Automata and Formal Languages — Homework 12

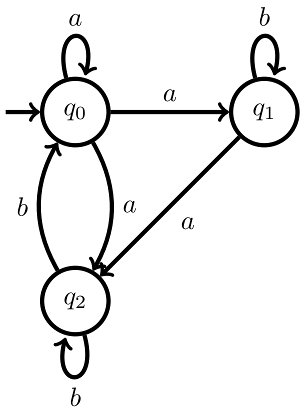
Due 22.01.2019

Exercise 12.1
Implement the Büchi complementation construction seen in the class using `Owl`[1].

The exercise template can be found in the branch `buchi-complementation`[2] and is called `Complementation.java`. Fill in the three locations marked with `// CODE HERE` computing the initial state, the transitions, and the accepting states.

You can test the correctness of your implementation by executing `./gradlew test` in the top-level directory of the project. Note that `Java 11` is the required minimal version for building and running the code.

Exercise 12.2
Consider the following automaton A:



(a) Interpret A as a Muller automaton with acceptance condition $\{\{q_1\}, \{q_0, q_2\}\}$. Use algorithms *NMAtoNGA* and *NGAtoNBA* from the lecture notes to construct a Büchi automaton that recognizes the same language as A.

(b) Interpret A as a Rabin automaton with acceptance condition $\{\langle\{q_0, q_2\}, \{q_1\}\rangle\}$. Follow the approach presented in class to construct a Büchi automaton that recognizes the same language as A.

Exercise 12.3
(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:

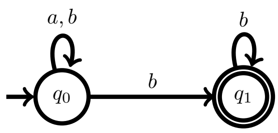
- $L_1 = \{w \in \{a, b\}^\omega : w \text{ contains infinitely many } a\text{'s}\}$,

[1] `https://owl.model.in.tum.de`
[2] `https://gitlab.lrz.de/i7/owl/commits/buchi-complementation`

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

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

Exercise 12.4
Consider the following Büchi automaton over $\Sigma = \{a, b\}$:



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

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

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

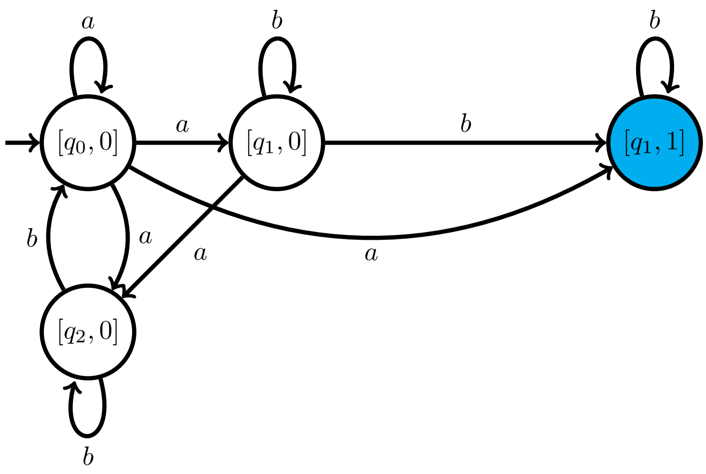
Are r_{abab^ω} and $r_{(ab)^\omega}$ odd rankings?

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

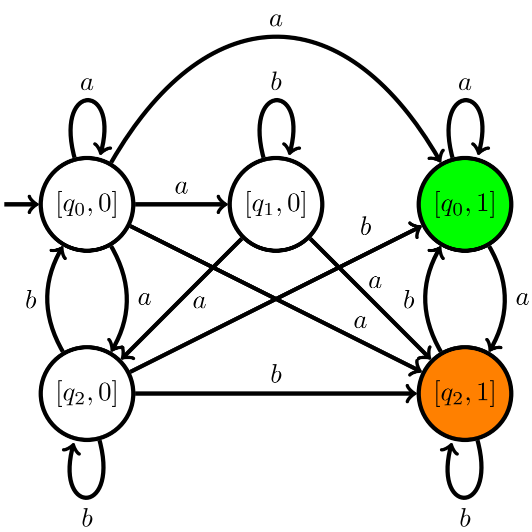
(d) Construct a Büchi automaton accepting $\overline{L_\omega(B)}$ using the construction seen in class. [Hint:]

