Büchi Automata

Lecture #9 of Model Checking

Joost-Pieter Katoen

Lehrstuhl 2: Software Modeling & Verification

E-mail: katoen@cs.rwth-aachen.de

November 18, 2008

Overview Lecture #11

$\Rightarrow$ Motivation: Peterson's Algorithm

- ω-Regular Languages
- Nondeterministic Büchi Automata (NBA)
- NBA and ω-Regular Languages

Recap: Verifying regular safety properties

- A safety property P is regular if $bp(P)$ is a regular language
- Thus, bad prefixes of P are recognizable by an NFA $\mathcal{A}$
- Checking $TS \models P$ for regular P reduces to
  - checking a (simple) invariant on the product automaton $TS \otimes \mathcal{A}$
  - i.e., performing a simple depth-first search on $TS \otimes \mathcal{A}$
- Time and space complexity in $\mathcal{O}(|TS| \cdot |\mathcal{A}|)$

for more general properties we need automata accepting infinite words!

Peterson's banking system

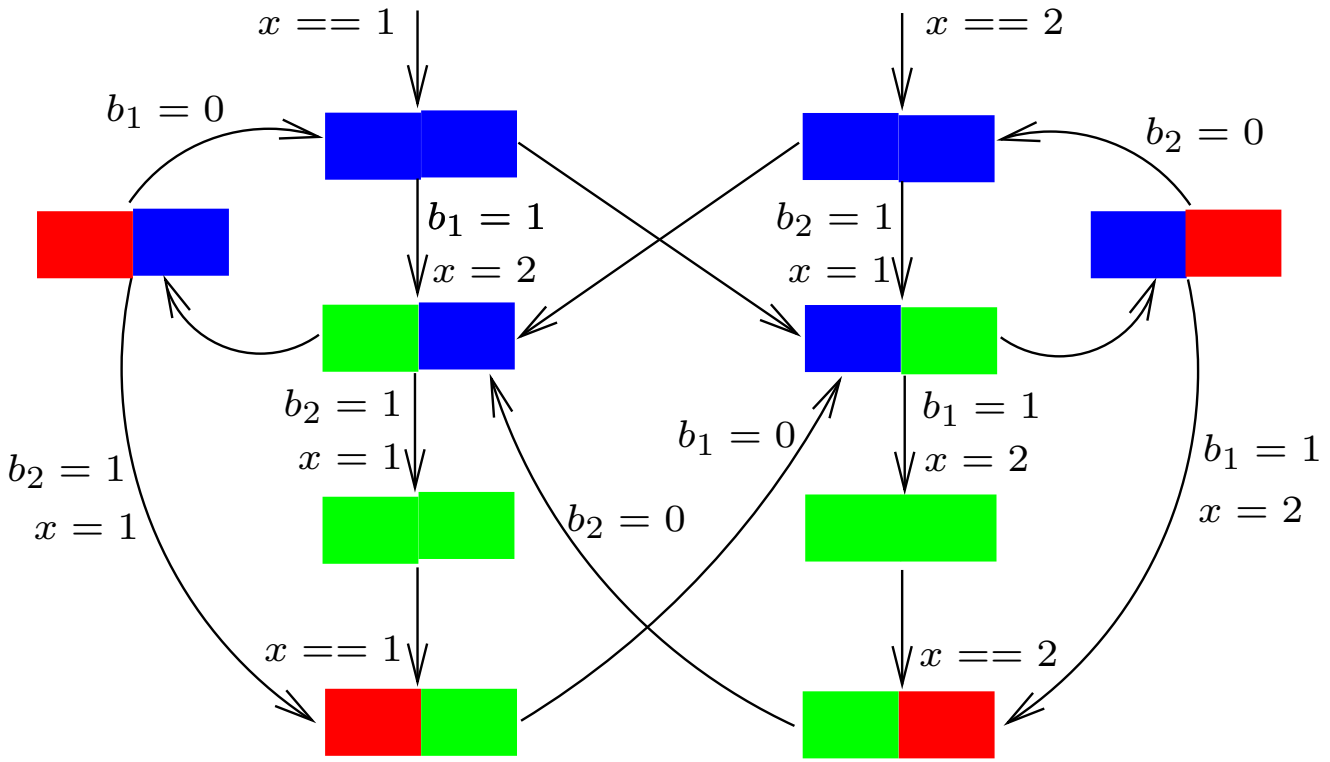
Person Left behaves as follows:

```
          while true {
              ......
rq :          b1, x = true, 2;
wt :          wait until(x == 1 || ¬ b2) {
cs :              ... @account_L ...}
              b1 = false;
              ......
          }
```

Person Right behaves as follows:

```
          while true {
              ......
rq :          b2, x = true, 1;
wt :          wait until(x == 2 || ¬ b1) {
cs :              ... @account_R ...}
              b2 = false;
              ......
          }
```

Is the banking system live?

