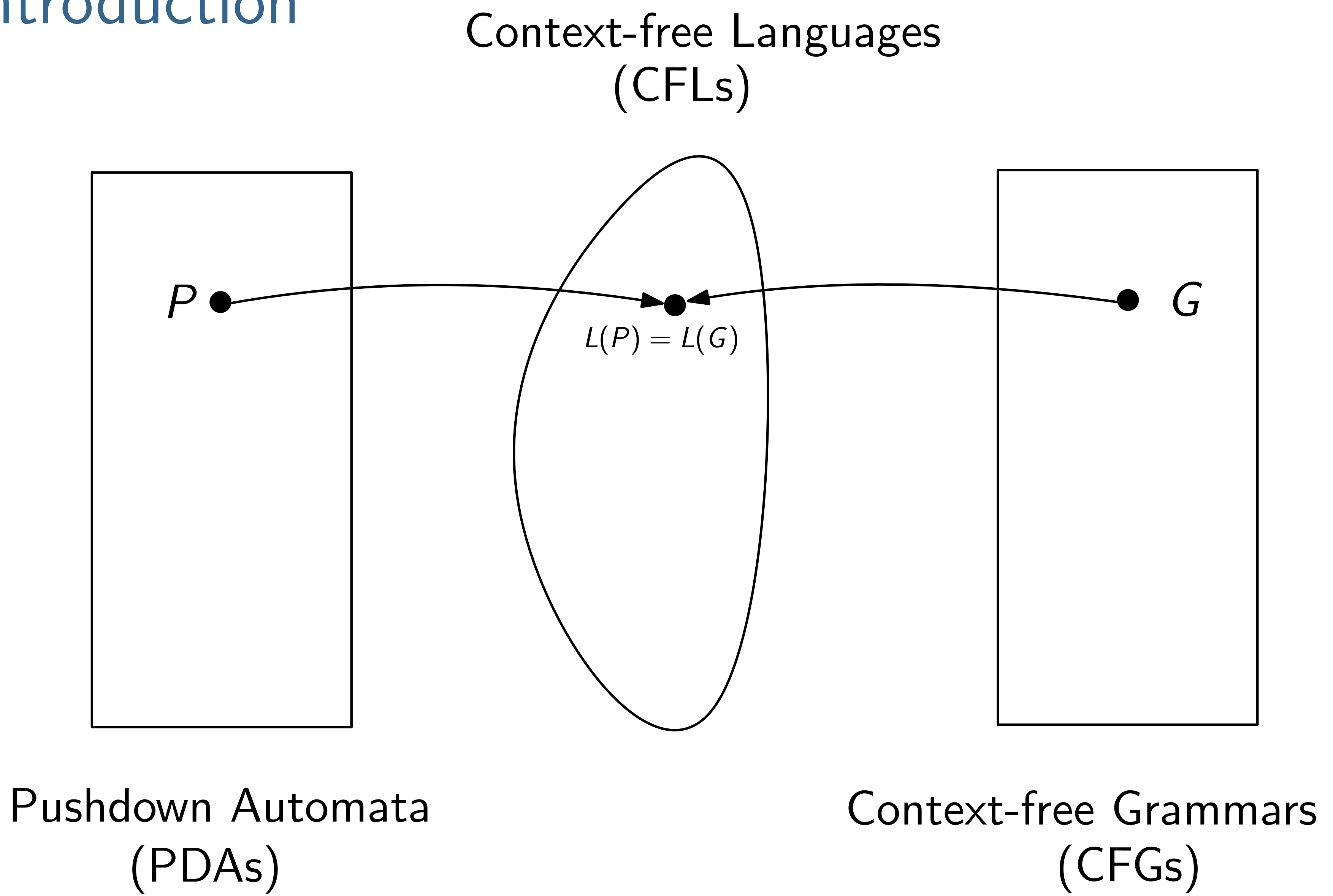
Parikh Image of Pushdown Automata

Elena Gutiérrez and Pierre Ganty

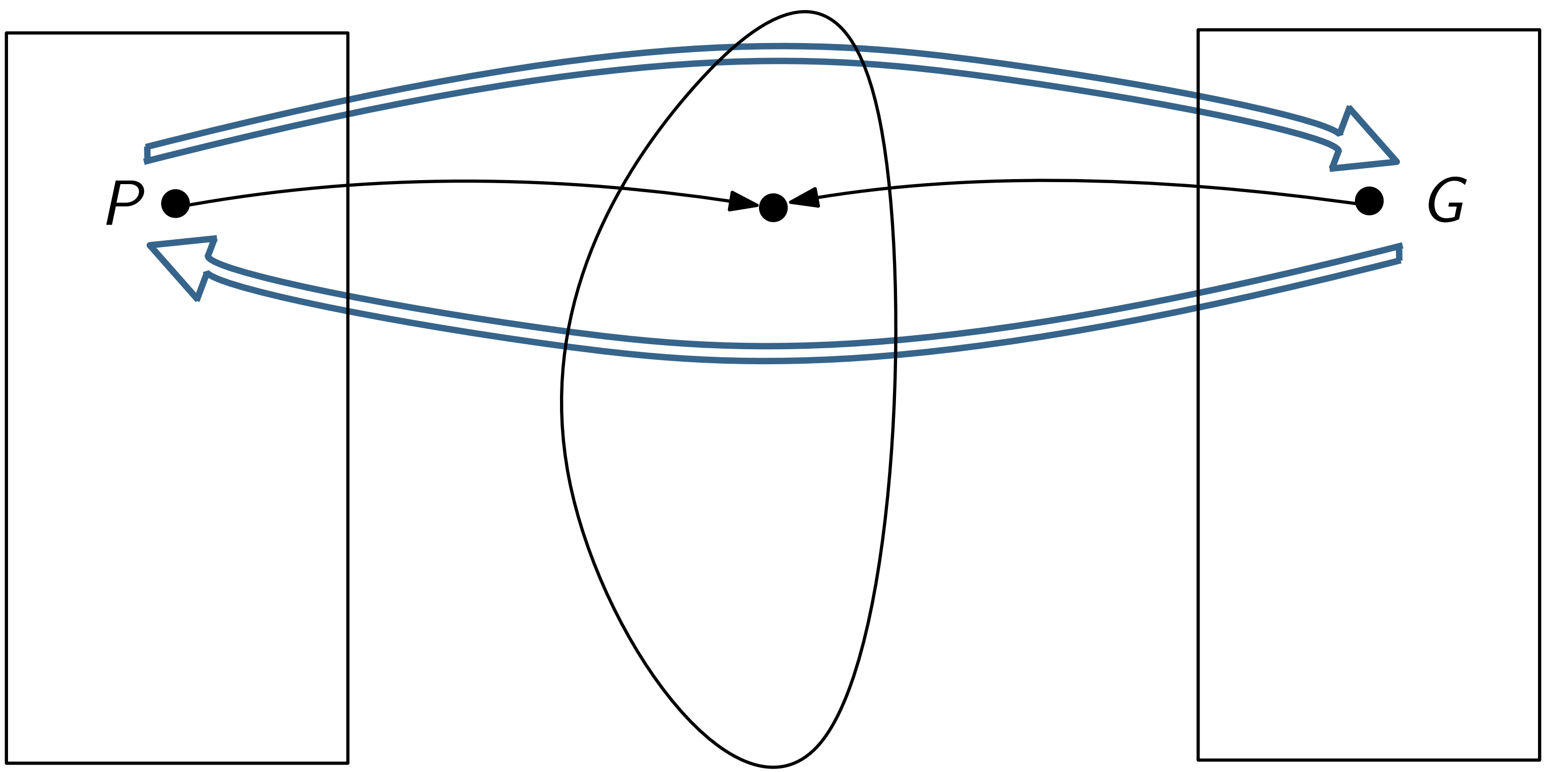
institute IMDEA software

Introduction



Introduction

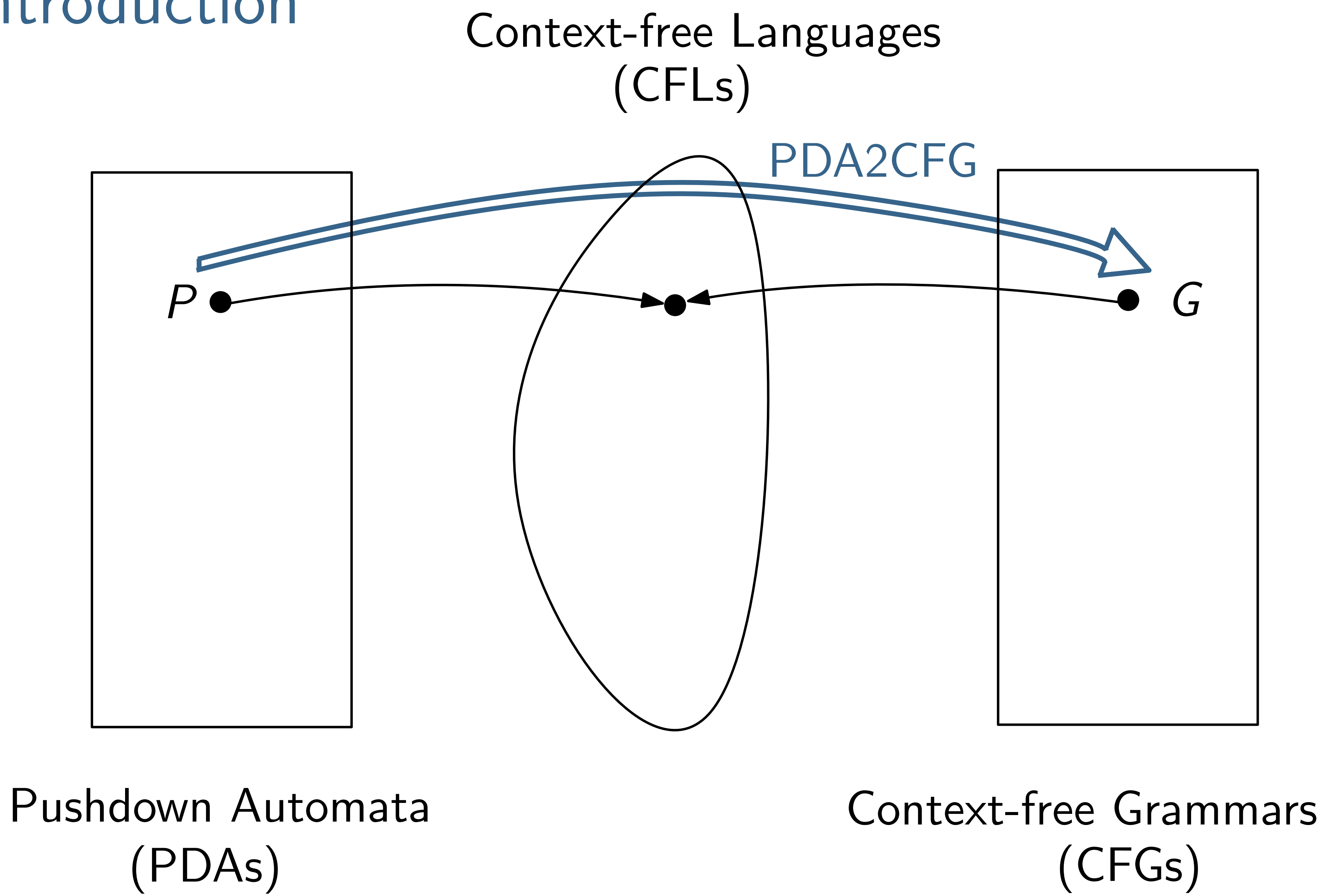
Context-free Languages
(CFLs)



