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
Introduction
  Computation
  Turing machines
  Finite automata

Descriptive complexity of finite automata
  Main questions and known results
  Outer-nondeterministic finite automata
  Determinization of outer-nondeterministic finite automata

Transducers
  One-way transducers
  Two-way transducers
  Hadamard operations
  Mirror operation
  Unary transducers

Conclusion
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 \( x \)
A computation is a sequence of successive elementary operations.

\[ f : x \mapsto 5x - 3 \]

Compute \( f(x) \)

\[ \text{with } + \text{ and } \times \]

- 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 ×*

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 ×

1. multiply by 5
2. add −3

with + only

1. 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
  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**

1. multiply by 5
2. add \(-3\)

**with + only**

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 ×**

- 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 

1. multiply by 5
2. add \(-3\)

**with** + only

1. start with \( x \)
2. add \( x \)
3. add \( x \)
4. add \( x \)
5. add \(-3\)
Computation

A computation is a sequence of successive elementary operations.

\[ g : x \mapsto x^2 + x \quad \text{and} \quad f : x \mapsto 5x - 3 \]

Compute \( f(x) \)
with + and \( \times \)

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

Compute \( f(x) \)
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.

\[ g : x \mapsto x^2 + x \quad \quad f : x \mapsto 5x - 3 \]

Compute \( f(x) \)

- start with \( x \)
  - 1. multiply by 5
  - 2. add \(-3\)

Compute \( g(x) \)

- start with \( x \)
Computation

A computation is a sequence of successive elementary operations.

\[ g : x \mapsto x^2 + x \quad \text{ and } \quad f : x \mapsto 5x - 3 \]

Compute \( f(x) \)

with + and ×

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Multiply by 5</td>
</tr>
<tr>
<td>2</td>
<td>Add –3</td>
</tr>
</tbody>
</table>

with + only

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Add ( x )</td>
</tr>
<tr>
<td>2</td>
<td>Add ( x )</td>
</tr>
<tr>
<td>3</td>
<td>Add ( x )</td>
</tr>
<tr>
<td>4</td>
<td>Add ( x )</td>
</tr>
<tr>
<td>5</td>
<td>Add –3</td>
</tr>
</tbody>
</table>

Compute \( g(x) \)

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Multiply by ( x )</td>
</tr>
</tbody>
</table>

|
Computation

A computation is a sequence of successive elementary operations.

\[ g : x \mapsto x^2 + x \]
\[ f : x \mapsto 5x - 3 \]

**Compute** \( f(x) \)

**with** + and \( \times \)

1. start with \( x \)
2. multiply by 5
3. add \( -3 \)

**Compute** \( g(x) \)

1. start with \( x \)
2. multiply by \( x \)
3. add \( x \)

**with** + only

1. start with \( x \)
2. add \( x \)
3. add \( x \)
4. add \( x \)
5. add \( -3 \)
Computation

A computation is a sequence of successive elementary operations.

\[ g: x \mapsto x^2 + x \quad \quad f: x \mapsto 5x - 3 \]

Compute \( f(x) \)

- start with \( x \)
  1. multiply by 5
  2. 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.

\[ g : x \mapsto x^2 + x \quad \quad \quad f : x \mapsto 5x - 3 \]

Compute \( f(x) \)

**with + and \times**

1. start with \( x \)
2. multiply by 5
3. add \( x \)
4. add \( x \)
5. add \( x \)
6. add \( x \)
7. add \( -3 \)

**with + only**

1. start with \( x \)
2. add \( x \)
3. add \( x \)
4. add \( x \)
5. add \( -3 \)

Compute \( g(x) \)

**with**

1. start with \( x \)
2. multiply by \( x \)
3. add \( x \)

**Impossible**
Turing machines

internal state: A
Turing machines

```plaintext
internal state: 0 0 1 0 0 1 1 0 1 1 1
```

```
A
```
Turing machines

internal state: B

A

0 | 1, →

B
Turing machines

internal state: B

0 0 1 0 1 1 1 0 1 1 1

A

read

B

or

C

A B

0 | 1, →

0 | 1, ←

1 | 1, ←

1 | 1, →
Turing machines

0 0 1 0 1 1 1 0 1 1 1

internal state: A

A
B

0 | 1, →

1 | 1, ←

nondeterministic choice:
Turing machines

| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |

Internal state: A

Input: 0 0 1 0 1 1 1 0 1 1 1

Transitions:
- 1 | 1, ←
- 0 | 1, →
- 1 | 0, →
- 1 | 1, ←
Turing machines

Internal state: A

Nondeterministic choice: □ or □
Turing machines

internal state: A

nondeterministic choice: □ or □
Turing machines

internal state: A

nondeterministic choice: □ or □
Turing machines

internal state: \( C \)

