Two-Way Automata Characterizations of L/poly versus NL

Christos A. Kapoutsis[1] Giovanni Pighizzini[2]

[1]LIAFA, Université Paris VII, France

[2]DI, Università degli Studi di Milano, Italia

CRS 2012 – Nizhny Novgorod, Russia
July 3–7, 2012

Nondeterminism with Bounded Resources

- Time complexity

  $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$ *polynomial time*

Nondeterminism with Bounded Resources

- Time complexity

  $\mathsf{P} \stackrel{?}{=} \mathsf{NP}$ *polynomial time*

- Space complexity

  $\mathsf{PSPACE} = \mathsf{NPSPACE}$ *polynomial space*

  $\mathsf{L} \stackrel{?}{=} \mathsf{NL}$ *logarithmic space*

Nondeterminism with Bounded Resources

- Time complexity

  $\mathsf{P} \stackrel{?}{=} \mathsf{NP}$ — *polynomial time*

- Space complexity

  $\mathsf{PSPACE} = \mathsf{NPSPACE}$ — *polynomial space*

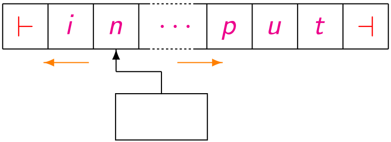
  $\mathsf{L} \stackrel{?}{=} \mathsf{NL}$ — *logarithmic space*

- State complexity

  $1\mathsf{D} \subsetneq 1\mathsf{N}$ — *one-way automata*

  $2\mathsf{D} \stackrel{?}{=} 2\mathsf{N}$ — *two-way automata*

Two-Way Automata



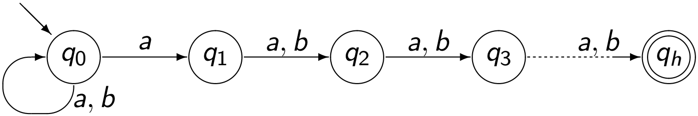
- The input head can be moved in both directions
- They recognize only regular language
- They can be *smaller* than one-way automata

Technical detail:

- Input surrounded by the *endmarkers* $\vdash$ and $\dashv$

An Example

$L_h = (a+b)^* a (a+b)^{h-1}$



- 1NFA: $h+1$ states
- 1DFA: 2^h states
- 2DFA: $h+2$ states

Classes

- Family of problems/languages $\mathcal{L} = (L_h)_{h \geq 1}$
- 2D class of families of problems solvable by poly-size 2DFAs:

  $\mathcal{L} \in$ 2D iff $\exists$polynomial p s.t.
  each L_h is solved by a 2DFA of size $p(h)$
- 1D, 1N, 2N ...

Classes

- Family of problems/languages $\mathcal{L} = (L_h)_{h \geq 1}$
- 2D class of families of problems solvable by poly-size 2DFAs:

  $\mathcal{L} \in$ 2D iff $\exists$polynomial p s.t.
  each L_h is solved by a 2DFA of size $p(h)$
- 1D, 1N, 2N ...

Example

$L_h = (a+b)^* a (a+b)^{h-1}$ $\mathcal{L} = (L_h)_{h \geq 1}$:

- 1NFA: $h+1$ states $\Rightarrow \mathcal{L} \in 1N$
- 1DFA: 2^h states $\Rightarrow \mathcal{L} \notin 1D$
- 2DFA: $h+2$ states $\Rightarrow \mathcal{L} \in 2D \subseteq 2N$

The Question of Sakoda and Sipser

Problem ([Sakoda&Sipser '78])

Do there exist polynomial simulations of

- 1NFAs *by* 2DFAs
- 2NFAs *by* 2DFAs *?*

The Question of Sakoda and Sipser

Problem ([Sakoda&Sipser '78])

Do there exist polynomial simulations of

- 1NFAs *by* 2DFAs
- 2NFAs *by* 2DFAs *?*

Conjecture

Both simulations are not polynomial!
i.e., $1N \neq 2D$ *and* $2N \neq 2D$

Two-Way Automata versus Logarithmic Space

Theorem ([Berman&Lingas '77])

If L = NL *then*
for every s-state σ*-symbol* 2NFA
there is a poly$(s\sigma)$*-state* 2DFA
which agrees with it on all inputs of length $\leq s$

