

Operational State Complexity under Parikh Equivalence*

(Extended Abstract)

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

¹ Dipartimento di Informatica, Università degli Studi di Milano, Italy
{lavado,pighizzini}@di.unimi.it**

² Helsinki Institute for Information Technology (HIIT)

³ Department of Information and Computer Science, Aalto University,
P. O. Box 15400, FI-00076, Aalto, Finland
shinnosuke.seki@aalto.fi***

Abstract. We investigate, under Parikh equivalence, the state complexity of some language operations which preserve regularity. For union, concatenation, Kleene star, complement, intersection, shuffle, and reversal, we obtain a polynomial state complexity over any fixed alphabet, in contrast to the intrinsic exponential state complexity of some of these operations in the classical version. For projection we prove a superpolynomial state complexity, which is lower than the exponential one of the corresponding classical operation. We also prove that 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 the intersection of the Parikh images of the languages accepted by A and B .

1 Introduction

The investigation of the state complexity of regular languages and their operations is extensively reported in the literature (e.g., [11,14,15]). In a previous work [9], we proposed to extend that investigation by considering the classical notion of Parikh equivalence [10], which has been extensively studied in the literature (e.g., [1,6]) even for the connections with semilinear sets [7] and with other fields such as Presburger Arithmetics [5], Petri Nets [3], logical formulas [13], and formal verification [12]. We remind the reader that two words over a same alphabet Σ are *Parikh equivalent* if and only if they are equal up to a permutation

* Paper accepted at the *16th International Workshop on Descriptive Complexity of Formal Systems*. In H. Jürgensen et al. (Eds.): DCFs 2014, LNCS 8614, pp. 294-305. Springer (2014)

** Their works are partially supported by MIUR under the project PRIN “Automati e Linguaggi Formali: Aspetti Matematici e Applicativi” H41J12000190001.

*** His work is supported in part by Academy of Finland, Postdoctoral Researcher Grant No. 13266670/T30606.

of their symbols or, equivalently, for each letter $a \in \Sigma$, the number of occurrences of a in the two words is the same (the vector $\psi(w)$ consisting of these numbers is also called *Parikh image* of a word $w \in \Sigma^*$). This notion extends in a natural way to languages (two languages L_1 and L_2 are *Parikh equivalent*, that is $\psi(L_1) = \psi(L_2)$, when for each word in L_1 there is a Parikh equivalent word in L_2 and vice versa) and to formal systems which are used to specify languages as, for instance, grammars and automata. A well-known result by Parikh states that context-free and regular languages are indistinguishable under Parikh equivalence [10]. More precisely, the Parikh image of a context-free language is a semilinear set and from each semilinear set a Parikh equivalent automaton can be immediately obtained.

In particular, in [9] we treated the conversion of one-way nondeterministic finite automata (NFAs) into Parikh equivalent one-way deterministic finite automata (DFAs). We proved that the state cost of this conversion is smaller than the exponential cost of the classical conversion. In fact, we showed that from each n -state NFA we can build a Parikh equivalent DFA with $e^{O(\sqrt{n \cdot \ln n})}$ states. Furthermore, this cost is tight. Quite surprisingly, this cost is due to the unary words in the language, i.e., to the words consisting only of occurrences of a same symbol. In fact, if the given NFA accepts only words containing at least two different letters then the cost reduces to a polynomial.

Motivated by the interest in regular languages, here we continue the same line of research by considering basic operations on regular languages and on DFAs. We reformulate under Parikh equivalence some classical questions on the state complexity of operations as, for instance, the following: given two arbitrary DFAs A and B of n_1 and n_2 states, respectively, how many states are sufficient and necessary in the worst case (as a function of n_1 and n_2) for a DFA to accept the concatenation of the languages accepted by A and B ? For this question an exponential cost is known [15]. Using our above mentioned bound on the conversion of NFAs into Parikh equivalent DFAs, this exponential bound can be reduced, under Parikh equivalence, to a superpolynomial upper bound. In this paper we further reduce it to a polynomial, namely we show that there exists a DFA with a number of states polynomial in n_1 and n_2 accepting a language that is Parikh equivalent to the concatenation of the languages accepted by A and B . We obtain a similar result for the Kleene star operation while, for the union, the cost is polynomial even in the classical case. We also present results for other operations as intersection, complement, reversal, shuffle and projection.

