Reachability Questions on (Partially Lossy) Queue Automata

28. Theorietag “Automaten und Formale Sprachen”, Wittenberg

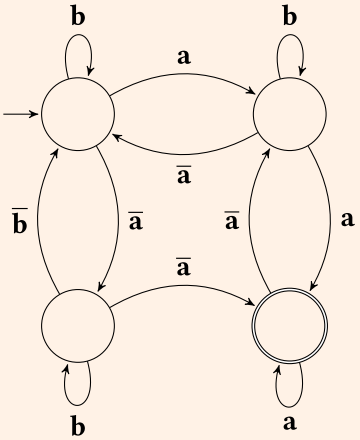
Chris Köcher

Automata and Logics Group
Technische Universität Ilmenau

September 26, 2018

- Let A be an alphabet ($|A| \geq 2$).
- Two actions for each $a \in A$:
  - write letter $a \rightsquigarrow \mathbf{a}$
  - read letter $a \rightsquigarrow \overline{\mathbf{a}}$
- $\overline{\mathbf{A}} := \{\overline{\mathbf{a}} \mid a \in A\}$
- $\boldsymbol{\Sigma} := \mathbf{A} \uplus \overline{\mathbf{A}}$

Example

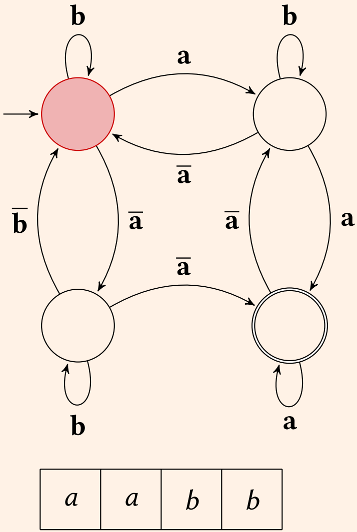


a	a	b	b

Queue Automata

- Let A be an alphabet ($|A| \geq 2$).
- Two actions for each $a \in A$:
  - write letter $a \rightsquigarrow \mathbf{a}$
  - read letter $a \rightsquigarrow \overline{\mathbf{a}}$
- $\overline{\mathbf{A}} := \{\overline{\mathbf{a}} \mid a \in A\}$
- $\mathbf{\Sigma} := \mathbf{A} \uplus \overline{\mathbf{A}}$

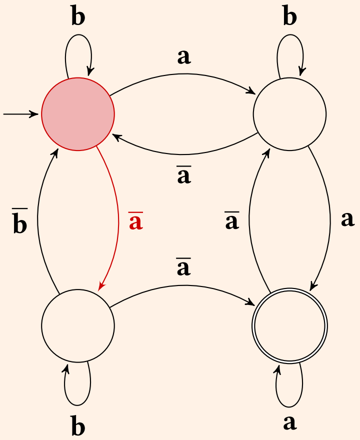
Example



Queue Automata

- Let A be an alphabet ($|A| \geq 2$).
- Two actions for each $a \in A$:
  - write letter $a \rightsquigarrow \mathbf{a}$
  - read letter $a \rightsquigarrow \overline{\mathbf{a}}$
- $\overline{\mathbf{A}} := \{\overline{\mathbf{a}} \mid a \in A\}$
- $\mathbf{\Sigma} := \mathbf{A} \uplus \overline{\mathbf{A}}$

