ω-Automata, Büchi and Generalized Büchi Automata

AmirReza Azari
Spring 2024 — Sharif University of Technology

Omega Languages

ω-Automata

Definition:

- 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-Regular Languages

Definition:

- Infinite words over the alphabet Σ are infinite sequences $A_0, A_1, A_2, \ldots$ of symbols $A_i \in \Sigma$.
- Σ^ω denotes the set of all infinite words over Σ.
- Any subset of Σ^ω is called a language of infinite words, called an ω-language.
- For instance, the infinite repetition of the finite word AB yields the infinite word $ABABABABAB \ldots$ (ad infinitum) and is denoted by $(AB)^\omega$.
- For the special case of the empty word, we have $\varepsilon^\omega = \varepsilon$.
- For an infinite word, infinite repetition has no effect, that is $\sigma^\omega = \sigma$ if $\sigma \in \Sigma^\omega$.

ω-Regular Expression

Definition:

- An ω-regular expression G over the alphabet Σ has the form:

$$G = E_1.F_1^{\omega} + \ldots + E_n.F_n^{\omega}$$

where $n \geq 1$ and $E_1, \ldots, E_n, F_1, \ldots, F_n$ are regular expressions over Σ such that $\varepsilon \notin L(F_i)$, for all $1 \leq i \leq n$.
- If $L(E) \subseteq \Sigma^*$ denotes the language (of finite words) induced by the regular expression E:

$$L_{\omega}(G) = L(E_1).L(F_1)^{\omega} \ldots L(E_n).L(F_n)^{\omega}$$

ω-Regular Expression

Definition (cont.)

- Example for ω-regular expressions over the alphabet $\Sigma = \{A, B, C\}$:

$$(A + B)^* A (AAB + C)^\omega$$

or

$$A(B + C)^* A^\omega + B(A + C)^\omega$$

ω-Regular Expression

Example:

1. A word in $aa\Sigma^*aa$ followed by only b → $aa\Sigma^*aa.b^\omega$

$$\{aaaabbbb\ldots\},\ \{aabbbbaaabbbb\ldots\}$$

2. Infinite words where b occurs only finitely often → $(a+b)^*.b^\omega$

$$\{aaaa\ldots\},\ \{babbaaaa\ldots\}$$

3. Infinite words where b occurs infinitely often → $(a^*b)^\omega$

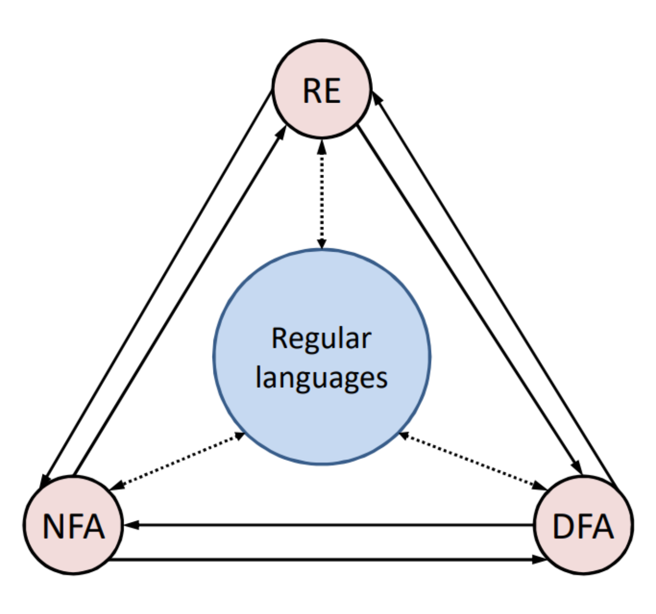
$$\{abababab\ldots\},\ \{bbbabbbabbbabbba\ldots\},\ \{bbbbbbbbbbbbb\ldots\}$$

ω-Regular Expression

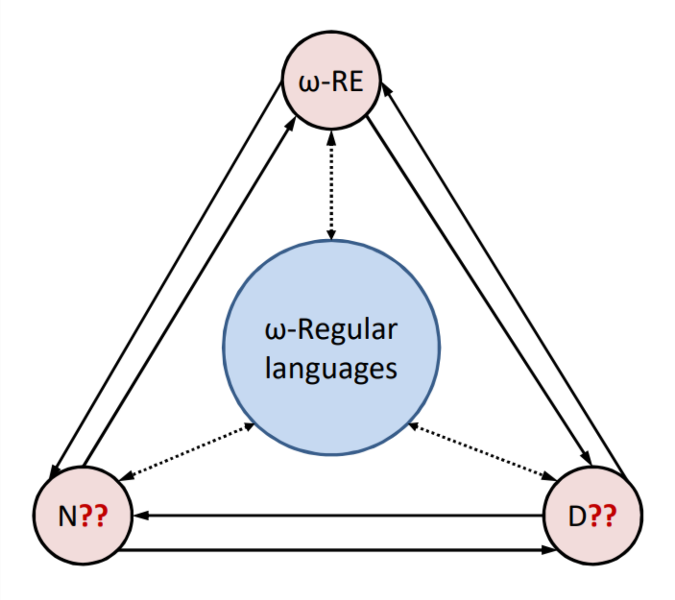
More Examples:

1. $(a + b)^\omega$ set of all infinite words.
2. $a(a + b)^\omega$ infinite words starting with an a
3. $(a + bc + c)^\omega$ words where every b is immediately followed by c
4. $((a + b)^* c)^\omega$ words where c occurs infinitely often
5. $(a + b)^* c(a + b)^\omega$ words with a single occurrence of c