Concerning intersection and complement, we observe that these operations do not commute with Parikh image, e.g., the Parikh image of the complement of a language L does not necessarily coincide with the complement of the Parikh image of L . However, semilinear sets are closed under intersection and complement [4]. Hence, we can formulate state complexity questions about intersections and complements of Parikh images of languages accepted by given DFAs. We solve the question for the intersection by proving, in a constructive way, that for each two DFAs there exists a DFA of polynomial size accepting a language whose Parikh image is the intersection of the Parikh images of the languages accepted

by the two given DFAs, while the analogous question for the complement will be the subject of future investigations.

2 Regular Operations under Parikh Equivalence

In this section, we consider problems in the following general form:

Problem 1. For DFAs A and B of n_1 and n_2 states, respectively, solve 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 $g(L(A), L(B))$?

Inspecting various regular operations we obtain the following results.

Theorem 2. *Let A and B be two DFAs with n_1 and n_2 states, respectively. Then:*

1. *There exist two DFAs both with n_1n_2 states that accept languages (Parikh) equivalent to $L(A) \cup L(B)$ and to $L(A) \cap L(B)$.*
2. *There exists a DFA with n_1 states that accepts a language (Parikh) equivalent to the complement of $L(A)$.*
3. *There exists a DFA with a number of states polynomial in n_1 and n_2 accepting a language Parikh equivalent to the concatenation and to the shuffle of $L(A)$ and $L(B)$.*
4. *There exists a DFA with a number of states polynomial in n_1 accepting a language Parikh equivalent to $L(A)^*$.*
5. *There exists a DFA with n_1 states that accepts a language Parikh equivalent to the reversal of $L(A)$.*
6. *There exists a DFA with $e^{O(\sqrt{n_1 \cdot \ln n_1})}$ states that accepts a language Parikh equivalent to the projection $P_{\Sigma'}(L(A))$ of $L(A)$ over $\Sigma' \subseteq \Sigma$.*

All these bounds are asymptotically tight.

We shortly comment the above results.

The state complexity of union and intersection is in the low order n_1n_2 even in the conventional sense over both unary and nonunary alphabets. Moreover, it is known to be tight already over a unary alphabet [14]. Similar considerations hold for the complement.

Unlike union or intersection, both concatenation and star are known to cost an exponential number of states on DFAs. In fact, the number of states which is necessary and sufficient in the worst case for a DFA to accept the concatenation of an n_1 -state DFA language and an n_2 -state DFA language over a binary alphabet is $(2n_1 - 1)2^{n_2 - 1}$ [15]; over a unary alphabet, the cost decreases to n_1n_2 [14].

As for star of an n -state DFA language, the tight bound is $2^{n-1} + 2^{n-2}$ over a binary alphabet, whereas it is $(n-1)^2 + 1$ over a unary alphabet [15].

For concatenation, we could first build an NFA with $n_1 + n_2$ states and then according to the superpolynomial conversion of NFAs into Parikh equivalent DFAs presented in [9] we could convert it into a Parikh equivalent DFA with a superpolynomial numbers of states. However, we give an *ad hoc* construction which produces a DFA with a polynomial number of states. We do the same for star. In the conventional sense, shuffle involves the exponential cost $2^{n_1 n_2} - 1$ and this bound is tight [2]. Since the Parikh image of the shuffle of two languages is equal to that of their concatenation, the cost for the shuffle under Parikh equivalence is the same as for concatenation.

Reversal is also expensive for DFAs. In fact, the tight bound 2^n is known for reversal [15]. Under Parikh equivalence, however, nothing need be said since Parikh image is invariant under this operation.

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' . We can extend this notion to languages in a standard way. It is easy to see that projection preserves regularity. However, transforming a DFA A of n states into a DFA for the projection can require a number of states that is exponential in n [8]. Even in this case, the bound can be reduced if we want to obtain a Parikh equivalent DFA: from A we can obtain an NFA of n states for the projection, and then we can transform it into a Parikh equivalent DFA of $e^{O(\sqrt{n \cdot \ln n})}$ states. By using a projection over a unary alphabet we can show that this bound cannot be reduced.