Example

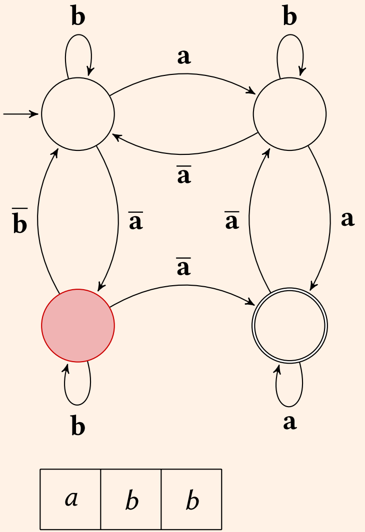


a	a	b	b

Queue Automata

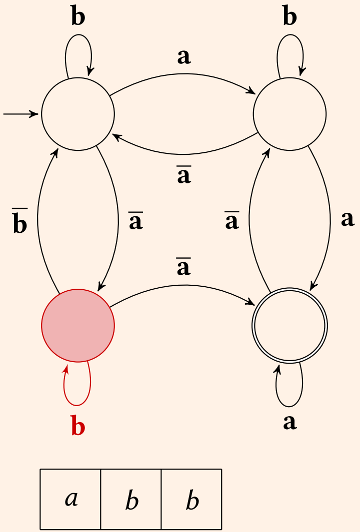
- Let A be an alphabet ($|A| \geq 2$).
- Two actions for each $a \in A$:
  - write letter $a \rightsquigarrow \mathbf{a}$
  - read letter $a \rightsquigarrow \overline{\mathbf{a}}$
- $\overline{\mathbf{A}} := \{\overline{\mathbf{a}} \mid a \in A\}$
- $\mathbf{\Sigma} := \mathbf{A} \uplus \overline{\mathbf{A}}$

Example



- Let A be an alphabet ($|A| \geq 2$).
- Two actions for each $a \in A$:
  - write letter $a \rightsquigarrow \mathbf{a}$
  - read letter $a \rightsquigarrow \overline{\mathbf{a}}$
- $\overline{\mathbf{A}} := \{\overline{\mathbf{a}} \mid a \in A\}$
- $\mathbf{\Sigma} := \mathbf{A} \uplus \overline{\mathbf{A}}$

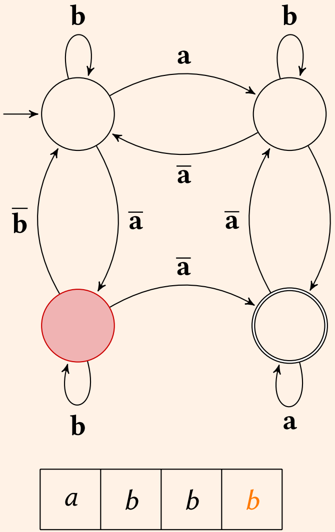
Example



Queue Automata

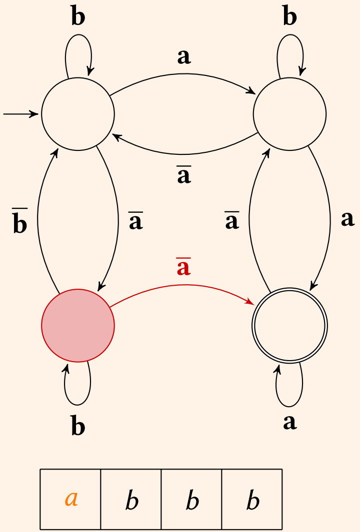
- Let A be an alphabet ($|A| \geq 2$).
- Two actions for each $a \in A$:
  - write letter $a \rightsquigarrow \mathbf{a}$
  - read letter $a \rightsquigarrow \overline{\mathbf{a}}$
- $\overline{\mathbf{A}} := \{\overline{\mathbf{a}} \mid a \in A\}$
- $\boldsymbol{\Sigma} := \mathbf{A} \uplus \overline{\mathbf{A}}$

Example



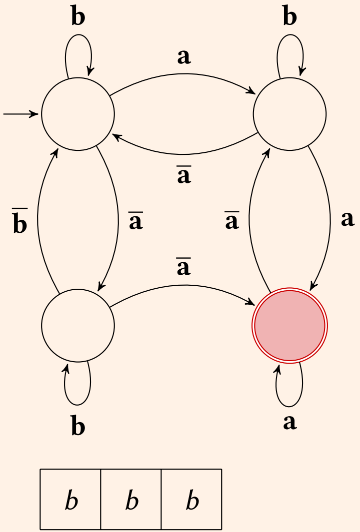
- Let A be an alphabet ($|A| \geq 2$).
- Two actions for each $a \in A$:
  - write letter $a \rightsquigarrow \mathbf{a}$
  - read letter $a \rightsquigarrow \overline{\mathbf{a}}$
- $\overline{\mathbf{A}} := \{\overline{\mathbf{a}} \mid a \in A\}$
- $\mathbf{\Sigma} := \mathbf{A} \uplus \overline{\mathbf{A}}$