Pushdown Automata
(PDAs)

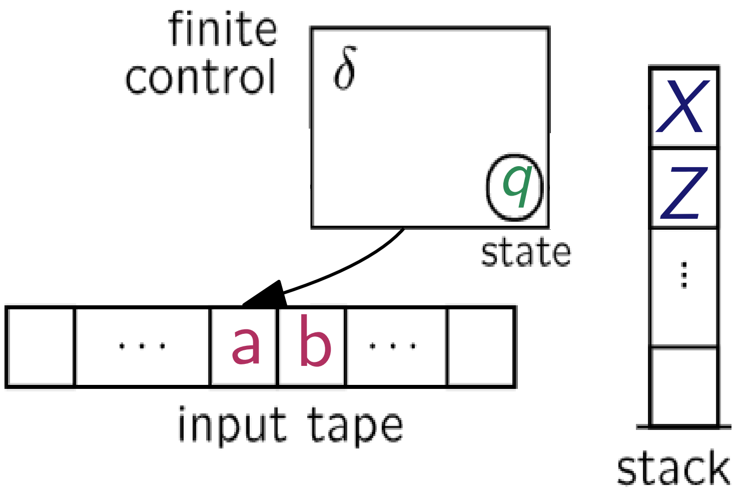
Context-free Grammars
(CFGs)

Introduction



PDAs and CFGs

- **Pushdown Automata**

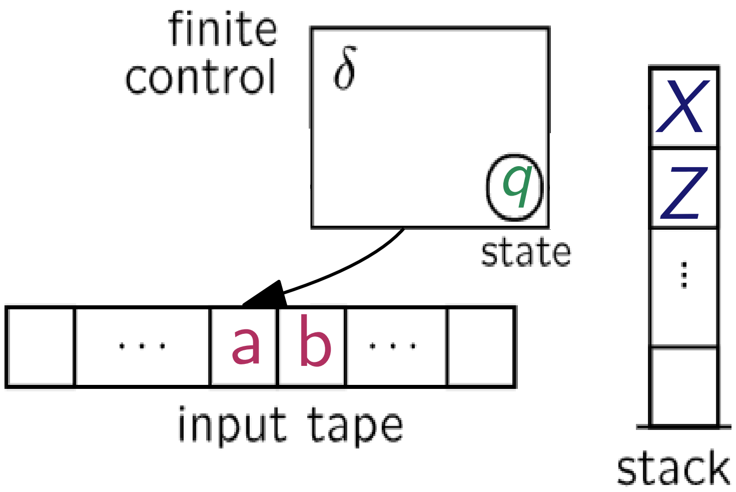


- **Context-free Grammar**

$$S \Rightarrow aSa \Rightarrow abSba \Rightarrow \ldots \Rightarrow abaaba$$

PDAs and CFGs

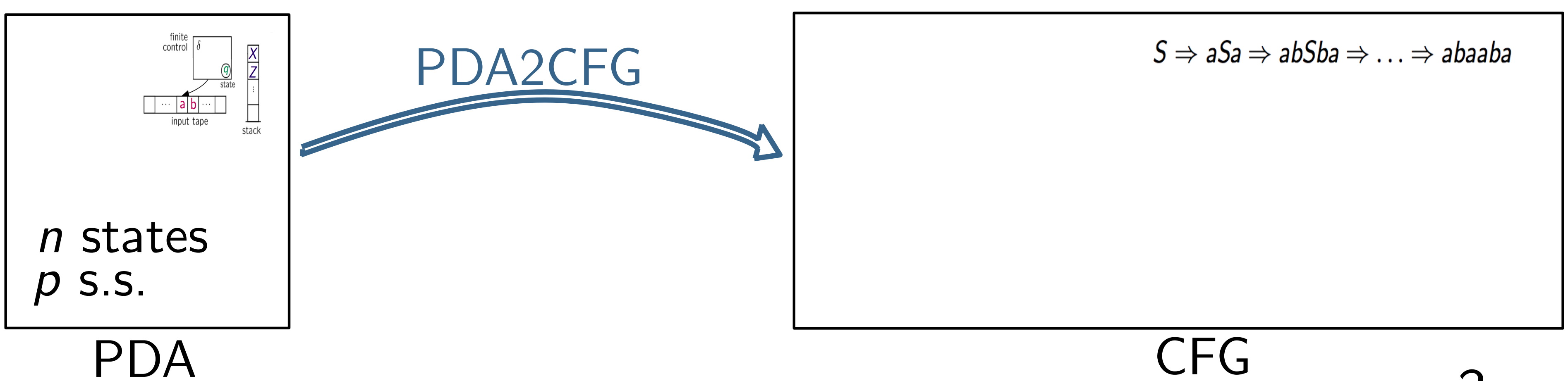
- **Pushdown Automata**



- **Context-free Grammar**

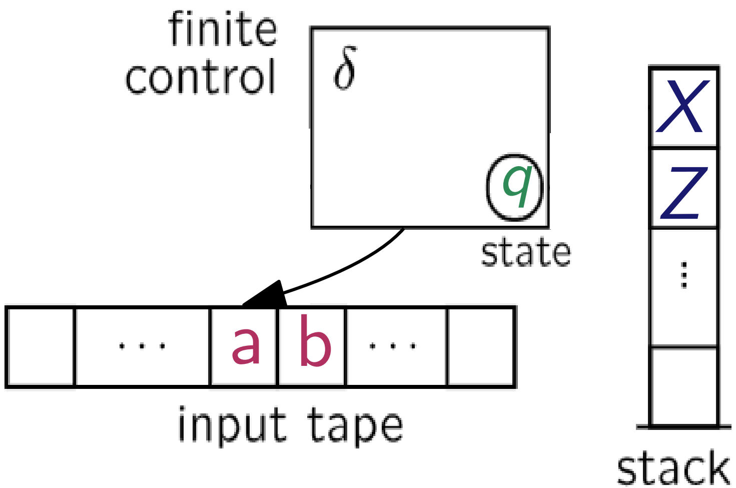
$$S \Rightarrow aSa \Rightarrow abSba \Rightarrow \ldots \Rightarrow abaaba$$

- **PDA2CFG**



PDAs and CFGs

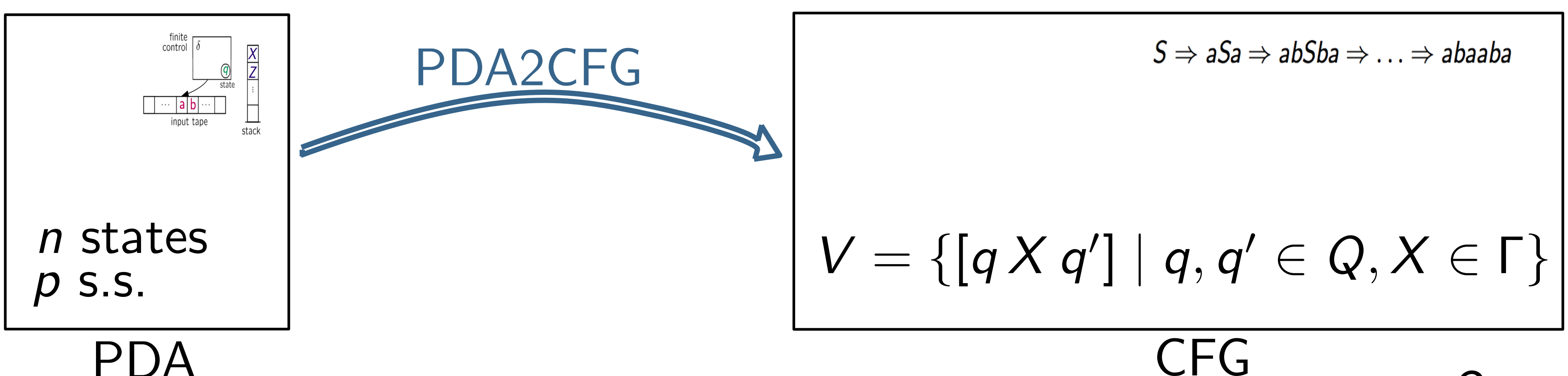
- **Pushdown Automata**



- **Context-free Grammar**

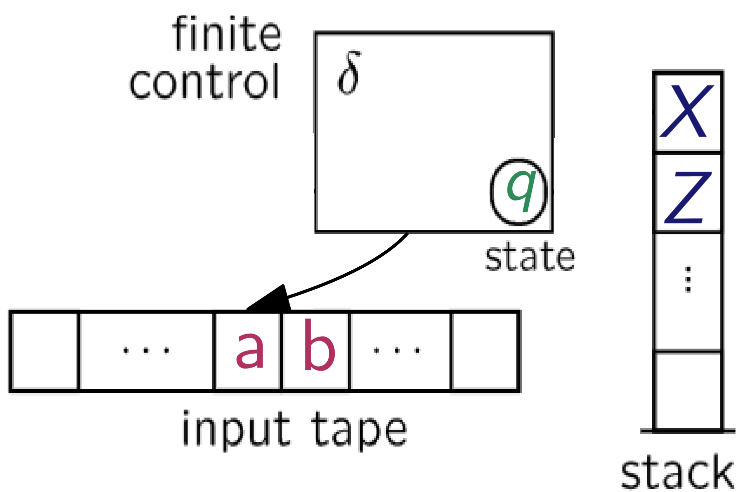
$$S \Rightarrow aSa \Rightarrow abSba \Rightarrow \ldots \Rightarrow abaaba$$