Two-Way Automata versus Logarithmic Space

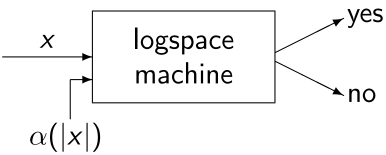
Theorem ([Berman&Lingas '77])

If L = NL *then*
for every s-state σ-symbol 2NFA
there is a poly($s\sigma$)-state 2DFA
which agrees with it on all inputs of length $\leq s$

Theorem ([Geffert&P '11])

If L = NL *then*
for every s-state unary 2NFA
there is an equivalent
poly(s)-state 2DFA

- L/poly
  class of languages accepted by deterministic logspace machines with a *polynomial advice*



Problem

L/poly $\supseteq$ NL ?

Two-Way Automata versus Logarithmic Space

Theorem ([Berman&Lingas '77])

If L $=$ NL *then*
for every s-state σ-symbol 2NFA
there is a poly$(s\sigma)$-state 2DFA
which agrees with it on all inputs of length $\leq s$

Theorem ([Geffert&P '11])

If L $=$ NL *then*
for every s-state unary 2NFA
there is an equivalent
poly(s)-state 2DFA

Two-Way Automata versus Logarithmic Space

Theorem ([Berman&Lingas '77])

If L = NL *then*
for every s-state σ*-symbol* 2NFA
there is a poly$(s\sigma)$*-state* 2DFA
which agrees with it on all inputs of length $\leq s$

↙ ↘

Theorem ([Geffert&P '11])

If L = NL *then*
for every s-state unary 2NFA
there is an equivalent
poly(s)*-state* 2DFA

Theorem ([Kapoutsis '11])

L/poly $\supseteq$ NL *iff*
for every s-state σ*-symbol* 2NFA
there is a poly(s)*-state* 2DFA
which agrees with it on all inputs
of length $\leq s$

Two-Way Automata versus Logarithmic Space

2N/unary := only *unary* inputs

Theorem ([Geffert&P '11])

L = NL ⇒ 2D ⊇ 2N/unary

2N/poly := only *short* inputs

Theorem ([Kapoutsis '11])

L/poly ⊇ NL ⇔ 2D ⊇ 2N/poly

Two-Way Automata versus Logarithmic Space

2N/unary := only *unary* inputs

Theorem ([Geffert&P '11])

$L = NL \Rightarrow 2D \supseteq 2N/unary$

$\downarrow$

- What about the weaker hypothesis $L/poly \supseteq NL$?
- What about the converse of this statement?

2N/poly := only *short* inputs

Theorem ([Kapoutsis '11])

$L/poly \supseteq NL \Leftrightarrow 2D \supseteq 2N/poly$

Two-Way Automata versus Logarithmic Space

2N/unary := only *unary* inputs

Theorem ([Geffert&P '11])

$\mathsf{L} = \mathsf{NL} \Rightarrow \mathsf{2D} \supseteq \mathsf{2N/unary}$

↓

- What about the weaker hypothesis $\mathsf{L/poly} \supseteq \mathsf{NL}$?
- What about the converse of this statement?

In this work:

$\mathsf{L/poly} \supseteq \mathsf{NL} \Leftrightarrow \mathsf{2D} \supseteq \mathsf{2N/unary}$

2N/poly := only *short* inputs

Theorem ([Kapoutsis '11])

$\mathsf{L/poly} \supseteq \mathsf{NL} \Leftrightarrow \mathsf{2D} \supseteq \mathsf{2N/poly}$

Two-Way Automata versus Logarithmic Space

2N/unary := only *unary* inputs

Theorem ([Geffert&P '11])

L = NL $\Rightarrow$ 2D $\supseteq$ 2N/unary

2N/poly := only *short* inputs

Theorem ([Kapoutsis '11])

L/poly $\supseteq$ NL $\Leftrightarrow$ 2D $\supseteq$ 2N/poly

$\downarrow$

In this work:

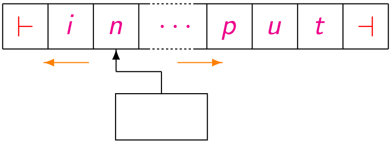
L/poly $\supseteq$ NL $\Leftrightarrow$ 2D $\supseteq$ 2N/unary

$\longrightarrow$

Furthermore:

- Investigation of the common behavior unary/short
- Characterizations of L/poly vs NL

1st Tool: Outer Nondeterministic Automata (2OFA)

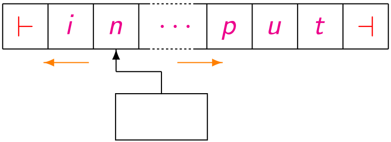


Nondeterministic choices are possible
only when the head is scanning the endmarkers

Lemma

For every s-state 2NFA *and integer* l
there is a $poly(sl)$*-state* 2OFA *which agrees with it*
on all inputs of length $\leq l$

1st Tool: Outer Nondeterministic Automata (2OFA)



Nondeterministic choices are possible
only when the head is scanning the endmarkers

Lemma ([Geffert et al. '03])

For every s-state unary 2NFA *there is an equivalent poly(s)-state* 2OFA

Lemma

For every s-state 2NFA *and integer l there is a poly(sl)-state* 2OFA *which agrees with it on all inputs of length* $\leq l$

2nd Tool: The Graph Accessibility Problem

GAP:

- Given $G = (V, E)$ an oriented graph, $s, t \in V$
- Decide whether or not G contains a path from s to t

2nd Tool: The Graph Accessibility Problem

GAP:

- Given $G = (V, E)$ an oriented graph, $s, t \in V$
- Decide whether or not G contains a path from s to t

Theorem ([Jones'75])

GAP *is complete for* NL *(under logspace reductions)*

$\Rightarrow$ GAP $\in$ L iff L $=$ NL

2nd Tool: The Graph Accessibility Problem

GAP:

- Given $G = (V, E)$ an oriented graph, $s, t \in V$
- Decide whether or not G contains a path from s to t

Theorem ([Jones '75])

GAP is complete for NL (under logspace reductions)

$\Rightarrow$ GAP $\in$ L iff L = NL

GAP$_h$:

- GAP restricted to graphs with vertex set $V_h = \{0, \ldots, h-1\}$

We show that
under suitable encodings
the family (GAP$_h$) is complete
for 2N/unary and 2N/poly

2nd Tool: The Graph Accessibility Problem

GAP:

- Given $G = (V, E)$ an oriented graph, $s, t \in V$
- Decide whether or not G contains a path from s to t

Theorem ([Jones '75])

GAP is complete for NL
(under logspace reductions)

$\Rightarrow$ GAP $\in$ L iff L $=$ NL

GAP$_h$:

- GAP restricted to graphs with vertex set $V_h = \{0, \ldots, h-1\}$

We show that
under suitable encodings
the family (GAP$_h$) is complete
for 2N/unary and 2N/poly

$\Rightarrow$ (GAP$_h$) $\in$ 2D iff
2D $\supseteq$ 2N/unary iff
2D $\supseteq$ 2N/poly iff
L/poly $\supseteq$ NL

Binary Encoding: The Family BGAP

- $G = (V_h, E)$, with $V_h = \{0, \ldots, h-1\}$
- *Binary encoding* of G:

  $\langle G \rangle_2 \in \{0,1\}^{h^2}$ standard encoding of the adjacency matrix
- $\text{BGAP}_h := \{\langle G \rangle_2 \mid G$ has a path from 0 to $h-1\}$
- 2NFA recognizing BGAP_h:
  - *input:* $x \in \{0,1\}^{h^2}$ *output:* $x \in \text{BGAP}_h$?
  - Nondeterministic choices only on the left endmarker
  - $O(h^3)$ states

Lemma

$\text{BGAP} \in 2O$

Binary Encoding: The Family BGAP

- $G = (V_h, E)$, with $V_h = \{0, \ldots, h-1\}$
- *Binary encoding* of G:

  $\langle G \rangle_2 \in \{0,1\}^{h^2}$ standard encoding of the adjacency matrix

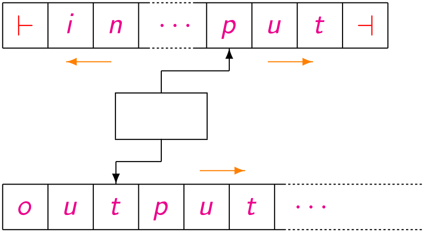
- $\mathrm{BGAP}_h := \{\langle G \rangle_2 \mid G \text{ has a path from } 0 \text{ to } h-1\}$
- 2NFA recognizing BGAP_h:
  - *input:* $x \in \{0,1\}^{h^2}$ *output:* $x \in \mathrm{BGAP}_h$?
  - Nondeterministic choices only on the left endmarker
  - $O(h^3)$ states

