Two-wayness: Automata & Transducers

Bruno Guillon

IRIF – Université Paris-Diderot, Paris 7
Dipartimento di Informatica – Università degli studi di Milano

May 30, 2016
PhD defense

Computation

A **computation** is a sequence of successive *elementary operations*.

Computation

A **computation** is a sequence of successive *elementary operations*.

$$f : x \mapsto 5x - 3$$

Computation

A **computation** is a sequence of successive *elementary operations*.

$$f : x \mapsto 5x - 3$$

Compute $f(x)$

with $+$ **and** $\times$

— start with x

Computation

A **computation** is a sequence of successive *elementary operations*.

$$f : x \mapsto 5x - 3$$

Compute $f(x)$

with $+$ **and** $\times$

— start with x

1. multiply by 5

Computation

A **computation** is a sequence of successive *elementary operations*.

$$f : x \mapsto 5x - 3$$

Compute $f(x)$

with $+$ **and** $\times$

— start with x

1. multiply by 5
2. add -3

Computation

A **computation** is a sequence of successive *elementary operations*.

$$f : x \mapsto 5x - 3$$

Compute $f(x)$

with $+$ **and** $\times$

— start with x

1. multiply by 5
2. add -3

with $+$ **only**

— start with x

Computation

A **computation** is a sequence of successive *elementary operations*.

$$f : x \mapsto 5x - 3$$

Compute $f(x)$

with + **and** ×

— start with x

1. multiply by 5
2. add -3

with + **only**

— start with x

1. add x

Computation

A **computation** is a sequence of successive *elementary operations*.

$$f : x \mapsto 5x - 3$$

Compute $f(x)$

with $+$ **and** $\times$

— start with x

1. multiply by 5
2. add -3

with $+$ **only**

— start with x

1. add x
2. add x

Computation

A **computation** is a sequence of successive *elementary operations*.

$$f : x \mapsto 5x - 3$$

Compute $f(x)$

with $+$ **and** $\times$

— start with x

1. multiply by 5
2. add -3

with $+$ **only**

— start with x

1. add x
2. add x
3. add x
4. add x

Computation

A **computation** is a sequence of successive *elementary operations*.

$$f : x \mapsto 5x - 3$$

Compute $f(x)$

with $+$ **and** $\times$

— start with x

1. multiply by 5
2. add -3

with $+$ **only**

— start with x

1. add x
2. add x
3. add x
4. add x
5. add -3

Computation

A **computation** is a sequence of successive *elementary operations*.

$$g : x \mapsto x^2 + x \qquad\qquad f : x \mapsto 5x - 3$$

Compute $f(x)$

with $+$ **and** $\times$

— start with x

1. multiply by 5
2. add -3

with $+$ **only**

— start with x

1. add x
2. add x
3. add x
4. add x
5. add -3

Computation

A **computation** is a sequence of successive *elementary operations*.

$g : x \mapsto x^2 + x$ $\qquad$ $f : x \mapsto 5x - 3$

Compute $f(x)$

with $+$ **and** $\times$

— start with x

1. multiply by 5
2. add -3

with $+$ **only**

— start with x

1. add x
2. add x
3. add x
4. add x
5. add -3

Compute $g(x)$

— start with x

Computation

A **computation** is a sequence of successive *elementary operations*.

$g : x \mapsto x^2 + x$ $\qquad$ $f : x \mapsto 5x - 3$

Compute $f(x)$

with $+$ **and** $\times$

— start with x

1. multiply by 5
2. add -3

with $+$ **only**

— start with x

1. add x
2. add x
3. add x
4. add x
5. add -3

Compute $g(x)$

— start with x

1. multiply by x

Computation

A **computation** is a sequence of successive *elementary operations*.

$g : x \mapsto x^2 + x$ $\qquad$ $f : x \mapsto 5x - 3$

Compute $f(x)$

with + and ×

— start with x

1. multiply by 5
2. add -3

with + only

— start with x

1. add x
2. add x
3. add x
4. add x
5. add -3

Compute $g(x)$

— start with x

1. multiply by x
2. add x

Computation

A **computation** is a sequence of successive *elementary operations*.

$$g : x \mapsto x^2 + x \qquad\qquad f : x \mapsto 5x - 3$$

Compute $f(x)$

with + and ×

— start with x
1. multiply by 5
2. add -3

with + only

— start with x
1. add x
2. add x
3. add x
4. add x
5. add -3

Compute $g(x)$

— start with x
1. multiply by x
2. add x

— start with x

Computation

A **computation** is a sequence of successive *elementary operations*.

$$g : x \mapsto x^2 + x \qquad\qquad f : x \mapsto 5x - 3$$

Compute $f(x)$

with $+$ **and** $\times$

— start with x
1. multiply by 5
2. add -3

with $+$ **only**

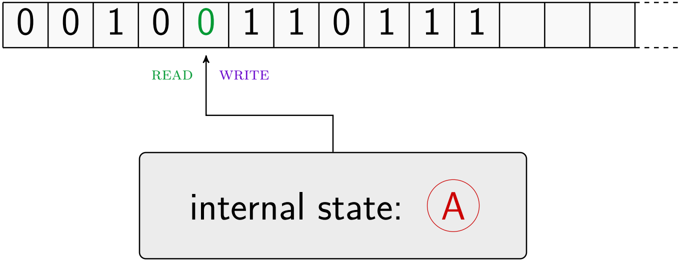
— start with x
1. add x
2. add x
3. add x
4. add x
5. add -3

Compute $g(x)$

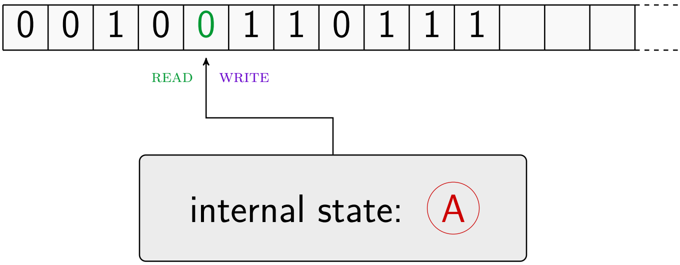
— start with x
1. multiply by x
2. add x