- **PDA2CFG**



PDAs and CFGs

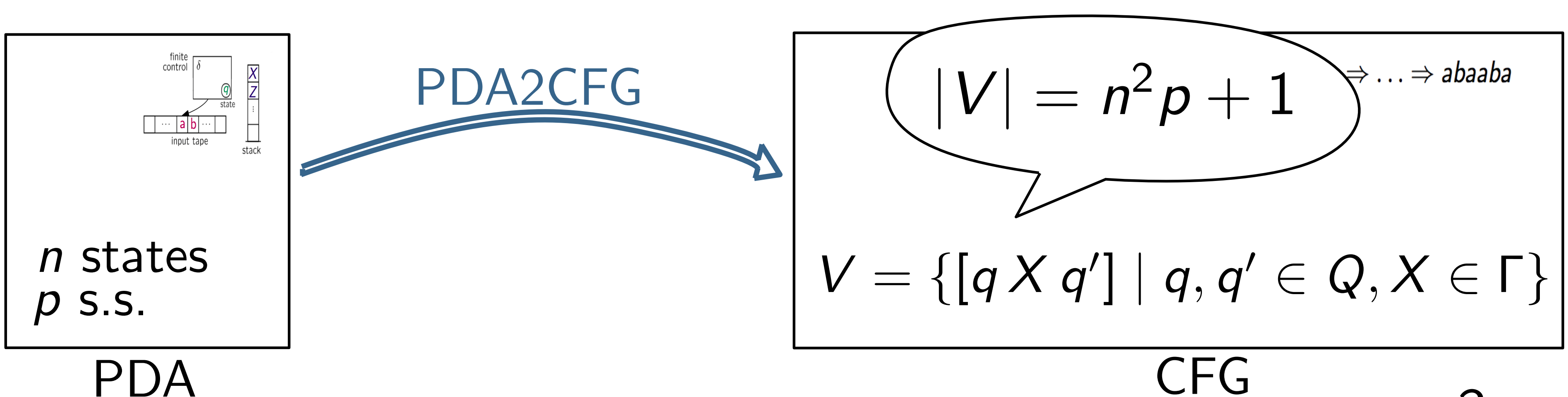
- **Pushdown Automata**



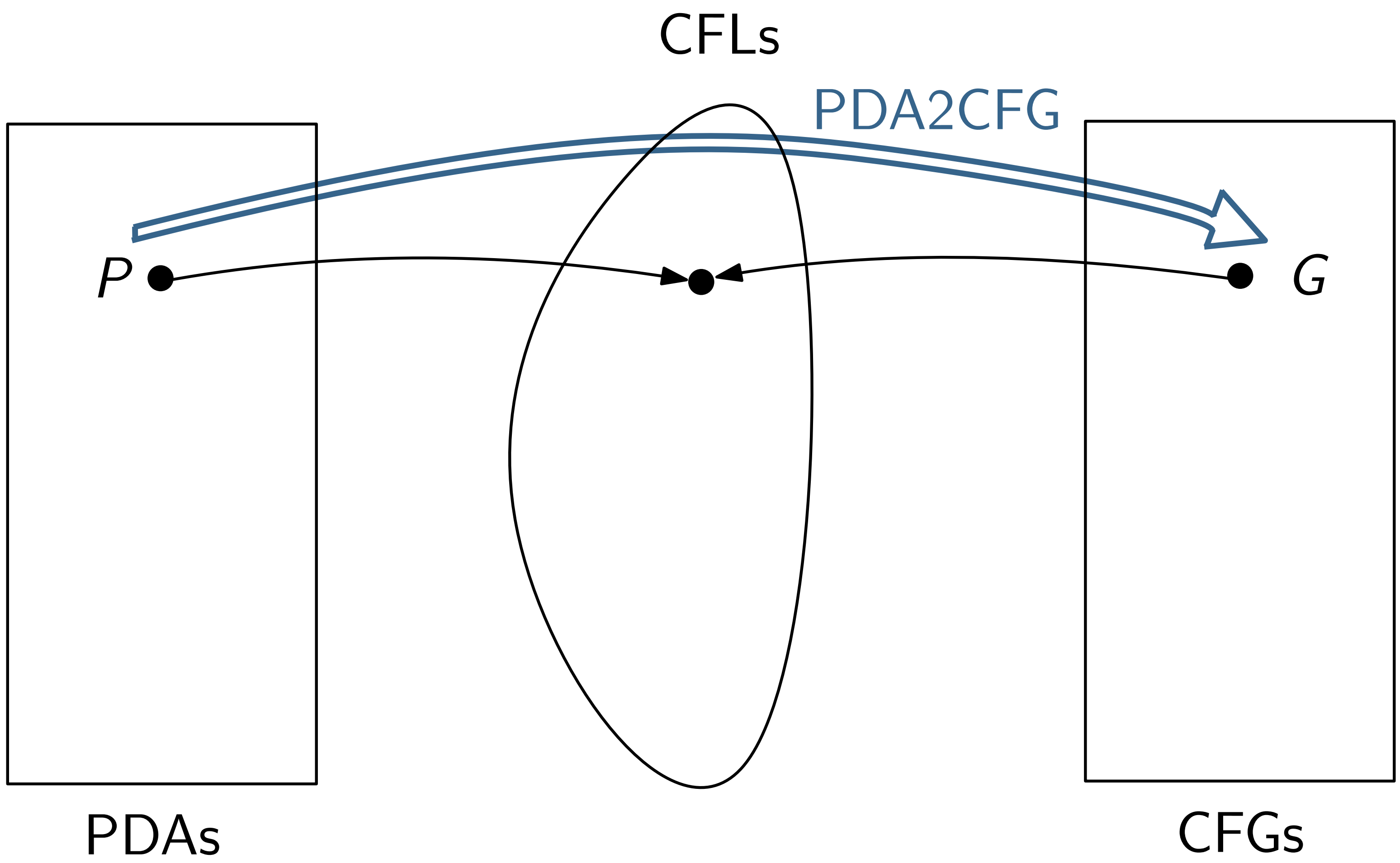
- **Context-free Grammar**

$$S \Rightarrow aSa \Rightarrow abSba \Rightarrow \ldots \Rightarrow abaaba$$

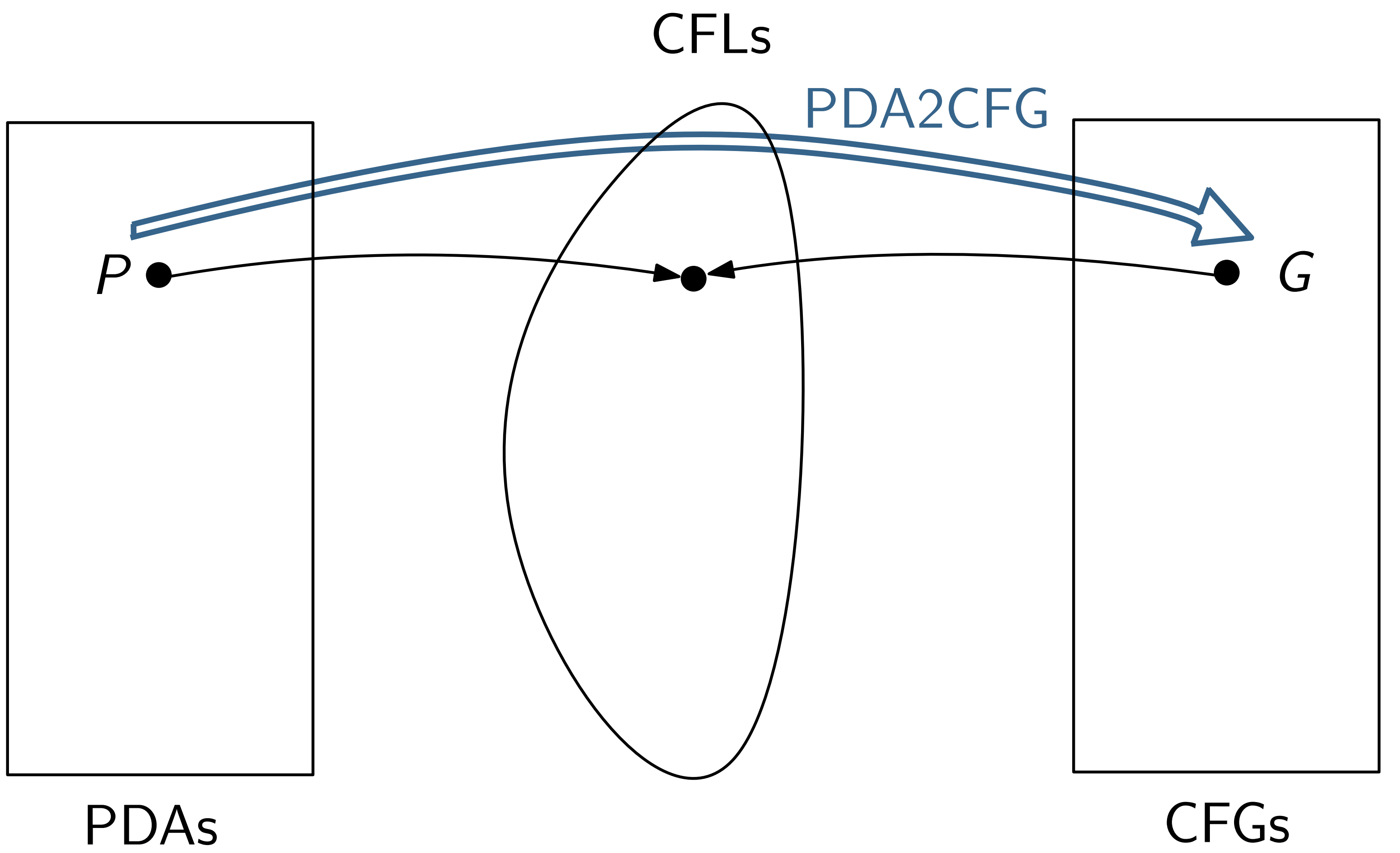
- **PDA2CFG**



Introduction



Introduction



Goldstine et. al.(1982): PDA2CFG is **optimal**

Introduction

A PUSHDOWN AUTOMATON OR A CONTEXT-FREE GRAMMAR—WHICH IS MORE ECONOMICAL?*,**

Jonathan GOLDSTINE, John K. PRICE*** and Detlef WOTSCHKE

Computer Science Department, The Pennsylvania State University, University Park, PA 16802, U.S.A.

