Converting Nondeterministic Automata and Context-Free Grammars into Parikh Equivalent Deterministic Automata

Giovanna J. Lavado[1] Giovanni Pighizzini[1] Shinnosuke Seki[2]

[1]Dipartimento di Informatica, Università degli Studi di Milano, Italy

[2]Department of Information and Computer Science, Aalto University, Finland

ICTCS 2012
Villa Toeplitz, Varese, Italy
September 19-21, 2012

UNIVERSITÀ DEGLI STUDI
DI MILANO

Standard equivalence: NFAs vs DFAs

Subset construction [Rabin&Scott 1959]

NFA
n states
L

$\Longrightarrow$

DFA
2^n states
L

Moreover, this state bound cannot be reduced
[Meyer&Fischer 1971, Moore 1971]

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: NFAs vs DFAs

Subset construction [Rabin&Scott 1959]

NFA
n states
L

$\Longrightarrow$

DFA
2^n states
L

Moreover, this state bound cannot be reduced
[Meyer&Fischer 1971, Moore 1971]

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 equivalence

- $\Sigma = \{a_1, \ldots, a_m\}$ alphabet of m symbols
- *Parikh's map* $\psi : \Sigma^* \to \mathbb{N}^m$

$$\forall w \in \Sigma^*, \psi(w) = (|w|_{a_1}, |w|_{a_2}, \ldots, |w|_{a_m})$$

  where $|w|_{a_i}$ is the number of occurrences of a_i in the word w
- $w, v \in \Sigma^*$ are *Parikh equivalent* (in symbols $w =_\pi v$) iff $\psi(w) = \psi(v)$
- *Parikh image* of a language $L \subseteq \Sigma^*$:

$$\psi(L) = \{\psi(w) \mid w \in L\}$$

- $L_1, L_2 \subseteq \Sigma^*$ are *Parikh equivalent* (in symbols $L_1 =_\pi L_2$) iff $\psi(L_1) = \psi(L_2)$

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:

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

have the same Parikh image, namely the set

$$\psi(L) = \psi(R) = \{(n, n) \mid n \geq 0\}$$

Our Goal

We investigate the conversion of nondeterministic automata and context-free grammars into *Parikh equivalent* deterministic automata, from a *descriptional complexity* point of view

Problem (NFAs to DFAs)

NFA
n states
$\Longrightarrow_{\pi}$
DFA
how many states?

Problem (CFGs to DFAs)

CFG
Chomsky normal form
h variables
$\Longrightarrow_{\pi}$
DFA
how many states?

Parikh Theorem: descriptional complexity aspects

We consider:

- CFGs in Chomsky normal form
- As a measure of size the number of variables, denoted by h [Gruska 1973]

Results:

- Classical proof: NFA $2^{2^{O(h^2)}}$ states
- Esparza et al. proof: NFA $O(4^h)$ states

Theorem ([Esparza&Ganty&Kiefer&Luttenberger 2011])

For any context-free grammar in Chomsky normal form with h variables, there exists a Parikh equivalent NFA with $O(4^h)$ states.

Parikh equivalence: unary languages

$$\forall L_1, L_2 \subseteq \{a\}^*, \qquad L_1 =_\pi L_2 \quad \textit{iff} \quad L_1 = L_2$$

Theorem ([Ginsburg&Rice 1962])

Each unary context-free language is regular.

CFG
Chomsky normal form
h variables

$\Longrightarrow$

DFA
$2^{\Theta(h^2)}$ states

[Pighizzini&Shallit&Wang 2002]

NFA
n states

$\Longrightarrow$

DFA
$e^{\Theta(\sqrt{n \ln n})}$ states

[Chrobak 1986]

From NFAs to Parikh equivalent DFAs

Our first contribution:

Problem (NFAs to DFAs)

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

- Upper bound: 2^n
  by subset construction [Rabin&Scott 1959]
- Lower bound: $e^{\sqrt{n \ln n}}$
  by unary case [Chrobak 1986]

Converting NFAs accepting only nonunary strings

A preliminary step:

Problem (NFAs to DFAs, restricted)

NFA s.t. each accepted string is nonunary
n states

$$\Longrightarrow_{\pi}$$

DFA
how many states?

Quite surprisingly, we can obtain a DFA with a number of states *polynomial* in n,

i.e., this conversion is less expensive than the conversion in the unary case, which costs $e^{\Theta(\sqrt{n \ln n})}$

Converting NFAs accepting only nonunary strings

The conversion uses a modification of the following result:

Theorem ([Kopczyński&To 2010])

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

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

where:

- *I is a set of at most $p(n)$ indexes*
- *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 $\{0, 1, \ldots, n\}^m$*

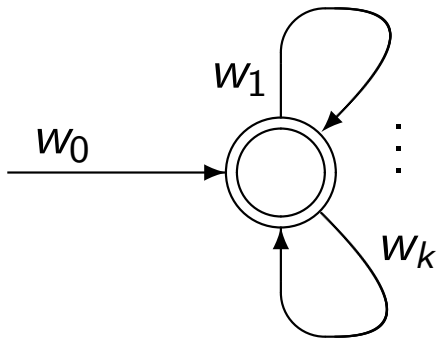
Each above linear set

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

can be converted into a poly size DFA 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*



Each above linear set

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

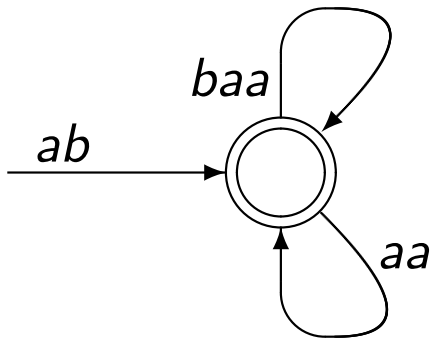
can be converted into a poly size DFA accepting a language

