Descriptional Complexity and Parikh Equivalence

Giovanna Janet Lavado

Università degli Studi di Milano

Supervisor: Prof. Giovanni Pighizzini
Co-supervisor: Prof. Carlo Mereghetti
Coordinator: Ernesto Damiani

March 13, 2015

Research in Descriptional Complexity

How succinctly a formal language can be described by a formalism in comparison with other formalisms.

Take the *length of description* as complexity measure.

- What is the blow-up when changing from one model to another?
- Are there languages such that this blow-up is achieved?

Investigation of *upper bounds* and *lower bounds*.

Standard equivalence: 1NFAs vs 1DFAs

Subset construction [Rabin&Scott '59]

1NFA
n states
L

$\Longrightarrow$

1DFA
2^n states
L

Moreover, this state bound cannot be reduced in the worst case
[Meyer&Fischer '71, Moore '71]

What happens if we do not care of the order of symbols in the strings?

This problem is related to the concept of *Parikh equivalence*
[Parikh 1966]

Standard equivalence: 1NFAs vs 1DFAs

Subset construction [Rabin&Scott '59]

1NFA
n states
L

$\Longrightarrow$

1DFA
2^n states
L

Moreover, this state bound cannot be reduced in the worst case
[Meyer&Fischer '71, Moore '71]

What happens if we do not care of the order of symbols in the strings?

This problem is related to the concept of *Parikh equivalence*
[Parikh 1966]

Parikh mapping: preliminaries

- $\Sigma = \{a_1, \ldots, a_m\}$ alphabet of m symbols
- $|w|_a$ be the number of occurrences of a in $w \in \Sigma^*$

Parikh map

The *Parikh map* $\psi : \Sigma^* \to \mathbb{N}^m$ associates with a word $w \in \Sigma^*$ the m-dimensional nonnegative vector $(|w|_{a_1}, |w|_{a_2}, \ldots, |w|_{a_m})$.

Parikh image

The *Parikh image of a language* $L \subseteq \Sigma^*$ is $\psi(L) = \{\psi(w) \mid w \in L\}$.

- $w_1 =_\pi w_2$ if $\psi(w_1) = \psi(w_2)$
- $L_1 =_\pi L_2$ if $\psi(L_1) = \psi(L_2)$

Parikh equivalence: Parikh's theorem

Theorem ([Parikh 1966])

For each context-free language $L \subseteq \Sigma^*$, there exists a Parikh equivalent regular language $R \subseteq \Sigma^*$.

Example ($L =_\pi R$)

$$L = \{a^n b^n \mid n \geq 0\} \quad \text{and} \quad R = (ab)^*$$

have the same Parikh image, namely the set

$$\{(n, n) \mid n \geq 0\}$$

Parikh mapping

Motivations and current research

Interesting theoretical properties:

- wrt Parikh equivalence regular and context-free languages are indistinguishable [Parikh 1966]

Connections with number theory and logic:

- Semilinear sets [Ginsburg&Spanier 1964, Ginsburg&Spanier 1966]
- Presburger arithmetics [Presburger 1929]
- Formal verification [To '10]

Applications in several fields:

- Abelian (approximate) pattern matching, sequence alignment [Christodoulakis&Christou '12, Benson '03]

Our Goal

We investigate the conversion of 1NFAs and CFGs into *Parikh equivalent* 1DFAs and 2DFAs, from a *descriptional complexity* point of view.

Problem (1NFAs to 1DFAs and 2DFAs)

1NFA
n states

$\Longrightarrow_{\pi}$

1DFA, 2DFA
how many states?

Problem (CFGs to 1DFAs and 2DFAs)

CFG
in Chomsky normal form
h variables

$\Longrightarrow_{\pi}$

1DFA, 2DFA
how many states?

From 1NFAs to Parikh equivalent 1DFAs

Our first contribution:

Problem (1NFAs to 1DFAs)

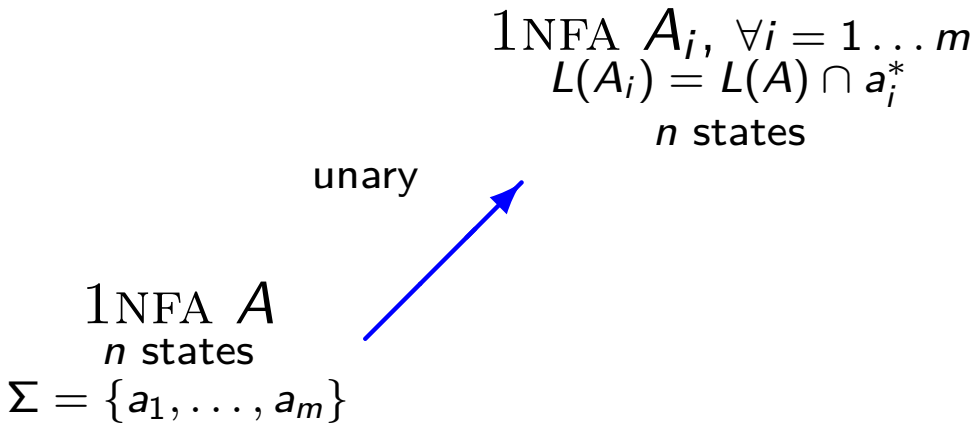
1NFA		1DFA
n states	$\Longrightarrow_{\pi}$	*how many states?*
L_1		L_2

- Upper bound: 2^n
  by subset construction [Rabin&Scott '59]
- Lower bound: $e^{\sqrt{n \ln n}}$
  This bound derives from the *unary case*:
  the state cost of the conversion of unary n-state 1NFAs
  into equivalent 1DFAs is $e^{\Theta(\sqrt{n \ln n})}$ [Chrobak '86]

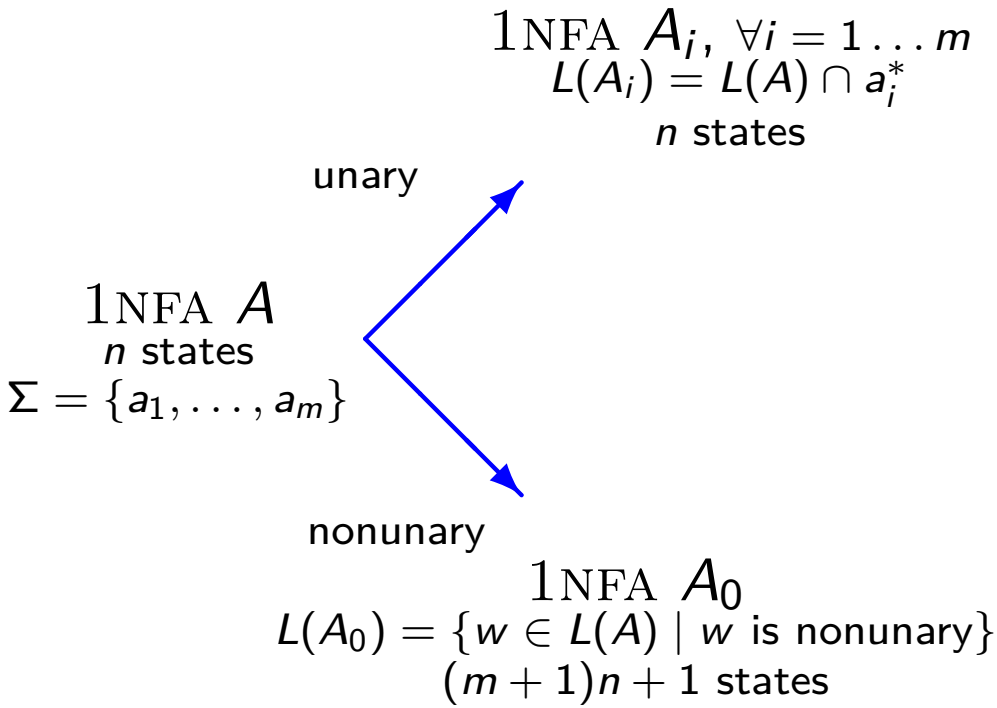
From 1NFAs to Parikh equivalent 1DFAs: general idea

1NFA A
n states
$\Sigma = \{a_1, \ldots, a_m\}$

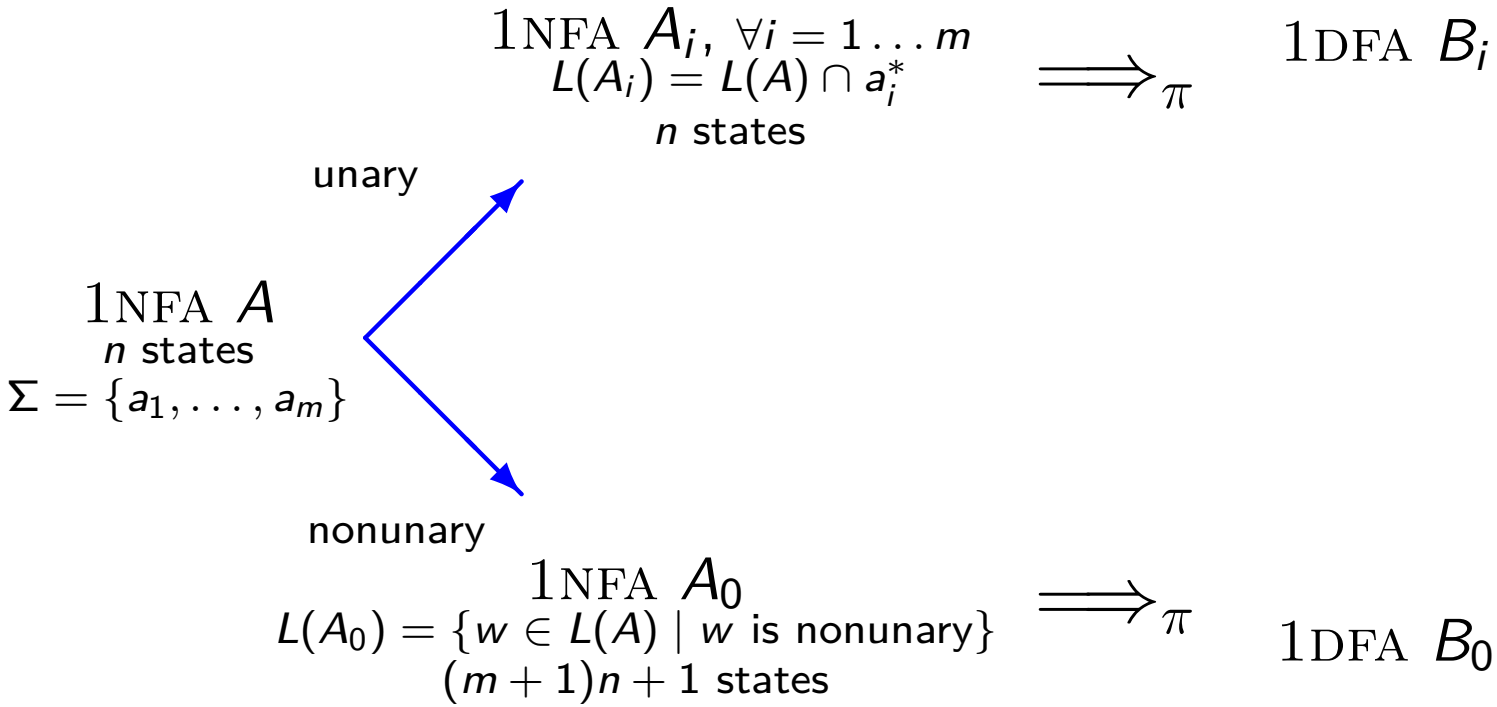
From 1NFAs to Parikh equivalent 1DFAs: general idea