Example



- Let A be an alphabet ($|A| \geq 2$).
- Two actions for each $a \in A$:
  - write letter $a \rightsquigarrow \mathbf{a}$
  - read letter $a \rightsquigarrow \overline{\mathbf{a}}$
- $\overline{\mathbf{A}} := \{\overline{\mathbf{a}} \mid a \in A\}$
- $\mathbf{\Sigma} := \mathbf{A} \uplus \overline{\mathbf{A}}$

Example



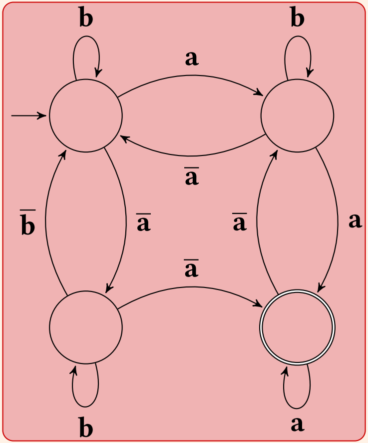
Inputs:

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

Compute:

- $\text{post}_{\mathbf{T}}(L) :=$
  $\{q \in A^* \mid \exists p \in L, \mathbf{t} \in \mathbf{T} \colon p \xrightarrow{\mathbf{t}} q\}$
- $\text{pre}_{\mathbf{T}}(L) :=$
  $\{q \in A^* \mid \exists p \in L, \mathbf{t} \in \mathbf{T} \colon q \xrightarrow{\mathbf{t}} p\}$

Example



a	a	b	b

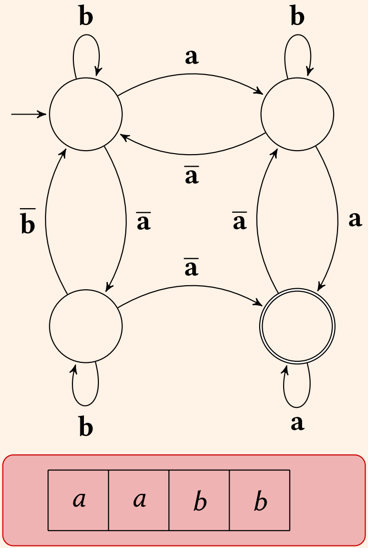
Inputs:

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

Compute:

- $\mathrm{post}_{\mathbf{T}}(L) :=$ $\{q \in A^* \mid \exists p \in L, \mathbf{t} \in \mathbf{T}\colon p \xrightarrow{\mathbf{t}} q\}$
- $\mathrm{pre}_{\mathbf{T}}(L) :=$ $\{q \in A^* \mid \exists p \in L, \mathbf{t} \in \mathbf{T}\colon q \xrightarrow{\mathbf{t}} p\}$

Example



Known Results

Theorem (Brand, Zafiropulo 1983)

Queue Automata are Turing-complete.

- $\mathrm{post}_{\mathbf{T}}(\varepsilon)$ / $\mathrm{pre}_{\mathbf{T}}(\varepsilon)$ can be any recursively enumerable language
- holds already for $\mathbf{T} = \{\mathbf{w_1}, \ldots, \mathbf{w_n}\}^*$ with $\mathbf{w_1}, \ldots, \mathbf{w_n} \in \mathbf{\Sigma}^*$

Theorem (Boigelot, Godefroid, Willems, Wolper 1997)

For $\mathbf{w} \in \mathbf{\Sigma}^*$ and $L \subseteq A^*$ regular, the sets $\mathrm{post}_{\mathbf{w}^*}(L)$ and $\mathrm{pre}_{\mathbf{w}^*}(L)$ are effectively regular.