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

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

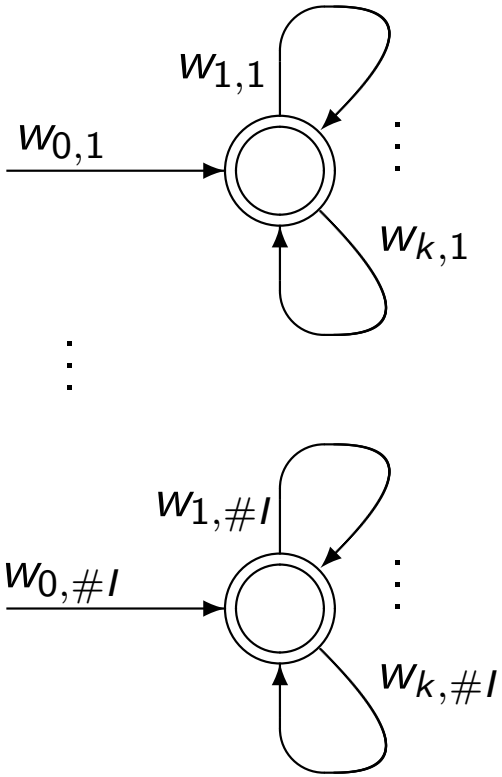
Example:

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



Converting NFAs accepting only nonunary strings

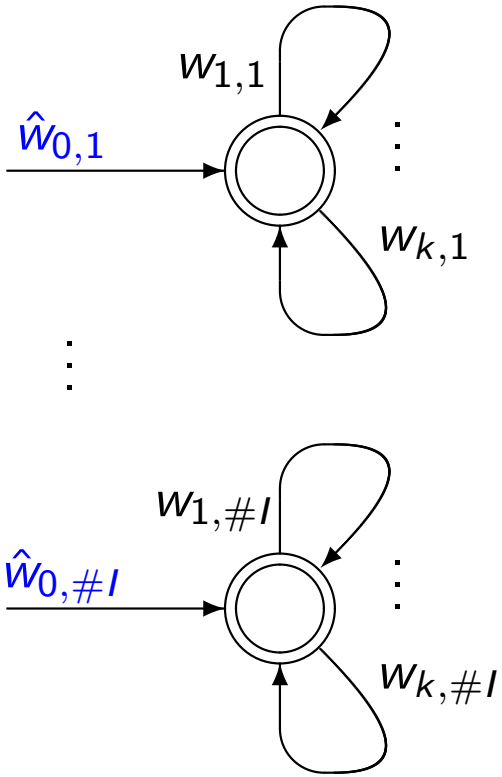
Outline: from linear to semilinear



- Standard construction for union of DFAs:
  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 NFAs accepting only nonunary strings

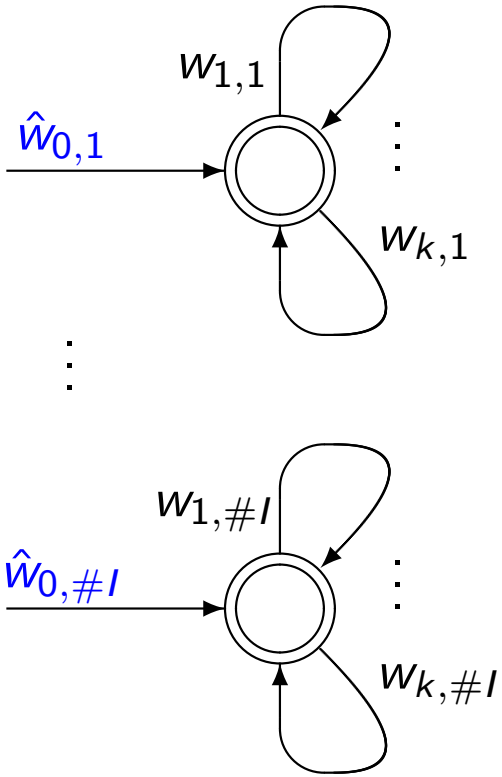
Outline: from linear to semilinear



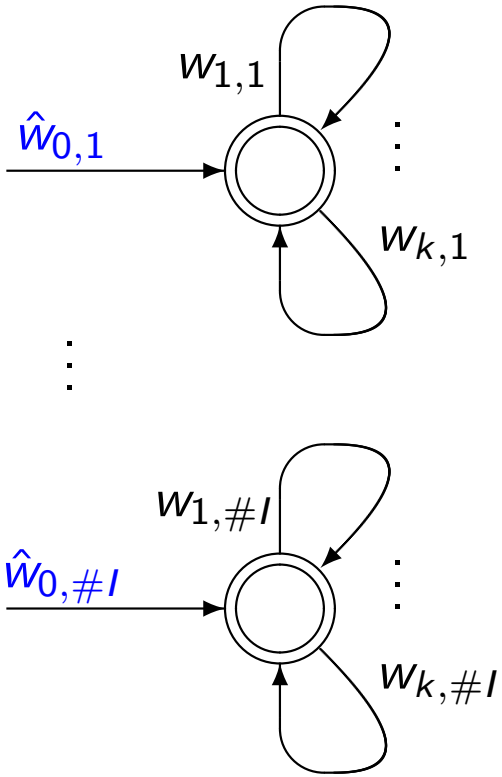
- Standard construction for union of DFAs:
  number of states = *product*
  $\#I \leq p(n) \Rightarrow$ Too large!!!
- Strings $w_{0,i}$ can be replaced by Parikh equivalent strings $\hat{w}_{0,i}$ in such a way that $W_0 = \{\hat{w}_{0,i} \mid i \in I\}$ is a *prefix code*
- After this change:
  number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

Converting NFAs accepting only nonunary strings

Outline: from linear to semilinear



- Standard construction for union of DFAs:
  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!!!