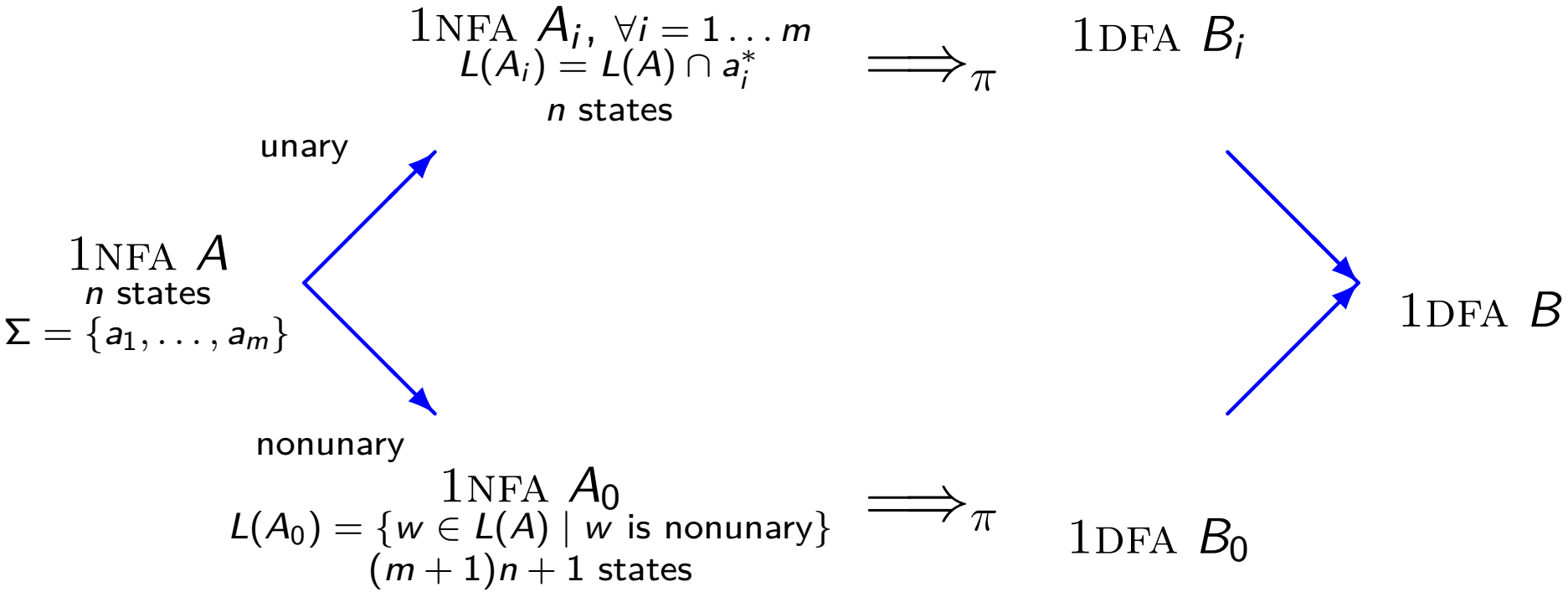
From 1NFAs to Parikh equivalent 1DFAs: general idea



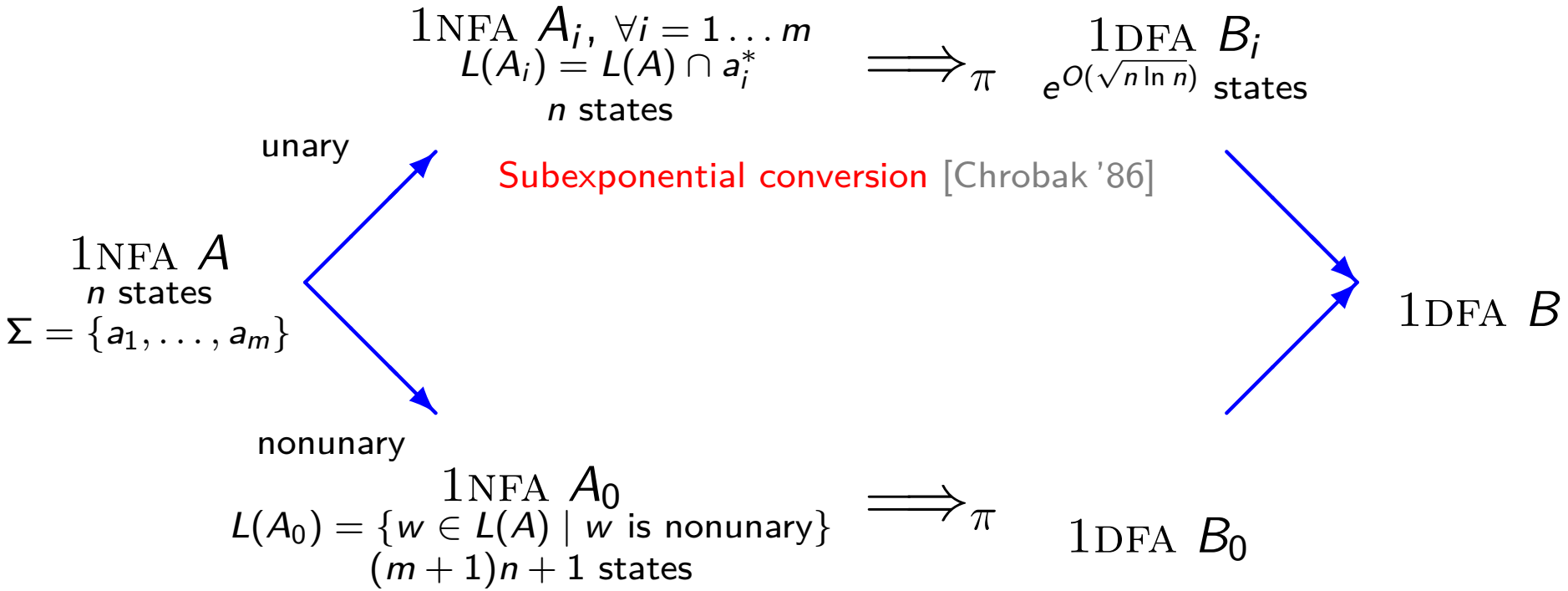
From 1NFAs to Parikh equivalent 1DFAs: general idea



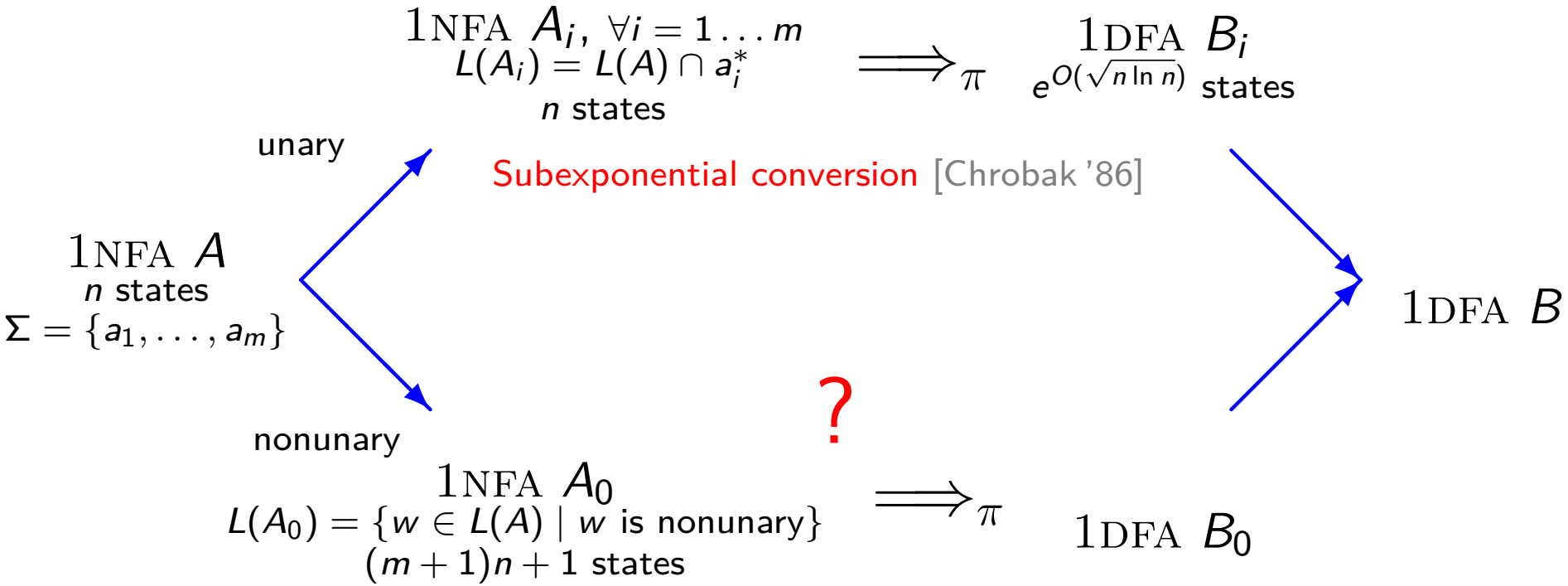
From 1NFAs to Parikh equivalent 1DFAs: general idea



