omega-Automat

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.

Omega-languages

Let Σ be an alphabet. An *infinite* word, also called an ω-*word*, is an infinite sequence $a_0a_1a_2\ldots$ of letters of Σ. The concatenation of a finite word $w_1 = a_1\ldots a_n$ and an ω-word $w_2 = b_1b_2\ldots$ is the ω-word $w_1w_2 = a_1\ldots a_nb_1b_2\ldots$, sometimes also denoted by $w_1 \cdot w_2$. Notice that $\varepsilon \cdot w = w$. We denote by Σ^ω the set of all ω-words over Σ. A set $L \subseteq \Sigma^\omega$ of ω-words is an *infinitary language* or ω-*language* over Σ.

The *concatenation* of a language L_1 and a language or ω-language L_2 is $L_1 \cdot L_2 = \{w_1w_2 \in \Sigma^\omega \mid w_1 \in L_1, w_2 \in L_2\}$. The ω-*iteration* of a language $L \subseteq \Sigma^*$ is the ω-language $L^\omega = \{w_1w_2w_3\ldots \mid w_i \in L \setminus \{\epsilon\}\}$. Observe that $\{\epsilon\}^\omega = \emptyset$, in contrast to the case of finite words, where $\{\epsilon\}^* = \{\emptyset\}$. Notice that $\{\epsilon\}^\omega = \{\emptyset\}$ does not make sense, because all the words of L^ω must have infinite length.

Omega-regular expressions

Definition 11.1 *ω-regular expressions s over an alphabet Σ are defined by the following grammar, where $r \in \mathcal{RE}(\Sigma)$ is a regular expression*

$$s ::= \emptyset \mid r^\omega \mid rs_1 \mid s_1 + s_2$$

Sometimes we write $r \cdot s_1$ insteadf rs_1. The set of all ω-regular expressions over Σ is written $\mathcal{RE}_\omega(\Sigma)$. The language $\mathcal{L}_\omega(s) \subseteq \Sigma$ *of an ω-regular expression $s \in \mathcal{RE}_\omega(\Sigma)$ is defined inductively as*

- $\mathcal{L}(\emptyset) = \emptyset$;
- $\mathcal{L}(r^\omega) = (\mathcal{L}(r))^\omega$;
- $\mathcal{L}_\omega(rs_1) = \mathcal{L}(r) \cdot \mathcal{L}_\omega(s_1)$; *and*
- $\mathcal{L}(s_1 + s_2) = \mathcal{L}(s_1) \cup \mathcal{L}(s_2)$.

A language L is ω-regular *if there is an ω-regular expression s such that $L = \mathcal{L}_\omega(s)$.*

Examples

Consider the alphabet {a,b}. We use ° instead of omega.

- Words containing infinitely many a's: (b*a)°

- Words containing only finitely a's:

- Words containing infinitely many a's
  and infinitely many b's:

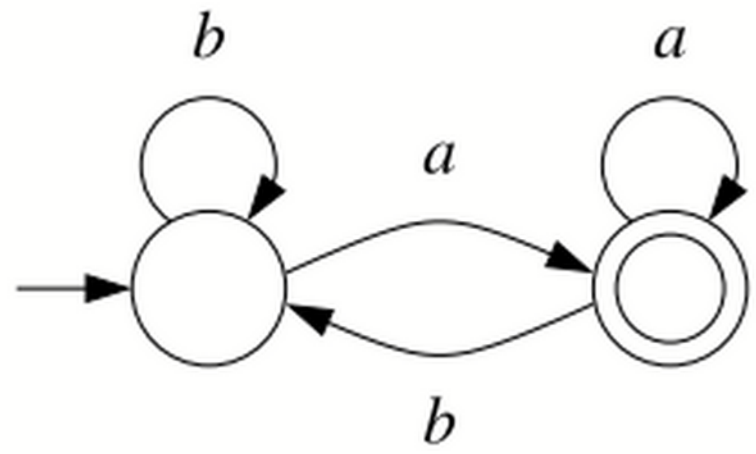
Consider now the alphabet {a,b,c}.

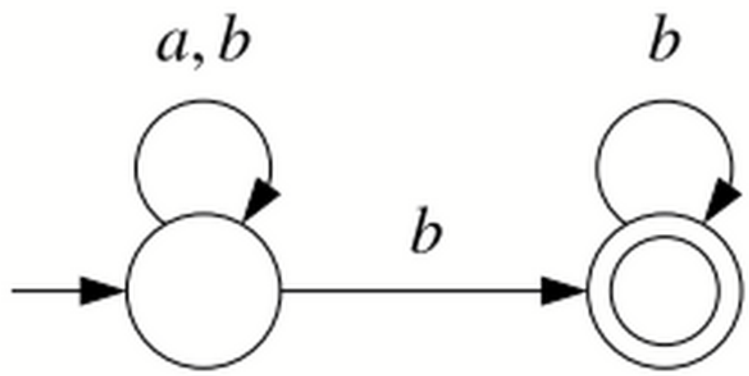
- Words containg infinitely many occurrences of ab and
  infinitely many occurrences of ba:

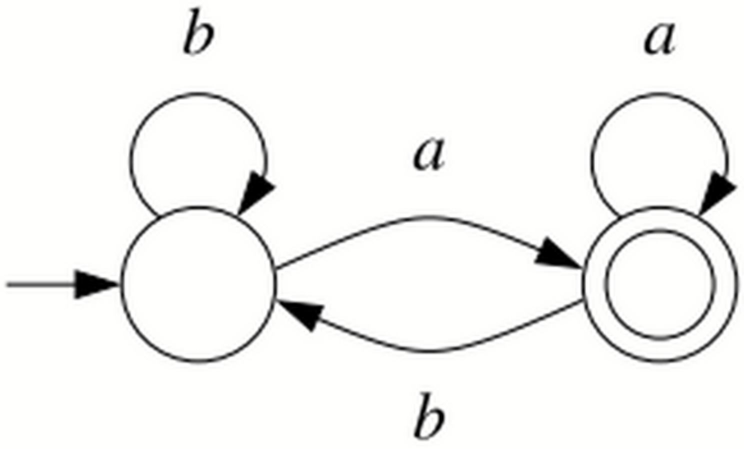
Büchi automata

Invented by J.R. Büchi for theoretical purposes (decision procedures in logic)

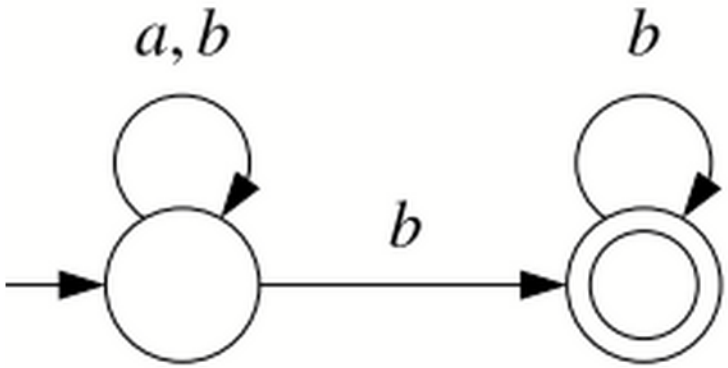
Same syntax as NFAs and DFAs, but different interpretation.







