

Two-Way Automata Making Choices Only at the Endmarkers

Viliam Geffert

Bruno Guillon

Giovanni Pighizzini

LATA 2012 – A Coruña, Spain – March 7, 2012

Two-Way Automata Making Choices Only at the Endmarkers

Viliam Geffert¹

Bruno Guillon

Giovanni Pighizzini



¹ Department of Computer Science
P. J. Šafárik University, Košice
Slovakia

LATA 2012 – A Coruña, Spain – March 7, 2012

Two-Way Automata Making Choices Only at the Endmarkers

Viliam Geffert¹ Bruno Guillon² Giovanni Pighizzini



¹ Department of Computer Science
P. J. Šafárik University, Košice
Slovakia

² Université Nice-Sophia Antipolis and
École Normale Supérieure de Lyon
France



LATA 2012 – A Coruña, Spain – March 7, 2012

Two-Way Automata Making Choices Only at the Endmarkers

Viliam Geffert¹ Bruno Guillon² Giovanni Pighizzini³



¹ Department of Computer Science
P. J. Šafárik University, Košice
Slovakia

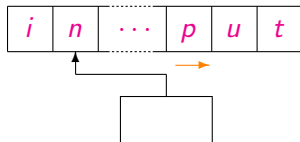
² Université Nice-Sophia Antipolis and
École Normale Supérieure de Lyon
France



³ Dipartimento di Informatica e Comunicazione
Università degli Studi di Milano
Italy

LATA 2012 – A Coruña, Spain – March 7, 2012

Finite State Automata



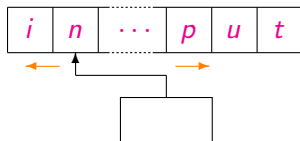
Base versions:

- ▶ one-way deterministic (1DFA)
- ▶ one-way nondeterministic (1NFA)

Possible variants:

- ▶ *two-way automata*: input head moving forth and back
 - 2DFA
 - 2NFA
- ▶ alternating automata
- ▶ ...

Finite State Automata



Base versions:

- ▶ one-way deterministic (1DFA)
- ▶ one-way nondeterministic (1NFA)

Possible variants:

- ▶ *two-way automata*: input head moving forth and back
 - 2DFA
 - 2NFA
- ▶ alternating automata
- ▶ ...

What about the power of these models?

What about the power of these models?

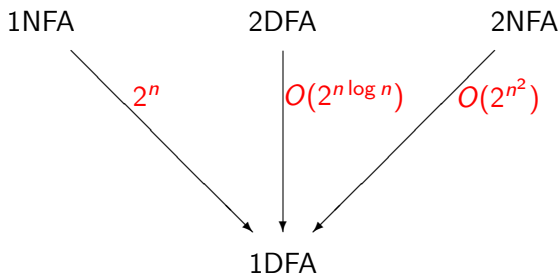
They share the same computational power, namely they characterize the class of *regular languages*,

What about the power of these models?

They share the same computational power, namely they characterize the class of *regular languages*, **however...**

...some of them are more succinct

Costs of the Optimal Simulations Between Automata

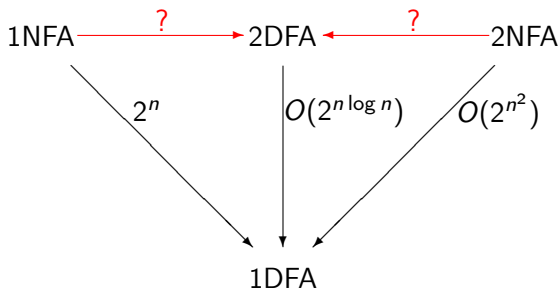


[Rabin&Scott '59, Shepardson '59, Meyer&Fischer '71, ...]

Question

How much the possibility of moving the input head forth and back is useful to eliminate the nondeterminism?