From 1NFAs to Parikh equivalent 1DFAs: general idea

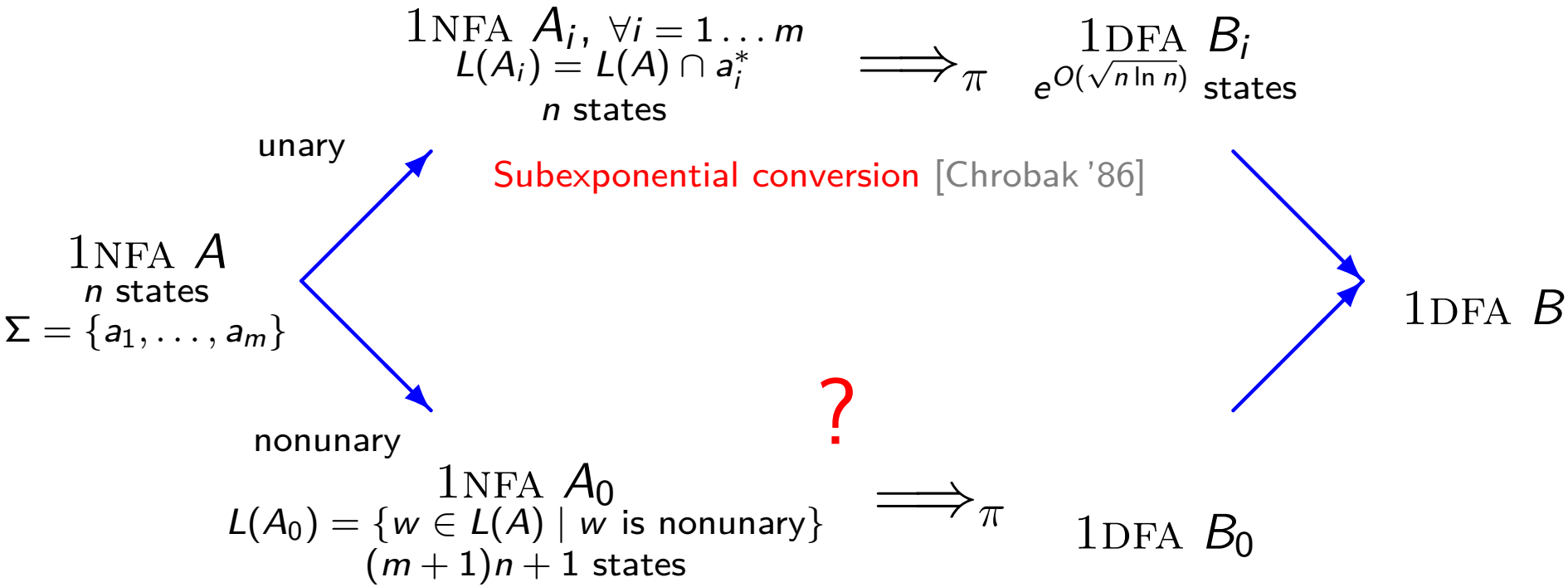


From 1NFAs to Parikh equivalent 1DFAs: general idea



How much does it cost the conversion of 1NFAs accepting *only nonunary strings* into Parikh equivalent 1DFAs?

From 1NFAs to Parikh equivalent 1DFAs: general idea



How much does it cost the conversion of 1NFAs accepting *only nonunary strings* into Parikh equivalent 1DFAs?

Only polynomial!

(less than in unary case)

Converting 1NFAs accepting only nonunary strings

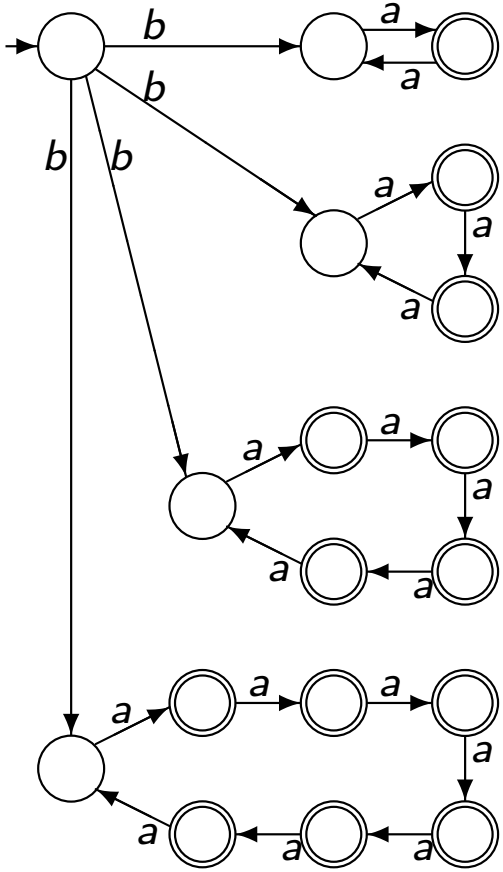
An example

$$L = \{ba^n \mid n \bmod 210 \neq 0\}$$

Converting 1NFAs accepting only nonunary strings

An example

$$L = \{ba^n \mid n \bmod 210 \neq 0\}$$



$L_1 = \{ba^n \mid n \bmod 2 \neq 0\}$

$L_2 = \{ba^n \mid n \bmod 3 \neq 0\}$

$L_3 = \{ba^n \mid n \bmod 5 \neq 0\}$

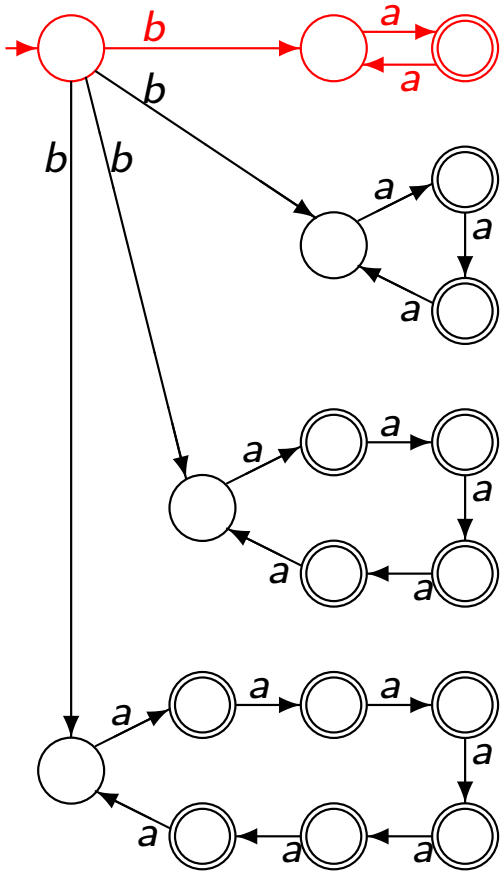
$L_4 = \{ba^n \mid n \bmod 7 \neq 0\}$

1DFA $\geq$ 211 states

Converting 1NFAs accepting only nonunary strings

An example

$$L = \{ba^n \mid n \bmod 210 \neq 0\}$$



$L_1 = \{ba^n \mid n \bmod 2 \neq 0\}$

$L_2 = \{ba^n \mid n \bmod 3 \neq 0\}$

$L_3 = \{ba^n \mid n \bmod 5 \neq 0\}$

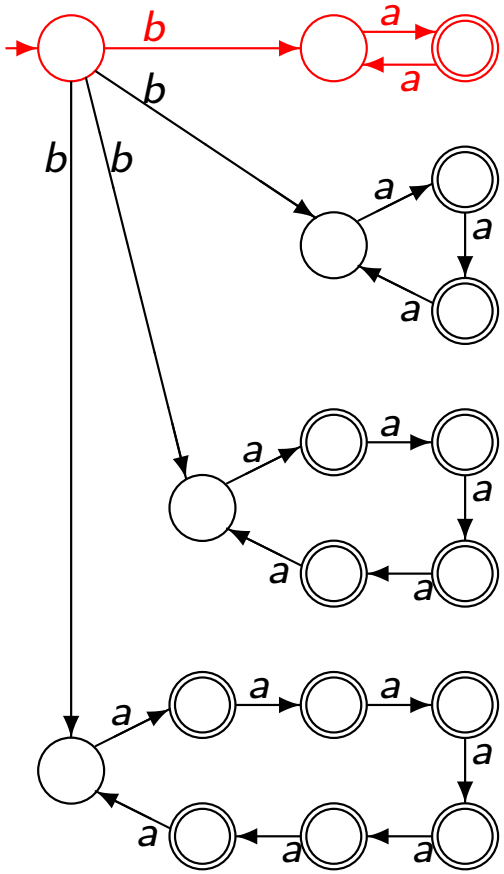
$L_4 = \{ba^n \mid n \bmod 7 \neq 0\}$

1DFA $\geq$ 211 states

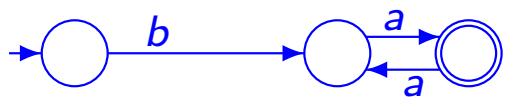
Converting 1NFAs accepting only nonunary strings

