ω-Automata

ω-Automata

- Automata that accept (or reject) words of infinite length.
- Languages of infinite words appear:
  - in verification, as encodings of non-terminating executions of a program.
  - in arithmetic, as encodings of sets of real numbers.

ω-Languages

- An ω-word is an infinite sequence of letters.
- The set of all ω-words is denoted by Σ^ω.
- An ω-language is a set of ω-words, i.e., a subset of Σ^ω.
- A language L_1 can be concatenated with an ω-language L_2 to yield the ω-language $L_1 L_2$, but two ω-languages cannot be concatenated.
- The ω-iteration of a language $L \subseteq \Sigma^\star$, denoted by L^ω, is an ω-language.
- Observe: $\emptyset^\omega = \{\epsilon\}^\omega = \emptyset$

ω-Regular Expressions

- ω-regular expressions have syntax

$$s ::= r^{\omega} \mid r s_1 \mid s_1 + s_2$$

where r is an (ordinary) regular expression.

- The ω-language $L_{\omega}(s)$ of an ω-regular expression s is inductively defined by

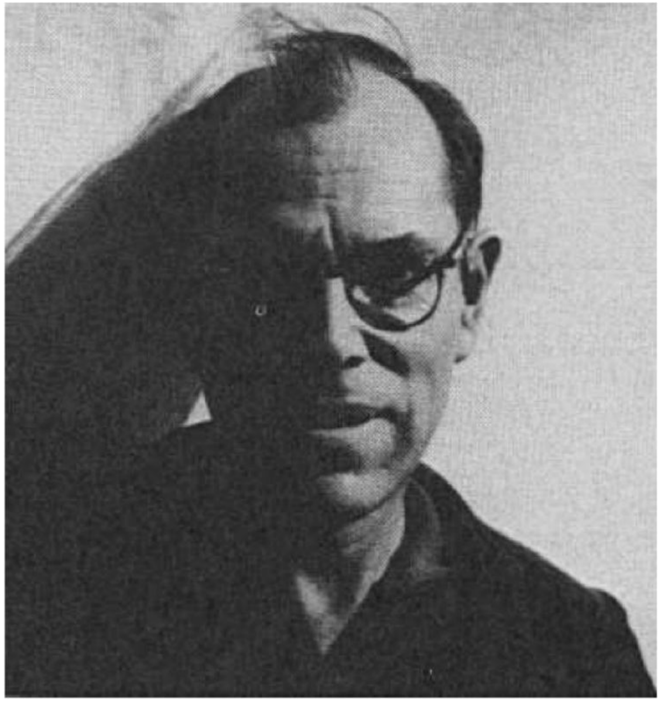
$$L_{\omega}(r^{\omega}) = \big(L(r)\big)^{\omega} \quad L_{\omega}(r s_1) = L(r) L_{\omega}(s_1)$$

$$L_{\omega}(s_1 + s_2) = L_{\omega}(s_1) \cup L_{\omega}(s_2)$$

- A language is ω-regular if it is the language of some ω-regular expression .

Büchi Automata

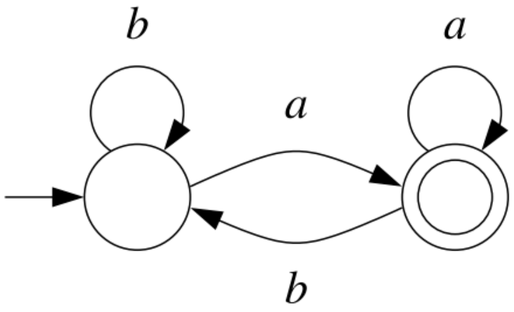
- Invented by J.R. Büchi, swiss logician.

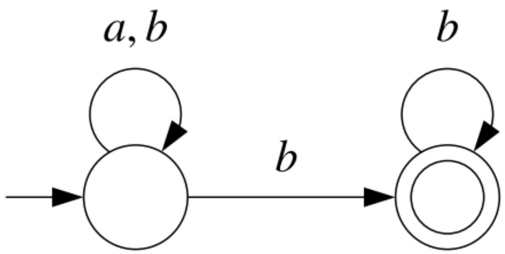


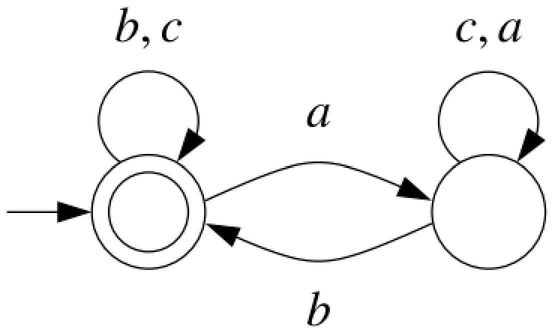
Büchi Automata

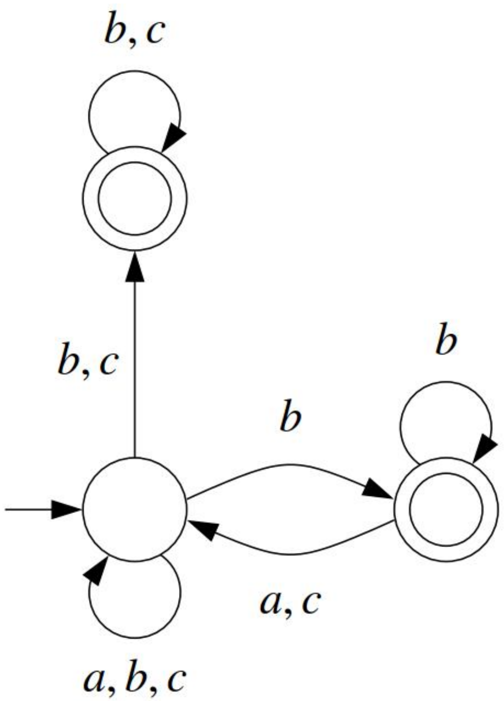
- Same syntax as DFAs and NFAs, but different acceptance condition.
- A run of a Büchi automaton on an ω-word is an infinite sequence of states and transitions.
- A run is accepting if it visits the set of final states infinitely often.
  - Final states renamed to accepting states.
- A DBA or NBA A accepts an ω-word if it has an accepting run on it; the ω-language $L_\omega(A)$ of A is the set of ω-words it accepts.