Words containing
infinitely many a's

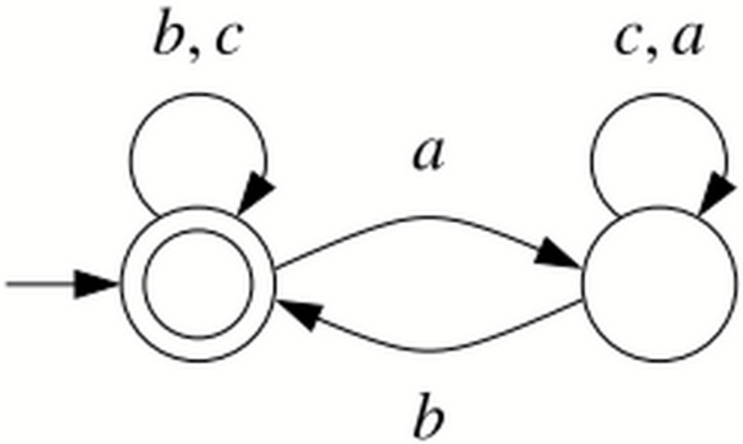


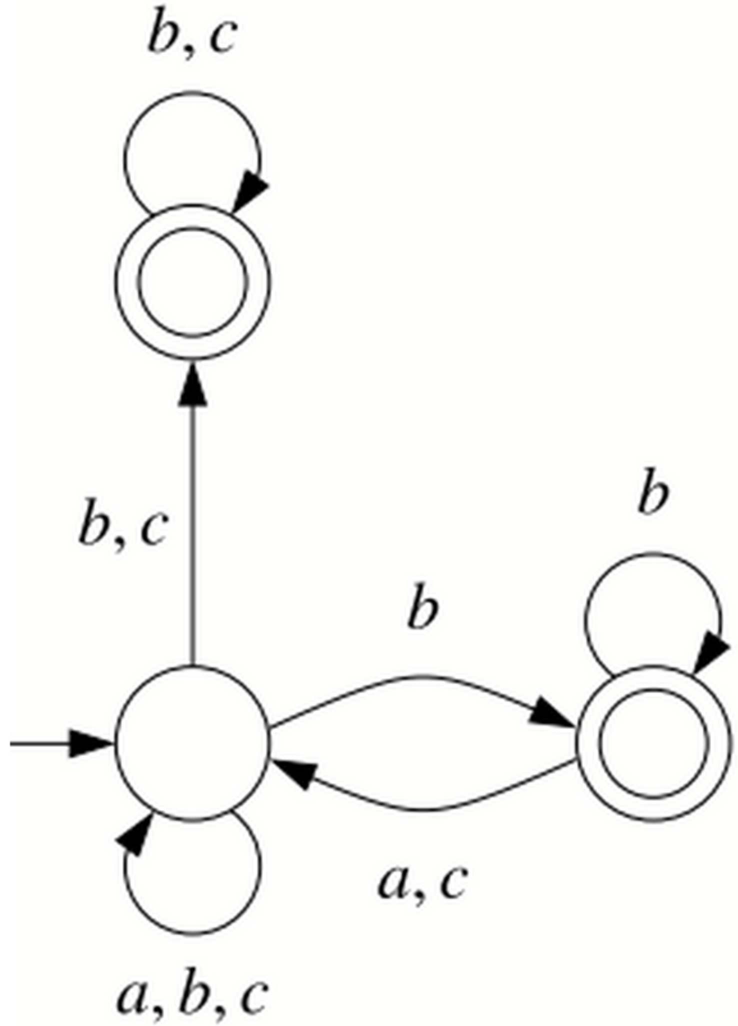
Words containing
only finitely many a's

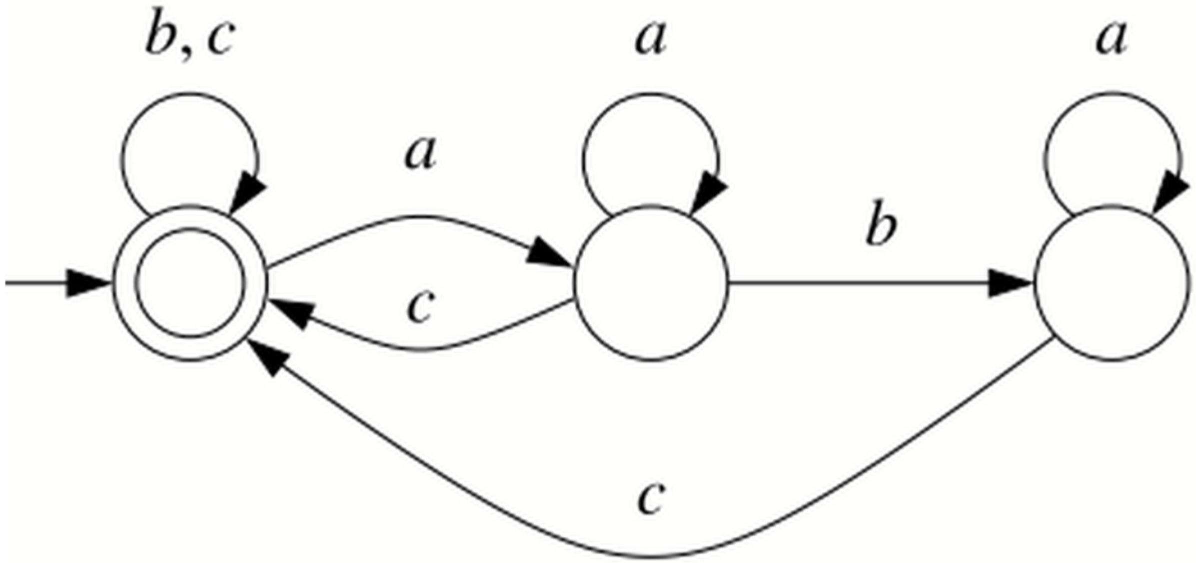
Definition 11.2 *A* nondeterministic Büchi automaton *(NBA) is a tuple* $A = (Q, \Sigma, \delta, q_0, F)$*, where* Q*,* Σ*,* δ*,* q_0*, and* F *are defined as for NFAs. A* run *of* A *on an* ω*-word* $a_0 a_1 a_2 \ldots$ *is an infinite sequence* $\rho = p_0 \xrightarrow{a_0} p_1 \xrightarrow{a_1} p_2 \ldots$*, such that* $p_i \in Q$ *for* $0 \leq i \leq n$ *and* $\delta(p_i, a_i) = p_{i+1}$ *for* $0 \leq i < n - 1$*. Let* $\inf(\rho)$ *be the set* $\{q \in Q \mid q = p_i$ *for infinitely many i's*$\}$*, i.e., the set of states that occur in* ρ *infinitely often. The run* ρ *is* accepting *if there is some accepting state that repeats in* ρ *infinitely often, i.e., if* $\inf(\rho) \cap F \neq \emptyset$*.* A accepts *an* ω*-word* $w \in \Sigma^\omega$ *if it has an accepting run on* w*. The* language recognized *by* A *is the set* $\mathcal{L}_\omega(A) = \{w \in \Sigma^\omega \mid w$ *is accepted by* $A\}$*.*

Deterministic Büchi Automata (DBAs) are defined as for finite words.