An example

$$L = \{ba^n \mid n \bmod 210 \neq 0\}$$



$L_1 = \{ba^n \mid n \bmod 2 \neq 0\}$



$L_1' = \{ba^n \mid n \bmod 2 \neq 0\}$

$L_2 = \{ba^n \mid n \bmod 3 \neq 0\}$

$L_3 = \{ba^n \mid n \bmod 5 \neq 0\}$

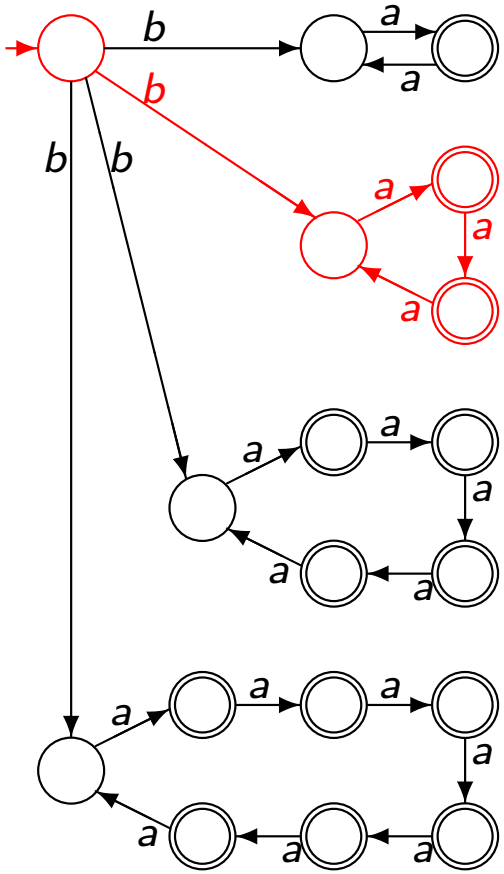
$L_4 = \{ba^n \mid n \bmod 7 \neq 0\}$

$1\text{DFA} \geq 211$ states

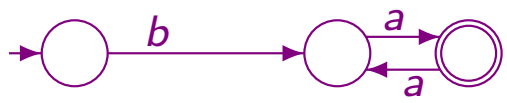
Converting 1NFAs accepting only nonunary strings

An example

$$L = \{ba^n \mid n \bmod 210 \neq 0\}$$



$L_1 = \{ba^n \mid n \bmod 2 \neq 0\}$



$L_1' = \{ba^n \mid n \bmod 2 \neq 0\}$

$L_2 = \{ba^n \mid n \bmod 3 \neq 0\}$

$L_3 = \{ba^n \mid n \bmod 5 \neq 0\}$

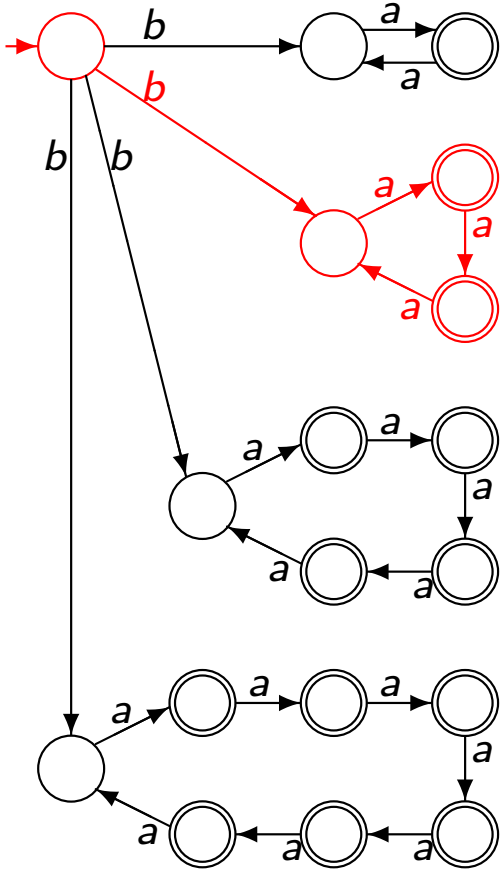
$L_4 = \{ba^n \mid n \bmod 7 \neq 0\}$

$1\text{DFA} \geq 211$ states

Converting 1NFAs accepting only nonunary strings

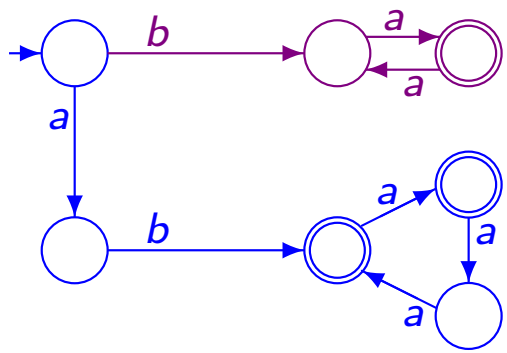
An example

$$L = \{ba^n \mid n \bmod 210 \neq 0\}$$



$L_1 = \{ba^n \mid n \bmod 2 \neq 0\}$

$L_1' = \{ba^n \mid n \bmod 2 \neq 0\}$



$L_2 = \{ba^n \mid n \bmod 3 \neq 0\}$

$L_2' = \{aba^{n-1} \mid n \bmod 3 \neq 0\}$

$L_3 = \{ba^n \mid n \bmod 5 \neq 0\}$

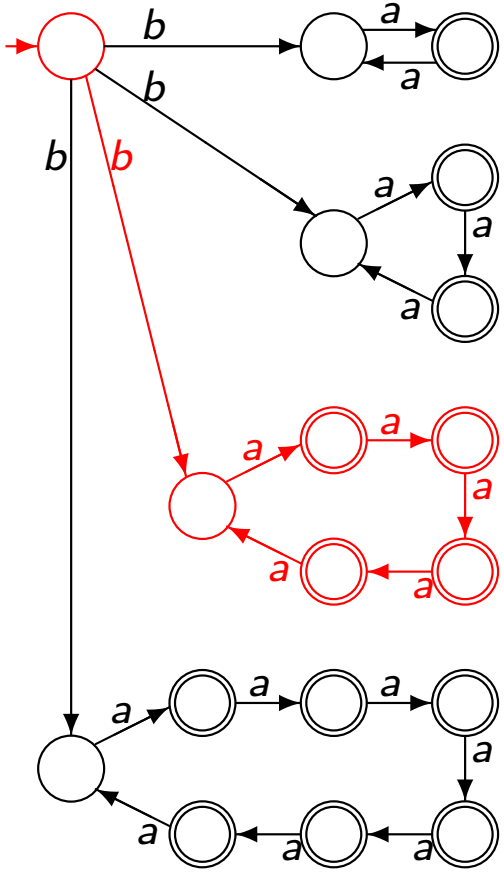
$L_4 = \{ba^n \mid n \bmod 7 \neq 0\}$

$1\text{DFA} \geq 211$ states

Converting 1NFAs accepting only nonunary strings

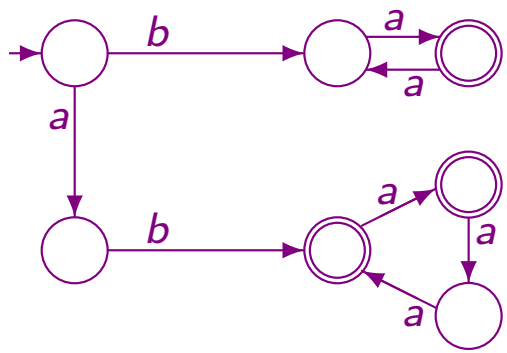
An example

$$L = \{ba^n \mid n \bmod 210 \neq 0\}$$



$L_1 = \{ba^n \mid n \bmod 2 \neq 0\}$

$L_1' = \{ba^n \mid n \bmod 2 \neq 0\}$



$L_2 = \{ba^n \mid n \bmod 3 \neq 0\}$

$L_2' = \{aba^{n-1} \mid n \bmod 3 \neq 0\}$

$L_3 = \{ba^n \mid n \bmod 5 \neq 0\}$

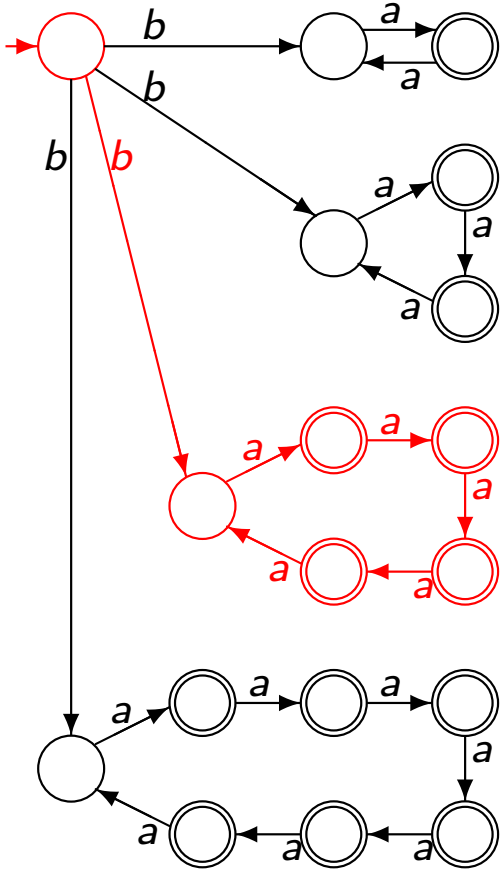
$L_4 = \{ba^n \mid n \bmod 7 \neq 0\}$

$1\text{DFA} \geq 211$ states

Converting 1NFAs accepting only nonunary strings

An example

$$L = \{ba^n \mid n \bmod 210 \neq 0\}$$



$$L_1 = \{ba^n \mid n \bmod 2 \neq 0\}$$

$$L_2 = \{ba^n \mid n \bmod 3 \neq 0\}$$

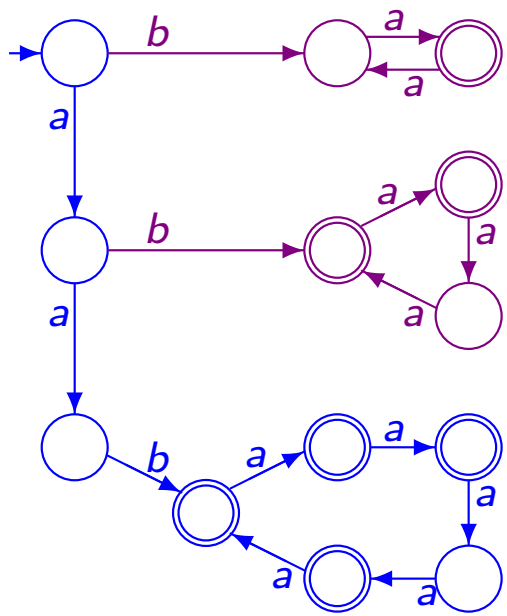
$$L_3 = \{ba^n \mid n \bmod 5 \neq 0\}$$