nondeterministic choice: \( \square \) or \( \square \)
Turing machines

internal state: C

A | 1, ←
B | 0, →

C | 1, →
A | 0, ←
B | 1, →

1 0 1 0 0 1 1 0 1 1 1
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

internal state: C

Huge computational power
- infinite memory
- universal

Complex dynamics
- undecidability of the halting problem

Huge computational power
Turing machines

Huge computational power
- infinite memory
- universal

Complex dynamics
- undecidability of the halting problem
- contribution of nondeterminism

\[ e.g., \ P \neq NP \text{ and } L \neq NL \]
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.

\[
a \quad | \quad b
\]

\[\text{A} \quad \rightarrow \quad \text{B}\]
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.

\[ a \mid b \rightarrow \]
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

![Diagram of a finite automaton accepting the language \( \{a, b\}^* \cdot a \cdot a \cdot b \cdot \{a, b\}^* \)]

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

Theorem (Kleene)

finite automata \( \equiv \) rational languages
Finite automata

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

Example

![Diagram of a finite automaton accepting \( \{a, b\}^* \cdot a \cdot a \cdot b \cdot \{a, b\}^* \)]

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

Theorem (Kleene)
finite automata \( \equiv \) rational languages

The smallest family including finite languages
closed under union, concatenation and Kleene star.
Two-wayness and nondeterminism
Two-wayness and nondeterminism

2DFA

2NFA

1DFA

1NFA

computationally equivalent

two-wayness

nondeterminism
Two-wayness and nondeterminism

natural simulations
Two-wayness and nondeterminism

\[ 2^n \text{ [MF71]} \]

\[ 2^{n^2} + 2 \]

\[ 2^n \text{ [MF71]} \]

\[ (n+1) \text{ [Kap05]} \]

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

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

2DFA

sweeping

2NFA

≥ $2^n$ [Sip80]

- the optimal cost of the simulation of 2NFA by 2DFA?
Two-wayness and nondeterminism

The two main questions (Sakoda & Sipser 1978)

- the optimal cost of the simulation of $2\text{NFA}$ by $2\text{DFA}$?

$2\text{DFA}$ $\geq 2^n$ $2\text{NFA}$

sweeping

$\geq 2^n$ [Sip80]
Two-wayness and nondeterminism

- the optimal cost of the simulation of $2\text{NFA}$ by $2\text{DFA}$?
Two-wayness and nondeterminism

2DFA 2ONFA 2NFA

sub-exponential

• the optimal cost of the simulation of 2NFA by 2DFA?
Outer-nondeterministic finite automata

Definition (2ONFA)

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

![Diagram of a 2ONFA with states and transitions:]

- States: $q_-$, $q_+$, $q_\times$, $q_\circ$
- Transitions:
  - $q_-$ transitions to $q_+$ on $\triangleright$
  - $q_+$ transitions to $q_-\times$ on $\triangleleft$
  - $q_\times$ transitions to $q_\circ$ on $\triangleright$
  - $q_\circ$ transitions to $q_\times$ on $\triangleleft$

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.*
Outer-nondeterministic finite automata

Definition (2ONFA)

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

\[
\begin{array}{cccccccc}
\text{i} & a & b & a & b & b & a & c & \text{o} \\
q^- & \downarrow & \text{a} & \text{b} & \text{a} & \text{b} & \text{b} & \text{a} & \text{c} & \downarrow \\
q^+ & \text{a} & \text{b} & \text{a} & \text{b} & \text{b} & \text{a} & \text{c} & \text{o} \\
\end{array}
\]

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 $q_+$:

<table>
<thead>
<tr>
<th>△</th>
<th>a</th>
<th>b</th>
<th>a</th>
<th>b</th>
<th>b</th>
<th>a</th>
<th>c</th>
<th>△</th>
</tr>
</thead>
</table>

Is there a segment?
Key point

Given $q_-$ and $q_+$:

<table>
<thead>
<tr>
<th>▶</th>
<th>a</th>
<th>b</th>
<th>a</th>
<th>b</th>
<th>b</th>
<th>a</th>
<th>c</th>
<th>▼</th>
</tr>
</thead>
</table>

Is there a segment?