Behavioral Equivalence: Definition

TECHNISCHE UNIVERSITÄT ILMENAU

Definition

$\mathbf{s}, \mathbf{t} \in \mathbf{\Sigma}^*$ behave equally (in symbols $\mathbf{s} \equiv \mathbf{t}$) if, and only if,

$$\forall p, q \in A^*: p \xrightarrow{\mathbf{s}} q \iff p \xrightarrow{\mathbf{t}} q$$

$\Rightarrow$ $\text{post}_{\mathbf{s}}(L) = \text{post}_{\mathbf{t}}(L)$ and $\text{pre}_{\mathbf{s}}(L) = \text{pre}_{\mathbf{t}}(L)$

Theorem (Huschenbett, Kuske, Zetzsche 2014)

$\equiv$ is the least congruence on $\mathbf{\Sigma}^*$ satisfying the following equations:

1. $\mathbf{a}\overline{\mathbf{b}} \equiv \overline{\mathbf{b}}\mathbf{a}$ *if* $a \neq b$
2. $\mathbf{a}\overline{\mathbf{a}\mathbf{c}} \equiv \overline{\mathbf{a}}\mathbf{a}\overline{\mathbf{c}}$
3. $\mathbf{c}\mathbf{a}\overline{\mathbf{a}} \equiv \mathbf{c}\overline{\mathbf{a}}\mathbf{a}$

for any $a, b, c \in A$.

Behavioral Equivalence: Result

Corollary

Let $\mathbf{T} \subseteq \mathbf{\Sigma}^*$ *be closed under* $\equiv$ *and* $L \subseteq A^*$. *Then*

$$\mathrm{post}_{\mathbf{T}}(L) = \mathrm{post}_{\mathbf{T} \cap \overline{\mathbf{A}}^* \mathbf{A}^* \overline{\mathbf{A}}^*}(L) \quad \textit{and} \quad \mathrm{pre}_{\mathbf{T}}(L) = \mathrm{pre}_{\mathbf{T} \cap \overline{\mathbf{A}}^* \mathbf{A}^* \overline{\mathbf{A}}^*}(L)\,.$$

Theorem

Let $\mathbf{T} \subseteq \mathbf{\Sigma}^*$ *be regular and closed under* $\equiv$ *and* $L \subseteq A^*$ *be regular. Then* $\mathrm{post}_{\mathbf{T}}(L)$ *and* $\mathrm{pre}_{\mathbf{T}}(L)$ *are effectively regular (in polynomial time).*

Proof. We have $\mathbf{T} \cap \overline{\mathbf{A}}^* \mathbf{A}^* \overline{\mathbf{A}}^* = \bigcup_i \overline{\mathbf{K_{i,1}}} \mathbf{K_{i,2}} \overline{\mathbf{K_{i,3}}}$ for finitely many regular languages $\mathbf{K_{i,1}}, \mathbf{K_{i,2}}, \mathbf{K_{i,3}} \subseteq \mathbf{A}^*$. Then

$$\mathrm{post}_{\mathbf{T}}(L) = \mathrm{post}_{\bigcup_i \overline{\mathbf{K_{i,1}}} \mathbf{K_{i,2}} \overline{\mathbf{K_{i,3}}}}(L) = \bigcup_i {}_{K_{i,3}}\backslash\left(\left({}_{K_{i,1}}\backslash L\right) \cdot K_{i,2}\right)$$

which is regular since class of regular languages is closed under quotients and products. □

Corollary

Let $\mathbf{T} \subseteq \mathbf{\Sigma}^*$ *be closed under* $\equiv$ *and* $L \subseteq A^*$. *Then*

$$\mathrm{post}_{\mathbf{T}}(L) = \mathrm{post}_{\mathbf{T} \cap \overline{\mathbf{A}}^* \mathbf{A}^* \overline{\mathbf{A}}^*}(L) \quad \textit{and} \quad \mathrm{pre}_{\mathbf{T}}(L) = \mathrm{pre}_{\mathbf{T} \cap \overline{\mathbf{A}}^* \mathbf{A}^* \overline{\mathbf{A}}^*}(L)\,.$$

Theorem

Let $\mathbf{T} \subseteq \mathbf{\Sigma}^*$ *be regular and closed under* $\equiv$ *and* $L \subseteq A^*$ *be regular. Then* $\mathrm{post}_{\mathbf{T}}(L)$ *and* $\mathrm{pre}_{\mathbf{T}}(L)$ *are effectively regular (in polynomial time).*

Example

Let $R_1, R_2 \subseteq A^*$ be regular. $R_1 \sqcup \overline{R_2}$ is regular and closed under $\equiv$.

