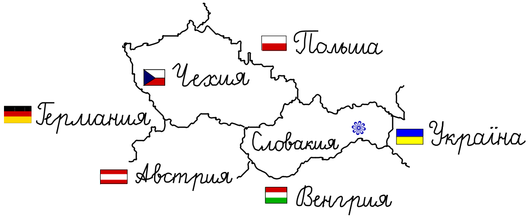
Operations on Self-Verifying Finite Automata

Galina Jirásková

Mathematical Institute, Slovak Academy of Sciences, Košice, Slovakia



Joint work with Jozef Jirásek, Jr. and Alexander Szabari

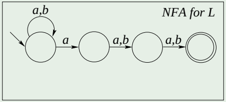
CSR 2015, Listvyanka, Russia

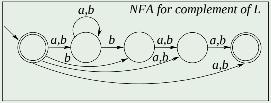
Self-Verifying Finite Automata

Definition (Self-Verifying Finite Automaton)

- nondeterministic automaton M over an alphabet Σ
  - states: accepting, rejecting, neutral
    ("yes", "no", "I don't know")
  - $L_a(M)$= strings with an accepting computation
    $L_r(M)$= strings with a rejecting computation
    1. $L_a(M) \cup L_r(M) = \Sigma^*$ and
    2. $L_a(M) \cap L_r(M) = \emptyset$
- the language accepted by SVFA M is $L_a(M)$

Example ($L = (a+b)^* a (a+b)^2$)



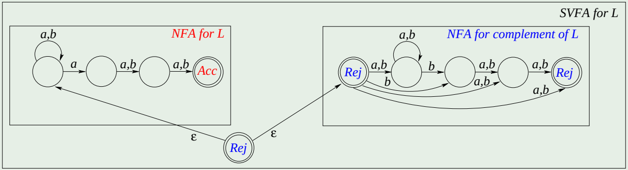


Self-Verifying Finite Automata

Definition (Self-Verifying Finite Automaton)

- nondeterministic automaton M over an alphabet Σ
  - states: accepting, rejecting, neutral
    ("yes", "no", "I don't know")
  - $L_a(M)$= strings with an accepting computation
    $L_r(M)$= strings with a rejecting computation
    1. $L_a(M) \cup L_r(M) = \Sigma^*$ and
    2. $L_a(M) \cap L_r(M) = \varnothing$
- the language accepted by SVFA M is $L_a(M)$

Example ($L = (a+b)^* a(a+b)^2$)



Self-Verifying Finite Automata Accept Regular Languages

Every DFA can be viewed as an SVFA:

- make all final states of the DFA accepting, and
  make all non-final states rejecting

If L is accepted by an n-state SVFA, then

- L is accepted by an n-state NFA (make accepting states final)
- L^c is accepted by an n-state NFA (make rejecting states final)

In this paper:

- all DFAs are complete
- all NFAs have a unique initial state

Why Self-Verifying Finite Automata?

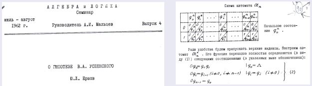
Motivation and History

- Ďuriš, Hromkovič, Rolim, Schnitger (STACS 1997)
  Hromkovič, Schnitger (Inform. Comput. 2001)
  - defined the model in connection with Las Vegas automata
  - complexity of Las Vegas automata problems (Ji DCFS 2004)
- Assent, Seibert (RAIRO-ITA 2007)
  - simulation of SVFAs by DFAs
- Jirásková, Pighizzini (LATA 2009, Inform. Comput. 2011)
  - optimal simulation by DFAs
- Why operations on SVFAs? (To come to Baikal:-)

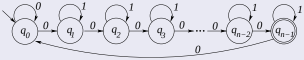
Known Results: NFA-to-DFA Conversion

- upper bound 2^n (subset construction: Rabin, Scott 1959)
- binary witnesses meeting the upper bound 2^n:

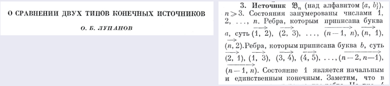
Yershov 1962



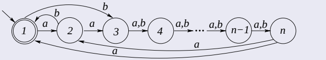
$\mathcal{A}_n$



Lupanov 1963

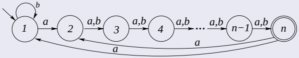


$\mathcal{B}_n$

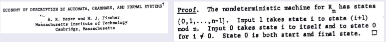


Moore 1971

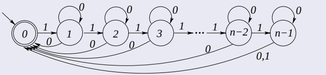
B_n



Meyer, Fischer 1971



R_n



Known Results: SVFA-to-DFA Conversion: Upper Bound

- upper bound 2^n reduced to $O(2^n/\sqrt{n})$ in [Assent, Seibert 07]
- further reduced to $g(n) \approx 3^{n/3}$ in [Jirásková, Pighizzini 2009]:
  - we can assign a graph $G(A)$ to an SVFA A
  - each reachable subset is a clique in $G(A)$
  - S and T are equivalent iff $S \cup T$ is a clique in $G(A)$

Moon, Moser 1960

The maximum number of cliques:

$$f(n) = \begin{cases} 3^{n/3}, & \text{if } n \bmod 3 = 0; \\ 4 \cdot 3^{\lfloor n/3 \rfloor - 1}, & \text{if } n \bmod 3 = 1; \\ 2 \cdot 3^{\lfloor n/3 \rfloor}, & \text{if } n \bmod 3 = 2. \end{cases}$$

Jirásková, Pighizzini 09

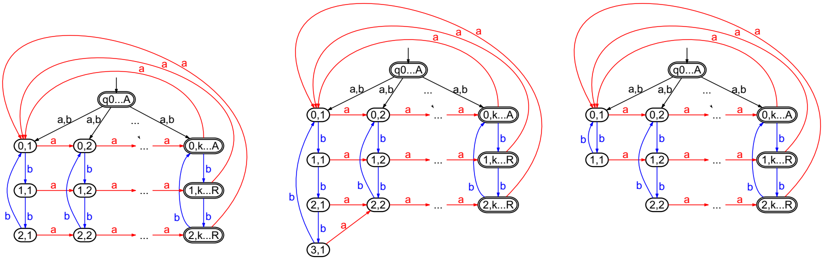
Every n-state SVFA can be simulated by a $g(n)$-state DFA, where

$$g(n) = 1 + f(n-1)$$

SVFA-to-DFA Conversion: Optimality of the Simulation

Jirásková, Pighizzini (LATA 2009)

The upper bound $g(n)$ is optimal, and it is met, depending on $n \bmod 3$, by the following automata:



Lower Bounds Methods

Well known: To prove that a DFA is minimal, show that

- all its states are reachable, and
- no two distinct states are equivalent.

Well known(?): To prove that an NFA is minimal, describe

a fooling set for the accepted language.

In this paper:

For a language L, we define a notion of an sv-fooling set, and we prove that its size provides a lower bound on the number of states in every SVFA for L.

The Complexity of Regular Operations on DFAs

Maslov 1970

Доклады Академии наук СССР
1970. Том 194, № 6

УДК 519.95 МАТЕМАТИКА

А. Н. МАСЛОВ

ОЦЕНКИ ЧИСЛА СОСТОЯНИЙ КОНЕЧНЫХ АВТОМАТОВ

Известно, что если $T(A)$ и $T(B)$ представимы в автоматах A и B с m и n состояниями соответственно ($m \geqslant 1$, $n \geqslant 1$), то

1) $T(A) \cup T(B)$ представимо в автомате с $m \cdot n$ состояниями.

2) $T(A) \cdot T(B)$ представимо в автомате с $(m-1) \cdot 2^n + 2^{n-1}$ состояниями ($n \geqslant 3$),

3) $T(A)^*$ представимо в автомате $\frac{3}{4} \cdot 2^m - 1$ состояниями ($m \geqslant 2$).

