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

(Extended Abstract)

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

[1] Dipartimento di Informatica, Università degli Studi di Milano
via Comelico 39, I-20135, Milano, Italy
giovanna.lavado@unimi.it
giovanni.pighizzini@unimi.it
[2] Department of Information and Computer Science, Aalto University,
P.O. Box 15400, FI-00076, Aalto, Finland
shinnosuke.seki@aalto.fi

Abstract. We investigate the conversion of nondeterministic finite automata and context-free grammars into Parikh equivalent deterministic finite automata, from a descriptional complexity point of view.
We prove that for each nondeterministic automaton with n states there exists a Parikh equivalent deterministic automaton with $e^{O(\sqrt{n \cdot \ln n})}$ states. Furthermore, this cost is tight. In contrast, if all the strings accepted by the given automaton contain at least two different letters, then a Parikh equivalent deterministic automaton with a polynomial number of states can be found.
Concerning context-free grammars, we prove that for each grammar in Chomsky normal form with n variables there exists a Parikh equivalent deterministic automaton with $2^{O(n^2)}$ states. Even this bound is tight.

1 Introduction

It is well-known that the state cost of the conversion of nondeterministic finite automata (NFAs) into equivalent deterministic finite automata (DFAs) is exponential: using the classical subset construction [10], from each n-state NFA we can build an equivalent DFA with 2^n states. Furthermore, this cost cannot be reduced.

In all examples witnessing such a state gap (e.g., [5–7]), input alphabets with at least two letters and proof arguments strongly relying on the structure of strings are used. As a matter of fact, for the unary case, namely the case of the one letter input alphabet, the cost reduces to $e^{\Theta(\sqrt{n \cdot \ln n})}$, as shown by Chrobak [1].

* Paper accepted at the *16th International Conference on Developments in Language Theory*. In H.–C. Yen and O.H. Ibarra (Eds.): DLT 2012, LNCS 7410, pp. 284–295. Springer (2012)

What happens if we do not care of the order of symbols in the strings, i.e., if we are interested only in obtaining a DFA accepting a set of strings which are equal, after permuting the symbols, to the strings accepted by the given NFA?

This question is related to the well-known notions of Parikh image and Parikh equivalence [8]. Two strings over a same alphabet Σ are Parikh equivalent if and only if they are equal up to a permutation of their symbols or, equivalently, for each letter $a \in \Sigma$ the number of occurrences of a in the two strings is the same. This notion extends in a natural way to languages (two languages L_1 and L_2 are Parikh equivalent when for each string in L_1 there is a Parikh equivalent string in L_2 and vice versa) and to formal systems which are used to specify languages as, for instance, grammars and automata. Notice that in the unary case Parikh equivalence is just the standard equivalence. So, in the unary case, the answer to our previous question is given by the above mentioned result by Chrobak.

Our first contribution in this paper is an answer to that question in the general case. In particular, we prove that the state cost of the conversion of n-state NFAs into Parikh equivalent DFAs is the same as in the unary case, i.e., it is $e^{\Theta(\sqrt{n \cdot \ln n})}$. More surprisingly, we prove that this is due to the unary parts of languages. In fact, we show that if the given NFA accepts only nonunary strings, i.e., each accepted string contains at least two different letters, then we can obtain a Parikh equivalent DFA with a polynomial number of states in n. Hence, while in standard determinization the most difficult part (with respect to the state complexity) is the nonunary one, in the "Parikh determinization" this part becomes easy and the most complex part is the unary one.

In the second part of the paper we consider context-free grammars (CFGs). Parikh Theorem [8] states that each context-free language is Parikh equivalent to a regular language. We study this equivalence from a descriptional complexity point of view. Recently, Esparza, Ganty, Kiefer, and Luttenberger proved that each context-free grammar in Chomsky normal form (CNFG) with h variables can be converted into a Parikh equivalent NFA with $O(4^h)$ states [2]. In [4] it was proven that if G generates a bounded language then we can obtain a DFA with $2^{h^{O(1)}}$ states, i.e., a number exponential in a polynomial of the number of variables. In this paper, we are able to extend such a result by removing the restriction to bounded languages. We also reduce the upper bound to $2^{O(h^2)}$. A milestone for obtaining such a result is the conversion of NFAs to Parikh equivalent DFAs presented in the first part of the paper. By suitably combining that result (in particular the polynomial conversion in the case of NFAs accepting nonunary strings) with the above mentioned result from [2] and with a result by Pighizzini, Shallit, and Wang [9] concerning the unary case, we prove that each context-free grammar in Chomsky normal form with h variables can be converted into a Parikh equivalent DFA with $2^{O(h^2)}$ states. From the results concerning the unary case, it follows that this bound is tight.

Even for this simulation, as for that of NFAs by Parikh equivalent DFAs, the main contribution to the state complexity of the resulting automaton is given by the unary part.

2 From NFAs to Parikh equivalent DFAs

In this section we present our first main contribution. From each n-state NFA A we derive a Parikh equivalent DFA A' with $e^{O(\sqrt{n \cdot \ln n})}$ states. Furthermore, we prove that this cost is tight.

