Undecidability of State Complexities Using Mirror Images

Arto Salomaa

Turku Centre for Computer Science,
Joukahaisenkatu 3–5 B, 20520 Turku, Finland
asalomaa@utu.fi

Abstract. We establish the undecidability of the state complexity of compositions of the operation mirror image and two other regularity-preserving operations. The undecidability of Hilbert's Tenth Problem is not needed; the weaker Davis-Putnam-Robinson Theorem suffices for the reduction. Special attention is paid to the maximal state complexity of mirror images and the maximal deterministic state complexity of nondeterministic finite automata.

Keywords: Finite automaton, State complexity, Undecidability, Mirror image, Nondeterminism, Exponential polynomials.

Dedication. This paper is dedicated to my friend and colleague *Jürgen Dassow* on the occasion of his 65th birthday. The first scientific contacts between us go back to early 70's. Jürgen was one of the first, maybe the very first, researcher in the Eastern bloc who investigated developmental languages. Our early cooperation was possible because it was easier to travel to Finland than elsewhere in the West. When life and travel became easier, many mutual visits and different forms of cooperation started to be possible between us, as well as our students. The conferences organized by Jürgen, such as the DLT-95 in Magdeburg following DLT-93 in Turku, made even further contacts possible. Jürgen was always interested in decision problems and, therefore, I believe my contribution to be appropriate for the volume. I wish Jürgen continuing success in science and life in general.

1 Introduction

It is well known that, for every regular language L, there is a unique, up to isomorphism, finite deterministic automaton accepting L which is minimal with respect to the number of states. The effect of a regularity-preserving operation on the number of states is customarily referred to as the *state complexity* of that operation. For instance, if $L_i, 1 \leq i \leq 3$, are regular languages accepted by automata with x_i states, respectively, how many states does the composition $(L_1 \cup L_2)L_3$ require in terms of the numbers x_i?

The recent study of state complexity has been motivated by many new applications of automata, e.g., in natural language and speech processing, software

H. Bordihn, M. Kutrib, and B. Truthe (Eds.): Dassow Festschrift 2012, LNCS 7300, pp. 221–235, 2012.

engineering, and parallel processing, which utilize finite automata of very large sizes. The state complexity gives a good estimate of the size of the application and a lower bound of its time and space complexities.

The effect of basic regularity-preserving operations was settled in [13]. Apart from the basic operations alone, also combined operations have been investigated, for instance, in [12,7,2,10,1]. The worst-case state complexity of the composition of two operations can be smaller than the one obtained directly from the (known) complexities of the two operations. For instance, the state complexity of the star operation on the result of the union of two regular languages, with the state complexities m and n, is $2^{m+n-1} - 2^{m-1} - 2^{n-1} + 1$. The direct composition of the two state complexities gives the result $2^{mn-1} + 2^{mn-2}$, which is much higher than the actual state complexity [7].

However, there is no general method of determining the state complexity of arbitrary compositions of operations. This undecidability result was established in [8], using compositions of two simple operations. Then a reduction of *Hilbert's Tenth Problem* could be used.

It is natural to investigate the effect and applicability of other operations within this framework. In this paper we focus the attention on the operation *mirror image*, denoted $mi(w), mi(L)$, (also called *reversal*, denoted w^r, L^r).

The state complexity of the *mirror image* of a regular language is of special interest because it is connected with the difference between nondeterminism and determinism in the following way. The mirror image of a language $L(\mathcal{A})$ is accepted by an automaton obtained from $\mathcal{A}$ by reversing all (labeled) arrows, and interchanging initial and final states. The latter automaton is nondeterministic. Thus, the (deterministic) state complexity of the mirror image is the number of states in the minimal equivalent deterministic automaton. Using the *subset construction*, [4], we see that the maximal increase in the number of states goes from n to 2^n. Thus, the state complexity of the language $mi(L)$ is between n and 2^n if the state complexity of L is n. Languages L where the mirror image $mi(L)$ actually reaches the upper bound 2^n can be used as "representations" of the exponential function. Consequently, we can, instead of Hilbert's Tenth Problem, use the weaker *Davis-Putnam-Robinson Theorem* as a basis of reduction.

A brief outline of the contents of the paper follows. In Section 2 we introduce the basics about state complexity, and discuss a special operation needed in the sequel. The next section investigates languages, with detailed proofs, whose mirror images possess the maximal state complexity. In fact, the results obtained there are interesting on their own right. They are stronger than what is actually needed for our undecidability result. Section 4 discusses exponential polynomials and modifies the Davis-Putnam-Robinson Theorem to suit for our purposes. Sections 5 and 6 present a method of associating with an exponential polynomial E a composition C of regular languages such that, for all tuples of values of the variables, the state complexity of C equals at most the value of E when the languages in C have state complexities defined by the tuple in question. For specific languages the value is actually reached. Moreover, Section 6 proves the following undecidability result. Given a sequence $C_i,\ i = 1, 2, \ldots,$ of compositions and a sequence $E_i,\ i = 1, 2, \ldots,$ of exponential polynomials, both effectively constructible, it is undecidable whether or not E_i is a state complexity function for C_i. Some open problems are presented in the final section.