More examples







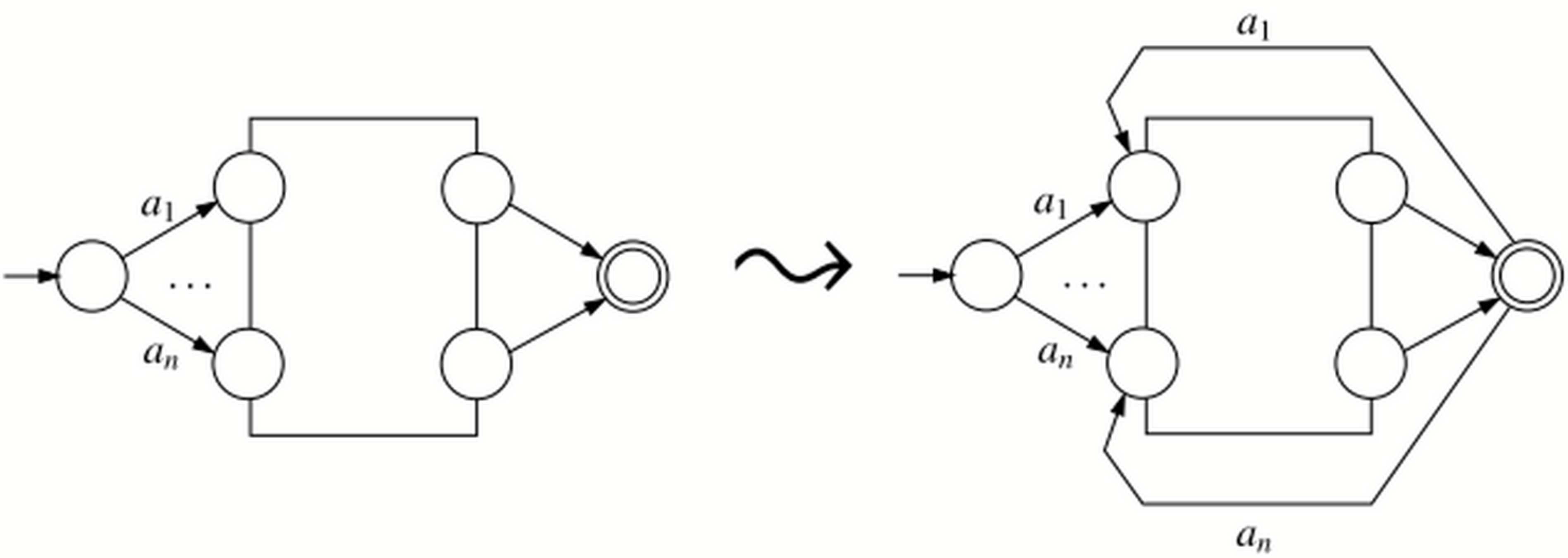
From omega-regular expressions to NBAs

Recall the syntax of omega-regular expressions:

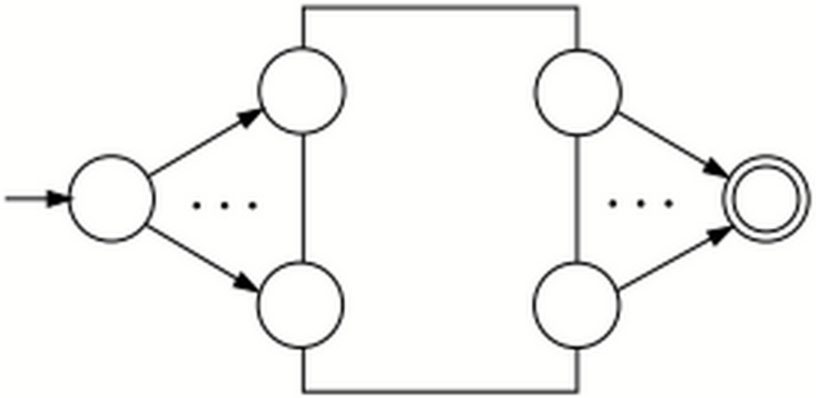
$$s ::= \emptyset \mid r^{\omega} \mid r s_1 \mid s_1 + s_2$$

We first preprocess the omega-regular expression to eliminate the occurrences of the emptyset-symbol.

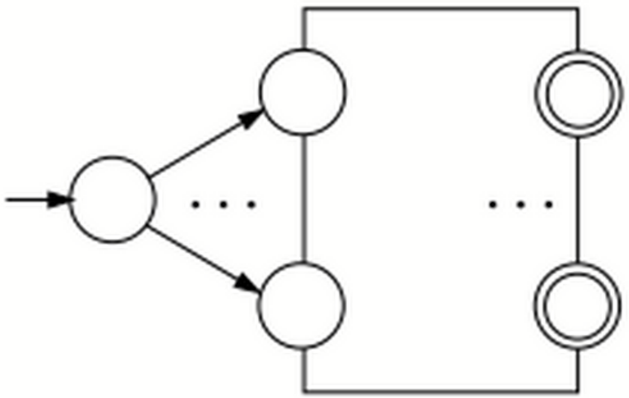
$$\begin{array}{rcl} \emptyset^{\omega} & \rightsquigarrow & \emptyset \\ \emptyset \cdot s & \rightsquigarrow & \emptyset \\ \emptyset + s & \rightsquigarrow & s \end{array} \qquad \begin{array}{rcl} & & \\ r \cdot \emptyset & \rightsquigarrow & \emptyset \\ s + \emptyset & \rightsquigarrow & s \end{array}$$