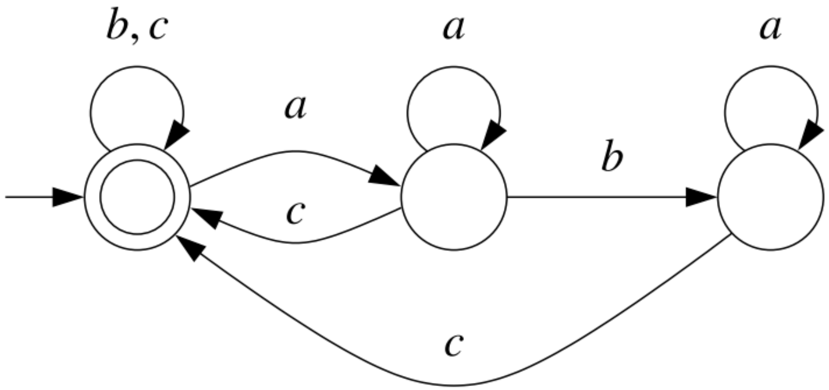
Some examples



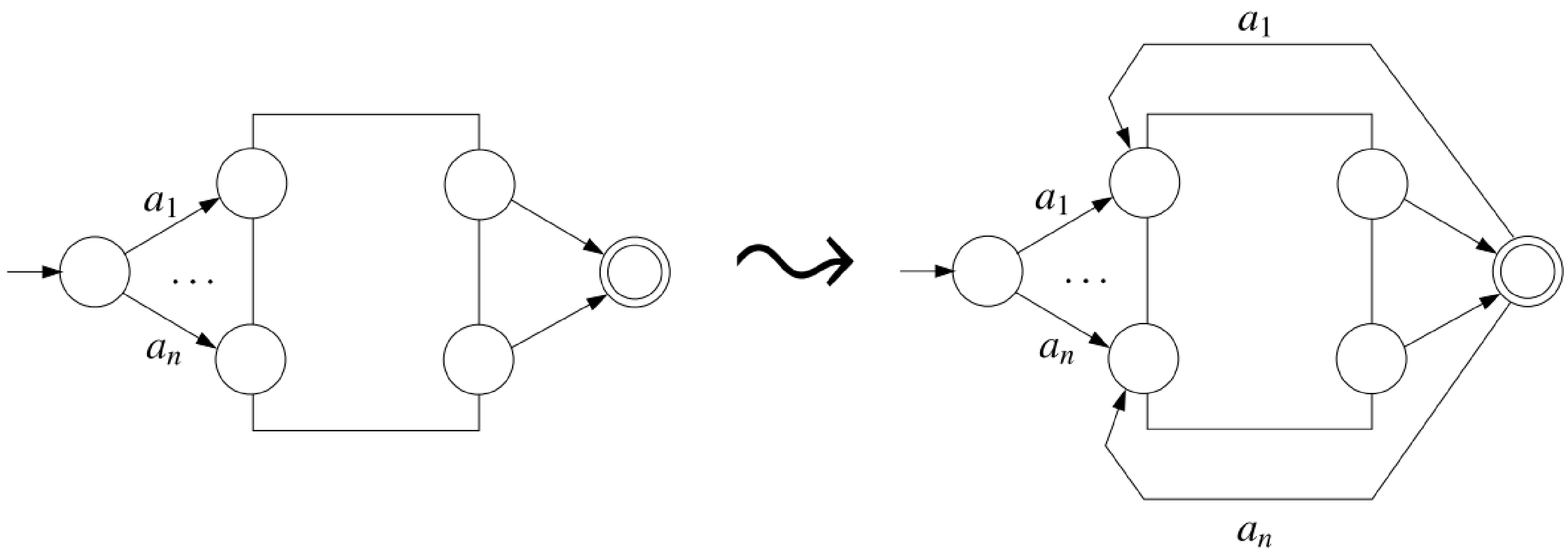






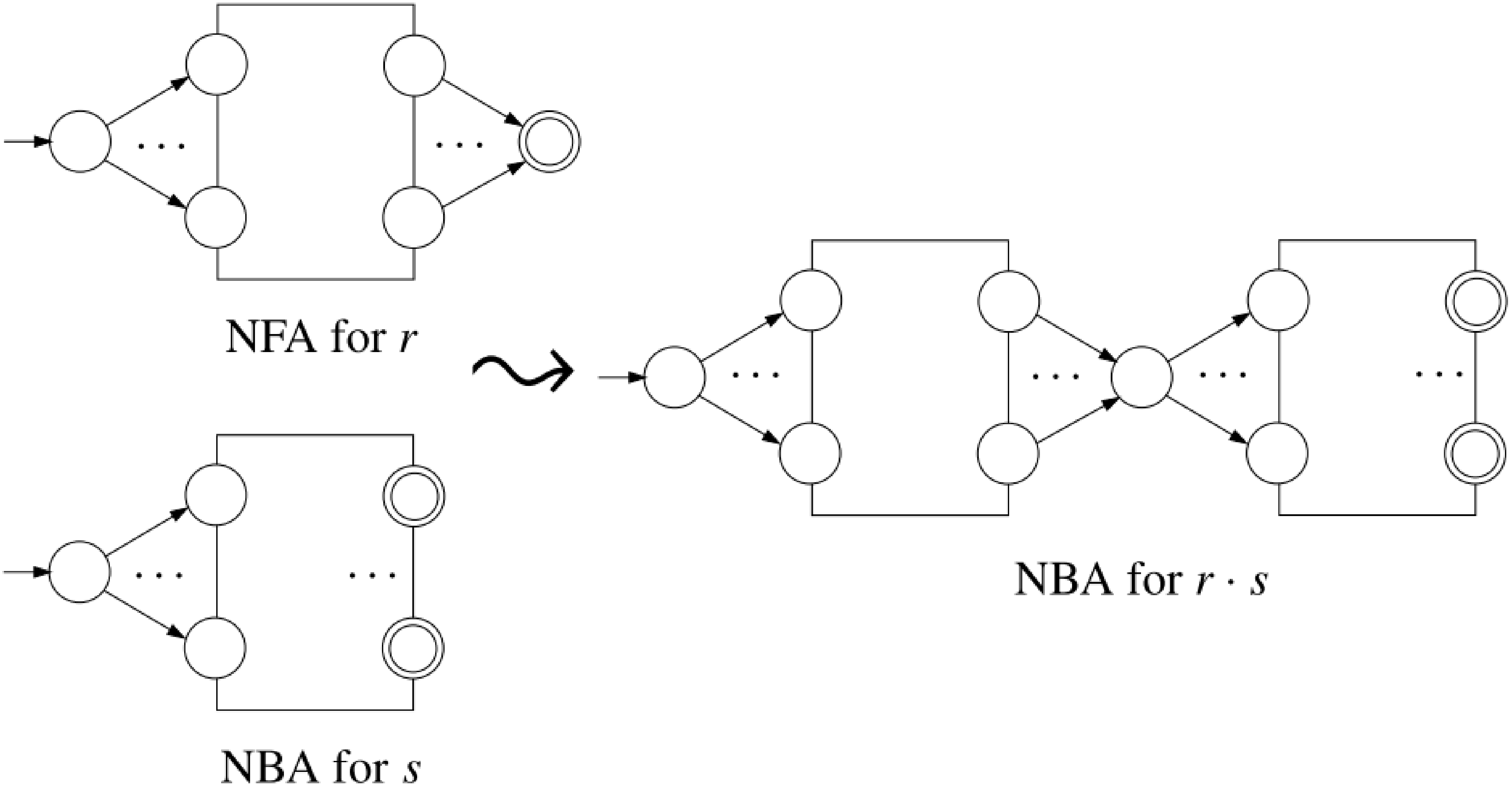


From ω-Regular Expressions to NBAs

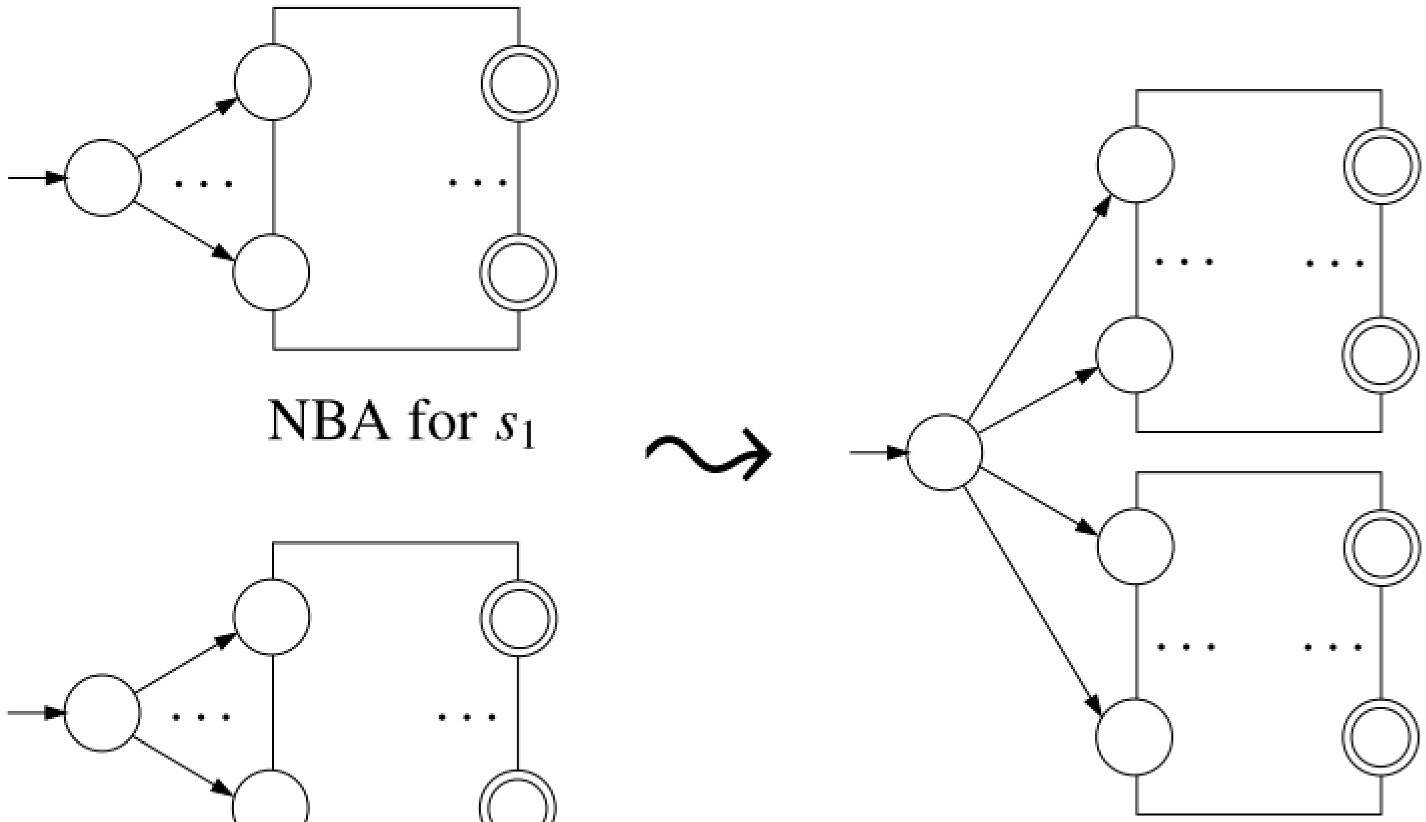


NFA for r NBA for r^ω

From ω-Regular Expressions to NBAs



From ω-Regular Expressions to NBAs



NBA for s_1

NBA for s_2

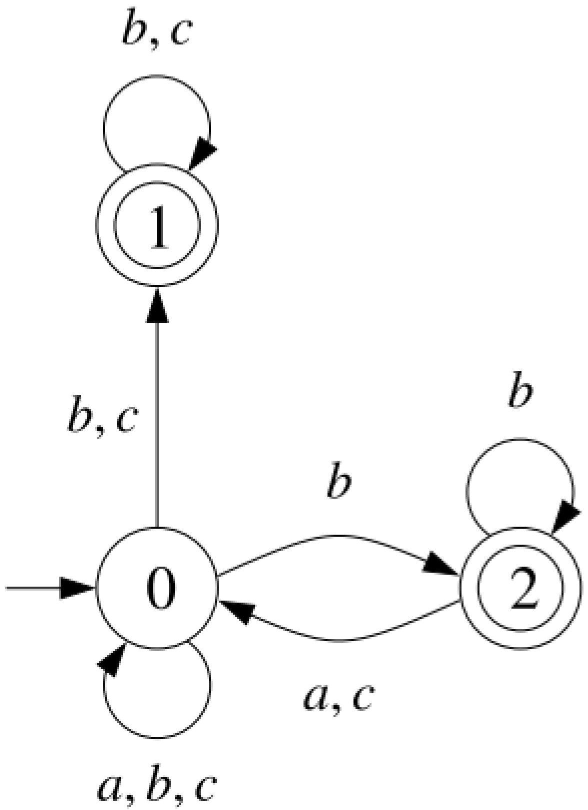
NBA for $s_1 + s_2$

From NBAs to ω-Regular Expressions

- Lemma: Let A be a NFA, and let q, q' be states of A . The language $L_q^{q'}$ of words with runs leading from q to q' and visiting q' exactly once is regular.
- Let $r_q^{q'}$ denote a regular expression for $L_q^{q'}$.