Regular Languages



ω-Regular Languages

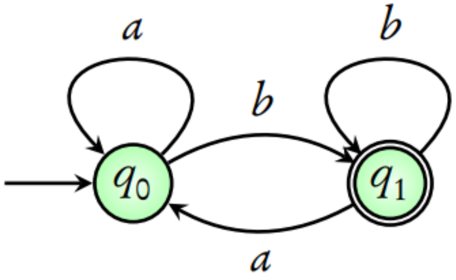


Büchi Automata

Run and acceptance

$$a\ b\ a\ b\ a\ a\ b\ b\ b\ b\ b\ b\ \ldots$$

$$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{b} q_1 \xrightarrow{b} q_1 \xrightarrow{b} q_1 \cdots$$

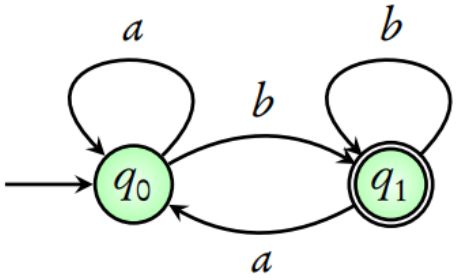


Run and acceptance

Run is accepting if some accepting state occurs infinitely often

Below word is not accepted by this automaton.

$$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \cdots$$



Non – deterministic Büchi Automata

Definition. Non-deterministic B*ü*chi Automata (NBA)

A Non-deterministic B*ü*chi automaton A is a tuple $A = (Q, \Sigma, \delta, Q_0, F)$ where:

- Q is a finite states,
- Σ is an alphabet,
- δ: $Q \times \Sigma \to 2^Q$ is a transition function,
- $Q_0 \subseteq Q$ is a set of initial states
- $F \subseteq Q$ is a set of accept states, called acceptance set.

A language $L \subseteq \Sigma^\omega$ is ω-regular if it can be accepted by some B*ü*chi automaton.

Non – deterministic Büchi Automata

Definition (cont.)

A run for $\sigma = A_0 A_1 A_2 \ldots \in \Sigma^\omega$ denotes an infinite sequence $q_0\, q_1\, q_2 \ldots$ of states in $\mathcal{A}$ such that $q_0 \in Q_0$ and $q_i \xrightarrow{A_i} q_{i+1}$ for $i \geqslant 0$. Run $q_0\, q_1\, q_2 \ldots$ is *accepting* if $q_i \in F$ for infinitely many indices $i \in \mathbb{N}$. The *accepted language* of $\mathcal{A}$ is

$$\mathcal{L}_\omega(\mathcal{A}) \;=\; \{\sigma \in \Sigma^\omega \mid \text{there exists an accepting run for } \sigma \text{ in } \mathcal{A}\}.$$

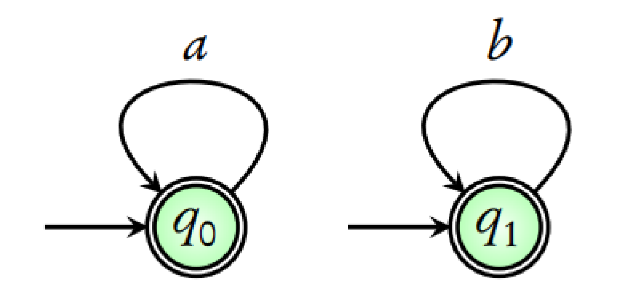
NFA vs. NBA

- Syntax differences between NFA and NBA : None
- Semantics differences between NFA and NBA: the accepted language of an NFA A is a language of finite words, whereas the accepted language of NBA A is an ω-language.
  The intuitive meaning of the acceptance criterion named after Buchi is that the accept set of A has to be visited infinitely often. Thus, the accepted language Lω(A) consists of all infinite words that have a run in which some accept state is visited infinitely often.

NBA

Example

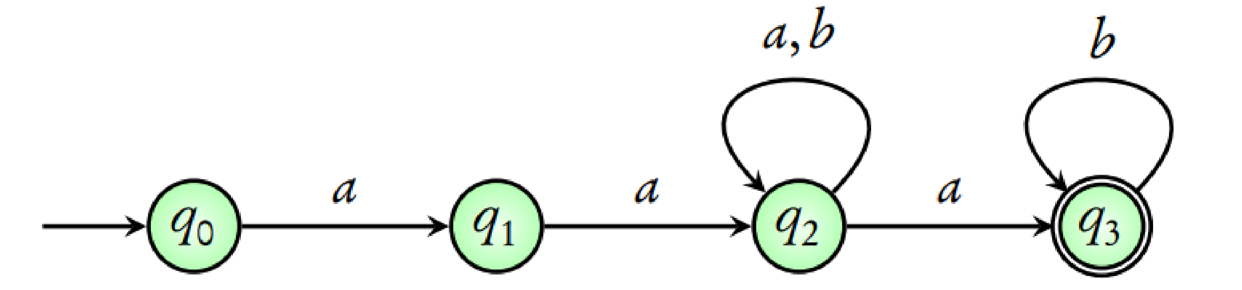
$a^\omega + b^\omega$:



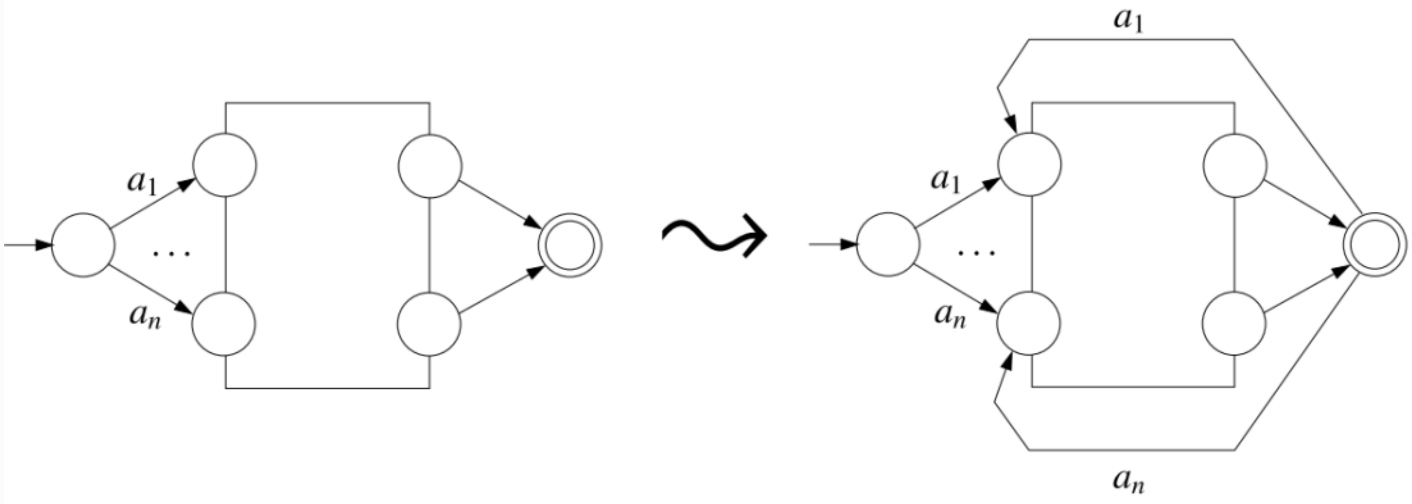
NBA

Example

$aa(a+b)^{*}ab^{\omega}$:



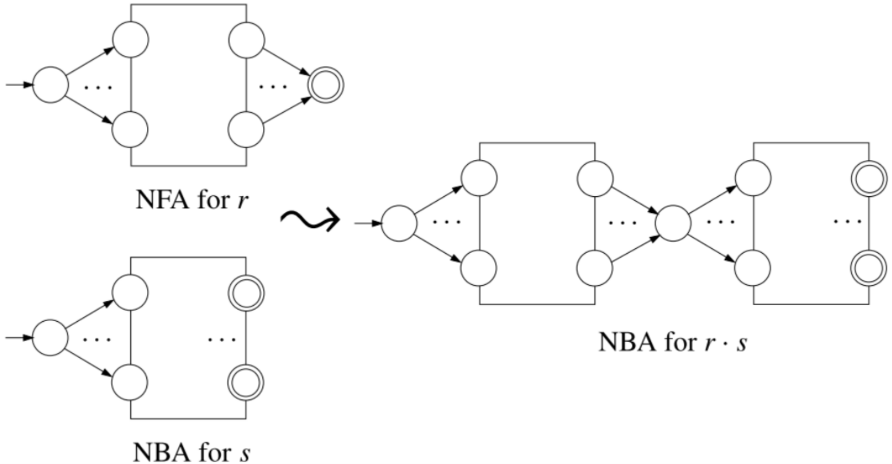
NBA - ω-operator



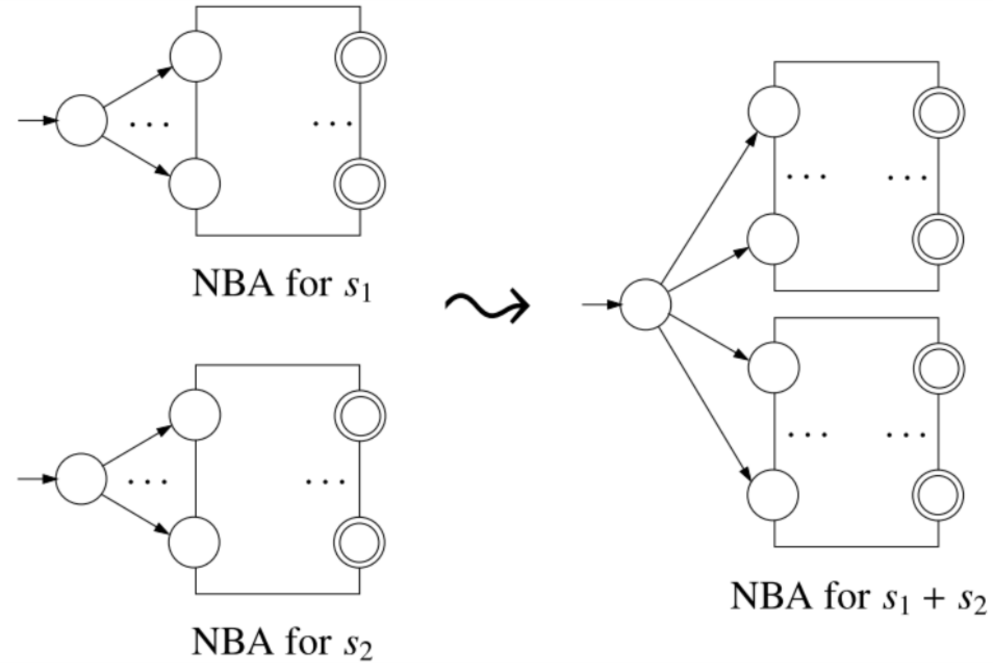
NFA for r NBA for r^ω

NBA - Concatenation

A language L_1 can be concatenated with an ω-language L_2 to yield the ω-language L_1L_2, but two ω-languages cannot be concatenated.



NBA - Union



NBA to ω-regular expression

Lemma.