2 State Complexity – Marked Catenations

We assume that the reader is familiar with the basics of finite automata and regular languages. Whenever necessary, the article of Sheng Yu in [4] can be consulted.

We use the customary notation

$$\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$$

for *deterministic finite automata*, DFA's. The five items are, respectively, the state set, the input alphabet, the transition function, the initial state, and the set of final states. We consider only *complete* automata: $\delta(q, a)$ is defined for all $q \in Q$ and $a \in \Sigma$. Very often in this paper, n refers to the cardinality of the state set: $\sharp(Q) = n$.

A state of an automaton is called a *sink* if no sequence of transitions leads from it to a final state. (Sinks are often also referred to as *garbage states*.)

The (regular) language accepted by the DFA $\mathcal{A}$ is denoted by $L(\mathcal{A})$. The *state complexity* of a regular language L is the number of states in the minimal DFA $\mathcal{A}$ such that $L = L(\mathcal{A})$.

The DFA $\mathcal{A}$ is *functionally complete* if the transition monoid of $\mathcal{A}$, that is the monoid generated by the functions $f_a(q) = \delta(q, a)$ where a ranges over Σ, consists of all of the n^n mappings of Q into Q. The notion of *functional completeness* can be extended to sets of functions $f : Q \to Q$, where Q is an arbitrary finite set. (For more details, see [5] or [11].)

We use natural graphical representations for DFA's, where states are represented by circles and transitions by labeled arrows.

We consider also *nondeterministic* finite automata, NFA's. Our NFA's may possess several initial states. (They are actually called NNFA's in [4].)

For an NFA $\mathcal{A}$, we denote by $S(\mathcal{A})$ the DFA obtained from $\mathcal{A}$ by the *subset construction*. The initial state of $S(\mathcal{A})$ is the set of initial states of $\mathcal{A}$. As states of $S(\mathcal{A})$ we consider only subsets reachable from the initial state. If $\sharp(Q) = n$, the automaton $S(\mathcal{A})$ has at most 2^n states. It is a direct consequence of the subset construction that the automaton $S(\mathcal{A})$ is complete.

We already pointed out that, for a regular language L, there is a unique minimal automaton accepting L. The number of states in this automaton is referred to as the *state complexity* of L. The situation is more involved if we consider classes of languages and *state complexity functions*.

We are interested in *compositions* of variable regular languages. We will now give a general definition of *state complexity functions*. The definition is given for arbitrary compositions although, for the undecidability result below, we actually need it only for some special compositions. The functions we are considering will always map some power of $\mathbb{N}_0$ into $\mathbb{N}_0$. Again, only some special functions (*exponential polynomials* defined below) will be needed for our undecidability result.

In the usual state complexity considerations, each variable of the function corresponds to a unique language. We allow also the more general case, where several languages are associated with the same variable.

Definition 1. *Consider a function $F(x_1, \ldots, x_m)$, $m \geq 1$,, some composition $C^n(L_1, \ldots, L_n)$, $n \geq m$, of languages involving only regularity-preserving operations, as well as a surjective mapping φ of the index set $\{1, \ldots, n\}$ onto the index set $\{1, \ldots, m\}$. Then the function $F(x_1, \ldots, x_m)$ is a* state complexity function *of the composition $C^n(L_1, \ldots, L_n)$ if the following condition is satisfied. Let $(x_1, \ldots, x_m)$ be an arbitrary m-tuple of nonnegative integers. Whenever $1 \leq i \leq m$ and each L_j, $j \in \varphi^{-1}(i)$, is a regular language with state complexity x_i, then the composition $C^n(L_1, \ldots, L_n)$ is accepted by an automaton with at most $F(x_1, \ldots, x_m)$ states.*

Note that when we say that a function $F(x_1, \ldots, x_m)$ is a state complexity function of the composition $C^n(L_1, \ldots, L_n)$, this means that the value of $F(x_1, \ldots, x_m)$ gives an *upper bound* for the state complexity of the language $C^n(L_1, \ldots, L_n)$ when each variable x_i is assigned the state complexity of the languages L_j such that $\varphi(j) = i$.

Marked Catenation. $L_1 \ddagger L_2$ is a special operation needed in the sequel. It is the catenation of the languages L_1, $\ddagger$, L_2, where $\ddagger$ is a letter not appearing in the alphabets of L_1 and L_2. Similarly we consider marked catenations of arbitrarily many languages. The following result is from [8]. In view of its importance, we give the proof also here.

