Deterministic and Generalised Büchi Automata

Lecture #10 of Model Checking

Joost-Pieter Katoen

Lehrstuhl 2: Software Modeling & Verification

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

November 19, 2008

© JPK

Overview Lecture #10

⇒ Checking Non-Emptiness

- Deterministic Büchi Automata (DBA)
- Generalized Nondeterministic Büchi Automata (GNBA)

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
- $\delta : Q \times \Sigma \longrightarrow 2^Q$ is a transition function
- $F \subseteq Q$ is a set of accept (or: final) states

The size of $\mathcal{A}$, denoted $|\mathcal{A}|$, is the number of states and transitions in $\mathcal{A}$:

$$|\mathcal{A}| \;=\; |Q| + \sum_{q \in Q} \sum_{A \in \Sigma} |\, \delta(q, A)\, |$$

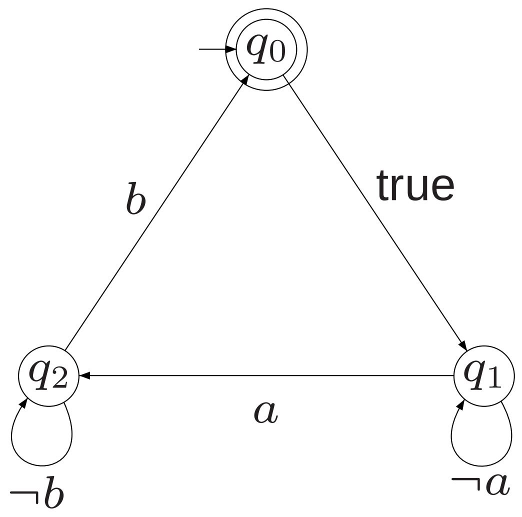
Language of an NBA

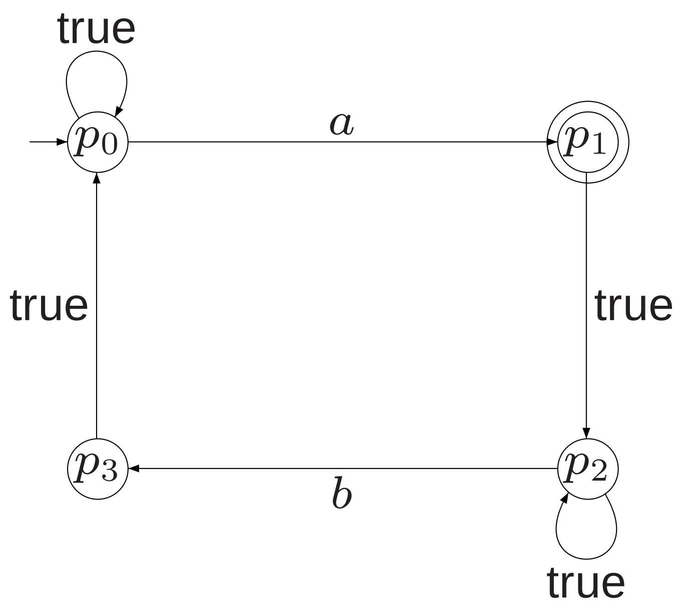
- NBA $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ and word $\sigma = A_0 A_1 A_2 \ldots \in \Sigma^\omega$
- A *run* for σ in $\mathcal{A}$ is an infinite sequence $q_0\, q_1\, q_2 \ldots$ such that:
  - $q_0 \in Q_0$ and $q_i \xrightarrow{A_i} q_{i+1}$ for all $0 \leqslant i$
- Run $q_0\, q_1\, q_2 \ldots$ 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 $\mathcal{A}$:

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

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

Equivalent NBA





infinitely often a and infinitely often b

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

Extended transition function

Extend the transition function δ to $\delta^* : Q \times \Sigma^* \to 2^Q$ by:

$$\delta^*(q, \varepsilon) = \{\, q \,\} \quad \text{and} \quad \delta^*(q, A) = \delta(q, A)$$

$$\delta^*(q, A_1 A_2 \ldots A_n) = \bigcup_{p \in \delta(q, A_1)} \delta^*(p, A_2 \ldots A_n)$$