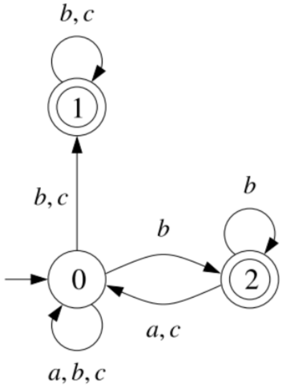
Let A be a NFA, and let $q,\ q'$ be states of A.
The language $L_q^{q'}$ of words with runs leading from q to q' and visiting q' exactly once after leaving q is regular.

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

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

NBA to ω-regular expression

Example:

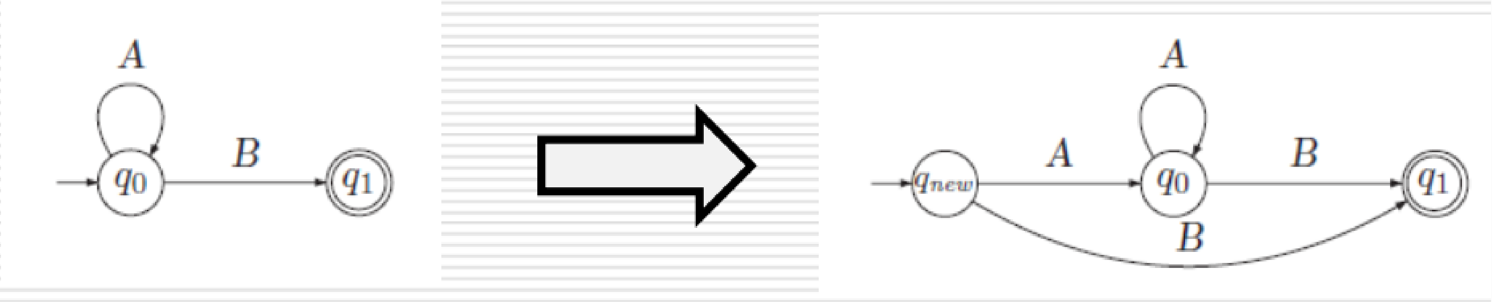


$$
\begin{aligned}
r_0^1 &= (a+b+c)^*(b+c) \\
r_0^2 &= (a+b+c)^*b \\
r_1^1 &= (b+c) \\
r_2^2 &= b+(a+c)(a+b+c)^*b
\end{aligned}
$$

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

Constructing a NBA from a NFA

Add a new initial (nonaccept) state q_{new} to Q with the transitions $q_{new} \xrightarrow{A} q$ if and only if $q_0 \xrightarrow{A} q$ for some initial state $q_0 \in Q_0$. All other transitions, as well as the accept states, remain unchanged.



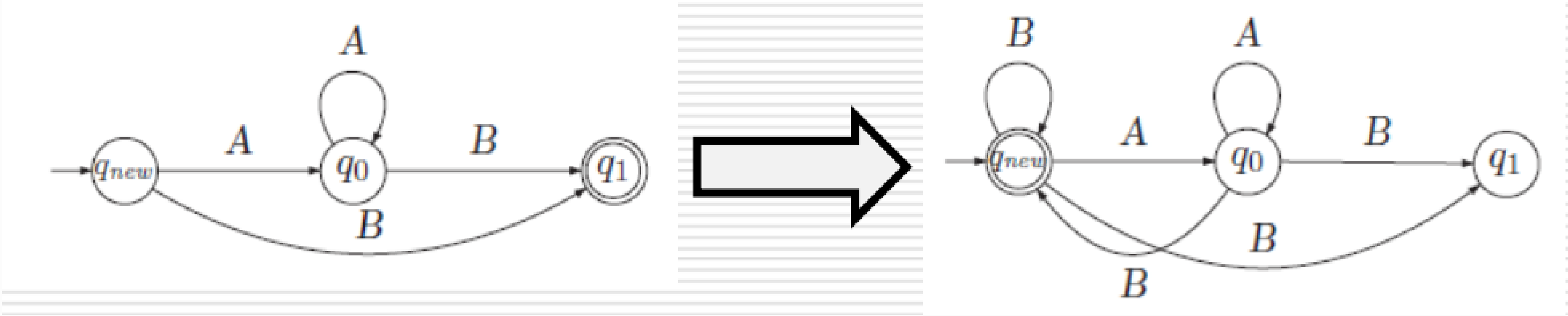
Constructing a NBA from a NFA

In the sequel, we assume that $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ is an NFA such that the states in Q_0 do not have any incoming transitions and $Q_0 \cap F = \varnothing$. We now construct an NBA $\mathcal{A}' = (Q, \Sigma, \delta', Q'_0, F')$ with $\mathcal{L}_\omega(\mathcal{A}') = \mathcal{L}(\mathcal{A})^\omega$. The basic idea of the construction of $\mathcal{A}'$ is to add for any transition in $\mathcal{A}$ that leads to an accept state new transitions leading to the initial states of $\mathcal{A}$. Formally, the transition relation δ' in the NBA $\mathcal{A}'$ is given by

$$\delta'(q, A) = \begin{cases} \delta(q, A) & \text{if } \delta(q, A) \cap F = \varnothing \\ \delta(q, A) \cup Q_0 & \text{otherwise.} \end{cases}$$

The initial states in the NBA $\mathcal{A}'$ agree with the initial states in $\mathcal{A}$, i.e., $Q'_0 = Q_0$. These are also the accept states in $\mathcal{A}'$, i.e., $F' = Q_0$.