Read-Write Independence: Definition

Definition

Let $\mathbf{T} \subseteq \mathbf{\Sigma}^*$. $\mathbf{T}$ is read-write independent if for all $\mathbf{s}, \mathbf{t} \in \mathbf{T}$ there is $\mathbf{u}_{\mathbf{s},\mathbf{t}} \in \mathbf{T}$ with

- write$(\mathbf{u}_{\mathbf{s},\mathbf{t}}) =$ write$(\mathbf{s})$ and
- read$(\mathbf{u}_{\mathbf{s},\mathbf{t}}) =$ read$(\mathbf{t})$.

Example

- $\{\mathbf{w}\}$ for $\mathbf{w} \in \mathbf{\Sigma}^*$
- $\overline{\mathbf{K}}\,\mathbf{L}\,\overline{\mathbf{M}}$ for $\mathbf{K}, \mathbf{L}, \mathbf{M} \subseteq \mathbf{A}^*$
- Perm$(\mathbf{w})$ ... the set of all permutations of $\mathbf{w} \in \mathbf{\Sigma}^*$

Theorem

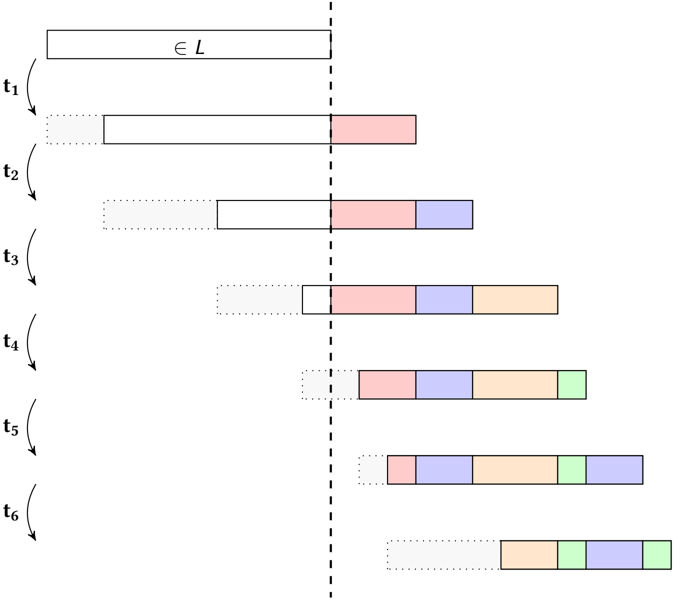
Let $\mathbf{T} \subseteq \mathbf{\Sigma}^*$ *be finite and read-write independent and* $L \subseteq A^*$ *be regular. Then* $\mathrm{post}_{\mathbf{T}^*}(L)$ *and* $\mathrm{pre}_{\mathbf{T}^*}(L)$ *are effectively regular (in polynomial space).*

Conjecture

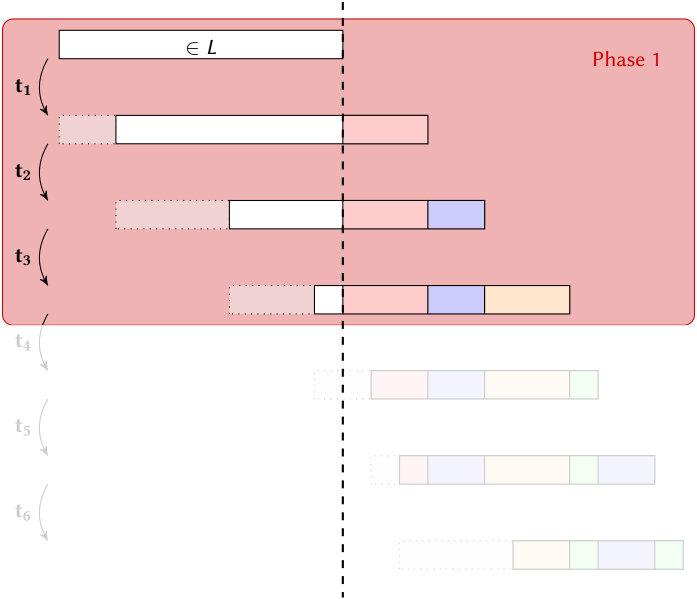
$\mathrm{post}_{\mathbf{T}^*}(L)$ and $\mathrm{pre}_{\mathbf{T}^*}(L)$ are effectively regular even if