Lemma

$\mathrm{BGAP} \in 2\mathrm{O}$

Reductions

Two-Way Deterministic Transducer (2DFT)



- $\mathcal{L} = (L_h)_{h \geq 1}$, $\mathcal{L}' = (L'_h)_{h \geq 1}$
- "Small" reduction:
  $\mathcal{L} \leq_{sm} \mathcal{L}'$ iff each L_h reduces to L'_h
  via "small" 2DFTs with "short" outputs

BGAP and Characterizations

Theorem

BGAP *is*

2N/poly-*complete*

2O-*complete*

under $\leq_{sm}$

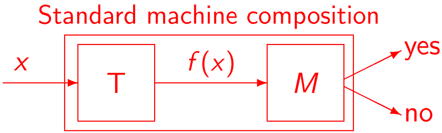
BGAP and Characterizations

Theorem

BGAP *is*
 2N/poly-*complete*
 2O-*complete*
under $\leq_{sm}$

Lemma

2D *is closed under* $\leq_{sm}$



BGAP and Characterizations

Theorem

BGAP *is*

- 2N/poly-*complete*
- 2O-*complete*

under $\leq_{sm}$

Lemma

2D *is closed under* $\leq_{sm}$

Hence the following statements are equivalent:

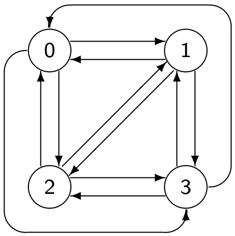
$$2D \ni \text{BGAP}$$

$$2D \supseteq 2N/\text{poly} \qquad 2D \supseteq 2O$$

Unary Encoding: The Family UGAP

- $K_h :=$ complete directed graph with vertex set $V_h = \{0, \ldots, h-1\}$
- With each edge (i, j) we associate a different prime number $p_{(i,j)}$
- A subgraph $G = (V_h, E)$ of K_h is encoded by the string a^{m_G}, where

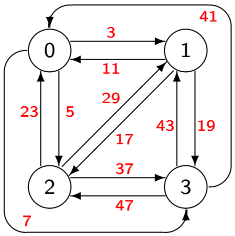
$$m_G = \prod_{(i,j) \in E} p_{(i,j)}$$



Unary Encoding: The Family UGAP

- $K_h :=$ complete directed graph with vertex set $V_h = \{0, \ldots, h-1\}$
- With each edge (i, j) we associate a different prime number $p_{(i,j)}$
- A subgraph $G = (V_h, E)$ of K_h is encoded by the string a^{m_G}, where

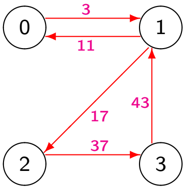
$$m_G = \prod_{(i,j) \in E} p_{(i,j)}$$



Unary Encoding: The Family UGAP

- $K_h :=$ complete directed graph with vertex set $V_h = \{0, \ldots, h-1\}$
- With each edge (i, j) we associate a different prime number $p_{(i,j)}$
- A subgraph $G = (V_h, E)$ of K_h is encoded by the string a^{m_G}, where

$$m_G = \prod_{(i,j) \in E} p_{(i,j)}$$

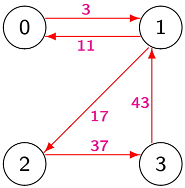


$$\begin{aligned} m_G &= 3 \cdot 11 \cdot 17 \cdot 37 \cdot 43 \\ &= 892551 \end{aligned}$$

Unary Encoding: The Family UGAP

- K_h := complete directed graph with vertex set $V_h = \{0, \dots, h-1\}$
- With each edge (i,j) we associate a different prime number $p_{(i,j)}$
- A subgraph $G = (V_h, E)$ of K_h is encoded by the string a^{m_G}, where

$$m_G = \prod_{(i,j) \in E} p_{(i,j)}$$



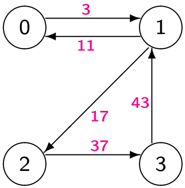
$$m_G = 3 \cdot 11 \cdot 17 \cdot 37 \cdot 43 = 892551$$