Proposition

*Answer with a 2DFA of linear size.*

Proof.

Adapt a Sipser’s construction to avoid deterministic central loops.
Theorem

Sub-exponential simulation of \(2\text{onfa}\) by \(2\text{dfa}\) \(O(n \log^2(n) + 7)\).

Further results

Simulation by unambiguous \(2\text{onfa}\) of polynomial size.

Simulation by a halting \(2\text{onfa}\) of polynomial size.

Complementation by a halting \(2\text{onfa}\) of polynomial size.
Theorem

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

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

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

Further results

- **Simulation by unambiguous \(2\text{ONFA} \) of polynomial size.**
Theorem

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

Further results

- Simulation by unambiguous $2\text{ONFA}$ of polynomial size.
- Simulation by a halting $2\text{ONFA}$ of polynomial size.
- Complementation by a halting $2\text{ONFA}$ of polynomial size.
Automata with output: 1-way transducers

Example:
- Replace $a$ by $b$
- Replace $b$ by $a$
- Ignore other letters

Input tape:

Output tape:
Automata with output: 1-way transducers

Example

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

input tape

output tape
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:
  \[ f_R : \Sigma^* \to 2^\Delta^* \]
  \[ u \mapsto \{ v \mid (u, v) \in R \} \]
Equivalent formalisms

- Relations on words:
  \[ R \subseteq \Sigma^* \times \Delta^* \]

- A function from words into languages:
  \[ f_R : \Sigma^* \rightarrow 2^{\Delta^*} \]
  \[ u \mapsto \{ v \mid (u, v) \in R \} \]

- A formal power series:
  \[ \sigma = \sum_{u \in \Sigma^*} \langle \sigma, u \rangle u \quad \text{with} \quad \langle \sigma, u \rangle = f_R(u) \]
Equivalent formalisms

- Relations on words:
  \[ R \subseteq \Sigma^* \times \Delta^* \]

- A function from words into languages:
  \[ f_R : \Sigma^* \rightarrow 2^{\Delta^*} \]
  \[ u \mapsto \{ v \mid (u, v) \in R \} \]

- A formal power series:
  \[ \sigma = \sum_{u \in \Sigma^*} \langle \sigma, u \rangle u \quad \text{with} \quad \langle \sigma, u \rangle = f_R(u) \]
Equivalent formalisms

- Relations on words:
  \[ R \subseteq \Sigma^* \times \Delta^* \]

- A function from words into languages:
  \[ f_R : \Sigma^* \rightarrow 2^{\Delta^*} \]
  \[ u \mapsto \{ v \mid (u, v) \in R \} \]

- A formal power series:
  \[ \sigma = \sum_{u \in \Sigma^*} \langle \sigma, u \rangle u \]
  with \( \langle \sigma, u \rangle = f_R(u) \)
Rational operations

- Union

- 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**

  \[
  R^* = \{ (u_1 u_2 \ldots u_k, v_1 v_2 \ldots v_k) \mid \forall i \ (u_i, v_i) \in R \}
  \]
Rational operations

- **Union**

- Componentwise concatenation

\[ R_1 \cdot R_2 = \{(u_1u_2, v_1v_2) \mid (u_1, v_1) \in R_1 \text{ and } (u_2, v_2) \in R_2\}\]

- Kleene star

\[ R^* = \{(u_1u_2\ldots u_k, v_1v_2\ldots v_k) \mid \forall i \ (u_i, v_i) \in R\}\]
Rational operations

- **Union**

- **Componentwise concatenation**
  \[
  R_1 \cdot R_2 = \{ (u_1u_2, v_1v_2) \mid (u_1, v_1) \in R_1 \text{ and } (u_2, v_2) \in R_2 \}
  \]

- **Kleene star**
  \[
  R^* = \{ (u_1u_2\cdots u_k, v_1v_2\cdots v_k) \mid \forall i (u_i, v_i) \in R \}
  \]
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

- 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**

- 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 (\( \mathcal{RAT} (\Sigma^* \times \Delta^*) \))

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 $\iff$ RAT.
One-way is rational

Theorem (Elgot, Mezei - 1965)

1-way transducers $\equiv$ RAT.
What about **two-way transducers**?

**Theorem (Elgot, Mezei - 1965)**

2-way transducers == ??

<table>
<thead>
<tr>
<th>Machine</th>
<th>one-way</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algebra</td>
<td>RAT</td>
<td>??</td>
</tr>
</tbody>
</table>
What about two-way transducers?