From NBAs to ω-Regular Expressions

- Example:



$$
\begin{aligned}
r_0^1 &= (a+b+c)^*(b+c) \\
r_0^2 &= (a+b+c)^*b \\
r_1^1 &= (b+c) \\
r_2^2 &= b+(a+c)(a+b+c)^*b
\end{aligned}
$$

From NBAs to ω-Regular Expressions

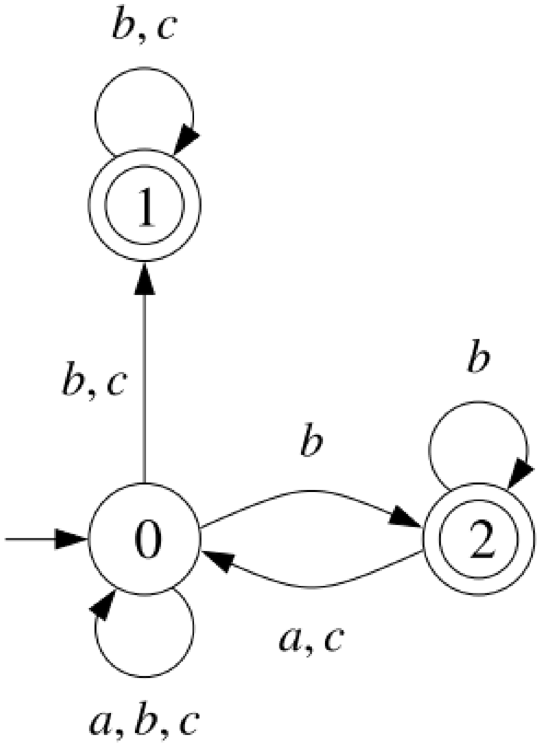
- Given a NBA A , we look at it as a NFA, and compute regular expressions $r_q^{q'}$.
- We show:

$$L_\omega(A) = L\left(\sum_{q\in F} r_{q_0}^{q} \left(r_q^q\right)^\omega\right)$$

  – An ω-word belongs to $L_\omega(A)$ iff it is accepted by a run that starts at q_0 and visits some accepting state q infinitely often.

From NBAs to ω-Regular Expressions

- Example:



$$
\begin{aligned}
r_0^1 &= (a+b+c)^*(b+c) \\
r_0^2 &= (a+b+c)^* b \\
r_1^1 &= (b+c) \\
r_2^2 &= b + (a+c)(a+b+c)^* b
\end{aligned}
$$

$$L_\omega(A) = r_0^1\,(r_1^1)^\omega + r_0^2\,(r_2^2)^\omega$$

DBAs are less expressive than NBAs

- Prop.: The ω-language $(a+b)^* b^\omega$ is not recognized by any DBA.
- Proof: By contradiction. Assume some DBA recognizes $(a+b)^* b^\omega$.
  - DBA accepts b^ω → DFA accepts b^{n_0}

    DBA accepts $b^{n_0} a\, b^\omega$ → DFA accepts $b^{n_0} a\, b^{n_1}$

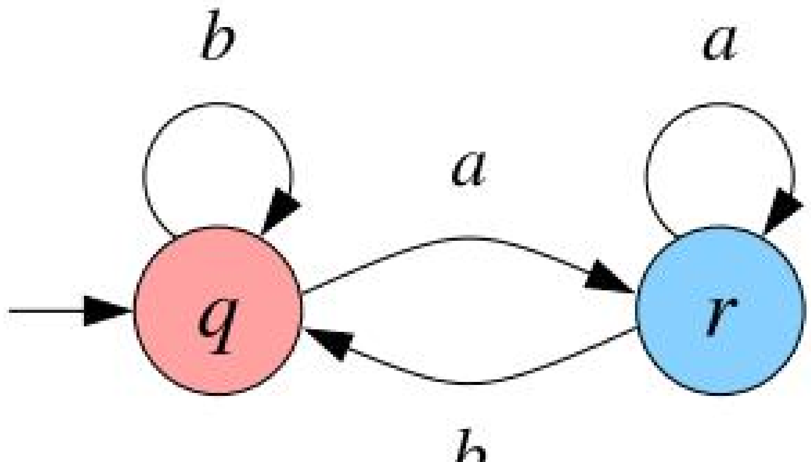
    DBA accepts $b^{n_0} a\, b^{n_1}\, ab^\omega$ → DFA accepts $b^{n_0} a\, b^{n_1} a\, b^{n_2}$ etc.
  - By determinism and finite number of states, the DBA accepts

    $$b^{n_0} a\, b^{n_1} a\, b^{n_2} \ldots a\, b^{n_i} (ab^{n_{i+1}} \ldots ab^{n_j})^\omega$$

    which does not belong to $(a+b)^* b^\omega$.

Generalized Büchi Automata

- Same power as Büchi automata, but more adequate for some constructions.
- Several sets of accepting states.
- A run is accepting if it visits each set of accepting states infinitely often.



$\mathcal{F} = \{ \{q\}, \{r\} \}$

From NGAs to NBAs

- Important fact:

All the sets $F_1, \ldots, F_n$ are visited infinitely often

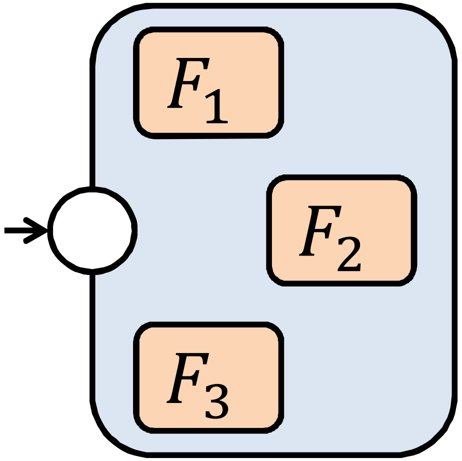
is equivalent to

F_1 is eventually visited

and

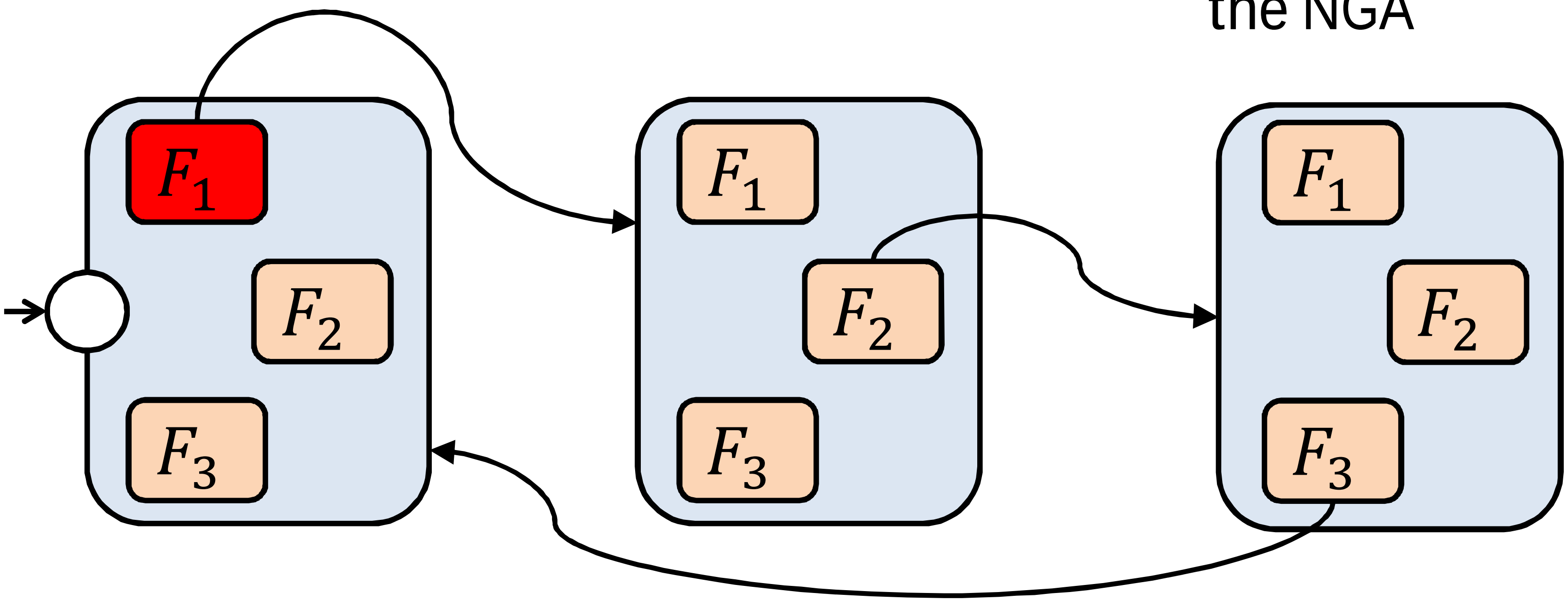
every visit to F_i is eventually followed by a visit to $F_{i \oplus 1}$

