Uncertainty and Synchronization

(Abstract)

Vojtěch Vorel

Faculty of Mathematics and Physics, Charles University,
Malostranské nám. 25, Prague, Czech Republic
vorel@ktiml.mff.cuni.cz

1 Introduction

In a system described by a deterministic finite automaton $A = (Q, \Sigma, \delta)$, uncertainty corresponds to a set of *possible states* $S \subseteq Q$. A *reset word* is an input sequence $w \in \Sigma^*$ that maps all the states of S to a single state, i.e., $|\delta(S, w)| = 1$. The process of modifying and reducing the current uncertainty by applying the letters of w is referred to as *synchronization*. We ask:

1. For a given DFA with a given uncertainty, does there exist a reset word?
2. For a given DFA with a given uncertainty, what is the minimum length of reset words?
3. For a given n, what is the greatest minimum length between all n-state DFA?

The special case of $S = Q$ is widely studied within the pursuit of resolving the Černý conjecture. In this case, shortest reset words are at most of cubic length in the number of states and their existence can be tested in polynomial time (see, e.g., [7]).

In the general scope of $S \subseteq Q$, the lengths of shortest reset words become exponential and the testing becomes PSPACE-complete. Though these facts became classical during the last century, the field was not explored with enough precision. This contribution presents recent results [8, 9] that give answers to the following key questions:

1. Is there a polynomial (or at least $2^{o(n)}$) upper bound on the length of shortest reset words in strongly connected n-state DFA?
2. Is there a $2^{o(n)}$ upper bound on the length of shortest reset words in n-state DFA with a fixed alphabet?

Note that $2^{\mathcal{O}(n)}$ is a general upper bound following from the number of possible uncertainties. Unfortunately, both the above questions turn out to have negative answers. Moreover, the new $2^{\Omega(n)}$ lower bound involves DFA that combine both the restrictions, i.e., are strongly connected and binary [8].

Minimum lengths of reset words form an important special case of *depths* in transformation semigroups, which are the worst-case lengths of shortest expressions needed to compose a given $f \in \mathcal{T}_n$ from the members of a given $G \subseteq \mathcal{T}_n$ [6].

2 Decreasing the Alphabet Size

As it was remarked in 1976 by Burkhard [1], for any $m \geq k \geq 1$ one can construct a DFA (Q, Σ, δ) with $|Q| = m + 2$ and $|\Sigma| = \binom{m}{k}$ such that the shortest reset words for an uncertainty $S \subseteq Q$ have length $\binom{m}{k}$. The key ideas are:

1. Use two sink states $\mathrm{D}, \overline{\mathrm{D}}$ and set $\mathrm{D} \in S$. Thus, D remains in the uncertainty all the time, while $\overline{\mathrm{D}}$ must remain outside.
2. List all the size-k possible uncertainties on the other m states and assign them to the members of Σ. Let the first subset be included in S. Transitions leading to $\overline{\mathrm{D}}$ enforce an order in which all the letters must be applied. Finally the last subset can be mapped to D all at once.

Fixing $k = \frac{n}{2}$ and using Stirling's approximation, we check that $\binom{n}{n/2} = \theta(2^n/\sqrt{n})$ and obtain a lower bound $\Omega(2^n/\sqrt{n})$, i.e., $2^{\Omega(n)}$ on the length of reset words. The first idea above is a key tool that remains involved in all the later improvements. In particular, some of them were originally formulated in the scope of *careful synchronization* of partial finite automata. Such constructions can be straightforwardly modified to our scope by adding the two sink states to each automaton (see [8, Lemma 1]).

A more sophisticated method, which we call *radix construction*, was first described in [3] and produces $2^{\Omega(n)}$ lower bounds using only linear-size alphabets. In the most simple variant, the DFA consist of two sink states D and $\overline{\mathrm{D}}$ and a number of two-state components. As far as exactly one state of each component lies in the uncertainty, one can see the uncertainty as the binary representation of a non-negative integer. For each component, i.e., each digit, there is a letter in Σ whose application corresponds to setting that digit from 0 to 1 and setting the less significant digits from 1 to 0. Transitions leading to $\overline{\mathrm{D}}$ enforce that the only letter that can be applied is the one that increases the represented integer by one. Until the represented number is the greatest possible, a special letter that maps the whole uncertainty to D cannot be applied.

With adding auxiliary states to each component, one can employ a binary encoding of the indices of digits, thus obtaining a binary DFA with properties similar to the one described above. The additional states form a kind of decision trees of size proportional to $\log n$ per component. Due to the additional states, only $2^{\theta\left(\frac{n}{\log n}\right)}$ lower bounds on length of reset words is obtained [4].

	alphabet size	**strong connectivity**	**min. length of reset words**
Subset listing construction [1]	$2^{\theta(n)}$	no	$2^{\theta(n)}$
Basic radix construction [3]	$\theta(n)$	no	$2^{\theta(n)}$
High-order permutation construction [2]	2	no	$2^{\theta\left(\sqrt[3]{n}\log n\right)}$
Extended radix construction [4]	2	no	$2^{\theta\left(\frac{n}{\log n}\right)}$
New radix construction [8]	2	no	$2^{\theta(n)}$
New radix construction + swapping [8]	2	yes	$2^{\theta(n)}$

Table 1. Length of shortest reset words in n-state DFA – history of lower bounds

In order to obtain a series of binary n-state automata and uncertainties with shortest reset words of length $2^{\Omega(n)}$, additional ideas were necessary [8]:

1. All the components are joint to a long cycle according to a De Bruijn sequence, which encodes their unique indices. Instead of adding many states to each component, many components are utilized for performing an operation in a single component.
2. A single special component is added in order to guarantee that in each $\log n$-th step a special letter occurs that restores the original roles of the components.