- Standard construction for union of DFAs:
  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 NFA with n states accepting a language none of whose words are unary, there exists a Parikh equivalent DFA with a number of states polynomial in n. Furthermore, this cost is tight.

From NFAs to Parikh equivalent DFAs: general case

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

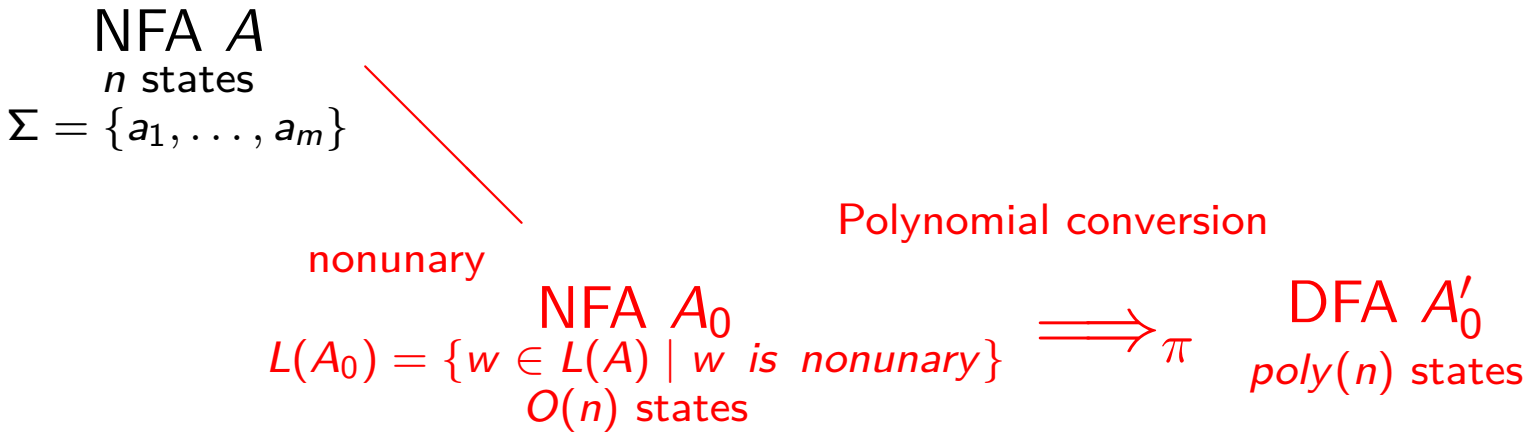
From NFAs to Parikh equivalent DFAs: general case