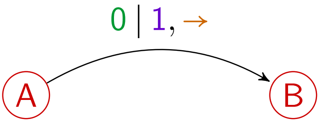
— start with x
- Impossible

Turing machines

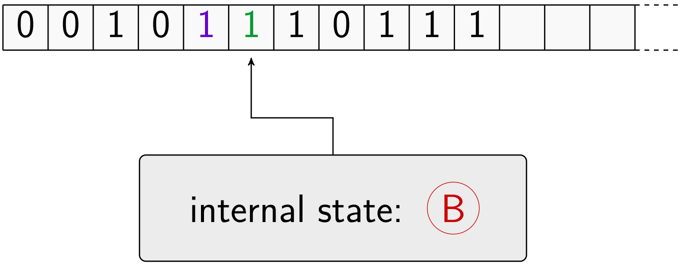


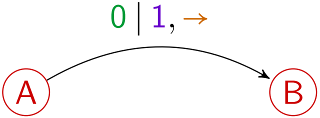
Turing machines



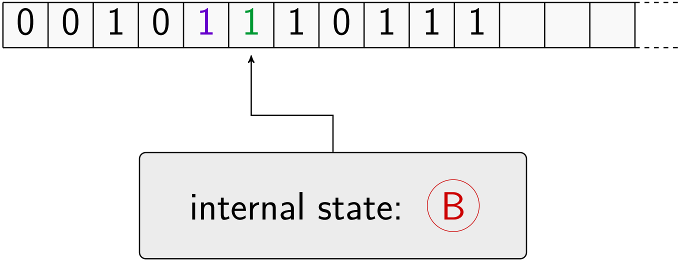


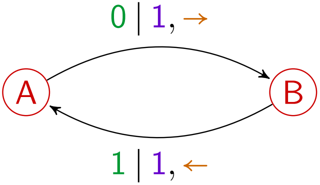
Turing machines



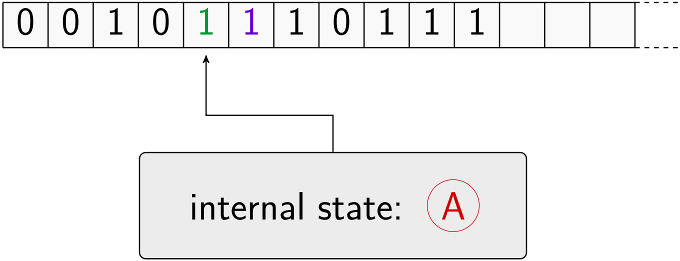


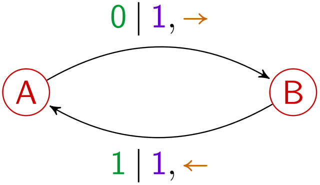
Turing machines



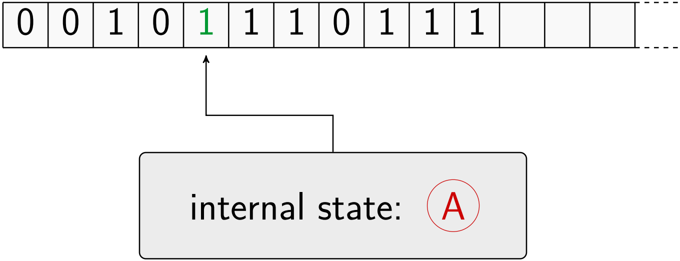


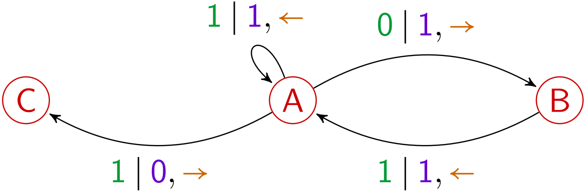
Turing machines



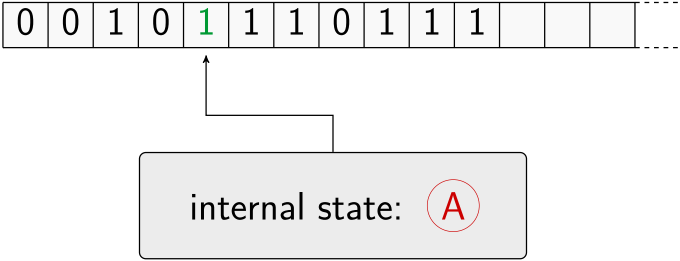


Turing machines

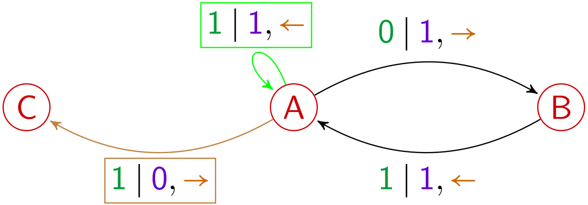




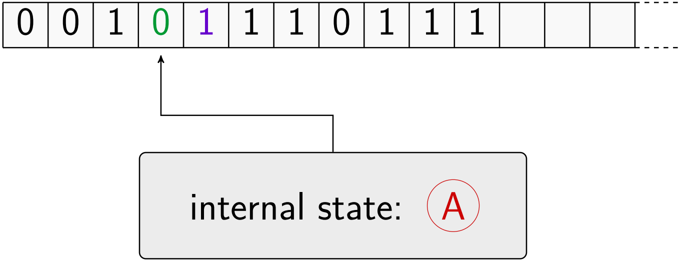
Turing machines



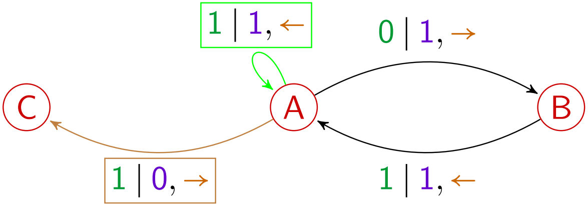
nondeterministic choice: □ or □



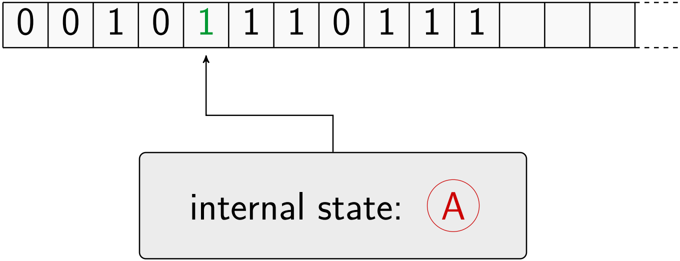
Turing machines



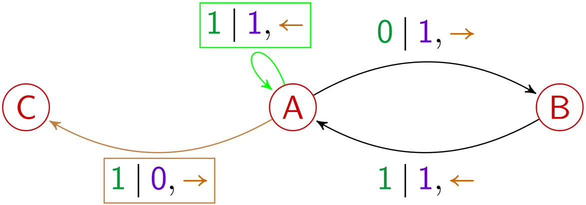
nondeterministic choice: □ or □



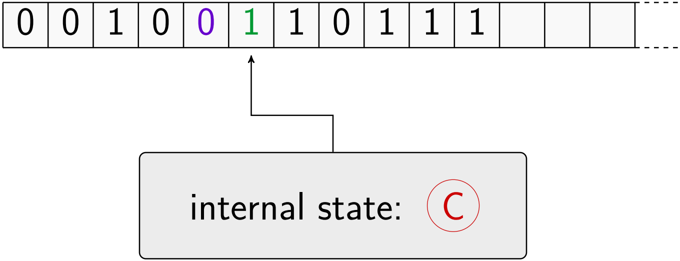
Turing machines



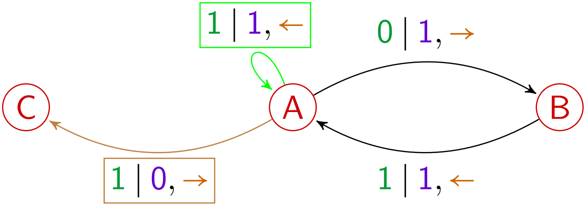
nondeterministic choice: □ or □



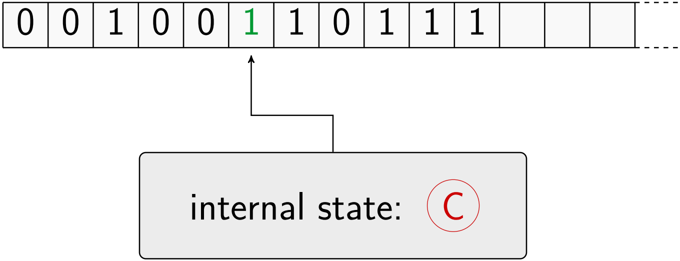
Turing machines

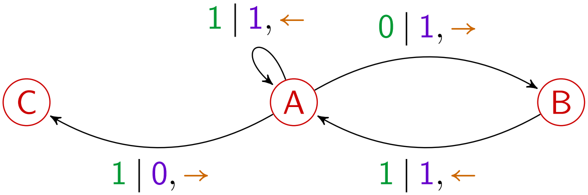


nondeterministic choice: □ or □

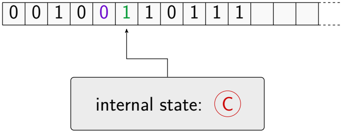


Turing machines

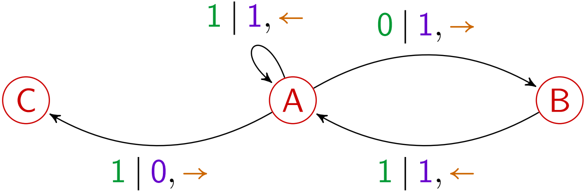




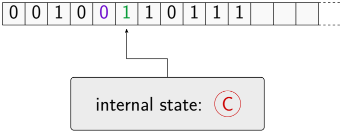
Turing machines



Huge computational power

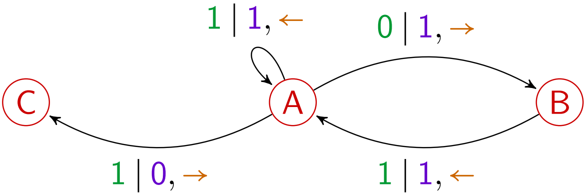


Turing machines

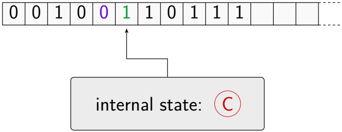


Huge computational power

- infinite memory

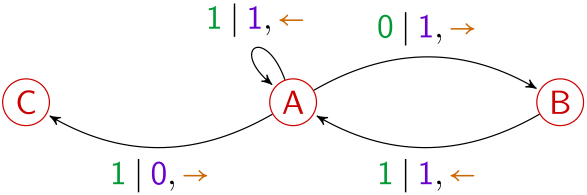


Turing machines

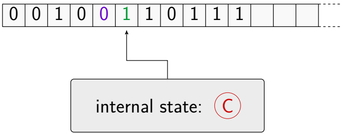


Huge computational power

- infinite memory
- universal



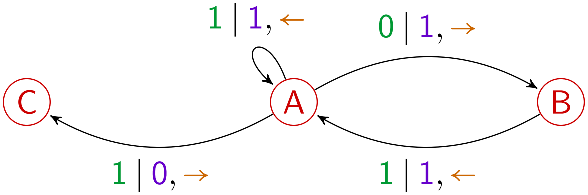
Turing machines



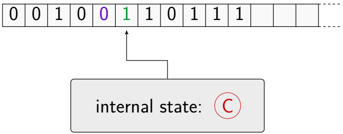
Complex dynamics

Huge computational power

- infinite memory
- universal



Turing machines

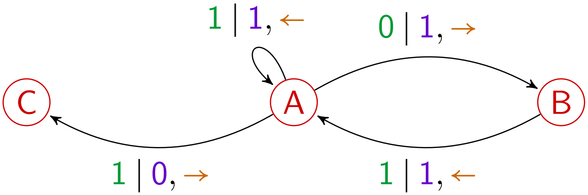


Complex dynamics

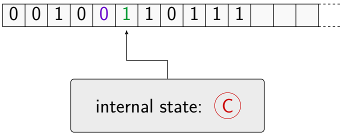
- undecidability of the halting problem

Huge computational power

- infinite memory
- universal



Turing machines



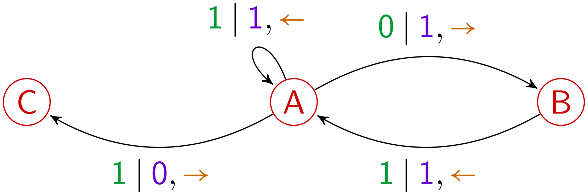
Complex dynamics

- undecidability of the halting problem
- contribution of nondeterminism

$$e.g., \quad \mathrm{P} \stackrel{?}{=} \mathrm{NP} \quad \text{and} \quad \mathrm{L} \stackrel{?}{=} \mathrm{NL}$$

Huge computational power

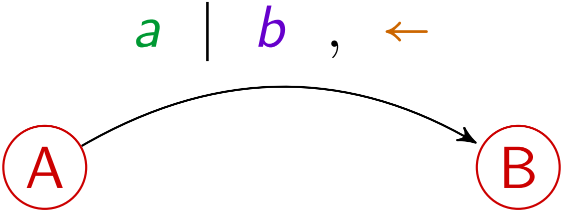
- infinite memory
- universal



Finite automata

Definition

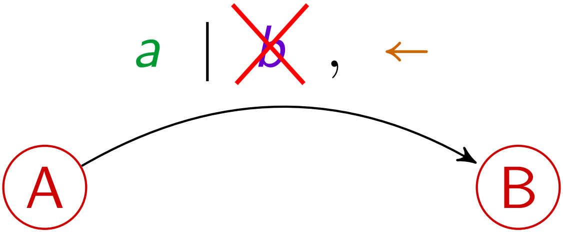
A **finite automata** (FA) is a one-way read-only Turing machine.



Finite automata

Definition

A **finite automata** (FA) is a one-way read-only Turing machine.

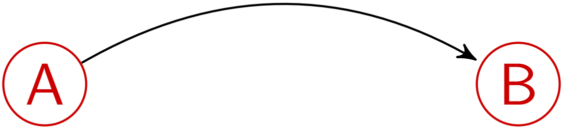


Finite automata

Definition

A **finite automata** (FA) is a one-way read-only Turing machine.

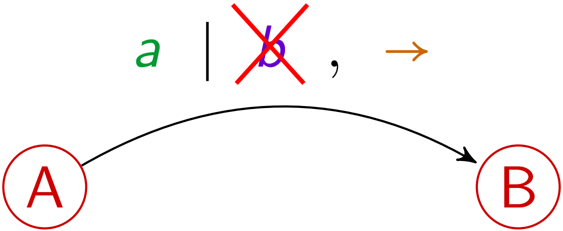
$$a \mid \cancel{b}, \cancel{\leftarrow}$$



Finite automata

Definition

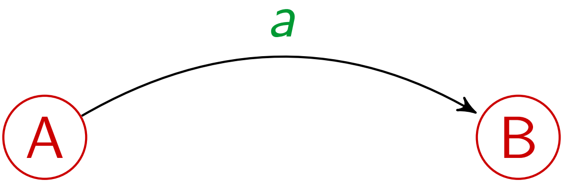
A **finite automata** (FA) is a one-way read-only Turing machine.



Finite automata

Definition

A **finite automata** (FA) is a one-way read-only Turing machine.



Finite automata

Definition

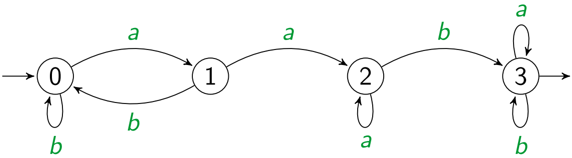
A **finite automata** (FA) is a one-way read-only Turing machine.
FAs are **recognizers**.

Finite automata

Definition

A **finite automata** (FA) is a one-way read-only Turing machine.
FAs are **recognizers**.

Example

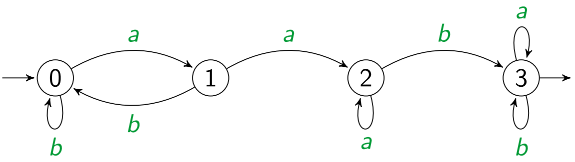


Finite automata

Definition

A **finite automata** (FA) is a one-way read-only Turing machine.
FAs are **recognizers**.

Example



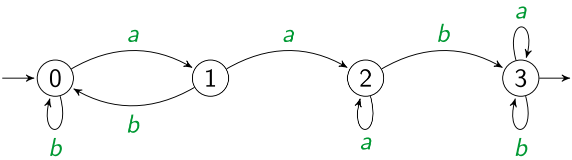
accepts the language $\{a, b\}^* \cdot a \cdot a \cdot b \cdot \{a, b\}^*$

Finite automata

Definition

A **finite automata** (FA) is a one-way read-only Turing machine.
FAs are **recognizers**.

Example



accepts the language $\{a, b\}^* \cdot a \cdot a \cdot b \cdot \{a, b\}^*$

Theorem (Kleene)

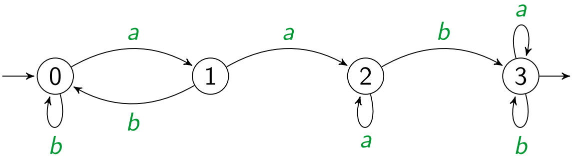
finite automata = *rational languages*

Finite automata

Definition

A **finite automata** (FA) is a one-way read-only Turing machine.
FAs are **recognizers**.

Example



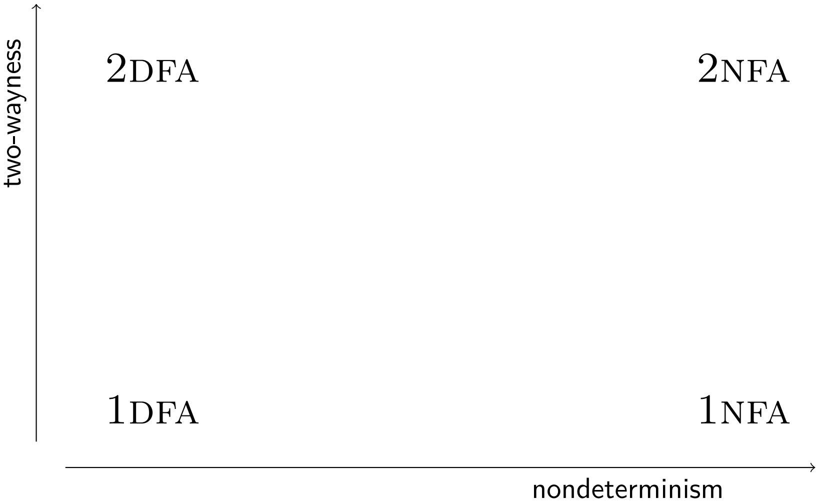
accepts the language $\{a, b\}^* \cdot a \cdot a \cdot b \cdot \{a, b\}^*$

Theorem (Kleene)

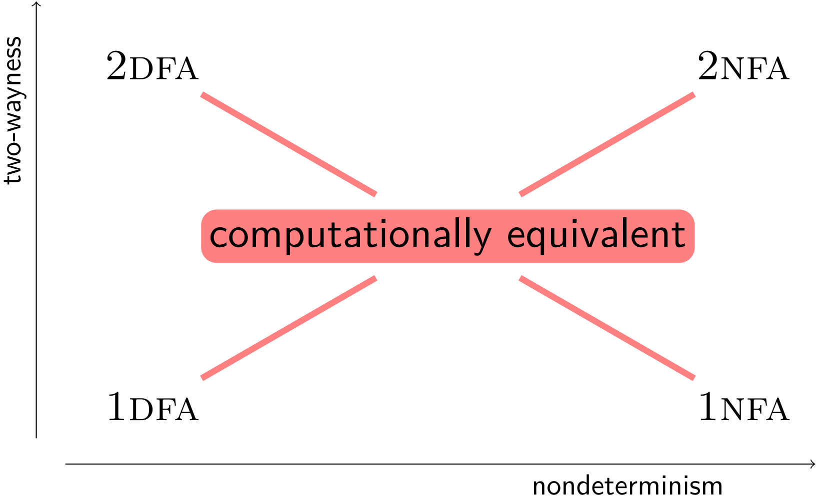
finite automata = *rational languages*

The smallest family including **finite languages**
closed under **union**, **concatenation** and **Kleene star**.