Theorem (Elgot, Mezei - 1965)

<table>
<thead>
<tr>
<th>Machine</th>
<th>one-way</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algebra</td>
<td>RAT</td>
<td>??</td>
</tr>
</tbody>
</table>

Most of the known results on 2-way transducers concern the **functional** (=deterministic) case…
A simple example: $\text{SQUARE} = \{(w, ww) \mid w \in \Sigma^*\}$
A simple example: \( \textsc{Square} = \{(w, \text{ww}) \mid w \in \Sigma^*\} \)

- copy the input word
A simple example: $\text{SQUARE} = \{(w, ww) \mid w \in \Sigma^*\}$

- copy the input word
- rewind the input tape
A simple example: \( \text{SQUARE} = \{(w, ww) \mid w \in \Sigma^*\} \)

- 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: \( \text{POWERS} = \{(w, w^*) \mid w \in \Sigma^*\} \)
Another example: $\text{POWERS} = \{(w, w^*) \mid w \in \Sigma^*\}$
Another example: $\text{POWERS} = \{(w, w^*) | w \in \Sigma^*\}$

copy the input word $\rightarrow$ rewind the input tape
Another example: $\text{POWERS} = \left\{ (w, w^*) \mid w \in \Sigma^* \right\}$

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

accept and halt with nondeterminism
Another example: \( \text{POWERS} = \{(w, w^*) \mid w \in \Sigma^*\} \)
Another example: \( \text{POWERS} = \{(w, w^*) \mid w \in \Sigma^*\} \)
Another example: \( \text{POWERS} = \{ (w, w^*) \mid w \in \Sigma^* \} \)
A last one: $2 \text{-} \text{PREF} = \{ (a^n, a^p b^p) \mid p \leq n \}$
A last one: $2\text{-}\mathcal{PREF} = \{ (a^n, a^p b^p) \mid p \leq n \}$
A last one: \(2\text{-PREF} = \{(a^n, a^p b^p) \mid p \leq n\} \)
A last one: $2\text{-}\text{Pref} = \{(a^n, a^p b^p) \mid p \leq n\}$
Hadamard operations

- Union

\[ R_1 \cup R_2 \]
Hadamard operations

- Union \[ R_1 \cup R_2 \]
- H-product \( R_1 \oplus R_2 = \{(u, v_1 v_2) \mid (u, v_1) \in R_1 \text{ and } (u, v_2) \in R_2\} \)
Hadamard operations

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

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

example:

- SQUARE = \( \text{Id} \oplus \text{Id} \)
Hadamard operations

- Union
  \[ R_1 \cup R_2 \]

- H-product
  \[ R_1 \oplus R_2 = \{ (u, v_1 v_2) \mid (u, v_1) \in R_1 \text{ and } (u, v_2) \in R_2 \} \]

- H-star
  \[ R^{H*} = \{ (u, v_1 v_2 \cdots v_k) \mid \forall i \ (u, v_i) \in R \} \]
Hadamard operations

- Union
  \[ R_1 \cup R_2 = \{ (u, v_1 v_2) \mid (u, v_1) \in R_1 \text{ and } (u, v_2) \in R_2 \} \]

- H-product
  \[ R_1 \oplus R_2 = \{ (u, v_1 v_2) \mid (u, v_1) \in R_1 \text{ and } (u, v_2) \in R_2 \} \]

- H-star
  \[ R^{H*} = \{ (u, v_1 v_2 \cdots v_k) \mid \forall i \ (u, v_i) \in R \} \]

- repeat
  - simulate \( T \)
    - rewind the input tape
  - or accept nondeterministically

example:
- \( \text{POWERS} = \text{ID}^{H*} \)
Hadamard operations

- **Union**
  \[ R_1 \cup R_2 \]

- **H-product**
  \[ R_1 \oplus R_2 = \{ (u, v_1 v_2) \mid (u, v_1) \in R_1 \text{ and } (u, v_2) \in R_2 \} \]

- **H-star**
  \[ R^{H*} = \{ (u, v_1 v_2 \cdots v_k) \mid \forall i (u, v_i) \in R \} \]

**Definition \( HAD (\Sigma^* \times \Delta^*) \)**

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 $\equiv$ RAT.
What about two-way transducers?

Theorem

1-way transducers $\equiv \text{RAT}$.

ex: $2\text{-PREF} \quad a^n \leftrightarrow \{a^p b^p, \ p \leq n\}$
What about rotating transducers?

Theorem

1-way transducers $\equiv \text{RAT}$. 