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

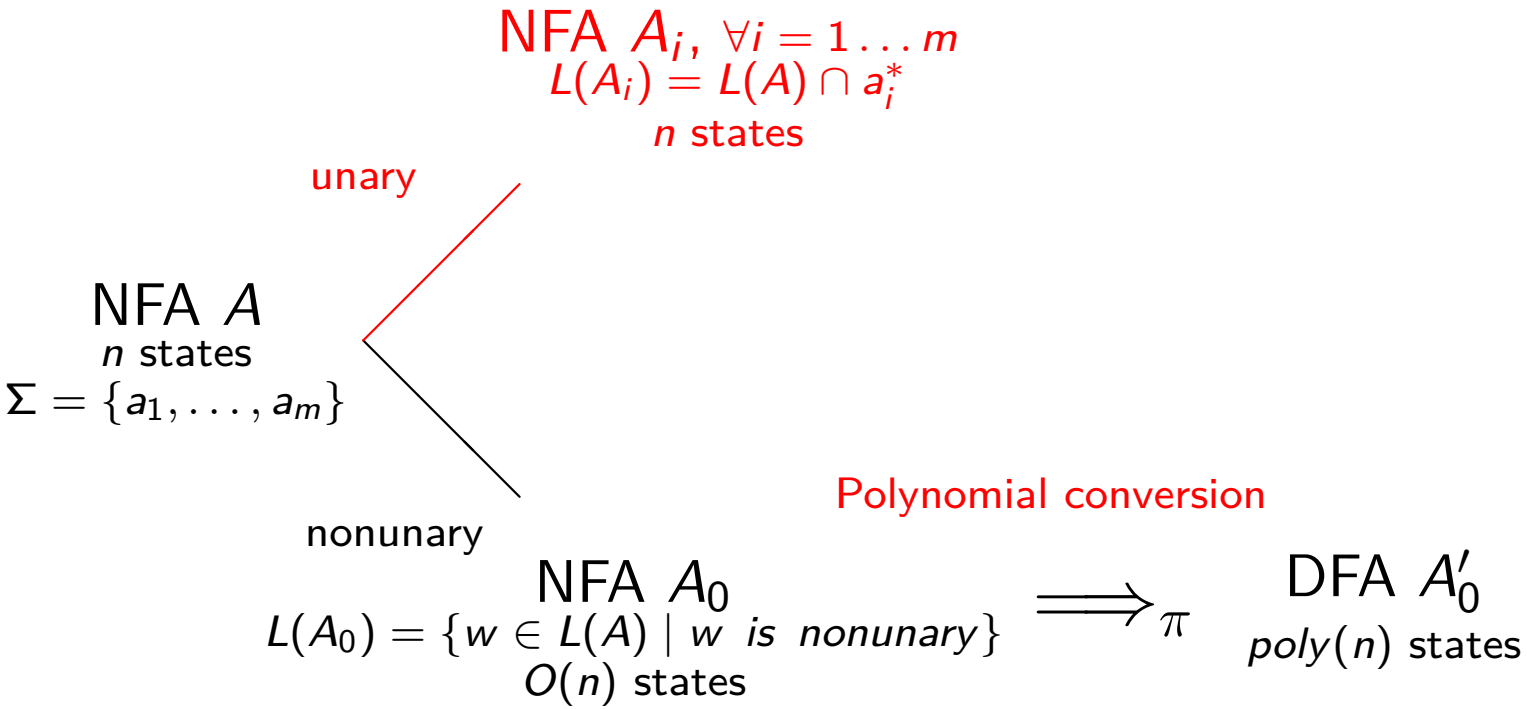
nonunary

NFA A_0
$L(A_0) = \{w \in L(A) \mid w \text{ is nonunary}\}$
$O(n)$ states

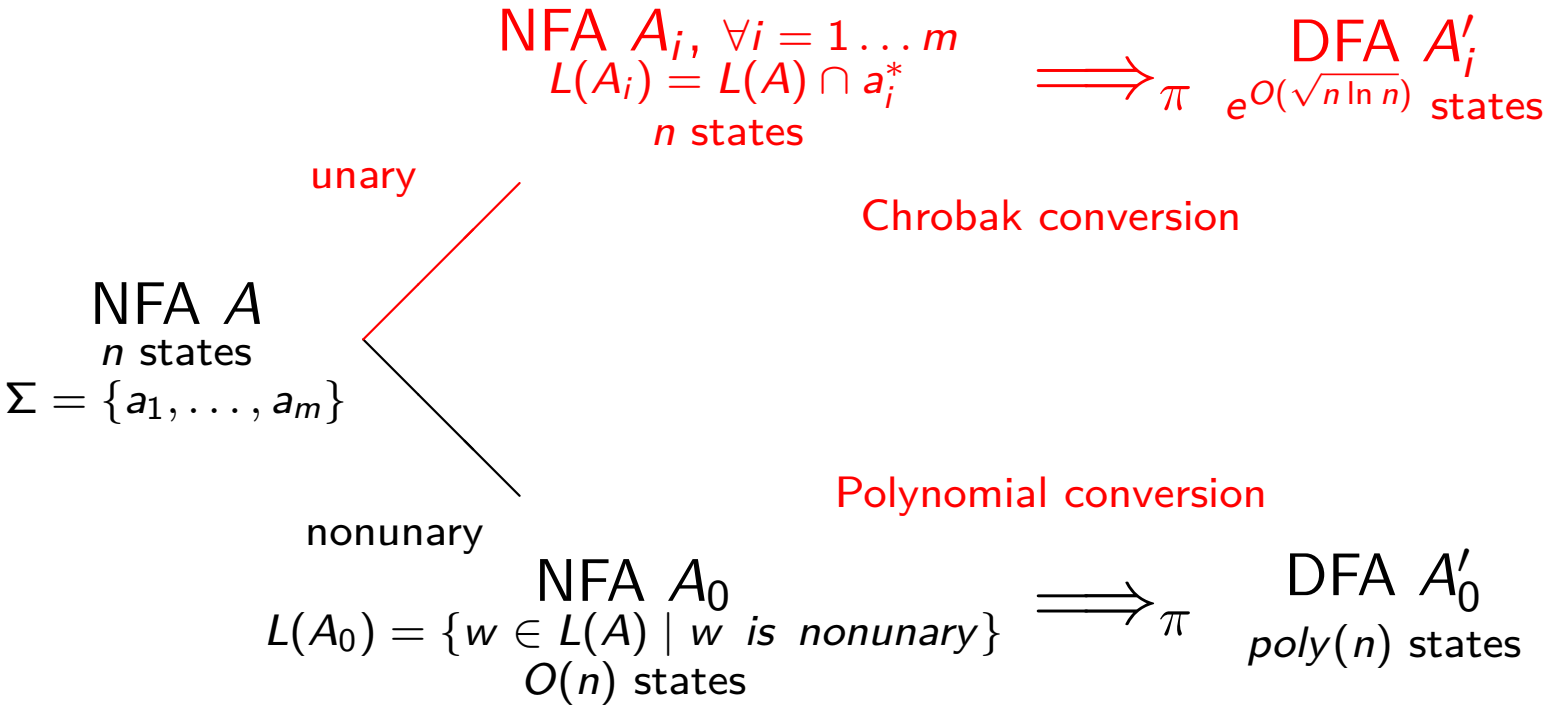
From NFAs to Parikh equivalent DFAs: general case