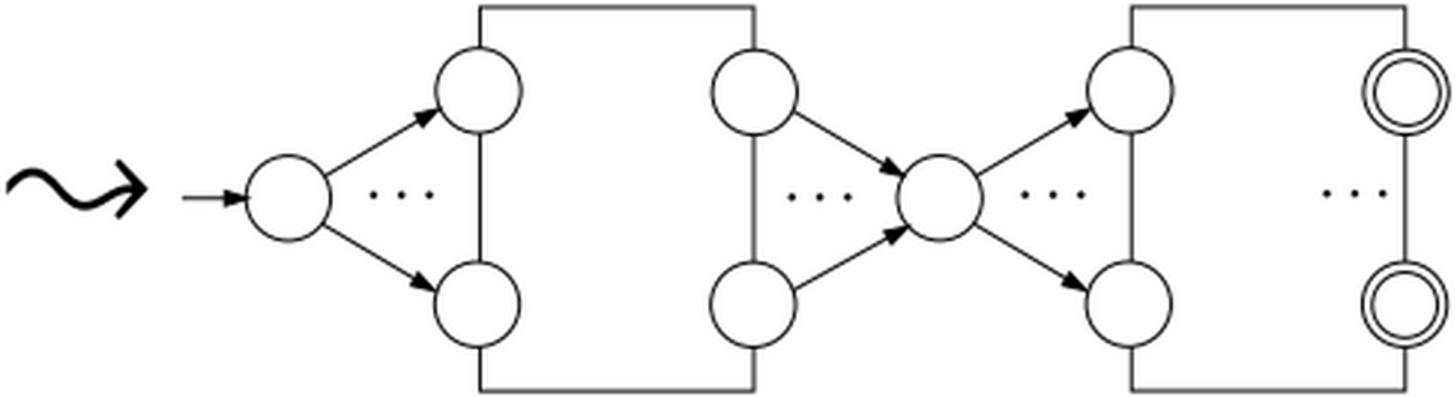
NFA for r NBA for r^ω



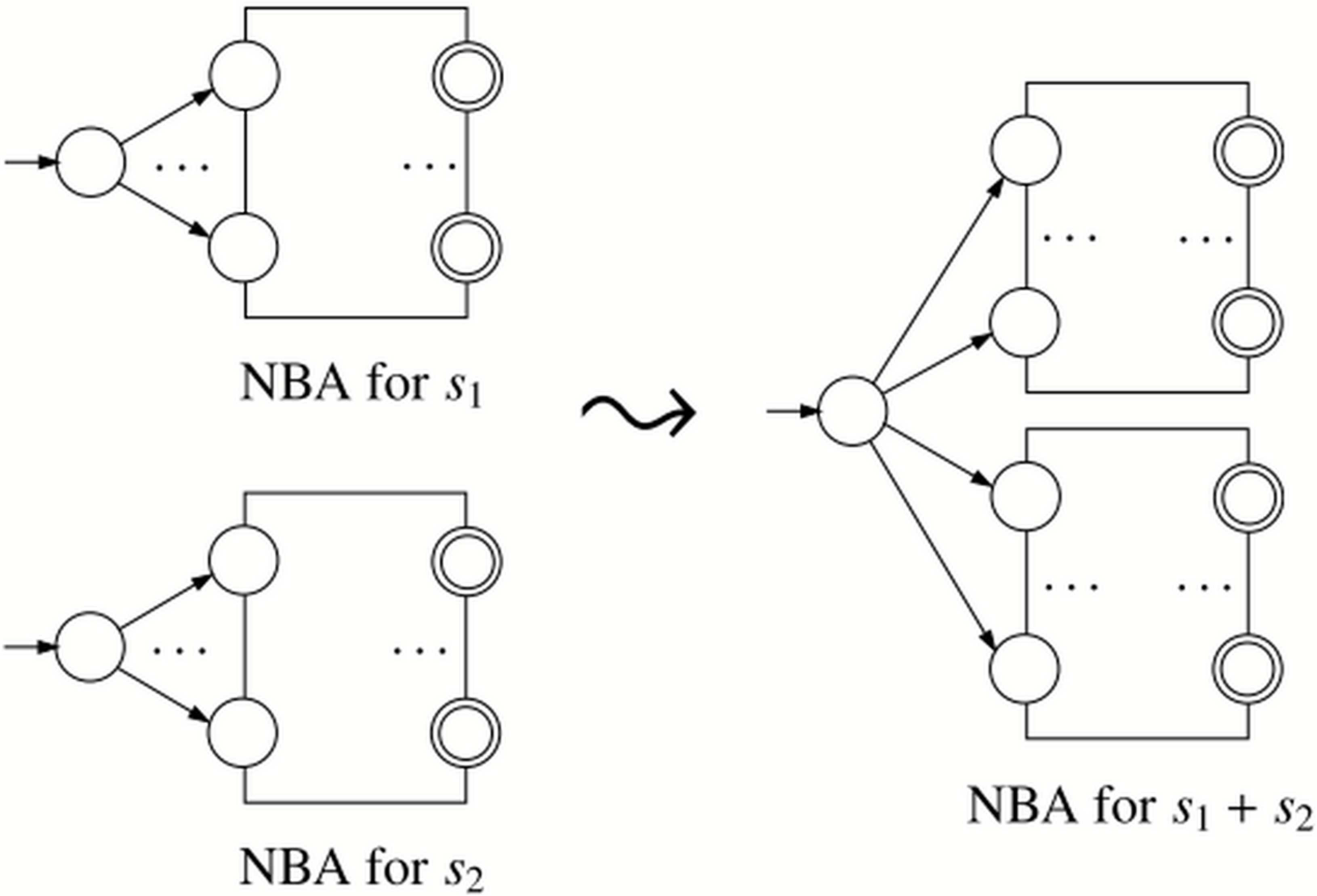
NFA for r



NBA for s



NBA for $r \cdot s$


NBA for s_1
NBA for s_2
NBA for $s_1 + s_2$

From NBAs to omega-regular expressions

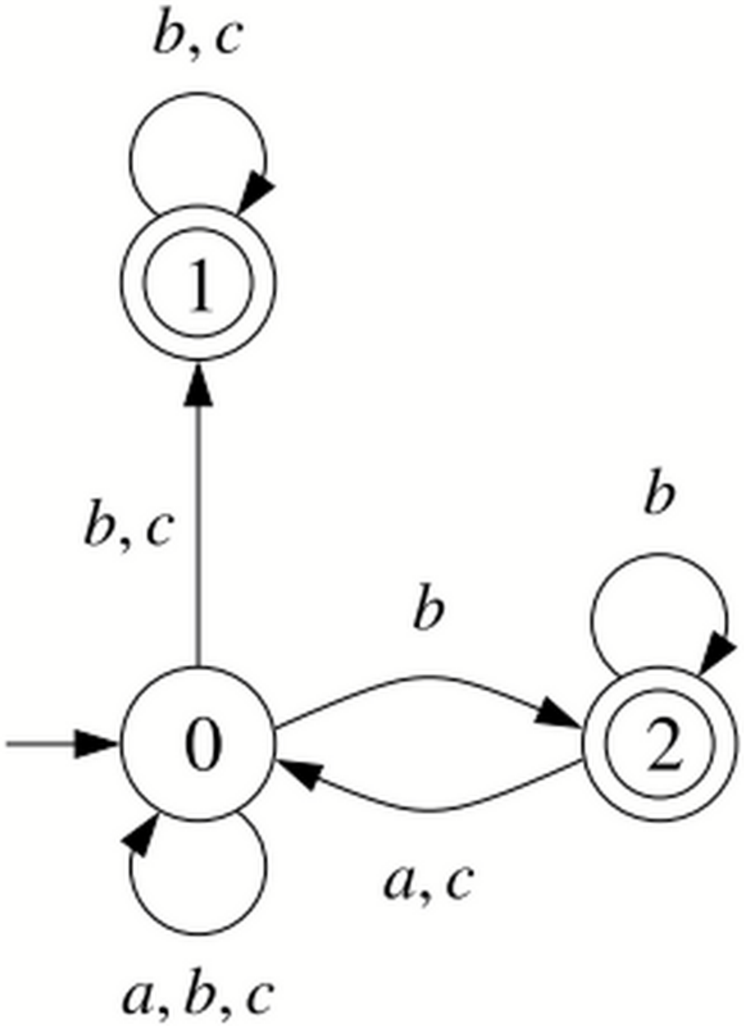
Let $A = (Q, \Sigma, \delta, q_0, F)$ be a NBA. For every two states $q, q' \in Q$, let $A_q^{q'} = (Q, \Sigma, \delta, q, \{q'\})$ be the NFA (not the NBA!) obtained from A by changing the initial state to q and the final state to q'. Using algorithm *NFAtoRE* we can construct a regular expression $r_q^{q'}$ such that $\mathcal{L}(A_q^{q'}) = \mathcal{L}(r_q^{q'})$.

We use these regular expressions to find an ω-regular expression for $\mathcal{L}_\omega(A)$. For every accepting state $q \in F$, let $L_q \subseteq \mathcal{L}_\omega(A)$ be the set of ω-words w such that some run of A on w visits the state q infinitely often. We have $\mathcal{L}_\omega(A) = \bigcup_{q \in F} L_q$.

Every word $w \in L_q$ can be split into an infinite sequence $w_1 w_2 w_3 \ldots$ of finite, nonempty words, where w_1 is the word read by the automaton until it visits q for the first time, and for every $i > 1$ w_i is the word read by the automaton between the i-th and the $(i+1)$-th visits to q. It follows $w_1 \in \mathcal{L}(r_{q_0}^q)$, and $w_i \in \mathcal{L}(r_q^q)$ for every $i > 1$. So we have $L_q = \mathcal{L}_\omega(r_{q_0}^q \left(r_q^q\right)^\omega)$, and so

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

is the ω-regular expression we are looking for.



