Operational State Complexity under Parikh Equivalence

Giovanna Lavado[1] Giovanni Pighizzini[1] Shinnosuke Seki[2,3]

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

ICTCS 2014
Università degli Studi di Perugia, Italy
September 17-19, 2014

UNIVERSITÀ DEGLI STUDI DI MILANO

Subset construction [Rabin&Scott '59]

NFA		DFA
n states	$\Longrightarrow$	2^n states
L		L

Moreover, this state bound cannot be reduced

[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 '66]

Standard equivalence: NFAs vs DFAs

Subset construction [Rabin&Scott '59]

NFA		DFA
n states	$\Longrightarrow$	2^n states
L		L

Moreover, this state bound cannot be reduced

[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 '66]

Parikh equivalence: 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* is $\psi(L) = \{\psi(w) \mid w \in L\}$.

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

Parikh equivalence: Parikh's theorem

Theorem ([Parikh '66])

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

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

From NFAs to Parikh equivalent DFAs

We have the following Parikh equivalent conversion:

Theorem (NFA to DFA)

NFA		DFA
n *states*	$\Longrightarrow_\pi$	$e^{O(\sqrt{n \cdot \ln n})}$ *states*
L_1		L_2

Moreover, this cost is tight.

Quite surprisingly:

Polynomial conversion

If the given NFA accepts only nonunary strings then the cost reduces to

a *polynomial* in n.

From NFAs to Parikh equivalent DFAs

We have the following Parikh equivalent conversion:

Theorem (NFA to DFA)

NFA		DFA
n *states*	$\Longrightarrow_\pi$	$e^{O(\sqrt{n \cdot \ln n})}$ *states*
L_1		L_2

Moreover, this cost is tight.

Quite surprisingly:

Polynomial conversion

If the given NFA accepts only nonunary strings then the cost reduces to

a *polynomial* in n.

Our Goal

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

Problem (DFAs to DFA)

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

where:

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

Standard equivalence: concatenation

A, B DFAs
n_1, n_2 states
$L(A)L(B)$

$\Longrightarrow$

C DFA
$2^{n_1+n_2}$ states
$L(C) = L(A)L(B)$

In the worst case: $(2n_1 - 1)2^{n_2-1}$ states

[Yu '00]

Under Parikh equivalence we reduce this bound.

Standard equivalence: concatenation

A, B DFAs n_1, n_2 states $L(A)L(B)$	$\Longrightarrow$	C DFA $2^{n_1+n_2}$ states $L(C) = L(A)L(B)$

In the worst case: $(2n_1 - 1)2^{n_2-1}$ states

[Yu '00]

Under Parikh equivalence we reduce this bound.

Concatenation under Parikh equivalence

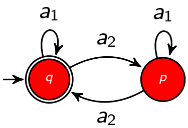
One of our contribution

Problem (DFAs to DFA)

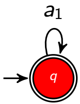
A, B DFAs		C DFA
n_1, n_2 *states*	$\Longrightarrow_\pi$	$L(C) =_\pi L$
$L = L(A)L(B)$		*how many states?*

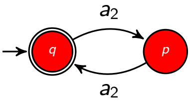
- Upper bound: $e^{\sqrt{n \cdot \ln n}}$, where $n = n_1 + n_2$
  by Parikh equivalent conversion
- Lower bound: $n_1 n_2$ states
  by unary case [Yu '00]

Unary and *nonunary* parts of a language

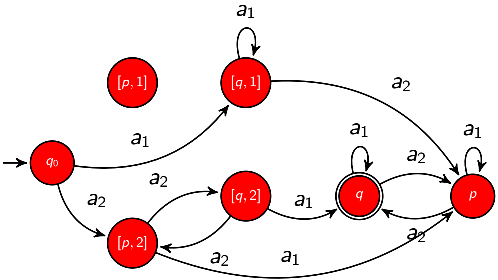


Unary parts:





Nonunary part:



$$L(A) = \bigcup_{i=0}^{m} L(A_i)$$

Concatenation under Parikh equivalence: proof idea

DFAs A, B
n_1, n_2 states
$L = L(A)L(B)$
$\Sigma = \{a_1, \ldots, a_m\}$

Concatenation under Parikh equivalence: proof idea

$\forall i = 1 \ldots m$
$A_i, O(n_1)$ states
$B_i, O(n_2)$ states

unary

DFAs A, B
n_1, n_2 states
$L = L(A)L(B)$
$\Sigma = \{a_1, \ldots, a_m\}$

Concatenation under Parikh equivalence: proof idea

$\forall i = 1 \dots m$
$A_i, O(n_1)$ states
$B_i, O(n_2)$ states

$\Longrightarrow$

$\forall i = 1 \dots m$
DFA M_i
$L(M_i) = L(A_i)L(B_i)$
$O(n_1 n_2)$ states

unary

[Yu '00]