$$L_4 = \{ba^n \mid n \bmod 7 \neq 0\}$$

$$L_1' = \{ba^n \mid n \bmod 2 \neq 0\}$$

$$L_2' = \{aba^{n-1} \mid n \bmod 3 \neq 0\}$$

$$L_3' = \{a^2ba^{n-2} \mid n \bmod 5 \neq 0\}$$

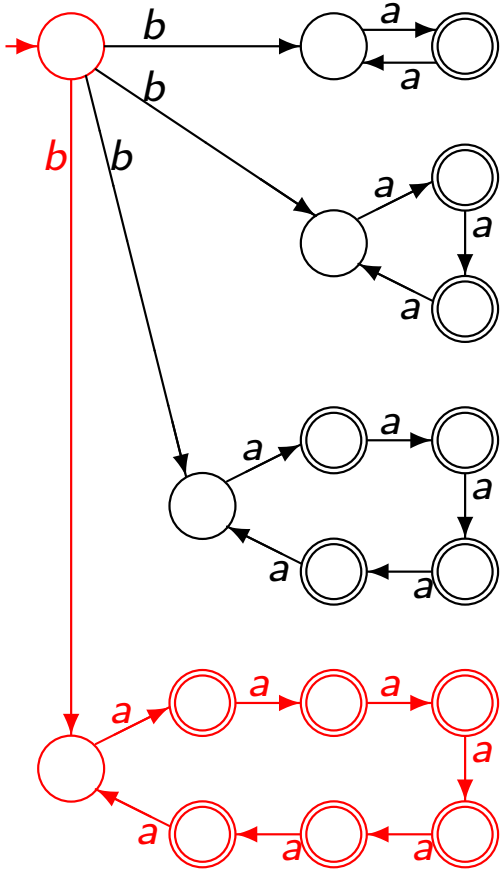


$1\text{DFA} \geq 211$ states

Converting 1NFAs accepting only nonunary strings

An example

$$L = \{ba^n \mid n \bmod 210 \neq 0\}$$



$$L_1 = \{ba^n \mid n \bmod 2 \neq 0\}$$

$$L_2 = \{ba^n \mid n \bmod 3 \neq 0\}$$

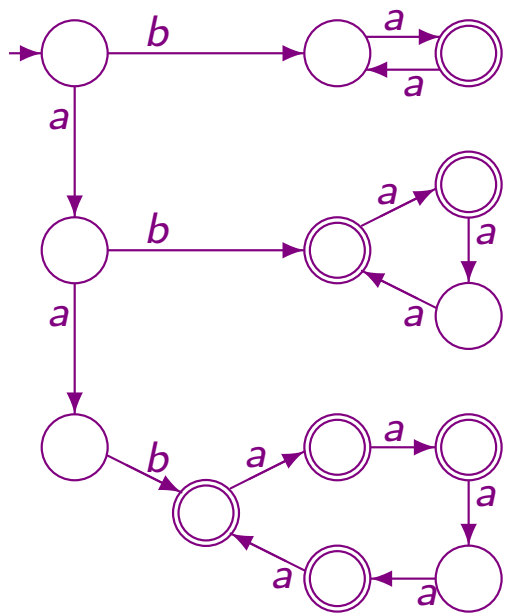
$$L_3 = \{ba^n \mid n \bmod 5 \neq 0\}$$

$$L_4 = \{ba^n \mid n \bmod 7 \neq 0\}$$

$$L_1' = \{ba^n \mid n \bmod 2 \neq 0\}$$

$$L_2' = \{aba^{n-1} \mid n \bmod 3 \neq 0\}$$

$$L_3' = \{a^2ba^{n-2} \mid n \bmod 5 \neq 0\}$$

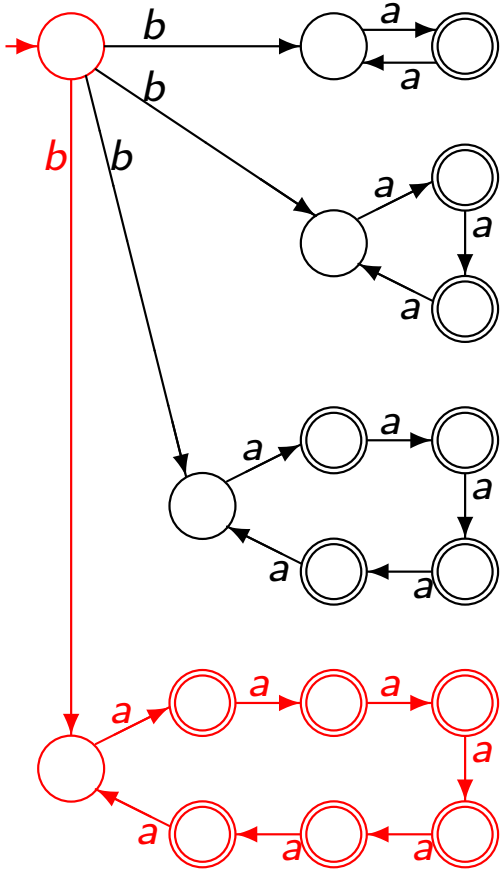


1DFA $\geq$ 211 states

Converting 1NFAs accepting only nonunary strings

An example

$$L = \{ba^n \mid n \bmod 210 \neq 0\}$$



$$L_1 = \{ba^n \mid n \bmod 2 \neq 0\}$$

$$L_2 = \{ba^n \mid n \bmod 3 \neq 0\}$$

$$L_3 = \{ba^n \mid n \bmod 5 \neq 0\}$$

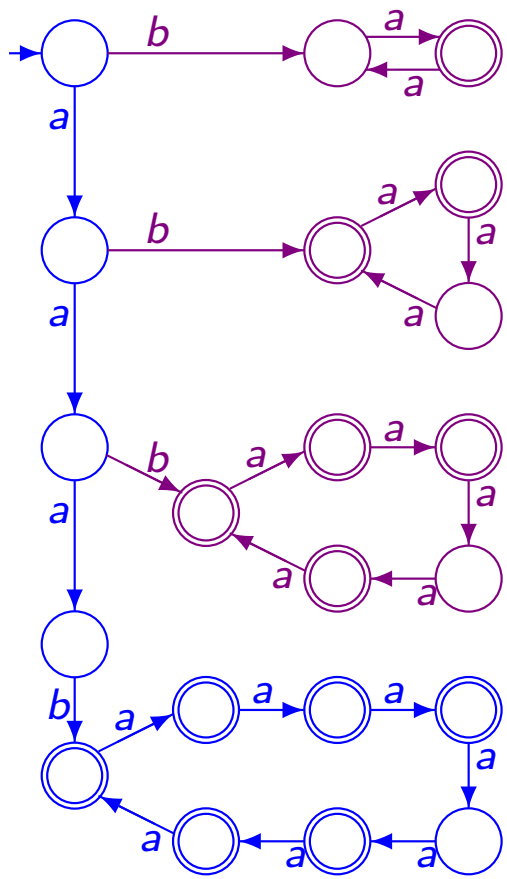
$$L_4 = \{ba^n \mid n \bmod 7 \neq 0\}$$