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

$$r_0^1\left(r_1^1\right)^\omega + r_0^2\left(r_2^2\right)^\omega$$

$$(a+b+c)^*(b+c)^+(b+c)^\omega + b^+(a+c)b^+(b+(a+c)(a+b+c)^*b)^\omega$$

Inequivalence of NBAs and DBAs

Proposition 11.4 *The language $L = (a + b)^* b^\omega$, (i.e., L consists of all infinite words in which a occurs only finitely many times) is not recognized by any DBA.*

Proof: Assume by way of contradiction that $L = \mathcal{L}_\omega(A)$, for some DBA $A = (\{a, b\}, Q, q_0, \delta, F)$. We extend δ to a mapping $Q \times \{a, b\}^* \rightarrow Q$ in the usual way: $\hat{\delta}(q, \epsilon) = q$ and $\hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a)$.

Consider the infinite word $w_0 = b^\omega$. Clearly, w_0 is accepted by A, so A has an accepting run on w_0. Thus, w_0 has a finite prefix u_0 such that $\hat{\delta}(q_0, u_0) \in F$. Consider now the infinite word $w_1 = u_0 a b^\omega$. Clearly, w_1 is also accepted by A, so A has an accepting run on w_1. Thus, w_1 has a finite prefix $u_0 b u_1$ such that $\hat{\delta}(q_0, u_0 a u_1) \in F$. In a similar fashion we can continue to find finite words u_i such that $\hat{\delta}(q_0, u_0 a u_1 a \ldots a u_i) \in F$. Since Q is finite, there are i, j, where $0 \leq i < j$, such that $\delta(q_0, u_0 a u_1 a \ldots a u_i) = \delta(q_0, u_0 a u_1 a \ldots a u_i a \ldots a u_j)$. It follows that A has an accepting run on

$$u_0 a u_1 a \ldots a u_i (a u_{i+1} \ldots u_{j-1} a u_j)^\omega.$$

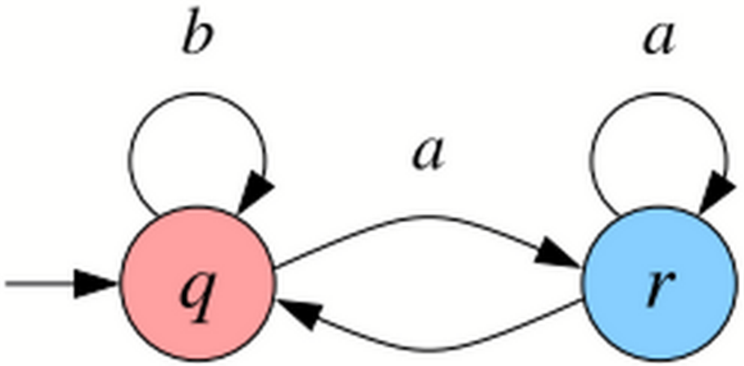
But the latter word has infinitely many occurrences of a, so it does not belong to L. $\square$

Generalized Büchi automata

Equivalent to Büchi automata, but more adequate for some constructions.

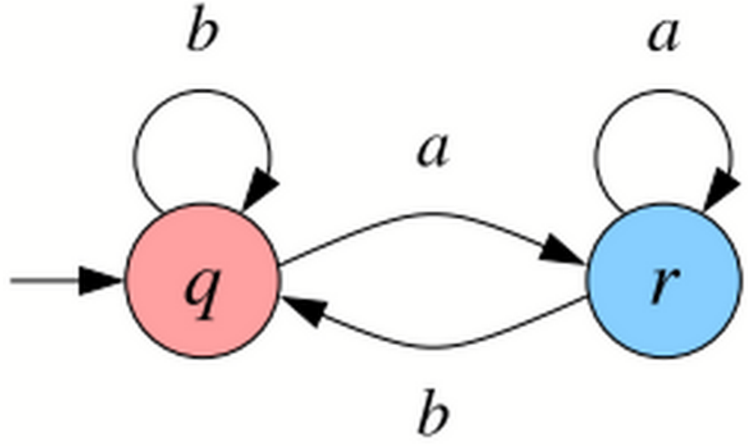
- Several sets of accepting states.

- A run is accepting if it visits at least one state OF EACH SET infinitely often.



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

A *generalized Büchi automaton* (NGA) differs from a Büchi automaton in that it has a *collection of sets of accepting states* $\mathcal{F} = \{F_0, \ldots, F_{m-1}\}$, instead of only one set F. A run ρ is accepting if *for every set* $F_i \in \mathcal{F}$ some state of F_i is visited by ρ infinitely often. Formally, ρ is accepting if $\inf(\rho) \cap F_i \neq \emptyset$ for every $i \in \{0, \ldots, m-1\}$. Abusing language, we speak of the *generalized Büchi condition* $\mathcal{F}$. Ordinary Büchi automata correspond to the special case $m = 1$.



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

From NGAs to NBAs

Important fact:

A1, ..., An all happen infinitely often

is equivalent to

A1 eventually happens
and
after every occurrence of Ai there is an occurrence of A(i+1)