Communicated by R. Book
Received January 1980
Revised September 1980

Abstract. For every pair of positive integers n and p, there is a language accepted by a real-time deterministic pushdown automaton with n states and p stack symbols and size $O(np)$, for which every context-free grammar needs at least n^2p+1 nonterminals if $n>1$ (or p non-terminals if $n=1$). It follows that there are context-free languages which can be recognized by pushdown automata of size $O(np)$, but which cannot be generated by context-free grammars of size smaller than $O(n^2p)$; and that the standard construction for converting a pushdown automaton to a context-free grammar is optimal in the sense that it infinitely often produces grammars with the fewest number of nonterminals possible.

Introduction

A PUSHDOWN AUTOMATON OR A CONTEXT-FREE GRAMMAR—WHICH IS MORE ECONOMICAL?*,**

Jonathan GOLDSTINE, John K. PRICE*** and Detlef WOTSCHKE

Computer Science Department, The Pennsylvania State University, University Park, PA 16802, U.S.A.

Communicated by R. Book
Received January 1980
Revised September 1980

Abstract. For every pair of positive integers n and p, there is a language accepted by a real-time deterministic pushdown automaton with n states and p stack symbols and size $O(np)$, for which

Abstract. For every pair of positive integers n and p, there is a language accepted by a real-time deterministic pushdown automaton with n states and p stack symbols and size $O(np)$, for which every context-free grammar needs at least n^2p+1 nonterminals if $n > 1$ (or p non-terminals if

Introduction

A PUSHDOWN AUTOMATON OR A CONTEXT-FREE GRAMMAR—WHICH IS MORE ECONOMICAL?*,**

Jonathan GOLDSTINE, John K. PRICE*** and Detlef WOTSCHKE

Computer Science Department, The Pennsylvania State University, University Park, PA 16802, U.S.A.

Communicated by R. Book
Received January 1980
Revised September 1980

Abstract. For every pair of positive integers n and p, there is a language accepted by a real-time deterministic pushdown automaton with n states and p stack symbols and size $O(np)$, for which every context-free grammar needs at least n^2p+1 nonterminals if $n>1$ (or p non-terminals if $n=1$). It follows that there are context-free languages which can be recognized by pushdown automata of size $O(np)$ but which cannot be generated by context-free grammars of size smaller

Notation. For positive integers n and p, let M_{np} be the PDA

$$M_{np} = (Q_n, \Sigma_{np}, \Gamma_p, \delta_{np}, q_1, Z_1, \emptyset),$$

where

$$Q_n = \{q_1, \ldots, q_n\}, \qquad \Gamma_p = \{Z_1, \ldots, Z_p\},$$

$$\Sigma_{np} = \{s_{ij}, r_{ij}, u, d \mid 1 \leq i \leq n, 1 \leq j \leq p\},$$

PDA2CFG is also optimal* in the unary case

Lower bound

Thm: There is a **family of unary PDAs** with n states and p stack symbols for which every equivalent CFG has $\Omega(n^2(p-2n-4))$ variables.

Family P(n,k)

- n states
- $p = 2n + k + 4$ stack symbols
- $\Sigma = \{a\}$

PDA2CFG is also optimal* in the unary case

Set of actions of P(n,k):

$$
\begin{aligned}
(q_0, a, S) &\hookrightarrow (q_0, X_k\, r_0) \\
(q_i, a, X_j) &\hookrightarrow (q_i, X_{j-1}\, r_m\, s_i\, X_{j-1}\, r_m) \;\; \forall\, i, m \in \{0, \ldots, n-1\}, \forall\, j \in \{1, \ldots, k\}, \\
(q_j, a, s_i) &\hookrightarrow (q_i, \varepsilon) \qquad \forall i, j \in \{0, \ldots, n-1\}, \\
(q_i, a, r_i) &\hookrightarrow (q_i, \varepsilon) \qquad \forall i \in \{0, \ldots, n-1\}, \\
(q_i, a, X_0) &\hookrightarrow (q_i, X_k \star) \qquad \forall i \in \{0, \ldots, n-1\}, \\
(q_i, a, X_0) &\hookrightarrow (q_{i+1}, X_k \$) \qquad \forall i \in \{0, \ldots, n-2\}, \\
(q_i, a, \star) &\hookrightarrow (q_{i-1}, \varepsilon) \qquad \forall i \in \{1, \ldots, n-1\}, \\
(q_0, a, \$) &\hookrightarrow (q_{n-1}, \varepsilon) \\
(q_{n-1}, a, X_0) &\hookrightarrow (q_{n-1}, \varepsilon)
\end{aligned}
$$

PDA2CFG is also optimal* in the unary case

Properties of P(n,k):

- P has only *one* accepting run
- $L(P) = \{a^\ell\}$ with $\ell \geq 2^{n^2 k}$

PDA2CFG is also optimal* in the unary case

Thm: There is a family of unary PDAs with n states and p stack symbols for which every equivalent CFG has $\Omega(n^2(p-2n-4))$ variables.

PDA2CFG is also optimal* in the unary case

Thm: There is a family of unary PDAs with n states and p stack symbols for which every equivalent CFG has $\Omega(n^2(p-2n-4))$ variables.

Proof:

- Find G s.t.: $L(G) = L(P) = \{a^\ell\}$ with $\ell \geq 2^{n^2 k}$.

PDA2CFG is also optimal* in the unary case

Thm: There is a family of unary PDAs with n states and p stack symbols for which every equivalent CFG has $\Omega(n^2(p-2n-4))$ variables.

Proof:

- Find G s.t.: $L(G) = L(P) = \{a^\ell\}$ with $\ell \geq 2^{n^2 k}$.
- **[Charikar et. al., 2005]**: The smallest CFG that generates exactly one word of length ℓ has $\Omega(log(\ell))$ variables.

PDA2CFG is also optimal* in the unary case

Thm: There is a family of unary PDAs with n states and p stack symbols for which every equivalent CFG has $\Omega(n^2(p-2n-4))$ variables.

Proof:

- Find G s.t.: $L(G) = L(P) = \{a^\ell\}$ with $\ell \geq 2^{n^2 k}$.
- **[Charikar et. al., 2005]**: The smallest CFG that generates exactly one word of length ℓ has $\Omega(log(\ell))$ variables.
- Then G has $\Omega(log(2^{n^2 k})) = \Omega(n^2 k)$ variables.

PDA2CFG is also optimal* in the unary case

Thm: There is a family of unary PDAs with n states and p stack symbols for which every equivalent CFG has $\Omega(n^2(p-2n-4))$ variables.

Proof:

- Find G s.t.: $L(G) = L(P) = \{a^\ell\}$ with $\ell \geq 2^{n^2 k}$.
- **[Charikar et. al., 2005]**: The smallest CFG that generates exactly one word of length ℓ has $\Omega(log(\ell))$ variables.
- Then G has $\Omega(log(2^{n^2 k})) = \Omega(n^2 k)$ variables.
- As $k = p - 2n - 4$, G has $\Omega(n^2(p-2n-4))$ variables.

$\square$

PDA2CFG is also optimal* in the unary case

	Equivalent **CFG**	
	Lower bound	Upper bound
$P(n, k)$	$\Omega(n^2 k)$	$\mathcal{O}(n^2(k+n))$

PDA2CFG is also optimal* in the unary case

	Equivalent **CFG**	
	Lower bound	Upper bound
$P(n,k)$	$\Omega(n^2 k)$	$\mathcal{O}(n^2(k+n))$