$\delta^*(q, w)$ = set of states reachable from q for the word w

Checking non-emptiness

$\mathcal{L}_\omega(\mathcal{A}) \neq \varnothing$ if and only if

$$\underbrace{\exists q_0 \in Q_0.\ \exists q \in F.\ \exists \mathsf{w} \in \Sigma^*.\ \exists \mathsf{v} \in \Sigma^+.\ \ q \in \delta^*(q_0, \mathsf{w})\ \wedge\ q \in \delta^*(q, \mathsf{v})}_{\text{there is a reachable accept state on a cycle}}$$

The emptiness problem for NBA $\mathcal{A}$ can be solved in time $\mathcal{O}(|\mathcal{A}|)$

Non-blocking NBA

- NBA $\mathcal{A}$ is *non-blocking* if $\delta(q, A) \neq \varnothing$ for all q and $A \in \Sigma$
  - for each input word there exists an infinite run
- For each NBA $\mathcal{A}$ there exists a non-blocking NBA *trap*$(\mathcal{A})$ with:
  - $|\mathit{trap}(\mathcal{A})| \;=\; \mathcal{O}(|\mathcal{A}|)$ and $\mathcal{A} \equiv \mathit{trap}(\mathcal{A})$
- For $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ let *trap*$(\mathcal{A}) = (Q', \Sigma, \delta', Q_0, F)$ with:
  - $Q' = Q \cup \{ q_{trap} \}$ where $\{ q_{trap} \} \not\subseteq Q$
  -

$$\delta'(q, A) = \begin{cases} \delta(q, A) & : \text{ if } q \in Q \text{ and } \delta(q, A) \neq \varnothing \\ \{q_{trap}\} & : \text{ otherwise} \end{cases}$$

Overview Lecture #10

- Checking Non-Emptiness

$\Rightarrow$ Deterministic Büchi Automata (DBA)

- Generalized Nondeterministic Büchi Automata (GNBA)

Deterministic BA

Büchi automaton $\mathcal{A}$ is called *deterministic* if

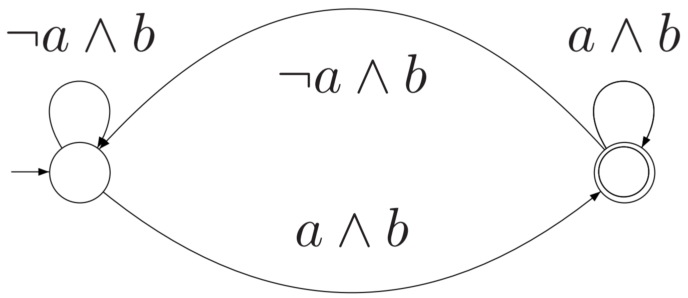
$$|Q_0| \leqslant 1 \quad \text{and} \quad |\delta(q, A)| \leqslant 1 \quad \text{for all } q \in Q \text{ and } A \in \Sigma$$

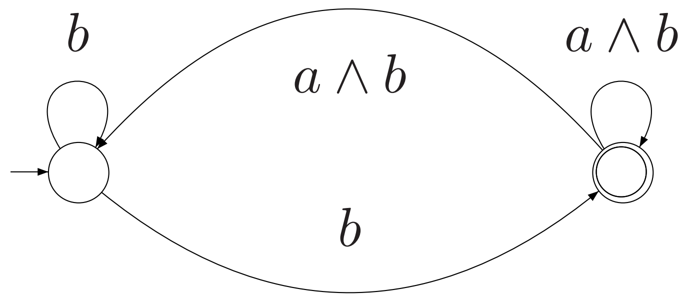
DBA $\mathcal{A}$ is called *total* if

$$|Q_0| = 1 \quad \text{and} \quad |\delta(q, A)| = 1 \quad \text{for all } q \in Q \text{ and } A \in \Sigma$$

total DBA provide unique runs for each input word

Example DBA for LT property





These NBA both represent the LT property ”always b and infinitely often a”

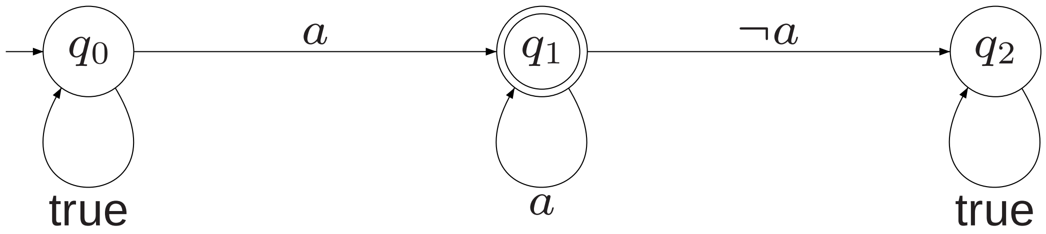
NBA are more expressive than DBA

NFA and DFA are equally expressive but NBA and DBA are not!