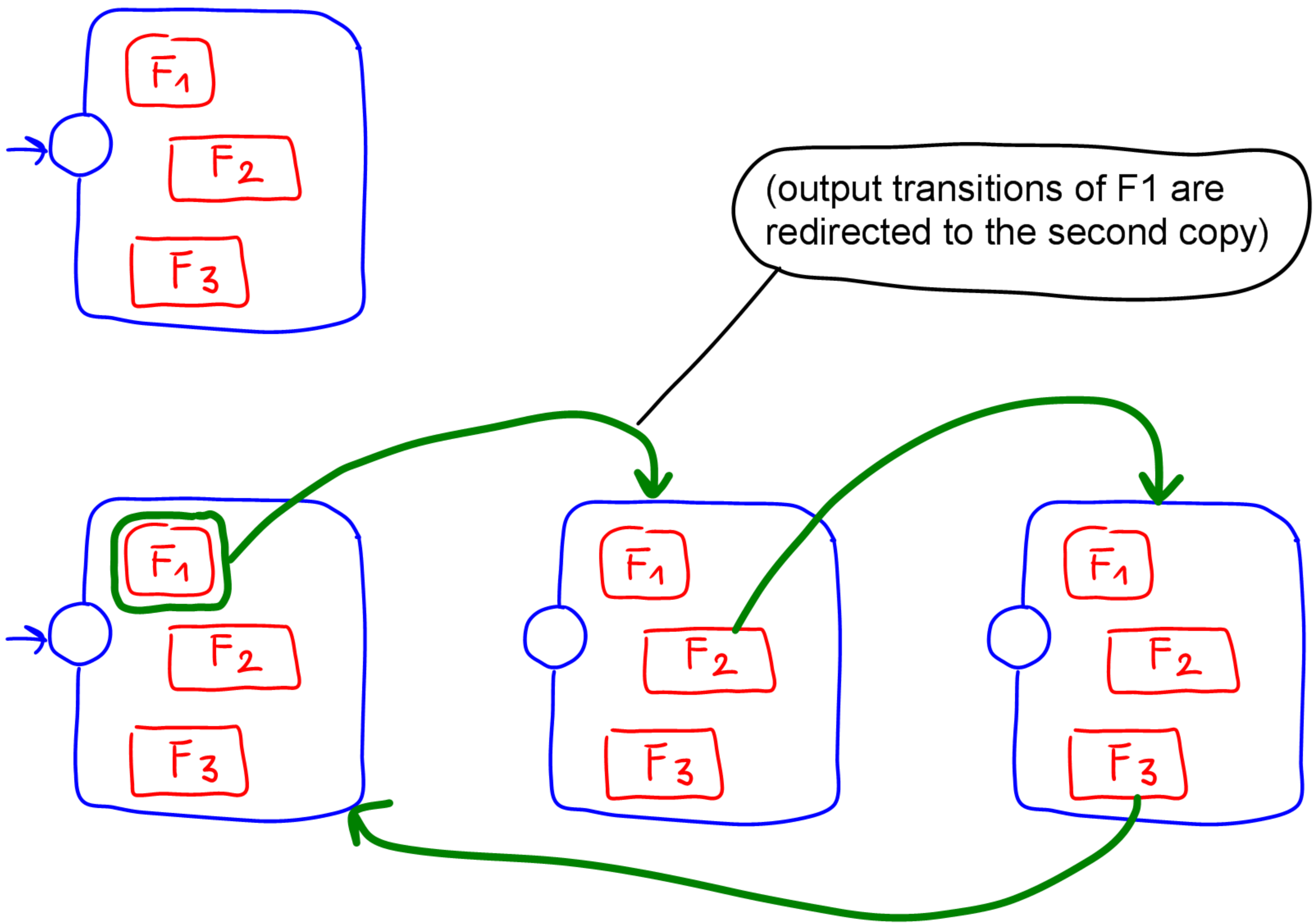
From NGAs to NBAs

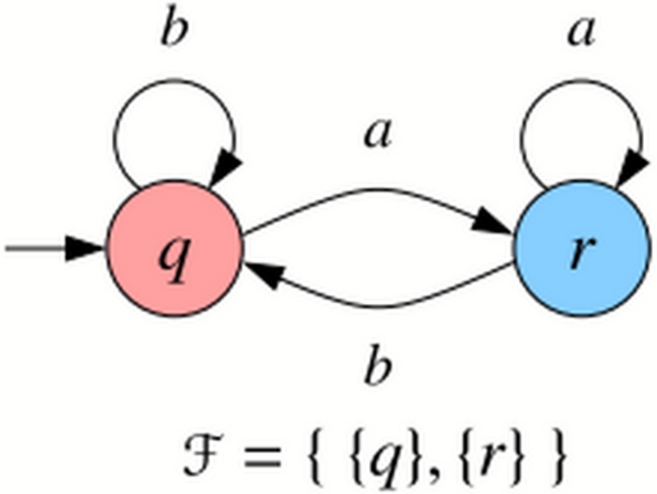
Application:

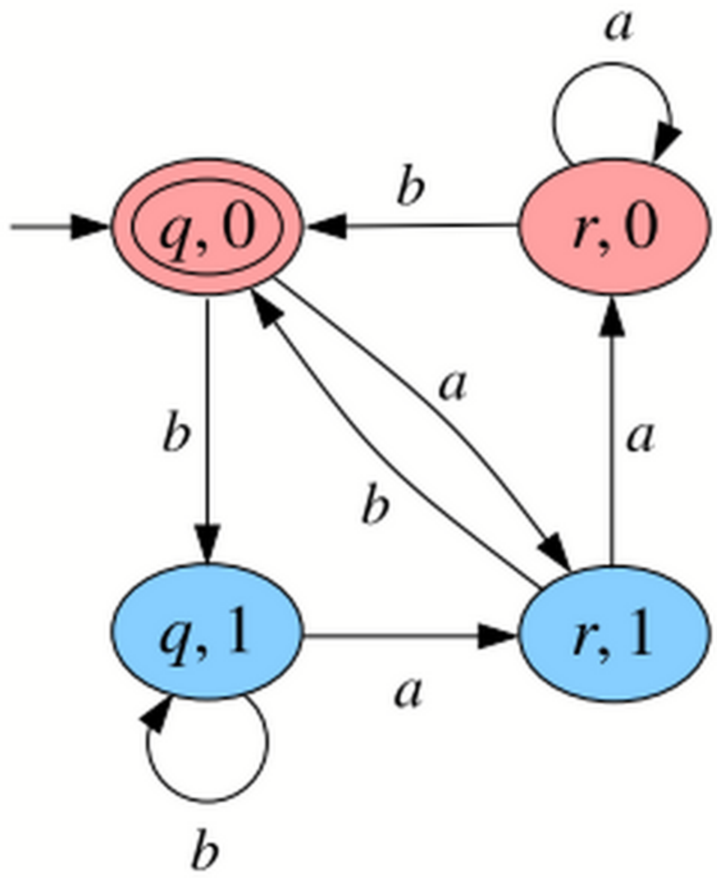
F1, ..., Fn are all visited infinitely often

is equivalent to

F1 is eventually visited
and
after every visit to Fi there is a visit to F(i+1)


F1
F2
F3
(output transitions of F1 are redirected to the second copy)
F1
F2
F3
F1
F2
F3
F1
F2
F3


b
a
a
q
r
b
ℱ = { {q}, {r} }


a
b
q, 0
r, 0
b
a
a
b
q, 1
r, 1
a
b

NGAtoNBA(A)

Input: NGA $A = (Q, \Sigma, q_0, \delta, \mathcal{F})$, where $\mathcal{F} = \{F_1, \ldots, F_m\}$

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

$Q', \delta', F' \leftarrow \emptyset$

$q'_0 \leftarrow [q_0, 0]$

$W \leftarrow \{[q_0, 0]\}$

while $W \neq \emptyset$ **do**

  pick $[q, i]$ **from** W

  add $[q, i]$ **to** Q'

  if $q \in F_0$ **and** $i = 0$ **then add** $[q, i]$ **to** F'

  for all $a \in \Sigma$ **do**

    for all $q' \in \delta(q, a)$ **do**

      if $q \notin F_i$ **then**

        if $[q', i] \notin Q'$ **then add** $[q', i]$ **to** W

        add $([q, i], a, [q', i])$ **to** δ'

      else /* $q \in F_i$ */

        if $[q', i \oplus 1] \notin Q'$ **then add** $[q', i \oplus 1]$ **to** W

        add $([q, i], a, [q', i \oplus 1])$ **to** δ'

return $(Q', \Sigma, \delta', q'_0, F')$

DGAs have the same expressive power as DBAs, and so are nt 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?

The only thing we are willing to change is the acceptance condition!

Muller automata

Muller automata only differ from Büchi automata in the acceptance condition. Like a generalized Büchi automaton, a *(nondeterministic) Muller automaton* (NMA) has a collection $\{F_0, \ldots, F_{m-1}\}$ of sets of accepting states. A run ρ is *accepting* if the set of states ρ visits infinitely often is equal to one of the F_i's. Formally, ρ is accepting if $\inf(\rho) = F_i$ for some $i \in \{0, \ldots, m-1\}$. We speak of the *Muller condition* $\{F_0, \ldots, F_{m-1}\}$.

From Büchi to Muller automata

Let A be a Büchi automaton with Büchi condition F.
Call a set of states of A "good" if it contains at least one state of F.
Let G be the set of all good sets of A.
Let A' be "the same automaton" as A, but with Muller condition G.

	run r of A is accepting
iff	inf(r) contains some state of F
iff	inf(r) is a good set of A
iff	run r of A' is accepting

From Muller to Büchi automata

It suffices to transform a given NMA into an equivalent NGA.