$$L'_1 = \{ba^n \mid n \bmod 2 \neq 0\}$$

$$L'_2 = \{aba^{n-1} \mid n \bmod 3 \neq 0\}$$

$$L'_3 = \{a^2ba^{n-2} \mid n \bmod 5 \neq 0\}$$

$$L'_4 = \{a^3ba^{n-3} \mid n \bmod 7 \neq 0\}$$

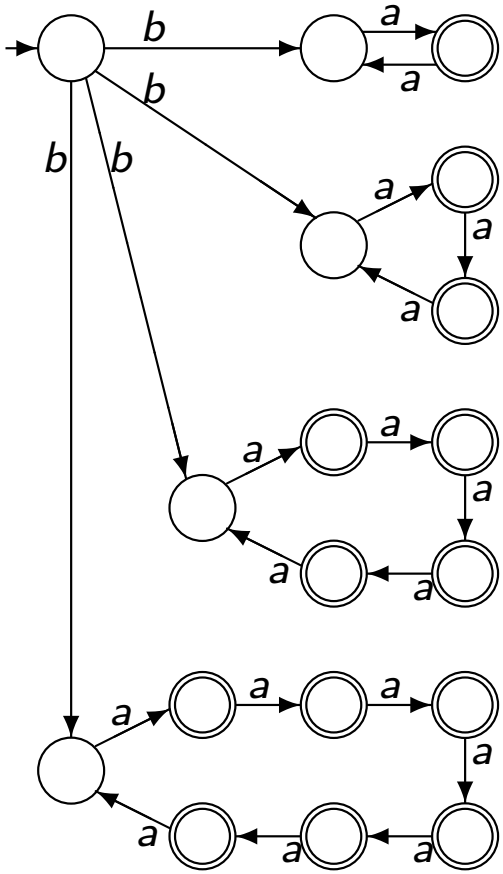


$1\text{DFA} \geq 211$ states

Converting 1NFAs accepting only nonunary strings

An example

$$L = \{ba^n \mid n \bmod 210 \neq 0\}$$



$L_1 = \{ba^n \mid n \bmod 2 \neq 0\}$

$L'_1 = \{ba^n \mid n \bmod 2 \neq 0\}$

$L_2 = \{ba^n \mid n \bmod 3 \neq 0\}$

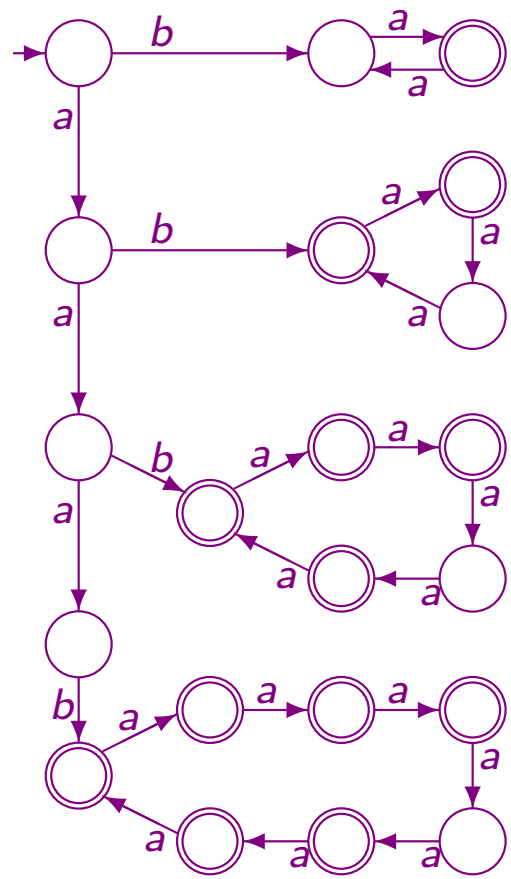
$L'_2 = \{aba^{n-1} \mid n \bmod 3 \neq 0\}$

$L_3 = \{ba^n \mid n \bmod 5 \neq 0\}$

$L'_3 = \{a^2ba^{n-2} \mid n \bmod 5 \neq 0\}$

$L_4 = \{ba^n \mid n \bmod 7 \neq 0\}$

$L'_4 = \{a^3ba^{n-3} \mid n \bmod 7 \neq 0\}$



$1\text{DFA} \geq 211$ states

$L' = L'_1 \cup L'_2 \cup L'_3 \cup L'_4$

1DFA with only 21 states!

Converting 1NFAs accepting only nonunary strings

The conversion uses a modification of the following result:

Theorem ([Kopczyński&To '10])

Given $\Sigma = \{a_1, \ldots, a_m\}$, there is a polynomial p s.t. for each n-state 1NFA A over Σ,

$$\psi(L(A)) = \bigcup_{i \in I} Z_i$$

where:

- *I is a set of at most $p(n)$ indices*
- *for $i \in I$, $Z_i \subseteq \mathbb{N}^m$ is a linear set of the form:*

$$Z_i = \{\alpha_0 + n_1\alpha_1 + \cdots + n_k\alpha_k \mid n_1, \ldots, n_k \in \mathbb{N}\}$$

with

  - $0 \leq k \leq m$
  - *the components of α_0 are bounded by $p(n)$*
  - *$\alpha_1, \ldots, \alpha_k$ are linearly independent vectors from $\{0, 1, \ldots, n\}^m$*

Converting 1NFAs accepting only nonunary strings

Outline: linear sets

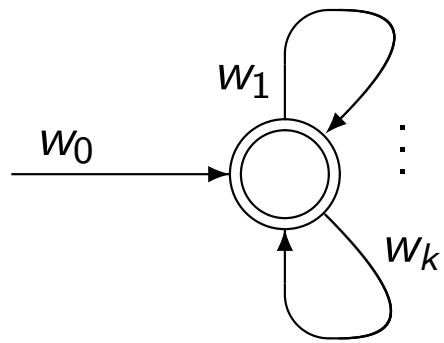
Each above linear set

$$Z_i = \{\alpha_0 + n_1\alpha_1 + \cdots + n_k\alpha_k \mid n_1, \ldots, n_k \in \mathbb{N}\}$$

can be converted into a poly size 1DFA accepting a language

$$R_i = w_0(w_1 + \cdots + w_k)^*$$

s.t. $\psi(w_j) = \alpha_j$, $j = 0, \ldots, k$, and
$w_1, \ldots, w_k$ *begin with different letters*



Converting 1NFAs accepting only nonunary strings

Outline: linear sets

Each above linear set

$$Z_i = \{\alpha_0 + n_1\alpha_1 + \cdots + n_k\alpha_k \mid n_1, \ldots, n_k \in \mathbb{N}\}$$

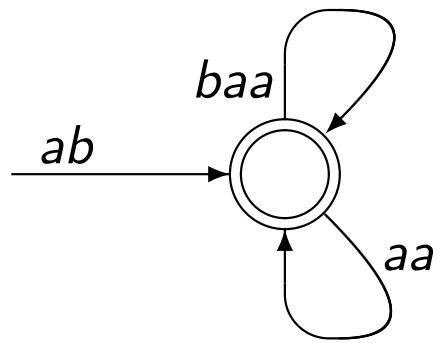
can be converted into a poly size 1DFA accepting a language

$$R_i = w_0(w_1 + \cdots + w_k)^*$$

s.t. $\psi(w_j) = \alpha_j$, $j = 0, \ldots, k$, and
$w_1, \ldots, w_k$ *begin with different letters*

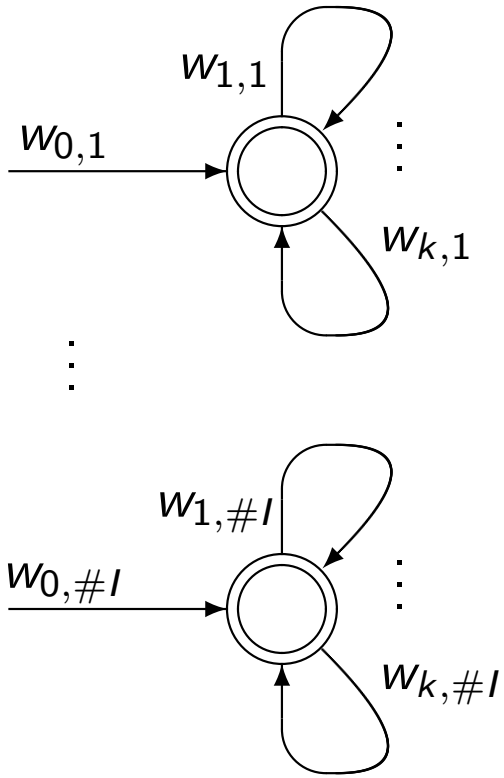
Example:

- $\{(1,1) + n_1(2,1) + n_2(2,0) \mid n_1, n_2 \geq 0\}$
- $ab(baa + aa)^*$



Converting 1NFAs accepting only nonunary strings

Outline: from linear to semilinear



- Standard construction for union of 1DFAs: number of states = *product*

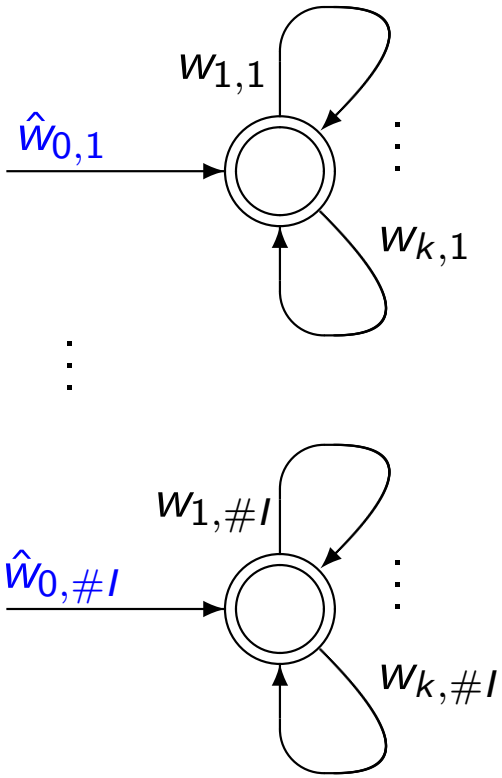
  $\#I \leq p(n) \Rightarrow$ Too large!!!

- Strings $w_{0,i}$ can be replaced by Parikh equivalent strings $\hat{w}_{0,i}$ in such a way that $W_0 = \{\hat{w}_{0,i} \mid i \in I\}$ is a *prefix code*

- After this change:
  number of states $\leq$ *sum* Polynomial!!!

Converting 1NFAs accepting only nonunary strings

Outline: from linear to semilinear



- Standard construction for union of 1DFAs: number of states = *product*

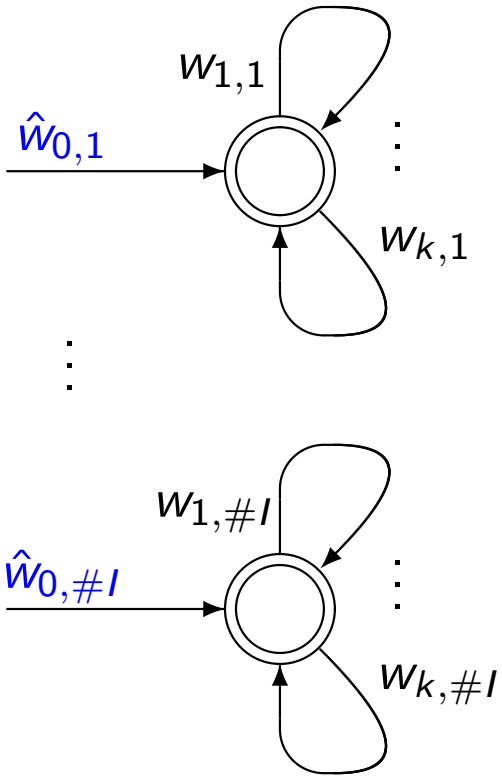
  $\#I \leq p(n) \Rightarrow$ Too large!!!

