

Operational state complexity under Parikh equivalence

Giovanna Lavado¹, Giovanni Pighizzini¹, Shinnosuke Seki^{2,3}

- ① Dipartimento di Informatica, Università degli Studi di Milano, Italy.
- ② Helsinki Institute for Information Technology (HIIT)
- ③ Department of Information and Computer Science, Aalto University

DCFS 2014, August 6th, 2014

Parikh map

- $\Sigma = \{a_1, a_2, \dots, a_m\}$ be an alphabet
- $|w|_a$ be the number of occurrences of a letter $a \in \Sigma$ on the word w

Parikh map

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

Parikh image

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

Parikh equivalence

Parikh equivalence $=_{\pi}$

Languages $L_1, L_2 \subseteq \Sigma^*$ are *Parikh equivalent* if $\psi(L_1) = \psi(L_2)$.
We write $L_1 =_{\pi} L_2$.

Parikh equivalence can be naturally extended among languages, grammars, and machines.

Example

Let $R = (ab)^*$ and M be a pushdown automaton (PDA) to accept

$$L = \{a^i b^i \mid i \geq 0\}.$$

Then $\psi(R) = \psi(L(M)) = \{(i, i) \mid i \geq 0\}$. Thus, the language R is Parikh equivalent to the PDA M ($R =_{\pi} M$).

Semilinear set

A set $S \subseteq \mathbb{N}^n$ is *linear* if there exist $\vec{v}_1, \dots, \vec{v}_k \subseteq \mathbb{N}^n$ such that

$$S = \{i_1 \vec{v}_1 + \dots + i_k \vec{v}_k \mid i_1, \dots, i_k \in \mathbb{N}\}.$$

A finite union of linear sets is called a *semilinear set*.

The semilinear set admits many other representations including

- regular language/expression
- (non)deterministic finite automaton (NFA/DFA)
- context-free language (Parikh's theorem [Parikh 66]).
- context-free grammar (CFG)
- pushdown automaton
- reversal-bounded counter machine (see, e.g., [Ibarra 78])
- etc.

State complexity of Parikh-equivalent conversion

CFG \implies_{π} DFA

Question

How costly is it to convert one representation to another?

By \implies_{π} , we mean the Parikh equivalent conversion.

CNFG \implies_{π} DFA [Lavado et al. 13]

For a CFG in Chomsky normal form (CNFG) G with h variables, there exists a Parikh equivalent DFA A with $2^{O(h^2)}$ states.

State complexity of Parikh-equivalent conversion

NFA \Rightarrow_{π} DFA

NFA \Rightarrow_{π} DFA [Lavado et al. 13]

NFA		DFA
n states	\Rightarrow_{π}	$e^{\sqrt{n \ln n}}$ states
A_1		A_2

Nonunary NFA \Rightarrow_{π} DFA [Lavado et al. 13]

NFA with no unary word		DFA
n states	\Rightarrow_{π}	$O(n^{3m^3+6m^2} m^{m^3/2+m^2+2m+5})$ states
A_1		A_2

Lemma [Lavado et al. 13]

For each n -state NFA A over an m -letter alphabet, there exist $m + 1$ NFAs A_0, A_1, \dots, A_m such that

- for $1 \leq i \leq m$, A_i consists of n states accepting $L(A) \cap \{a_i\}^*$;
- A_0 consists of $n(m + 1) + 1$ states accepting $L(A) \setminus (\bigcup_{i=1}^m L(A_i))$.

Moreover, if A is deterministic, then so are A_0, A_1, \dots, A_m .

Regular operations under Parikh equivalence

Problems

For DFAs A and B of n_1 and n_2 states, respectively, we consider the following problems:

- 1 For a unary operation f , how small can we make a DFA M that is Parikh equivalent to $f(L(A))$?
- 2 For a binary operation g , how small can we make a DFA M that is Parikh equivalent to $g(L(A), L(B))$?