Let A be a NMA with condition {F1,...,Fn}.
Let A1,..., An be NMAs with the same structure as A but Muller conditions {F1}, {F2},..., {Fn}, respectively.
We have:

$$L(A) = L(A1) \cup L(A2) \cup \ldots \cup L(An)$$

We proceed in two steps:

(1) we construct for each NMA Ai an NGA Ai'
(2) we construct an NGA A' such that

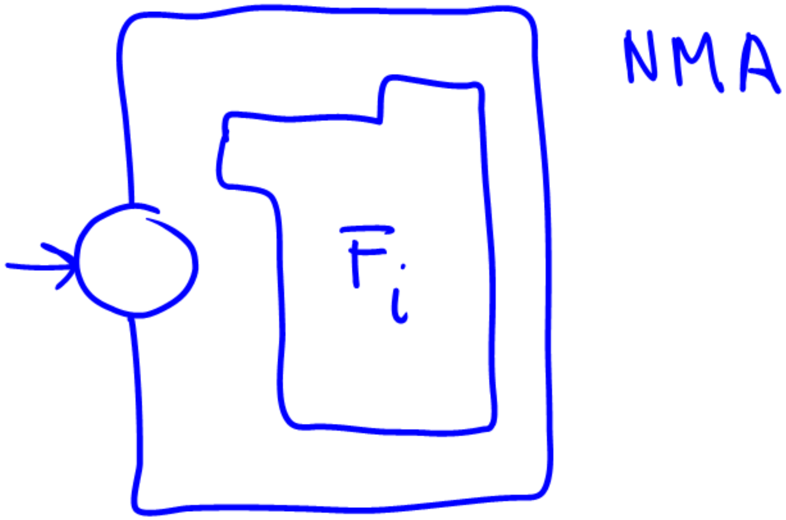
$$L(A') = L(A1') \cup L(A2') \cup \ldots \cup L(An')$$

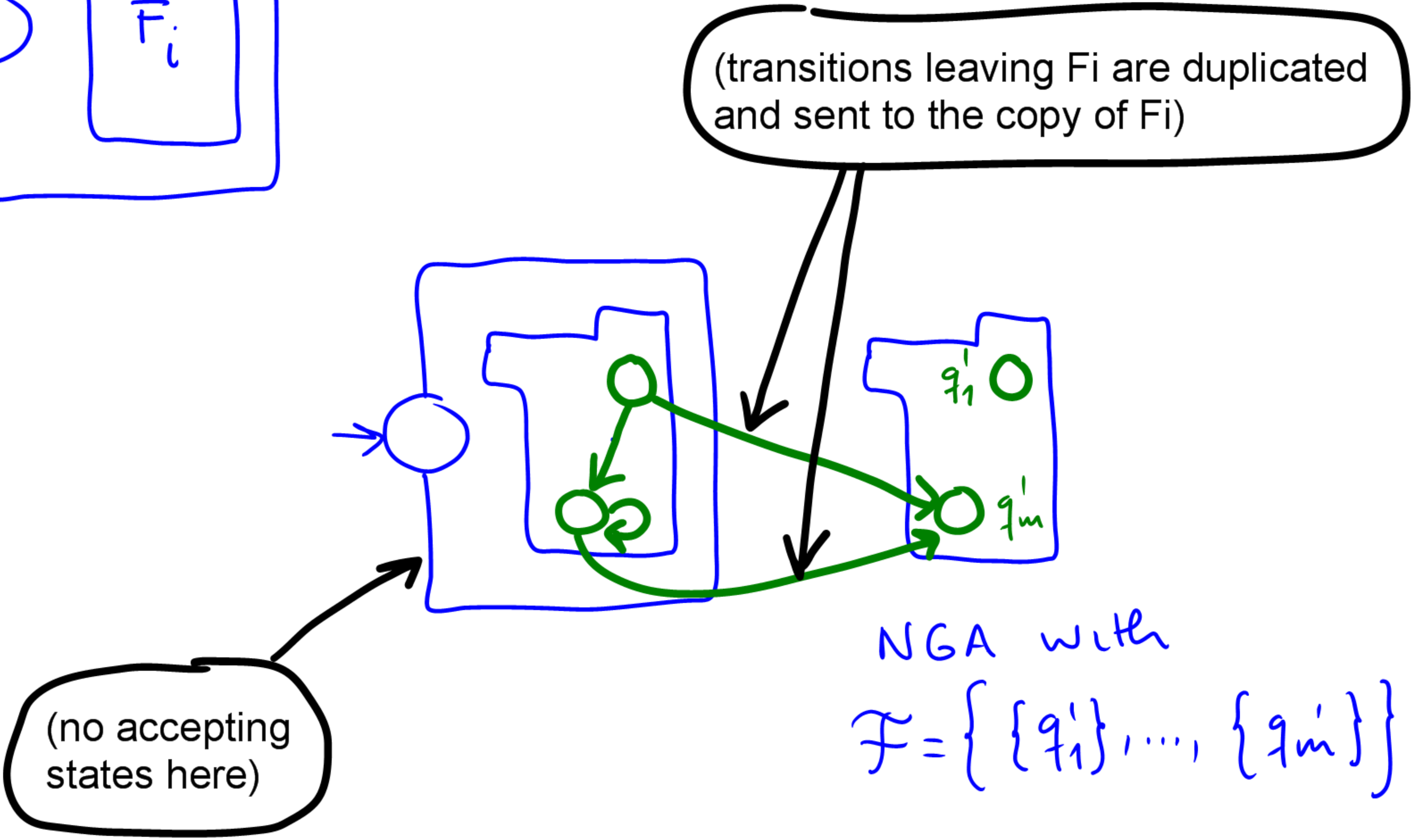
(1) we construct for each NMA Ai an NGA Ai'

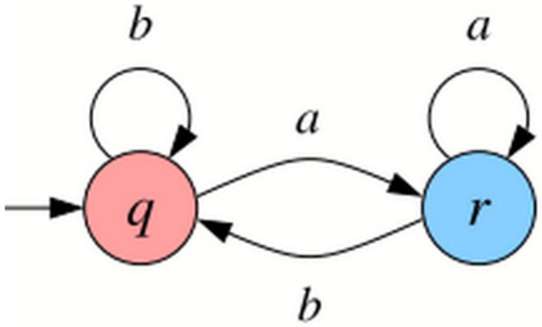
A run of Ai is accepting iff
- it visits infinitely often every state of Fi, and
- it only visits finitely often every other state.

If Fi = {q1,...,qm}, this is equivalent to:

A run of Ai is accepting iff
- from some point on it "stays within" Fi, and
- it visits infinitely often each of the sets {q1}, {q2},....,{qm}
  (a generalized Büchi condition).


NMA
F_i

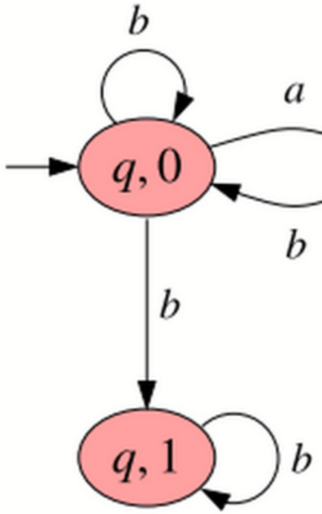

(transitions leaving Fi are duplicated and sent to the copy of Fi)
q_1'
q_m'
(no accepting states here)
NGA with
$\mathcal{F} = \{ \{q_1'\}, \ldots, \{q_m'\} \}$