Rotating transducers
What about rotating transducers?

Theorem

\[ \text{rotating transducers} \iff \text{HAD}. \]

<table>
<thead>
<tr>
<th>one-way</th>
<th>rotating</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAT</td>
<td>HAD</td>
<td>HAD??</td>
</tr>
</tbody>
</table>

ex: \(2\text{-PREF} \quad a^n \leftrightarrow \{a^p b^p, \ p \leq n\} \)
Mirror operation:
\[ \overline{R} = \{ (\overline{u}, v) \mid (u, v) \in R \} \]

Example
\[ \overline{\text{ID}} = \{ w, \overline{w} \} \]
Mirror operation:
\[ \overline{R} = \left\{ (\overline{u}, v) \mid (u, v) \in R \right\} \]

Example
\[ \overline{ID} = \{ w, \overline{w} \} \]

Definition (\( \text{MHAD}(\Sigma^* \times \Delta^*) \))

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 $\equiv$ MHAD.

<table>
<thead>
<tr>
<th>one-way</th>
<th>rotating</th>
<th>sweeping</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAT</td>
<td>HAD</td>
<td>MHAD</td>
<td>MHAD??</td>
</tr>
</tbody>
</table>

19/26
What about sweeping transducers?

**Theorem**

{sweeping transducers} \( \equiv \) MHAD.

<table>
<thead>
<tr>
<th>one-way</th>
<th>rotating</th>
<th>sweeping</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAT</td>
<td>HAD</td>
<td>MHAD</td>
<td>MHAD??</td>
</tr>
</tbody>
</table>

ex: 2-PREF \[ a^n \mapsto \{a^p b^p, \ p \leq n\} \]
What about sweeping transducers?

**Theorem**

sweeping transducers $\equiv$ MHAD.

<table>
<thead>
<tr>
<th>transducer</th>
<th>one-way</th>
<th>rotating</th>
<th>sweeping</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Diagram" /></td>
<td><img src="image2.png" alt="Diagram" /></td>
<td><img src="image3.png" alt="Diagram" /></td>
<td><img src="image4.png" alt="Diagram" /></td>
<td><img src="image5.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>general</strong></th>
<th>RAT</th>
<th>HAD</th>
<th>MHAD</th>
<th>MHAD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>input unary</strong></td>
<td>$a^n \mapsto {a^p b^p, \ p \leq n}$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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

- $\text{UID} = \{(a^n, a^n) \mid n \in \mathbb{N}\}$ $\in \text{RAT}$
Unary transducers

We focus on a weaker problem:

\[ \Sigma = \{ a \} \quad \text{and} \quad \Delta = \{ a \} \]

Examples

- \text{uId} = \{(a^n, a^n) \mid n \in \mathbb{N}\} \quad \in \text{RAT}

- \text{uSquare} = \text{uId} \oplus \text{uId} = \{(a^n, a^{2n}) \mid n \in \mathbb{N}\} \quad \in \text{RAT}
Unary transducers

We focus on a weaker problem:

\[ \Sigma = \{ a \} \quad \text{and} \quad \Delta = \{ a \} \]

Examples

- \text{uID} = \{(a^n, a^n) \mid n \in \mathbb{N}\} \quad \in \text{RAT}
- \text{uSquare} = \text{uID} \oplus \text{uID} = \{(a^n, a^{2n}) \mid n \in \mathbb{N}\} \quad \in \text{RAT}
- \text{uPowers} = \text{uID}^{H^*} = \{(u^n, u^{kn}) \mid k, n \in \mathbb{N}\} \quad \in \text{HAD \setminus RAT}
Characterization of unary two-way transductions

<table>
<thead>
<tr>
<th>transducer</th>
<th>one-way</th>
<th>rotating</th>
<th>sweeping</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>general</td>
<td><img src="image" alt="RAT" /></td>
<td><img src="image" alt="HAD" /></td>
<td>MHAD</td>
<td>MHAD</td>
</tr>
<tr>
<td>input unary</td>
<td><img src="image" alt="RAT" /></td>
<td><img src="image" alt="HAD" /></td>
<td>MHAD</td>
<td>MHAD</td>
</tr>
<tr>
<td>input and output unary</td>
<td><img src="image" alt="RAT" /></td>
<td><img src="image" alt="HAD" /></td>
<td>??</td>
<td>??</td>
</tr>
</tbody>
</table>
Characterization of unary two-way transductions

**Theorem**

2-way unary transducers $\equiv$ HAD

<table>
<thead>
<tr>
<th>transducer</th>
<th>one-way</th>
<th>rotating</th>
<th>sweeping</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
</tr>
</tbody>
</table>

- **general**
- **input unary**
- **input and output unary**
Characterization of unary two-way transductions

Theorem

2-way unary transducers $\equiv$ $\text{HAD}$

Corollary

2-way unary transducers $\rightarrow$ rotating transducers.