There is no DBA that accepts $\mathcal{L}_\omega((A + B)^* B^\omega)$

Proof

The need for nondeterminism



let $\{\ a\ \} = AP$, i.e., $2^{AP} = \{A, B\}$ where $A = \{\}$ and $B = \{a\}$

"eventually for ever a" equals $(A + B)^* B^\omega \ = \ (\{\} + \{a\})^* \{a\}^\omega$

Overview Lecture #10

- Checking Non-Emptiness
- Deterministic Büchi Automata (DBA)

$\Rightarrow$ Generalized Nondeterministic Büchi Automata (GNBA)

Generalized Büchi automata

- NBA are as expressive as ω-regular languages
- Variants of NBA exist that are equally expressive
  - Muller, Rabin, and Streett automata
  - *generalized Büchi automata* (GNBA)
- GNBA are like NBA, but have a distinct *acceptance criterion*
  - a GNBA requires to visit several sets $F_1, \ldots, F_k$ ($k \geqslant 0$) infinitely often
  - for $k{=}0$, all runs are accepting
  - for $k{=}1$ this boils down to an NBA
- GNBA are useful to relate temporal logic and automata
  - but they are equally expressive as NBA

Generalized Büchi automata

A *generalized NBA* (GNBA) $\mathcal{G}$ is a tuple $(Q, \Sigma, \delta, Q_0, \mathcal{F})$ where:

- Q is a finite set of states with $Q_0 \subseteq Q$ a set of initial states
- Σ is an alphabet
- $\delta : Q \times \Sigma \rightarrow 2^Q$ is a transition function
- $\mathcal{F} = \{ F_1, \ldots, F_k \}$ is a (possibly empty) subset of 2^Q

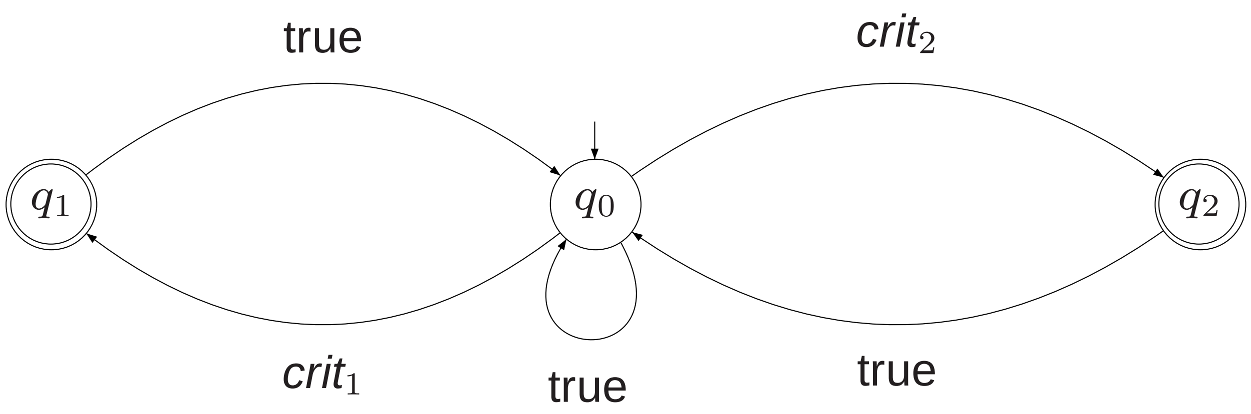
The size of $\mathcal{G}$, denoted $|\mathcal{G}|$, is the number of states and transitions in $\mathcal{G}$:

$$|\mathcal{G}| \; = \; |Q| + \sum_{q \in Q} \sum_{A \in \Sigma} | \, \delta(q, A) \, |$$