From NGAs to NBAs



NGA with 3 sets of accepting states

Equivalent NBA with 3 copies of the NGA

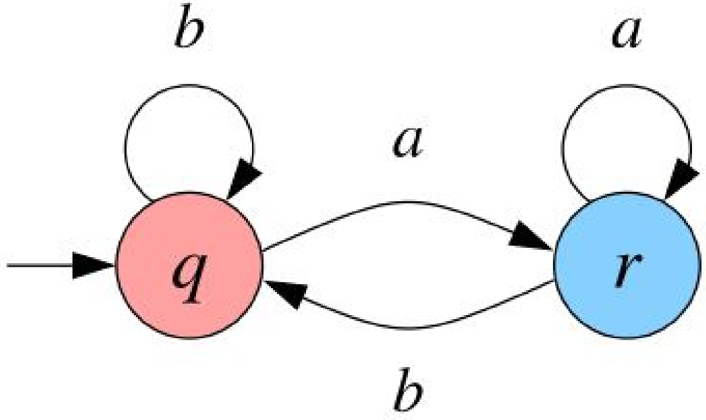


NGAtoNBA(A)

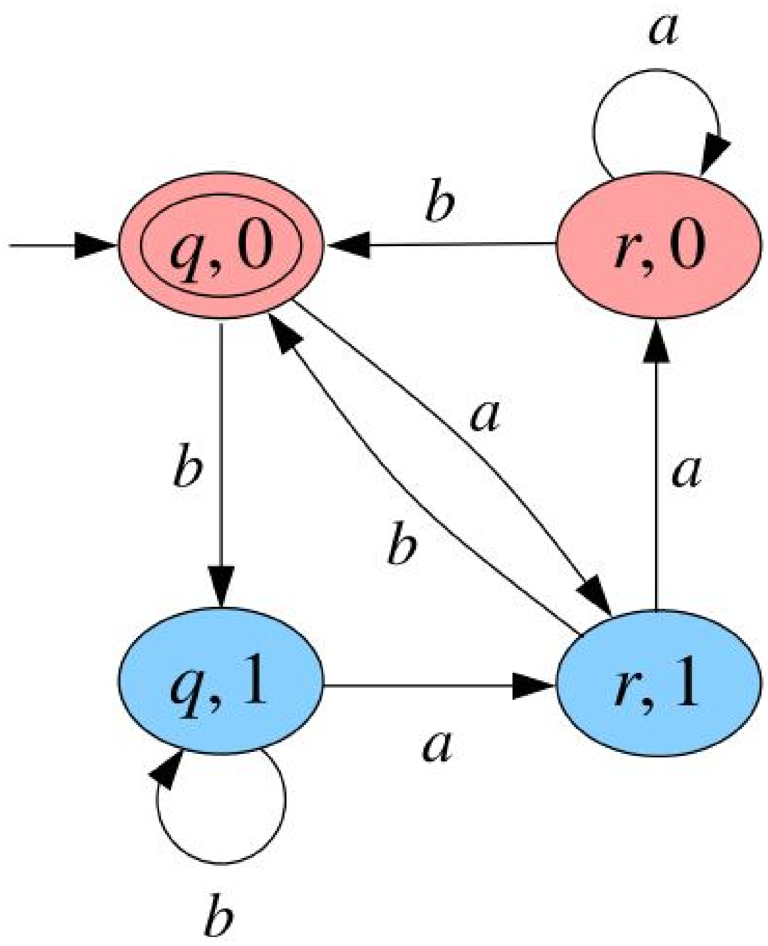
Input: NGA $A = (Q, \Sigma, Q_0, \delta, \mathcal{F})$, where $\mathcal{F} = \{F_0, \ldots, F_{m-1}\}$

Output: NBA $A' = (Q', \Sigma, \delta', Q'_0, F')$

```
Q', δ', F' ← ∅; Q'_0 ← {[q_0, 0] | q_0 ∈ Q_0}
W ← Q'_0
while W ≠ ∅ do
    pick [q, i] from W
    add [q, i] to Q'
    if q ∈ F_0 and i = 0 then add [q, i] to F'
    for all a ∈ Σ, q' ∈ δ(q, a) do
        if q ∉ F_i then
            if [q', i] ∉ Q' then add [q', i] to W
            add ([q, i], a, [q', i]) to δ'
        else /* q ∈ F_i */
            if [q', i ⊕ 1] ∉ Q' then add [q', i ⊕ 1] to W
            add ([q, i], a, [q', i ⊕ 1]) to δ'
return (Q', Σ, δ', Q'_0, F')
```



$\mathcal{F} = \{\ \{q\}, \{r\}\ \}$



DGAs have the same expressive power as DBAs, and so are not equivalent to NGAs.

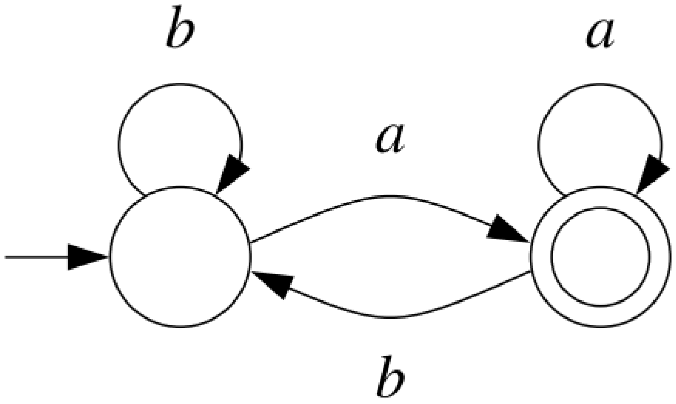
- Question: Are there other classes of omega-automata with
  - the same expressive power as NBAs or NGAs, and
  - with equivalent deterministic and nondeterministic versions?

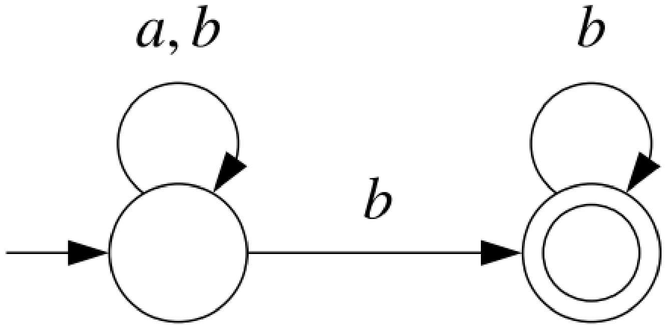
We are only willing to change the acceptance condition!

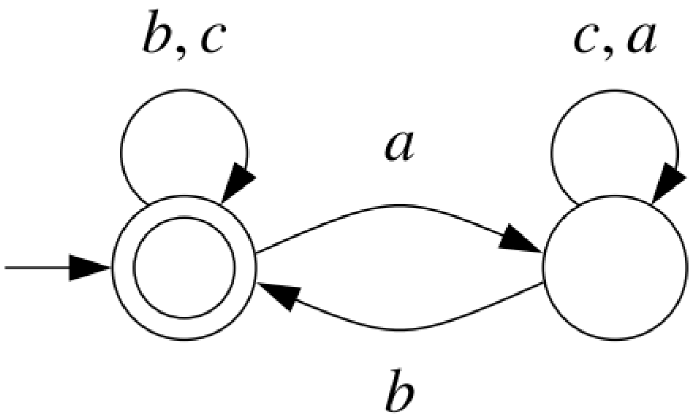
Co-Büchi automata

- A nondeterministic co-Büchi automaton (NCA) is syntactically identical to a NBA, but a run is accepting iff it only visits accepting states finitely often.
- Fact: Given an automaton A, let B the result of swapping accepting and non-accepting states. If A as NBA recognizes a language L, then B as NCA recognizes $\overline{L}$.

Which are the languages?