Two-wayness and nondeterminism



Two-wayness and nondeterminism

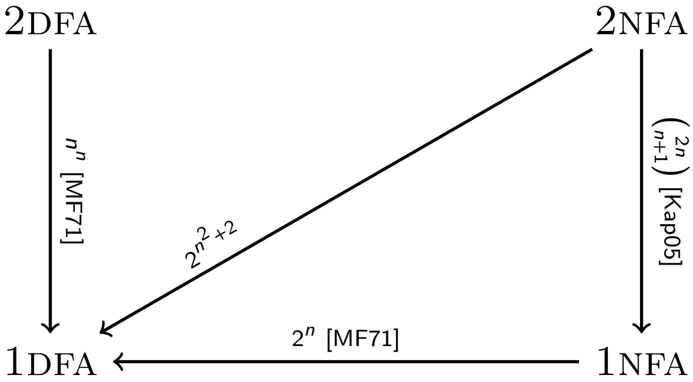


Two-wayness and nondeterminism

$$
\begin{tikzcd}
2\textsc{dfa} \arrow[r, "n"] & 2\textsc{nfa} \\
1\textsc{dfa} \arrow[u, "n"] \arrow[ur, "n"] \arrow[r, "n"] & 1\textsc{nfa} \arrow[u, "n"]
\end{tikzcd}
$$

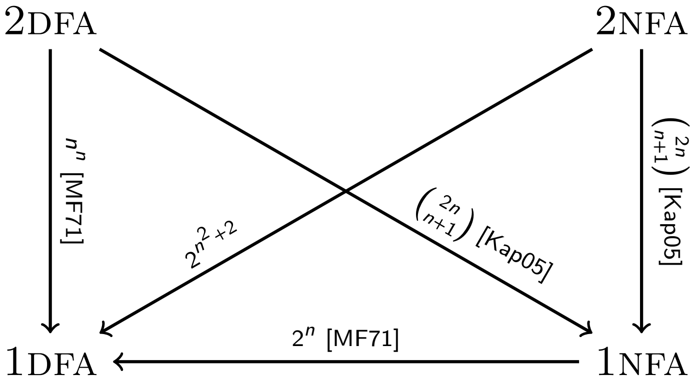
natural simulations

Two-wayness and nondeterminism



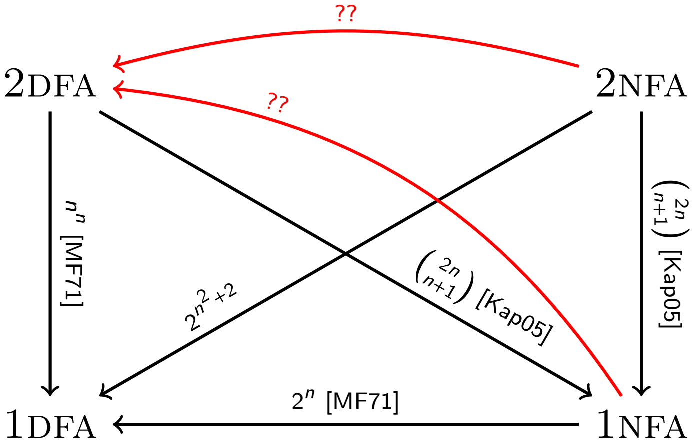
known results on simulations

Two-wayness and nondeterminism



known results on simulations

Two-wayness and nondeterminism



The two main questions (Sakoda & Sipser 1978)

- the optimal cost of the simulation of 1NFA by 2DFA?
- the optimal cost of the simulation of 2NFA by 2DFA?

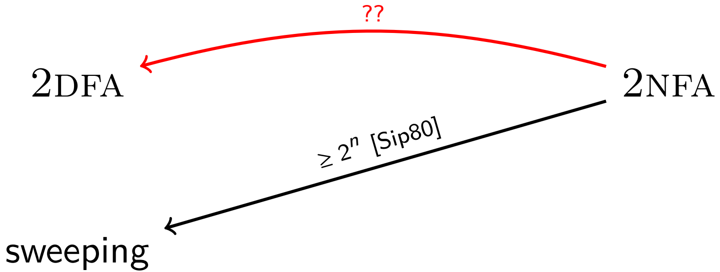
Two-wayness and nondeterminism

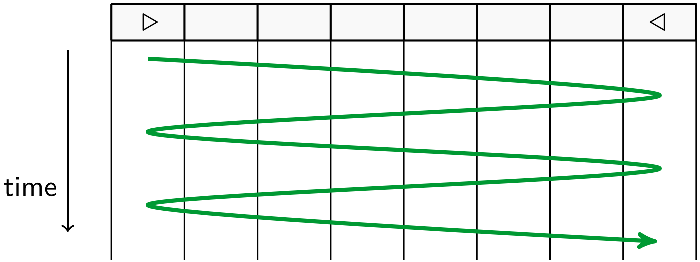
$$2\textsc{dfa} \xleftarrow{\;??\;} 2\textsc{nfa}$$

The two main questions (Sakoda & Sipser 1978)

- the optimal cost of the simulation of $1\textsc{nfa}$ by $2\textsc{dfa}$?
- the optimal cost of the simulation of $2\textsc{nfa}$ by $2\textsc{dfa}$?

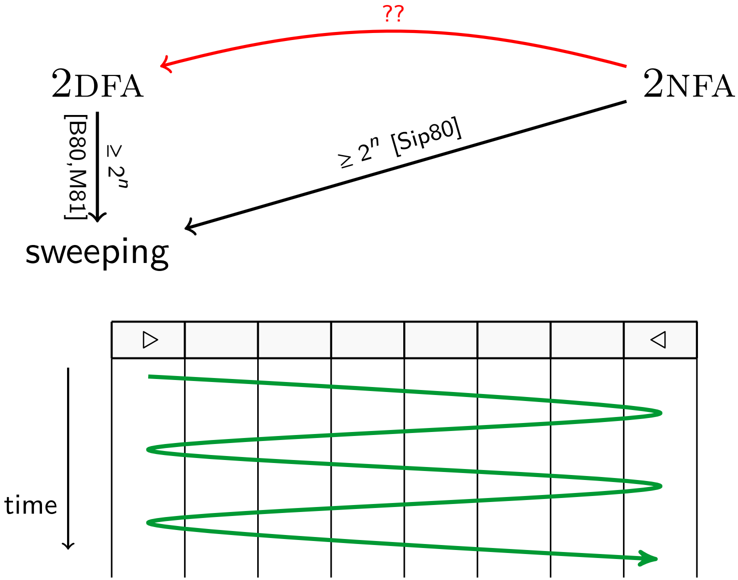
Two-wayness and nondeterminism





- the optimal cost of the simulation of 2NFA by 2DFA?

Two-wayness and nondeterminism



- the optimal cost of the simulation of 2NFA by 2DFA?

Two-wayness and nondeterminism

$$
\begin{tikzcd}
2\textsc{dfa} & 2\textsc{onfa} & 2\textsc{nfa} \arrow[ll, bend right, "??"']
\end{tikzcd}
$$

- the optimal cost of the simulation of $2\textsc{nfa}$ by $2\textsc{dfa}$?

Two-wayness and nondeterminism

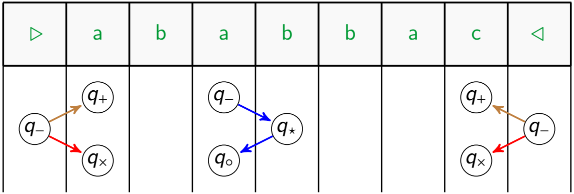
$$\begin{tikzcd} 2\textsc{dfa} & & 2\textsc{onfa} \arrow[ll, bend left, "\text{sub-exponential}"] & & 2\textsc{nfa} \arrow[llll, bend right, "??"'] \end{tikzcd}$$

- the optimal cost of the simulation of $2\textsc{nfa}$ by $2\textsc{dfa}$?

Outer-nondeterministic finite automata

Definition (2ONFA)

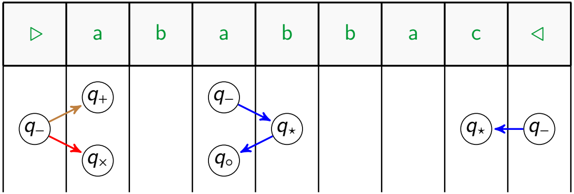
An 2-way automaton is outer-nondeterministic
if nondeterministic choices are restricted to the endmarkers only.



Outer-nondeterministic finite automata

Definition (2ONFA)

An 2-way automaton is outer-nondeterministic
if nondeterministic choices are restricted to the endmarkers only.



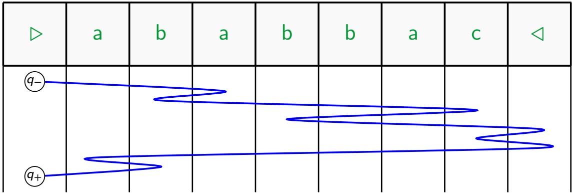
Proposition

With a linear increase of the number of states,
nondeterministic choices are restricted to the left endmarker only.

Outer-nondeterministic finite automata

Definition (2ONFA)

An 2-way automaton is outer-nondeterministic
if nondeterministic choices are restricted to the endmarkers only.



Proposition

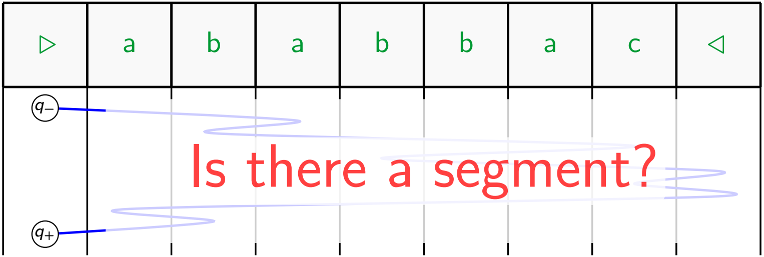
With a linear increase of the number of states, nondeterministic choices are restricted to the left endmarker only.

Definition

A segment is a computational path between two successive visits of the left endmarker.

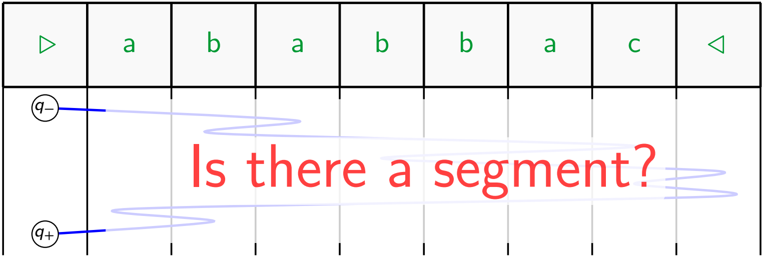
Key point

Given q_- and q_+:



Key point

Given q_- and q_+:



Proposition

Answer with a 2DFA *of linear size.*

Proof.

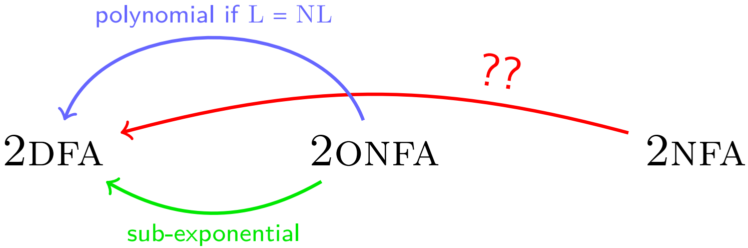
Adapt a Sipser's construction to avoid deterministic central loops. □

$$\begin{tikzcd}
\mathrm{2DFA} & \mathrm{2ONFA} & \mathrm{2NFA} \arrow[ll, bend right, "??"']
\end{tikzcd}$$

$$\begin{tikzcd}
\mathrm{2DFA} & \mathrm{2ONFA} \arrow[l, bend left, "\text{sub-exponential}"] & \mathrm{2NFA} \arrow[ll, bend right, "??"']
\end{tikzcd}$$

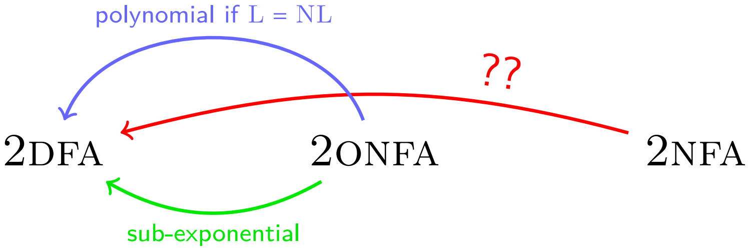
Theorem

- *Sub-exponential simulation of* 2ONFA *by* 2DFA $\mathcal{O}(n^{\log_2(n)+7})$.



Theorem

- *Sub-exponential simulation of* 2ONFA *by* 2DFA $\mathcal{O}(n^{\log_2(n)+7})$.
  - *polynomial if* L = NL.

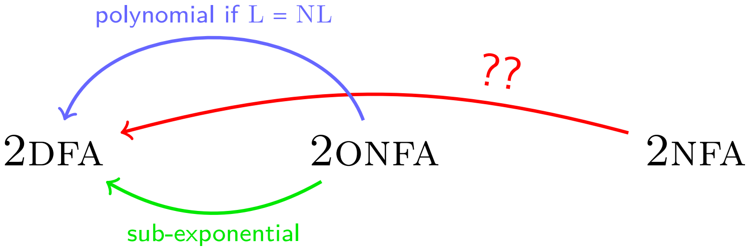


Theorem

- *Sub-exponential simulation of* 2ONFA *by* 2DFA $\mathcal{O}(n^{\log_2(n)+7})$.
    - *polynomial if* $\mathrm{L} = \mathrm{NL}$.

Further results

- *Simulation by unambiguous* 2ONFA *of polynomial size.*



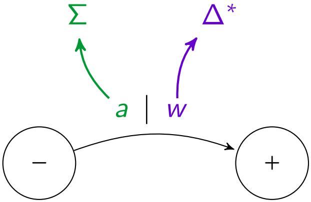
Theorem

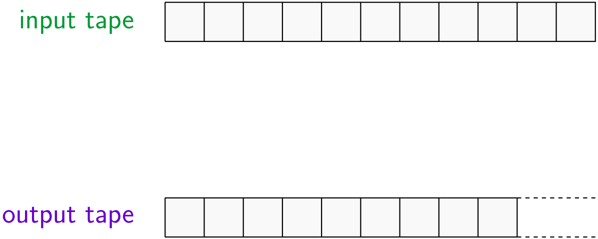
- *Sub-exponential simulation of* 2ONFA *by* 2DFA $\mathcal{O}(n^{\log_2(n)+7})$.
  - *polynomial if* L = NL.