Language of a GNBA

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

Example



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

$$\mathcal{F} = \{ \{ q_1 \}, \{ q_2 \} \}$$

From GNBA to NBA

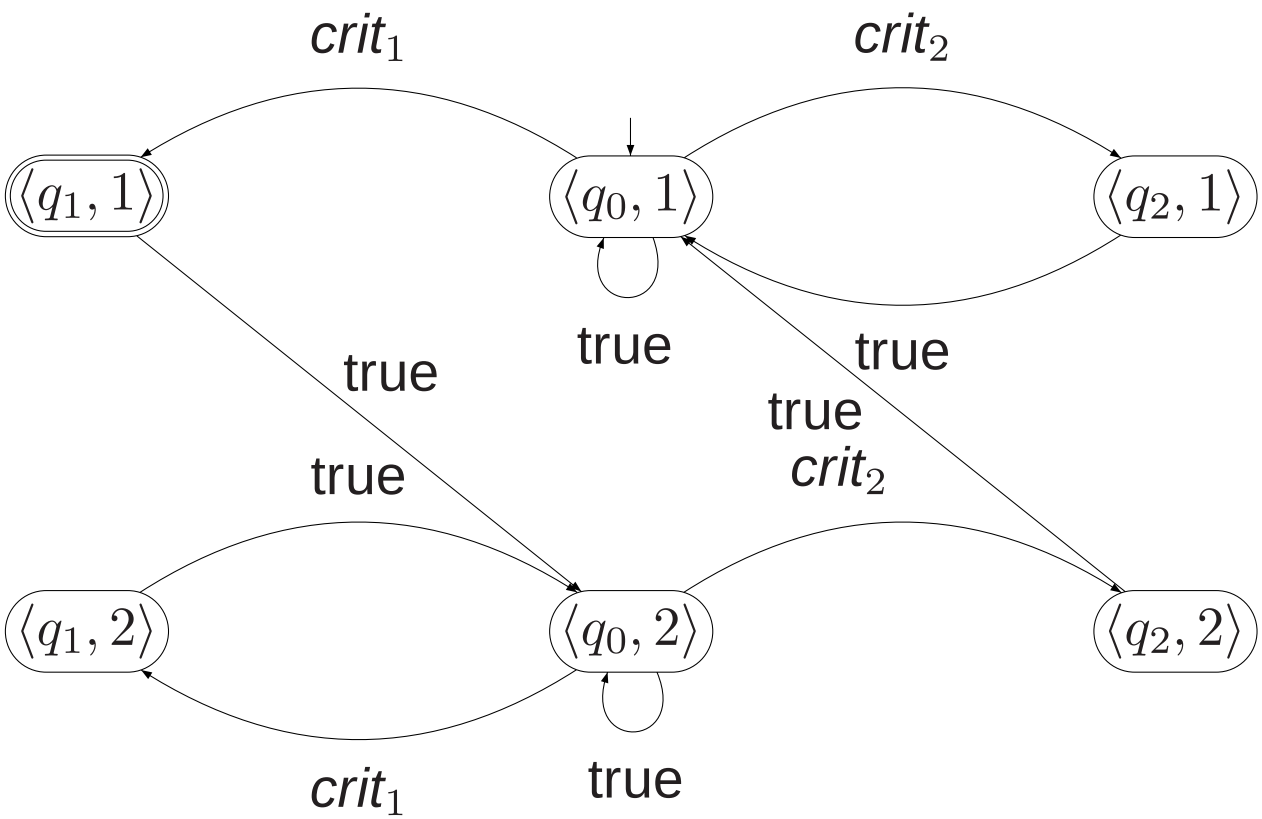
For any GNBA $\mathcal{G}$ there exists an NBA $\mathcal{A}$ with:

$\mathcal{L}_\omega(\mathcal{G}) = \mathcal{L}_\omega(\mathcal{A})$ and $|\mathcal{A}| = \mathcal{O}(|\mathcal{G}| \cdot |\mathcal{F}|)$

