Verification

Lecture 10

Martin Zimmermann

UNIVERSITÄT
DES
SAARLANDES

Plan for today

- Equivalence of Büchi automata & ω-regular expressions
- Generalized Büchi automata

Review: NBA and ω-regular languages

The class of languages accepted by NBA
agrees with the class of ω-regular languages

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

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

Rely on operations for NBA that mimic operations 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 $\mathcal{L}.\mathcal{L}_{\omega}(\mathcal{A}')$

Concatenation of 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

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

Proof

Let $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$, $\mathcal{A}' = (Q', \Sigma, \delta', Q_0', F')$ with $Q \cap Q' = \varnothing$.
Define NBA $\mathcal{A}'' = (Q'', \Sigma, \delta'', Q_0'', F'')$ with

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

For each (accepting) run $\rho = q_0 q_1 q_2 \cdots$ of $\mathcal{A}''$ on $A_0 A_1 A_2 \cdots \in \Sigma^\omega$:

- either $q_0 q_1 q_2 \cdots$ is an (accepting) run of $\mathcal{A}''$ on $A_0 A_1 A_2 \cdots$ (in case $Q_0 \cap F \neq \varnothing$), or
- there is an $n \geq 0$ such that
  - $q_0 \cdots q_n q$ is an accepting run of $\mathcal{A}$ on $A_0 \cdots A_n$ for some $q \in F$, and
  - $q_{n+1} q_{n+2} q_{n+3} \cdots$ is an (accepting) run of $\mathcal{A}'$ on $A_{n+1} A_{n+2} A_{n+3} \cdots$.

Summarizing the results so far

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

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

NBA accept ω-regular languages

$\boxed{\text{For each NBA } \mathcal{A}\text{: } \mathcal{L}_\omega(\mathcal{A}) \text{ is } \omega\text{-regular}}$

Proof

Let $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$. Define the NFA $\mathcal{A}_{q,p} = (Q, \Sigma, \delta, \{q\}, \{p\})$, (for $q, p \in Q$).

- Let $\sigma \in \mathcal{L}_\omega(\mathcal{A})$ with accepting run $q_0 q_1 q_2 \cdots$ that visits $q \in F$ infinitely often.

$$\sigma = \underbrace{w_0}_{\in \mathcal{L}(\mathcal{A}_{q_0,q})} \underbrace{w_1}_{\in \mathcal{L}(\mathcal{A}_{q,q})} \underbrace{w_2}_{\in \mathcal{L}(\mathcal{A}_{q,q})} \cdots$$

- On the other hand, each word of this form has an accepting run of $\mathcal{A}$.

Thus:

$$\mathcal{L}_\omega(\mathcal{A}) = \bigcup_{q_0 \in Q_0, q \in F} \mathcal{L}(\mathcal{A}_{q_0,q}).\left(\mathcal{L}(\mathcal{A}_{q,q})\right)^\omega$$

which is ω-regular.

Checking non-emptiness

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

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

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

Generalized Büchi automata

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 $F_1, \ldots, F_k$ ($k \geq 0$) 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

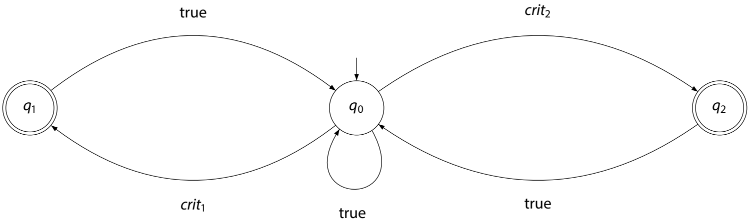
The size of $\mathcal{G}$, denoted $|\mathcal{G}|$, is the number of states and transitions in $\mathcal{G}$:

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

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 $0 \leq i$
- 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