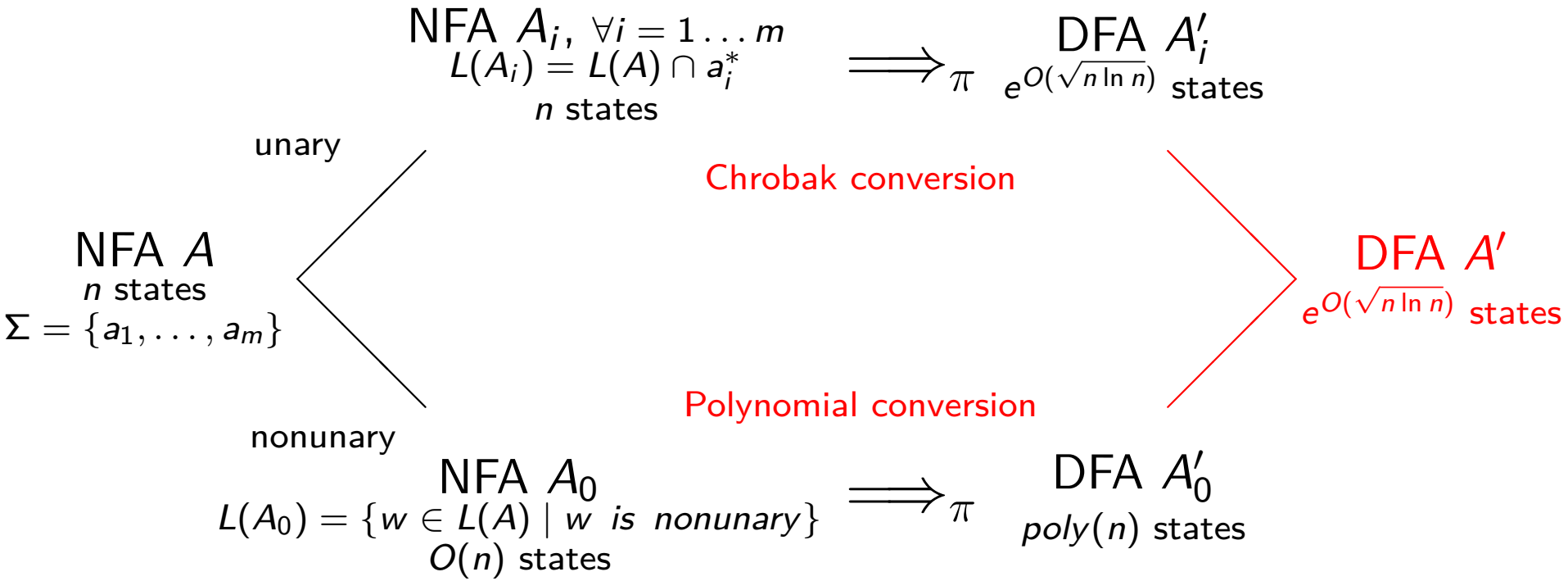
From NFAs to Parikh equivalent DFAs: general case



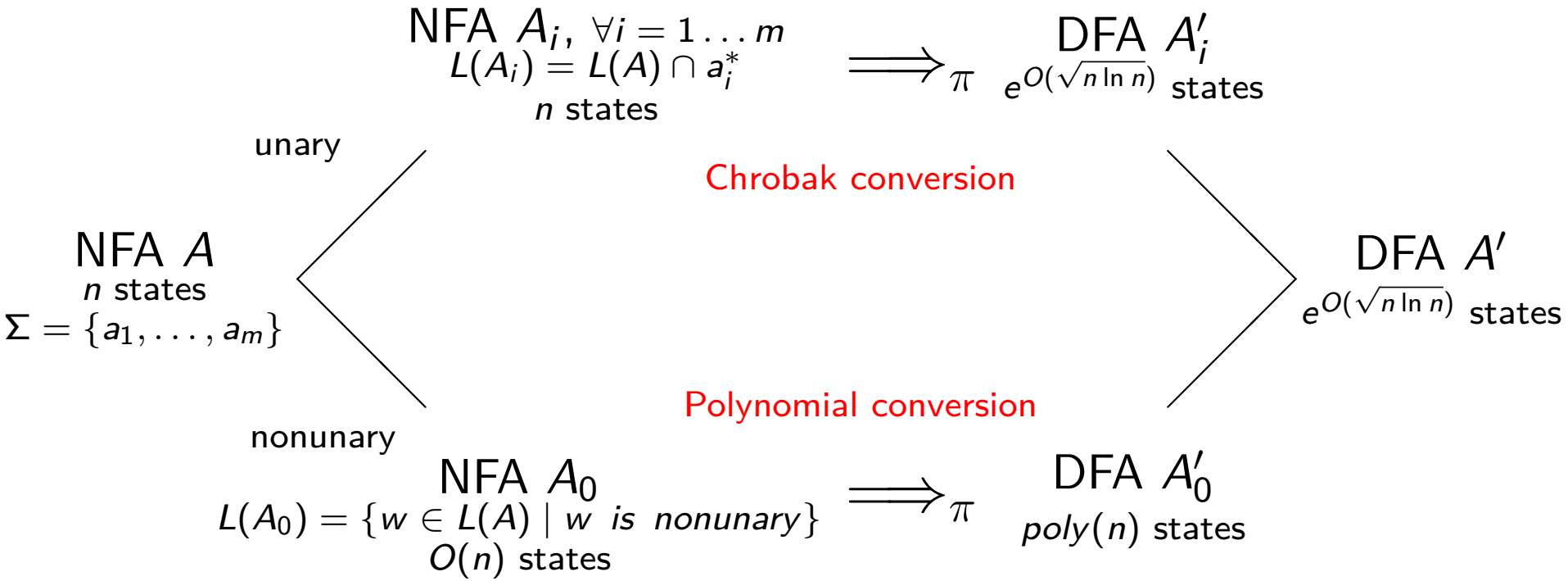
From NFAs to Parikh equivalent DFAs: general case



From NFAs to Parikh equivalent DFAs: general case



From NFAs to Parikh equivalent DFAs: general case



Theorem

For each NFA with n states, there exists a Parikh equivalent DFA with $e^{O(\sqrt{n \ln n})}$ states. Furthermore, this cost is tight.