Theorem 1. *Assume that L_i are regular languages (maybe over different alphabets) with state complexities σ_i, $1 \leq i \leq r$, $r \geq 2$. Assume, further, that for each i, $1 \leq i \leq r$, the minimal automaton $\mathcal{A}_i$ for L_i has no sinks. Then the marked catenation*

$$L_1 \ddagger L_2 \ddagger \cdots \ddagger L_r$$

is accepted by an automaton $\mathcal{A}$ with

$$\sum_{i=1}^{r} \sigma_i + 1 = \sigma$$

states but by no automaton with fewer than σ states. The alphabet of $\mathcal{A}$ consists of the union of the alphabets of L_i and of $\ddagger$. The initial state of $\mathcal{A}_1$ is the initial state of $\mathcal{A}$, and the final states of $\mathcal{A}_r$ constitute the set of final states of $\mathcal{A}$.

Proof. An automaton $\mathcal{A}$ accepting the marked catenation is obtaining by joining the automata $\mathcal{A}_i$, $1 \leq i \leq r$, in the following way. From each final state of $\mathcal{A}_i$, $1 \leq i \leq r-1$, introduce a transition labeled by $\ddagger$ to the initial state of $\mathcal{A}_{i+1}$. From all other states of $\mathcal{A}_i$, $1 \leq i \leq r-1$, as well as from all states of $\mathcal{A}_r$, introduce a transition labeled by $\ddagger$ to an additional sink state. It is clear that $\mathcal{A}$ accepts the marked catenation and has σ states. On the other hand, no automaton with fewer states can accept the marked catenation. Each word has to have exactly $r-1$ occurrences of ‡. States in two different automata $\mathcal{A}_i$ cannot be combined because this would result into too many occurrences of the letter ‡. □

If some of the automata $\mathcal{A}_i$ would possess a sink, then the various sinks can be combined, and the total number σ can be reduced accordingly.

3 Mirror Images

For a word $w = b_1 b_2 \dots b_k$, $b_i \in \Sigma$, its *mirror image* (also called *reversal*) is defined by

$$mi(w) = b_k \dots b_2 b_1.$$

The mirror image $mi(L)$ of a language L consists of the mirror images of its words. For a DFA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, we denote by $R(\mathcal{A})$ the NFA obtained from $\mathcal{A}$ by reversing all arrows and interchanging the initial and final states. It is obvious that $R(\mathcal{A})$ accepts the language $mi(L(\mathcal{A}))$. If $\sharp Q = n$, then $S(R(\mathcal{A}))$ has at most 2^n states. Consequently, the state complexity of $mi(L(\mathcal{A}))$ is at most 2^n. For a proof of the following well-known result, see [4], Vol. 1, p. 95.

Lemma 1. *If in a DFA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ all states of Q are reachable from q_0, then $S(R(\mathcal{A}))$ is the minimal DFA accepting $mi(L(\mathcal{A}))$.*

Thus, assuming that the state complexity of a language $L = L(\mathcal{A})$ equals n, the state complexity of $mi(L)$ equals 2^n if and only if all of the 2^n subsets of Q appear as states of $S(R(\mathcal{A}))$. We now consider a sequence of automata where this actually happens. Some of the automata were discussed, omitting many details, in [9] which was one of the very last joint works of the present author with the late *Derick Wood.* Therefore, we call them here *Wood automata.* Wood automata are investigated, from a different point of view, also in [6].

The Wood automaton $W(n)$ with n states is over the binary alphabet $\{a, b\}$. (If some other letters, say c, d, are used, this will be indicated in the notation: $W(n)(c, d)$). The state set is $Q = \{1, 2, \dots, n\}$. The transitions $f_a(x) = \delta(x, a)$ and $f_b(x) = \delta(x, b)$ are defined as follows. The function $f_a(x)$ is the circular permutation $(123 \cdots n)$, whereas

$$f_b(1) = f_b(3) = 1,\ f_b(4) = 3,\ f_b(x) = x \text{ otherwise.}$$

We assume first that $n \geq 5$ and that n is not divisible by 4. Then the state 1 is both the initial and the only final state. In this case the automaton $W(n)$ is depicted in Figure 1. (Final states are marked by double circles, the incoming arrow points to the initial state.) We will return later to the remaining cases.

The essential tool in our considerations is the subset construction, and the main problem the connectedness of the resulting graph. The framework can be described in terms of *subset functions* as follows.

Consider a finite set $Q = \{1, 2, \dots, n\}$ and mappings $f : Q \to 2^Q$. Extend such a mapping additively to a mapping from 2^Q to 2^Q. (Thus, for $X \subseteq Q$, the value $f(X)$ is the union Y of the values $f(x)$, where $x \in X$.)

