RAIRO-Inf. Theor. Appl. **41** (2007) 261–265
DOI: 10.1051/ita:2007017

AN UPPER BOUND FOR TRANSFORMING SELF-VERIFYING AUTOMATA INTO DETERMINISTIC ONES

Ira Assent[1] and Sebastian Seibert[2]

Abstract. This paper describes a modification of the *power set construction* for the transformation of self-verifying nondeterministic finite automata to deterministic ones. Using a set counting argument, the upper bound for this transformation can be lowered from 2^n to $O(\frac{2^n}{\sqrt{n}})$.

Mathematics Subject Classification. 68Q10, 68Q19.

1. Introduction

One of the fundamental research problems of current theoretical computer science is the investigation of the computational power of nondeterministic and randomized computations, especially in comparison with deterministic ones. A subarea where some progress has been made is the study of finite automata w.r.t. their descriptional complexity.

In this note, we study the transformation from self-verifying nondeterministic to deterministic finite automata. From the study of nondeterministic finite automata, we know that nondeterminism can be used to effectively accept regular languages by always "guessing" the correct decision to be taken. However, this means in practice, that from the fact that such an automaton recognizes a certain language, we can only derive that there exists at least one sequence of states corresponding to the input word which leads to an accepting computation. Whenever the automaton

Keywords and phrases. Self-verifying nondeterministic automata, descriptional complexity, power set construction.

[1] RWTH Aachen, Lehrstuhl für Informatik IX, Ahornstr. 55, 52074 Aachen, Germany; assent@informatik.rwth-aachen.de

[2] ETH Zürich, Department Informatik, Informationstechnologie und Ausbildung, ETH Zentrum, 8092 Zürich, Switzerland; sseibert@inf.ethz.ch

Supported under SNF project 200021-107327/1 "Computational power of randomization and nondeterminism"

does not accept a word, this means that there exists no such computation among all possible choices which is much harder to verify. This is where self-verifying nondeterministic finite automata come into play (see *e.g.* [2]). They are the natural answer to the call for automata which use nondeterministic strategies to search for a solution, but are also able to provide a definite "no" whenever a word is not in the language accepted.

That is, for all runs of the automaton where it doesn't get to the correct answer, it will never give a wrong answer but rather "I do not know", *i.e.* it will halt in a so called neutral state.

Definition 1.1. A self-verifying finite automaton (SVFA) $A = (Q, \Sigma, \delta, q_0, F, R)$ consists of:

- a finite state set Q;
- a finite alphabet Σ;
- a transition function $\delta \to 2^Q$;
- an initial state q_0;
- two disjoint sets $F, R \subset Q$ of final and rejecting states, respectively.

The remaining states $Q \setminus (F \cup R)$ are called neutral.
Additionally, we demand

(i) no input word w has both computations finishing in accepting states and finishing in rejecting states;
(ii) each input word w has at least one computation finishing in an accepting or a rejecting state.

An input word w is accepted if there is at least one computation on w finishing in an accepting state, and it is rejected if there is at least one computation on w finishing in a rejecting state.

For deterministic and nondeterministic finite automata (DFA, and NFA, respectively) $A = (Q, \Sigma, \delta, q_0, F)$ we use the standard definition (see [1] for example).

As we can see, self-verifying finite automata always give an "answer" to the question whether or not a word is in the language. This leads to an interesting property, in that they can immediately be used for recognition of the complement language by just interchanging the sets F and R. So we have always automata of the same minimal size for a language and its complement, an important difference to the case of conventional nondeterministic automata.

Immediately, this leads to the question how the SVFA relate in size to the known models of DFA and NFA. In [2], Hromkovič and Schnitger observed that the size of a minimal SVFA is at most 1 plus the sum of the sizes of a minimal NFA for the same language and the size of a minimal NFA for the complement. As a consequence, they obtained an example that has an n-state SVFA but only DFA of size $\Omega(2^{\sqrt{n}})$. Thus no polynomially bound transformation from SVFA to DFA can exists, and the question arises whether the blow-up for such a transformation can be as large as from NFA to DFA.

