Reachability Problems on (Partially Lossy) Queue Automata

Highlights on Games, Logic and Automata 2019, Warsaw

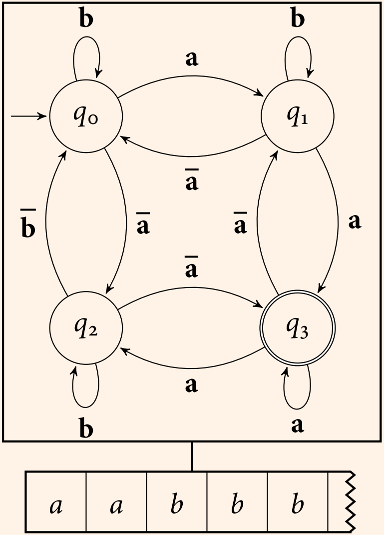
Chris Köcher

Automata and Logics Group
Technische Universität Ilmenau

September 19, 2019

- Let A be an alphabet.
- Two actions for each $a \in A$:
  - write letter $a \rightsquigarrow \mathbf{a}$
  - read letter $a \rightsquigarrow \overline{\mathbf{a}}$
- $\mathbf{A} := \{\mathbf{a} \mid a \in A\}$, $\overline{\mathbf{A}} := \{\overline{\mathbf{a}} \mid a \in A\}$
- $\Sigma := \mathbf{A} \uplus \overline{\mathbf{A}}$

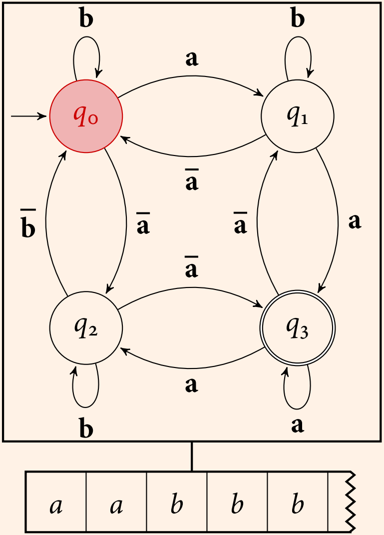
Example



TECHNISCHE UNIVERSITÄT ILMENAU

- Let A be an alphabet.
- Two actions for each $a \in A$:
  - write letter $a \rightsquigarrow \mathbf{a}$
  - read letter $a \rightsquigarrow \overline{\mathbf{a}}$
- $\mathbf{A} := \{\mathbf{a} \mid a \in A\}$, $\overline{\mathbf{A}} := \{\overline{\mathbf{a}} \mid a \in A\}$
- $\Sigma := \mathbf{A} \uplus \overline{\mathbf{A}}$

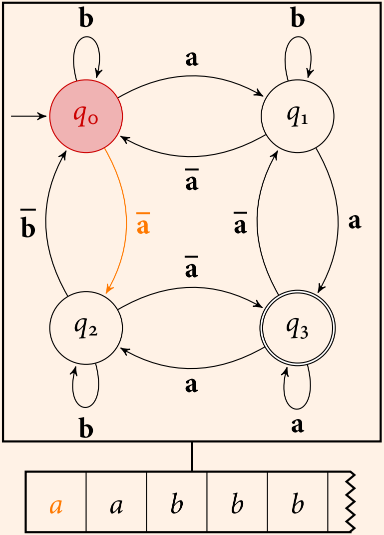
Example



Queue Automata

- Let A be an alphabet.
- Two actions for each $a \in A$:
    - write letter $a \rightsquigarrow \mathbf{a}$
    - read letter $a \rightsquigarrow \overline{\mathbf{a}}$
- $\mathbf{A} := \{\mathbf{a} \mid a \in A\}$, $\overline{\mathbf{A}} := \{\overline{\mathbf{a}} \mid a \in A\}$
- $\Sigma := \mathbf{A} \uplus \overline{\mathbf{A}}$

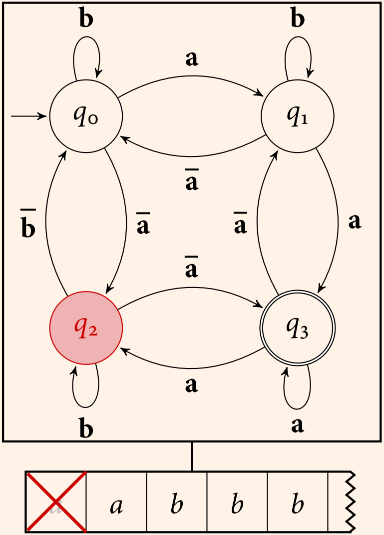
Example



Queue Automata

- Let A be an alphabet.
- Two actions for each $a \in A$:
  - write letter $a \rightsquigarrow \mathbf{a}$
  - read letter $a \rightsquigarrow \overline{\mathbf{a}}$
- $\mathbf{A} := \{\mathbf{a} \mid a \in A\}$, $\overline{\mathbf{A}} := \{\overline{\mathbf{a}} \mid a \in A\}$
- $\Sigma := \mathbf{A} \uplus \overline{\mathbf{A}}$

