, \perp, \emptyset] \xrightarrow{b} [0, 0, \{q_0, q_1\}]$ both states have even rank 0 after reading b so both are in S'. If $S \neq \emptyset$ then S' is the set of the states from S that have an even rank (after the transition), for example, in $[1, \perp, \{q_1\}] \xrightarrow{b} [0, 0, \{q_1\}]$ both states have even rank 0 after reading b, but since q_0 was not in S, we have $S' = \{q_1\}$.

The accepting states are breakpoints, those with $S = \emptyset$.

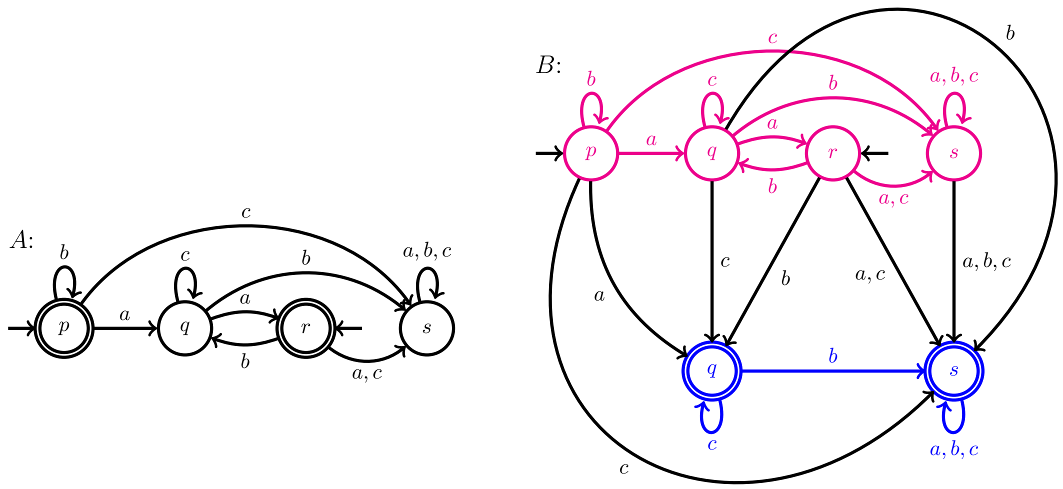
★ It is enough to only consider the blue states (as the part (d) of this exercise suggests), as any other state cannot reach a level in which there is an odd rank; descendants of *dag* states with rank 0 can never be assigned an odd rank.

Solution 11.4

Observe that A rejects a word w iff its *single* run on w stops visiting accepting states at some point. Hence, we construct an NBA B that reads a prefix as in A and non deterministically decides to stop visiting accepting states by moving to a copy of A without its accepting states.

More precisely, we assume that each letter can be read from each state of A, i.e. that A is complete. If this is not the case, it suffices to add a rejecting sink state to A. The NBA B consists of two copies of A. The first copy is exactly as A. The second copy is as A but restricted to its non accepting states. We add transitions from the first copy to the second one as follows. For each transition (p, a, q) of A, we add a transition that reads letter a from state p of the first copy to state q of the second copy. All states of the first copy are made non accepting and all states of the second copy are made accepting. Note that B contains at most $2n$ states as desired.

Here is an example of the construction:



This construction does not work on NBAs. Indeed, we have $A = B = \{a^\omega\}$ below: