Automata and Formal Languages — Exercise Sheet 13

Exercise 13.1

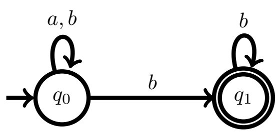
(a) Give deterministic Büchi automata for L_a, L_b, L_c where $L_\sigma = \{w \in \{a,b,c\}^\omega : w \text{ contains infinitely many } \sigma\text{'s}\}$, and intersect these automata.

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

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

and intersect these automata. Decide if this automaton is the smallest Büchi automaton for that language.

Exercise 13.2
Consider the following Büchi automaton over $\Sigma = \{a, b\}$:



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

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

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

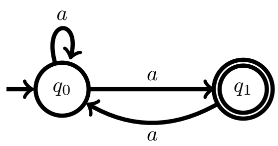
Are r_{abab^ω} and $r_{(ab)^\omega}$ odd rankings?

(c) Show that r_w is an odd ranking if and only if $w \notin L_\omega(B)$.

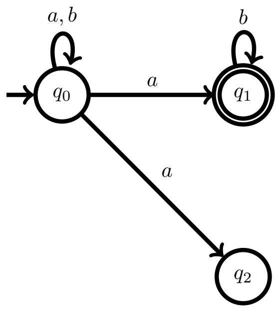
(d) Construct a Büchi automaton accepting $\overline{L_\omega(B)}$ using the construction seen in class. *Hint*: by (c), it is sufficient to use $\{0, 1\}$ as ranks.

Exercise 13.3
Convert the following NBAs into DMAs using Safra's translation.

1. Consider

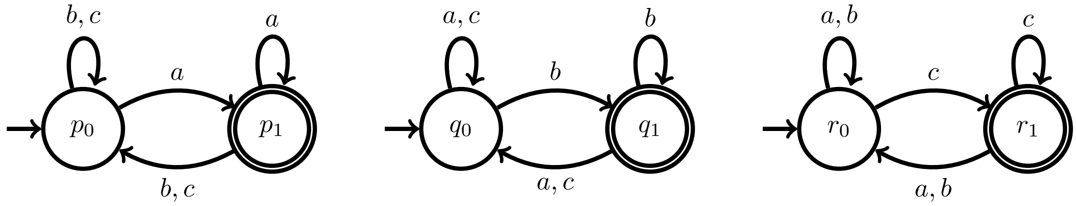


2. Consider

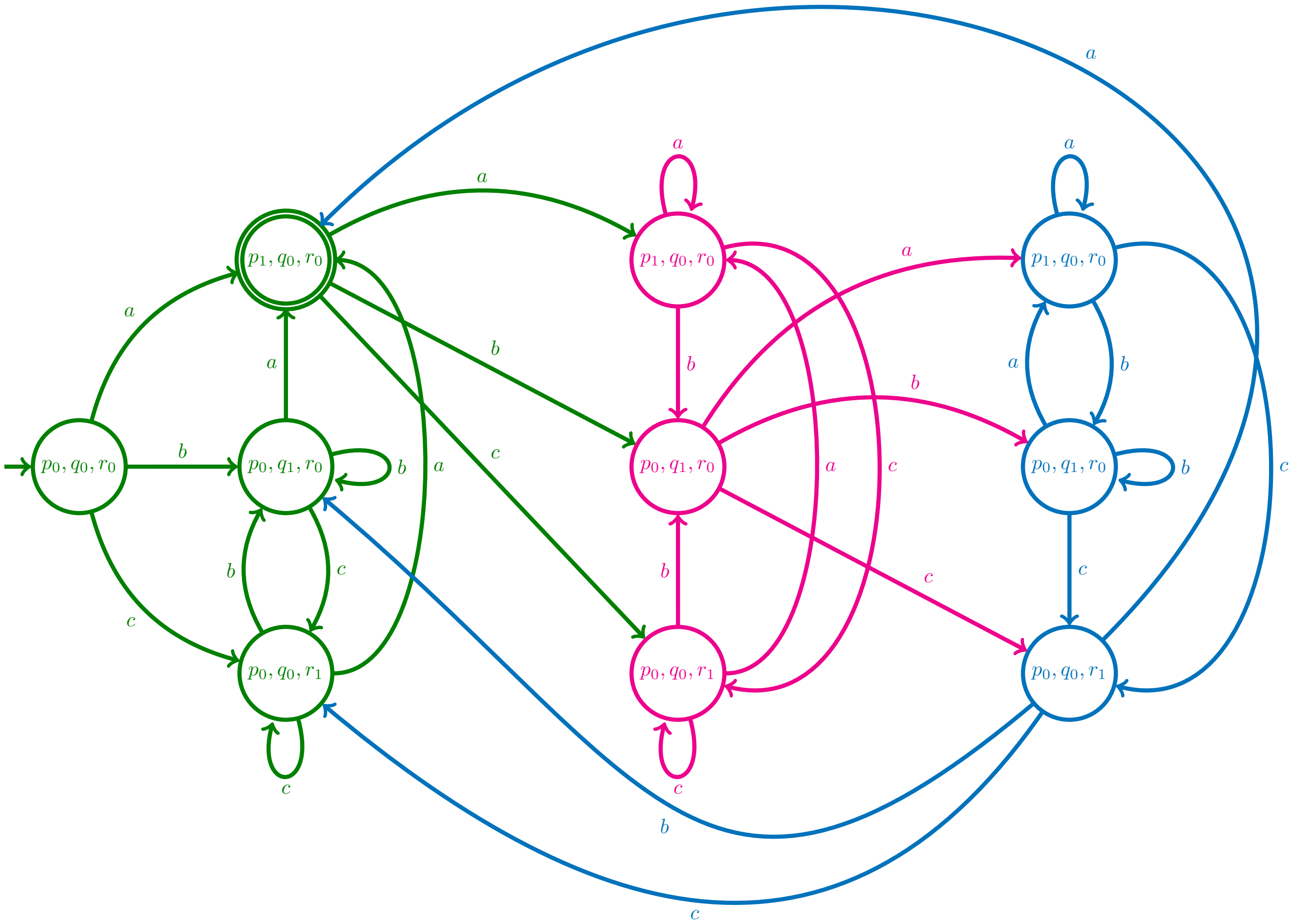


Solution 13.1

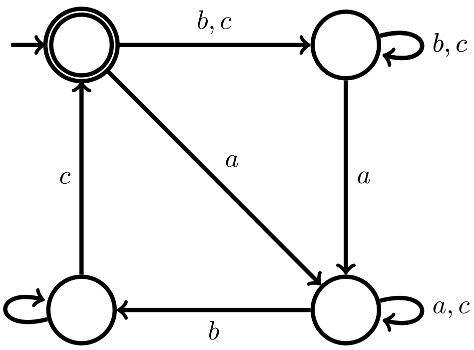
(a) The following deterministic Büchi automata respectively accept L_a, L_b and L_c:



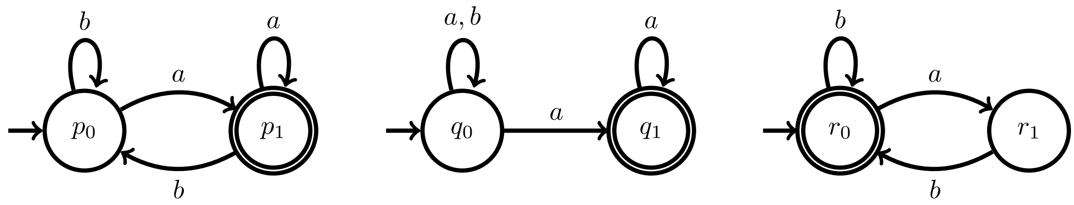
Taking the intersection of these automata leads to the following deterministic Büchi automaton:



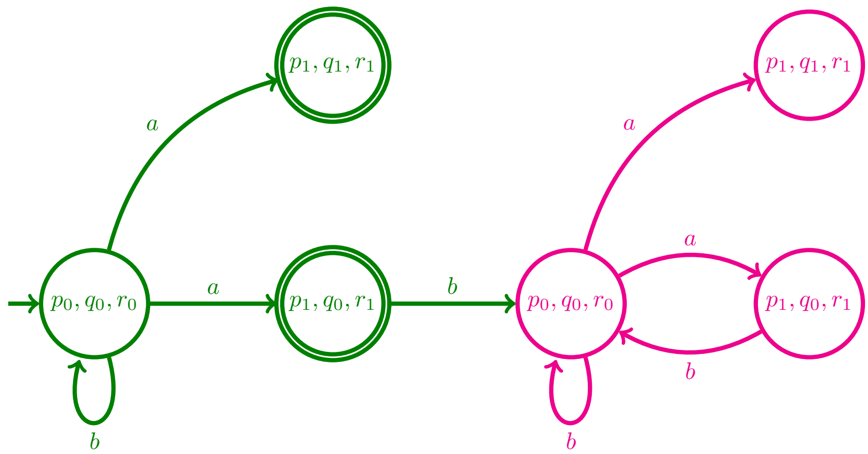
★ As seen in #11.1(d), $L_a \cap L_b \cap L_b$ is accepted by a smaller deterministic Büchi automaton:



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



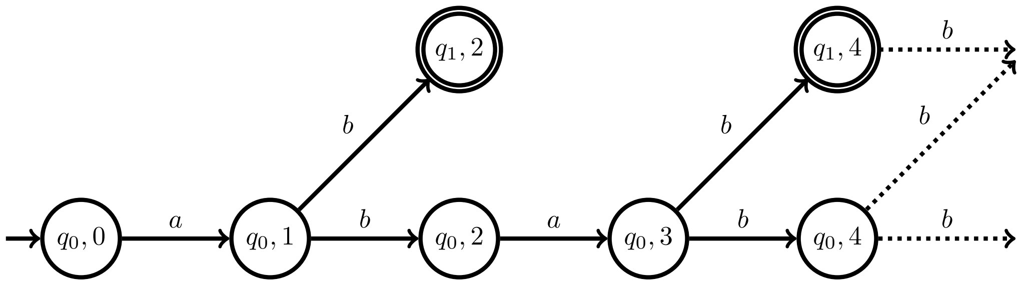
Taking the intersection of these automata leads to the following Büchi automaton:



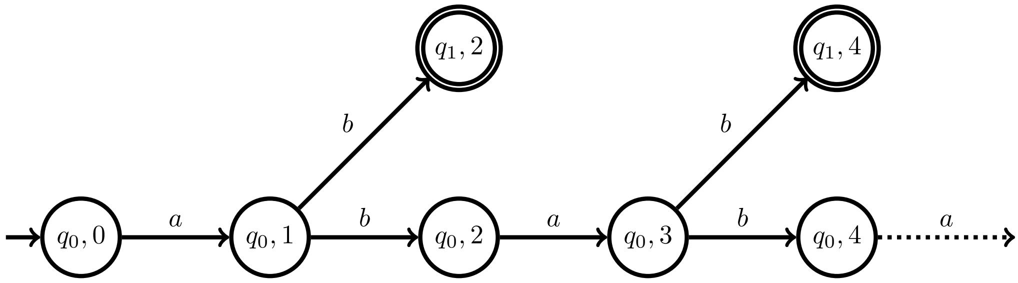
★ Note that the language of this automaton is the empty language.

Solution 13.2

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



dag($(ab)^\omega$):



(b)
- r *is not* an odd rank for 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 dag($abab^\omega$) not visiting odd nodes infinitely often.

- r *is* an odd rank for 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) $\Rightarrow$) Let $w \in L_\omega(B)$. We have $w = ub^\omega$ for some $u \in \{a, b\}^*$. This implies that