Constructing a NBA from a NFA

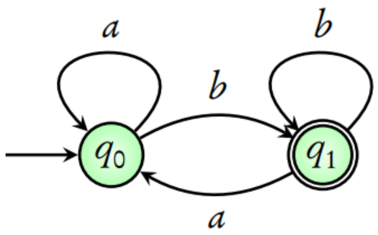


Deterministic Büchi Automata

Definition:

- Single initial state
- From every state - on an alphabet, there is a unique transition

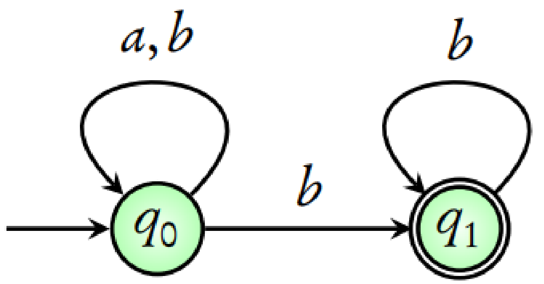
Words where b occurs infinitely often



DBA

Question: Can every NBA be converted to an equivalent DBA?

$(a+b)^*b^\omega$: a occurs only finitely often



DBA

DBA less powerful than NBA

- Automaton has to guess the point from where only b occurs
- A deterministic Büchi automaton cannot make this guess
- The above language cannot be accepted by a DBA.

Proof.

By contradiction. Assume some DBA recognizes $(a + b)^* b^\omega$.

DBA

Proof (cont.)

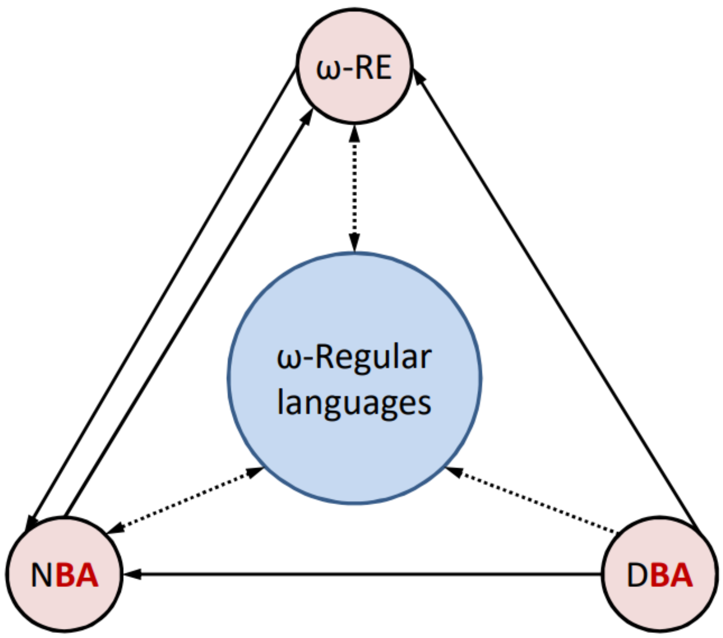
– DBA accepts b^{ω} → DFA accepts b^{n_0}
DBA accepts $b^{n_0} a\, b^{\omega}$ → DFA accepts $b^{n_0} a\, b^{n_1}$
DBA accepts $b^{n_0} a\, b^{n_1}\, ab^{\omega}$ → DFA accepts $b^{n_0} a\, b^{n_1} a\, b^{n_2}$ etc.

– By determinism and finite number of states, the DBA accepts

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

for some $i < j$. This word does not belong to $(a + b)^* b^{\omega}$.

ω-Regular languages



Generalized Büchi Automata

GBA

Definition.

- Generalized Büchi automaton (GBA) is a variant of Büchi automaton
- The difference with the Büchi automaton is its accepting condition, i.e., a set of sets of states.
- A run is accepted by the automaton if it visits at least one state of every set of the accepting condition infinitely often.
- Generalized Büchi automata (GBA) is equivalent in expressive power with Büchi automata

GBA

Definition (cont.)

- A generalized Buchi automaton (GBA) over Σ is:

$$A = (Q, \Sigma, \delta, I, F)$$

- Q is a finite set of states
- $\Sigma = \{a, b, \ldots\}$ is a finite alphabet set of A
- $\delta \subseteq Q \times \Sigma \times Q$ is a transition relation
- $I \subseteq Q$ is a set of initial states
- $F = \{F_1, \ldots, F_k\} \subseteq 2^Q$ is a set of sets of final states.

GBA

Definition (cont.)

- A accepts exactly those runs in which the set of infinitely often occurring states contains at least a state from each $F_1, \ldots, F_n$.
- A run σ of a GBA is said to e accepting iff,

$$\text{for all } 1 \leq i \leq k, \text{ we have } \mathrm{inf}(\sigma) \cap F_i \neq \emptyset.$$

inf(σ)

The set of states visited infinitely often by a run σ is denoted inf(σ).

GBA

(inf(σ) (cont.))

- $\rho_1 = q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_1 \xrightarrow{a} q_1 \xrightarrow{a} q_1 \cdots$ $\quad \mathrm{inf}(\rho_1) = \{q_1\}$
- $\rho_2 = q_0 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \xrightarrow{b} q_0 \cdots$ $\quad \mathrm{inf}(\rho_2) = \{q_0\}$
- $\rho_3 = q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \cdots$ $\quad \mathrm{inf}(\rho_3) = \{q_0, q_1\}$

NBA & GNBA

Nondeterministic Büchi Automaton

$A = (Q, \Sigma, \delta, I, F)$, where $F \subseteq Q$ is the set of accepting states.