Further results

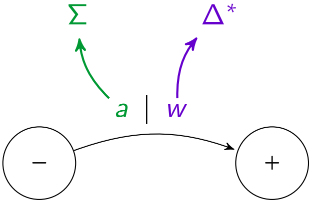
- *Simulation by unambiguous* 2ONFA *of polynomial size.*
- *Simulation by a halting* 2ONFA *of polynomial size.*
- *Complementation by a halting* 2ONFA *of polynomial size.*

Automata with output: 1-way transducers





Automata with output: 1-way transducers

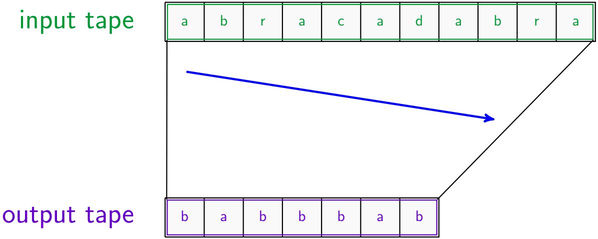


Example

u →

- replace a by b
- replace b by a
- ignore other letters

→ v



Equivalent formalisms

- Relations on words:

$$R \subseteq \Sigma^* \times \Delta^*$$

Equivalent formalisms

- Relations on words:

$$R \subseteq \Sigma^* \times \Delta^*$$

- A function from words into languages:

$$f_R : \begin{array}{rcl} \Sigma^* & \to & 2^{\Delta^*} \\ u & \mapsto & \{v \mid (u, v) \in R\} \end{array}$$

Equivalent formalisms

- Relations on words:

$$R \subseteq \Sigma^* \times \Delta^*$$

- A function from words into languages:

$$f_R : \begin{array}{ccc} \Sigma^* & \to & 2^{\Delta^*} \\ u & \mapsto & \{v \mid (u, v) \in R\} \end{array}$$

- A formal power series:

$$\sigma = \sum_{u \in \Sigma^*} \langle \sigma, u \rangle u \qquad \text{with } \langle \sigma, u \rangle = f_R(u)$$

Equivalent formalisms

- Relations on words:

$$R \subseteq \Sigma^* \times \Delta^*$$

Transducers

- A function from words into languages:

$$f_R : \begin{array}{ccc} \Sigma^* & \to & 2^{\Delta^*} \\ u & \mapsto & \{v \mid (u,v) \in R\} \end{array}$$

- A formal power series:

$$\sigma = \sum_{u \in \Sigma^*} \langle \sigma, u \rangle u \qquad \text{with } \langle \sigma, u \rangle = f_R(u)$$

Equivalent formalisms

- Relations on words:

$$R \subseteq \Sigma^* \times \Delta^*$$

Transducers

- A function from words into languages:

$$f_R : \begin{array}{ccc} \Sigma^* & \to & 2^{\Delta^*} \\ u & \mapsto & \{v \mid (u, v) \in R\} \end{array}$$

- A formal power series:

$$\sigma = \sum_{u \in \Sigma^*} \langle \sigma, u \rangle u \qquad \text{with } \langle \sigma, u \rangle = f_R(u)$$

Weighted Automata

Rational operations

- Union $R_1 \cup R_2$
- Componentwise concatenation

$$R_1 \cdot R_2 = \{(u_1 u_2, v_1 v_2) \mid (u_1, v_1) \in R_1 \text{ and } (u_2, v_2) \in R_2\}$$

- Kleene star

$$R^* = \{(u_1 u_2 \cdots u_k, v_1 v_2 \cdots v_k) \mid \forall i \ (u_i, v_i) \in R\}$$

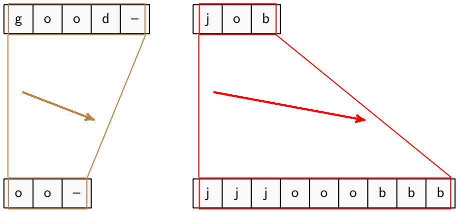
Rational operations

- Union $\qquad R_1 \cup R_2$
- Componentwise concatenation

$$R_1 \cdot R_2 = \{(u_1 u_2, v_1 v_2) \mid (u_1, v_1) \in R_1 \text{ and } (u_2, v_2) \in R_2\}$$

- Kleene star

$$R^* = \{(u_1 u_2 \cdots u_k, v_1 v_2 \cdots v_k) \mid \forall i \ (u_i, v_i) \in R\}$$



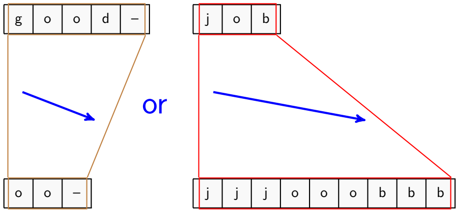
Rational operations

- Union $\qquad R_1 \cup R_2$
- Componentwise concatenation

$$R_1 \cdot R_2 = \{(u_1 u_2, v_1 v_2) \mid (u_1, v_1) \in R_1 \text{ and } (u_2, v_2) \in R_2\}$$

- Kleene star

$$R^* = \{(u_1 u_2 \cdots u_k, v_1 v_2 \cdots v_k) \mid \forall i \ (u_i, v_i) \in R\}$$



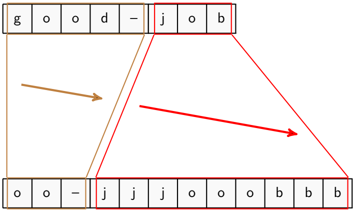
Rational operations

- Union $\quad R_1 \cup R_2$
- Componentwise concatenation

$$R_1 \cdot R_2 = \{(u_1 u_2, v_1 v_2) \mid (u_1, v_1) \in R_1 \text{ and } (u_2, v_2) \in R_2\}$$

- Kleene star

$$R^* = \{(u_1 u_2 \cdots u_k, v_1 v_2 \cdots v_k) \mid \forall i \ (u_i, v_i) \in R\}$$



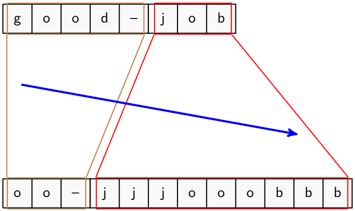
Rational operations

- Union $R_1 \cup R_2$
- Componentwise concatenation

$$R_1 \cdot R_2 = \{(u_1 u_2, v_1 v_2) \mid (u_1, v_1) \in R_1 \text{ and } (u_2, v_2) \in R_2\}$$

- Kleene star

$$R^* = \{(u_1 u_2 \cdots u_k, v_1 v_2 \cdots v_k) \mid \forall i \ (u_i, v_i) \in R\}$$



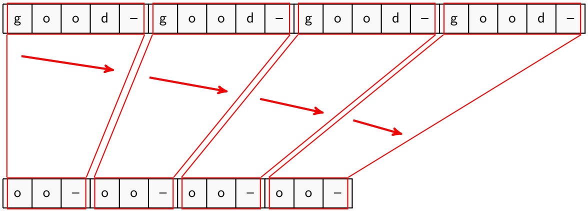
Rational operations

- Union $R_1 \cup R_2$
- Componentwise concatenation

$$R_1 \cdot R_2 = \{(u_1 u_2, v_1 v_2) \mid (u_1, v_1) \in R_1 \text{ and } (u_2, v_2) \in R_2\}$$

- Kleene star

$$R^* = \{(u_1 u_2 \cdots u_k, v_1 v_2 \cdots v_k) \mid \forall i \ (u_i, v_i) \in R\}$$



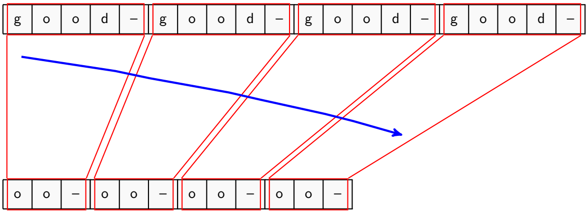
Rational operations

- Union $R_1 \cup R_2$
- Componentwise concatenation

$$R_1 \cdot R_2 = \{(u_1 u_2, v_1 v_2) \mid (u_1, v_1) \in R_1 \text{ and } (u_2, v_2) \in R_2\}$$

- Kleene star

$$R^* = \{(u_1 u_2 \cdots u_k, v_1 v_2 \cdots v_k) \mid \forall i \ (u_i, v_i) \in R\}$$



Rational operations

- Union $R_1 \cup R_2$
- Componentwise concatenation

$$R_1 \cdot R_2 = \{(u_1 u_2, v_1 v_2) \mid (u_1, v_1) \in R_1 \text{ and } (u_2, v_2) \in R_2\}$$

- Kleene star

$$R^* = \{(u_1 u_2 \cdots u_k, v_1 v_2 \cdots v_k) \mid \forall i\ (u_i, v_i) \in R\}$$

Definition ($\mathrm{RAT}(\Sigma^* \times \Delta^*)$)

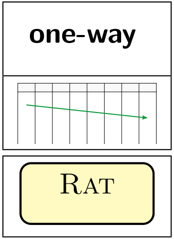
The family of Rational relations is the smallest family:

- including **finite relations**
- closed under **Rational operations**

One-way is rational

Theorem (Elgot, Mezei - 1965)

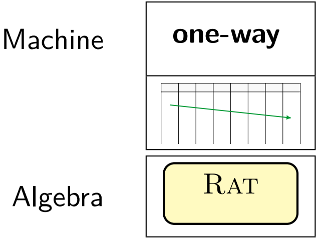
1-way transducers $=$ $\textsc{Rat}$.



One-way is rational

Theorem (Elgot, Mezei - 1965)

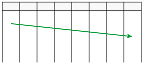
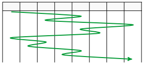
1-way transducers $=$ $\textsc{Rat}$.



What about two-way transducers?

Theorem (Elgot, Mezei - 1965)

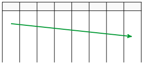
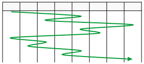
2-way transducers = ??

Machine	**one-way**	**two-way**
		
Algebra	RAT	??

What about two-way transducers?

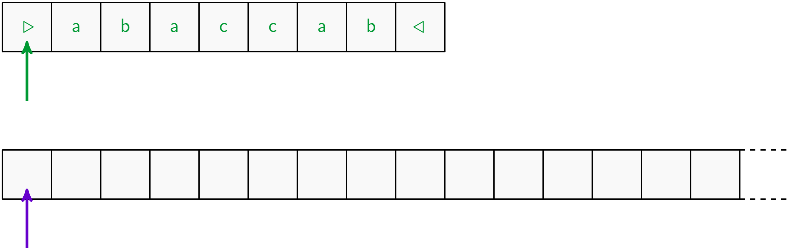
Theorem (Elgot, Mezei - 1965)

2-way transducers = ??

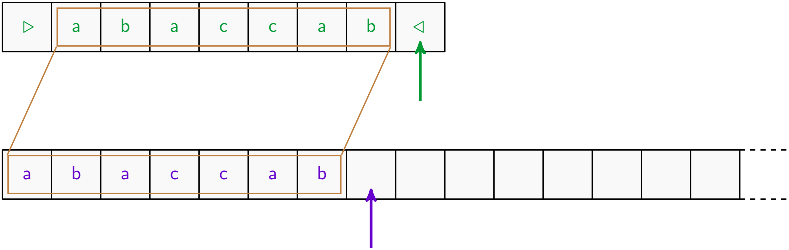
Machine	**one-way**	**two-way**
		
Algebra	RAT	??

Most of the known results on 2-way transducers concern the **functional** (=**deterministic**) case...

A simple example: $\textsc{Square} = \{(w, ww) \mid w \in \Sigma^*\}$

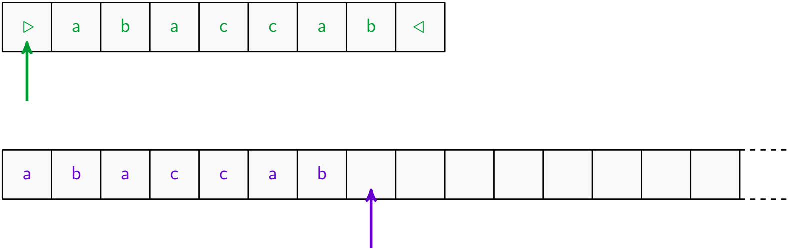


A simple example: $\textsc{Square} = \{(w, ww) \mid w \in \Sigma^*\}$



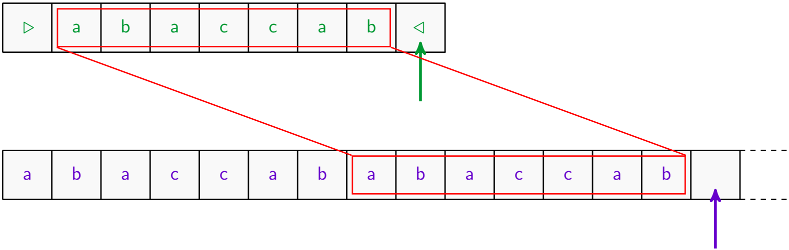
- copy the input word

A simple example: $\textsc{Square} = \{(w, ww) \mid w \in \Sigma^*\}$



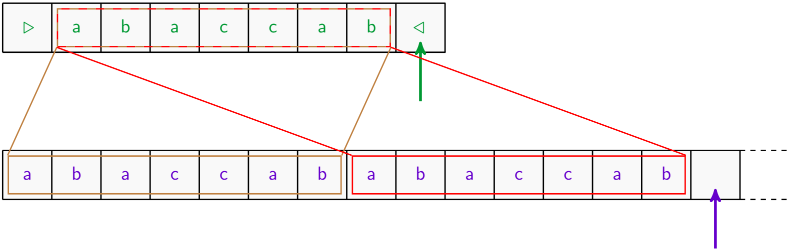
- copy the input word
- rewind the input tape