Actually, as a preliminary step we obtain a result which is interesting *per se* (Theorem 1): if each string accepted by the given NFA A contains at least two different symbols, i.e., it is nonunary, then the Parikh equivalent DFA A' can be obtained with polynomially many states. Hence, the superpolynomial blowup is due to the unary part of the accepted language.

The proof of Theorem 1 gives a construction which uses a normal form for the Parikh image of the languages accepted by NFAs. Such a form is a refinement of a form presented in [3, 11].

Theorem 1. *For each n-state* NFA *accepting a language none of whose words are unary, there exists a Parikh equivalent* DFA *with a number of states polynomial in n.*

For the unary part, the following result proved by Chrobak in 1986 is useful.

Theorem 2 ([1]). *The state cost of the conversion of n-state unary* NFA*s into equivalent* DFA*s is $e^{\Theta(\sqrt{n \cdot \ln n})}$.*

Theorem 1 and Theorem 2 are useful to study the general case. From a given n-state NFA A with input alphabet $\Sigma = \{a_1, a_2, \ldots, a_m\}$, for each $i = 1, \ldots, m$, we first build an n-state NFA A_i accepting the unary language $L(A) \cap a_i^*$. Using Theorem 2, we convert A_i into an equivalent DFA A'_i with $e^{O(\sqrt{n \cdot \ln n})}$ states. We can also build an $O(n)$-state NFA A_0 accepting all the nonunary strings belonging to $L(A)$. The NFA A_0 can be converted into a Parikh equivalent DFA A_n with a number of states polynomial in n. Using standard constructions, we combine DFAs $A'_1, \ldots, A'_m$ and A_n to finally obtain a DFA accepting a language Parikh equivalent to the language accepted by the original NFA A and with a number of states polynomial in n.

From this argument and from the optimality of the upper bound for the unary case (Theorem 2) we obtain the following result.

Theorem 3. *For each n-state* NFA*, there exists a Parikh equivalent* DFA *with $e^{O(\sqrt{n \cdot \ln n})}$ states. Furthermore, this cost is tight.*

3 From CFGs to Parikh Equivalent DFAs

In this section we extend the results of Section 2 to the conversion of CFGs in Chomsky normal form to Parikh equivalent DFAs. Actually, Theorem 1 will play an important role in order to obtain the main result of this section.

Even in this case the proof is given by splitting the unary and the nonunary parts of the language under consideration, converting the corresponding grammars into Parikh equivalent DFAs and, finally, recombining the DFAs so obtained into a DFA.

For the unary part, the conversion is done by using a result from [9] stating that for any CNFG with h variables that generates a unary language, there exists an equivalent DFA with less than 2^{h^2} states.

For the nonunary part, we first use a result from [2] stating that for a CNFG with h variables there exists a Parikh equivalent NFA with $O(4^h)$ variables. Then, we apply the construction used to prove Theorem 1 to the resulting NFA.

Theorem 4. *For any* CNFG *with h variables, there exists a Parikh equivalent* DFA *with at most $2^{O(h^2)}$ states.*

We finally observe that in [9] it was proven that there is a constant $c > 0$ such that for infinitely many $h > 0$ there exists a CNFG with h variables generating a unary language such that each equivalent DFA requires at least 2^{ch^2} states. This implies that the upper bound given in Theorem 4 cannot be improved.

References

1. Chrobak, M.: Finite automata and unary languages. Theoretical Computer Science 47, 149–158 (1986), corrigendum, ibid. 302 (2003) 497-498
2. Esparza, J., Ganty, P., Kiefer, S., Luttenberger, M.: Parikh's theorem: A simple and direct automaton construction. Information Processing Letters 111(12), 614–619 (2011)
3. Kopczyński, E., To, A.W.: Parikh images of grammars: Complexity and applications. In: Symposium on Login in Computer Science. pp. 80–89 (2010)
4. Lavado, G.J., Pighizzini, G.: Parikh's theorem and descriptional complexity. In: Proceedings of SOFSEM 2012. LNCS, vol. 7147, pp. 361–372. Springer (2012)
5. Lupanov, O.: A comparison of two types of finite automata. Problemy Kibernet 9, 321–326 (1963), (in Russian). German translation: Über den Vergleich zweier Typen endlicher Quellen, Probleme der Kybernetik 6, 329–335 (1966)
6. Meyer, A.R., Fischer, M.J.: Economy of description by automata, grammars, and formal systems. In: FOCS. pp. 188–191. IEEE (1971)
7. Moore, F.: On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic, and two-way finite automata. IEEE Transactions on Computers C-20(10), 1211–1214 (1971)
8. Parikh, R.J.: On context-free languages. Journal of the ACM 13(4), 570–581 (1966)
9. Pighizzini, G., Shallit, J., Wang, M.: Unary context-free grammars and pushdown automata, descriptional complexity and auxiliary space lower bounds. Journal of Computer and System Sciences 65(2), 393–414 (2002)
10. Rabin, M., Scott, D.: Finite automata and their decision problems. IBM J. Res. Develop. 3, 114–125 (1959)
11. To, A.W.: Parikh images of regular languages: Complexity and applications (February 2010), arXiv:1002.1464v2