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

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

From GNBA to NBA

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

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

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

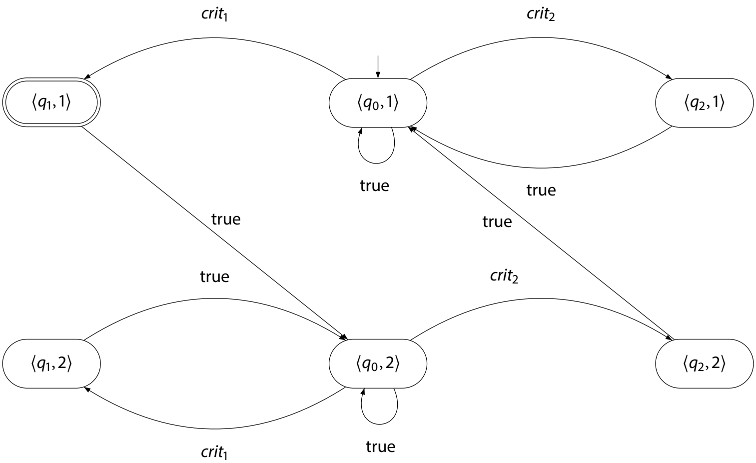
Proof

Let $\mathcal{G} = (Q, \Sigma, \delta, Q_0, \mathcal{F})$. W.l.o.g.: $\mathcal{F} = \{ F_1, \dots, F_k \}$, $k > 0$.
Define $\mathcal{A} = (Q', \Sigma, \delta', Q'_0, F')$ with

- $Q' = Q \times \{1, \dots, k\}$,
- $Q'_0 = Q \times \{1\}$,
- $F' = F_1 \times \{1\}$,
- $\delta((q, i), A) = \begin{cases} \{(q', i) \mid q' \in \delta(q, A)\} & \text{if } q \notin F_i, \\ \{(q', i+1) \mid q' \in \delta(q, A)\} & \text{otherwise.} \end{cases}$
  where $k + 1 = 1$.

A run $(q_0, i_0)(q_1, i_1)(q_2, i_2)\cdots$ of $\mathcal{A}$ on $A_0A_1A_2\cdots$ is accepting $\Leftrightarrow$ the run $q_0q_1q_2\cdots$ of $\mathcal{G}$ on $A_0A_1A_2\cdots$ is accepting.

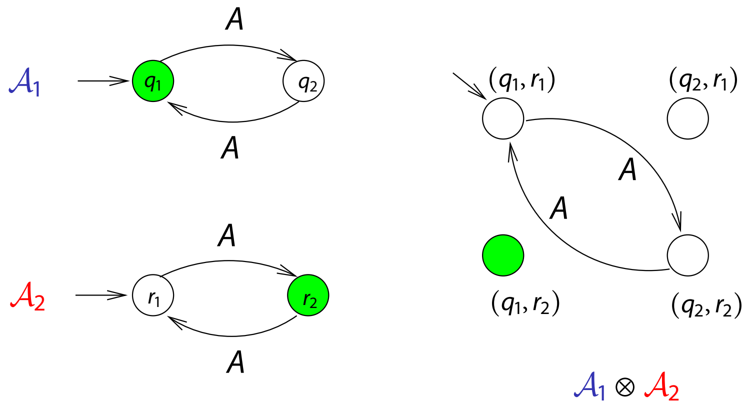
Example



crit1
crit2
⟨q1,1⟩
⟨q0,1⟩
⟨q2,1⟩
true
true
true
true
true
crit2
⟨q1,2⟩
⟨q0,2⟩
⟨q2,2⟩
crit1
true

Product of Büchi automata

The product construction for finite automata does not work:



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

Intersection

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

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

Proof

Let $\mathcal{G}_i = (Q_i, \Sigma, \delta_i, Q_{0,i}, \mathcal{F}_i)$ with $Q_1 \cap Q_2 = \varnothing$.