Let F be a (finite) set of such *subset functions* For $X, Y \subseteq Q$, we use the notation $X \Rightarrow_F Y$ to indicate that $f(X) = Y$, for some $f \in F$. Finally, let $\Rightarrow_F^*$ be the reflexive transitive closure of the relation $\Rightarrow_F$.

Definition 2. *A set F of subset functions is* complete *if, for any $X \subseteq Q$, $X \neq \emptyset, Q$ and any $Y \subseteq Q$, we have*

$$X \Rightarrow_F^* Y.$$

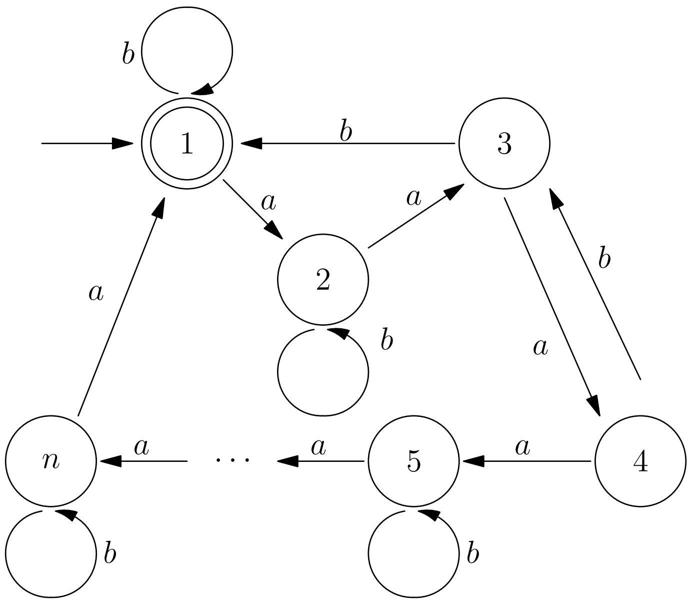

Fig. 1. Wood automaton $W(n)$, $n \geq 5,\ 4 \nmid n$

If $X \Rightarrow_F^* Y$, we say that Y is *reachable* from X (via F). Although functional completeness is well understood (see [5] and the references given there), the completeness of sets of subset functions is **an open problem area**. The restrictions in Definition 2, concerning $\emptyset, Q$, become obvious below.

The following general considerations are independent of initial and final states and concern an arbitrary $n \geq 4$. For convenience, we denote the inverses of the functions f_a and f_b by A and B, respectively. We denote also $F = \{A, B\}$. Clearly, A and B are subset functions in the sense defined above. Thus, A affects the circular permutation $(n \cdots 321)$, whereas B maps 1 to $\{1, 3\}$, 3 to 4, 4 to $\emptyset$, and x to x, otherwise. In connection with the set Q, additions will be carried out *modulo* n, that is, $i + j$ stands for the smallest *positive* remainder modulo n. For $X = \{x_1, \ldots, x_k\} \subseteq Q$, we consider sets

$$X^{+i} = \{x_{1+i}, \ldots, x_{k+i}\},\ 1 \leq i \leq n.$$

(Observe that $X^{+n} = X$.) Since A is a circular permutation, we have, for all i,

$$X \Rightarrow_A^* X^{+i}.$$

This fact will be used frequently in our following arguments.

Observe that $4 \Rightarrow_B \emptyset$. We now claim that, from any

$$X \subseteq Q,\ 2 \leq \sharp X = k \leq n-1,$$

a subset Y of Q with cardinality $k-1$ is reachable. If, for some $i,\ 0 \leq i \leq n-1$, the element $i+4$ is in X, whereas $i+1$ is not there, then we apply B to the set $X^{+(n-i)}$, and obtain a subset Y as required. If no such pair of elements exists in X, then n is divisible by 3, and X consists of one or two residue classes modulo 3. (This follows because $X \neq Q$.) We may assume, by applying A if necessary, that the numbers $3, 6, \ldots, n$ are in X, whereas the numbers $1, 4, \ldots, n-2$ are not there. Now $B(X)$ is obtained from X by replacing 3 with 4. Hence, $Y = B(B(X))$ is of cardinality $k-1$. This completes the proof of our claim.

We now work inductively "upwards", increasing the cardinality of the reachable sets. We will investigate which subsets Y are reachable from the singleton $\{1\}$. Reachability is immediately verified for $\emptyset$ and all singletons. We now assume inductively that every subset X' of Q with cardinality $k-1,\ 2 \leq k \leq n$, is reachable, and consider an arbitrary subset X of Q with cardinality k. Given X, we want to show how to choose an X' of cardinality $k-1$ such that $X' \Rightarrow_F^* X$.