A simple example: $\textsc{Square} = \{(w, ww) \mid w \in \Sigma^*\}$



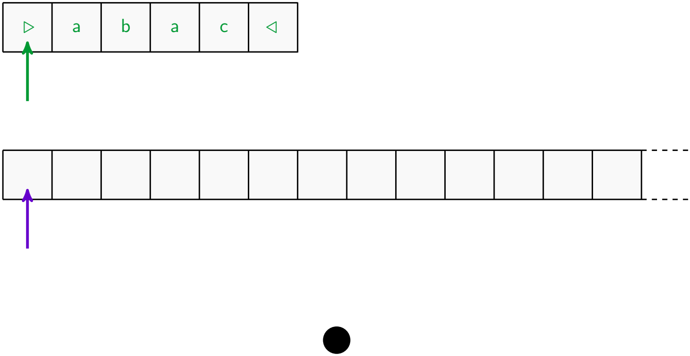
- copy the input word
- rewind the input tape
- append a copy of the input word

A simple example: $\textsc{Square} = \{(w, ww) \mid w \in \Sigma^*\}$

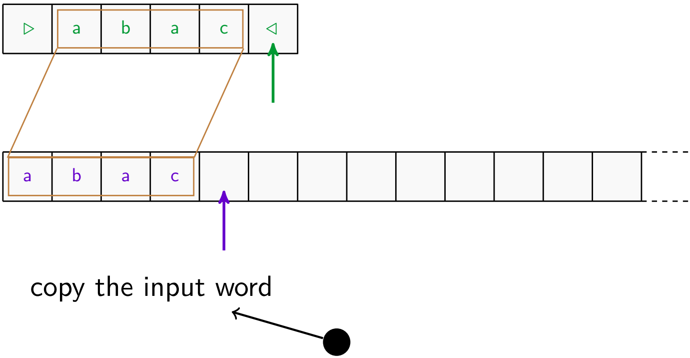


- copy the input word
- rewind the input tape
- append a copy of the input word

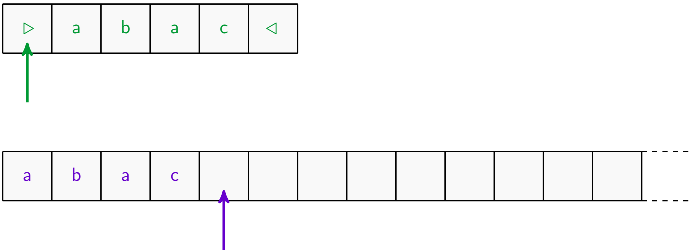
Another example: $\textsc{Powers} = \{(w, w^*) \mid w \in \Sigma^*\}$



Another example: $\textsc{Powers} = \{(w, w^*) \mid w \in \Sigma^*\}$

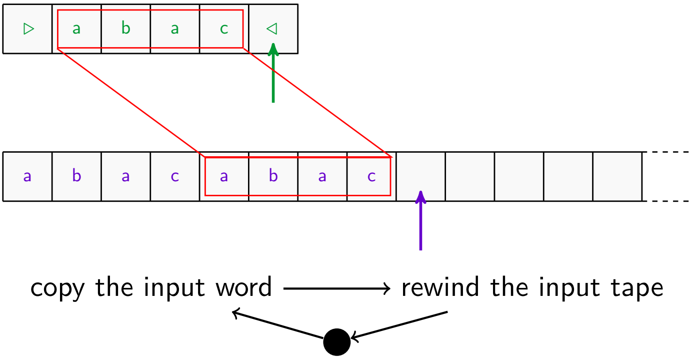


Another example: $\textsc{Powers} = \{(w, w^*) \mid w \in \Sigma^*\}$

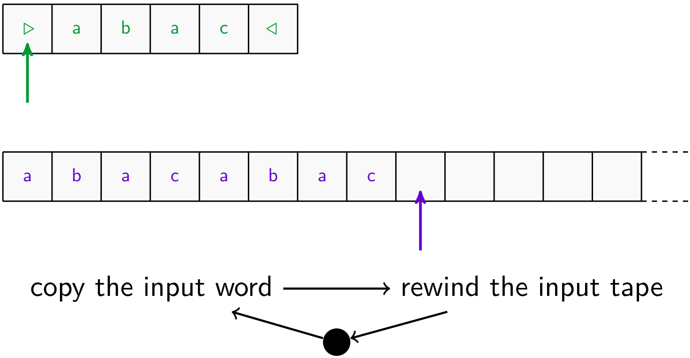


copy the input word $\longrightarrow$ rewind the input tape

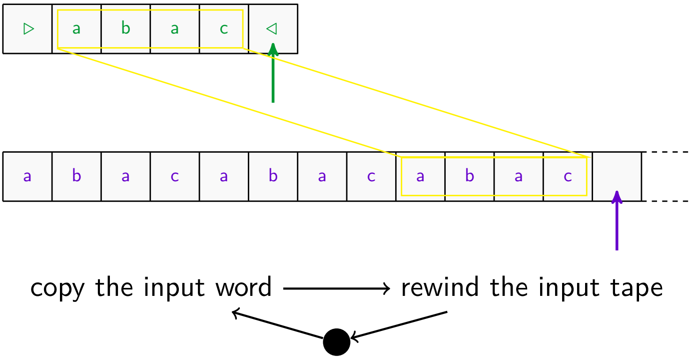
Another example: $\textsc{Powers} = \{(w, w^*) \mid w \in \Sigma^*\}$



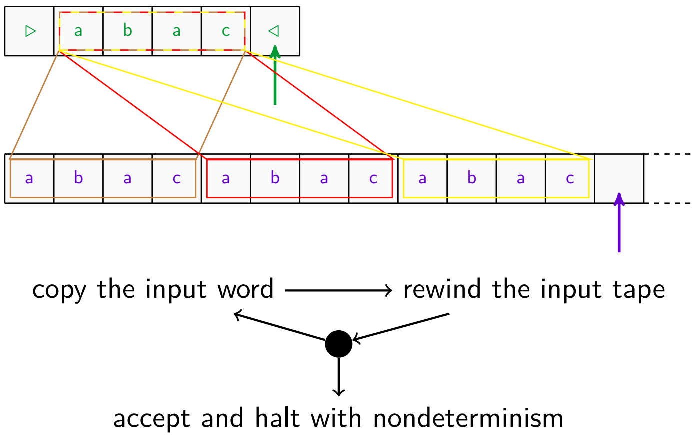
Another example: $\textsc{Powers} = \{(w, w^*) \mid w \in \Sigma^*\}$



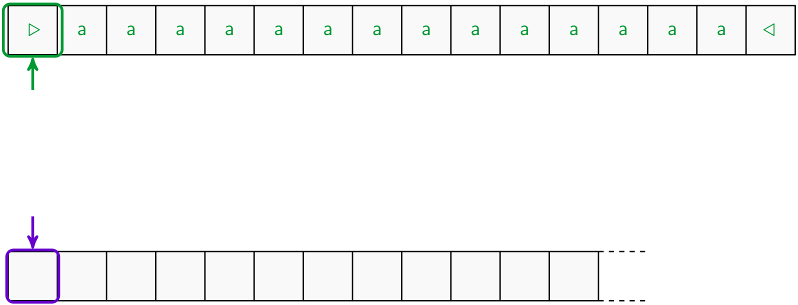
Another example: $\textsc{Powers} = \{(w, w^*) \mid w \in \Sigma^*\}$



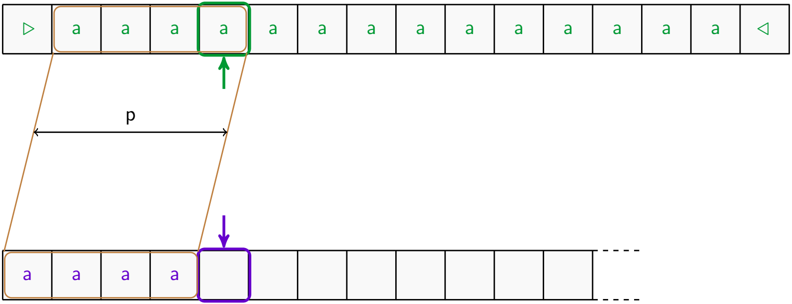
Another example: $\textsc{Powers} = \{(w, w^*) \mid w \in \Sigma^*\}$



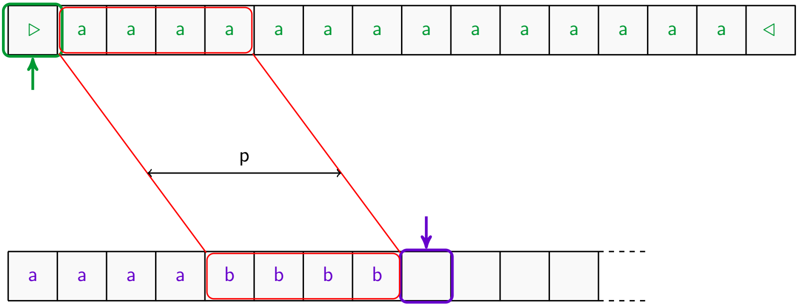
A last one: 2-$\textsc{Pref} = \{(a^n, a^p b^p) \mid p \leq n\}$



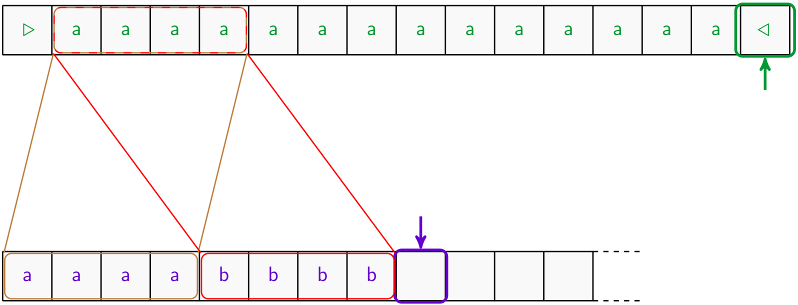
A last one: 2-$\text{PREF} = \{(a^n, a^p b^p) \mid p \leq n\}$



A last one: 2-$\textsc{Pref} = \{(a^n, a^p b^p) \mid p \leq n\}$



A last one: 2-$\textsc{Pref} = \{(a^n, a^p b^p) \mid p \leq n\}$



Hadamard operations

- Union $R_1 \cup R_2$

Hadamard operations

- Union $R_1 \cup R_2$
- H-product $R_1 \circledH R_2 = \{(u, v_1 v_2) \mid (u, v_1) \in R_1 \text{ and } (u, v_2) \in R_2\}$

Hadamard operations

- Union $R_1 \cup R_2$
- H-product $R_1 \circledcirc R_2 = \{(u, v_1 v_2) \mid (u, v_1) \in R_1 \text{ and } (u, v_2) \in R_2\}$

- simulate $\mathcal{T}_1$
- rewind the input tape
- simulate $\mathcal{T}_2$

example:

- $\textsc{Square} = \textsc{Id} \circledcirc \textsc{Id}$

Hadamard operations

- Union $R_1 \cup R_2$
- H-product $R_1 \circledcirc R_2 = \{(u, v_1 v_2) \mid (u, v_1) \in R_1 \text{ and } (u, v_2) \in R_2\}$
- H-star $R^{\mathsf{H}\star} = \{(u, v_1 v_2 \cdots v_k) \mid \forall i \ (u, v_i) \in R\}$

Hadamard operations

- Union $R_1 \cup R_2$
- H-product $R_1 \circledH R_2 = \{(u, v_1 v_2) \mid (u, v_1) \in R_1 \text{ and } (u, v_2) \in R_2\}$
- H-star $R^{\mathsf{H}\star} = \{(u, v_1 v_2 \cdots v_k) \mid \forall i\ (u, v_i) \in R\}$

- repeat
  - simulate $\mathcal{T}$
  - rewind the input tape
- or accept nondeterministically

example:

- $\textsc{Powers} = \textsc{Id}^{\mathsf{H}\star}$

Hadamard operations

- Union $R_1 \cup R_2$
- H-product $R_1 \circledast R_2 = \{(u, v_1 v_2) \mid (u, v_1) \in R_1 \text{ and } (u, v_2) \in R_2\}$
- H-star $R^{\mathsf{H}\star} = \{(u, v_1 v_2 \cdots v_k) \mid \forall i \; (u, v_i) \in R\}$

Definition ($\textsc{Had}(\Sigma^* \times \Delta^*)$)

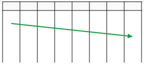
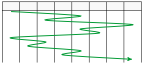
The family of Hadamard relations is the smallest family:

- including **rational relations**
- closed under **Hadamard operations**

What about two-way transducers?

Theorem

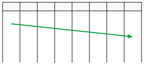
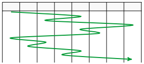
1-way transducers = RAT.

one-way	two-way
	
RAT	HAD??

What about two-way transducers?

Theorem

1-way transducers = RAT.

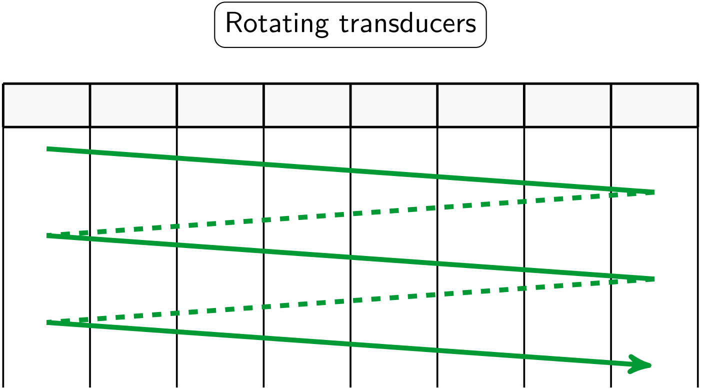
one-way	two-way
	
RAT	~~HAD??~~

ex: 2-PREF

$a^n \mapsto \{a^p b^p,\ p \leq n\}$

What about rotating transducers?