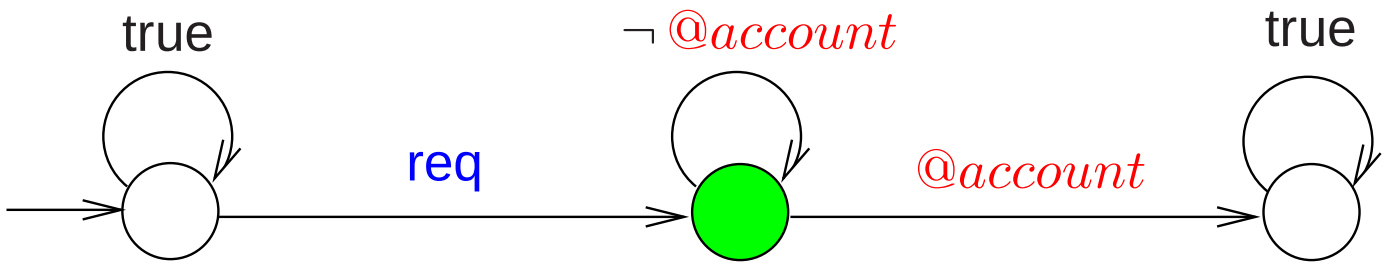


If someone wants to update the account, does he ever get the opportunity to do so?

"always ($req_L \Rightarrow$ eventually @account$_L$) $\wedge$ always ($req_R \Rightarrow$ eventually @account$_R$)"

Is the banking system live (revisited)?

- Live = when you want access to account, you eventually get it
- Unlive: once you want access to the account, you never get it
  - unlive behaviour can be characterized as a (set of) infinite traces
  - or, equivalently, by a Büchi-automaton *Live*:



- Checking liveness: $\mathit{Traces}(\mathit{TS}_{Pet}) \cap L_\omega(\overline{\mathit{Live}}) = \varnothing$?
  - (explicit) complementation, intersection and emptiness of Büchi automata!

Overview Lecture #11

- Motivation: Peterson's Algorithm

$\Rightarrow$ ω-Regular Languages

- Nondeterministic Büchi Automata (NBA)
- NBA and ω-Regular Languages

Regular expressions

- Let Σ be an alphabet with $A \in \Sigma$
- Regular expressions over Σ have *syntax*:

$$\mathsf{E} ::= \underline{\varnothing} \mid \underline{\varepsilon} \mid \underline{A} \mid \mathsf{E} + \mathsf{E}' \mid \mathsf{E}.\mathsf{E}' \mid \mathsf{E}^*$$

- The *semantics* of regular expression E is a language $\mathcal{L}(\mathsf{E}) \subseteq \Sigma^*$:

$$\mathcal{L}(\underline{\varnothing}) = \varnothing, \quad \mathcal{L}(\underline{\varepsilon}) = \{\varepsilon\}, \quad \mathcal{L}(\underline{A}) = \{A\}$$

$$\mathcal{L}(\mathsf{E}+\mathsf{E}') = \mathcal{L}(\mathsf{E}) \cup \mathcal{L}(\mathsf{E}') \quad \mathcal{L}(\mathsf{E}.\mathsf{E}') = \mathcal{L}(\mathsf{E}).\mathcal{L}(\mathsf{E}') \quad \mathcal{L}(\mathsf{E}^*) = \mathcal{L}(\mathsf{E})^*$$