---

<table>
<thead>
<tr>
<th>transducer</th>
<th>one-way</th>
<th>rotating</th>
<th>sweeping</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>general</td>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Diagram" /></td>
<td><img src="image3" alt="Diagram" /></td>
<td><img src="image4" alt="Diagram" /></td>
</tr>
<tr>
<td>input unary</td>
<td><img src="image5" alt="Diagram" /></td>
<td><img src="image6" alt="Diagram" /></td>
<td><img src="image7" alt="Diagram" /></td>
<td><img src="image8" alt="Diagram" /></td>
</tr>
<tr>
<td>input and output unary</td>
<td><img src="image9" alt="Diagram" /></td>
<td><img src="image10" alt="Diagram" /></td>
<td><img src="image11" alt="Diagram" /></td>
<td><img src="image12" alt="Diagram" /></td>
</tr>
</tbody>
</table>
Characterization of unary two-way transductions

Theorem

2-way unary transducers $\equiv \text{HAD}$

Corollary

2-way unary transducers $\rightarrow$ rotating transducers.

Example

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

<table>
<thead>
<tr>
<th>transducer</th>
<th>one-way</th>
<th>rotating</th>
<th>sweeping</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image-url" alt="Diagram" /></td>
<td><img src="image-url" alt="Diagram" /></td>
<td><img src="image-url" alt="Diagram" /></td>
<td><img src="image-url" alt="Diagram" /></td>
<td><img src="image-url" alt="Diagram" /></td>
</tr>
</tbody>
</table>

- **general**
- **input unary**
- **input and output unary**
Key points of the proof

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

22 / 26
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

<table>
<thead>
<tr>
<th>transducer</th>
<th>one-way</th>
<th>rotating</th>
<th>sweeping</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image.png" alt="Diagram" /></td>
<td><img src="image.png" alt="Diagram" /></td>
<td><img src="image.png" alt="Diagram" /></td>
<td><img src="image.png" alt="Diagram" /></td>
<td><img src="image.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>general</th>
<th>RAT</th>
<th>HAD</th>
<th>MHAD</th>
<th>MHAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>input unary</td>
<td></td>
<td></td>
<td>MHAD</td>
<td>MHAD</td>
</tr>
<tr>
<td>output unary</td>
<td></td>
<td></td>
<td></td>
<td>??</td>
</tr>
<tr>
<td>input and output unary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

arbitrary $\Sigma$ and $\Delta = \{a\}$
The output-unary case

 Arbitrary $\Sigma$ and $\Delta = \{a\}$

### Proposition

$\text{HAD} = \text{MHAD}$

<table>
<thead>
<tr>
<th>transducer</th>
<th>one-way</th>
<th>rotating</th>
<th>sweeping</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>general</th>
<th>RAT</th>
<th>MHAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>input unary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>output unary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>input and output unary</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The output-unary case

Proposition

\[ \text{HAD} = \text{MHAD} = \bigcup_{\text{finite}} \mathbb{R} \upharpoonright \Sigma^* \]

arbitrary \( \Sigma \) and \( \Delta = \{ a \} \)

<table>
<thead>
<tr>
<th>transducer</th>
<th>one-way</th>
<th>rotating</th>
<th>sweeping</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \cdot \downarrow \cdot \cdot )</td>
<td>( \cdot \uparrow \cdot \cdot )</td>
<td>( \cdot \downarrow \cdot \cdot )</td>
<td>( \cdot \uparrow \cdot \cdot )</td>
<td></td>
</tr>
</tbody>
</table>

General

Input unary

Output unary

Input and output unary

\( \text{RAT} \)

\( \text{HAD} \)

\( \text{MHAD} \)