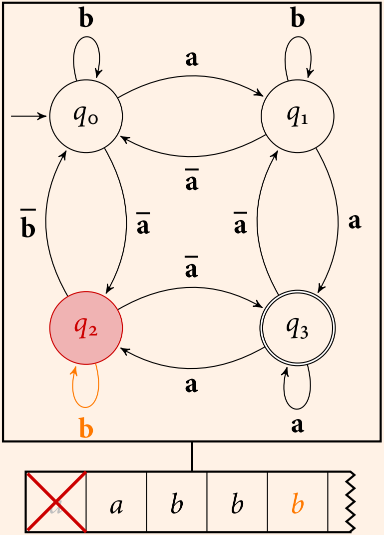
Example



Queue Automata

- Let A be an alphabet.
- Two actions for each $a \in A$:
  - write letter $a \rightsquigarrow \mathbf{a}$
  - read letter $a \rightsquigarrow \overline{\mathbf{a}}$
- $\mathbf{A} := \{\mathbf{a} \mid a \in A\}$, $\overline{\mathbf{A}} := \{\overline{\mathbf{a}} \mid a \in A\}$
- $\Sigma := \mathbf{A} \uplus \overline{\mathbf{A}}$

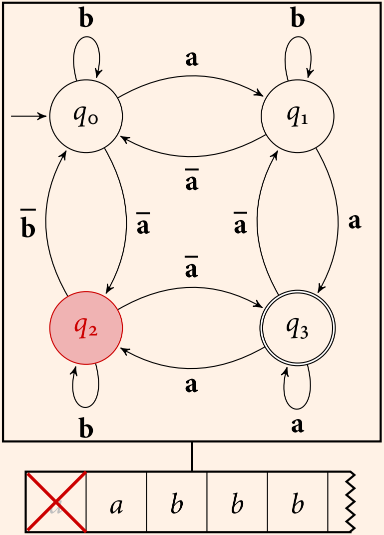
Example



Queue Automata

- Let A be an alphabet.
- Two actions for each $a \in A$:
  - write letter $a \rightsquigarrow \mathbf{a}$
  - read letter $a \rightsquigarrow \overline{\mathbf{a}}$
- $\mathbf{A} := \{\mathbf{a} \mid a \in A\}$, $\overline{\mathbf{A}} := \{\overline{\mathbf{a}} \mid a \in A\}$
- $\Sigma := \mathbf{A} \uplus \overline{\mathbf{A}}$

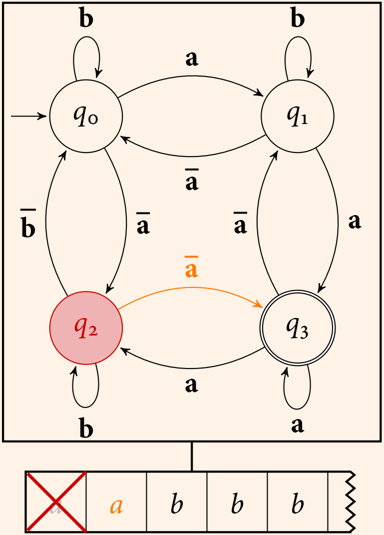
Example



TECHNISCHE UNIVERSITÄT ILMENAU

- Let A be an alphabet.
- Two actions for each $a \in A$:
  - write letter $a \rightsquigarrow \mathbf{a}$
  - read letter $a \rightsquigarrow \overline{\mathbf{a}}$
- $\mathbf{A} := \{\mathbf{a} \mid a \in A\}$, $\overline{\mathbf{A}} := \{\overline{\mathbf{a}} \mid a \in A\}$
- $\Sigma := \mathbf{A} \uplus \overline{\mathbf{A}}$

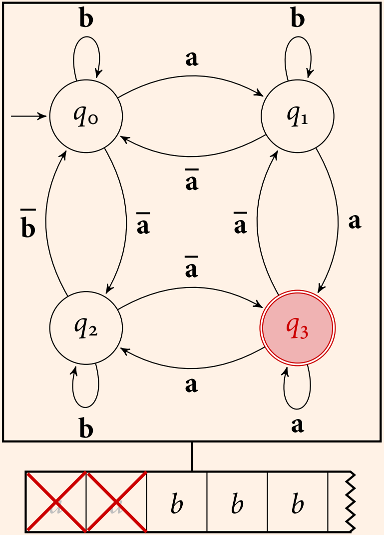
Example



Queue Automata

- Let A be an alphabet.
- Two actions for each $a \in A$:
  - write letter $a \rightsquigarrow \mathbf{a}$
  - read letter $a \rightsquigarrow \bar{\mathbf{a}}$
- $\mathbf{A} := \{\mathbf{a} \mid a \in A\}$, $\overline{\mathbf{A}} := \{\bar{\mathbf{a}} \mid a \in A\}$
- $\Sigma := \mathbf{A} \uplus \overline{\mathbf{A}}$