- Strings $w_{0,i}$ can be replaced by Parikh equivalent strings $\hat{w}_{0,i}$ in such a way that $W_0 = \{\hat{w}_{0,i} \mid i \in I\}$ is a *prefix code*

- After this change:
  number of states $\leq$ *sum* Polynomial!!!

Converting 1NFAs accepting only nonunary strings

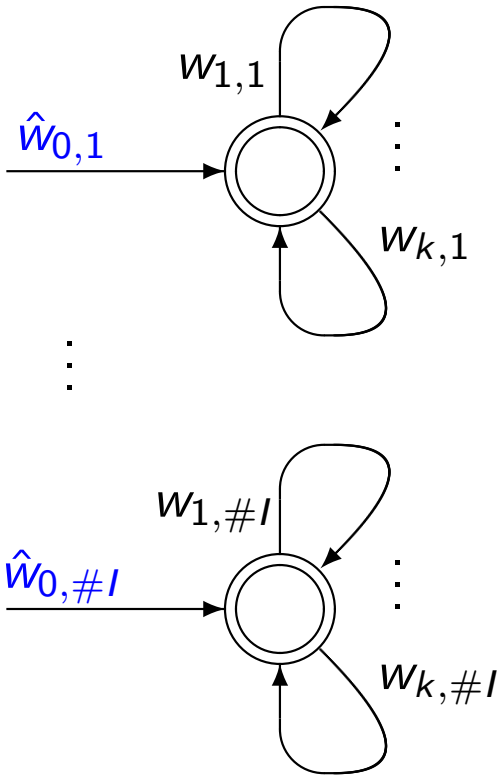
Outline: from linear to semilinear



- Standard construction for union of 1DFAs:
  number of states = *product*
  $\#I \leq p(n) \Rightarrow$ Too large!!!
- Strings $w_{0,i}$ can be replaced by Parikh equivalent strings $\hat{w}_{0,i}$ in such a way that $W_0 = \{\hat{w}_{0,i} \mid i \in I\}$ is a *prefix code*
- After this change:
  number of states $\leq$ *sum* Polynomial!!!

Converting 1NFAs accepting only nonunary strings

Outline: from linear to semilinear



- Standard construction for union of 1DFAs: number of states = *product*
  $\#I \leq p(n) \Rightarrow$ Too large!!!
- Strings $w_{0,i}$ can be replaced by Parikh equivalent strings $\hat{w}_{0,i}$ in such a way that $W_0 = \{\hat{w}_{0,i} \mid i \in I\}$ is a *prefix code*
- After this change:
  number of states $\leq$ *sum* Polynomial!!!

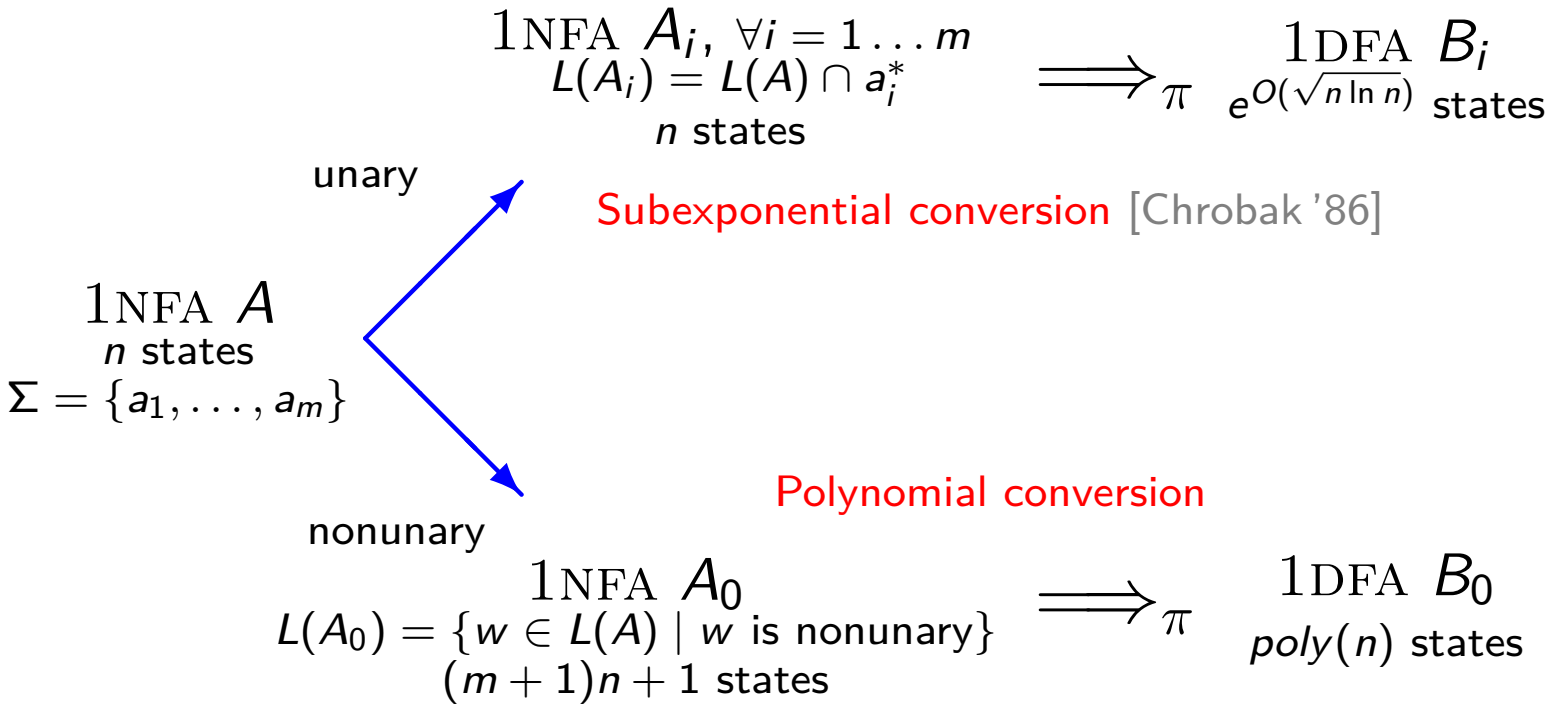
Theorem

For each n-state 1NFA *accepting a language none of whose words are unary, there exists a Parikh equivalent* 1DFA *with a number of states polynomial in n.*

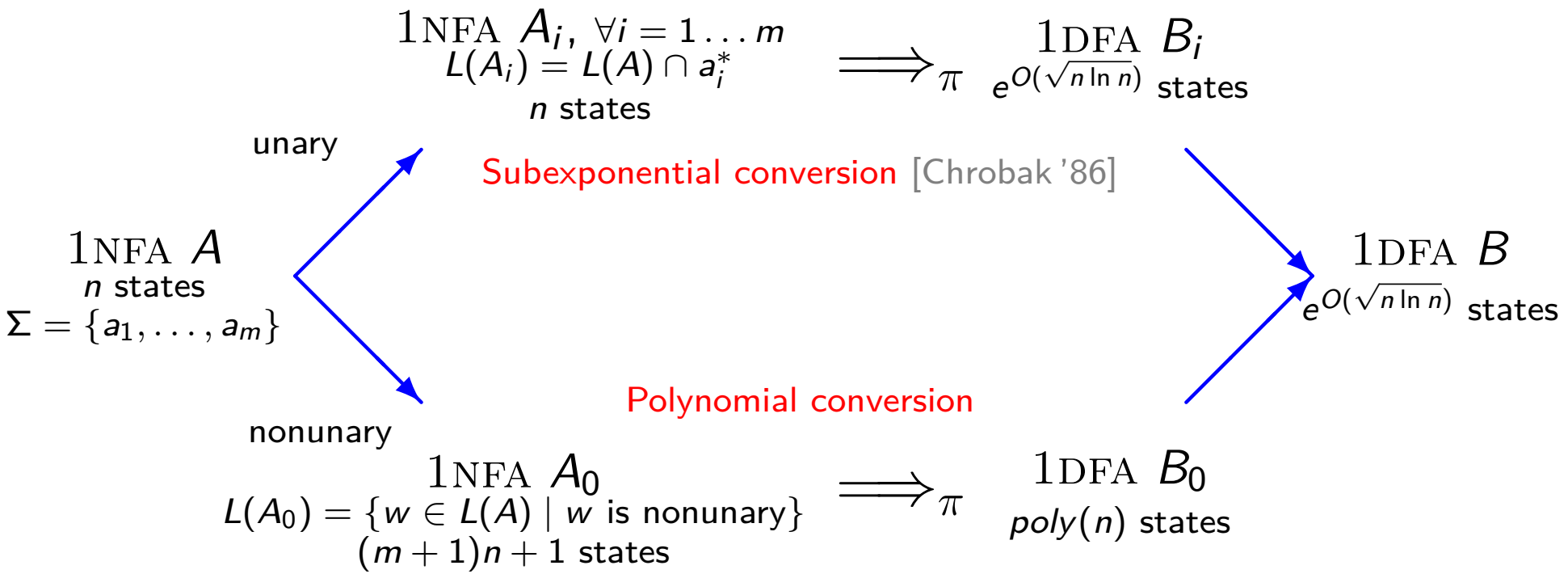
From 1NFAs to Parikh equivalent 1DFAs: general case

1NFA A
n states
$\Sigma = \{a_1, \ldots, a_m\}$

From 1NFAs to Parikh equivalent 1DFAs: general case



From 1NFAs to Parikh equivalent 1DFAs: general case



Theorem

For each n-state 1NFA *over* Σ, *there exists a Parikh equivalent* 1DFA *with* $e^{O(\sqrt{n \ln n})}$ *states. Furthermore, this cost is optimal.*