DFAs A, B
n_1, n_2 states
$L = L(A)L(B)$
$\Sigma = \{a_1, \dots, a_m\}$

Concatenation under Parikh equivalence: proof idea

DFAs A, B
n_1, n_2 states
$L = L(A)L(B)$
$\Sigma = \{a_1, \ldots, a_m\}$

unary

$\forall i = 1 \ldots m$
$A_i, O(n_1)$ states
$B_i, O(n_2)$ states

$\Longrightarrow$ [Yu '00]

$\forall i = 1 \ldots m$
DFA M_i
$L(M_i) = L(A_i)L(B_i)$
$O(n_1 n_2)$ states

$\Longrightarrow$

DFA M'
$L(M') = \bigcup_{i=1}^{m} L(M_i)$
$poly(n_1, n_2)$ states

Concatenation under Parikh equivalence: proof idea

DFAs A, B
n_1, n_2 states
$L = L(A)L(B)$
$\Sigma = \{a_1, \ldots, a_m\}$

unary

$\forall i = 1 \ldots m$
$A_i, O(n_1)$ states
$B_i, O(n_2)$ states

$\Longrightarrow$ [Yu '00]

$\forall i = 1 \ldots m$
DFA M_i
$L(M_i) = L(A_i)L(B_i)$
$O(n_1 n_2)$ states

$\Longrightarrow$

DFA M'
$L(M') = \bigcup_{i=1}^{m} L(M_i)$
$poly(n_1, n_2)$ states

nonunary

NFA M
$L(M) = L$
$n_1 + n_2$ states

Concatenation under Parikh equivalence: proof idea

DFAs A, B
n_1, n_2 states
$L = L(A)L(B)$
$\Sigma = \{a_1, \ldots, a_m\}$

unary

$\forall i = 1 \ldots m$
$A_i, O(n_1)$ states
$B_i, O(n_2)$ states

$\Longrightarrow$ [Yu '00]

$\forall i = 1 \ldots m$
DFA M_i
$L(M_i) = L(A_i)L(B_i)$
$O(n_1 n_2)$ states

$\Longrightarrow$

DFA M'
$L(M') = \bigcup_{i=1}^{m} L(M_i)$
$poly(n_1, n_2)$ states

nonunary

NFA M
$L(M) = L$
$n_1 + n_2$ states

$\Longrightarrow$

NFA M_0
$L \setminus L(M')$
$(n_1+n_2)(m+1)+1$ states

Concatenation under Parikh equivalence: proof idea

DFAs A, B
n_1, n_2 states
$L = L(A)L(B)$
$\Sigma = \{a_1, \ldots, a_m\}$

unary

$\forall i = 1 \ldots m$
$A_i, O(n_1)$ states
$B_i, O(n_2)$ states

$\Longrightarrow$ [Yu '00]

$\forall i = 1 \ldots m$
DFA M_i
$L(M_i) = L(A_i)L(B_i)$
$O(n_1 n_2)$ states

$\Longrightarrow$

DFA M'
$L(M') = \bigcup_{i=1}^{m} L(M_i)$
$poly(n_1, n_2)$ states

nonunary

NFA M
$L(M) = L$
$n_1 + n_2$ states

$\Longrightarrow$

NFA M_0
$L \setminus L(M')$
$(n_1+n_2)(m+1)+1$ states

Parikh equivalent conversion $\Longrightarrow_\pi$

DFA M'_0
$poly(n_1, n_2)$ states

Concatenation under Parikh equivalence: proof idea

DFAs A, B
n_1, n_2 states
$L = L(A)L(B)$
$\Sigma = \{a_1, \ldots, a_m\}$

unary

$\forall i = 1 \ldots m$
$A_i, O(n_1)$ states
$B_i, O(n_2)$ states

$\Longrightarrow$ [Yu '00]