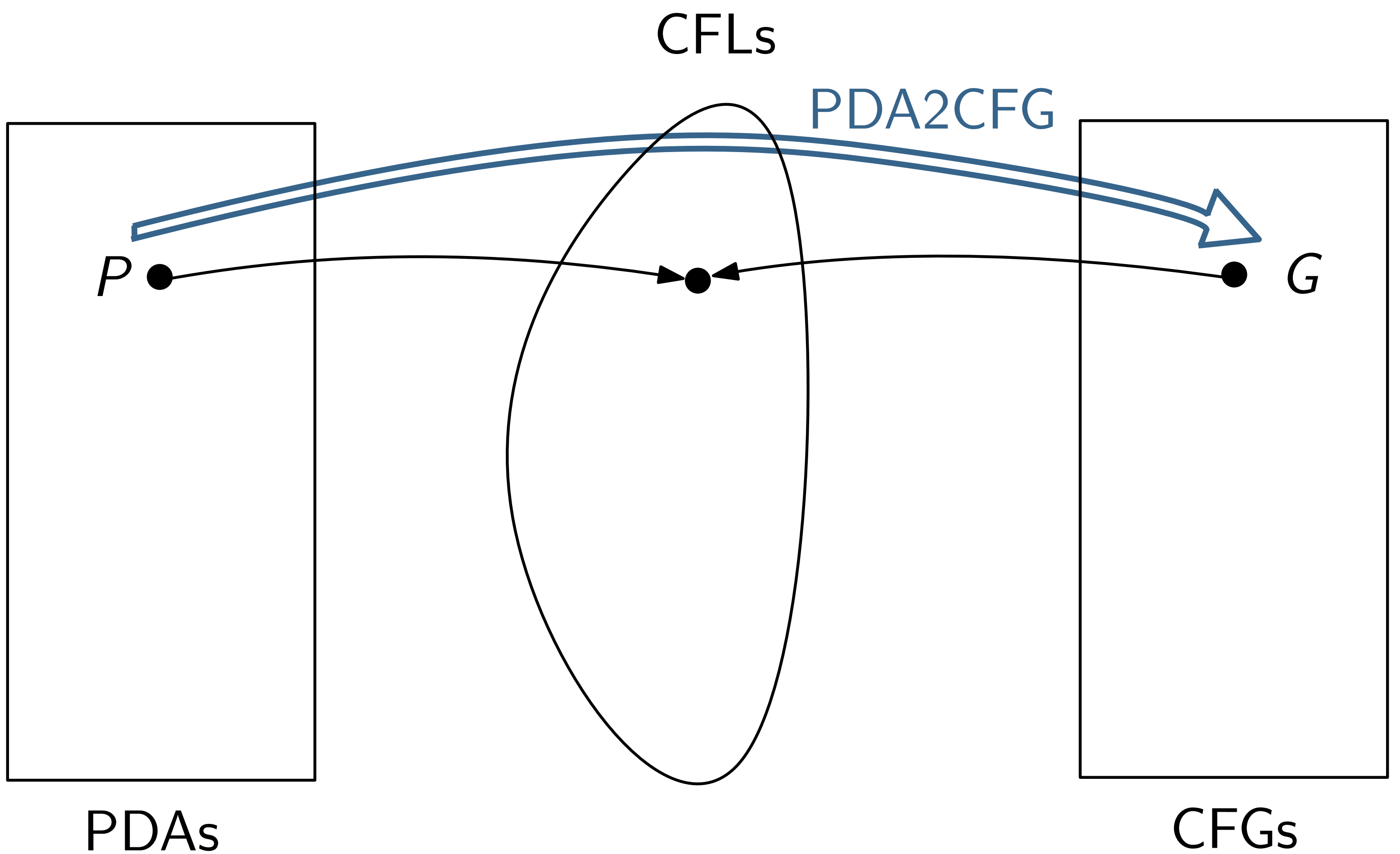
Asymptotically tight
if $n \leq Ck$ with $C > 0$

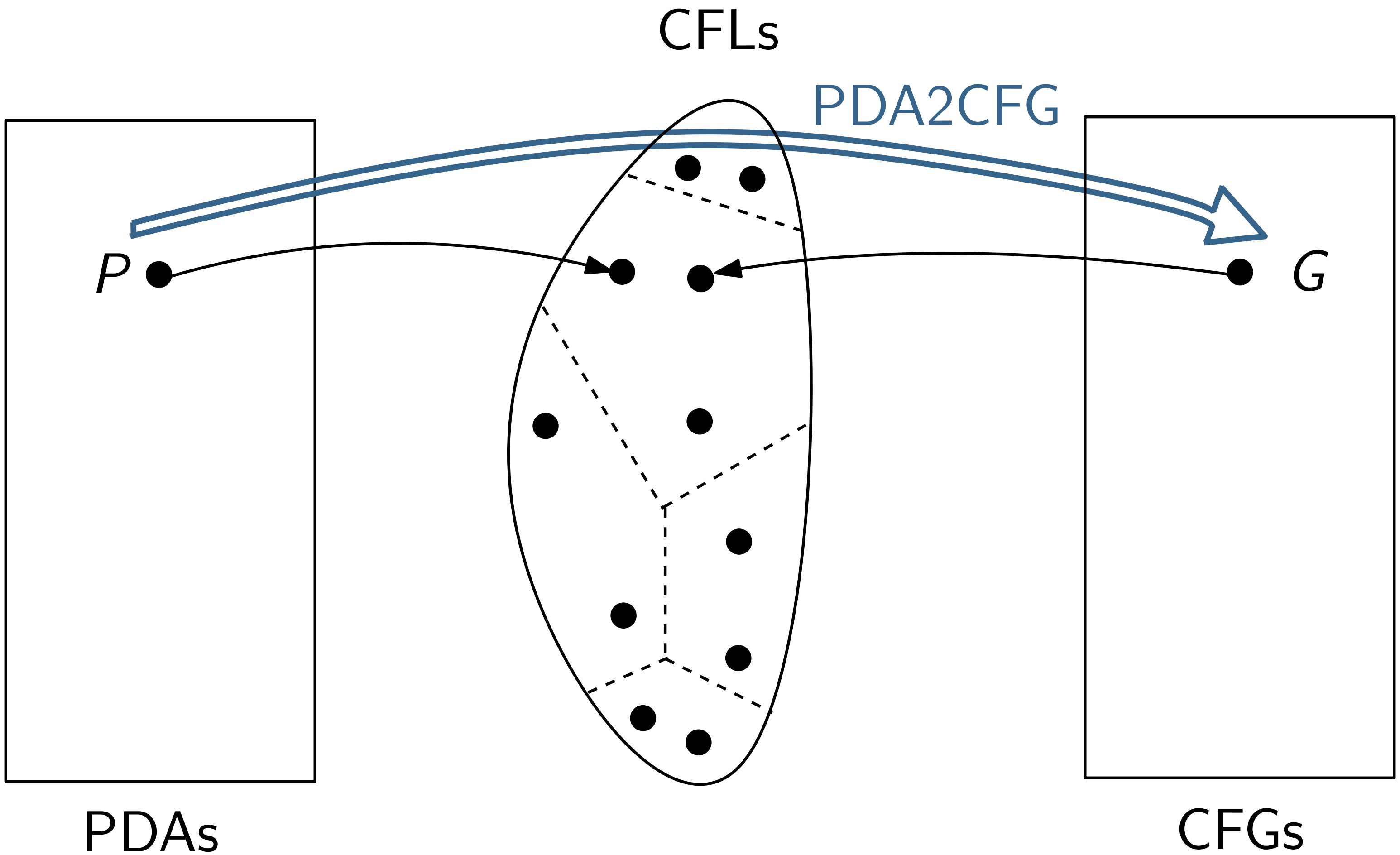
PDA2CFG is also optimal* in the unary case

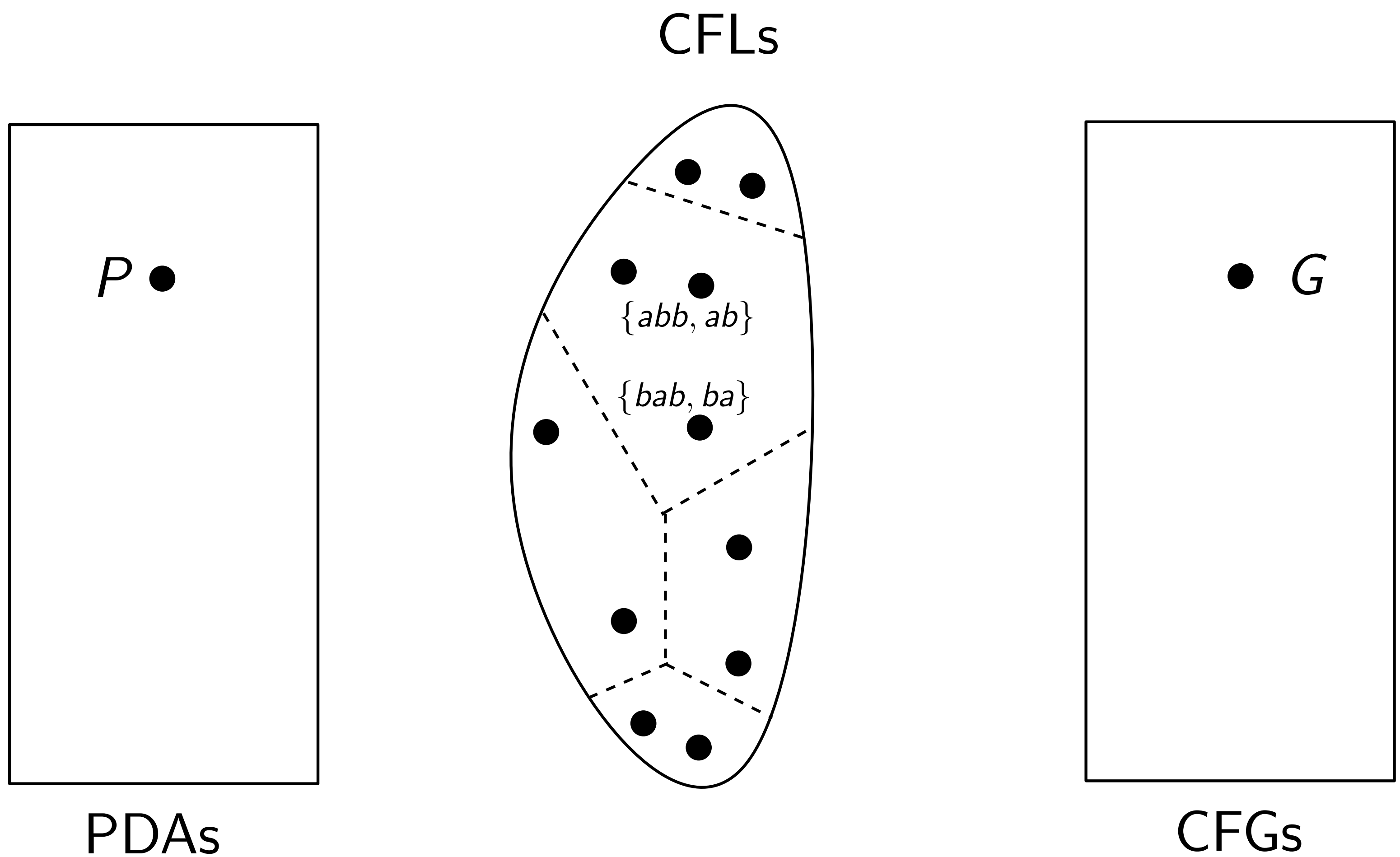
	PDA2CFG is **optimal**
$\lvert\Sigma\rvert > 1$	✔
$\lvert\Sigma\rvert = 1$	