3 Intersection and Complement, Revisited

We consider one more time the intersection and the complement. In fact, the noncommutativity of those operations with the Parikh mapping brings us a second problem of interest. The noncommutativity in the case of intersection is illustrated in the inequality $\psi(a^+b^+ \cap b^+a^+) \neq \psi(a^+b^+) \cap \psi(b^+a^+)$; the left-hand side is the empty set, while the right-hand side is the linear set $\mathbb{N} \times \mathbb{N}$. In the case of complement the reader may consider the language $(ab)^*$.

Note that each of the other operations examined so far is either commutative with the Parikh mapping (i.e., union and projection) or not defined naturally over the set of nonnegative integer vectors (i.e., concatenation, star, shuffle, and reversal). The problem of interest asks: given two DFAs A and B of n_1 and n_2 states, respectively, how small can we make a DFA whose Parikh image is equal to $\psi(L(A)) \cap \psi(L(B))$? We can formulate a similar problem in case of the complement. The fact that the Parikh image of a language accepted by an NFA is semilinear and the closure property of semilinear sets under intersection and complement [4] makes these problems meaningful. We solve the problem for intersection, leaving the one for complement for future investigations.

Over a unary alphabet, the problem is degenerated into the problem addressed in Theorem 2(1) because over such an alphabet, intersection commutes

with the Parikh mapping. Therefore, in the following, we examine the problem over a nonunary alphabet, and solve it by showing that a polynomial number of states in n_1 and n_2 are sufficient. The proof consists of revisiting Ginsburg and Spanier's proof [4,7] of the closure property of semilinear sets under intersection with a careful analysis of the size of the resulting semilinear set.

Theorem 3. *Given two DFAs A and B with n_1 and n_2 states, respectively, there exists a DFA of a polynomial number of states in n_1 and n_2 whose Parikh image is equal to $\psi(L(A)) \cap \psi(L(B))$.*

References

1. Aceto, L., Ésik, Z., Ingólfssdóttir, A.: A fully equational proof of Parikh's theorem. *RAIRO-Theor. Inf. Appl.* 36(2), 129–153 (2002)
2. Câmpeanu, C., Salomaa, K., Yu, S.: Tight lower bound for the state complexity of shuffle of regular languages. *Journal of Automata, Languages and Combinatorics* 7(3), 303–310 (2002)
3. Esparza, J.: Petri nets, commutative context-free grammars, and basic parallel processes. *Fund. Inform.* 31(1), 13–25 (1997)
4. Ginsburg, S., Spanier, E.H.: Bounded ALGOL-like languages. *T. Am. Math. Soc.* 113, 333–368 (1964)
5. Ginsburg, S., Spanier, E.H.: Semigroups, Presburger formulas, and languages. *Pac. J. Math.* 16(2), 285–296 (1966)
6. Goldstine, J.: A simplified proof of Parikh's theorem. *Discrete Math.* 19(3), 235 – 239 (1977)
7. Huynh, D.T.: The complexity of semilinear sets. In: *Automata, Languages and Programming, LNCS*, vol. 85, pp. 324–337. Springer (1980)
8. Jirásková, G., Masopust, T.: On a structural property in the state complexity of projected regular languages. *Theor. Comput. Sci.* 449, 93–105 (2012)
9. Lavado, G.J., Pighizzini, G., Seki, S.: Converting nondeterministic automata and context-free grammars into Parikh equivalent one-way and two-way deterministic automata. *Inform. Comput.* 228, 1–15 (2013)
10. Parikh, R.J.: On context-free grammars. *J. ACM* 13(4), 570–581 (1966)
11. Pighizzini, G., Shallit, J.: Unary language operations, state complexity and Jacobsthal's function. *Int. J. Found. Comput. S.* 13(1), 145–159 (2002)
12. To, A.W.: *Model Checking Infinite-State Systems: Generic and Specific Approaches*. Ph.D. thesis, School of Informatics, University of Edinburgh (August 2010)
13. Verma, K., Seidl, H., Schwentick, T.: On the complexity of equational Horn clauses. In: *CADE 2005. LNCS*, vol. 3632, pp. 337–352. Springer (2005)
14. Yu, S.: State complexity of regular languages. *Journal of Automata, Languages and Combinatorics* 6, 221–234 (2000)
15. Yu, S., Zhuang, Q., Salomaa, K.: The state complexity of some basic operations on regular languages. *Theor. Comput. Sci.* 125, 315–328 (1994)