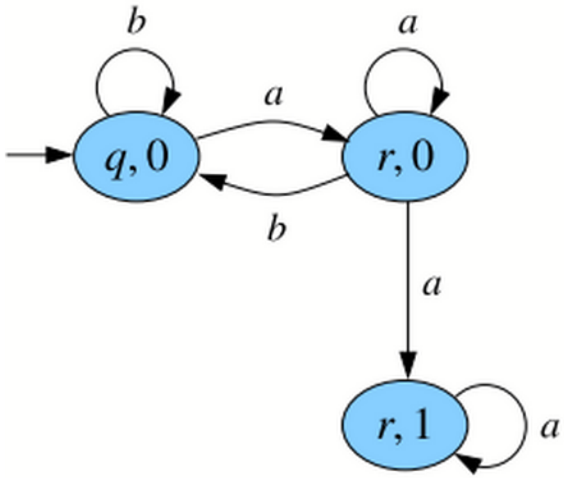
$$\mathcal{F} = \{F_0, F_1\}$$

$$F_0 = \{q\}$$

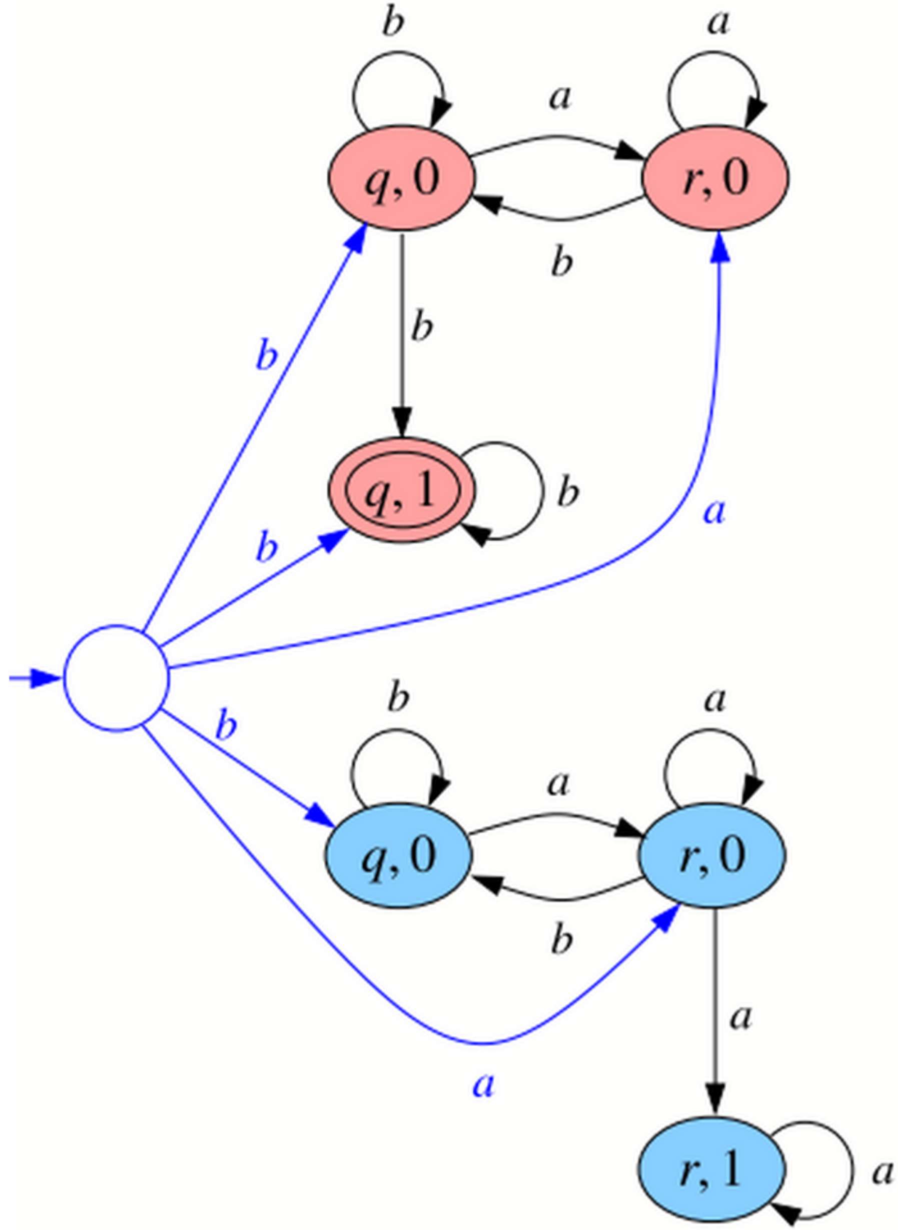
$$F_1 = \{r\}$$



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



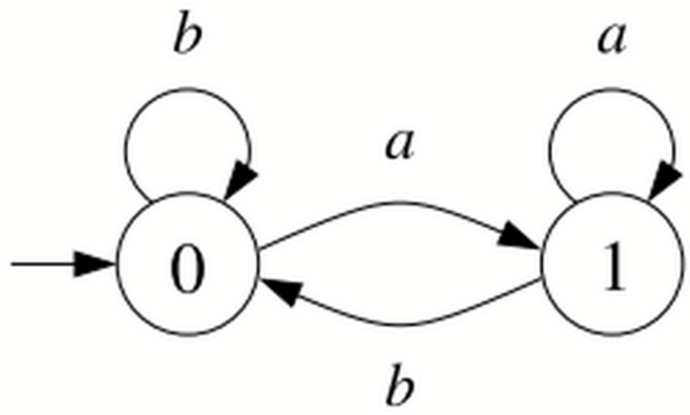
$\mathcal{F}'_1 = \{\ [r, 1]\ \}$



Equivalence of NMAs and DMAs

Theorem 11.6 (Safra) *Any NBA with n states can be effectively transformed into a DMA of size* $n^{O(n)}$.

We can easily give a deterministic Muller automaton for the language $L = (a + b)^* b^\omega$, which, as shown in Proposition 11.4, is not recognized by any DBA. The automaton is



with Muller condition { {1} }. The accepting runs are the runs ρ such that $\inf(\rho) = \{1\}$,

Rabin automata

Muller automata recognize all omega-regular languages, and can be determinized.

But the translation from a Büchi automaton to a Muller automaton has exponential complexity.

Rabin automata enjoy the same properties as Muller automata, and there are back and forth polynomial translations between Büchi and Rabin automata.

The acceptance condition of a *nondeterministic Rabin automaton* (NRA) is a set of pairs $\mathcal{F} = \{\langle F_0, G_0\rangle, \ldots, \langle F_m, G_m\rangle\}$, where the F_i's and G_i's are sets of states. A run ρ is *accepting* if there is a pair $\langle F_i, G_i\rangle$ such that ρ visits *some* state of F_i *infinitely often* and *all* states of G_i *finitely often*. Formally, ρ is accepting if there is $i \in \{1, \ldots, m\}$ such that $\inf(\rho) \cap F_i \neq \emptyset$ and $\inf(\rho) \cap G_i = \emptyset$.

NBA → NRA. A Büchi condition $\{q_1, \ldots, q_k\}$ corresponds to the Rabin condition $\{(\{q_1\}, \emptyset), \ldots, (\{q_n\}, \emptyset)\}$.

NRA → NBA. Given a Rabin automaton $A = (Q, \Sigma, q_0, \delta, \{\langle F_0, G_0\rangle, \ldots, \langle F_{m-1}, G_{m-1}\rangle\})$, it is easy to see that, as for Muller automata, we have $\mathcal{L}_\omega(A) = \bigcup_{i=0}^{m-1} \mathcal{L}_\omega(A_i)$, where $A_i = (Q, \Sigma, q_0, \delta, \{\langle F_i, G_i\rangle\})$. In this case we directly translate each A_i into an NBA. Since an accepting run ρ of A_i satisfies $\inf(\rho) \cap G_i = \emptyset$, from some point on the run only visits states of $Q_i \setminus G_i$. So ρ consists of an initial *finite* part, say ρ_0, that may visit all states, and an infinite part, say ρ_1, that only visits states of $Q \setminus G_i$. Again, we take two copies of A_i. Intuitively, A'_i simulates ρ by executing ρ_0 in the first copy, and ρ_1 in the second. The condition that ρ_1 must visit some state of F_i infinitely often is enforced by taking F_i as Büchi condition.