Assume first that X contains, for some i, the elements i and $i+2$. By applying A, we may assume that 1 and 3 are contained in X. If 4 is (resp. is not) in the set X thus modified, we let X' be the set obtained from X by removing the element 4 (resp. 3). Then $X' \Rightarrow_F^* X$. (Of course, we still have to use A to get the original X.)

From now on we assume that no elements i and $i+2$ are in X. This implies that X contains no three consecutive elements $i, i+1, i+2$ and, whenever i is in X but $i+1$ is not in X, then also $i+2$ is not in X. Intuitively, X consists of singletons and pairs of two consecutive elements, all separated by at least two "non-elements". (All the time we are using the modular arithmetic: n and 1 are consecutive.)

Assume that, for some i, the element i is in X, whereas $i+1$ and $i-1$ are not. By the preceding paragraph, also $i+2$ and $i-2$ are not in X. By an A-shift, we may assume that 1 is in X, whereas $2, 3, n-1, n$ are not.

Let j be the smallest element, apart from 1, in X. We know that $j \geq 4$. Construct X' by removing j from X. In the following reachability sequence we have marked down only the relevant elements in the sets. It is essential that n is not in X. (Observe that B alters elements $1, 3, 4$ only.)

$$\begin{aligned} X' \Rightarrow_B^* \{1,3,4\} \Rightarrow_A^* \{2,4,5\} \Rightarrow_B^* \{2,5\} \Rightarrow_A^* \{1,4\} \\ \Rightarrow_A^* \{n,3,\} \Rightarrow_B^* \{n,4\} \Rightarrow_A^* \{1,5\} \end{aligned}$$

This shows how X is reachable if $j=4$ or $j=5$. For an arbitrary j, we reach X by repeating the transformations on the second line.

Hence, we may assume that X does not contain such isolated elements. This implies, by our previous constructions, that X consists of pairs of consecutive elements, separated by at least two "non-elements". Suppose that, for some i, the elements i and $i+1$ are in X, whereas none of the elements $i-3, i-2, i-1, i+2, i+3$ is in X. By an A-shift, we may again assume that 1 and 2 are in X, whereas none of the elements $n-2, n-1, n, 3, 4$ is in X.

We now let X' be the following subset of cardinality $k-1$:

$$X' = \{1\} \cup \{j+1 \mid j \in X,\ j \neq 1, 2\}.$$

Then the following reachability chain is valid:

$$\begin{aligned} X' \Rightarrow_B^* \{1,3\} \cup \{j+1 \mid j \in X,\ j \neq 1,2\} = X_1 \\ \Rightarrow_A^* \{3,5\} \cup \{j+3 \mid j \in X,\ j \neq 1,2\} = X_2 \\ \Rightarrow_B^* \{4,5\} \cup \{j+3 \mid j \in X,\ j \neq 1,2\} \Rightarrow_A^* X \end{aligned}$$

It is important to observe that neither $n-1$ nor n is in X_1 and, consequently, neither 1 nor 2 is in X_2.

Thus, we have reached the conclusion that X consists of pairs of consecutive elements, separated by exactly two "non-elements". But this means that n is divisible by $4:\ n = 4m$.

For $n = 4m$, we now define the Wood automaton $W(n)$ by choosing the set

$$WF(n) = \{4i+1, 4i+2 \mid 0 \leq i \leq m-1\}$$

as the set of final states. Otherwise, the definition of $W(n)$ remains unaltered. The automaton $W(8)$ is illustrated in Figure 2.