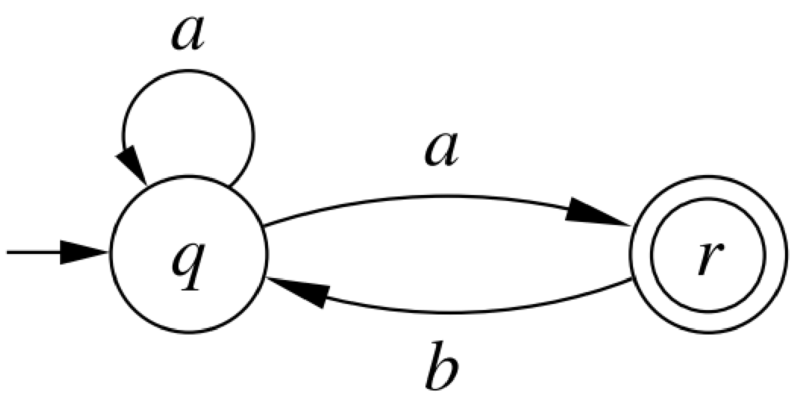


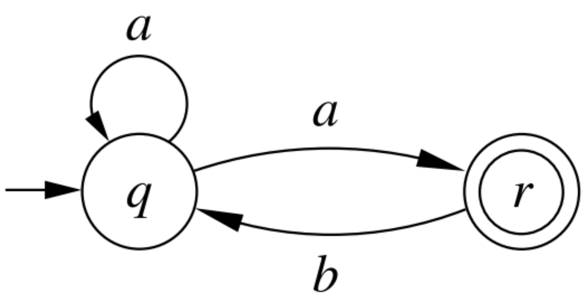


Determinizing co-Büchi automata

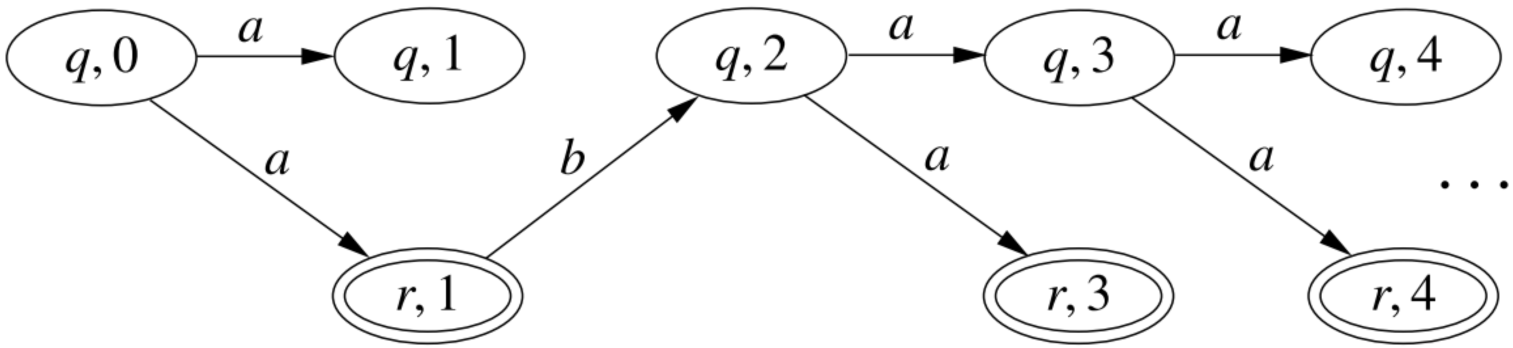
- Given a NCA A we construct a DCA B such that $L(A) = L(B)$.
- We proceed in three steps:
  - We assign to every ω-word w a directed acyclic graph $dag(w)$ that ``contains´´ all runs of A on w.
  - We prove that w is accepted by A iff $dag(w)$ is infinite but contains only finitely many breakpoints.
  - We construct a DCA B such that w is accepted by B iff $dag(w)$ is infinite but contains only finitely many breakpoints.

- Running example:

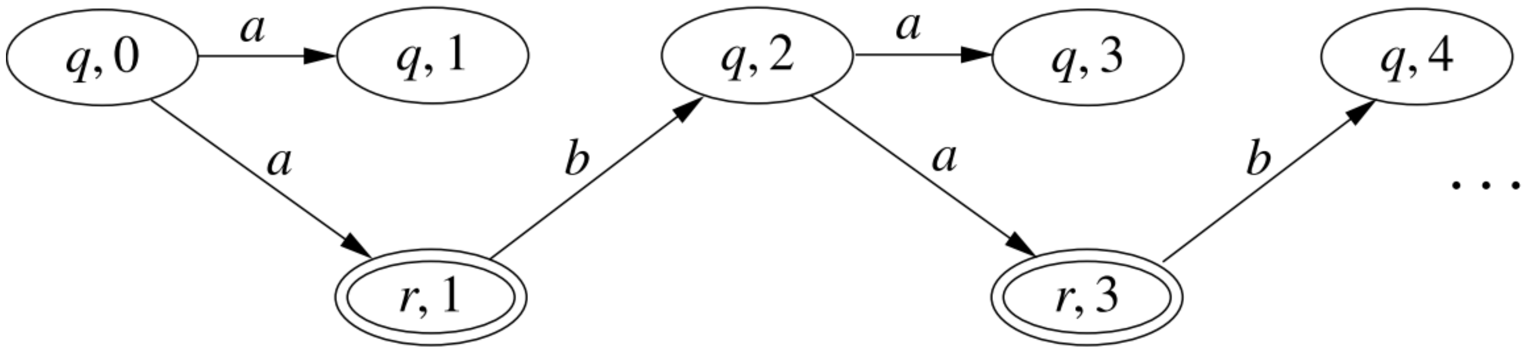




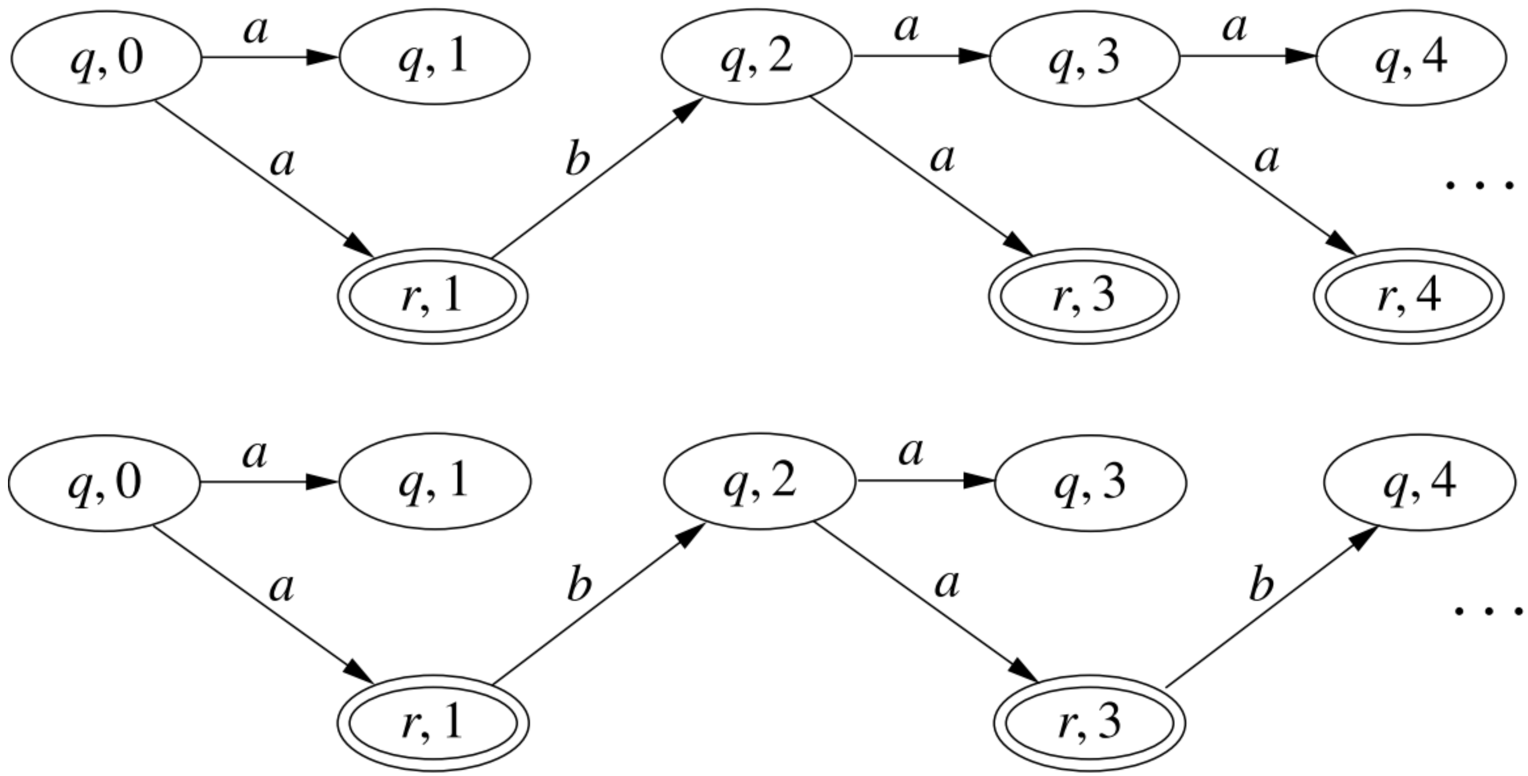
$dag(aba^{\omega})$



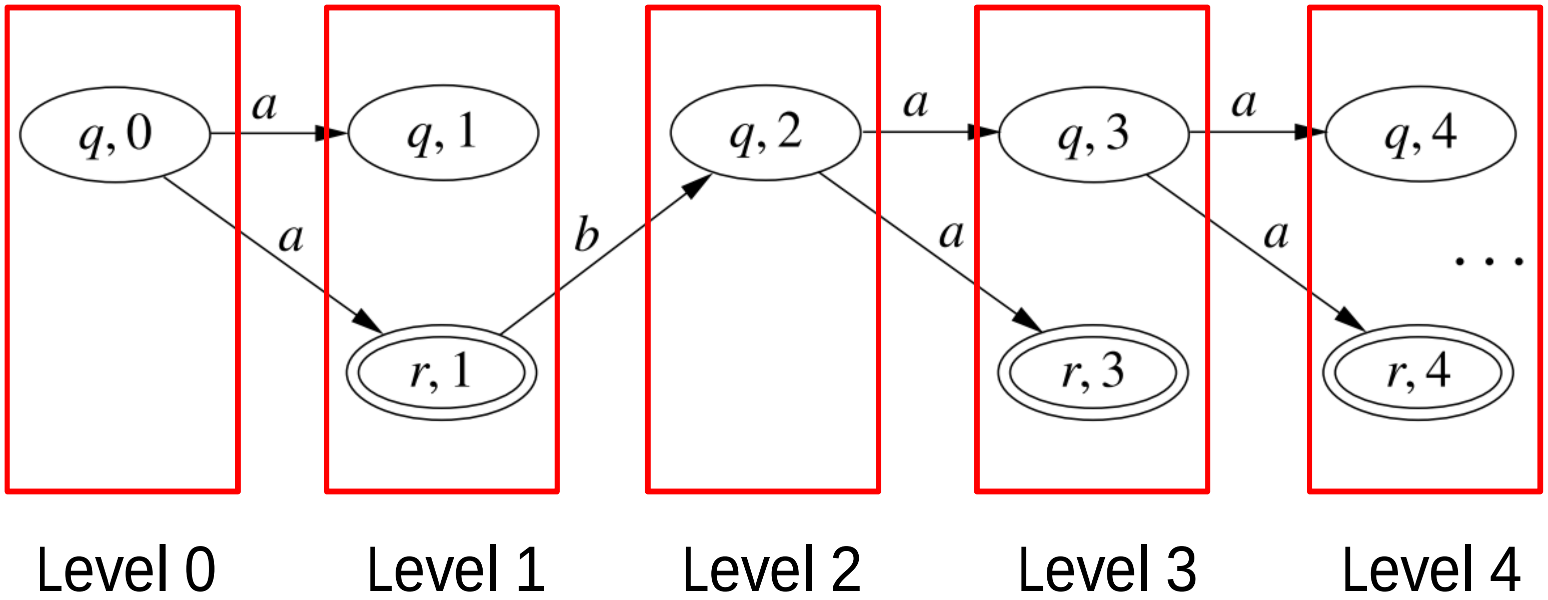
$dag((ab)^{\omega})$



- A accepts w iff some infinite path of $dag(w)$ only visits accepting states finitely often