Syntax of ω-regular expressions

- Regular expressions denote languages of finite words
- ω-Regular expressions denote languages of infinite words
- An ω-*regular expression* G over Σ has the form:

$$\mathsf{G} \;=\; \mathsf{E}_1.\mathsf{F}_1^{\omega} \;+\; \ldots \;+\; \mathsf{E}_n.\mathsf{F}_n^{\omega} \quad \text{for } n > 0$$

where $\mathsf{E}_i, \mathsf{F}_i$ are regular expressions over Σ with $\varepsilon \notin \mathcal{L}(\mathsf{F}_i)$

- Some examples: $(A + B)^*.B^{\omega}$, $(B^*.A)^{\omega}$, and $A^*.B^{\omega} + A^{\omega}$

Semantics of ω-regular expressions

- For $\mathcal{L} \subseteq \Sigma^*$ let $\mathcal{L}^\omega \;=\; \{\, w_1 w_2 w_3 \ldots \mid \forall i \geqslant 0.\, w_i \in \mathcal{L} \,\}$
- Let ω-regular expression $\mathsf{G} \;=\; \mathsf{E}_1.\mathsf{F}_1^\omega \;+\; \ldots \;+\; \mathsf{E}_n.\mathsf{F}_n^\omega$
- The *semantics* of G is a language $\mathcal{L}(\mathsf{G}) \subseteq \Sigma^\omega$:

$$\mathcal{L}_\omega(\mathsf{G}) \;=\; \mathcal{L}(\mathsf{E}_1).\mathcal{L}(\mathsf{F}_1)^\omega \;\cup\; \ldots \;\cup\; \mathcal{L}(\mathsf{E}_n).\mathcal{L}(\mathsf{F}_n)^\omega$$

- G_1 and G_2 are *equivalent*, denoted $\mathsf{G}_1 \equiv \mathsf{G}_2$, if $\mathcal{L}_\omega(\mathsf{G}_1) = \mathcal{L}_\omega(\mathsf{G}_2)$

ω-Regular languages

- $\mathcal{L}$ is ω-regular if $\mathcal{L} = \mathcal{L}_\omega(\mathrm{G})$ for some ω-regular expression G
- Examples over $\Sigma = \{ A, B \}$:
  - language of all words with infinitely many As:

$$(B^*.A)^\omega$$

  - language of all words with finitely many As:

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

  - the empty language

$$\varnothing^\omega$$

- ω-Regular languages are closed under $\cup$, $\cap$, and complementation

ω-Regular safety properties

- Definition:

LT property P over *AP* is ω-regular if

P is an ω-regular language over the alphabet 2^{AP}

- Or, equivalently:

LT property P over *AP* is ω-regular if P is a language accepted by a nondeterministic Büchi automaton over 2^{AP}

Abstract example ω-regular properties

- Any invariant P is an ω-regular property
  - as Φ^ω describes P with invariant condition Φ
- Any regular safety property P is an ω-regular property
  - as $\overline{P} = \mathit{bp}(P).\left(2^{AP}\right)^\omega$ is ω-regular
  - and the fact that ω-regular languages are closed under complement
- Any liveness property P is an ω-regular property

Concrete ω-regular properties

Nondeterministic Büchi automata

- NFA (and DFA) are incapable of accepting infinite words
- Automata on infinite words
  - suited for accepting ω-regular languages
  - we consider nondeterministic Büchi automata (NBA)
- Accepting runs have to "check" the entire input word $\Rightarrow$ are infinite

  $\Rightarrow$ acceptance criteria for infinite runs are needed
- NBA are like NFA, but have a distinct *acceptance criterion*
  - one of the accept states must be visited infinitely often

Büchi automata

A *nondeterministic Büchi automaton* (NBA) $\mathcal{A}$ is a tuple $(Q, \Sigma, \delta, Q_0, F)$ where:

- Q is a finite set of states with $Q_0 \subseteq Q$ a set of initial states
- Σ is an alphabet
- $\delta : Q \times \Sigma \to 2^Q$ is a transition function
- $F \subseteq Q$ is a set of accept (or: final) states

An example NBA

Language of an NBA

- NBA $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ and word $\sigma = A_0 A_1 A_2 \ldots \in \Sigma^\omega$
- A *run* for σ in $\mathcal{A}$ is an infinite sequence $q_0\, q_1\, q_2 \ldots$ such that:
  - $q_0 \in Q_0$ and $q_i \xrightarrow{A_{i+1}} q_{i+1}$ for all $0 \leqslant i$
- Run $q_0\, q_1\, q_2 \ldots$ is *accepting* if $q_i \in F$ for infinitely many i
- $\sigma \in \Sigma^\omega$ is *accepted* by $\mathcal{A}$ if there exists an accepting run for σ
- The *accepted language* of $\mathcal{A}$:

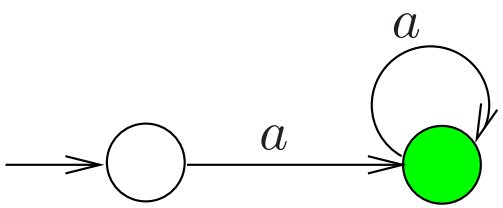
$$\mathcal{L}_\omega(\mathcal{A}) = \left\{ \sigma \in \Sigma^\omega \mid \text{ there exists an accepting run for } \sigma \text{ in } \mathcal{A} \right\}$$

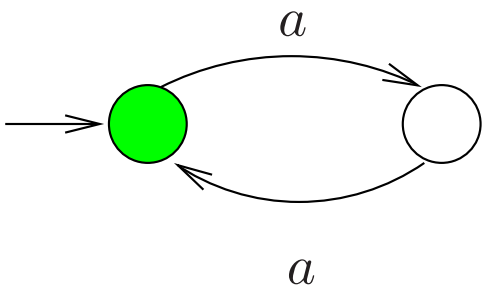
- NBA $\mathcal{A}$ and $\mathcal{A}'$ are *equivalent* if $\mathcal{L}_\omega(\mathcal{A}) = \mathcal{L}_\omega(\mathcal{A}')$

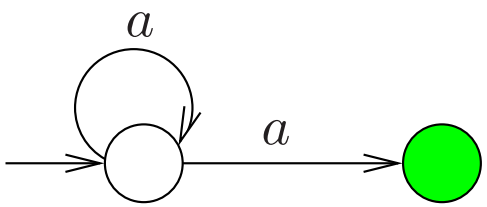
Example runs and accepted words

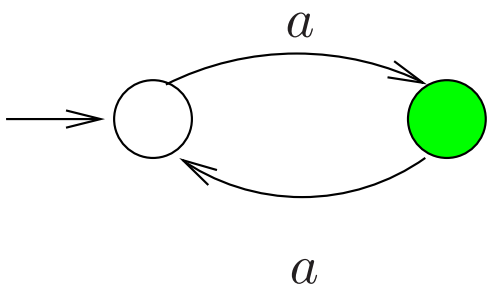
Some NBA for ω-regular properties

NBA versus NFA









finite equivalence $\not\Rightarrow$ ω-equivalence

$\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$, but $\mathcal{L}_\omega(\mathcal{A}) \neq \mathcal{L}_\omega(\mathcal{A}')$

ω-equivalence $\not\Rightarrow$ finite equivalence

$\mathcal{L}_\omega(\mathcal{A}) = \mathcal{L}_\omega(\mathcal{A}')$, but $\mathcal{L}(\mathcal{A}) \neq \mathcal{L}(\mathcal{A}')$

NBA and ω-regular languages

The class of languages accepted by NBA

agrees with the class of ω-regular languages

(1) any ω-regular language is recognized by an NBA

(2) for any NBA $\mathcal{A}$, the language $\mathcal{L}_\omega(\mathcal{A})$ is ω-regular

For any ω-regular language there is an NBA

- How to construct an NBA for the ω-regular expression:

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

where E_i and F_i are regular expressions over alphabet Σ with $\varepsilon \notin F_i$

- Use operators on NBA mimicking operators on ω-regular expressions:

(1) for NBA $\mathcal{A}_1$ and $\mathcal{A}_2$ there is an NBA accepting $\mathcal{L}_\omega(\mathcal{A}_1) \cup \mathcal{L}_\omega(\mathcal{A}_2)$
(2) for any regular language $\mathcal{L}$ with $\varepsilon \notin \mathcal{L}$ there is an NBA accepting $\mathcal{L}^\omega$
(3) for regular language $\mathcal{L}$ and NBA $\mathcal{A}'$ there is an NBA accepting $\mathcal{L}.\mathcal{L}_\omega(\mathcal{A}')$

- We will discuss these three operators in detail

Union of NBA

For NBA $\mathcal{A}_1$ and $\mathcal{A}_2$ (both over the alphabet Σ)