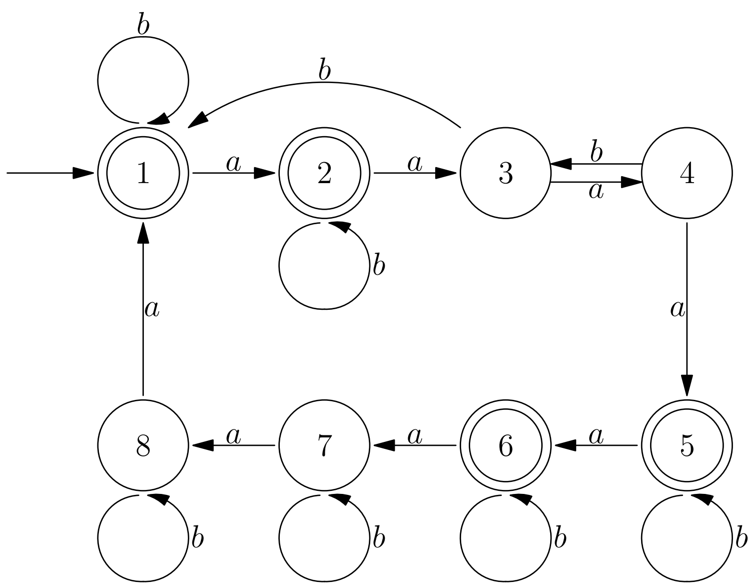

Fig. 2. Wood automaton $W(8)$

Our argument above shows, by Lemma 1 and the reachability of all subsets, that the state complexity of the language $mi(L(W(n)))$, $n \geq 5$, equals 2^n, provided n is not divisible by 4. In fact, in this case we are free, [6], to choose the initial and the set of final states. However, our choice of the final state set $WF(n)$ guarantees that the state complexity result holds true also if n is divisible by 4. (There is no change in the proof when we reduce the cardinality of reachable subsets. The argument applies also if we want to increase the cardinality or keep it $2m$ which is the cardinality of $WF(n)$.) Hence, we have established the following result.

Theorem 2. *For $n \geq 4$, the state complexity of the language $mi(L(W(n)))$ equals 2^n.*

We still have to deal with the small values of n. The automata $W(2)$ and $W(3)$ are depicted in Figure 3. It is immediately verified that the state complexities of the mirror images are 4 and 8 in these cases. Hence, we obtain the following corollary of Theorem 2.

Theorem 3. *For $n \geq 2$, the state complexity of the language $mi(L(W(n)))$ equals 2^n.*

Summarizing, we obtain the following result.

Theorem 4. *For every $n \geq 2$, the Wood automaton $W(n)$ has n states but the minimal deterministic automaton equivalent to $R(W(n))$ has 2^n states.*

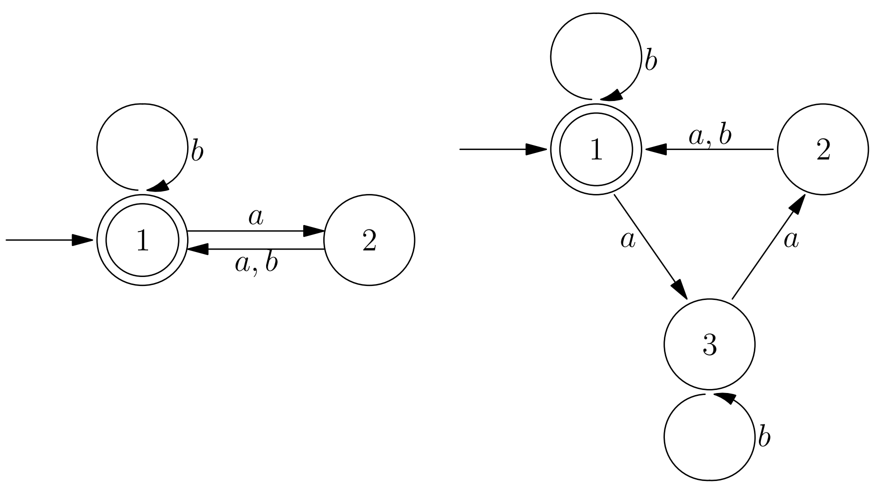

Fig. 3. Wood automata $W(2)$ and $W(3)$

4 Modifications of the Davis-Putnam-Robinson Theorem

By an *exponential polynomial*, briefly *E-polynomial*, we mean a finite sum of terms of the form $\alpha_1\alpha_2\cdots\alpha_n$, $n \geq 1$, where each α_i, $i \geq 1$, is a variable, or of the form 2^x, for some variable x. An exponential polynomial may contain several identical terms, which will be expressed with coefficients in $\mathbb{N}_0$. For instance,

$$2^{x_1}x_3^2x_4 + 4x_1x_2 + 2^{x_3}2^{x_1}x_2x_3x_4$$

is an exponential polynomial.

By the *Davis-Putnam-Robinson Theorem*, for every recursively enumerable set S of nonnegative integers, there are (effectively constructible) exponential polynomials $E_i(x_0, x_1, \ldots, x_m)$, $i = 1, 2$, such that $x_0 \in S$ if and only if the equation

$$E_1(x_0, x_1, \ldots, x_m) = E_2(x_0, x_1, \ldots, x_m)$$

has a solution in nonnegative integers $(x_1, \ldots, x_m)$. (For details and a proof using register machines, see [3].) Using the universal Turing machine and the undecidability of the emptiness of recursively enumerable languages, the result can be expressed in the following form. There are (effectively constructible) exponential polynomials $E(x_0, x_1, \ldots, x_m)$ and $E'(x_0, x_1, \ldots, x_m)$ such that, given $x_0 \geq 0$, it is undecidable whether or not the equation

$$E(x_0, x_1, \ldots, x_m) = E'(x_0, x_1, \ldots, x_m)$$

has a solution in nonnegative integers $(x_1, \ldots, x_m)$. By substituting the value $x_0 = i \geq 1$ to the exponential polynomials E and E', we obtain two infinite sequences E_i and E'_i, $i = 1, 2, \ldots$, such that, given $i \geq 1$, it is undecidable whether or not the equation

$$E_i(x_1, \ldots, x_m) = E'_i(x_1, \ldots, x_m)$$

has a solution in nonnegative integers $(x_1, \ldots, x_m)$. Denote

$$P_i(x_1, \ldots, x_m) = E_i(x_1, \ldots, x_m) - E'_i(x_1, \ldots, x_m),\ i \geq 1.$$