Solution 12.2

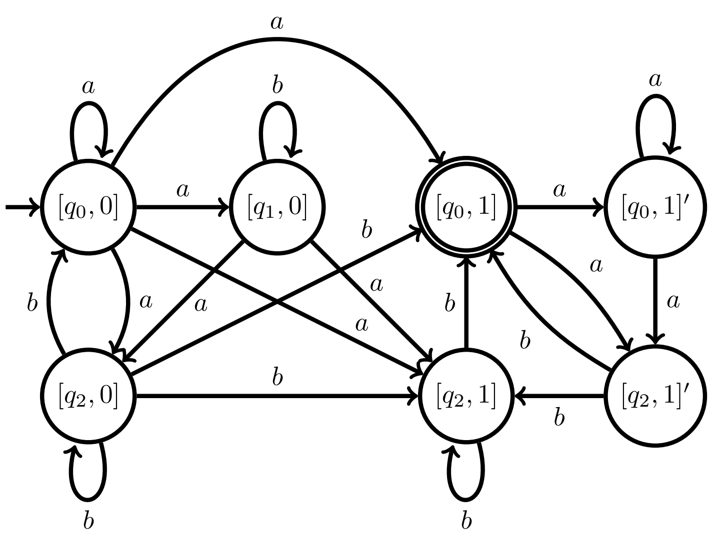
(a) We must first construct two generalized Büchi automata A and B for $\{q_1\}$ and $\{q_0, q_2\}$ respectively. Automaton A is as follows with acceptance condition $\{\{q_1\}\}$:



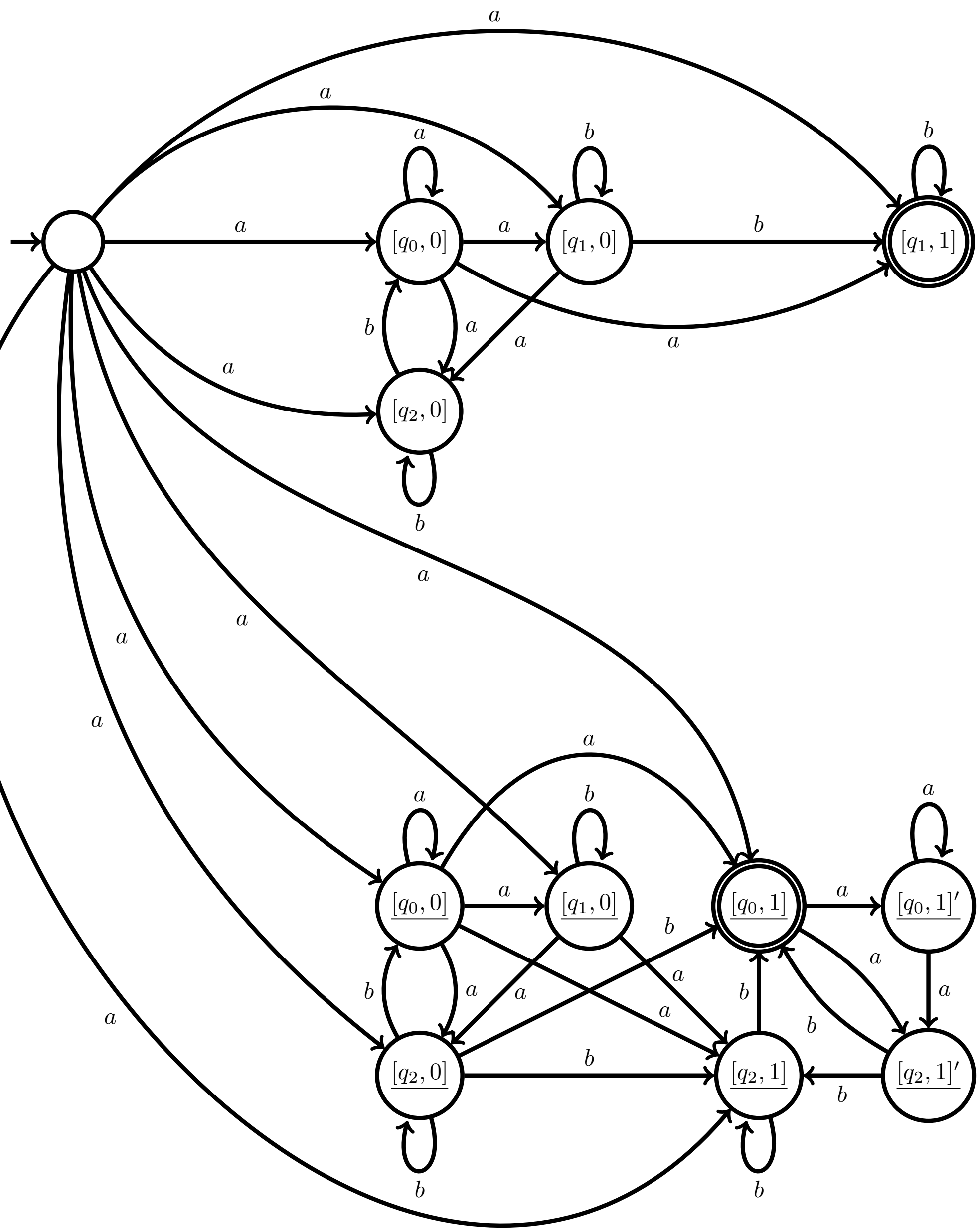
Automaton B is as follows with acceptance condition $\{\{q_0\}, \{q_2\}\}$:



The resulting generalized Büchi automaton is the union of A and B. Note that A is essentially already a standard Büchi automaton, it suffices to make state $[q_1, 1]$ accepting. However, it remains to convert B into a standard Büchi automaton B':

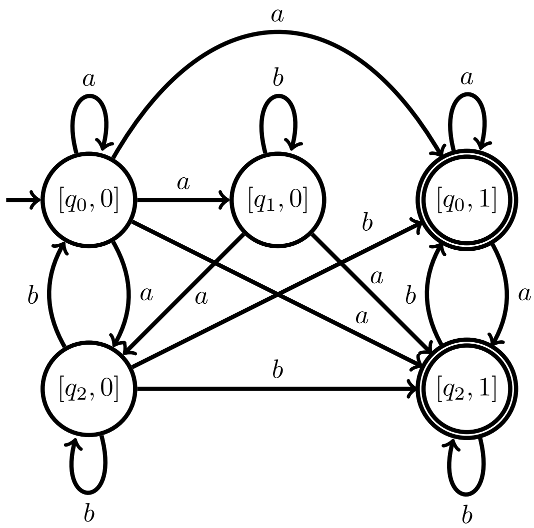


Altogether, we obtain the following Büchi automaton:



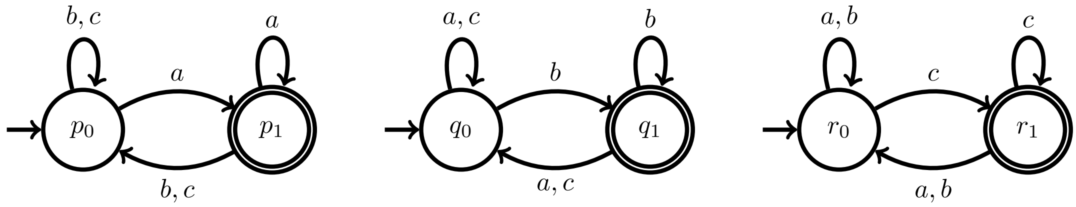
★ Since Büchi automata can have multiple initial states, we can also simply take the disjoint union of both automata, i.e. have them side by side instead of adding a single new initial.

(b)