3 Strong Connectivity

Automata with multiple sink states seem quite artificial, but all the classical lower bound relied heavily on them. It was very unclear whether the situation is similar in the scope of strongly connected DFA. The classical constructions used the two sink states to force application of particular letters during the synchronization. A common step in the proof looks like *„The letter x cannot be applied since that would make* $\overline{\mathrm{D}}$ *active, while* D *is active all the time"*. We developed the following alternative mechanism [8]:

A congruence ρ is a *swap congruence* of a DFA if, for each equivalence class C of ρ and each letter $x \in \Sigma$, the restricted function $\delta : C \times \{x\} \to Q$ is injective. The key property of a swap

congruence ρ is that an uncertainty S lack reset words whenever it contains distinct states r and s with $r\rho s$. A reduction turns an arbitrary DFA with a given uncertainty to a strongly connected case, where:

1. there are two copies of the original automaton and two additional states $\mathrm{E}, \overline{\mathrm{E}}$;
2. the number of letters is increased by the number of former strongly connected components;
3. the new uncertainty contains one copy of the original one plus the state E.

There is a fixed swap congruence where each equivalence class, besides of the class $\{\mathrm{E}, \overline{\mathrm{E}}\}$, consists of the two copies of an original state. The additional letters make the resulting DFA strongly connected, but their possible effective application during synchronization makes both E and $\overline{\mathrm{E}}$ appear in the resulting uncertainty. Thus, though the resulting DFA is strongly connected, the only reset words correspond to reset words of the original DFA.

4 A Note on Depths in Transformation Semigroups

It has been pointed out by Arto Salomaa [6] in 2001 that very little is known about the minimum length of a composition needed to generate a trasformation from a given set of generators. We denote by $\mathcal{T}_n$ the semigroup of all transformations of $\{1, \ldots, n\}$. Given $\mathbf{G}, \mathbf{F} \subseteq \mathcal{T}_n$, we are interested in the length k of a shortest sequence $g_1, \ldots, g_k \in \mathbf{G}$ such that $g_1 \circ \cdots \circ g_k \in \mathbf{F}$. There is a trivial upper bound $n!$ given by the size of $\mathcal{T}_n$. Arto Salomaa refers to a single nontrivial lower bound, namely $(\sqrt[3]{n})!$. In fact, he omits the $2^{\Omega(n)}$ lower bounds for synchronization, which apply easily to the scope of depths.

Recently, a tight bound of the form $2^n e^{\sqrt{(n/2)\ln n}(1+o(n))}$ was settled by Panteleev [5].

References

1. Burkhard, H.: Zum Längenproblem homogener Experimente an determinierten und nichtdeterministischen Automaten. Elektronische Informationsverarbeitung und Kybernetik 12(6), 301–306 (1976)
2. Goralčík, P., Hedrlín, Z., Koubek, V., Ryšlinková, J.: A game of composing binary relations. RAIRO - Theoretical Informatics and Applications - Informatique Theorique et Applications 16(4), 365–369 (1982)
3. Martyugin, P.V.: A lower bound for the length of the shortest carefully synchronizing words. Russian Mathematics 54(1), 46–54 (2010)
4. Martyugin, P.V.: Careful synchronization of partial automata with restricted alphabets. In: Bulatov, A.A., Shur, A.M. (eds.) Computer Science - Theory and Applications, Lecture Notes in Computer Science, vol. 7913, pp. 76–87. Springer Berlin Heidelberg (2013)
5. Panteleev, P.: Preset distinguishing sequences and diameter of transformation semigroups. In: Dediu, A.H., Formenti, E., Martín-Vide, C., Truthe, B. (eds.) Language and Automata Theory and Applications, Lecture Notes in Computer Science, vol. 8977, pp. 353–364. Springer International Publishing (2015)
6. Salomaa, A.: Compositions over a finite domain: From completeness to synchronizable automata. In: A Half-century of Automata Theory, pp. 131–143. World Scientific Publishing Co., Inc., River Edge, NJ, USA (2001)
7. Volkov, M.V.: Synchronizing automata and the Černý conjecture. In: Martín-Vide, C., Otto, F., Fernau, H. (eds.) Language and Automata Theory and Applications, Lecture Notes in Computer Science, vol. 5196, pp. 11–27. Springer Berlin Heidelberg (2008)
8. Vorel, V.: Subset synchronization and careful synchronization of binary finite automata. International Journal of Foundations of Computer Science (accepted in August 2015), http://arxiv.org/abs/1403.3972
9. Vorel, V.: Subset synchronization of transitive automata. In: Proceedings 14th International Conference on Automata and Formal Languages, AFL 2014. pp. 370–381 (2014)