Regular operations under Parikh equivalence

Summary table

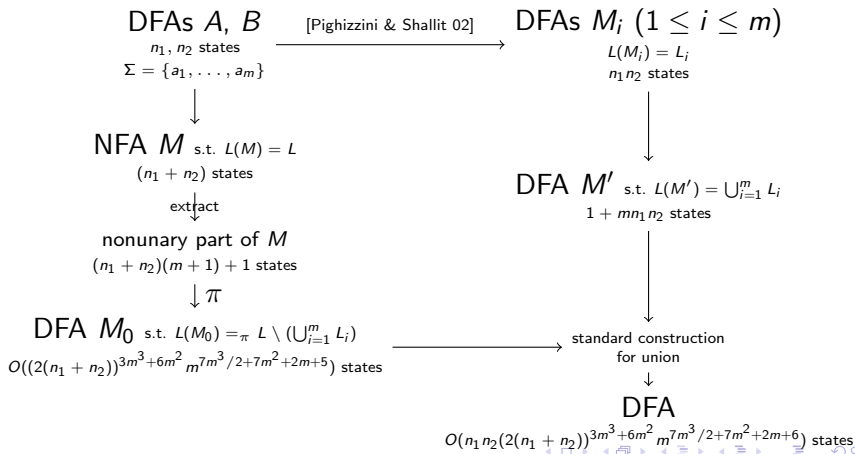
Let $\Sigma = \{a_1, a_2, \dots, a_m\}$.

	conventionally		under Parikh equivalence
	$m \geq 2$	$m = 1$	
\cup, \cap	$n_1 n_2$ [Yu 00]		$n_1 n_2$
•	$(2n_1 - 1)2^{n_2 - 1}$ [Yu et al. 94]	$n_1 n_2$ [Yu 00]	$O(n_1 n_2 (2(n_1 + n_2))^{3m^3 + 6m^2} m^{7m^3 / 2 + 7m^2 + 2m + 6})$
Shuffle	$2^{n_1 n_2} - 1$ [Câmpeanu et al. 02]		
*	$2^{n-1} + 2^{n-2}$ [Yu et al. 94]	$(n - 1)^2 + 1$ [Yu et al. 94]	$O((2n)^{3m^3 + 6m^2 + 1} n m^{7m^3 / 2 + 7m^2 + 2m + 6})$
Reversal	2^n [Yu et al. 94]	n	n

State complexity under Parikh equivalence

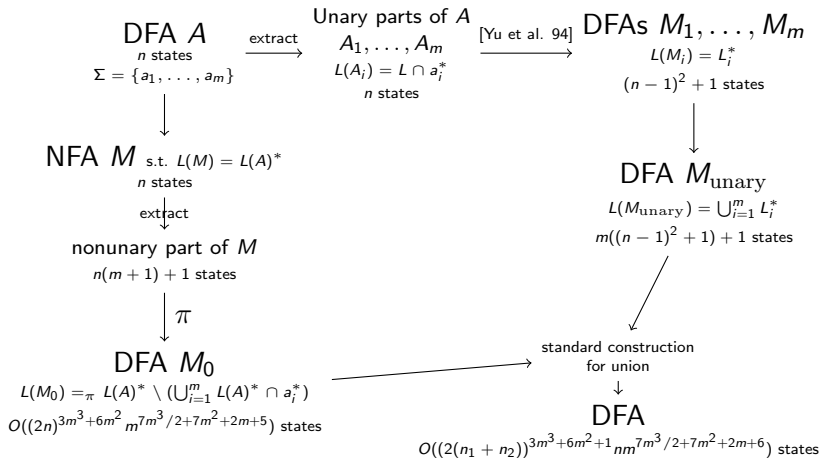
Catenation

Let $L = L(A)L(B)$, and let $L_i = L \cap \{a_i\}^*$ for $1 \leq i \leq m$.