Costs of the Optimal Simulations Between Automata

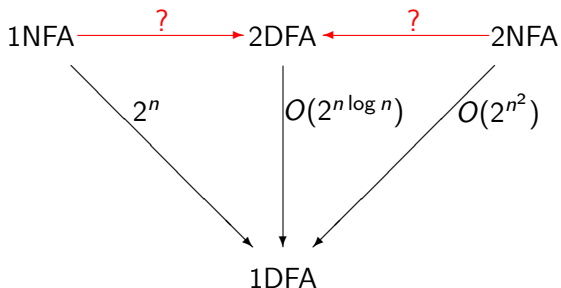


[Rabin&Scott '59, Shepardson '59, Meyer&Fischer '71, ...]

Question

How much the possibility of moving the input head forth and back is useful to eliminate the nondeterminism?

Costs of the Optimal Simulations Between Automata

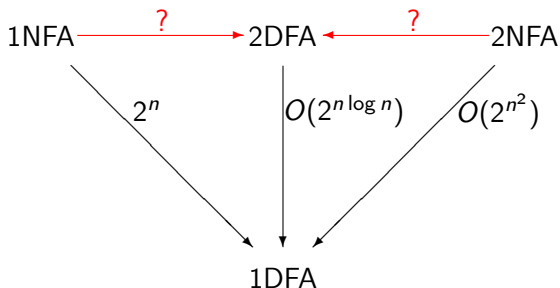


Problem ([Sakoda&Sipser '78])

Do there exist polynomial simulations of

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

Costs of the Optimal Simulations Between Automata



Problem ([Sakoda&Sipser '78])

Do there exist polynomial simulations of

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

Conjecture

These simulations are not polynomial

Sakoda&Sipser Question: Upper and Lower Bounds

- ▶ **Exponential upper bounds**
deriving from the simulations by 1DFAs
- ▶ **Polynomial lower bounds**
for the cost $c(n)$ of simulation of 1NFAs by 2DFAs:
 - $c(n) \in \Omega\left(\frac{n^2}{\log n}\right)$ [Berman&Lingas '77]
 - $c(n) \in \Omega(n^2)$ [Chrobak '86]

Sakoda&Sipser Question: Upper and Lower Bounds

- ▶ **Exponential upper bounds**
deriving from the simulations by 1DFAs
- ▶ **Polynomial lower bounds**
for the cost $c(n)$ of simulation of 1NFAs by 2DFAs:
 - $c(n) \in \Omega\left(\frac{n^2}{\log n}\right)$ [Berman&Lingas '77]
 - $c(n) \in \Omega(n^2)$ [Chrobak '86]

Sakoda and Sipser Question

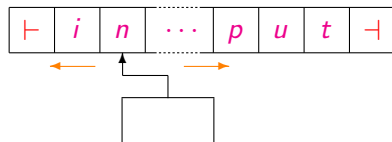
- ▶ Very difficult in its general form
- ▶ Not very encouraging obtained results:

Lower and upper bounds too far
(Polynomial vs exponential)

- ▶ Hence:

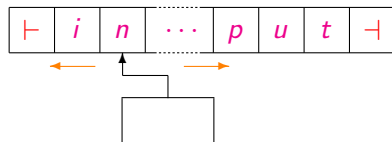
Try to attack restricted versions of the problem!

Two-Way Automata: Few Technical Details



- ▶ **Input surrounded by the endmarkers \vdash and \dashv**
- ▶ $w \in \Sigma^*$ is accepted iff there is a computation
 - with input tape $\vdash w \dashv$
 - starting at the left endmarker \vdash in the initial state
 - reaching a final state (on the left endmarker)

Two-Way Automata: Few Technical Details



- ▶ Input surrounded by the endmarkers \vdash and \dashv
- ▶ $w \in \Sigma^*$ is accepted iff there is a computation
 - with input tape $\vdash w \dashv$
 - starting at the left endmarker \vdash in the initial state
 - reaching a final state (on the left endmarker)

2NFAs vs 2DFAs: Restricted Versions

Previous works:

(i) Restrictions on the *simulating* machines (i.e., resulting 2DFAs)

- ▶ sweeping automata [Sipser '80]
- ▶ oblivious automata [Hromkovič&Schnitger '03]
- ▶ “few reversal” automata [Kapoutsis '11]