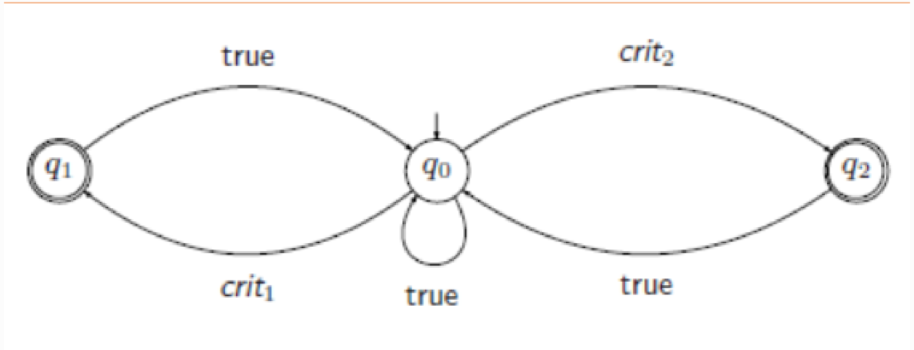
- A run ρ of A on omega word α is an infinite sequence $\rho = q_o, q_1, q_2, \ldots$ s.t. $q_0 \in I$ and $q_i \xrightarrow{a_i} q_{i+1}$ for $0 \leq i$.
- The run ρ is accepting if
$$Inf(\rho) \cap F \neq \emptyset.$$
- The language accepted by A
$\mathcal{L}(A) = \{\alpha \in \Sigma^\omega \mid A \text{ has an accepting run on } \alpha\}$

A Generalized Büchi Automaton is $A := (Q, \Sigma, \delta, I, FT)$ where $FT = < F_1, F_2, \ldots, F_k >$ with $F_i \subseteq Q$.

A run ρ of A is accepting if $Inf(\rho) \cap F_i \neq \emptyset$ for each $1 \leq i \leq k$.

NBA & GNBA

A GNBA for the property ”both processes are infinitely often in their critical section”:

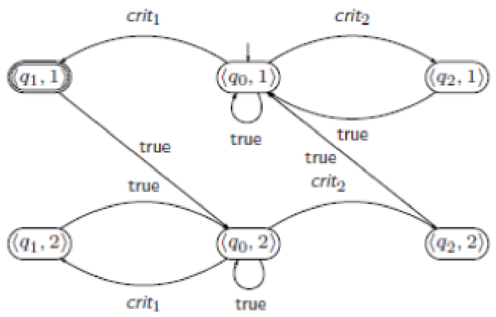


- GNBA are like NBA, but have a distinct acceptance criterio. A GNBA requires to visit several sets $F_1, \ldots, F_k (k \geq 0)$ infinitely often.

De-generalization of GBA

Sketch of transformation GNBA (with $|\mathcal{F}| = k$) into equivalent NBA:

- make k copies of the GNBA
- initial states of NBA := the initial states in the first copy
- final states of NBA := accept set F_1 in the first copy
- on visiting in i-th copy a state in F_i, then move to the $(i+1)$-st copy



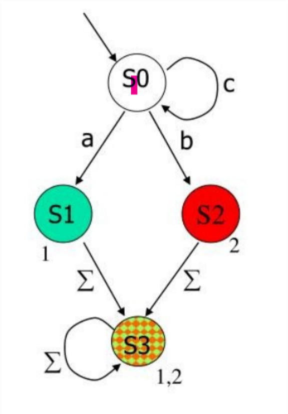
De-generalization of GBA

Algorithm:

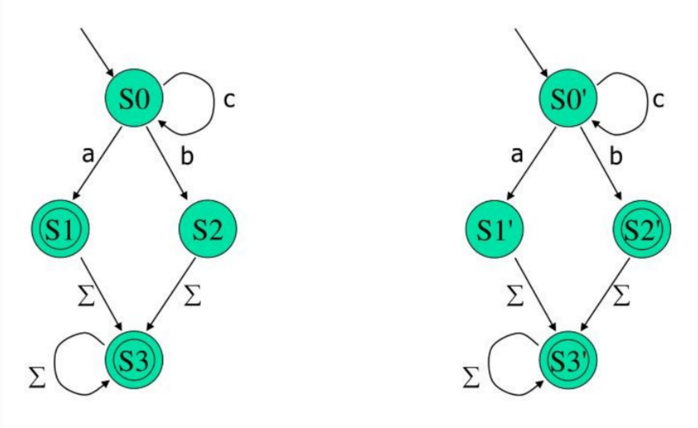
- Turn a generalized Büchi automaton into a Büchi automaton.
- The idea:
  - Each cycle must go through every copy.
  - Each cycle must contain accepting states from each accepting set.
- Algorithm:
  - Duplicate the GBA to as many copies as the number of accepting sets
  - Redirect outgoing edges from accepting states to the next copy.

De-generalization of GBA

Example:



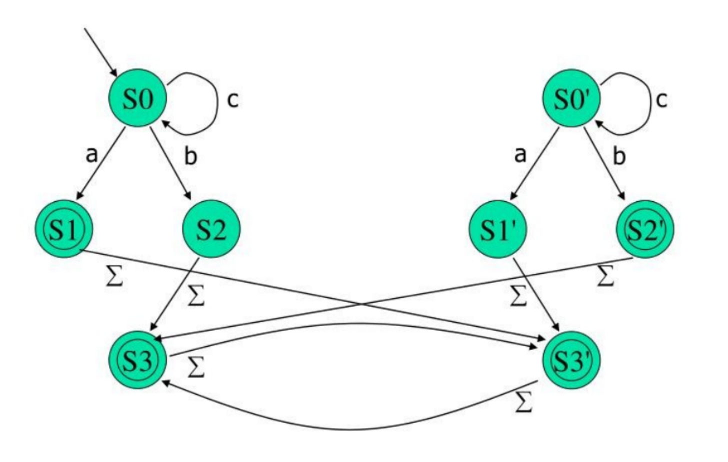
1,2 correspond to F_1 and F_2, the accepting sets