- $\mathbf{T}$ is regular, closed under $\equiv$, and read-write independent.
- $\mathbf{T} = \overline{\mathbf{K}}\,\mathbf{L}\,\overline{\mathbf{M}}$ for $\mathbf{K}, \mathbf{L}, \mathbf{M} \subseteq \mathbf{A}^*$ regular.

Read-Write Independence: Proof - Overview

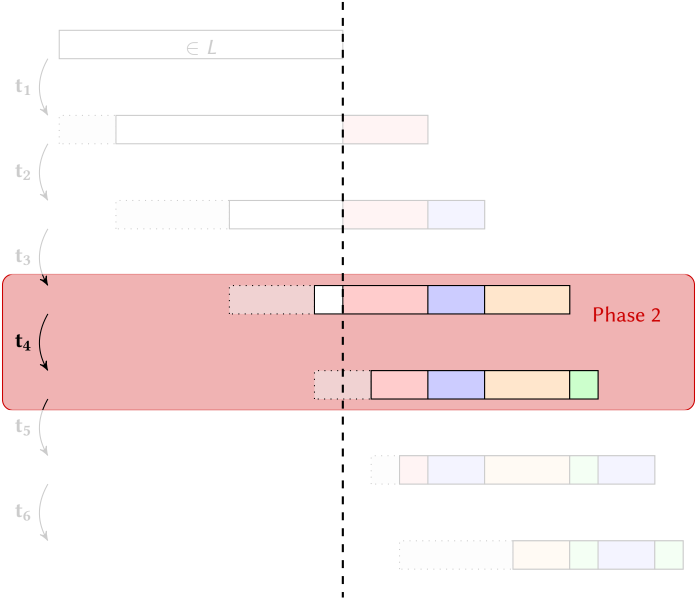


Read-Write Independence: Proof - Overview



1. Read only letters from the queue's initial contents
   - Let $\mathcal{A} = (Z, A, I, \delta, F)$ be NFA accepting L.
   - for each $z \in Z$:
     - Compute $R_z := L(\mathcal{A}_{F:=\{z\}}) \cap \mathsf{read}(\mathbf{T})^*$
     - Replace each occurrence of a word from $\mathsf{read}(\mathbf{T})$ in R_z by $\mathsf{write}(\mathbf{T}) \quad \rightsquigarrow \quad W_z \subseteq \mathsf{write}(\mathbf{T})^*$
     - Result: $L(\mathcal{A}_{I:=\{z\}}) \cdot W_z$
   - $X_1 := \bigcup_{z \in Z} L(\mathcal{A}_{I:=\{z\}}) \cdot W_z$