(ii) Restrictions on the *languages*

- ▶ unary regular languages [Geffert Mereghetti&Pighizzini '03]

In this work we use a different approach:

(iii) Restrictions on the *simulated* machines (i.e., given 2NFAs)

2NFAs vs 2DFAs: Restricted Versions

Previous works:

- (i) Restrictions on the *simulating* machines (i.e., resulting 2DFAs)
 - ▶ sweeping automata [Sipser '80]
 - ▶ oblivious automata [Hromkovič&Schnitger '03]
 - ▶ “few reversal” automata [Kapoutsis '11]

- (ii) Restrictions on the *languages*
 - ▶ unary regular languages [Geffert Mereghetti&Pighizzini '03]

In this work we use a different approach:

- (iii) Restrictions on the *simulated* machines (i.e., given 2NFAs)

2NFAs vs 2DFAs: Restricted Versions

Previous works:

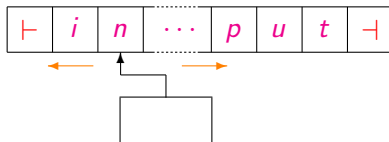
- (i) Restrictions on the *simulating* machines (i.e., resulting 2DFAs)
 - ▶ sweeping automata [Sipser '80]
 - ▶ oblivious automata [Hromkovič&Schnitger '03]
 - ▶ “few reversal” automata [Kapoutsis '11]

- (ii) Restrictions on the *languages*
 - ▶ unary regular languages [Geffert Mereghetti&Pighizzini '03]

In this work we use a different approach:

- (iii) Restrictions on the *simulated* machines (i.e., given 2NFAs)

Outer Nondeterministic Automata (ONFAs)



In the paper, we consider the following model:

Definition

A two-way automaton is said to be *outer nondeterministic* iff nondeterministic choices are allowed *only* when the input head is scanning the endmarkers

Unary 2NFAs vs ONFAs

Normal Form for Unary 2NFAs [Geffert Mereghetti&Pighizzini '03]

- ▶ *Nondeterministic choices* only at the endmarkers
- ▶ *Head reversals* only at the endmarkers
- ▶ In each sweep the input length modulo one integer is counted

Unary 2NFAs vs ONFAs

Normal Form for Unary 2NFAs [Geffert Mereghetti&Pighizzini '03]

- ▶ *Nondeterministic choices* only at the endmarkers
- ▶ *Head reversals* only at the endmarkers
- ▶ In each sweep the input length modulo one integer is counted

Outer Nondeterministic Automata

- ▶ No restrictions on the *input alphabet*
- ▶ No restrictions on *head reversals*
- ▶ *Deterministic transitions* on “real” input symbols
- ▶ *Nondeterministic choices* only at the endmarkers

Unary 2NFAs vs ONFAs

Normal Form for Unary 2NFAs [Geffert Mereghetti&Pighizzini '03]

- ▶ *Nondeterministic choices* only at the endmarkers
- ▶ *Head reversals* only at the endmarkers
- ▶ In each sweep the input length modulo one integer is counted

Outer Nondeterministic Automata

- ▶ No restrictions on the *input alphabet*
- ▶ No restrictions on *head reversals*
- ▶ *Deterministic transitions* on “real” input symbols
- ▶ *Nondeterministic choices* only at the endmarkers

Unary 2NFAs are a very restricted version of 2ONFAs!

- ▶ We extended to 2ONFAs previous results on unary 2NFAs

Outer nondeterministic automata (ONFAs): tools

Main tool: procedure *reach*(p, q)

- ▶ Checks the existence of a computation segment
 - from the left endmarker in the state p
 - to the left endmarker in the state q
 - not visiting the left endmarker in between

Outer nondeterministic automata (ONFAs): tools

Main tool: procedure *reach*(p, q)

- ▶ Checks the existence of a computation segment
 - from the left endmarker in the state p
 - to the left endmarker in the state q
 - not visiting the left endmarker in between

Outer nondeterministic automata (ONFAs): tools

Main tool: procedure *reach*(p, q)