??
The output-unary case

arbitrary $\Sigma$ and $\Delta = \{a\}$

Proposition

$$HAD = MHAD = \bigcup_{\text{finite}} R \oplus S^{H*}$$

Theorem

2-way output-unary $\not\equiv HAD$

<table>
<thead>
<tr>
<th>transducer</th>
<th>one-way</th>
<th>rotating</th>
<th>sweeping</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="transducer diagram" /></td>
<td><img src="image" alt="one-way diagram" /></td>
<td><img src="image" alt="rotating diagram" /></td>
<td><img src="image" alt="sweeping diagram" /></td>
<td><img src="image" alt="two-way diagram" /></td>
</tr>
</tbody>
</table>

general

input unary

output unary

input and output unary

$RAT$

$HAD$

$MHAD$
An non-Hadamard output-unary transduction

\[ \Sigma = \{ a, \# \} \quad \text{and} \quad \Delta = \{ a \} \]

\[ R = \{(u, a^{kn}) \mid k, n \in \mathbb{N}, \#a^k\# \text{ is a factor of } u \} \]
An non-Hadamard output-unary transduction

\[ \Sigma = \{ a, \# \} \quad \text{and} \quad \Delta = \{ a \} \]

\[ R = \left\{ (u, a^{kn}) \mid k, n \in \mathbb{N}, \#a^k\# \text{ is a factor of } u \right\} \]
An non-Hadamard output-unary transduction

\[ \Sigma = \{ a, \# \} \quad \text{and} \quad \Delta = \{ a \} \]

\[ R = \left\{ (u, a^{kn}) \mid k, n \in \mathbb{N}, \#a^k\# \text{ is a factor of } u \right\} \]
An non-Hadamard output-unary transduction

\[ \Sigma = \{a, \#\} \quad \text{and} \quad \Delta = \{a\} \]

\[ R = \{(u, a^{kn}) \mid k, n \in \mathbb{N}, \#a^k\# \text{ is a factor of } u\} \]
An non-Hadamard output-unary transduction

\[ \Sigma = \{ a, \# \} \quad \text{and} \quad \Delta = \{ a \} \]

\[ R = \left\{ (u, a^{kn}) \mid k, n \in \mathbb{N}, \#a^k\# \text{ is a factor of } u \right\} \]
An non-Hadamard output-unary transduction

\[ \Sigma = \{ a, \# \} \quad \text{and} \quad \Delta = \{ a \} \]

\[ R = \{ (u, a^{kn}) \mid k, n \in \mathbb{N}, \ a^k\# \text{ is a factor of } u \} \]
An non-Hadamard output-unary transduction

\[ \Sigma = \{ a, \# \} \quad \text{and} \quad \Delta = \{ a \} \]

\[ R = \{ (u, a^{kn}) \mid k, n \in \mathbb{N}, \#a^k\# \text{ is a factor of } u \} \]
An non-Hadamard output-unary transduction

\[ \Sigma = \{ a, \# \} \quad \text{and} \quad \Delta = \{ a \} \]

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

\[ \Delta = \{ a \} \]

\[ \Sigma = \{ a, \# \} \]

\[ R = \{ (u, a^{kn}) \mid k, n \in \mathbb{N}, \#a^k\# \text{ is a factor of } u \} \]
Two-way transducers **VERSUS** Algebra

<table>
<thead>
<tr>
<th>transducer</th>
<th>one-way</th>
<th>rotating</th>
<th>sweeping</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
</tr>
</tbody>
</table>

- **general**
  - **input** unary
  - **output** unary
  - **input and output** unary

**Legend:***
- **RAT**
- **HAD**
- **MHAD**

- **Notation:** $a, -1 \mid b$
Two-way transducers **VERSUS** Algebra

### Functional Case

<table>
<thead>
<tr>
<th>Transducer</th>
<th>One-way</th>
<th>Rotating</th>
<th>Sweeping</th>
<th>Two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
</tr>
</tbody>
</table>

- **General**
- **Input Unary**
- **Output Unary**
- **Input and Output Unary**
Conclusion

Descriptional complexity

-polynomial if $L = NL$

Two-way transducers

<table>
<thead>
<tr>
<th>transducer</th>
<th>one-way</th>
<th>rotating</th>
<th>Sweeping</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>general</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>input unary</td>
<td><em>RAT</em></td>
<td></td>
<td>MHAD</td>
<td>MHAD</td>
</tr>
<tr>
<td>output unary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>input and output unary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conclusion

