DBAs are less expressive than NBAs

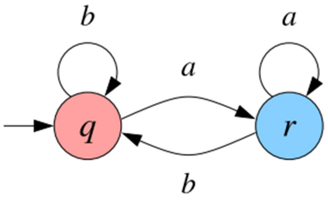
- Prop.: The ω-language $(a+b)^* b^\omega$ is not recognized by any DBA.
- Proof: By contradiction. Assume some DBA recognizes $(a+b)^* b^\omega$.
  - DBA accepts b^ω → DFA accepts b^{n_0}

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

    DBA accepts $b^{n_0} a\, b^{n_1}\, a b^\omega$ → DFA accepts $b^{n_0} a\, b^{n_1} a\, b^{n_2}$ etc.
  - By determinism, the DBA accepts $b^{n_0} a\, b^{n_1} a\, b^{n_2} \ldots a\, b^{n_i} \ldots$, 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

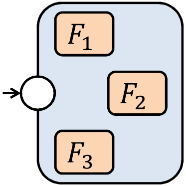
is equivalent to

F_1 is eventually visited

and

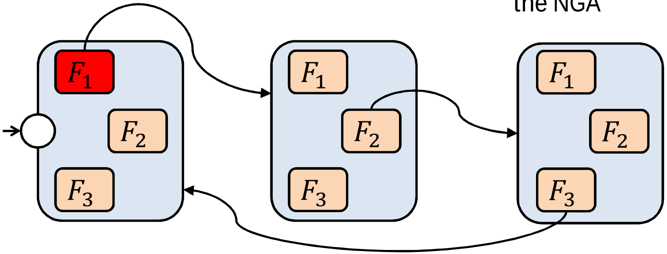
every visit to F_i is eventually followed by a visit to $F_{i \oplus 1}$

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 $\mathcal{F} = \{F_1, \ldots, F_m\}$

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

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

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

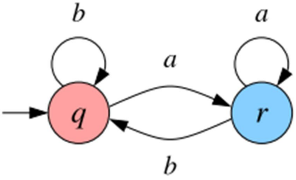
            add $([q, i], a, [q', i])$ **to** δ'

        else /* $q \in F_i$ */

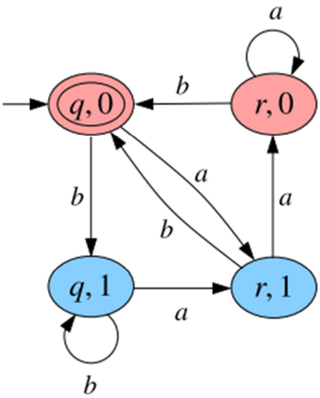
            if $[q', i \oplus 1] \notin Q'$ **then add** $[q', i \oplus 1]$ **to** W

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

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



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



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

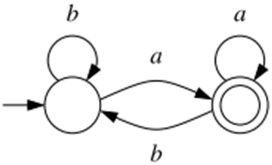
- Question: Are there other classes of omega-automata with
  - the same expressive power as NBAs or NGAs, and
  - with equivalent deterministic and nondeterministic versions?

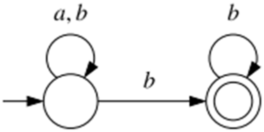
We are only willing to change the acceptance condition!

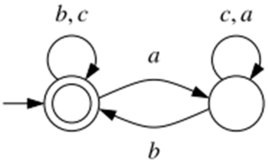
Co-Büchi automata

- 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.

Which are the languages?



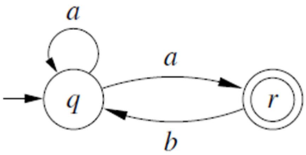


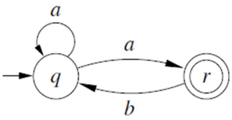


Determinizing co-Büchi automata

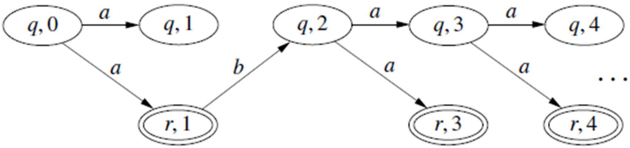
- Given a NCA A we construct a DCA B such that $L(A) = L(B)$.
- We proceed in three steps:
  - We assign to every ω-word w a directed acyclic graph $dag(w)$ that ``contains´´ all runs of A on w.
  - We prove that w is accepted by A iff $dag(w)$ is infinite but contains only finitely many breakpoints.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

- Running example:

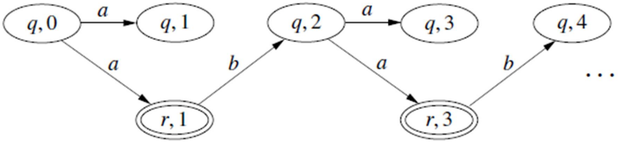




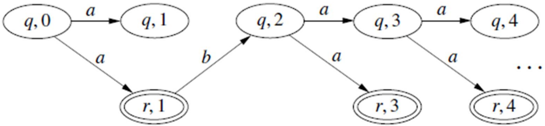
$dag(aba^\omega)$

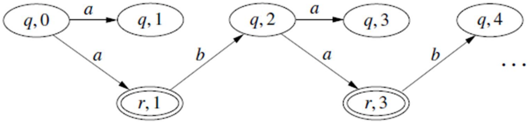


$dag((ab)^\omega)$

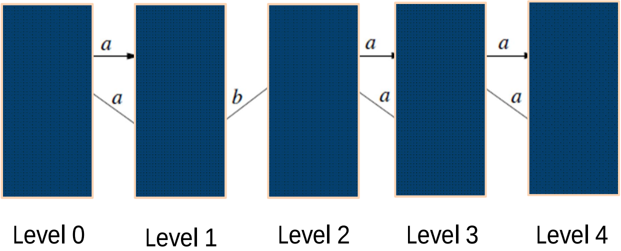


- A accepts w iff some infinite path of $dag(w)$ only visits accepting states finitely often





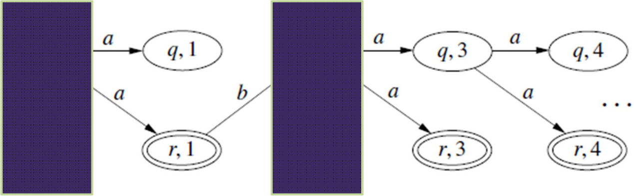
Levels of a dag

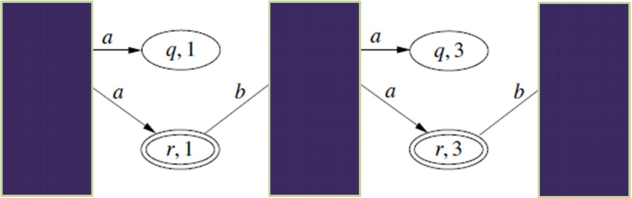


Breakpoints of a dag

- 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 between l and 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 A 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.

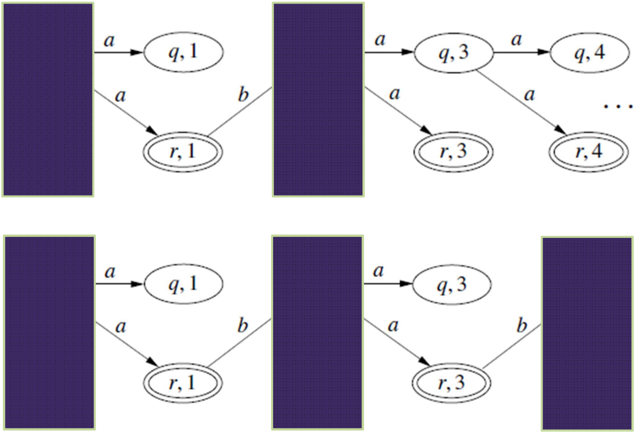
If $dag(w)$ is infinite, then it has an infinite path, and so A has at least one run on w. Since $dag(w)$ has finitely many breakpoints, then every infinite path visits accepting states only finitely often.

Constructing the DCA

- If we could tell if a level is a breakpoint by looking at it, we could take the set of breakpoints as states of the DCA.
- However, we also need some information about its ``history´´.
- Solution: add that information to the level!
- 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 accepting states‘‘. Formally: $q \in O$ if q is the

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 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.


a
q, 1
a
r, 1
b
a
q, 3
a
q, 4
a
r, 3
a
r, 4
...
a
q, 1
a
r, 1
b
a
q, 3
a
r, 3
b

Constructing the DCA

- States: pairs $[P, O]$
- Initial state: pair $[\{q_0\}, \emptyset]$ if $q_0 \in F$, and $[\{q_0\}, \{q_0\}]$ otherwise.
- 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)

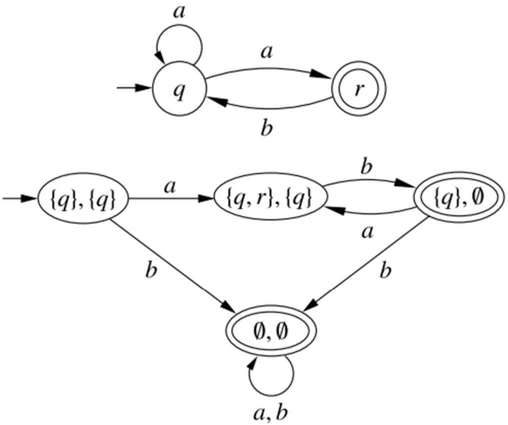
Input: NCA $A = (Q, \Sigma, \delta, q_0, F)$

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

$\tilde{Q}, \tilde{\delta}, \tilde{F} \leftarrow \emptyset$; **if** $q_0 \in F$ **then** $\tilde{q}_0 \leftarrow [q_0, \emptyset]$ **else** $\tilde{q}_0 \leftarrow [\{q_0\}, \{q_0\}]$
$W \leftarrow \{ \tilde{q}_0 \}$
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 ...