A NOTE ON THE NUMBER OF TRANSITIONS OF NONDETERMINISTIC FINITE AUTOMATA

Hermann Gruber and Markus Holzer
Institut für Informatik, Technische Universität München,
Boltzmannstraße 3, D-85748 Garching bei München, Germany
e-mail: {gruberh,holzer}@in.tum.de

ABSTRACT

Finite automata are among he most fundamental computational models in theoretical computer science. The vast variety of possible practical applications led to a widespread use of this concept, and gives an additional incentive for doing research on this topic. During the last decade, the investigation of questions regarding descriptional complexity has attracted the interest of many researchers; see, e.g., [1, 4, 6]. One measure for the economy of description by automata seem to dominate this area of research, namely state complexity, which equals the minimal number of states needed by a deterministic or nondeterministic finite automaton to accept a regular language. Let $sc(L)$ ($nsc(L)$, respectively) denote the deterministic (nondeterministic, respectively) state complexity of the regular language L. To our knowledge only few results are known with respect to transition complexity—see, e.g., the recent publications [2, 5]. Here, transition complexity is analogously defined as state complexity and we abbreviate the deterministic (nondeterministic, respectively) transition complexity of a regular language L by $tc(L)$ ($ntc(L)$, respectively). To be more precise, for a nondeterministic finite automaton with transition function $\delta : Q \times \Sigma \to 2^Q$, where Q is the finite set of states, Σ the finite set of input symbols, and 2^Q the power set of Q, the number of transitions equals $|\{(q, a, p) \mid p \in \delta(q, a)\}|$. This naturally extends to deterministic finite automata. Observe, that only non-blocking transitions are counted; a transition is *blocking*, if $\delta(q, a) = \emptyset$, for some $q \in Q$ and $a \in \Sigma$.

Obviously, a deterministic finite automaton with n states and input alphabet Σ has exactly $|\Sigma| \cdot n$ transitions, because every state has at most $|\Sigma|$ transitions leaving it. Moreover, it is easy to see that for deterministic finite automata the state minimal finite automaton is also transition minimal. For nondeterministic finite automata, a pessimistic estimate shows that the overall number of transitions is at most $|\Sigma| \cdot n^2$, since every state has at most $|\Sigma| \cdot n$ transitions leaving it. In contrast to the deterministic case, here it turns out, that state and transition minimization cannot be carried out simultaneously. To this end we define the following (infinite) languages

$$L_n = \{\, 0^r 10^s 10^t \mid 0 \leq r, t < n \text{ and } s > 0 \,\} \cup \{\, 0^r 110^r \mid 0 \leq r < n \,\}$$

for $n \geq 1$. The result on the languages L_n w.r.t. state and transition complexity read as follows:

Theorem 1 *Let $n \geq 1$. There is a unique (up to the renaming of states) state minimal nondeterministic finite automaton accepting the language L_n which has $3n$ states and $n^2 + 4n - 2$ transitions. Moreover, $4n + 1$ states are sufficient for a deterministic finite automaton to accept the language L_n and this finite automaton induces that the nondeterministic transition complexity is less than or equal to $6n + 1$.*

In other words, $ntc(L_n) \leq 6n+1$ and $3n = nsc(L_n) \leq sc(L_n) \leq 4n+1$. Moreover, by increasing the number of states by $n+1$ decreases the number of transitions from a quadratic to a linear polynomial. Hence, in this case state minimization for nondeterministic finite automata leads to an undesirable complexity w.r.t. the number of transitions. A similar result can be obtained for the finite languages

$$L'_n = \{\, 0^i 10^j 10^k \mid i+k+j < n \text{ and } k > 0 \,\} \cup \{\, 0^i 110^k \mid i < n \text{ and } k = n-i-1 \,\}.$$

It is well known that minimization (state or transition) for deterministic finite automata can be carried out in polynomial time. On the other hand, it is **PSPACE**-complete to find a nondeterministic finite automaton that is state or transition minimal. Recently it was shown in [2] that even finding an approximately transition minimal nondeterministic finite automaton is hard to approximate unless $\mathbf{P} = \mathbf{PSPACE}$. In [2] also the case of unary languages is covered, for which the analogous question was shown to be **coNP**-hard. We complete the picture by giving an approximation hardness results for the case of finite languages. Observe, that the upper bound on the complexity of the stated problem is $\Sigma_2^{\mathbf{P}}$, the second level of the Polynomial Hierarchy. Our non-approximation result is based on a reduction from a variant of the **NP**-complete SAT-problem to the problem under consideration. A Boolean assignment, is encoded as a binary string of length n, and is accepted by the finite automaton A, if the assignment does *not* satisfy the formula. Therefore, if the formula is not satisfiable, then all strings of length n are accepted by A, and hence $nsc(L(A)) = n+1$ and $ntc(L(A)) = 2n$. On the other hand, by carefully choosing the variable order, we ensure that the nondeterministic state and transition complexity of the language L rises sharply. This allows us to prove the following result.

Theorem 2 *Unless* $\mathbf{P} = \mathbf{NP}$*, it is impossible to efficiently approximate the nondeterministic transition complexity of the finite language* $L(A)$ *within an approximation factor of* $c < 3/2$ *when given a acyclic nondeterministic finite automaton* A*. A similar statement is valid for nondeterministic state minimization and a factor* $c < 2$*.*

We can improve the previous statement, by a more involved construction, to the case where the input is specified by a deterministic automaton, and more importantly to a factor t^ϵ, for some $\epsilon > 0$, where t is the number of transitions of the given finite automaton. This answers an open question stated in [2].

References

[1] J. Goldstine, M. Kappes, C. M. R. Kintala, H. Leung, A. Malcher, and D. Wotschke. Descriptional complexity of machines with limited resources. *Journal of Universal Computer Science*, 8(2):193–234, 2002.

[2] G. Gramlich and G. Schnitger. Minimizing NFA's and regular expressions. In V. Diekert and B. Durand, editors, 22*nd Annual Symposium on Theoretical Aspects of Computer Science*, number 3404 in LNCS, pages 399–411, Stuttgart, Germany, February 2005, Springer.

[3] J. Hromkovič. Descriptional complexity of finite automata: Concepts and open problems. *Journal of Automata, Languages and Combinatorics*, 7(4):519–531, 2002.

[4] J. Hromkovič, S. Seibert, and T. Wilke. Translating regular expressions into small ϵ-free automata. *Journal on Computers and Systems Sciences*, 62:565–588, 2001.

[5] Y. Lifshits. A lower bound on the size of ϵ-free NFA corresponding to a regular expression. *Information Processing Letters*, 85(6):293–299, 2003.

[6] S. Yu. State complexity of regular languages. *Journal of Automata, Languages and Combinatorics*, 6(2):221–234, 2001.