Here, we show that at least SVFA are not as far from DFA as NFA in that the gap in state number that can occur in such a transformation is not n versus 2^n in the worst case but at most n versus $O(\frac{2^n}{\sqrt{n}})$.

2. From self-verifying to deterministic automata

Since SVFA are a variant of NFA, we can easily adapt the power set construction to transform a SVFA to DFA, as will be shown below. A central idea to evidence a lower state number blow-up than in the NFA case, is to show that there are certain combinations of states which exist only in a minimal DFA resulting from NFA but not in a minimal DFA obtained from SVFA. Thus we do not use the entire power set as we do in the unrestricted nondeterministic case, but only a smaller subset of the power set. (It is known that in the NFA to DFA case the bound is tight [3–5].) We therefore use considerably less states and consequently obtain a better upper bound.

Theorem 2.1. *For any n-state SVFA an equivalent DFA can be constructed using at most* $\mathrm{O}\left(\frac{2^n}{\sqrt{n}}\right)$ *states.*

Proof. As mentioned above, the adapted power set construction is central to this proof. We will now give a formal definition of the generalized power set construction, valid for NFA and SVFA. This is a modification of the construction for NFA as it is used in standard computer science literature (see *e.g.* [1]). □

Definition 2.2. (power set construction).

Let $A = (Q, \Sigma, \delta, q_0, F)$ or $A = (Q, \Sigma, \delta, q_0, F, R)$ be a nondeterministic or self-verifying finite automaton respectively. Then an equivalent deterministic finite automaton $B = (Q', \Sigma, \delta', \{q_0\}, F')$ can be constructed as follows:

(i) The set of states Q' is the subset of the power set of Q, 2^Q, consisting of the states reachable from $\{q_0\}$ via δ' defined below.

(ii) The new transition function δ' is defined by
$\delta'(q', a) := \bigcup_{q \in q'} \delta(q, a)$ where $q' \in Q', a \in \Sigma$.

(iii) $F' := \{q' \in Q' \mid q' \cap F \neq \emptyset\}$ is the new set of accepting states consisting of those sets of states from Q' which contain an accepting state from A.

Note that in the case of SVFA, this definition explicitly relies only on final versus non-final states, not on the distinction between neutral and rejecting states. But implicitly, the definition of SVFA assures that there is no rejecting state in any $q' \in F'$, and that at least one rejecting state is in each $q' \in Q' \setminus F'$.

We now show that from the definition of self-verifying finite automata which do not allow for computations on any input word finishing in both accepting and rejecting states, we obtain considerably less states in the equivalent minimal DFA.

Claim 2.3. Let $B = (Q', \Sigma, \delta', \{q_0\}, F')$ be a DFA obtained from an SVFA $A = (Q, \Sigma, \delta, q_0, F, R)$ by power set construction.

If there are two states $S_1, S_2 \in Q'$ such that $S_1 \subset S_2$, then S_1 and S_2 are equivalent, *i.e.* can be identified without changing the language accepted by B.

In other words this means that after minimization B cannot contain two such states that are subsets of each other.

Recall that equivalence of S_1 and S_2 means that for all $x \in \Sigma^*$ we have $S_1 \xrightarrow{x} F'$ iff $S_2 \xrightarrow{x} F'$.

To prove Claim 2.3, assume to the contrary that there is an input word x which, read in S_1, leads to an accepting state whereas the same word, read in S_2, leads to a rejecting state in our DFA B. (By symmetry of the SVFA and DFA, the converse case is handled identically.)

Let us now take a look at the original SVFA A for the above situation. If in the DFA in state S_1 B reads x and reaches the set of accepting states, one of the original states in S_1, which we call q_1, has a path leading to an accepting state of A, that is $q_1 \xrightarrow{x} q_f \in F$. Analogously, there exists $q_2 \in S_2$ such that $q_2 \xrightarrow{x} q_r \in R$.