$\forall i = 1 \ldots m$
DFA M_i
$L(M_i) = L(A_i)L(B_i)$
$O(n_1 n_2)$ states

$\Longrightarrow$

DFA M'
$L(M') = \bigcup_{i=1}^{m} L(M_i)$
$poly(n_1, n_2)$ states

nonunary

NFA M
$L(M) = L$
$n_1 + n_2$ states

$\Longrightarrow$

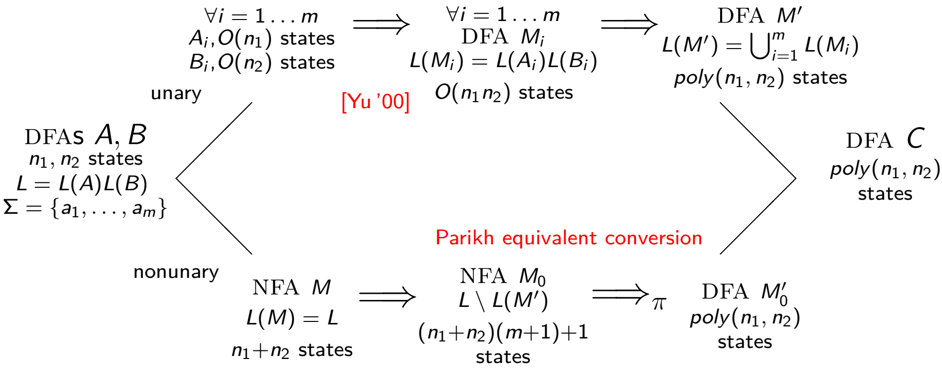
NFA M_0
$L \setminus L(M')$
$(n_1 + n_2)(m + 1) + 1$
states

$\Longrightarrow_{\pi}$ Parikh equivalent conversion

DFA M'_0
$poly(n_1, n_2)$
states

DFA C
$poly(n_1, n_2)$
states

Concatenation under Parikh equivalence: proof idea



Theorem

Given two DFAs A and B of n_1 and n_2 states, respectively, there exists a DFA of polynomial number of states in n_1 and n_2 that is Parikh equivalent to $L(A)L(B)$. Moeover, this cost is tight.

Projection under Parikh equivalence

Given a word $w \in \Sigma^*$, the *projection* of w over an alphabet $\Sigma' \subseteq \Sigma$, is the word $P_{\Sigma'}(w)$ obtained by removing from w all the symbols which are not in Σ'. (see, e.g., [Jirásková & Masopust 12]).

Example:

$$P_{\{a,b\}}(a^n b^n c^n) = a^n b^n$$

Projection under Parikh equivalence

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

DFA A		NFA A'		DFA M
$L(A)$	$\Longrightarrow$	$L(A') = P_{\Sigma'}(L(A))$	$\Longrightarrow_\pi$	$L(M) =_\pi L(A')$
n states		n states		$e^{O(\sqrt{n \cdot \ln n})}$ states

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]

Intersection does not commute 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 does not commute with Parikh mapping

$\psi((a^*b^*)^c) \neq (\psi(a^*b^*))^c$ holds; in fact,

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

Problem: intersection

A, B DFAs
n_1, n_2 states

$\Longrightarrow$

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

Problem: complement (left open!)

A DFA
n states

$\Longrightarrow$

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

Intersection: revisited

We use a modification of the following result:

Theorem ([Kopczyński&To '10])

There is a polynomial p such that for each n-state NFA A *over* $\Sigma = \{a_1, \ldots, a_m\}$,

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

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* $\{0, 1, \ldots, n\}^m$

Intersection: revisited

Theorem

Let A, B be DFAs with respectively n_1, n_2 states over $\Sigma = \{a_1, \ldots, a_m\}$. 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:

- $n = \max\{n_1, n_2\}(m+1) + 1$
- $p(n) = O(n^{3m^2} m^{m^2/2+2})$

Proof.

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

Conclusion

Under Parikh equivalence:

- For $\cup$, $\cdot$, *, ^{c}, $\cap$, $\sqcup\!\sqcup$, and ^{R}, we obtain a polynomial state complexity, in contrast to the intrinsic exponential state complexity in the classical equivalence.
- For P_{Σ_0} we prove a superpolynomial state complexity, which is lower than the exponential one of the corresponding classical operation.
- For each two deterministic automata A and B, it is possible to obtain a deterministic automaton with a polynomial number of states, whose accepted language has as Parikh image $\psi(L(A)) \cap \psi(L(B))$.

Conclusion

Under Parikh equivalence:

- For $\cup$, $\cdot$, *, ^{c}, $\cap$, $\sqcup\!\sqcup$, and ^{R}, we obtain a polynomial state complexity, in contrast to the intrinsic exponential state complexity in the classical equivalence.
- For P_{Σ_0} we prove a superpolynomial state complexity, which is lower than the exponential one of the corresponding classical operation.
- For each two deterministic automata A and B, it is possible to obtain a deterministic automaton with a polynomial number of states, whose accepted language has as Parikh image $\psi(L(A)) \cap \psi(L(B))$.

Conclusion

Under Parikh equivalence:

- For $\cup$, $\cdot$, *, ^{c}, $\cap$, $\sqcup\!\sqcup$, and ^{R}, we obtain a polynomial state complexity, in contrast to the intrinsic exponential state complexity in the classical equivalence.
- For P_{Σ_0} we prove a superpolynomial state complexity, which is lower than the exponential one of the corresponding classical operation.
- For each two deterministic automata A and B, it is possible to obtain a deterministic automaton with a polynomial number of states, whose accepted language has as Parikh image $\psi(L(A)) \cap \psi(L(B))$.

Thank you for your attention