- ▶ Checks the existence of a computation segment
 - from the left endmarker in the state p
 - to the left endmarker in the state q
 - not visiting the left endmarker in between

Accepting computation:

sequence of states q_0, q_1, \dots, q_f visited at the left endmarker:

- ▶ q_0 initial state
- ▶ for $i = 1, \dots, f$ $reach(q_{i-1}, q_i) = true$
- ▶ q_f final state

Outer nondeterministic automata (ONFAs): tools

- ▶ How to deal with loops?
- ▶ Two kinds of loops:
 - loops visiting the endmarkers
 - loops inside the “real” input

Outer nondeterministic automata (ONFAs): tools

- ▶ How to deal with loops?
- ▶ Two kinds of loops:
 - loops visiting the endmarkers
 - loops inside the “real” input

Loops visiting the endmarkers

- ▶ Loops involving endmarkers can contain nondeterministic choices
- ▶ If a computation visits the left endmarker twice in the same state q then there is a shorter “equivalent” computation
- ▶ We can consider only computations visiting the left endmarker $\leq \#Q$ times

Loops visiting the endmarkers

- ▶ Loops involving endmarkers can contain nondeterministic choices
- ▶ If a computation visits the left endmarker twice in the same state q then there is a shorter “equivalent” computation
- ▶ We can consider only computations visiting the left endmarker $\leq \#Q$ times

Loops visiting the endmarkers

- ▶ Loops involving endmarkers can contain nondeterministic choices
- ▶ If a computation visits the left endmarker twice in the same state q then there is a shorter “equivalent” computation
- ▶ We can consider only computations visiting the left endmarker $\leq \#Q$ times

Loops inside the “real” input

Procedure $reach(p, q)$:

- ▶ “Backward search” from q to p
- ▶ In this way loops are avoided
- ▶ Finite control with a *linear number of states*

The technique:

- ▶ Introduced by Sipser for the complementation of space bounded Turing machines [Sipser '80]
- ▶ Modified for the complementation of 2DFAs [Geffert Mereghetti&Pighizzini '07]
- ▶ Extended in our paper to 2ONFAs

Loops inside the “real” input

Procedure $reach(p, q)$:

- ▶ “Backward search” from q to p
- ▶ In this way loops are avoided
- ▶ Finite control with a *linear number of states*

The technique:

- ▶ Introduced by Sipser for the complementation of space bounded Turing machines [Sipser '80]
- ▶ Modified for the complementation of 2DFAs [Geffert Mereghetti&Pighizzini '07]
- ▶ Extended in our paper to 2ONFAs

Results

(i) Subexponential simulation of 2ONFAs by 2DFAs

Verify that q_f is reachable from q_0
by visiting the left endmarker $\leq \#Q$ times
(divide-and-conquer algorithm)

(ii) Polynomial complementation of 2ONFAs

Inductive counting argument

(iii) Polynomial simulation of 2ONFAs by 2DFAs

under the condition $L = NL$

Reduction to graph accessibility problem

(iv) Polynomial simulation of 2ONFAs by unambiguous 2ONFAs

Reduction to graph accessibility problem

combined with $NL/poly \subseteq UL$ [Reinhardt&Allender '00]

Results

- (i) Subexponential simulation of 2ONFAs by 2DFAs
Verify that q_f is reachable from q_0
by visiting the left endmarker $\leq \#Q$ times
(divide-and-conquer algorithm)
- (ii) Polynomial complementation of 2ONFAs
Inductive counting argument
- (iii) Polynomial simulation of 2ONFAs by 2DFAs
under the condition $L = NL$
Reduction to graph accessibility problem
- (iv) Polynomial simulation of 2ONFAs by unambiguous 2ONFAs
Reduction to graph accessibility problem
combined with $NL/poly \subseteq UL$ [Reinhardt&Allender '00]