From CFGs to Parikh equivalent DFAs

Our second contribution:

Problem (CFGs to DFAs)

CFG
Chomsky normal form
h variables

$\Longrightarrow_{\pi}$

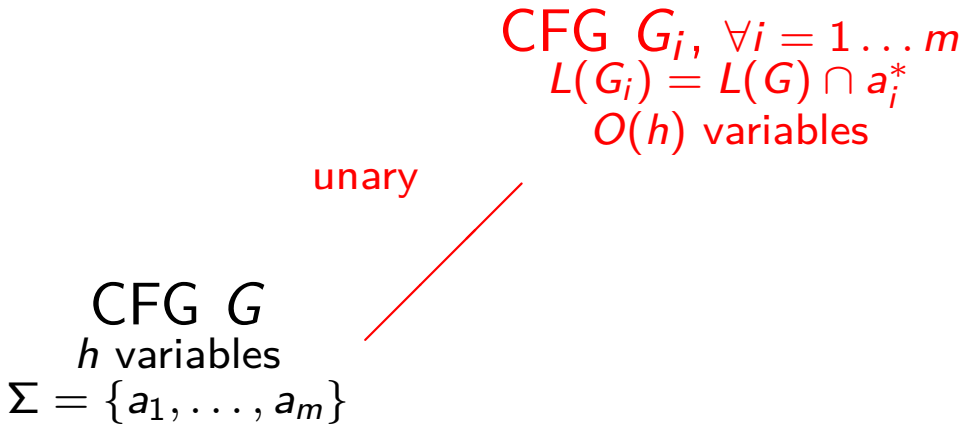
DFA
how many states?

- Upper bound: $2^{O(4^h)}$
  by subset construction
  and [Esparza&Ganty&Kiefer&Luttenberger 2011]
- Lower bound: 2^{ch^2}
  tight bound for the unary case $2^{\Theta(h^2)}$
  [Pighizzini&Shallit&Wang 2002]