Consider the inequalities

$$0 \leq (P_i(x_1, \ldots, x_m)^2) - 1,\ i = 1, 2, \ldots$$

Clearly, for any given i, this inequality is valid for all m-tuples $(x_1, \ldots, x_m)$ of nonnegative integers exactly in case the equation

$$E_i(x_1, \ldots, x_m) = E'_i(x_1, \ldots, x_m)$$

has *no solution* in nonnegative integers. Therefore, by the Davis-Putnam-Robinson Theorem, there is no algorithm of deciding, given i, whether or not the inequality

$$0 \leq (P_i(x_1, \ldots, x_m)^2) - 1$$

holds for all m-tuples $(x_1, \ldots, x_m)$ of nonnegative integers. We now move all negative terms from the right side to the left side. This gives rise to a new inequality, equivalent to the original one,

$$E_i^{(l)}(x_1, \ldots, x_m) \leq E_i^{(r)}(x_1, \ldots, x_m),$$

where $E_i^{(l)}$ and $E_i^{(r)}$ are E-polynomials.

These considerations are summarized in the following Theorem.

Theorem 5. *There is no algorithm of deciding, given a positive integer i, whether or not the inequality*

$$E_i^{(l)}(x_1, \ldots, x_m) \leq E_i^{(r)}(x_1, \ldots, x_m)$$

holds for all m-tuples $(x_1, \ldots, x_m)$ of nonnegative integers. Here $E_i^{(l)}$ and $E_i^{(r)}$ are effectively constructible E-polynomials over the set of variables $\{x_1, \ldots, x_m\}$. Moreover, there is a finite set S of terms of the form

$$y_1^{j_1} \cdots y_{2m}^{j_{2m}},\ j_\mu \geq 0,\ 1 \leq \mu \leq 2m,$$

such that every polynomial $E_i^{(l)}$, $i = 1, 2, \ldots$, equals the sum of some terms in S, provided with positive integer coefficients.

Thus, each $E_i^{(l)}$ is a (finite) sum of terms of the form

$$y_1^{j_1} \cdots y_{2m}^{j_{2m}},\ j_\nu \geq 0,\ 1 \leq \nu \leq 2m,$$

provided with positive integer coefficients depending on i. The choice of i affects only the multiplicity of each term, i.e., it tells how many times each term appears in the polynomial $E_i^{(l)}$. (We have $2m$ instead of m because a term may contain both x and 2^x, for some variable x.)

In the sequel we will associate the E-polynomials $E_i^{(l)}$ with specific compositions of regular operations, whereas the polynomials $E_i^{(r)}$ will constitute the proposed state complexities.

5 Special Compositions and Associated *E*-Polynomials

The specific compositions we will need use the three regularity-preserving operations of mirror image, intersection and marked catenation. Therefore, we will call them *three-compositions.* The operations are not nested arbitrarily. The way of nesting is specified in the following definition.

Definition 3. *A* three-composition *over the set $\{L_1, \ldots, L_n\}$, $n \geq 2$, of language variables is an expression*

$$\gamma_1 \ddagger \gamma_2 \ddagger \cdots \ddagger \gamma_r,\ r \geq 2,$$

where each γ_i, $1 \leq i \leq r$, is of the form

$$\gamma_i = M_1 \cap \cdots \cap M_{j(i)},\ j(i) \geq 1,$$

such that the M's are different ones among the language variables L_ν, $1 \leq \nu \leq n$, either appearing as plain L_ν, or in the form $mi(L_\nu)$. A language variable L_ν can appear both as such and in the form $mi(L_\nu)$ in the same γ_i.

Parentheses can be added for clarity. For instance,

$$(mi(L_2) \cap L_3)\ddagger(L_1 \cap L_2 \cap L_3)\ddagger(mi(L_1) \cap mi(L_3) \cap L_1)$$

is a three-composition over the set $\{L_1, L_2, L_3\}$ of language variables.

Above we defined the Wood automata $W(n),\ n \geq 2$, and showed that the state complexity of the language $mi(L(W(n)))$ equals 2^n. We make the formal convention that the state complexity of $mi(L(W(1)))$ (resp. $mi(L(W(0)))$) equals 2 (resp. 1).