State complexity under Parikh equivalence

Star



State complexity under Parikh equivalence

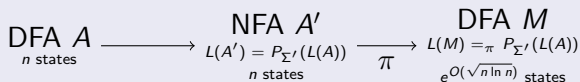
Projection

The *projection* of a word $w \in \Sigma^*$ over $\Sigma' \subseteq \Sigma$, $P_{\Sigma'}(w)$, is the word obtained by removing all the non- Σ' symbols from w (see, e.g., [Jirásková & Masopust 12]).

Given a DFA A of n states, an exponential number of states in n is required for a DFA to accept $P_{\Sigma'}(L(A))$.

Projection under Parikh equivalence

Under Parikh equivalence, $e^{O(\sqrt{n \ln n})}$ is enough and this is tight.



Intersection and complement: revisited

Non-commutativity with Parikh mapping

Intersection is not commutative with Parikh mapping

$\psi(a^+b^+ \cap b^+a^+) \neq \psi(a^+b^+) \cap \psi(b^+a^+)$ holds; in fact,

$$\begin{aligned}\psi(a^+b^+ \cap b^+a^+) &= \emptyset \\ \psi(a^+b^+) \cap \psi(b^+a^+) &= \{(i, j) \mid i, j \geq 1\}.\end{aligned}$$

Complement is not commutative with Parikh mapping

$\psi(\overline{a^*b^*}) \neq \overline{\psi(a^*b^*)}$ holds; in fact,

$$\begin{aligned}\psi(\overline{a^*b^*}) &= \{(i, j) \mid i, j \geq 0\} \\ \overline{\psi(a^*b^*)} &= \emptyset.\end{aligned}$$

Intersection and complement: revisited

Problem setting

Problem: intersection

DFA A, B
 n_1, n_2 states

→

DFA M
 $\psi(L(M)) = \psi(L(A)) \cap \psi(L(B))$
How many states needed?

Problem: complement (left open!)

DFA A
 n states

→

DFA M
 $\psi(L(M)) = \overline{\psi(L(A))}$
How many states needed?

Intersection and complement: revisited

Theorem

Let A, B be DFAs with respectively n_1, n_2 states over $\Sigma = \{a_1, \dots, a_m\}$. There exists a DFA M whose Parikh map is equal to $\psi(L(A)) \cap \psi(L(B))$ and which contains

$$O(n^{(2m-1)(3m^3+6m^2)+2} p(n)^{2(3m^3+6m^2)+m})$$

states, where $p(n) = O(n^{3m^2} m^{m^2/2+2})$.

Proof.

Revisiting the Ginsburg and Spanier's proof [Ginsburg & Spanier 64] of the closure property of semilinear sets under intersection. □

Complexity of transforming FIN to equivalent CFG

In the conventional sense

FIN \implies CNFG

Finite language
 $L = \{w_1, w_2, \dots, w_k\}$
over $\Sigma_m = \{a_1, \dots, a_m\}$

\longrightarrow

CNFG G s.t. $L(G) = L$
 $m + \sum_{i=1}^k |w_i|$ variables

This bound cannot be reduced significantly.

Lemma ([Domaratzki et al. 02])

For each $k \geq 1$, a singleton language $L_k = \{w\}$ with $|w| = 2^k + k - 1$ such that any CNFG for L_k requires $O(2^k/k)$ variables.

Complexity of transforming FIN to equivalent CFG

Under Parikh equivalence

A grammar G is *binary normal form grammar* (BNFG) if every production is in one of the following forms:

$$A \rightarrow a, A \rightarrow \lambda, A \rightarrow B, A \rightarrow BC,$$

where A, B, C are variables and $a \in \Sigma$.

Lemma

Let $n \geq 1$ and $T \subseteq \{0, 1, 2, \dots, n-1\}^m$. There exists a BNFG G of $O(n^{m/3})$ variables s.t. $\psi(L(G)) = T$. The bound is asymptotically tight.