- Graph $K_h(m)$: $\exists$ edge (i,j) iff $p_{(i,j)}$ divides m
- $\text{UGAP}_h := \{a^m \mid K_h(m) \text{ has a path from } 0 \text{ to } h-1\}$

Unary Encoding: The Family UGAP

- K_h := complete directed graph with vertex set $V_h = \{0, \ldots, h-1\}$
- With each edge (i, j) we associate a different prime number $p_{(i,j)}$
- A subgraph $G = (V_h, E)$ of K_h is encoded by the string a^{m_G}, where

$$m_G = \prod_{(i,j) \in E} p_{(i,j)}$$



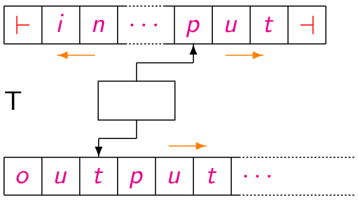
$$m_G = 3 \cdot 11 \cdot 17 \cdot 37 \cdot 43 = 892551$$

- Graph $K_h(m)$: $\exists$ edge (i, j) iff $p_{(i,j)}$ divides m
- $\text{UGAP}_h := \{a^m \mid K_h(m) \text{ has a path from } 0 \text{ to } h-1\}$

Lemma

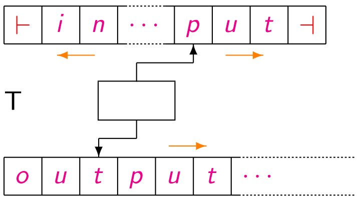
UGAP $\in$ 2O

Prime Reductions



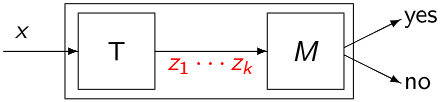
- Producing a unary output a^m could require too many states!
- Output: a list $z_1 \cdots z_k$ of prime powers factorizing m
- "Small" prime reduction $\preceq_{sm}$

Prime Reductions



- Producing a unary output a^m could require too many states!
- Output: a list $z_1 \cdots z_k$ of prime powers factorizing m
- "Small" prime reduction $\preceq_{sm}$

Machine composition



- Unary 2DFAs can be modified to read prime encodings
- This allows to prove that 2D is closed under $\preceq_{sm}$

UGAP and Characterizations

Lemma

2D *is closed under* $\preceq_{sm}$

Theorem

UGAP *is*

- 2N/unary-*complete*
- 2O-*complete*

under $\preceq_{sm}$

UGAP and Characterizations

Lemma

2D is closed under $\preceq_{sm}$

Theorem

UGAP *is*

- 2N/unary-*complete*
- 2O-*complete*

under $\preceq_{sm}$

Hence the following statements are equivalent:

$2D \ni UGAP$

$2D \supseteq 2O$

$2D \supseteq 2N/unary$

UGAP and Characterizations

Lemma

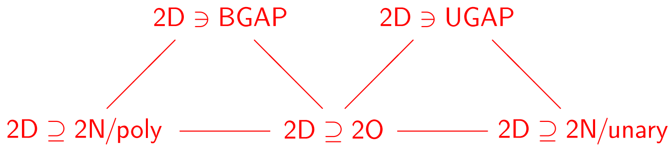
2D *is closed under* $\preceq_{sm}$

Theorem

UGAP *is*
- 2N/unary-*complete*
- 2O-*complete*

under $\preceq_{sm}$

Hence the following statements are equivalent:



UGAP and Characterizations

Lemma

2D *is closed under* $\preceq_{sm}$

Theorem

UGAP *is*

- 2N/unary*-complete*
- 2O*-complete*

under $\preceq_{sm}$

Hence the following statements are equivalent:

- 2D $\ni$ BGAP
- 2D $\ni$ UGAP
- 2D $\supseteq$ 2N/poly
- 2D $\supseteq$ 2O
- 2D $\supseteq$ 2N/unary
- L/poly $\supseteq$ NL (equivalent to 2D $\supseteq$ 2N/poly by (a))

(a) [Kapoutsis '11]

Directions for Further Investigations

- Characterizations in terms of two-way automata of *uniform* L vs NL
- Comparison of two-way automata on unary vs short inputs
- Use of the reductions introduced in the paper for other purposes

Thank you for your attention!