Нами построены примеры автоматов над алфавитом $\Sigma = \{0, 1\}$, на которых эти оценки достигаются.

1. Объединение: A имеет состояния $\{S_0, \ldots, S_{m-1}\}$ и переходы $S_{m-1}1 = S_0$, $S_i1 = S_{i+1}$ при $i \neq m-1$, $S_i0 = S_i$, S_{m-1} — заключительное состояние; B имеет состояния $\{P_0, \ldots, P_{n-1}\}$ и переходы $P_i1 = P_i$, $P_{n-1}0 = P_0$, $P_i0 = P_{i+1}$ при $i \neq n-1$, P_{n-1} — заключительное состояние.

2. Произведение: B имеет состояния $\{P_0, \ldots, P_{n-1}\}$ и переходы $P_{n-1}1 = P_{n-2}$, $P_{n-2}1 = P_{n-1}$, $P_i1 = P_i$ при $i < n-2$, $P_{n-1}0 = P_{n-1}$, $P_i0 = P_{i+1}$ при $i \neq n-1$, P_{n-1} — заключительное состояние; автомат A такой же, как для объединения.

3. Итерация: A имеет состояния $\{S_0, \ldots, S_{m-1}\}$ и переходы $S_{m-1}1 = S_0$, $S_i1 = S_{i+1}$ при $i \neq m-1$, $S_00 = S_0$, $S_i0 = S_{i-1}$ при $i > 0$, S_{m-1} — заключительное состояние.

По A и B строим соответствующие автоматы, как в ([2], [4]), и находим необходимое число достижимых и различных состояний, что и доказывает минимальность ([3]).

Upper bounds:

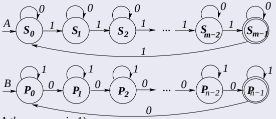
1) Union: $m.n$

2) Concatenation : $(m-1).2^n + 2^{n-1}$

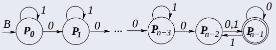
3) Star: $\frac{3}{4}.2^m - 1$

Lower bounds:

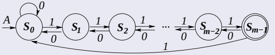
1)



2) A the same as in 1)

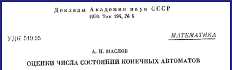


3)



General Formulation of the Problem

Maslov 1970

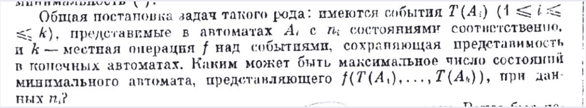


Доклады Академии наук СССР
1970. Том 194, № 6

УДК 519.95 — МАТЕМАТИКА

А. Н. МАСЛОВ

ОЦЕНКИ ЧИСЛА СОСТОЯНИЙ КОНЕЧНЫХ АВТОМАТОВ



Общая постановка задач такого рода: имеются события $T(A_i)$ $(1 \leqslant i \leqslant k)$, представимые в автоматах A_i с n_i состояниями соответственно, и k — местная операция f над событиями, сохраняющая представимость в конечных автоматах. Каким может быть максимальное число состояний минимального автомата, представляющего $f(T(A_1), \ldots, T(A_k))$, при данных n_i?

"Given languages $L(A_i)$ $(1 \leq i \leq k)$
accepted by automata A_i with n_i states
and a k-ary regular operation f,
what is the maximal number of states
in the minimal automaton for $f(L(A_1), \ldots, L(A_k))$,
considered as a function of n_i s?"