Results

- (i) Subexponential simulation of 2ONFAs by 2DFAs
Verify that q_f is reachable from q_0
by visiting the left endmarker $\leq \#Q$ times
(divide-and-conquer algorithm)
- (ii) Polynomial complementation of 2ONFAs
Inductive counting argument
- (iii) Polynomial simulation of 2ONFAs by 2DFAs
under the condition $L = NL$
Reduction to graph accessibility problem
- (iv) Polynomial simulation of 2ONFAs by unambiguous 2ONFAs
Reduction to graph accessibility problem
combined with $NL/poly \subseteq UL$ [Reinhardt&Allender '00]

Results

- (i) Subexponential simulation of 2ONFAs by 2DFAs
Verify that q_f is reachable from q_0
by visiting the left endmarker $\leq \#Q$ times
(divide-and-conquer algorithm)

- (ii) Polynomial complementation of 2ONFAs
Inductive counting argument

- (iii) Polynomial simulation of 2ONFAs by 2DFAs
under the condition $L = NL$
Reduction to graph accessibility problem

- (iv) Polynomial simulation of 2ONFAs by unambiguous 2ONFAs
Reduction to graph accessibility problem
combined with $NL/poly \subseteq UL$ [Reinhardt&Allender '00]

Results: Alternating Case (2ONFAs)

At the endmarkers, universal and existential states are allowed

(v) Polynomial simulation of 2OAFAs by 2DFAs under $L = P$

(vi) Polynomial simulation of 2OAFAs by 2NFAs under $NL = P$

For both:

Reduction to the Alternating Graph Accessibility Problem

Results: Alternating Case (2ONFAs)

At the endmarkers, universal and existential states are allowed

(v) Polynomial simulation of 2OAFAs by 2DFAs under $L = P$

(vi) Polynomial simulation of 2OAFAs by 2NFAs under $NL = P$

For both:

Reduction to the Alternating Graph Accessibility Problem

Final Remarks

- ▶ We extended several results from the unary to the general case for 2NFAs
- ▶ In the unary case, restricting the nondeterminism to the endmarkers does not significantly change the size of 2NFAs (normal form)
- ▶ In the general case, is there some “simple way” to restrict the nondeterminism?
- ▶ Does it is possible to extend our results to some wider class of 2NFAs?
- ▶ Interesting connections with complexity theory:
 - Results connected with classical complexity questions
 - Proof techniques derived from space complexity

Final Remarks

- ▶ We extended several results from the unary to the general case for 2NFAs
- ▶ In the unary case, restricting the nondeterminism to the endmarkers does not significantly change the size of 2NFAs (normal form)
- ▶ In the general case, is there some “simple way” to restrict the nondeterminism?
- ▶ Does it is possible to extend our results to some wider class of 2NFAs?
- ▶ Interesting connections with complexity theory:
 - Results connected with classical complexity questions
 - Proof techniques derived from space complexity

Final Remarks

- ▶ We extended several results from the unary to the general case for 2NFAs
- ▶ In the unary case, restricting the nondeterminism to the endmarkers does not significantly change the size of 2NFAs (normal form)
- ▶ In the general case, is there some “simple way” to restrict the nondeterminism?
- ▶ Does it is possible to extend our results to some wider class of 2NFAs?
- ▶ Interesting connections with complexity theory:
 - Results connected with classical complexity questions
 - Proof techniques derived from space complexity

Final Remarks

- ▶ We extended several results from the unary to the general case for 2NFAs
- ▶ In the unary case, restricting the nondeterminism to the endmarkers does not significantly change the size of 2NFAs (normal form)
- ▶ In the general case, is there some “simple way” to restrict the nondeterminism?
- ▶ Does it is possible to extend our results to some wider class of 2NFAs?
- ▶ Interesting connections with complexity theory:
 - Results connected with classical complexity questions
 - Proof techniques derived from space complexity

Final Remarks

- ▶ We extended several results from the unary to the general case for 2NFAs
- ▶ In the unary case, restricting the nondeterminism to the endmarkers does not significantly change the size of 2NFAs (normal form)
- ▶ In the general case, is there some “simple way” to restrict the nondeterminism?
- ▶ Does it is possible to extend our results to some wider class of 2NFAs?
- ▶ Interesting connections with complexity theory:
 - Results connected with classical complexity questions
 - Proof techniques derived from space complexity