From CFGs to Parikh equivalent 1DFAs

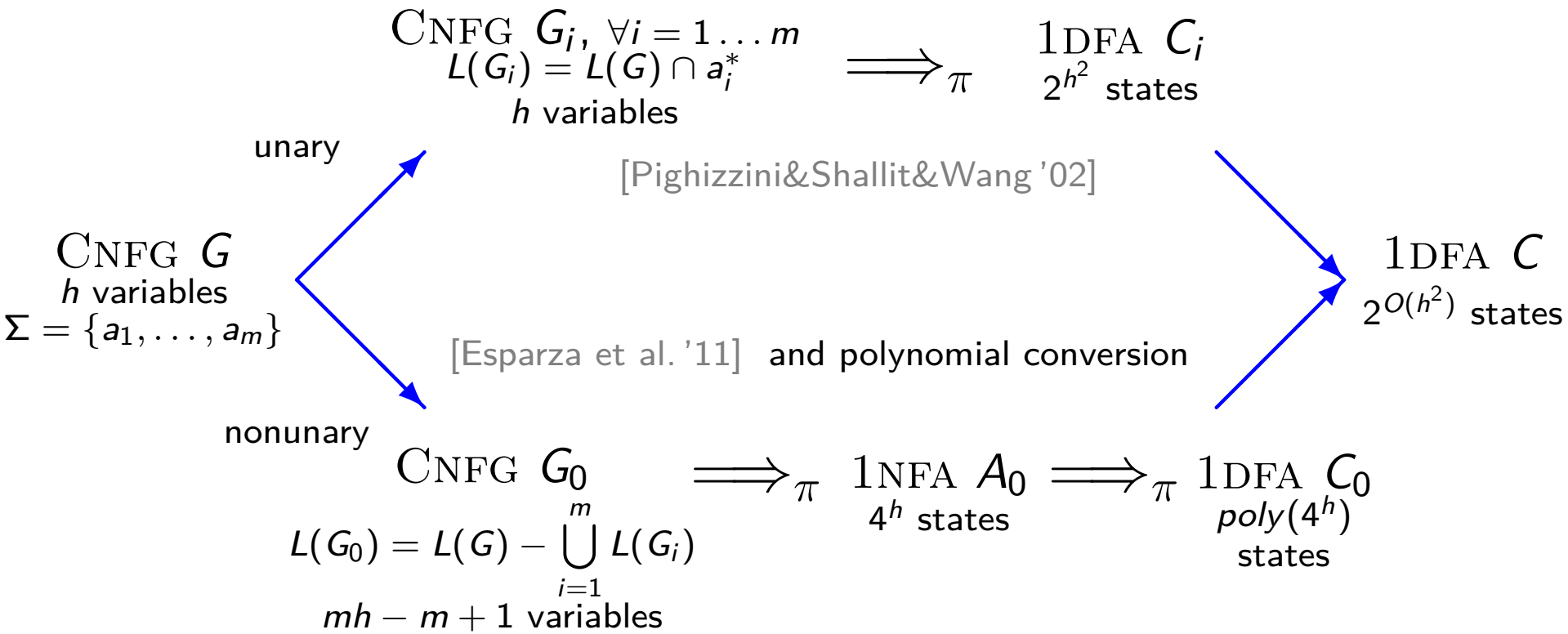
Our second contribution:

Problem (CFGs to 1DFAs)

$$\underset{\substack{\text{CFG } \textit{in Chomsky normal form}\\ h \textit{ variables}\\ L_1}}{\text{CNFG}} \quad \Longrightarrow_{\pi} \quad \underset{\substack{\textit{how many states?}\\ L_2}}{\text{1DFA}}$$

- Upper bound: $2^{O(4^h)}$
  by subset construction
  and [Esparza&Ganty&Kiefer&Luttenberger '11]
- Lower bound: 2^{ch^2}
  This bound derives from the *unary case*: the state cost of the conversion of unary h-variable CNFGs into equivalent 1DFAs is less than 2^{h^2} [Pighizzini&Shallit&Wang '02]

From CFGs to Parikh equivalent 1DFAs: general idea



Theorem

For each h-variable CNFG *over Σ, there exists a Parikh equivalent* 1DFA *with at most $2^{O(h^2)}$ states. Furthermore, this cost is optimal.*

Conversions into Parikh equivalent 2DFAs

Summary table

	1DFA	2DFA
1NFA n states	$e^{O(\sqrt{n \ln n})}$	
CNFG h variables	$2^{O(h^2)}$	

Conversions into Parikh equivalent 2DFAs

Summary table

Our third contribution

	1DFA	2DFA
1NFA n states	$e^{O(\sqrt{n \ln n})}$	$poly(n)$
CNFG h variables	$2^{O(h^2)}$	$2^{O(h)}$

The idea is to split the accepted language into its unary and nonunary parts:

- The state cost of the conversion of unary n-state 1NFAs into equivalent 2DFAs is $O(n^2)$ [Chrobak '86]
- The state cost of the conversion of unary h-variable CNFGs into equivalent 1NFAs is at most $2^{2h-1} + 1$ [Pighizzini&Shallit&Wang '02]

Operational state complexity under Parikh mapping

- Automata are used in many areas of computer science to model the behaviour of systems or protocols.
- The need to perform operations between them, brings to an explosion of the number of states.

Our Goal

We investigate, under Parikh equivalence, the state complexity of some language operations which preserve regularity $(\cup, \cap, ^c, \cdot, ^*, \sqcup, ^R, P_{\Sigma_0})$.

Problem (1DFAs to 1DFA)

$$\begin{array}{c} A,\ B\ \text{1DFAs} \\ n_1,\ n_2\ \text{states} \\ L(A), L(B) \end{array} \quad \Longrightarrow_{\pi} \quad \begin{array}{c} C\ \text{1DFA} \\ L(C) =_{\pi} L \\ \textit{how many states?} \end{array}$$

where:

- $L = L(A) \cup L(B)$
- $L = L(A) \cap L(B)$
- $L = L(A)L(B)$
- ...

Concatenation under Parikh equivalence

Problem setting

Problem (1DFAs to 1DFA)

A, B 1DFAs		C 1DFA
n_1, n_2 states	$\Longrightarrow_\pi$	$L(C) =_\pi L$
$L = L(A)L(B)$		*how many states?*

- Upper bound: $e^{\sqrt{n \cdot \ln n}}$, where $n = n_1 + n_2$
  by Parikh equivalent conversion
- Lower bound: $n_1 n_2$ states
  by unary case

[Yu '00]

Operational state complexity under Parikh equivalence

Summary table

Our fourth contribution:

Operation	Standard equivalence	Parikh equivalence
$L_1 \cup L_2$	$n_1 n_2$	$n_1 n_2$
$L_1 \cap L_2$	$n_1 n_2$	$n_1 n_2$
L_1^c	n_1	n_1
$L_1 L_2$	$(2n_1 - 1)2^{n_2 - 1}$	$poly(n_1, n_2)$
L_1^*	$2^{n_1 - 1} + 2^{n_1 - 2}$	$poly(n_1)$
$L_1 \sqcup\!\sqcup L_2$	$2^{n_1 n_2} - 1$	$poly(n_1, n_2)$
L_1^R	2^{n_1}	n_1
$P_{\Sigma_0}(L_1)$	$3 \cdot 2^{n_1 - 2} - 1$	$e^{O(\sqrt{n_1 \cdot \ln n_1})}$

[Yu '00, Campeanu&Salomaa&Yu '02, Yu&Zhuang&Salomaa '94, Jiraskova&Masopust '12]

Conclusion and future works

- Quite surprisingly, the costs into Parikh equivalent 1DFA is due to the unary parts of the languages.
- Conversion of the parts consisting of nonunary strings is less expensive.
- Conversions into Parikh equivalent 2DFA are less expensive than the corresponding ones into 1DFA.
- State complexity of some operations between regular languages has been studied under Parikh equivalence.

Conclusion and future works

Complement does not commute with Parikh mapping

A 1DFA
n states $\Longrightarrow$ M 1DFA
$\psi(L(M)) = (\psi(L(A)))^c$
How many states needed?

- Given a deterministic automaton, finding the smallest Parikh equivalent deterministic automaton is an open question.
- It could be interesting to study conversions from 2DFA (or 2NFA) to Parikh equivalent 1DFA.

Publications (journals and conferences)

1. Parikh's Theorem and Descriptional Complexity.
   G.J. Lavado and G. Pighizzini.
   In 38th International Conference on Current Trends in Theory and Practice of Computer Science *SOFSEM 2012*,
   LNCS 7147, pages 361-372, 2012.
2. Converting Nondeterministic Automata and Context-free Grammars into Parikh Equivalent Deterministic Automata.
   G.J. Lavado, G. Pighizzini, and S. Seki.
   In Proceedings of Development in Language Theory *DLT 2012*,
   LNCS 7410, pages 284-295, 2012.
3. Converting Nondeterministic Automata and Context-Free Grammars into Parikh Equivalent One-Way and Two-Way Deterministic Automata.
   G.J. Lavado, G. Pighizzini, and S. Seki.
   In *Information and Computation*, volumes 228-229, pages 1-15, July, 2013.
4. Operational State Complexity under Parikh Equivalence.
   G.J. Lavado, G. Pighizzini, and S. Seki.
   Proceedings LNCS Vol. 8614. *DCFS 2014* - 16th International Workshop on Descriptional Complexity of Formal Systems. Finland, 5-8 August, 2014.

Presentations (conferences, workshops and seminars)

- Converting Nondeterministic Automata and Context-free Grammars into Parikh Equivalent Deterministic Automata.
  ICTCS 2012 - 13th Italian Conference on Theoretical Computer Science, Università degli Studi d'Insubria, 19-21 September, 2012
- Descriptional Complexity and Parikh Equivalence.
  Workshop Progetto MIUR *PRIN 2010-2011*,
  Automi e Linguaggi Formali: Aspetti Matematici e Applicativi,
  Politecnico di Milano, 28-29 November, 2013.
- Operational State Complexity under Parikh Equivalence.
  ICTCS 2014 - 15th Italian Conference on Theoretical Computer Science, Università degli Studi di Perugia, 17-19 September, 2014.
- Descriptional Complexity under Parikh Equivalence.
  Università degli Studi di Milano, 22 January, 2015.