Descriptive complexity

- Polynomial if $L = \text{NL}$
- Sub-exponential

Two-way transducers

<table>
<thead>
<tr>
<th>Transducer</th>
<th>One-way</th>
<th>Rotating</th>
<th>Sweeping</th>
<th>Two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td><img src="#" alt="Rat" /></td>
<td><img src="#" alt="Had" /></td>
<td><img src="#" alt="MHad" /></td>
<td><img src="#" alt="MHad" /></td>
</tr>
<tr>
<td>Input unary</td>
<td><img src="#" alt="Rat" /></td>
<td><img src="#" alt="Had" /></td>
<td><img src="#" alt="MHad" /></td>
<td><img src="#" alt="MHad" /></td>
</tr>
<tr>
<td>Output unary</td>
<td><img src="#" alt="Rat" /></td>
<td><img src="#" alt="Had" /></td>
<td><img src="#" alt="MHad" /></td>
<td><img src="#" alt="MHad" /></td>
</tr>
<tr>
<td>Input and output unary</td>
<td><img src="#" alt="Rat" /></td>
<td><img src="#" alt="Had" /></td>
<td><img src="#" alt="MHad" /></td>
<td><img src="#" alt="MHad" /></td>
</tr>
</tbody>
</table>
Conclusion

Descriptional complexity

- polynomial if $L = \text{NL}$
- sub-exponential

Two-way transducers

<table>
<thead>
<tr>
<th>transducer</th>
<th>one-way</th>
<th>rotating</th>
<th>sweeping</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{RAT}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{HAD}$</td>
<td>MHAD</td>
<td></td>
<td></td>
<td>MHAD</td>
</tr>
<tr>
<td>$\text{input unary}$</td>
<td>RAT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{output unary}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{input and output unary}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Alternating 2ONFA
- Other restrictions on nondeterminism of 2NFA
Conclusion

Descriptive complexity

- Exponential: $2^{\text{DFA}} \rightarrow 2^{\text{NFA}} \rightarrow 2^{\text{ONFA}}$
- Sub-exponential: $L = \text{NL}$
- Polynomial: $L = \text{P}$

Two-way transducers

- General
- Input unary
- Output unary
- Input and output unary

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

- Uniformization
Conclusion

Descriptional complexity

- Polynomial if $L = NL$
- Sub-exponential
- 2DFA → 2ONFA → 2NFA

Two-way transducers

- Alternating 2ONFA
- Other restrictions on nondeterminism of 2NFA
- Uniformization
- Composition $R_1 \circ R_2$
- Transitive closure

<table>
<thead>
<tr>
<th>transducer</th>
<th>one-way</th>
<th>rotating</th>
<th>sweeping</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a \cdot 1 \mid b$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>general</td>
<td>RAT</td>
<td>HAD</td>
<td>MHAD</td>
<td></td>
</tr>
<tr>
<td>input unary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>output unary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>input and output unary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Thanks for your attention
Conclusion

Descriptive complexity

- Polynomial if $L = \text{NL}$
- Sub-exponential

Two-way transducers

- Alternating 2onfa
- Other restrictions on nondeterminism of 2NFA
- Uniformization
- Composition $R_1 \circ R_2$
- Transitive closure
- Extend to series

Table:

<table>
<thead>
<tr>
<th>Transducer</th>
<th>One-way</th>
<th>Rotating</th>
<th>Sweeping</th>
<th>Two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>RAT</td>
<td>HAD</td>
<td>MHAD</td>
<td>MHAD</td>
</tr>
<tr>
<td>Input unary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output unary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input and output unary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conclusion

Descriptional complexity

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

Two-way transducers

<table>
<thead>
<tr>
<th>transducer</th>
<th>one-way</th>
<th>rotating</th>
<th>sweeping</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>general</td>
<td>✅</td>
<td>✅</td>
<td>✅</td>
<td>✅</td>
</tr>
<tr>
<td>input unary</td>
<td>✅</td>
<td>✅</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>output unary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>input and output unary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Uniformization
- Composition $R_1 \circ R_2$
- Transitive closure
- Extend to series
- Describe the mirror
Conclusion

Descriptional complexity

- **2DFA**
- **2ONFA**
- **2NFA**

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

Two-way transducers

<table>
<thead>
<tr>
<th>transducer</th>
<th>one-way</th>
<th>rotating</th>
<th>sweeping</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MHAD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Uniformization
- Composition $R_1 \circ R_2$
- Transitive closure
- Extend to series
- Describe the mirror
- Cost of simulations

Thanks for your attention
Conclusion

Descriptional complexity

- Polynomial if $L = NL$
- Sub-exponential

Two-way transducers

<table>
<thead>
<tr>
<th>transducer</th>
<th>one-way</th>
<th>rotating</th>
<th>sweeping</th>
<th>two-way</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>input unary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>output unary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>input and output unary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 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