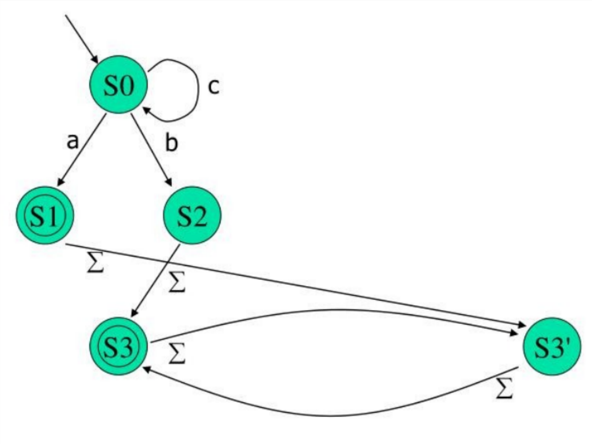
Two copies, because we have two accepting sets

De-generalization of GBA

Example (cont.):



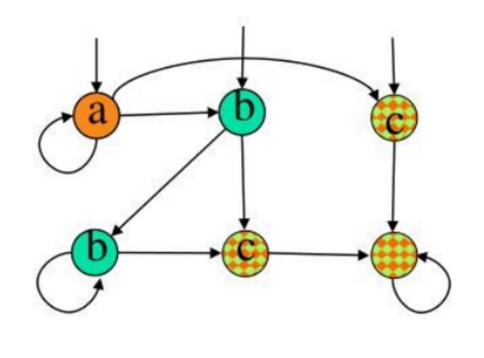
Choose one cope as initial and redirect edges from accepting edges



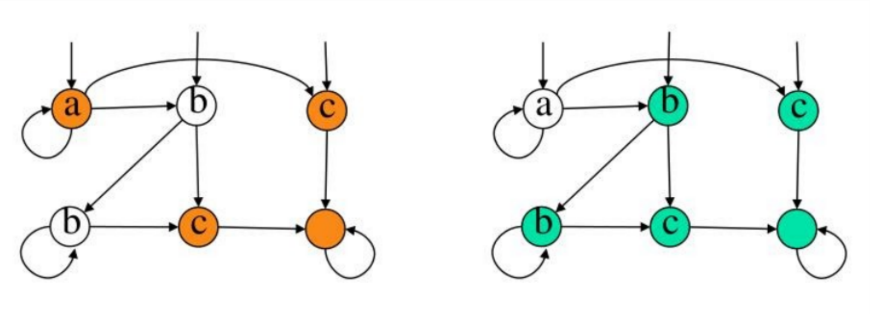
remove unreachable states

De-generalization of GBA

Another Example:



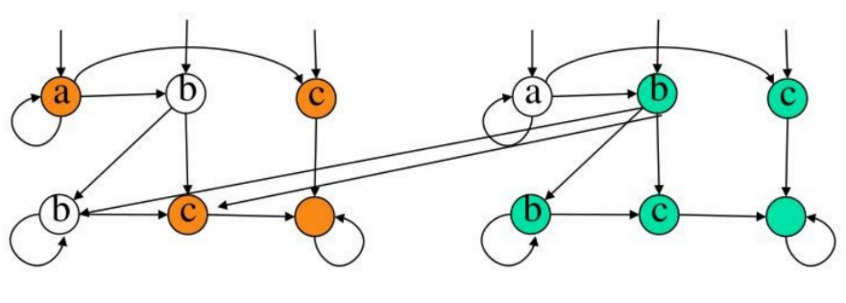
A GBA



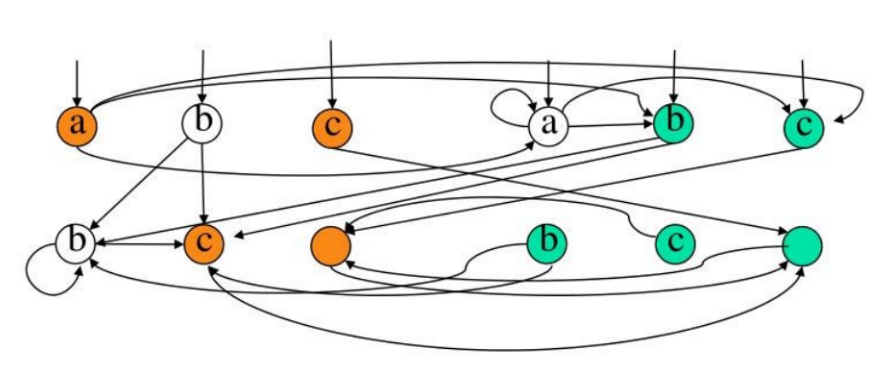
One copy for each accepting set

De-generalization of GBA

Another Example (cont.):