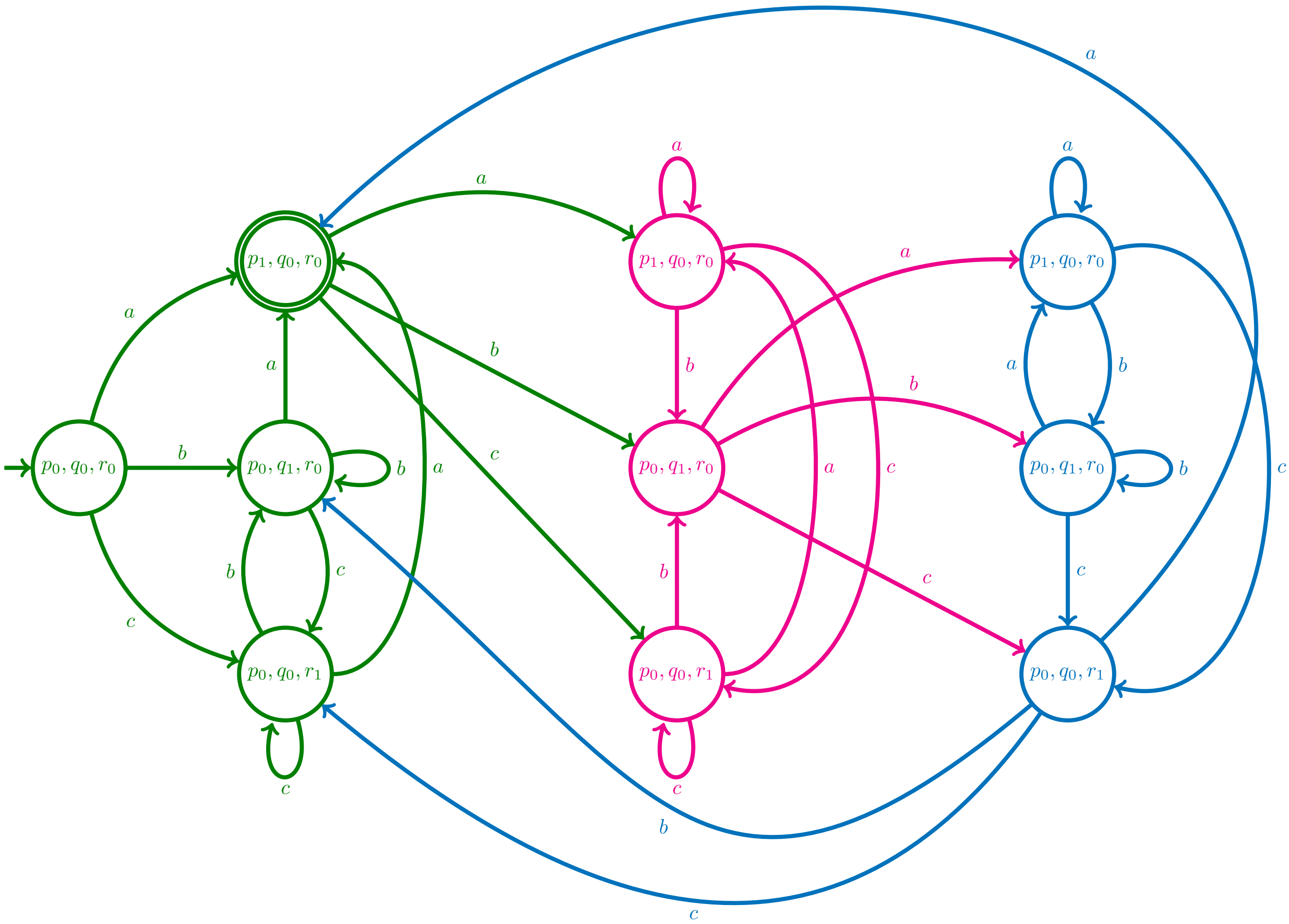


Solution 12.3

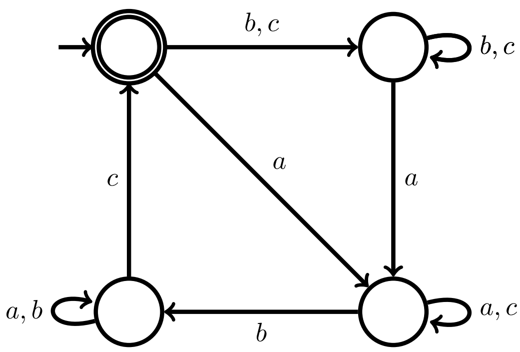
(a) The following deterministic Büchi automata respectively accept L_a, L_b and L_c:



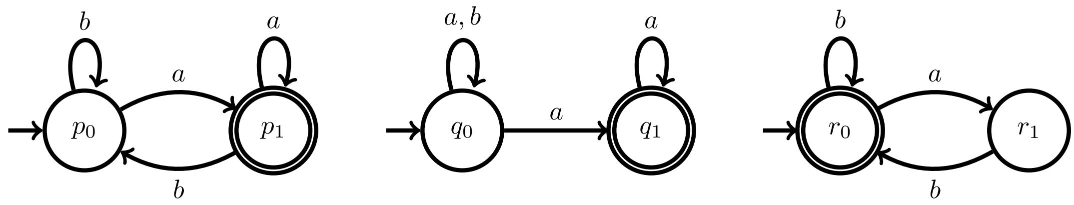
Taking the intersection of these automata leads to the following deterministic Büchi automaton:



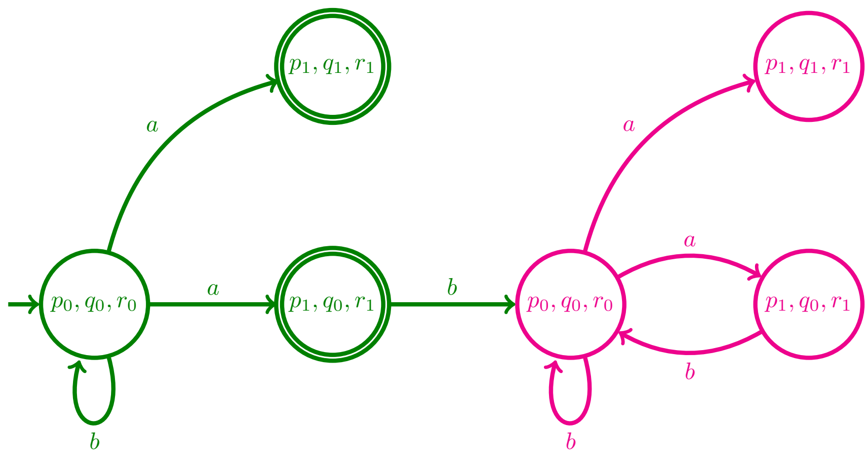
★ As seen in #11.1(d), $L_a \cap L_b \cap L_b$ is accepted by a smaller deterministic Büchi automaton:



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



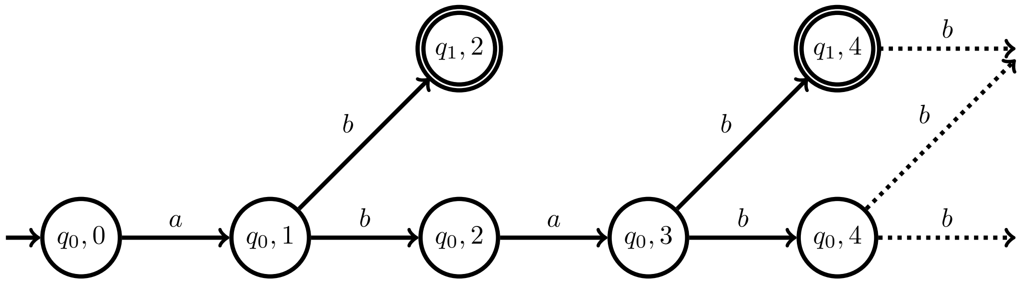
Taking the intersection of these automata leads to the following Büchi automaton:



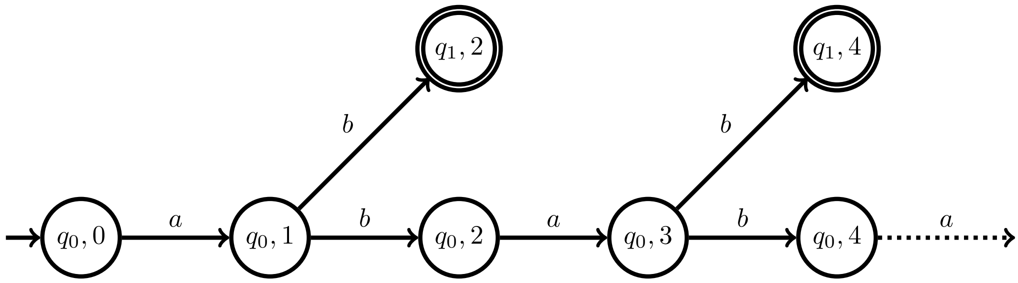
★ Note that the language of this automaton is the empty language.

Solution 12.4

(a) 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

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

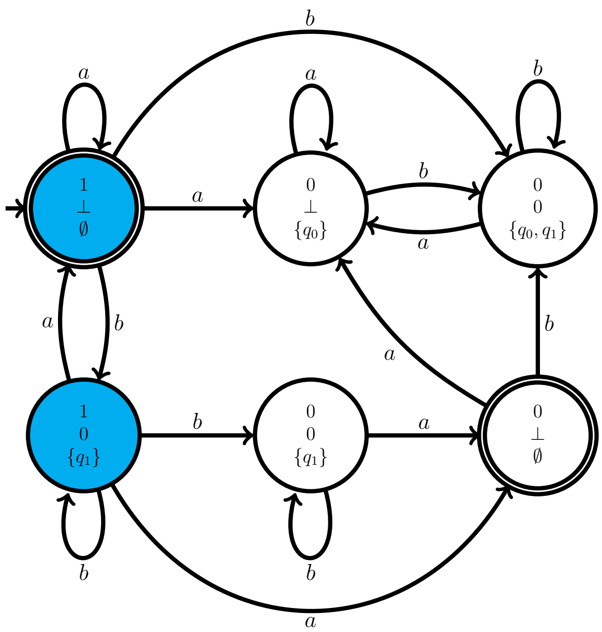
is an infinite path of dag(w). Since this path does not visit odd nodes infinitely often, r is not odd for dag(w).

$\Leftarrow$) 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) 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. We obtain the following Büchi automaton:



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