Theorem

Let $L \subseteq \{w \mid |w|_a < n \text{ for any } a \in \Sigma\}$. Then there is a CNFG with $O(n^{m/3})$ variables which is Parikh equivalent to L . This bound is asymptotically tight.

Proof idea for the lemma

Let $r = \lceil n^{1/3} \rceil$. Any integer less than n can be expressed in base r using at most 3 digits as $ir^2 + jr + k$ for some $0 \leq i, j, k < r$.

- 1 Prepare r^m variables G_{k_1, \dots, k_m} for $0 \leq k_1, \dots, k_m < r$ such that $L(G_{k_1, \dots, k_m}) = \{a_1^{k_1} \dots a_m^{k_m}\}$.
- 2 Based on them, prepare r^m variables F_{j_1, \dots, j_m} for $0 \leq j_1, \dots, j_m < r$ such that $L(F_{j_1, \dots, j_m}) = \{a_1^{j_1 r} \dots a_m^{j_m r}\}$.
- 3 Based on them, prepare r^m variables E_{i_1, \dots, i_m} for $0 \leq i_1, \dots, i_m < r$ such that $L(E_{i_1, \dots, i_m}) = \{a_1^{i_1 r^2} \dots a_m^{i_m r^2}\}$.
- 4 Finally, we define

$$\begin{array}{ll}
 S & \rightarrow E_{i_1, \dots, i_m} S_{i_1, \dots, i_m} \quad \text{for } 0 \leq i_1, \dots, i_m < r \\
 S_{i_1, \dots, i_m} & \rightarrow F_{j_1, \dots, j_m} G_{k_1, \dots, k_m} \quad \text{for all } 0 \leq j_1, \dots, j_m, k_1, \dots, k_m < r \\
 & \text{s.t. } (i_1 r^2 + j_1 r + k_1, \dots, i_m r^2 + j_m r + k_m) \in T
 \end{array}$$

Thank you very much!

References I



[Câmpeanu et al. 02]

C. Câmpeanu, K. Salomaa, S. Yu.

Tight lower bound for the state complexity of shuffle of regular languages.

Journal of Automata, Languages, and Combinatorics 7(3), 303-310, 2002.



[Domaratzki et al. 02]

M. Domaratzki, G. Pighizzini, J. Shallit.

Simulating finite automata with context-free grammars.

Information Processing Letters 84(6), pp.339-344, 2002.

References II



[Ginsburg & Spanier 64]

S. Ginsburg, E. H. Spanier.

Bounded ALGOL-like languages.

Transactions of American Mathematical Society 113,
pp.333-368, 1964.



[Ibarra 78] O. H. Ibarra.

Reversal-bounded multicounter machines and their decision
problems.

Journal of the ACM 25(1), pp.116-133, 1978.

References III



[Jirásková & Masopust 12]

G. Jirásková, T. Masopust.

On a structural property in the state complexity of projected regular languages.

Theoretical Computer Science 449, pp.93-105, 2012.



[Lavado et al. 13] G. J. Lavado, G. Pighizzini, S. Seki.

Converting nondeterministic automata and context-free grammars into Parikh equivalent one-way and two-way deterministic automata.

Information and Computation 228, pp.1-15, 2013.





[Parikh 66] R. J. Parikh.

On context-free languages.

Journal of the ACM 13(4), 570-581, 1966.

References IV

-  [Pighizzini & Shallit 02] G. Pighizzini, J. Shallit.
Unary language operations, state complexity and jacobsthal's
function.
International Journal of Foundations of Computer Science
13(1), pp.145-159, 2002.
-  [Yu 00]
S. Yu.
State complexity of regular languages
Journal of Automata, Languages, and Combinatorics 6,
pp.221-234, 2000.

References V



[Yu et al. 94]

S. Yu, Q. Zhuang, K. Salomaa.

The state complexity of some basic operations on regular languages.

Theoretical Computer Science 125, pp.315-328, 1994.