By construction of S_2 there exists $y \in \Sigma^*$ such that $q_0 \xrightarrow{y} q_1 \xrightarrow{x} q_f$ and $q_0 \xrightarrow{y} q_2 \xrightarrow{x} q_r$.

This results in a contradiction to the definition of a self-verifying finite automaton where no input word can have computations finishing in both accepting and rejecting states, which proves Claim 2.3.

Now, since states of B that are subsets of each other can be identified, we no longer can have 2^n states in a minimal deterministic finite automaton resulting from minimizing B, if we start with a self-verifying finite automaton A with n states. How many do we have at most? To answer this question we have to take a closer look at the construction of power sets. The problem is to find the maximal number of set combinations without strict inclusions.

As the greatest quantity is that of sets containing half the number of elements possible, the largest possible number of pairwise incomparable sets is $\binom{n}{\lfloor n/2 \rfloor}$.

We now have to determine an upper bound for $\binom{n}{\lfloor n/2 \rfloor}$. The result is the difference we can show between nondeterministic finite automata and self-verifying finite automata for the power set construction and thus the improvement in the upper bound for the conversion from self-verifying finite automata to deterministic finite automata.

What we still have to show is that $\binom{n}{\lfloor n/2 \rfloor}$ is indeed in $\mathrm{O}\left(\frac{2^n}{\sqrt{n}}\right)$ as we claimed in Theorem 2.1.

Recall the following property of binomial coefficients: $\binom{a}{b} = \frac{a!}{(a-b)!b!}$. Using this, we get:

$$\binom{n}{\lfloor n/2 \rfloor} \approx \frac{n!}{\left(\left(\frac{n}{2}\right)!\right)^2}. \tag{1}$$

To determine the value of this expression we use Stirling's formula:

$$n! \approx \left(\frac{n}{e}\right)^n \sqrt{2\pi n}.$$

This formula is also applicable for broken rational numbers, which is very convenient in our case. Note that we do not need an exact expression, as we are interested in asymptotic bounds.

We now simply apply Stirling's formula to the above expression (1) to obtain the upper bound we claimed earlier.

$$\begin{aligned}\frac{n!}{\left(\left(\frac{n}{2}\right)!\right)^2} &\approx \frac{\left(\frac{n}{e}\right)^n\sqrt{2\pi n}}{\left(\left(\frac{\frac{n}{2}}{e}\right)^{\frac{n}{2}}\sqrt{2\pi\frac{n}{2}}\right)^2}\\ &= \left(\frac{n}{e}\frac{e}{\frac{n}{2}}\right)^n\frac{\sqrt{2\pi n}}{\pi n}\\ &= \frac{2^n}{\sqrt{\pi\frac{n}{2}}} = \frac{2^n}{\sqrt{n}}\sqrt{\frac{2}{\pi}}.\end{aligned}$$

$\square$

Acknowledgements. We would like to thank Juraj Hromkovič for inciting this research.

References

[1] J.E. Hopcroft and J.D. Ullman, *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Reading, MA (1979).

[2] J. Hromkovič and G. Schnitger, On the Power of Las Vegas for One-way Communication Complexity, OBDDS, Finite Automata. *Inform. Comput.* **169** (2001) 284–296.

[3] O.B. Lupanov, A comparision of two types of finite automata. *Problemy Kibernetiki* **9** (1963) 321–326 (Russian).

[4] A.R. Meyer and M.J. Fischer, Economy of Description by Automata, Grammars, and Formal Systems, in *Proc. IEEE 12th Ann. Symp. on Switching and Automata Theory* 1971, 188–191.

[5] F.R. Moore, On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic and two-way finite automata. *IEEE Trans. Comput.* **20** (1971) 1211–1214.

Communicated by J. Karhumäki.
Received May 20, 2005. Accepted August 17, 2005.