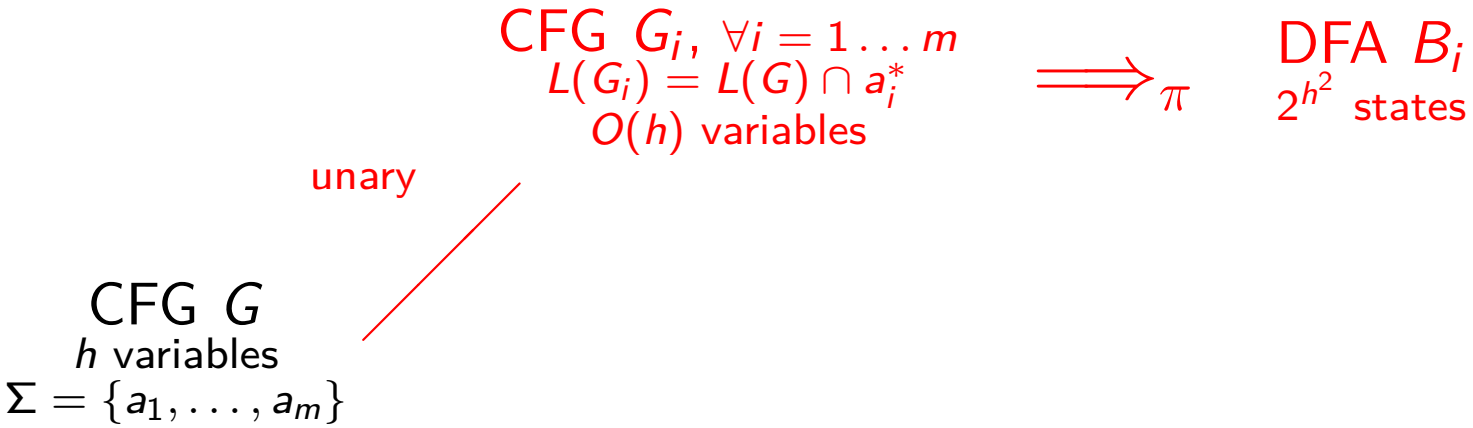
From CFGs to Parikh equivalent DFAs

CFG G
h variables
$\Sigma = \{a_1, \ldots, a_m\}$

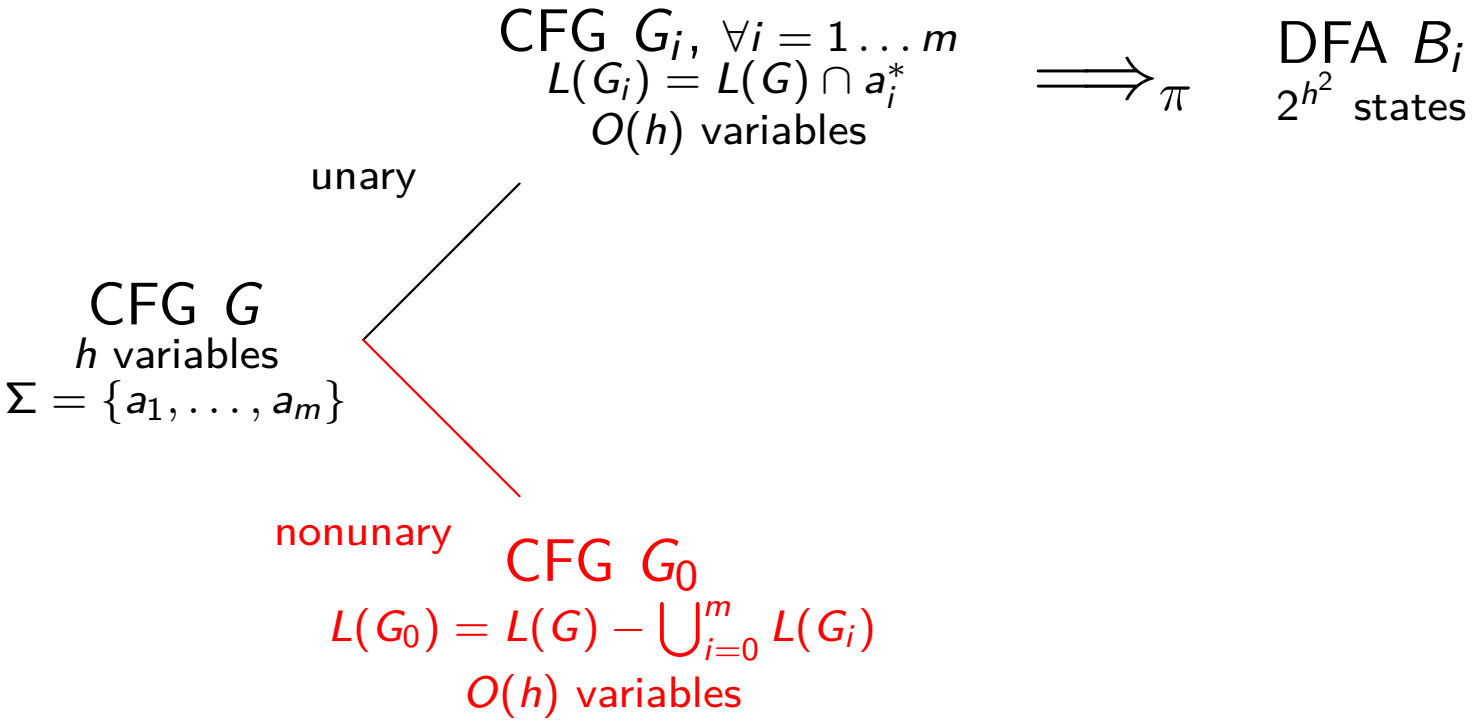
From CFGs to Parikh equivalent DFAs



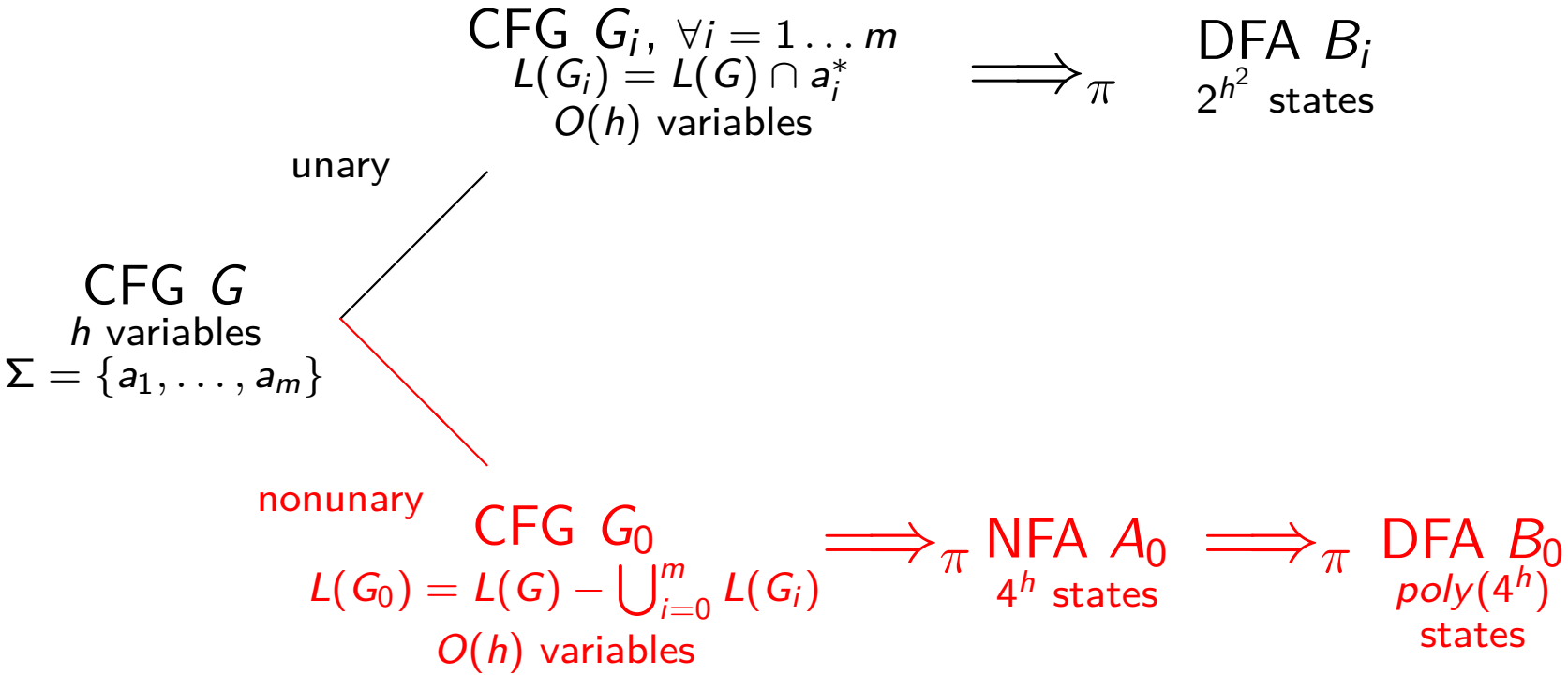
From CFGs to Parikh equivalent DFAs



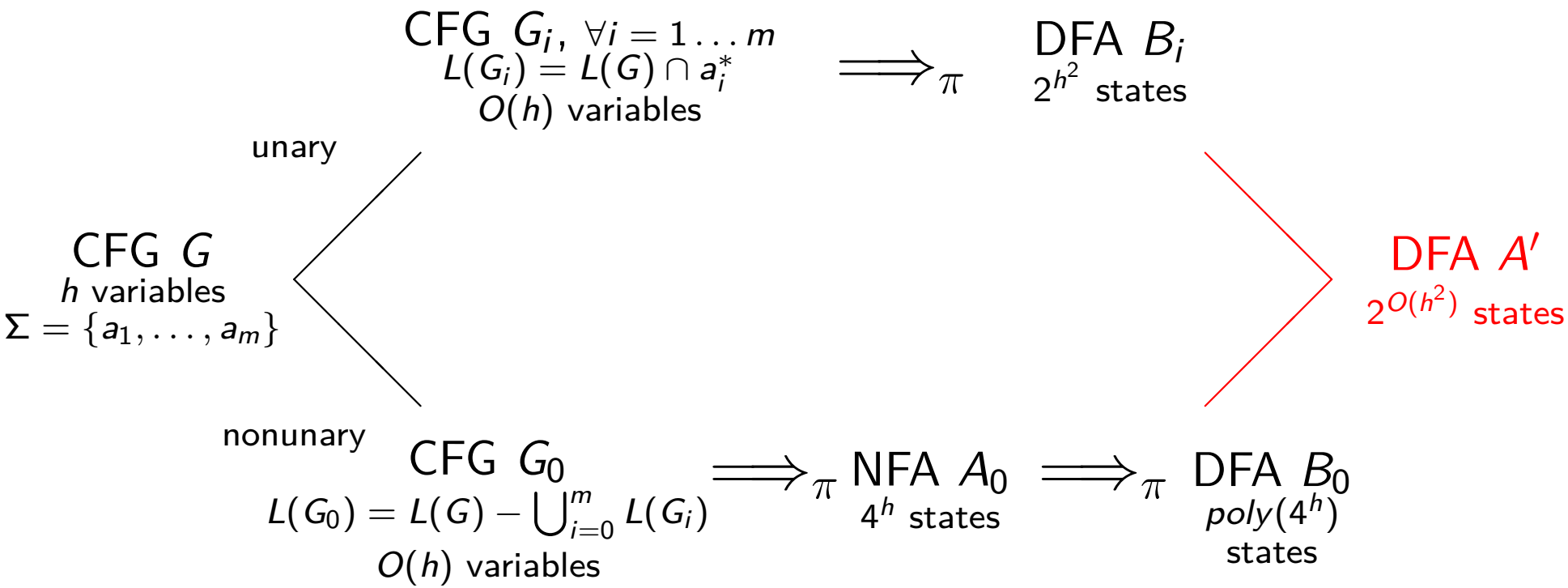
From CFGs to Parikh equivalent DFAs



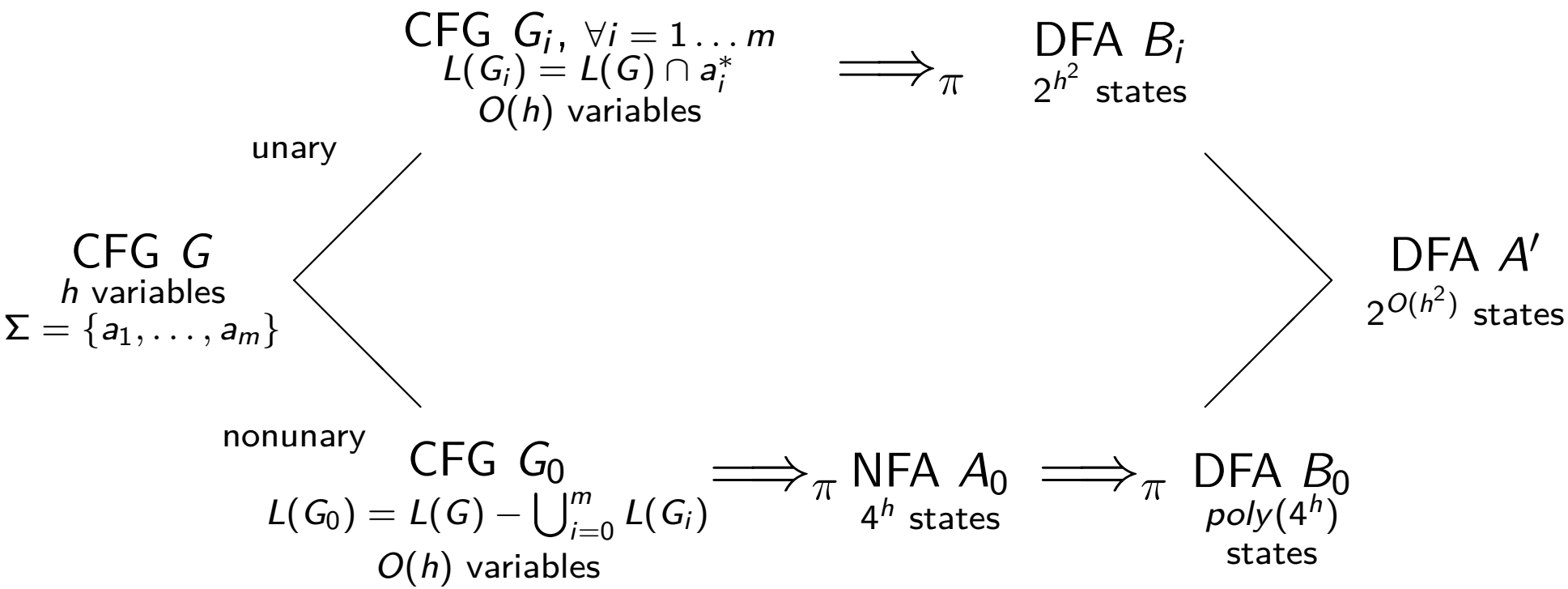
From CFGs to Parikh equivalent DFAs



From CFGs to Parikh equivalent DFAs



From CFGs to Parikh equivalent DFAs



Theorem

For any CFG in Chomsky normal form with h variables, there exists a Parikh equivalent DFA with at most $2^{O(h^2)}$ states. Furthermore, this cost is tight.

Conclusion and further work

- Quite surprisingly, in both cases the cost is due to the unary parts of the languages.
- The conversion of the parts consisting of nonunary strings is less expensive.
- It could be interesting to investigate if for some other constructions related to regular and context-free languages similar phenomena happen
  (e.g., DFA minimization, state complexity of operations).

Thank you for your attention