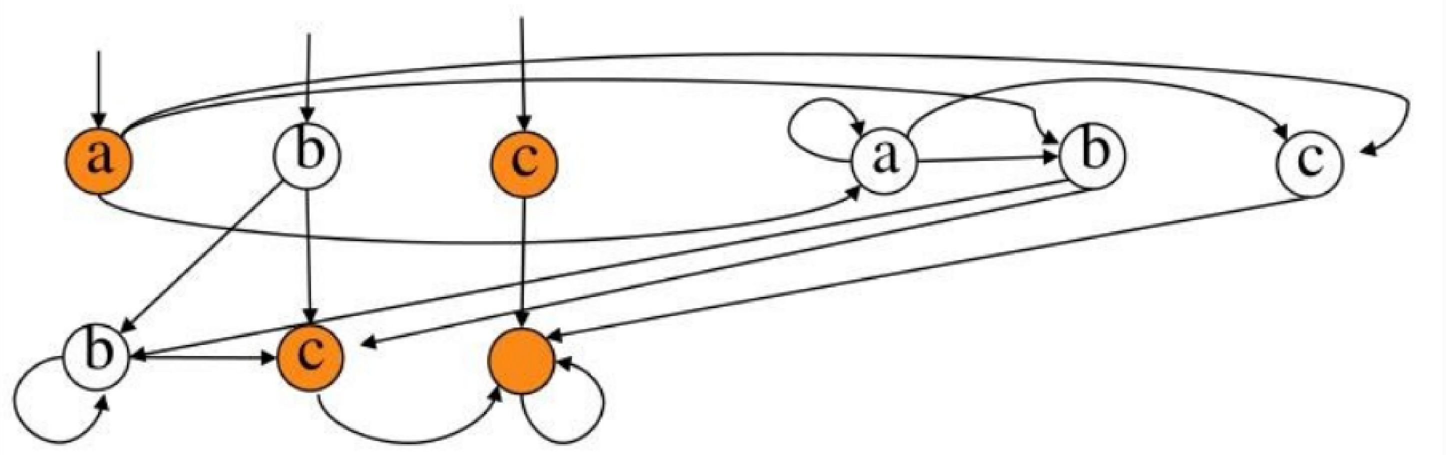
Redirecting edges



and so forth...

De-generalization of GBA

Another Example (cont.):

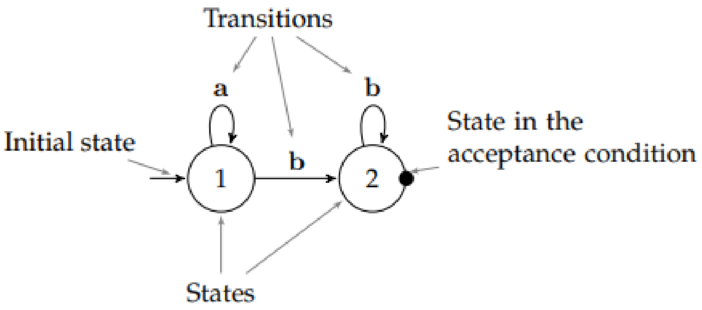


Review

ω-automata

An ω-automaton is a quintuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ with:

- Q a finite set of states.
- Σ the alphabet.
- $\delta : Q \times \Sigma \to 2^Q$ the transition function.
- $q_0 \in Q$ the initial state.
- F the acceptance condition, which is a formula on states. The differ on finite words lies in these acceptance conditions.



An ω-automaton

Büchi Automata

Büchi acceptance condition

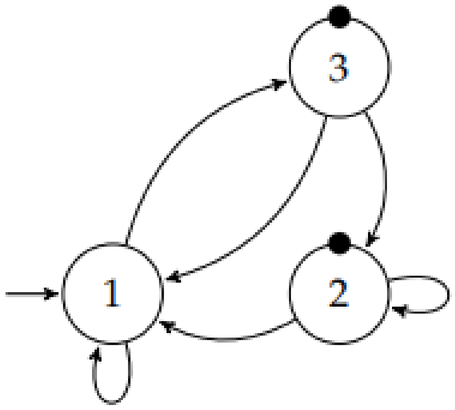
Büchi (1962) was the first to introduce ω-automata with his acceptance condition. The Büchi acceptance condition is the most adapted to model checking since it supports all the operations presented in Section 1.2: Operations on ω-automata.

With his definition, the acceptance condition F is a set of states, and a run must visit infinitely often some states from F to be accepting.

More formally, a run π of a Büchi automaton with $F \subseteq Q$ as acceptance condition is accepting, iff $inf(\pi) \cap F \neq \emptyset$.

Büchi Automata

Example. Figure 1.3 presents a Büchi automaton with its states in the acceptance condition F marked with ●. A run of this automaton is accepting if it visits infinitely often states ② OR ③.



A Büchi automaton.

Generalized Büchi Automata

Generalized Büchi acceptance condition

Generalized Büchi automata are a variant of Büchi automata that is more succint, since it allows to have automata that recognize the same language than Büchi automata but with a smaller number of states and transitions.

The Generalized Büchi acceptance condition has more than one set of acceptance conditions. A run is accepting if it passes through at least one state of each set infinitely often. Figure 1.4 illustrates this acceptance condition.

More formally, the definition is $\forall i \mid inf(\pi) \cap F_i \neq \emptyset$ with $F = \{F_1, F_2, \cdots, F_n\}$ and $F_i \subseteq Q$.

Generalized Büchi Automata

Example. Figure 1.4 presents a generalized Büchi automaton with an accepting run if a run visits infinitely often both acceptance conditions (states denoted with ● and ○).

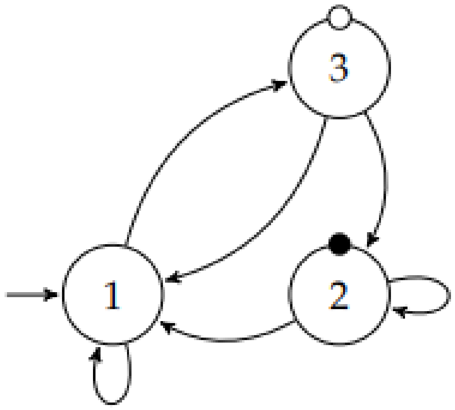


Figure 1.4: A generalized Büchi automaton.

References

References

- https://www.cmi.ac.in/ kumar/words/lecture07.pdf
- https://www.cs.colostate.edu/ france/CS614/Slides/ModelCheckingChapter4.pdf
- https://www.youtube.com/watch?v=KOu6lUssxbs
- https://www.lrde.epita.fr/ sadegh/buchi-complementation-techrep.pdf
- https://www.irif.fr/ jep/PDF/InfiniteWords/Chapter1.pdf

دانشگاه صنعتی شریف