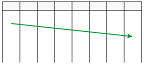
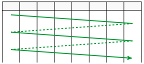
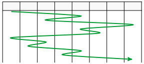
Theorem
1-way transducers $=$ $\textsc{Rat}$.



What about rotating transducers?

Theorem

rotating transducers = HAD.

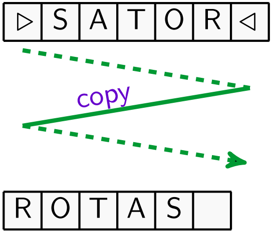
one-way	rotating	two-way
		
RAT	HAD	~~HAD??~~

ex: 2-PREF

$a^n \mapsto \{a^p b^p,\ p \leq n\}$

Right to left scan of the input

- Mirror operation:
  $\overline{R} = \{(\overline{u}, v) \mid (u, v) \in R\}$

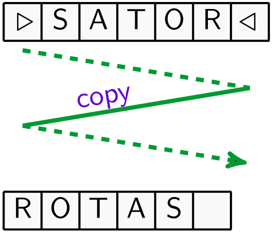


Example

$\overline{\mathrm{ID}} = \{w, \overline{w}\}$

Right to left scan of the input

- Mirror operation:
  $\overline{R} = \{(\overline{u}, v) \mid (u, v) \in R\}$



Example
$\overline{\mathrm{ID}} = \{w, \overline{w}\}$

Definition ($\textsc{MHad}(\Sigma^* \times \Delta^*)$)

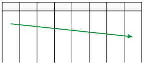
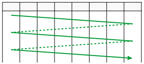
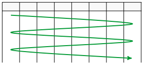
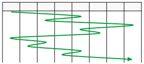
The family of Mirror-Hadamard relations is the smallest family:

- including **rational relations**
- closed under **Hadamard operations** and **mirror**

What about sweeping transducers?

Theorem

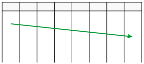
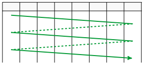
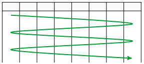
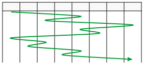
sweeping transducers = MHad.

one-way	rotating	sweeping	two-way
			
Rat	Had	MHad	MHad??

What about sweeping transducers?

Theorem

sweeping transducers = MHAD.

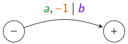
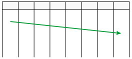
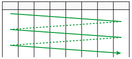
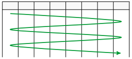
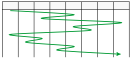
one-way	rotating	sweeping	two-way
			
RAT	HAD	MHAD	~~MHAD??~~

ex: 2-PREF $a^n \mapsto \{a^p b^p,\ p \leq n\}$

What about sweeping transducers?

Theorem

sweeping transducers = $\textsc{MHad}$.

transducer	one-way	rotating	sweeping	two-way
				
general	$\textsc{Rat}$	$\textsc{Had}$	$\textsc{MHad}$	$\cancel{\textsc{MHad}}$
input unary				

ex: 2-$\textsc{Pref}$ $a^n \mapsto \{a^p b^p,\ p \leq n\}$

Unary transducers

We focus on a weaker problem:

$$\Sigma = \{a\} \quad \text{and} \quad \Delta = \{a\}$$

Unary transducers

We focus on a weaker problem:

$$\Sigma = \{a\} \quad \text{and} \quad \Delta = \{a\}$$

Examples

- $\textsc{uId} = \{(a^n, a^n) \mid n \in \mathbb{N}\}$ $\in \textsc{Rat}$

Unary transducers

We focus on a weaker problem:

$$\Sigma = \{a\} \quad \text{and} \quad \Delta = \{a\}$$

Examples

- $\textsc{uId} = \{(a^n, a^n) \mid n \in \mathbb{N}\}$ $\in \textsc{Rat}$
- $\textsc{uSquare} = \textsc{uId} \circledast \textsc{uId} = \{(a^n, a^{2n}) \mid n \in \mathbb{N}\}$ $\in \textsc{Rat}$

Unary transducers

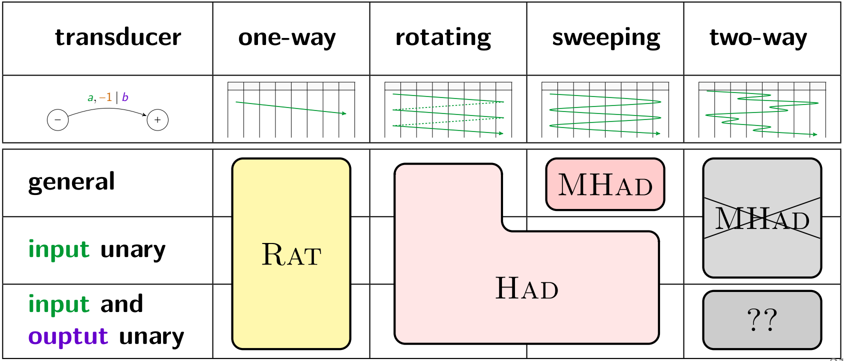
We focus on a weaker problem:

$$\Sigma = \{a\} \quad \text{and} \quad \Delta = \{a\}$$

Examples

- $\textsc{uId} = \{(a^n, a^n) \mid n \in \mathbb{N}\} \qquad \in \textsc{Rat}$
- $\textsc{uSquare} = \textsc{uId} \circledH \textsc{uId} = \{(a^n, a^{2n}) \mid n \in \mathbb{N}\} \qquad \in \textsc{Rat}$
- $\textsc{uPowers} = \textsc{uId}^{\mathsf{H}\star} = \{(u^n, u^{kn}) \mid k, n \in \mathbb{N}\} \quad \in \textsc{Had} \smallsetminus \textsc{Rat}$

Characterization of unary two-way transductions

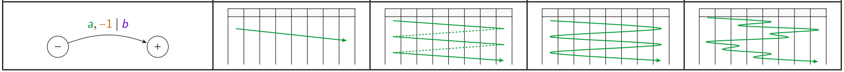


Characterization of unary two-way transductions

Theorem

2-way unary transducers $=$ Had

transducer	one-way	rotating	sweeping	two-way



	one-way	rotating	sweeping	two-way
general	Rat	Had	MHad	~~MHad~~
input unary	Rat	Had	Had	~~MHad~~
input and ouptut unary	Rat	Had	Had	Had

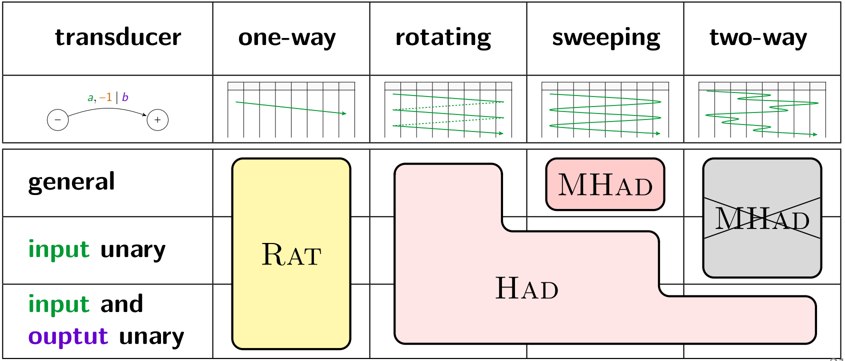
Characterization of unary two-way transductions

Theorem

2-way unary transducers $=$ Had

Corollary

2-way unary transducers $\longrightarrow$ *rotating transducers*.



Characterization of unary two-way transductions

Theorem

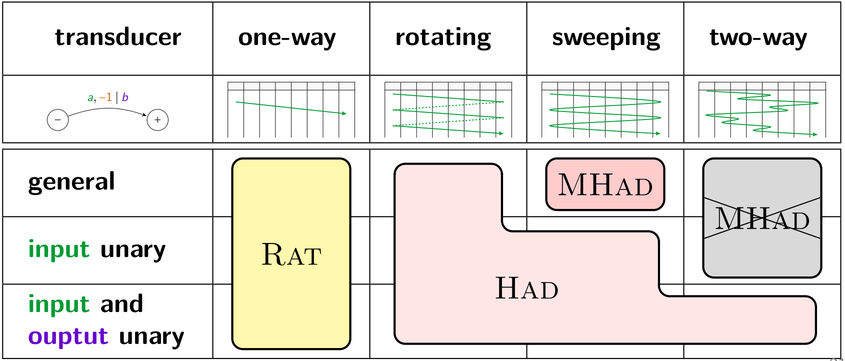
2-way unary transducers $=$ $\textsc{Had}$

Corollary

2-way unary transducers $\longrightarrow$ *rotating transducers*.

Example

$\textsc{uPowers} = \{(a^n, a^{kn}) \mid k, n \in \mathbb{N}\}$

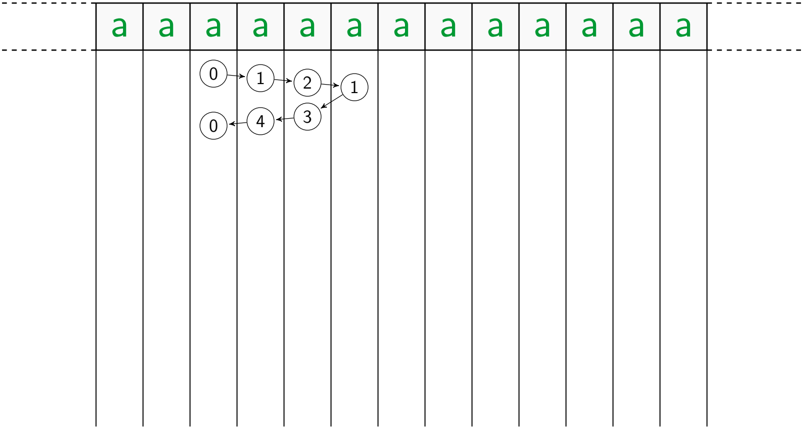


Key points of the proof

- commutative output
- deal with nondeterministic central loops ($\Sigma = \{a\}$ and $\Delta = \{a\}$).

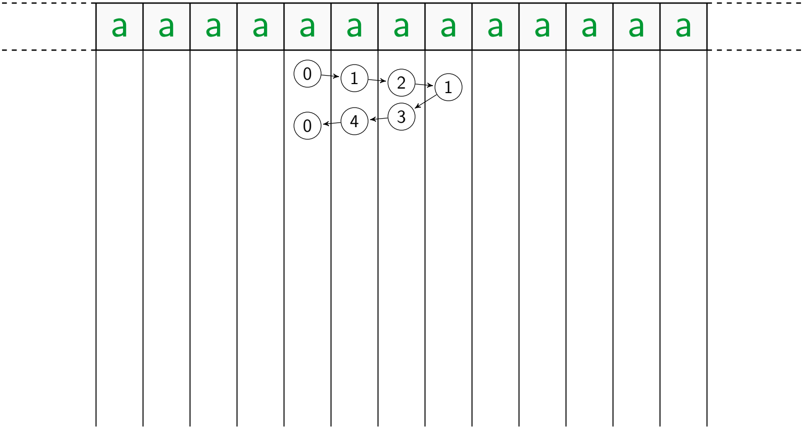
Key points of the proof

- commutative output
- deal with nondeterministic central loops ($\Sigma = \{a\}$ and $\Delta = \{a\}$).



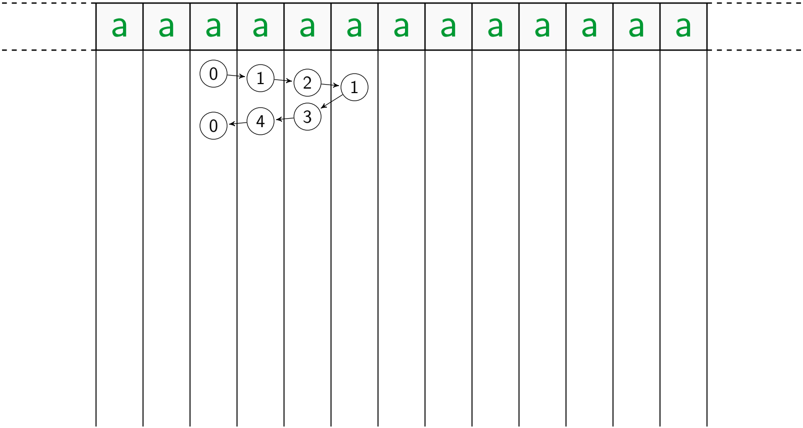
Key points of the proof

- commutative output
- deal with nondeterministic central loops ($\Sigma = \{a\}$ and $\Delta = \{a\}$).



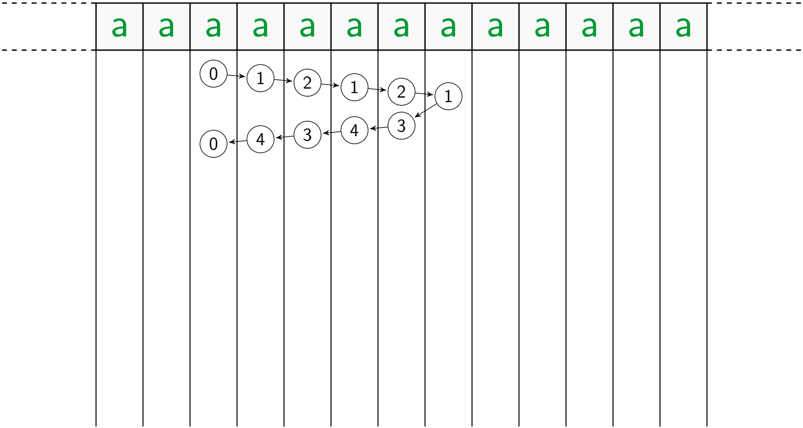
Key points of the proof

- commutative output
- deal with nondeterministic central loops ($\Sigma = \{a\}$ and $\Delta = \{a\}$).