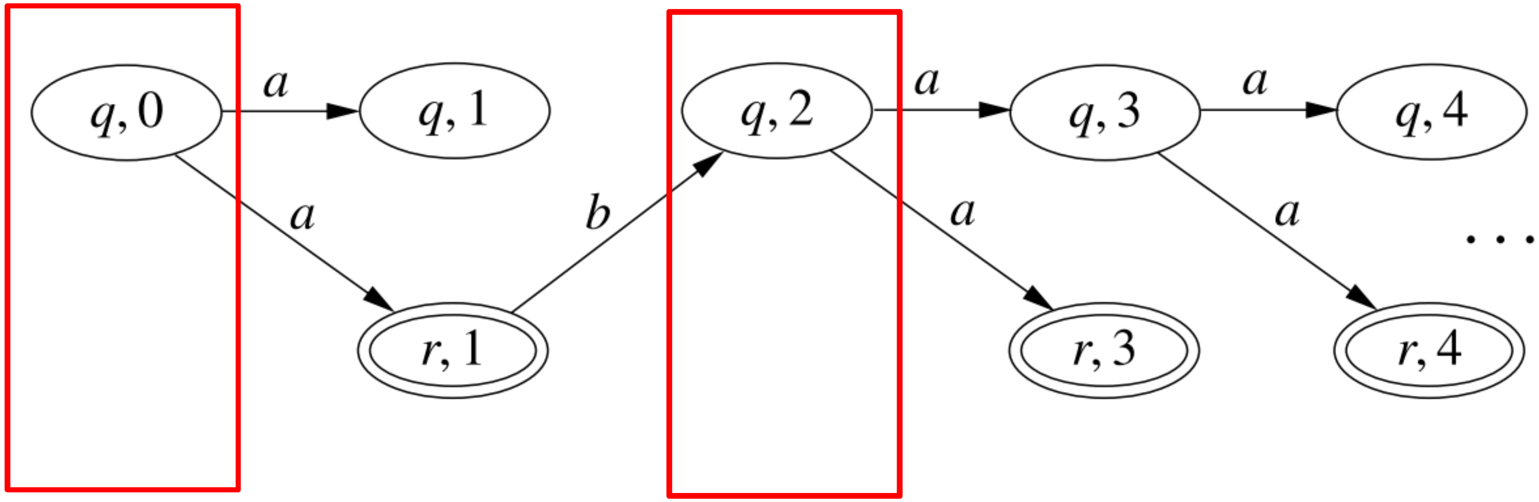
Levels of a dag

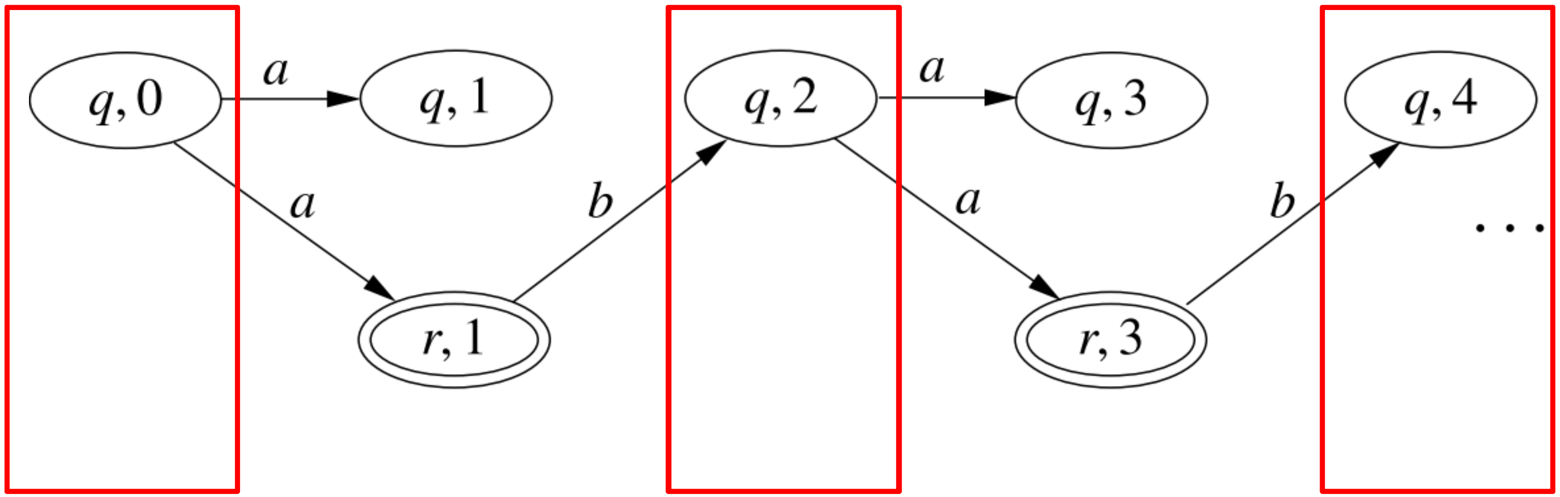


Breakpoints of a dag

- We defined inductively the set of levels that are breakpoints:
  - Level 0 is always a breakpoint
  - If level l is a breakpoint, then the next level l' such that every path from l to l' visits an accepting state is also a breakpoint.

Only two breakpoints





Infinitely many breakpoints

Lemma: A accepts w iff $dag(w)$ is infinite and has only finitely many breakpoints.

Proof:

(⇒) If A accepts w, then it has at least one run on w, and so $dag(w)$ is infinite. Moreover, the run visits accepting states only finitely often, and so after it stops visiting accepting states there are no further breakpoints.

Lemma: A accepts w iff $dag(w)$ is infinite and has only finitely many breakpoints.

Proof:

(⇐) Assume $dag(w)$ is infinite and has only finitely many breakpoints. Let l be the last breakpoint. Since $dag(w)$ is infinite, for every $l' > l$ there is a path from l to l' that visits no accepting states. The subdag containing all these paths is infinite and has finite degree. By König's Lemma the dag contains an infinite path.

Constructing the DCA

- If we could tell if a level is a breakpoint by looking at it, we could take the set of all breakpoints as the set of states of the DCA.
- However, in oder to decide if a level is a breakpoint we need information about its ``history´´.
- Solution: add that information to the level.

Constructing the DCA

- States: pairs $[P, O]$ where:
  - P is the set of states of a level, and
  - $O \subseteq P$ is the set of states ``that owe a visit to the set of accepting states''.
- Formally: $q \in O$ if q is the endpoint of a path starting at the last breakpoint that has not yet visited any accepting state.

Constructing the DCA

- States: pairs $[P, O]$
- Initial state: pair $[Q_0, \emptyset]$.
- Transitions: $\delta([P, Q], a) = [P', O']$ where $P' = \delta(P, a)$, and
  - $O' = \delta(O, a) \setminus F$ if $O \neq \emptyset$
    (automaton updates set of owing states)
  - $O' = \delta(P, a) \setminus F$ if $O = \emptyset$
    (automaton starts search for next breakpoint)
- Accepting states: pairs $[P, \emptyset]$ (no owing states)

NCAtoDCA(A)

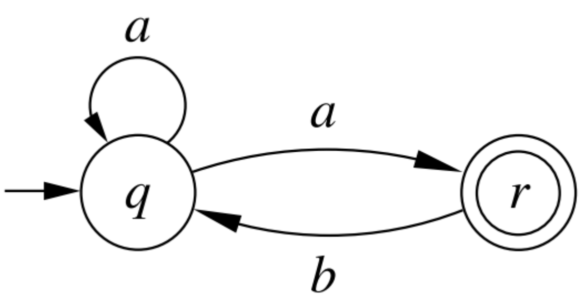
Input: NCA $A = (Q, \Sigma, \delta, Q_0, F)$

