

# Two-Way Automata Characterizations of L/poly versus NL

Christos A. Kapoutsis<sup>1</sup>    Giovanni Pighizzini<sup>2</sup>

<sup>1</sup>LIAFA, Université Paris VII, France

<sup>2</sup>DI, Università degli Studi di Milano, Italia

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

# Nondeterminism with Bounded Resources

- ▶ Time complexity

$$P \stackrel{?}{=} NP$$

*polynomial time*

- ▶ Space complexity

$$PSPACE = NPSPACE$$

*polynomial space*

$$L \stackrel{?}{=} NL$$

*logarithmic space*

- ▶ State complexity

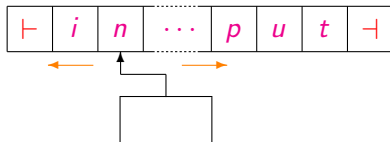
$$1D \subsetneq 1N$$

*one-way automata*

$$2D \stackrel{?}{=} 2N$$

*two-way automata*

# Two-Way Automata



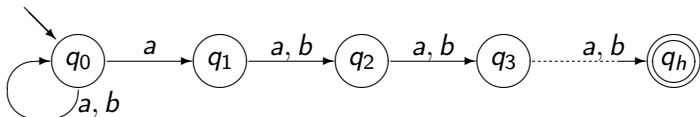
- ▶ The input head can be moved in both directions
- ▶ They recognize only regular language
- ▶ They can be *smaller* than one-way automata

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 ...

## Example

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

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

$$\mathcal{L} = (L_h)_{h \geq 1}:$$

- $\Rightarrow \mathcal{L} \in 1N$
- $\Rightarrow \mathcal{L} \notin 1D$
- $\Rightarrow \mathcal{L} \in 2D \subseteq 2N$

# The Question of Sakoda and Sipser

## Problem ([Sakoda&Sipser '78])

*Do there exist polynomial simulations of*

- ▶ 1NFAs by 2DFAs
- ▶ 2NFAs by 2DFAs ?

## 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  $\sigma$ -symbol 2NFA*

*there is a  $\text{poly}(s\sigma)$ -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*

*$\text{poly}(s)$ -state 2DFA*

## Theorem ([Kapoutsis '11])

*$L/\text{poly} \supseteq NL$  iff*

*for every  $s$ -state  $\sigma$ -symbol 2NFA*

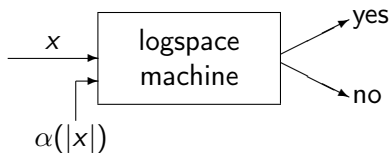
*there is a  $\text{poly}(s)$ -state 2DFA*

*which agrees with it on all inputs of length  $\leq s$*

# Two-Way Automata versus Logarithmic Space

L/poly: Nonuniform Deterministic Logspace

- ▶ 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

$2N/\text{unary}$  := only *unary* inputs

Theorem ([Geffert&P '11])

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



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

$2N/\text{poly}$  := only *short* inputs

Theorem ([Kapoutsis '11])

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

In this work:

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

# Two-Way Automata versus Logarithmic Space

$2N/\text{unary}$  := only *unary* inputs

Theorem ([Geffert&P '11])

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

$2N/\text{poly}$  := only *short* inputs

Theorem ([Kapoutsis '11])

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

In this work:

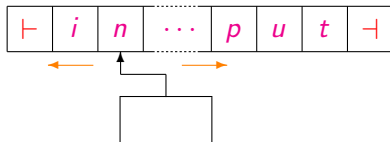
$L/\text{poly} \supseteq NL \Leftrightarrow 2D \supseteq 2N/\text{unary}$



↓  
Furthermore:

- ▶ Investigation of the common behavior unary/short
- ▶ Characterizations of  $L/\text{poly}$  vs  $NL$

# 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  $\text{poly}(s)$ -state 2OFA*

Lemma

*For every  $s$ -state 2NFA and integer  $l$  there is a  $\text{poly}(sl)$ -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  $t$