In this paper:

- automata are self-verifying
- f: boolean op., reversal, star, left and right qoutients, product

Intersection on Self-Verifying Automata

Intersection:
$K \cap L = \{w \mid w \in K \text{ and } w \in L\}$

Known results for intersection:

DFA:	mn	binary	[Maslov 1970]
NFA:	mn	binary	[Holzer Kutrib 2003]

Our result for intersection on self-verifying automata:

SVFA: mn $|\Sigma| \geq 2$

Proof idea for the upper bound:

- construct a product automaton, in which
accepting: (p, q) where both p and q are accepting;
rejecting: (p, q) where p or q is rejecting. □

Complementation, Union, Difference, Symmetric Diffrence

Complementation:
$L^c = \Sigma^* \setminus L$

Known results for complementation:

DFA:	n	unary	[folklore]
NFA:	2^n	binary	[Birget 1993, Ji 2005]

An observation for self-verifying automata:

SVFA n (interchange *acc* and *rej* states)

Union and difference on self-verifying automata:

SVFA: mn $K \cup L = (K^c \cap L^c)^c$

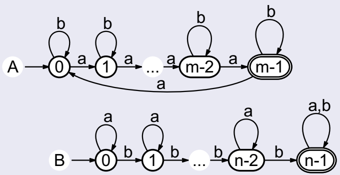
$K \setminus L = K \cap L^c$

Complementation:
$L^c = \Sigma^* \setminus L$

Known results for complementation:

DFA:	n	unary	[folklore]
NFA:	2^n	binary	[Birget 1993, Ji 2005]

Symmetric difference on SVFAs: Worst-case example

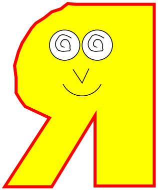


Reversal on Self-Verifying Automata

Reversal:
$L^R = \{w^R \mid w \in L\}$, where w^R is the mirror image of w

Known results for the reversal operation:

DFA:	2^n	binary	[Leiss 1981]
NFA:	$n+1$	binary	[Holzer, Kutrib 2003]



Our result for reversal:

SVFA: $2n+1$ $|\Sigma| \geq 2$

Proof idea for the upper bound:

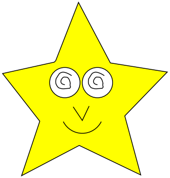
n-state SVFA for $L \Rightarrow$
n-state NFA for L and n-state NFA for $L^c \Rightarrow$
n-state NFA for L^R and n-state NFA for $(L^R)^c \Rightarrow$
$(2n+1)$-state SVFA for L^R $\square$

Star:

$L^* = \{u_1 u_2 \cdots u_k \mid k \geq 0 \text{ and } u_i \in L \text{ for all } i\}$

Known results for the star operation:

DFA:	$3/4 \cdot 2^n$	binary	[Maslov 1970]
NFA:	$n+1$	unary	[Holzer, Kutrib 2003]



Our results for star on self-verifying automata:

1. SVFA: $3/4 \cdot 2^n$ $|\Sigma| \geq 3/4 \cdot 2^n$
2. SVFA: $\geq 2^{n-1}$ for a quaternary alphabet

Proof idea for the lower bound:

- start with Maslov's DFA with $Q = \{1, \ldots, n\}$;
- define a new symbol c_S for each $S \subseteq Q$;
- describe an sv-fooling set for its star. □

Applications

Want to do something useful??? As a woman?

iron

do shopping

make a dinner

• • •

Applications II

. . . as a mother:

The work at my office (mathematics +TCS):

- going into a fairy tail
- going into a world with
  - truth
  - beauty
  - infinity

Nevertheless:

- Las Vegas computations
- unambiguous automata
- . . .

Summary and Open Problems

The complexity of operations on self-verifying finite automata:

	DFAs	SVFAs	NFAs
complement	n	n	2^n
intersection	mn	mn	mn
union	mn	mn	$m+n+1$
difference	mn	mn	?
symmetric difference	mn	mn	?
reversal	2^n	$2n+1$	$n+1$
star	$3/4 \cdot 2^n$	$3/4 \cdot 2^n$	$n+1$
left quotient	$2^n - 1$	$2^n - 1$	$n+1$
right quotient	n	$g(n)$	n
concatenation	$(m-1/2) \cdot 2^n$	$\Theta(3^{m/3} \cdot 2^n)$	$m+n$

Thank You for Your Attention