Define $\mathcal{G} = (Q_1 \times Q_2, \Sigma, \delta, Q_{0,1} \times Q_{0,2}, \mathcal{F})$ with

$$\frac{q_1' \in \delta_1(q_1, A) \ \wedge \ q_2' \in \delta_2(q_2, A)}{\langle q_1', q_2' \rangle \in \delta(\langle q_1, q_2 \rangle, A)}$$

and

$$\mathcal{F} = \{F_1 \times Q_2 \mid F_1 \in \mathcal{F}_1\} \cup \{Q_1 \times F_2 \mid F_2 \in \mathcal{F}_2\}$$

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 $\mathcal{O}(|\mathcal{A}|)$

Linear-time Temporal Logic

Syntax

modal logic over infinite sequences [Pnueli 1977]

- Propositional logic
  - $a \in AP$ — atomic proposition
  - $\neg\phi$ and $\phi \wedge \psi$ — negation and conjunction
- Temporal operators
  - $\bigcirc \phi$ — next state fulfills ϕ
  - $\phi \, \mathsf{U} \, \psi$ — ϕ holds Until a ψ-state is reached

linear temporal logic is a logic for describing LT properties

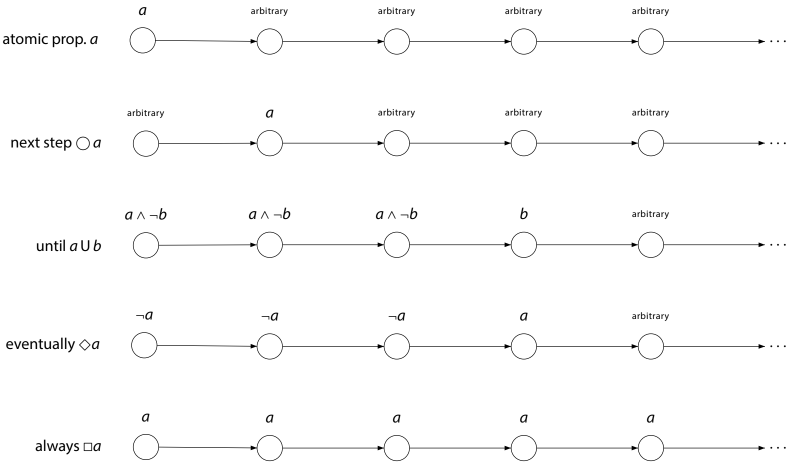
Derived operators

$$
\begin{aligned}
\phi \lor \psi &\equiv \neg(\neg\phi \land \neg\psi) \\
\phi \Rightarrow \psi &\equiv \neg\phi \lor \psi \\
\phi \Leftrightarrow \psi &\equiv (\phi \Rightarrow \psi) \land (\psi \Rightarrow \phi) \\
\phi \oplus \psi &\equiv (\phi \land \neg\psi) \lor (\neg\phi \land \psi) \\
\text{true} &\equiv \phi \lor \neg\phi \\
\text{false} &\equiv \neg\,\text{true} \\
\Diamond\phi &\equiv \text{true}\ \mathsf{U}\ \phi && \text{“sometimes in the future”} \\
\Box\phi &\equiv \neg\Diamond\neg\phi && \text{“from now on forever”}
\end{aligned}
$$

precedence order: the unary operators bind stronger than the binary ones.

$\neg$ and $\bigcirc$ bind equally strong. U takes precedence over $\wedge$, $\vee$, and $\rightarrow$

Intuitive semantics



Traffic light properties

- Once red, the light cannot become green immediately:

$$\Box\,(red \;\Rightarrow\; \neg \bigcirc green)$$

- The light becomes green eventually: $\Diamond\, green$
- Once red, the light always becomes green eventually: $\Box\,(red \;\Rightarrow\; \Diamond\, green)$
- Once red, the light always becomes green eventually after being yellow for some time inbetween:

$$\Box(red \rightarrow \bigcirc\,(red\, \mathrm{U}\,(yellow \;\wedge\; \bigcirc\,(yellow\, \mathrm{U}\, green))))$$