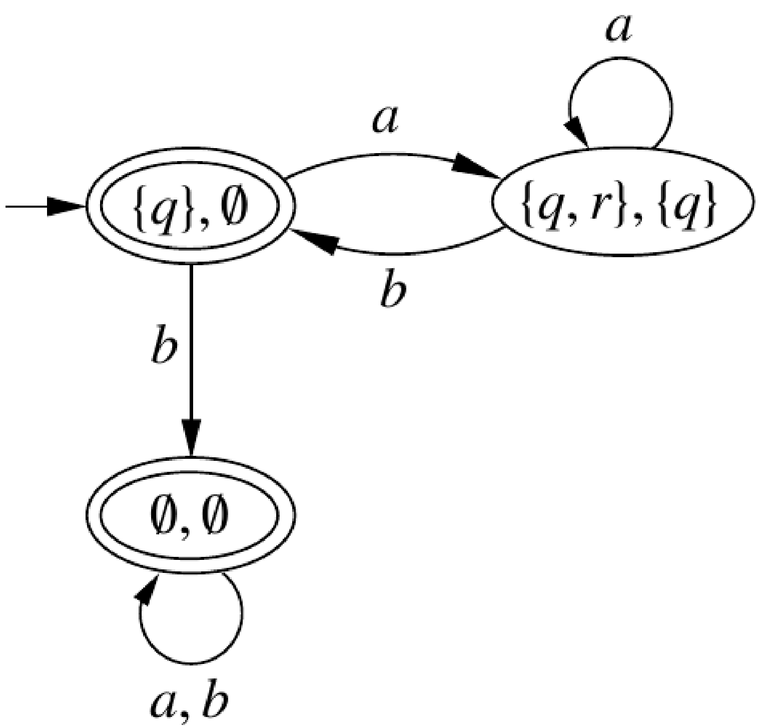
Output: DCA $B = (\tilde{Q}, \Sigma, \tilde{\delta}, \tilde{q}_0, \tilde{F})$ with $L_\omega(A) = L_\omega(B)$

$\tilde{Q}, \tilde{\delta}, \tilde{F} \leftarrow \emptyset$; $\tilde{q}_0 \leftarrow [Q_0, \emptyset]$
$W \leftarrow \{ \tilde{q}_0 \}$
while $W \neq \emptyset$ **do**
    pick $[P, O]$ **from** W; **add** $[P, O]$ **to** $\tilde{Q}$
    if $P = \emptyset$ **then add** $[P, O]$ **to** $\tilde{F}$
    for all $a \in \Sigma$ **do**
        $P' = \delta(P, a)$
        if $O \neq \emptyset$ **then** $O' \leftarrow \delta(O, a) \setminus F$ **else** $O' \leftarrow \delta(P, a) \setminus F$
        add $([P, O], a, [P', O'])$ **to** $\tilde{\delta}$
        if $[P', O'] \notin \tilde{Q}$ **then add** $[P', Q']$ **to** W

- Complexity: at most 3^n states

Running example





Recall ...

- Question: Are there other classes of omega-automata with
  - the same expressive power as NBAs or NGAs, and
  - with equivalent deterministic and nondeterministic versions?

Are co-Büchi automata a positive answer?

Unfortunately no ...

Lemma: No DCA recognizes the language $(b^* a)^\omega$.

Proof: Assume the contrary. Then the same automaton seen as a DBA recognizes the complement $(a + b)^* b^\omega$. Contradiction.

So the quest goes on ...

Muller automata

- A nondeterministic Muller automaton (NMA) has a collection $\{F_0, F_1, \ldots, F_{m-1}\}$ of sets of accepting states.
- A run is accepting if the set of states it visits infinitely often is equal to one of the sets in the collection.

From Büchi to Muller automata

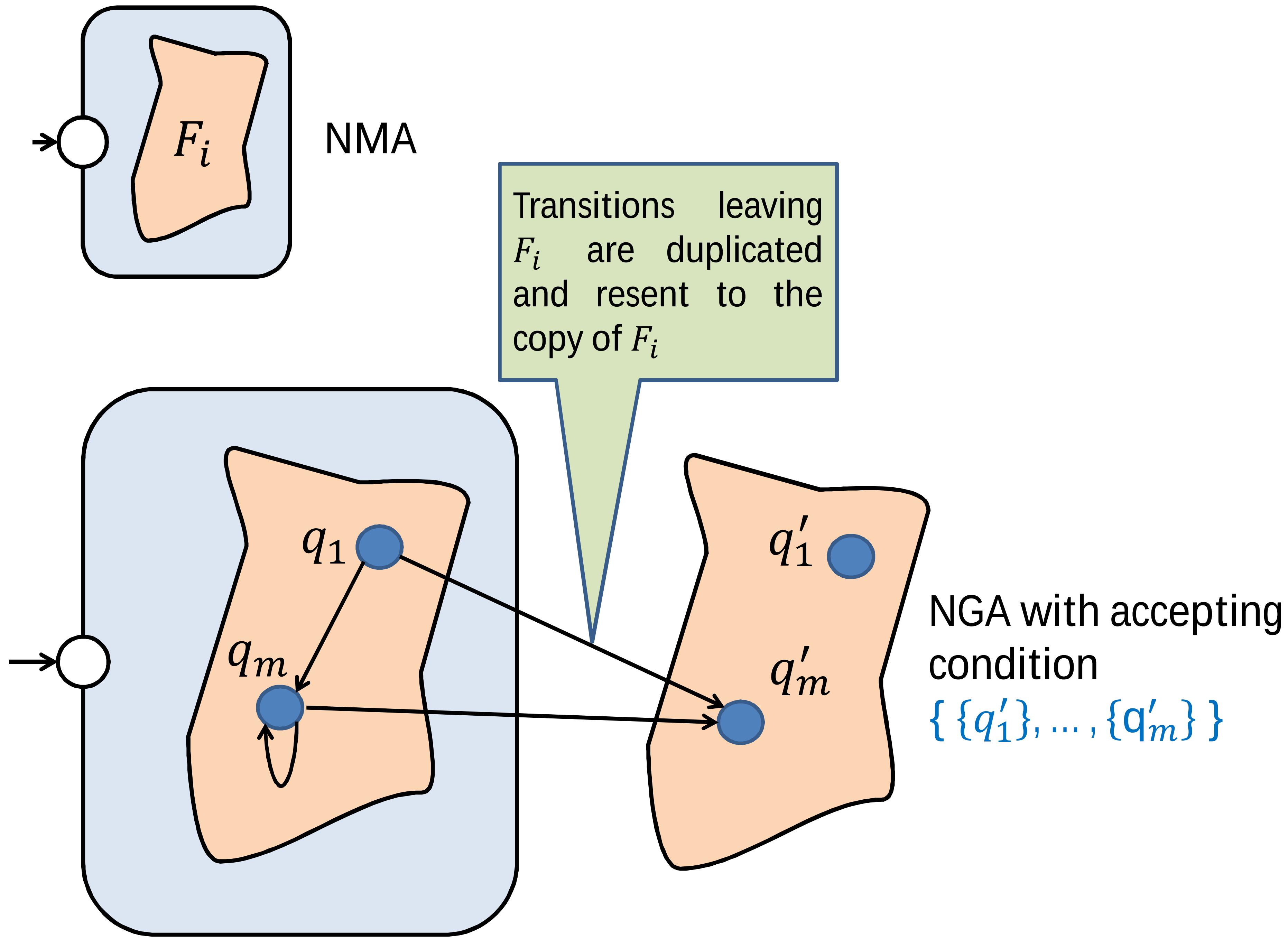
- Let A be a NBA with set F of accepting states.
- A set of states of A is good if it contains some state of F.
- Let $\mathcal{G}$ be the set of all good sets of A.
- Let A' be "the same automaton" as A, but with Muller condition $\mathcal{G}$.
- Let ρ be an arbitrary run of A and A'. We have

	ρ is accepting in A
iff	$\inf(\rho)$ contains some state of F
iff	$\inf(\rho)$ is a good set of A
iff	ρ is accepting in A'

From Muller to Büchi automata

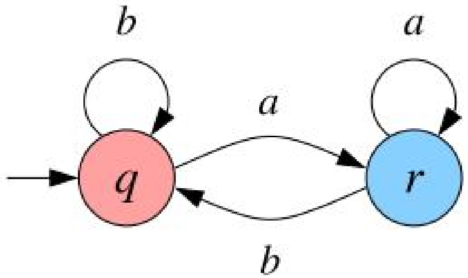
- Let A be a NMA with condition $\{F_0, F_1, \dots, F_{m-1}\}$.
- Let $A_0, \dots, A_{m-1}$ be NMAs with the same structure as A but Muller conditions $\{F_0\}, \{F_1\}, \dots, \{F_{m-1}\}$ respectively.
- We have: $L(A) = L(A_0) \cup \cdots \cup L(A_{m-1})$
- We proceed in two steps:
  1. we construct for each NMA A_i an NGA A'_i such that $L(A_i) = L(A'_i)$
  2. we construct an NGA A' such that

$$L(A') = L\left(A'_0\right) \cup \dots \cup L(A'_{m-1})$$


F_i
NMA
Transitions leaving F_i are duplicated and resent to the copy of F_i
q_1
q_m
q'_1
q'_m
NGA with accepting condition
$\{\{q'_1\}, \ldots, \{q'_m\}\}$

Input: NMA $A = (Q, \Sigma, Q_0, \delta, \{F\})$
Output: NGA $A = (Q', \Sigma, Q'_0, \delta', \mathcal{F}')$