Read-Write Independence: Proof - Overview

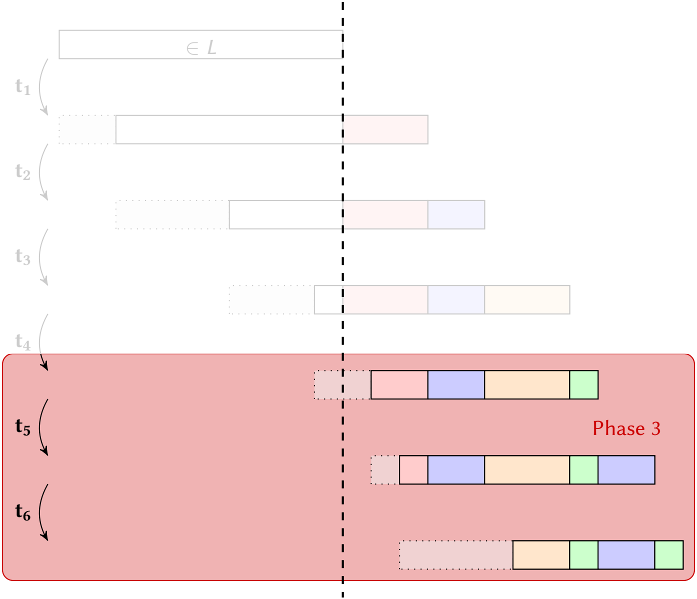


Read-Write Independence: Proof - Phases 1 + 2

TECHNISCHE UNIVERSITÄT ILMENAU

1. Read only letters from the queue's initial contents
   - Let $\mathcal{A} = (Z, A, I, \delta, F)$ be NFA accepting L.
   - for each $z \in Z$:
     - Compute $R_z := L(\mathcal{A}_{F:=\{z\}}) \cap \text{read}(\mathbf{T})^*$
     - Replace each occurrence of a word from $\text{read}(\mathbf{T})$ in R_z by $\text{write}(\mathbf{T})$ $\rightsquigarrow$ $W_z \subseteq \text{write}(\mathbf{T})^*$
     - Result: $L(\mathcal{A}_{I:=\{z\}}) \cdot W_z$
   - $X_1 := \bigcup_{z \in Z} L(\mathcal{A}_{I:=\{z\}}) \cdot W_z$
2. Read some letters from the queue's initial contents and some letters written in Phase 1
   - $X_2 := \text{post}_{\mathbf{T}}(X_1) \cap \text{suffixes}(\text{write}(\mathbf{T}))\,\text{write}(\mathbf{T})^*$

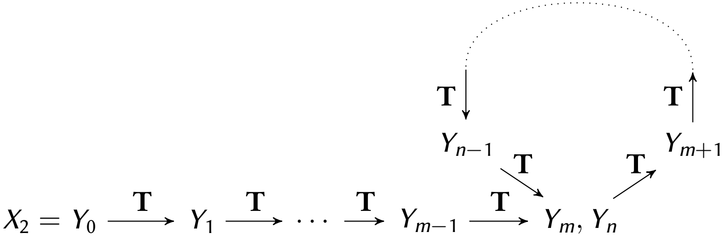
Read-Write Independence: Proof - Overview



Read-Write Independence: Proof - Phase 3

TECHNISCHE UNIVERSITÄT ILMENAU

3 Read only letters written in Phases 1-3



- $Y_i = \mathrm{post}_{\mathbf{T}^i}(X_2) = \bigcup_{s \in S_i} s\,\mathrm{write}(\mathbf{T})^{E_{s,i}}$
  - $S_i \subseteq \mathrm{suffixes}(\mathrm{write}(\mathbf{T}))$ minimal
  - $E_{s,i} \subseteq \mathbb{N}$ semi-linear
- $S_m = S_n$ and arithmetical difference of the $E_{s,n}$'s and $E_{s,m}$'s is effectively semi-linear
- X_3 is effectively regular

$\Rightarrow \mathrm{post}_{\mathbf{T}^*}(L) = X_1 \cup X_2 \cup X_3.$ □

(Partially) Lossy Queue Automata

- can forget some contents of their queue content at any time
- Known Results (for fully lossy queues):
  - $\text{post}_{\mathbf{T}}(L)$ is regular
  - NFA accepting $\text{post}_{\mathbf{T}}(L)$ cannot be computed [Mayr 2003]
  - Membership of $\text{post}_{\mathbf{T}}(L)$ is decidable [Abdulla, Jonsson 1996], but not primitive recursive [Schnoebelen 2002]
  - $\text{pre}_{\mathbf{T}}(L)$ is effectively regular [Abdulla, Jonsson 1996]
- Results from this talk hold for arbitrary Partially Lossy Queue Automata

Thank you!