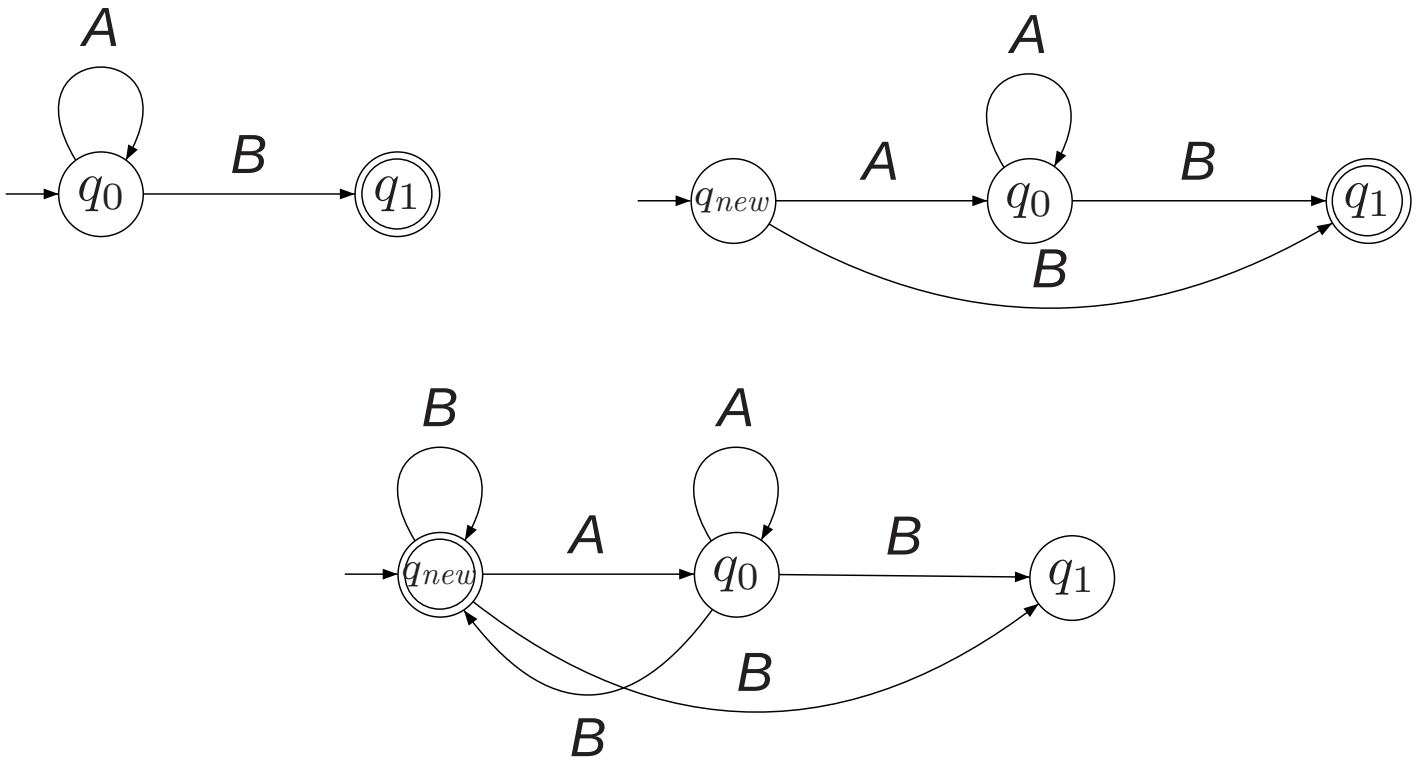
there exists an NBA $\mathcal{A}$ such that:

$$\mathcal{L}_\omega(\mathcal{A}) = \mathcal{L}_\omega(\mathcal{A}_1) \cup \mathcal{L}_\omega(\mathcal{A}_2) \quad \text{and} \quad |\mathcal{A}| = \mathcal{O}(|\mathcal{A}_1| + |\mathcal{A}_2|)$$