Theorem ([Jones '75])

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

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

GAP<sub>h</sub>:

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

We show that  
under suitable encodings  
the family (GAP<sub>h</sub>) is complete  
for 2N/unary and 2N/poly

$\Rightarrow$  (GAP<sub>h</sub>)  $\in$  2D iff  
2D  $\supseteq$  2N/unary iff  
2D  $\supseteq$  2N/poly iff  
L/poly  $\supseteq$  NL

# Binary Encoding: The Family BGAP

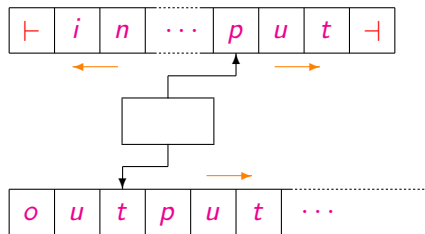
- ▶  $G = (V_h, E)$ , with  $V_h = \{0, \dots, h - 1\}$
- ▶ *Binary encoding* of  $G$ :  
 $\langle G \rangle_2 \in \{0, 1\}^{h^2}$  standard encoding of the adjacency matrix
- ▶  $\text{BGAP}_h := \{\langle G \rangle_2 \mid G \text{ has a path from } 0 \text{ to } h - 1\}$
- ▶ 2NFA recognizing  $\text{BGAP}_h$ :
  - *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

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

# 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

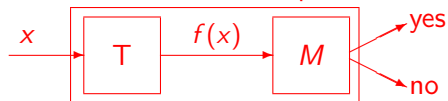
## Theorem

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

## Lemma

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

Standard machine composition



# BGAP and Characterizations

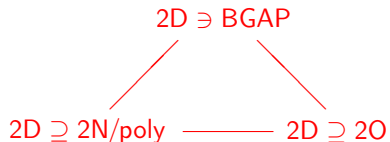
## Theorem

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

## Lemma

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

Hence the following statements are equivalent:



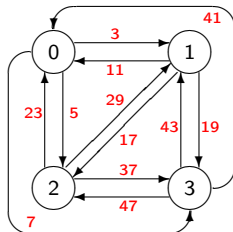


# Unary Encoding: The Family UGAP

- ▶  $K_h :=$  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

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

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

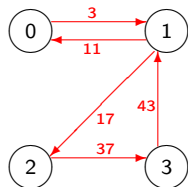


# Unary Encoding: The Family UGAP

- ▶  $K_h :=$  complete directed graph with vertex set  $V_h = \{0, \dots, h-1\}$
- ▶ 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

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

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

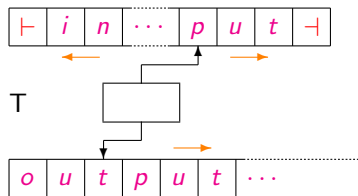


$$\begin{aligned} m_G &= 3 \cdot 11 \cdot 17 \cdot 37 \cdot 43 \\ &= 892551 \end{aligned}$$

Lemma

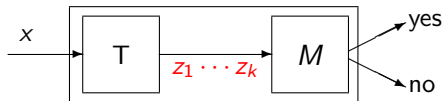
UGAP  $\in$  20

# 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

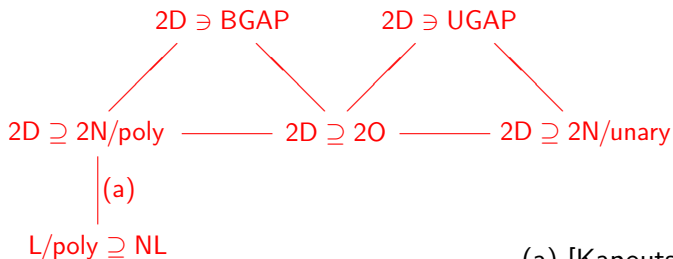
## Lemma

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

## Theorem

$UGAP$  is  
 $2N/unary$ -complete  
 $2O$ -complete  
under  $\preceq_{sm}$

Hence the following statements are equivalent:



(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!