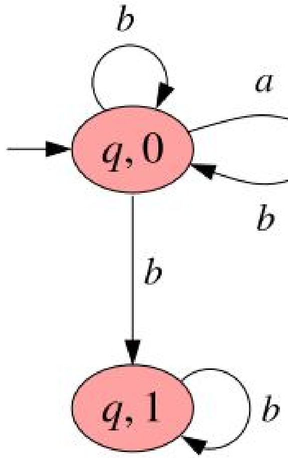
$Q', \delta', \mathcal{F}' \leftarrow \emptyset$
$Q'_0 \leftarrow \{[q_0, 0] \mid q_0 \in Q_0\}$
$W \leftarrow Q'_0$
while $W \neq \emptyset$ **do**
  pick $[q, i]$ **from** W; **add** $[q, i]$ **to** Q'
  if $q \in F$ **and** $i = 1$ **then add** $\{[q, 1]\}$ **to** $\mathcal{F}'$
  for all $a \in \Sigma,\ q' \in \delta(q, a)$ **do**
    if $i = 0$ **then**
      add $([q, 0], a, [q', 0])$ **to** δ'
      if $[q', 0] \notin Q'$ **then add** $[q', 0]$ **to** W
      if $q \in F$ **and** $q' \in F$ **then**
        add $([q, 0], a, [q', 1])$ **to** δ'
        if $[q', 1] \notin Q'$ **then add** $[q', 1]$ **to** W
    else /* $i = 1$ */
      if $q' \in F$ **then**
        add $([q, 1], a, [q', 1])$ **to** δ'
        if $[q', 1] \notin Q'$ **then add** $[q', 1]$ **to** W
return $(Q', \Sigma, q'_0, \delta', \mathcal{F}')$



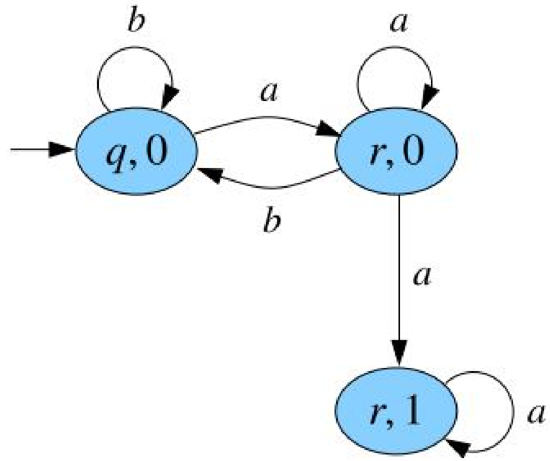
$\mathcal{F} = \{F_0, F_1\}$

$F_0 = \{q\}$

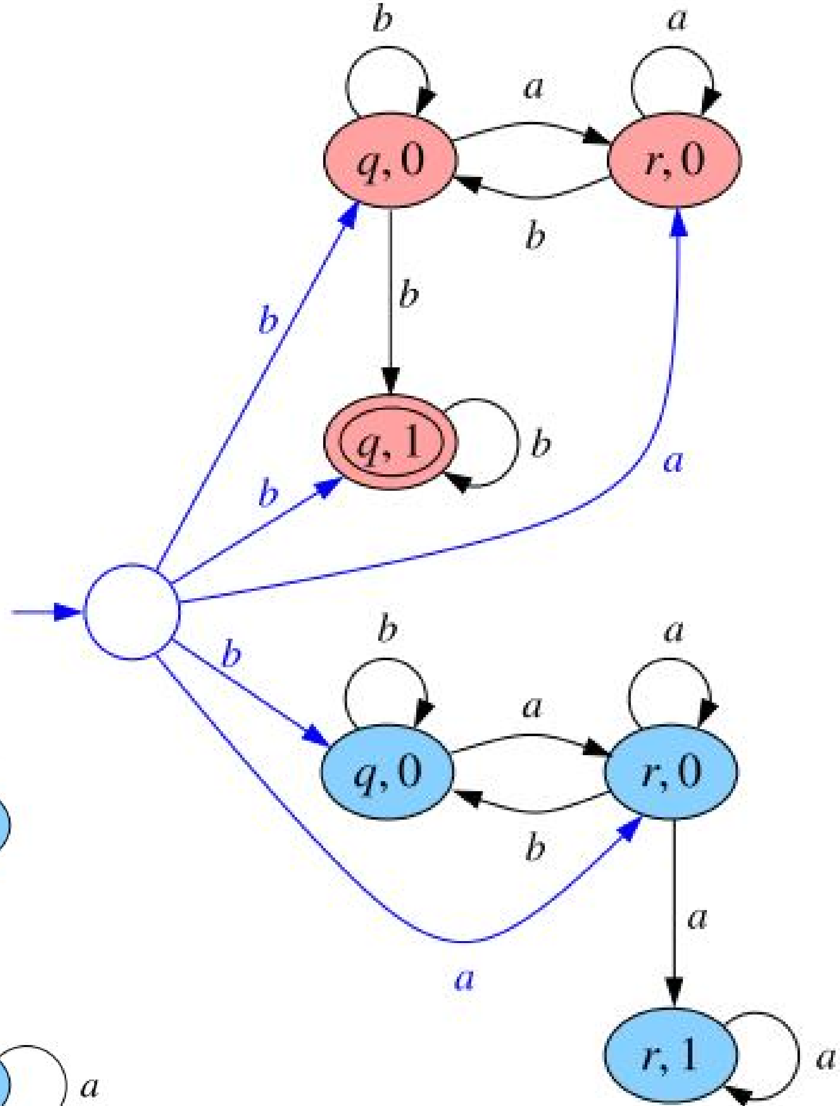
$F_1 = \{r\}$



$\mathcal{F}'_0 = \{\ [q, 1]\ \}$



$\mathcal{F}'_1 = \{\ [r, 1]\ \}$

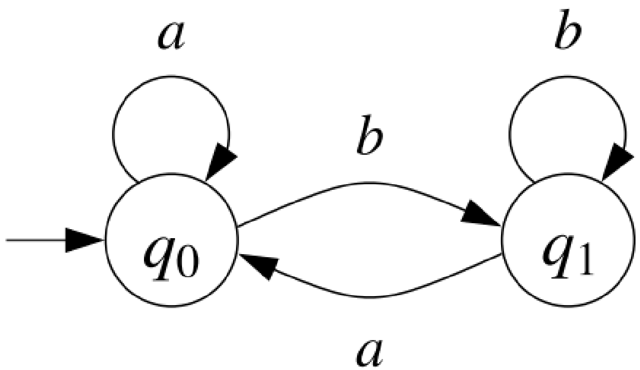


Equivalence of NMAs and DMAs

- Theorem (Safra): Any NBA with n states can be effectively transformed into a DMA of size $n^{O(n)}$.

  Proof: Omitted.

- DMA for $(a + b)^* b^\omega$:



with accepting condition $\{\{q_1\}\}$

- Question: Are there other classes of omega-automata with
  - the same expressive power as NBAs or NGAs, and
  - with equivalent deterministic and nondeterministic versions?
- Answer: Yes, Muller automata

Is the quest over?

- Recall the translation NBA→NMA
- The NMA has the same structure as the NBA; its accepting condition are all the good sets of states.
- The translation has exponential complexity.

New question: Is there a class of ω-automata with

- the same expressive power as NBAs,
- equivalent deterministic and nondeterministic versions, and
- polynomial conversions to and from Büchi automata?

Rabin automata

- The acceptance condition is a set of pairs $\{ \langle F_0, G_0 \rangle, \ldots, \langle F_{m-1}, G_{m-1} \rangle \}$
- A run ρ is accepting if there is a pair $\langle F_i, G_i \rangle$ such that ρ visits the set F_i infinitely often and the set G_i finitely often.
- Translations NBA→NRA and NRA→NBA are left as an exercise.
- Theorem (Safra): Any NBA with n states can be effectively transformed into a DRA with $n^{O(n)}$ states and $O(n)$ accepting pairs.

Is the quest over?

- The accepting condition of Rabin automata is not closed under negation: the negation of
  $$\exists i \in \{1, \ldots, m\}\colon \mathrm{inf}(\rho) \cap F_i \neq \emptyset \wedge \mathrm{inf}(\rho) \cap G_i = \emptyset$$
  is of the form
  $$\forall i \in \{1, \ldots, m\}\colon \mathrm{inf}(\rho) \cap F_i = \emptyset \vee \mathrm{inf}(\rho) \cap G_i \neq \emptyset$$
  or, equivalently
  $$\forall i \in \{1, \ldots, m\}\colon \mathrm{inf}(\rho) \cap G_i = \emptyset \Rightarrow \mathrm{inf}(\rho) \cap F_i = \emptyset$$
- This is the Streett condition.
- The Büchi condition is a special case of the Streett condition.
- However, the translation from Streett to Büchi is exponential.

Is the quest over?

New question: Is there a class of ω-automata with

- the same expressive power as NBAs,
- equivalent deterministic and nondeterministic versions,
- polynomial conversions to and from Büchi automata, and
- an accepting condition closed under negation?

Parity automata

- The acceptance condition is a sequence $(F_1, \dots, F_{2k})$ of sets of states such that $F_1 \subseteq F_2 \subseteq \cdots \subseteq F_{2k} = Q$.
- A run ρ is accepting if the minimal index i such that ρ visits the set F_i infinitely often is even.
- NBA→NPA. $F \rightarrow (\emptyset, F, Q, Q)$
- NPA→NBA. NPA→NRA →NBA.
- NPA→NRA. $(F_1, \dots, F_{2k}) \rightarrow \{\langle F_{2k}, F_{2k-1}\rangle, \dots, \langle F_4, F_3\rangle, \langle F_2, F_1\rangle\}$
- Theorem (Safra,Piterman): Any NBA with n states can be effectively transformed into a DPA with $n^{O(n)}$ states and $O(n)$ accepting sets.
- Complementation of DPAs. $(F_1, \dots, F_{2k}) \rightarrow (\emptyset, F_1, \dots, F_{2k}, Q)$

Parity automata

New question: Is there a class of ω-automata with

- the same expressive power as NBAs,
- equivalent deterministic and nondeterministic versions,
- polynomial conversions to and from Büchi automata, and
- an accepting condition closed under negation?

- Answer: Yes, parity automata