Example



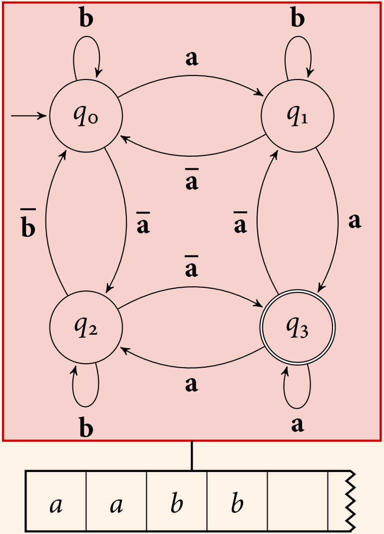
Inputs:

- $\mathbf{T} \subseteq \Sigma^*$ regular language of transformation sequences
- $L \subseteq A^*$ regular language of queue contents

Compute:

- $\textsc{Reach}(L, \mathbf{T})$:= the set of all queue contents after application of $\mathbf{T}$ on L

Example



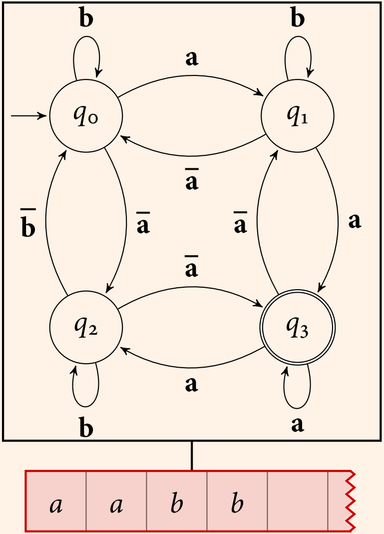
Inputs:

- $\mathbf{T} \subseteq \mathbf{\Sigma}^*$ regular language of transformation sequences
- $L \subseteq A^*$ regular language of queue contents

Compute:

- $\textsc{Reach}(L, \mathbf{T})$:= the set of all queue contents after application of $\mathbf{T}$ on L

Example



Theorem (Brand, Zafiropulo 1983)

Queue Automata can simulate Turing-machines.

- $\textsc{Reach}(L, \mathbf{T})$ can be any recursively enumerable language
- Approximate $\textsc{Reach}(L, \mathbf{T})$ step-by-step

Theorem (Boigelot, Godefroid, Willems, Wolper 1997)

Let $L \subseteq A^*$ be regular and $\mathbf{t} \in \mathbf{\Sigma}^*$. Then $\textsc{Reach}(L, \mathbf{t}^*)$ is effectively regular.

- Combine multiple iterations of a loop to a meta-transformation

Aim

Generalize this result.

Multiple Loops

- There is fixed $\mathbf{T} = \{\mathbf{t_1}, \ldots, \mathbf{t_n}\}$ such that $\textsc{Reach}(L, \mathbf{T}^*)$ can be any recursively enumerable language

Theorem (Main Theorem)

Let $L, W, R \subseteq A^*$ be regular. Then $\textsc{Reach}(L, (\mathbf{W}\overline{\mathbf{R}})^*)$ is effectively regular (in polynomial time).

- Proved by some reduction to reachability in pushdown automata
- Uses that reachability in pushdown automata preserves regularity by [Bouajjani, Esparza, Maler 1997]

Consequences

Corollary

Let $L \subseteq A^*$ and $\mathbf{T} \subseteq \Sigma^*$ be regular. Then $\textsc{Reach}(L, \mathbf{T}^*)$ *is regular if*

1. $\mathbf{T} = \overline{\mathbf{R_1}}\mathbf{W}\overline{\mathbf{R_2}}$ *for regular* $W, R_1, R_2 \subseteq A^*$,
2. $\mathbf{T} = \mathbf{W} \cup \overline{\mathbf{R}}$ *for regular* $W, R \subseteq A^*$,
3. $\mathbf{T} = \mathrm{shuffle}(\mathbf{W}, \overline{\mathbf{R}})$ *for regular* $W, R \subseteq A^*$, *or*
4. $\mathbf{T} = \{\mathbf{t}\}$ *for* $\mathbf{t} \in \Sigma^*$ *(cf. [Boigelot et al. 1997]).*

- Remark: Proofs of 3 and 4 use some result from [K. 2018, cf. STACS'18]

Thank you!