PDA2CFG is also optimal* in the unary case

	PDA2CFG is **optimal**
$\|\Sigma\| > 1$	✓
$\|\Sigma\| = 1$	✓


CFLs
PDA2CFG
P
G
PDAs
CFGs


CFLs
PDA2CFG
P
G
PDAs
CFGs


CFLs
P
{abb, ab}
{bab, ba}
G
PDAs
CFGs

Parikh equivalence

Parikh-equivalent **words**

$$abb$$

$$bab$$

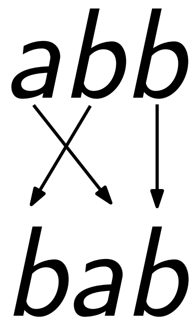
Parikh-equivalent **languages**

$$\{abb, ab\}$$

$$\{bab, ba\}$$

Parikh equivalence

Parikh-equivalent **words**



Parikh-equivalent **languages**

$$\{abb, ab\}$$

$$\{bab, ba\}$$

Parikh equivalence

Parikh-equivalent **words**

$$abb \approx bab$$

Parikh-equivalent **languages**

$$\{abb, ab\}$$

$$\{bab, ba\}$$

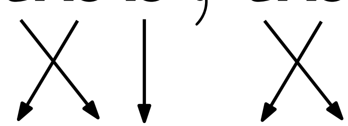
Parikh equivalence

Parikh-equivalent **words**

$$abb \approx bab$$

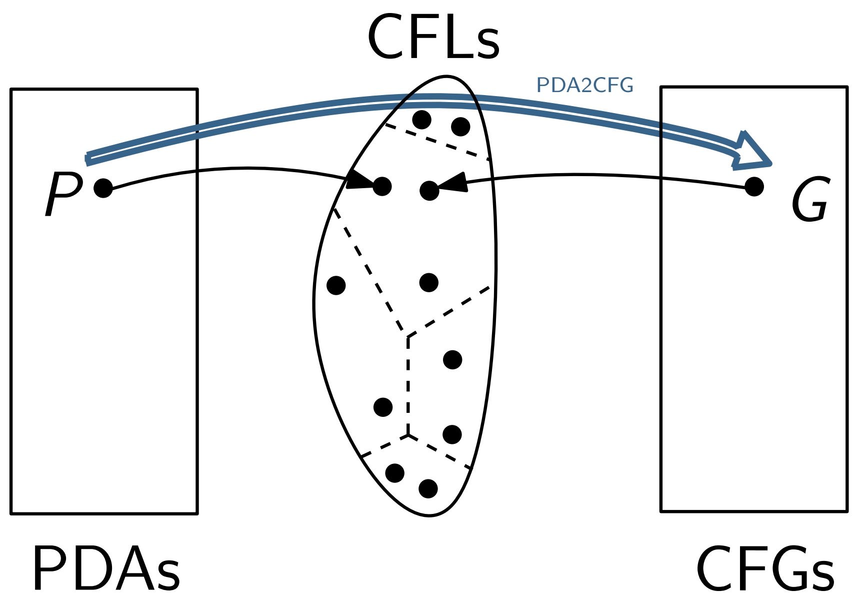
Parikh-equivalent **languages**

$$\{abb, ab\}$$



$$\{bab, ba\}$$

PDA2CFG for Parikh equivalence

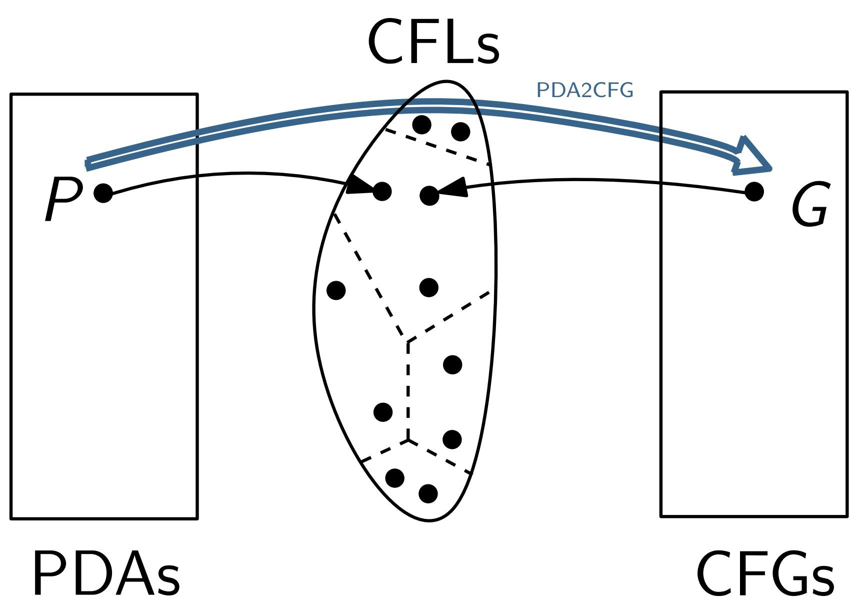


Idea:

Find F such that:

For all $L \in F$: every CFG G with $L(G) \approx L$ needs $\Omega(n^2\, p)$ variables

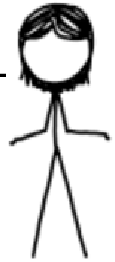
PDA2CFG for Parikh equivalence



Idea:

Find F such that:

For all $L \in F$: every CFG G with $L(G) \approx L$ needs $\Omega(n^2\, p)$ variables



$$\{abb, ab\}$$

$$\{abb, ab\}$$

$$\{abb, ab\}$$

$$\{abb, ab\}$$

$$\boxed{L = L' \Rightarrow L \approx L'}$$

$$\{abb, ab\}$$

$$\downarrow\downarrow\downarrow \quad \downarrow\downarrow$$

$$\{abb, ab\}$$

$$\boxed{L = L' \Rightarrow L \approx L'}$$

$$\{abb, ab\}$$

$$\downarrow\downarrow\downarrow \qquad \downarrow\downarrow$$

$$\{abb, ab\}$$

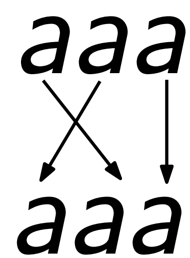
$$\boxed{L = L' \Rightarrow L \approx L'}$$

$$\not\Leftarrow$$

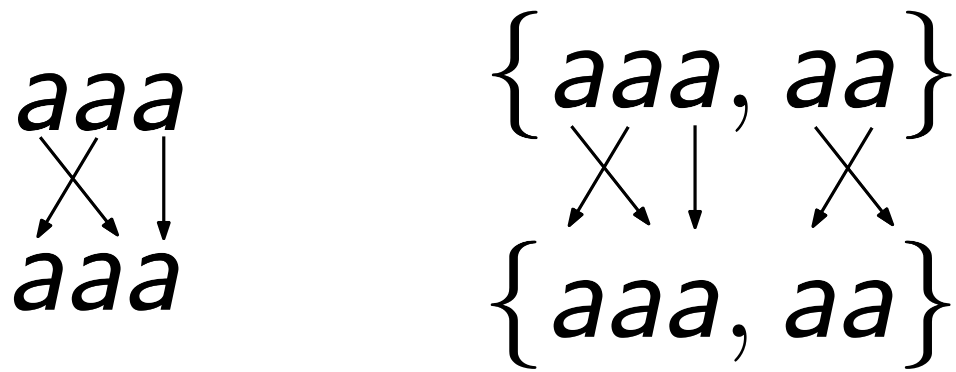
- If $|\Sigma| = 1$:

$$aaa$$

- If $|\Sigma| = 1$:



- If $|\Sigma| = 1$:



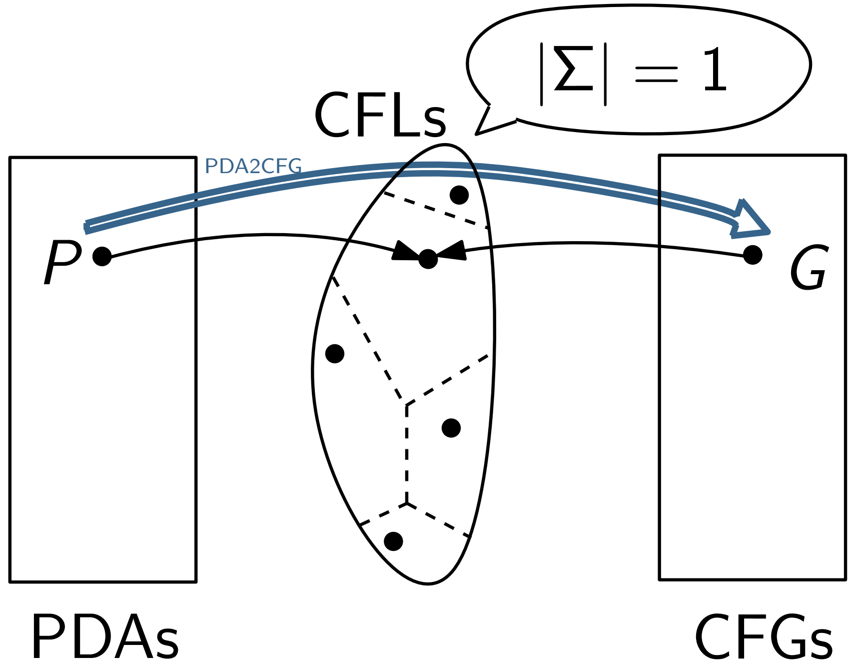
- If $|\Sigma| = 1$:

$$aaa \quad \to \quad aaa$$

$$\{aaa, aa\} \quad \to \quad \{aaa, aa\}$$

If $|\Sigma| = 1$:

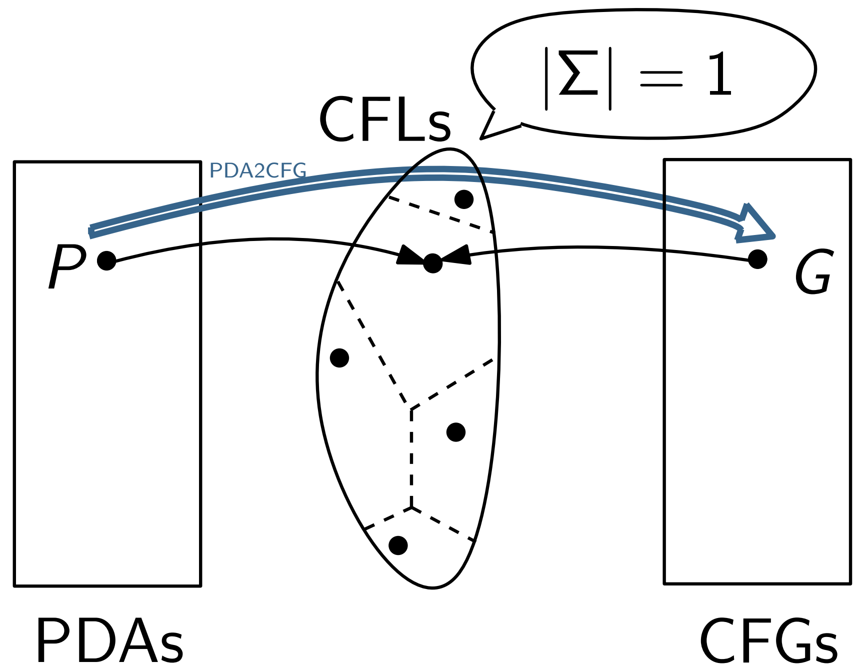
$$L = L' \iff L \approx L'$$



Idea:

Find F with $|\Sigma| = 1$ such that:

For all $L \in F$: every CFG G with $L(G) \approx L$ needs $\Omega(n^2\, p)$ variables



Idea:

$P(n, k)$ is **unary**

Find F with $|\Sigma| = 1$ such that:

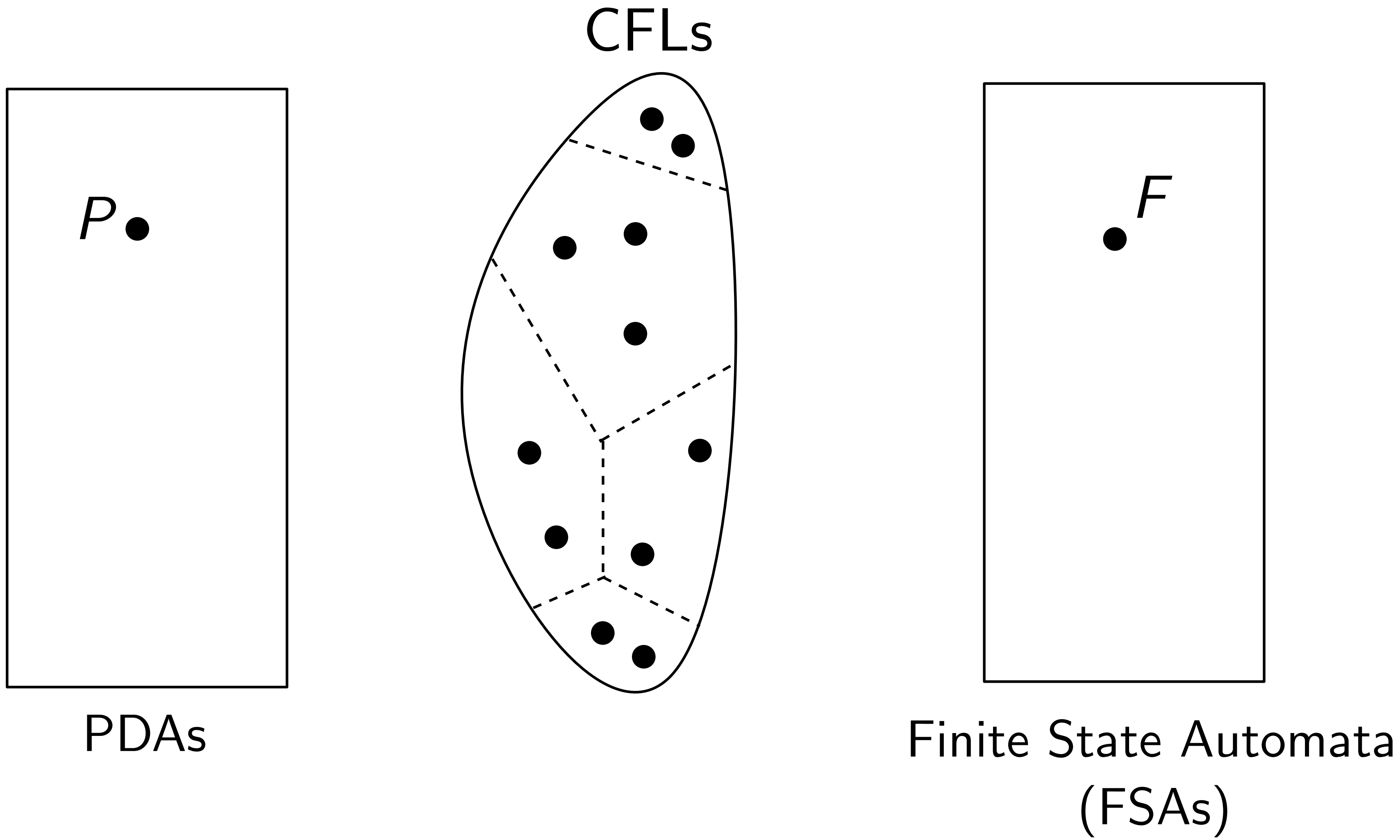
For all $L \in F$: every CFG G with $L(G) \approx L$ needs $\Omega(n^2 p)$ variables

PDA2CFG is optimal* for Parikh equivalence

	PDA2CFG is **optimal**
$\|\Sigma\| > 1$	✓
$\|\Sigma\| = 1$	✓
Parikh equivalence	✓

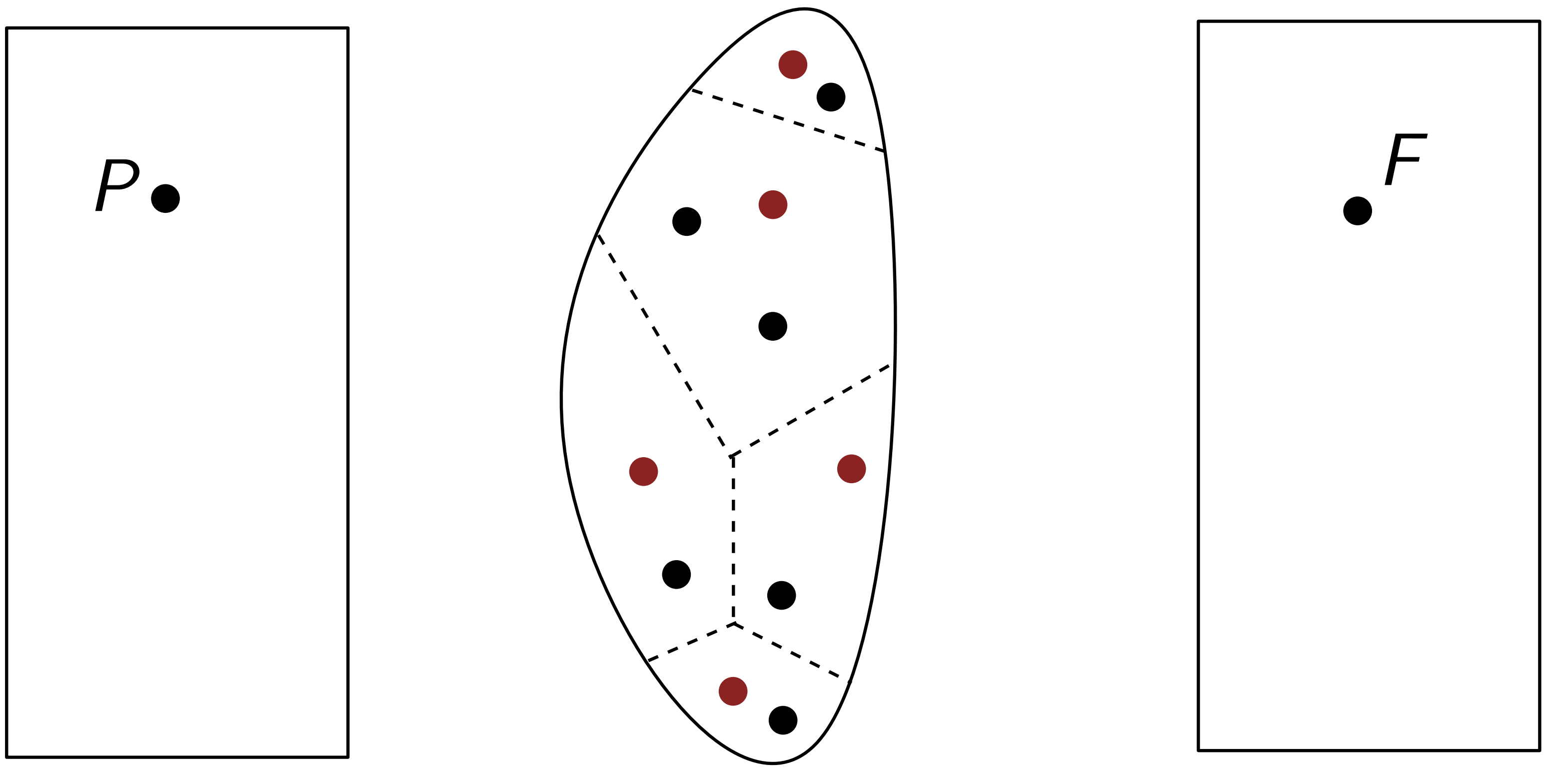
2-step procedure for Parikh-equivalent FSA