Definition of ω-operator for NFA

- Let $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ be an NFA with $\varepsilon \notin \mathcal{L}(\mathcal{A})$.
- Assume no initial state in $\mathcal{A}$ has incoming transitions and $Q_0 \cap F = \varnothing$
  - otherwise introduce a new initial state $q_{new} \notin F$
  - let $q_{new} \xrightarrow{A} q$ iff $q_0 \xrightarrow{A} q$ for some $q_0 \in Q_0$
  - keep all transitions in $\mathcal{A}$
- Construct an NBA $\mathcal{A}' = (Q, \Sigma, \delta', Q_0', F')$ as follows
  - if $q \xrightarrow{A} q' \in F$ then add $q \xrightarrow{A} q_0$ for any $q_0 \in Q_0$
  - keep all transitions in $\mathcal{A}$
  - $Q_0' = Q_0$ and $F' = Q_0$.

Example for ω-operator for NFA



From an NFA accepting A^*B to an NBA accepting $(A^*B)^\omega$

Proof of $\mathcal{L}_\omega(\mathcal{A}') \subseteq \mathcal{L}(\mathcal{A})^\omega$

- Let $\sigma \in \mathcal{L}_\omega(\mathcal{A}')$ and $q_0 q_1 q_2 \ldots$ be an accepting run for σ in $\mathcal{A}'$
  - hence, $q_i \in F' = Q_0$ for infinitely many indices i
  - let $i_0 = 0 < i_1 < i_2 < \ldots$ such that $q_{i_k} \in F'$ and $q_j \notin F'$ for $j \neq i_k$
- Divide σ into infinitely many nonempty finite subwords $w_i \in \Sigma^*$:

$$\sigma = w_1 w_2 w_3 \ldots \quad \text{such that} \quad q_{i_k} \in \delta'^*(q_{i_{k-1}}, w_k) \quad \text{for all } k > 0$$

- It follows $\delta^*(q_{i_{k-1}}, w_k) \cap F \neq \varnothing$
  - $q_{i_k} \in Q_0$ and $q_{i_k} \in Q_0$ has no incoming transitions, thus $q_{i_k} \in F$
- Thus: $w_k \in \mathcal{L}(\mathcal{A})$ for any $k > 0$, and hence $\sigma \in \mathcal{L}(\mathcal{A})^\omega$

Proof of $\mathcal{L}_\omega(\mathcal{A}') \supseteq \mathcal{L}(\mathcal{A})^\omega$

- Let $\sigma = w_1 w_2 w_3 \ldots \in \Sigma^\omega$ such that $w_k \in \mathcal{L}(\mathcal{A})$ for all $k > 0$
  - that, is $\sigma \in \mathcal{L}(\mathcal{A})^\omega$
- Let $q_0^k q_1^k \ldots q_{n_k}^k$ be an accepting run for w_k in $\mathcal{A}$
- By definition of $\mathcal{A}'$, we have $q_0^{k+1} \in \delta'^*(q_0^k, w_k)$ for all $k > 0$

$\Rightarrow$ $q_0^1 \ldots q_{n_1-1}^1 q_0^2 \ldots q_{n_2-1}^2 q_0^3 \ldots q_{n_3-1}^3 \ldots$ is an accepting run for σ in $\mathcal{A}'$

- Hence, $\sigma \in \mathcal{L}_\omega(\mathcal{A}')$

Summarizing: ω-operator for NFA

For each NFA $\mathcal{A}$ with $\varepsilon \notin \mathcal{L}(\mathcal{A})$ there exists an NBA $\mathcal{A}'$ such that:

$$\mathcal{L}_\omega(\mathcal{A}') = \mathcal{L}(\mathcal{A})^\omega \quad \text{and} \quad |\mathcal{A}'| = \mathcal{O}(|\mathcal{A}|)$$

Concatenating an NFA and an NBA

For NFA $\mathcal{A}$ and NBA $\mathcal{A}'$ (both over the alphabet Σ)

there exists an NBA $\mathcal{A}''$ with

$$\mathcal{L}_\omega(\mathcal{A}'') = \mathcal{L}(\mathcal{A}).\mathcal{L}_\omega(\mathcal{A}') \quad \text{and} \quad |\mathcal{A}''| = \mathcal{O}(|\mathcal{A}| + |\mathcal{A}'|)$$

Summarizing the results so far

For any ω-regular language $\mathcal{L}$

there exists an NBA $\mathcal{A}$ with $\mathcal{L}_\omega(\mathcal{A}) = \mathcal{L}$