where $\mathcal{F}$ denotes the set of acceptance sets in $\mathcal{G}$

Proof

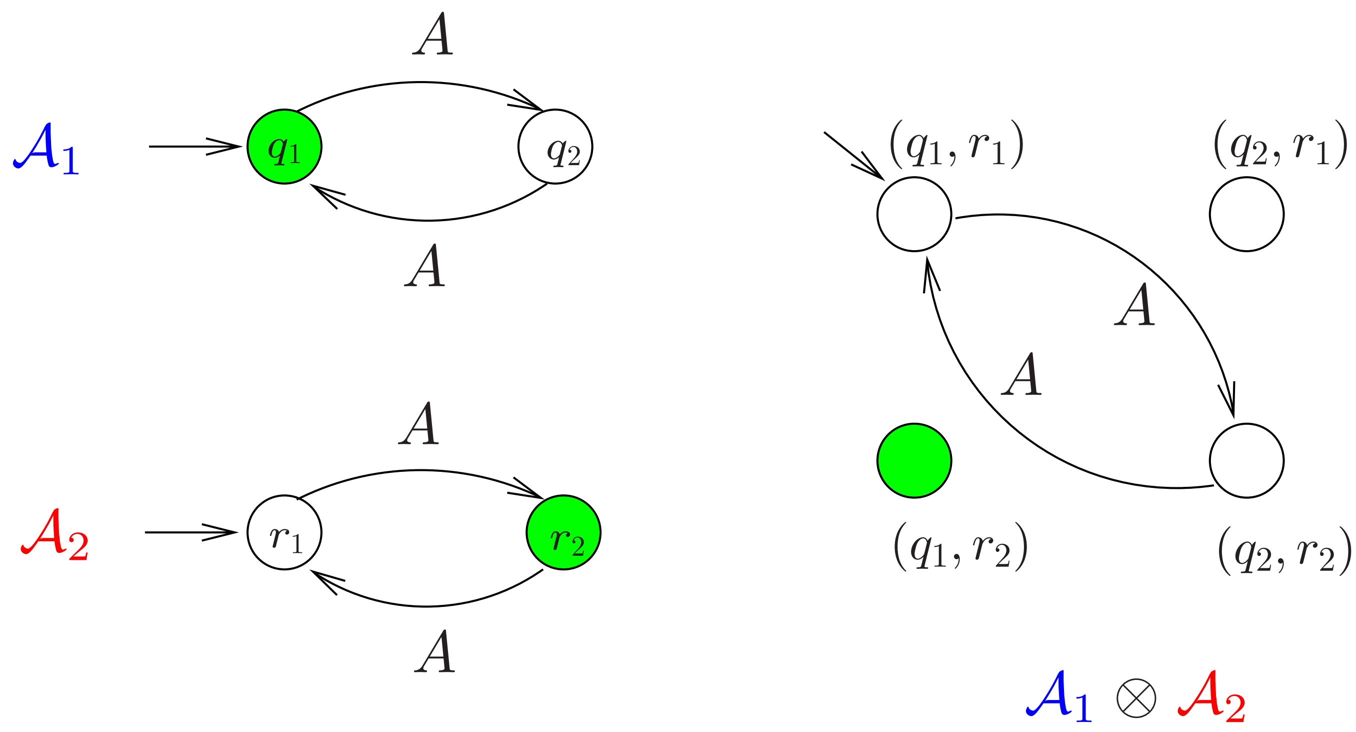
Example



crit$_1$
crit$_2$
$\langle q_1, 1\rangle$
$\langle q_0, 1\rangle$
$\langle q_2, 1\rangle$
true
true
true
true
true
crit$_2$
$\langle q_1, 2\rangle$
$\langle q_0, 2\rangle$
$\langle q_2, 2\rangle$
crit$_1$
true

Product of Büchi automata

The product construction for finite automata does *not* work:



$\mathcal{L}_\omega(\mathcal{A}_1) = \mathcal{L}_\omega(\mathcal{A}_2) = \{ A^\omega \}$, but $\mathcal{L}_\omega(\mathcal{A}_1 \otimes \mathcal{A}_2) = \varnothing$

Intersection

For GNBA $\mathcal{G}_1$ and $\mathcal{G}_2$ there exists a GNBA $\mathcal{G}$ with

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

Proof

Facts about Büchi automata

- They are as expressive as ω-regular languages
- They are closed under various operations and also under $\cap$
  - *deterministic* automaton $-\mathcal{A}$ accepts $-\mathcal{L}_\omega(\mathcal{A})$
- Nondeterministic BA are more expressive than deterministic BA
- Emptiness check = check for reachable recurrent accept state
  - this can be done in $\mathcal{O}(|\mathcal{A}|)$