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

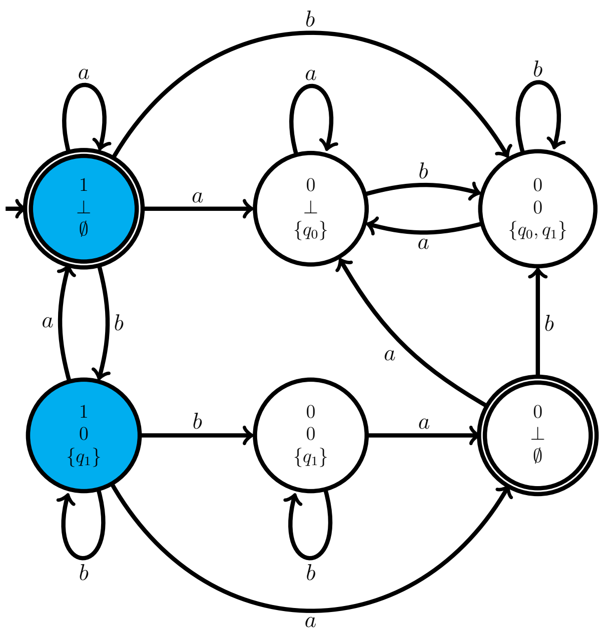
is an infinite path of dag(w). Since this path does not visit odd nodes infinitely often, r is not odd for dag(w).

$\Leftarrow$) Let $w \notin L_\omega(B)$. Suppose there exists an infinite path of 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.

(d) Recall that we construct an NBA with an infinite number of states whose runs on an ω-word w are the rankings of $dag(w)$. The automaton accepts a ranking R iff every infinite path of R visits nodes of odd rank i.o. By (c), for every $w \in \{a, b\}^\omega$, if 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 (and our NBA is now finite). We obtain the following Büchi automaton, for which some intuition is given below:



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.

★ It would have even been sufficient to only explore the blue states as they correspond to the family of rankings $\{r_w : w \in \Sigma^\omega\}$.

Solution 13.3

The Safra determinization procedure converts an NBA A to a DMA B recognizing the same language. It relies on the idea of breakpoints. Consider a run in the automata of the classical subset construction (from NFA to DFA):

$$\begin{array}{ccccccc} \{q_0\} \xrightarrow{u_1} & Q_1 & \xrightarrow{v_1} R_1 & \xrightarrow{u_2} \ldots \xrightarrow{u_i} & Q_i & \xrightarrow{v_i} R_i \\ & \supseteq & = & & \supseteq & = \\ & F_1 & \xrightarrow{v_1} G_1 & & F_i & \xrightarrow{v_i} G_i \end{array}$$

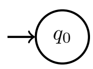
The F_i are the subset of final states of Q_i, and the u_i, v_i are words. We call the moment $R_i = G_i$ a *breakpoint.* If a run over ω-word w visits breakpoints infinitely often then there is a run in the classical subset automata where w visits final states infinitely often. We want to identify these breakpoints, which will be the final states of our DMA. To do so we take as states of B trees whose nodes are sets of states.

Let A be the NBA illustrated in 1., and B the DMA we want to build.

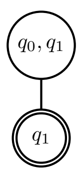
From a tree-state S (whose nodes are sets of states), we get the next tree-state by applying the four steps

1. apply letter a to all nodes of the tree-state S
2. for each node that contains final states, create a child node containing those final states
3. horizontal-merge: if the states of a node n are contained in the node of an older sibling node, delete this node n
4. vertical-merge: if the union of all the children of a node n are equal to that node n, then delete the children and add n to the list of marked nodes. The marked nodes help identify the breakpoints of our automata.

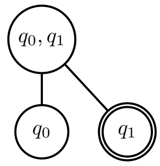
The initial state of B is the 1 node tree



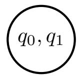
We apply the steps (1 and 2) to our initial tree-state and obtain our second state of B



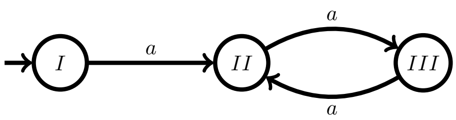
We apply the steps to this new tree, first (1 and 2)



then step (4) which marks $\{q_0, q_1\}$ and obtain the third state of B



Applying the steps to this third state results in the second state again. We now draw the resulting DMA automaton B:



The final sets $F_1, F_2, ..., F_k$ are defined such that each marked node defines one such F_i. There is only one marked node in this case, $\{q_0, q_1\}$, and the final set F_1 is the tree-states that contain $\{q_0, q_1\}$ so $\{II, III\}$.