Thm: Every CFL is Parikh-equivalent to some regular language



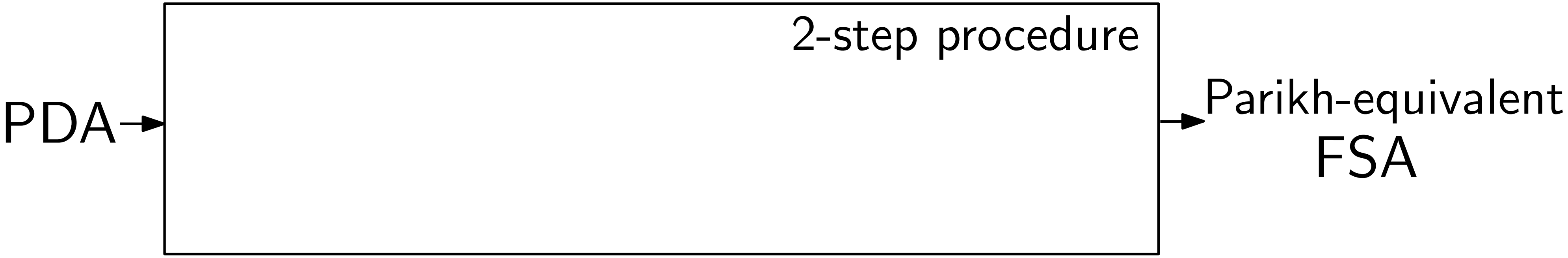
2-step procedure for Parikh-equivalent FSA

Thm: Every CFL is Parikh-equivalent to some regular language



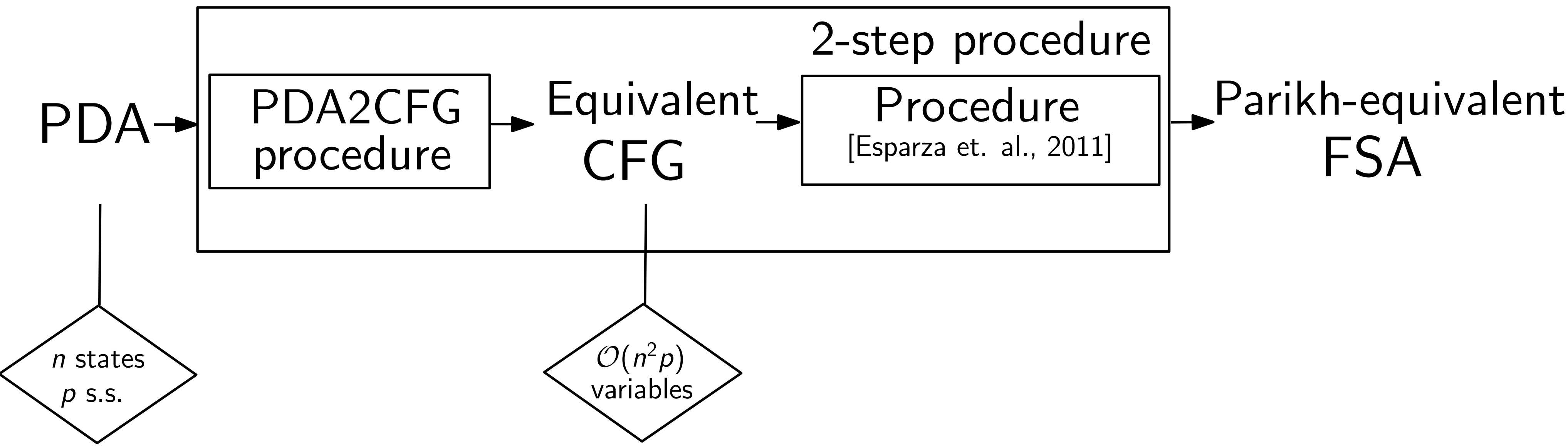
2-step procedure for Parikh-equivalent FSA

Upper bound



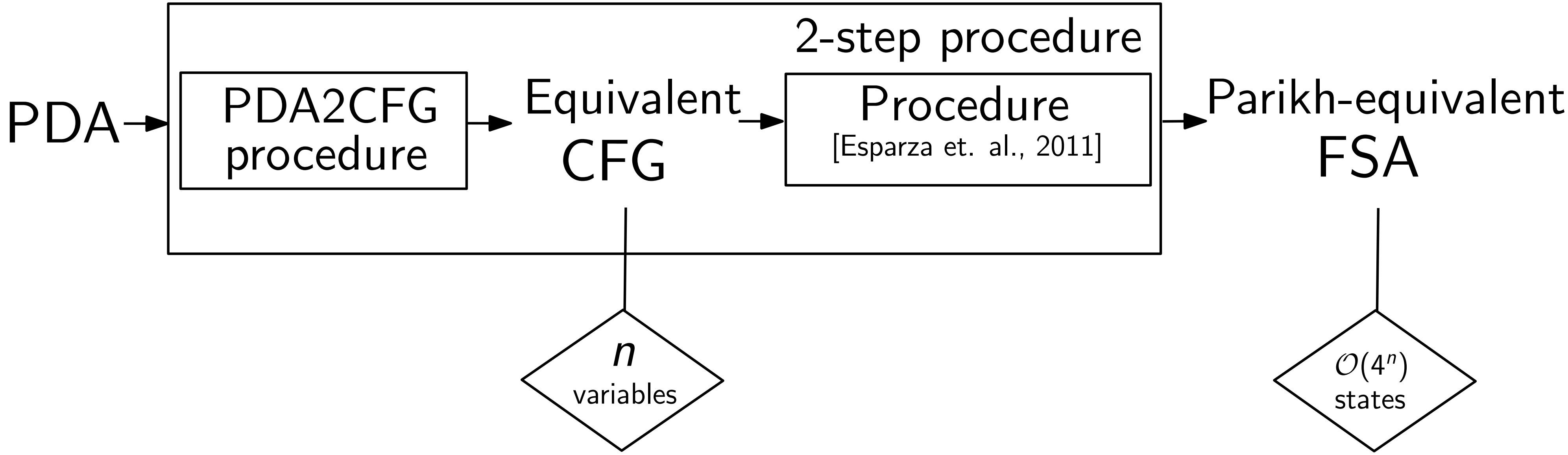
2-step procedure for Parikh-equivalent FSA

Upper bound



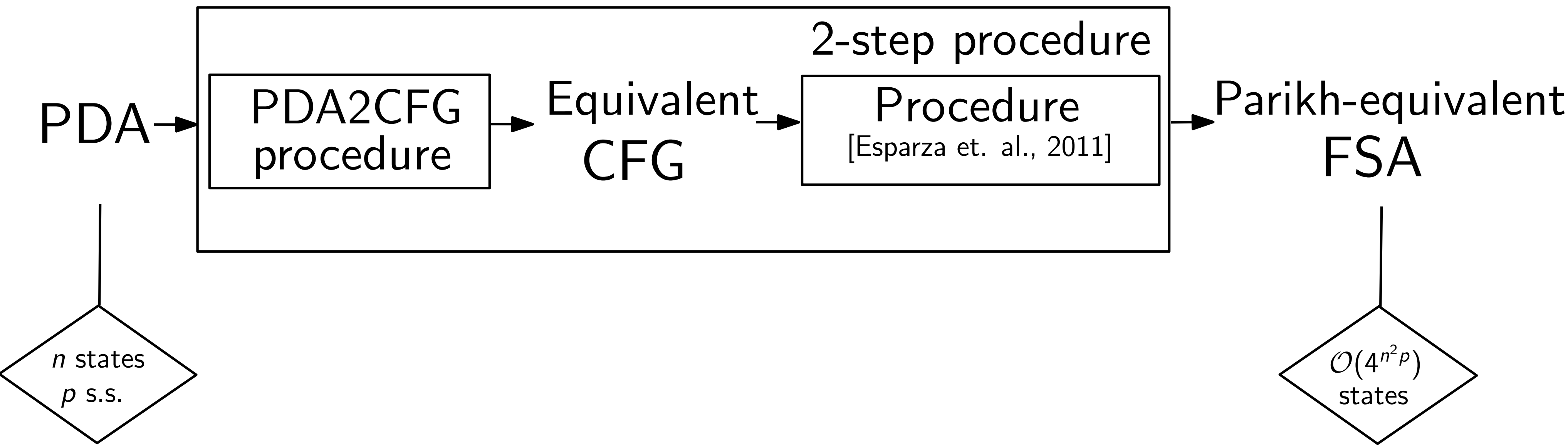
2-step procedure for Parikh-equivalent FSA

Upper bound



2-step procedure for Parikh-equivalent FSA

Upper bound

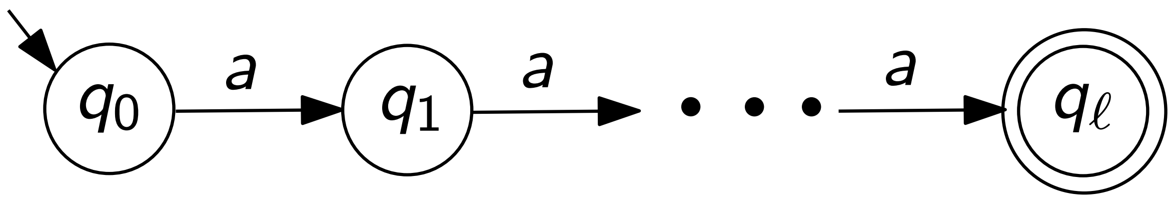


Thm: Given a PDA with n states and p s.s., there is a **Parikh-equivalent FSA** with $\mathcal{O}(4^{n^2 p})$ **states**.

2-step procedure for Parikh-equivalent FSA

Lower bound

- Using the family $P(n, k)$
- $L(P) = \{a^\ell\}$ with $\ell \geq 2^{n^2 k}$



Thm: There is a family of unary PDAs with n states and p stack symbols for which every equivalent FSA needs at least $2^{n^2(p-2n-4)} + 1$ **states**.

2-step procedure for Parikh-equivalent FSA

	Parikh-equivalent **FSA**	
	Lower bound	Upper bound
$P(n,k)$	$\Omega(2^{n^2 k})$	$\mathcal{O}(4^{n^2(k+2n+4)})$

Asymptotically tight
if $n \leq Ck$ with $C > 0$

Conclusions

- PDA2CFG is also **optimal** in the **unary case**
  - PDA2CFG is **optimal** for **Parikh-equivalence**
- PDA2CFG-based procedure for Parikh-equivalent FSA **is close to optimal**

Conclusions

- PDA2CFG is also **optimal** in the **unary case**
  - PDA2CFG is **optimal** for **Parikh-equivalence**
- PDA2CFG-based procedure for Parikh-equivalent FSA **is close to optimal**

Thank you!