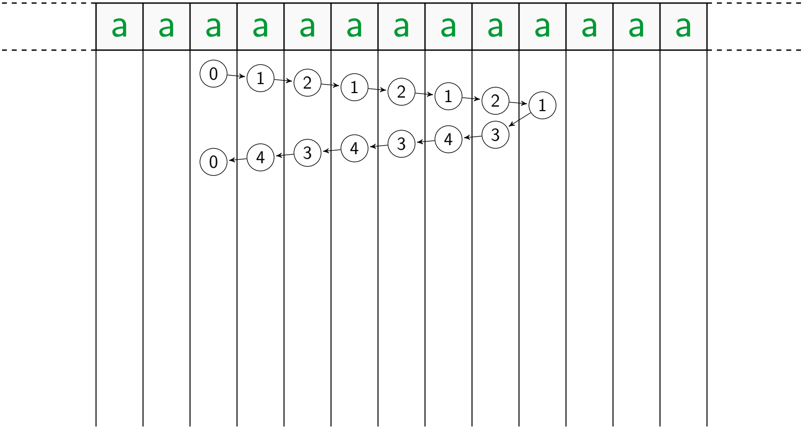
Key points of the proof

- commutative output
- deal with nondeterministic central loops ($\Sigma = \{a\}$ and $\Delta = \{a\}$).



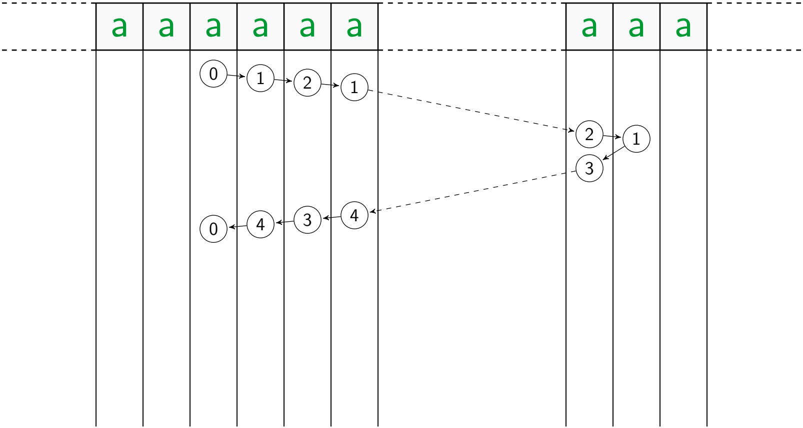
Key points of the proof

- commutative output
- deal with nondeterministic central loops ($\Sigma = \{a\}$ and $\Delta = \{a\}$).



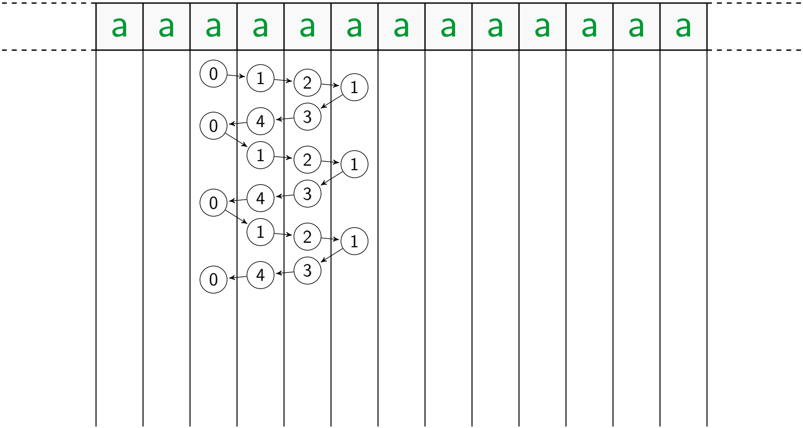
Key points of the proof

- commutative output
- deal with nondeterministic central loops ($\Sigma = \{a\}$ and $\Delta = \{a\}$).



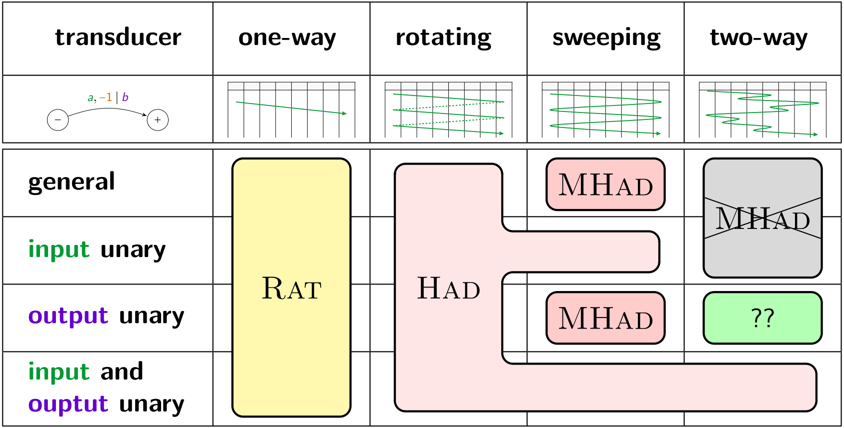
Key points of the proof

- commutative output
- deal with nondeterministic central loops ($\Sigma = \{a\}$ and $\Delta = \{a\}$).



The output-unary case

arbitrary Σ and $\Delta = \{a\}$

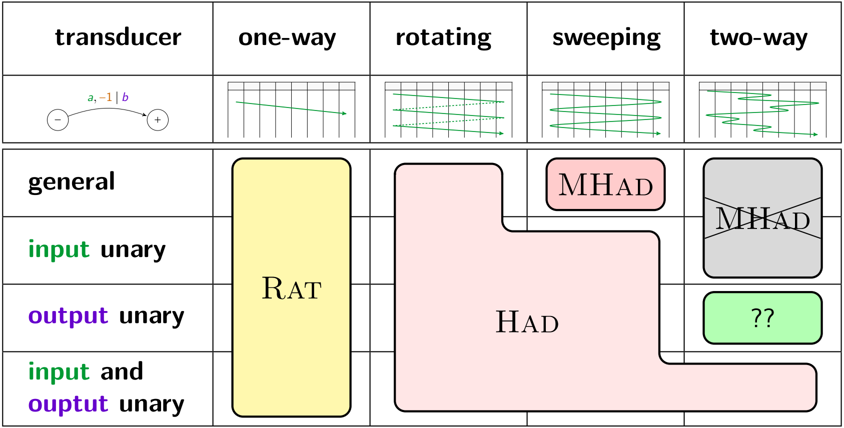


The output-unary case

arbitrary Σ and $\Delta = \{a\}$

Proposition

$\textsc{Had} = \textsc{MHad}$



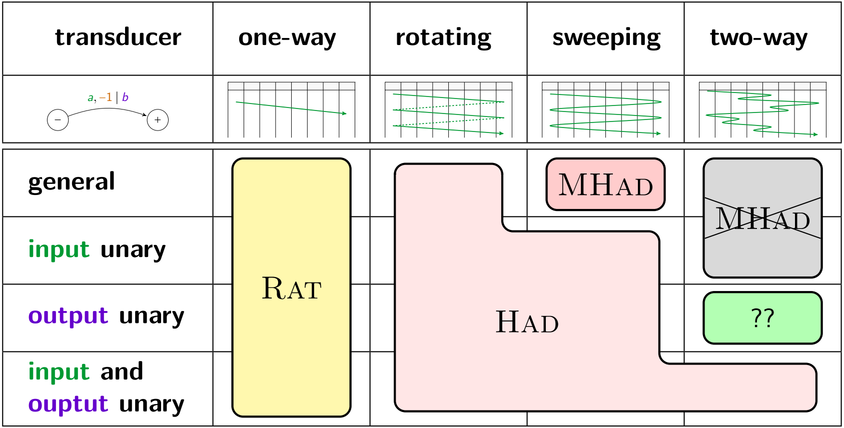
The output-unary case

arbitrary Σ and $\Delta = \{a\}$

Proposition

$\in \textsc{Rat}$

$$\textsc{Had} = \textsc{MHad} = \bigcup_{\mathit{finite}} R \;Ⓗ\; S^{H\star}$$



The output-unary case

arbitrary Σ and $\Delta = \{a\}$

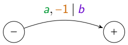
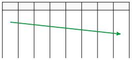
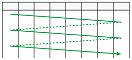
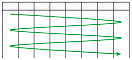
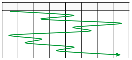
Proposition

$\in \textsc{Rat}$

$$\textsc{Had} = \textsc{MHad} = \bigcup_{\textit{finite}} R \circledast S^{H\star}$$

Theorem

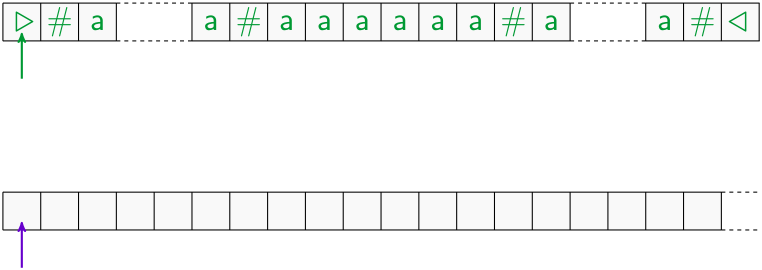
2-way output-unary $\neq$ $\textsc{Had}$

transducer	one-way	rotating	sweeping	two-way
				
general	Rat	Had	MHad	~~MHad~~
input unary	Rat	Had	Had	~~MHad~~
output unary	Rat	Had	Had	~~MHad~~
input and ouptut unary	Rat	Had	Had	Had

An non-Hadamard output-unary transduction

$\Sigma = \{a, \#\}$ and $\Delta = \{a\}$

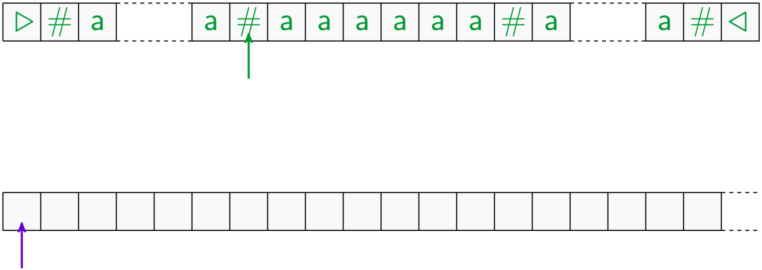
$$R = \left\{ \left(u, a^{kn} \right) \mid k, n \in \mathbb{N},\ \#a^k\# \text{ is a factor of } u \right\}$$



An non-Hadamard output-unary transduction

$\Sigma = \{a, \#\}$ and $\Delta = \{a\}$

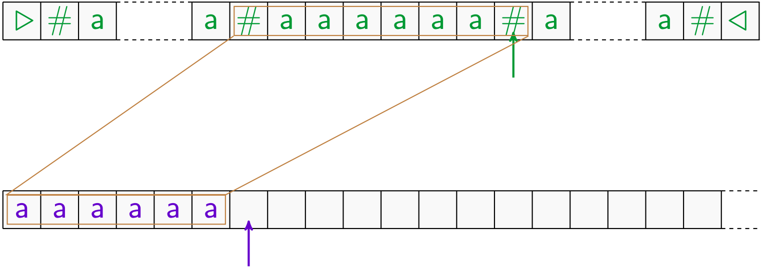
$$R = \left\{ \left(u, a^{kn} \right) \mid k, n \in \mathbb{N},\ \#a^k\# \text{ is a factor of } u \right\}$$



An non-Hadamard output-unary transduction

$\Sigma = \{a, \#\}$ and $\Delta = \{a\}$

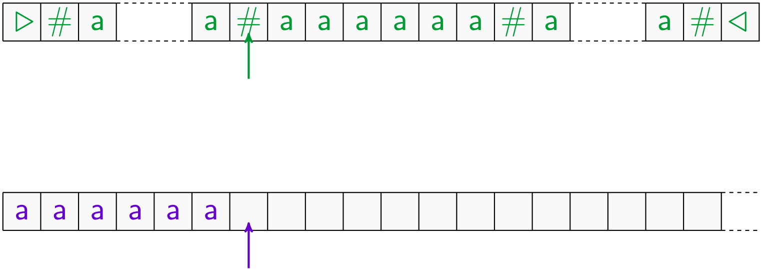
$$R = \left\{ \left(u, a^{kn} \right) \mid k, n \in \mathbb{N},\ \#a^k\# \text{ is a factor of } u \right\}$$



An non-Hadamard output-unary transduction

$\Sigma = \{a, \#\}$ and $\Delta = \{a\}$

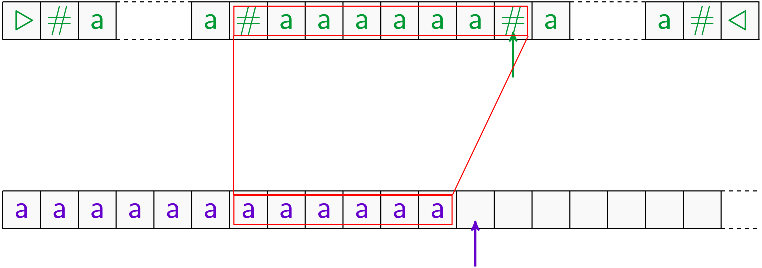
$$R = \left\{ \left(u, a^{kn} \right) \mid k, n \in \mathbb{N},\ \#a^k\# \text{ is a factor of } u \right\}$$



An non-Hadamard output-unary transduction

$$\Sigma = \{a, \#\} \qquad \text{and} \qquad \Delta = \{a\}$$

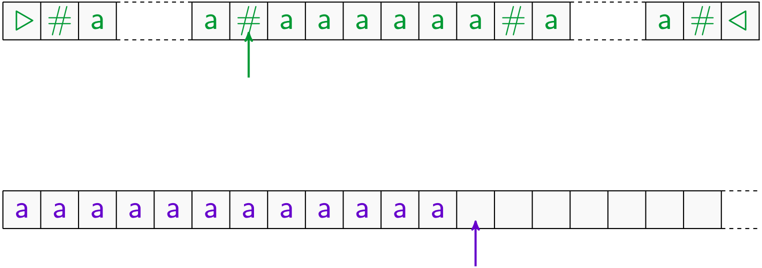
$$R = \left\{ \left(u, a^{kn}\right) \mid k, n \in \mathbb{N},\ \#a^k\# \text{ is a factor of } u \right\}$$



An non-Hadamard output-unary transduction

$\Sigma = \{a, \#\}$ and $\Delta = \{a\}$

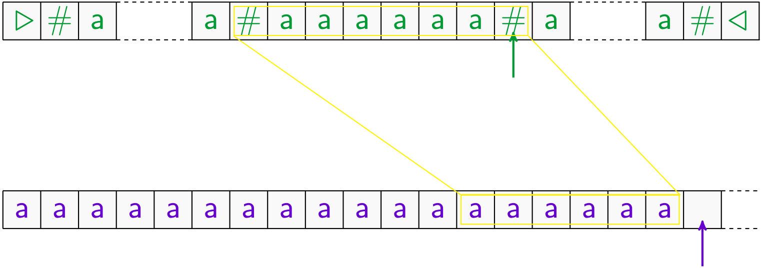
$$R = \Big\{ \big(u, a^{kn}\big) \mid k, n \in \mathbb{N},\ \#a^k\# \text{ is a factor of } u \Big\}$$



An non-Hadamard output-unary transduction

$\Sigma = \{a, \#\}$ and $\Delta = \{a\}$

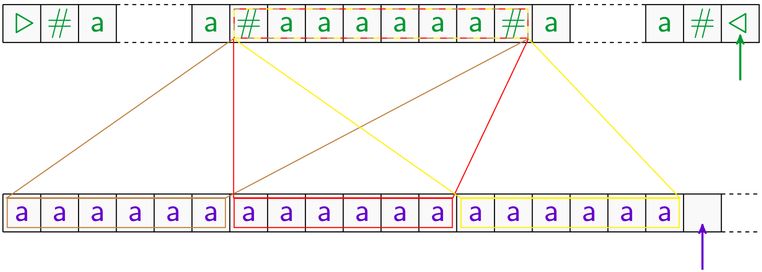
$$R = \left\{ \left(u, a^{kn} \right) \mid k, n \in \mathbb{N},\ \#a^k\# \text{ is a factor of } u \right\}$$



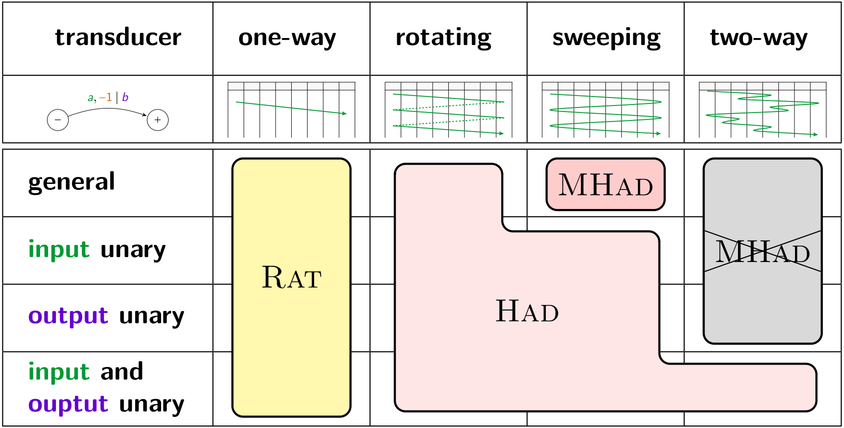
An non-Hadamard output-unary transduction

$\Sigma = \{a, \#\}$ and $\Delta = \{a\}$

$$R = \left\{\left(u, a^{kn}\right) \mid k, n \in \mathbb{N},\ \#a^k\# \text{ is a factor of } u\right\}$$

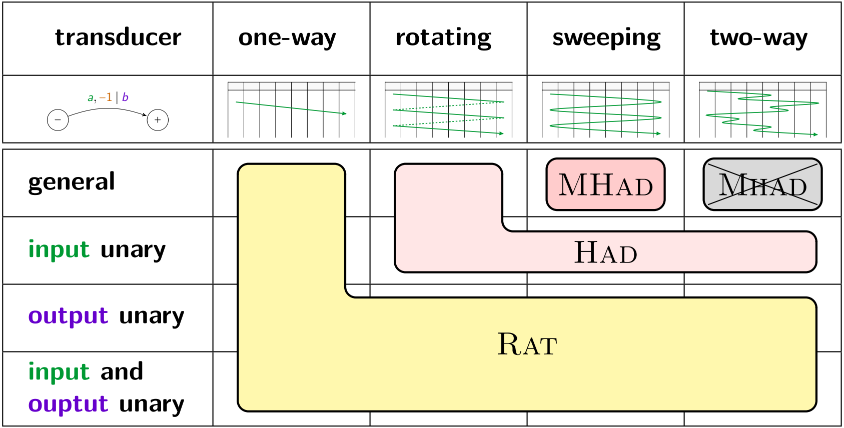


Two-way transducers VERSUS Algebra



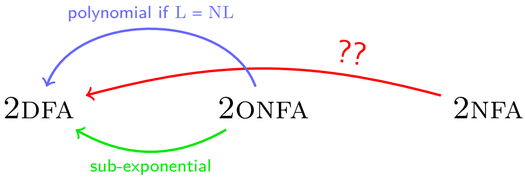
Two-way transducers VERSUS Algebra

functional case

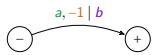
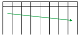
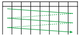
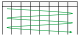
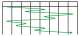


Conclusion

Descriptional complexity

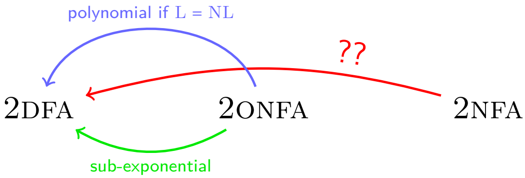


Two-way transducers

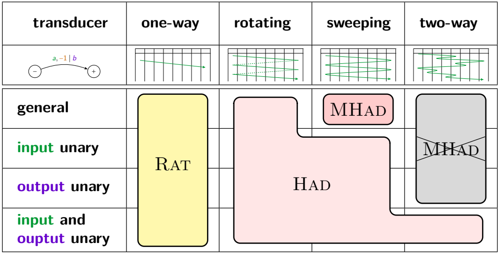
transducer	one-way	rotating	sweeping	two-way
				
general	RAT	HAD	MHAD	~~MHAD~~
input unary	RAT	HAD	HAD	~~MHAD~~
output unary	RAT	HAD	HAD	~~MHAD~~
input and ouptut unary	RAT	HAD	HAD	HAD

Conclusion

Descriptional complexity



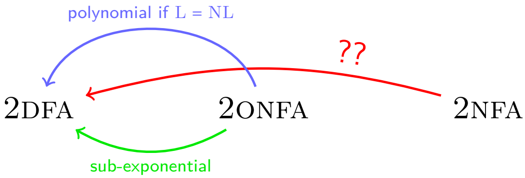
Two-way transducers



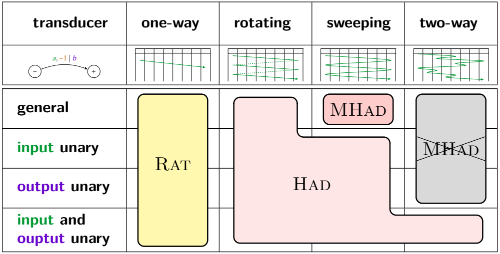
- **Alternating 2onfa**

Conclusion

Descriptional complexity



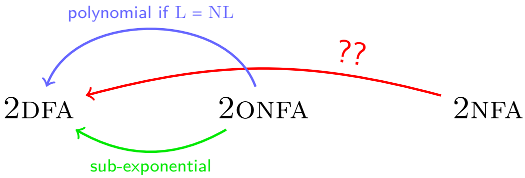
Two-way transducers



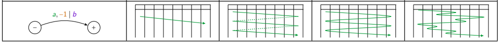
- **Alternating 2onfa**
- Other restrictions on nondeterminism of 2NFA

Conclusion

Descriptional complexity



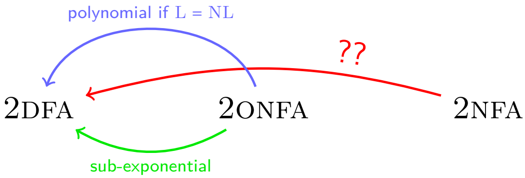
Two-way transducers

transducer	one-way	rotating	sweeping	two-way
				
general			MHAD	
input unary				~~MHAD~~
output unary	RAT	HAD		
input and ouptut unary				

- **Alternating 2onfa**
- Other restrictions on nondeterminism of 2NFA
- **Uniformization**

Conclusion

Descriptional complexity



Two-way transducers

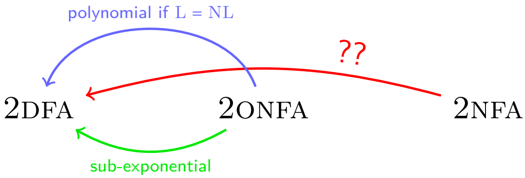
transducer	one-way	rotating	sweeping	two-way
general			MHAD	
input unary	RAT			~~MHAD~~
output unary		HAD		
input and ouptut unary				

- **Alternating 2onfa**
- Other restrictions on nondeterminism of 2NFA

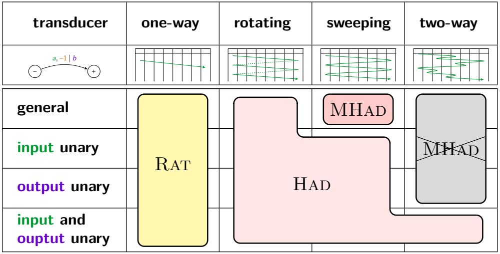
- **Uniformization**
- **Composition** $R_1 \circ R_2$
- **Transitive closure**

Conclusion

Descriptional complexity



Two-way transducers



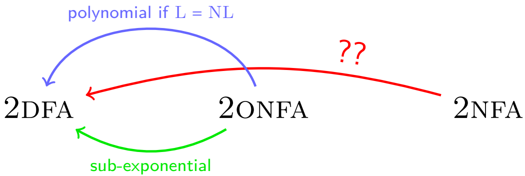
- Alternating 2onfa
- Other restrictions on nondeterminism of 2NFA

- Uniformization
- Composition $R_1 \circ R_2$
- Transitive closure

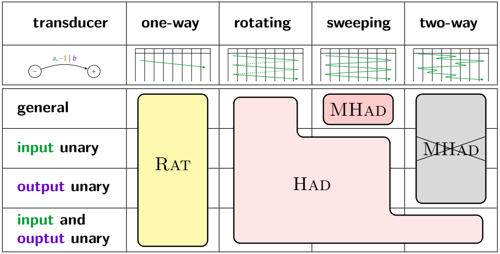
- Extend to series

Conclusion

Descriptional complexity



Two-way transducers



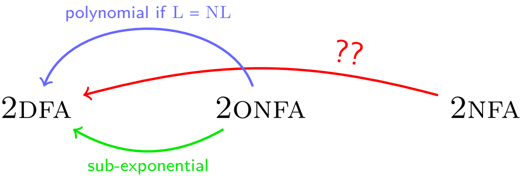
- **Alternating 2onfa**
- Other restrictions on nondeterminism of 2NFA

- **Uniformization**
- **Composition** $R_1 \circ R_2$
- **Transitive closure**

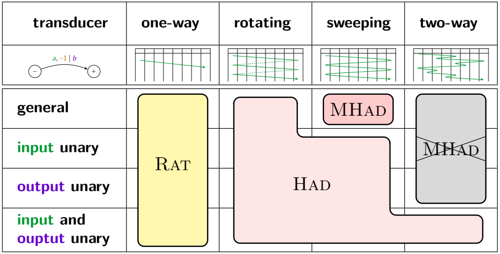
- Extend to series
- Describe the mirror

Conclusion

Descriptional complexity



Two-way transducers



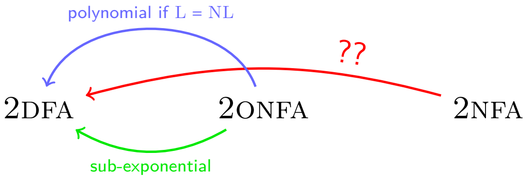
- **Alternating 2onfa**
- Other restrictions on nondeterminism of 2NFA

- **Uniformization**
- **Composition** $R_1 \circ R_2$
- **Transitive closure**

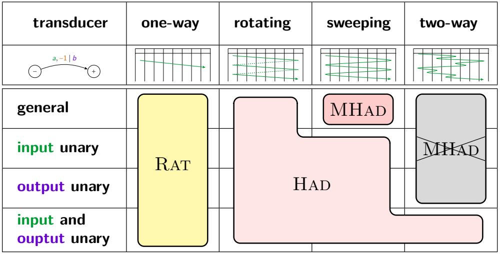
- Extend to series
- Describe the mirror
- Cost of simulations

Conclusion

Descriptional complexity



Two-way transducers



- **Alternating 2onfa**
- Other restrictions on nondeterminism of 2NFA

- **Uniformization**
- **Composition** $R_1 \circ R_2$
- **Transitive closure**

- Extend to series
- Describe the mirror
- Cost of simulations

Thanks for your attention