Let us go back to the E-polynomials $E_i^{(l)}$ defined in the preceding section. They use the fixed set of variables $\{x_1, \ldots, x_m\}$. As already pointed out, the variables may appear either by themselves or as exponents of 2. However, each variable x and each power 2^x appears only a bounded number of times in each product in each $E_i^{(l)}$. Thus, for each $j,\ 1 \leq j \leq m$, there is a number K_j such that every exponent of x_j in every product equals at most K_j, and that 2^{x_j} appears as a factor in every product at most K_j times, no matter what $E_i^{(l)}$ we are considering. This important fact follows because, as explained above, a change of the index i in $E_i^{(l)}$ does not affect the summands in $E_i^{(l)}$, only their multiplicities.

Consider now language variables

$$L_j^\nu,\ 1 \leq j \leq m,\ 1 \leq \nu \leq K_j.$$

The variables $L_j^\nu,\ 1 \leq \nu \leq K_j$ correspond to x_j in the sense of the mapping φ in Definition 1. With each summand

$$(2^{x_1})^{\mu_1} \cdots (2^{x_m})^{\mu_m} x_1^{\nu_1} \cdots x_m^{\nu_m},$$

where $0 \leq \mu_j, \nu_j \leq K_j,\ 1 \leq j \leq m$, in $E_i^{(l)}$, we associate an intersection as follows. (We consider an arbitrary index i. For readability, we do not include it in the notation.) Consider an arbitrary $j,\ 1 \leq j \leq m$. The part $(2^{x_j})^{\mu_j}$ is associated with the intersection

$$mi(L_j^1) \cap \ldots \cap mi(L_j^{\mu_j}).$$

The part $x_j^{\nu_j}$ is associated with the intersection

$$L_j^1 \cap \ldots \cap L_j^{\nu_j}.$$

Finally, the *three-composition* $C(E_i^{(l)})$ *associated with* $E_i^{(l)}$ is the marked catenation of the summands appearing in $E_i^{(l)}$.

As an example, consider the E-polynomial

$$(2^{x_1})^2 x_1 x_3 + 2x_1 x_2^2 x_3 + 2^{x_2}.$$

Now we have $K_1 = K_2 = 2,\ K_3 = 1$. The three-composition associated with this E-polynomial is

$$(mi(L_1^1) \cap mi(L_1^2) \cap L_1^1 \cap L_3^1)\ddagger(L_1^1 \cap L_2^1 \cap L_2^2 \cap L_3^1)$$
$$\ddagger(L_1^1 \cap L_2^1 \cap L_2^2 \cap L_3^1)\ddagger(mi(L_2^1)).$$

We will need the following result from [13]. It is proved also in [8].

Theorem 6. *Assume that* L_i, $1 \leq i \leq r$, *are regular languages with the state complexities* σ_i. *Then the state complexity of the regular language* $L_1 \cap \ldots \cap L_r$ *is at most the product* $\sigma = \sigma_1 \cdots \sigma_r$. *Moreover, for any* r*-tuple* $(\sigma_1, \ldots, \sigma_r)$ *of nonnegative integers, it is possible to construct regular languages* $K_i, 1 \leq i \leq r$, *with the state complexity* σ_i *such that the intersection of the languages* K_i *has exactly the state complexity* $\sigma = \sigma_1 \cdots \sigma_r$.

The following theorem is an immediate corollary of Theorems 1 and 6. (The additional summand +1 of Theorem 1 is not needed if we assume that at least one exponential term appears in the E-polynomial.)

Theorem 7. *For each* $i = 1, 2, \ldots$, *the* E*-polynomial* $E_i^{(l)}(x_1, \ldots, x_m)$ *is a state complexity function of the three-composition* $C(E_i^{(l)})$.

In the next section specific languages will be substituted in three-compositions in such a way that the alphabets of the languages appearing in intersections will be pairwise disjoint. (We do not estimate here the size of the total alphabet. Such an estimation was done in an analogous situation in [8].)

6 Undecidability

We consider in the sequel an arbitrary but fixed E-polynomial $E_i^{(l)}(x_1, \ldots, x_m)$, and the numbers K_j as defined above. Let $C(E_i^{(l)})$ be the three-composition associated with $E_i^{(l)}(x_1, \ldots, x_m)$. Introduce the alphabet Σ consisting of the letters

$$a_j^\nu, b_j^\nu, c_j^\nu,\ 1 \leq j \leq m,\ 1 \leq \nu \leq K_j.$$

The specific languages defined below will be over the alphabet Σ. The languages will depend on a fixed nonnegative integer n. (It will be the value assigned for the variable x_j in our E-polynomial.) By definition,

$$A_j^\nu(n) = mi(W(n)(b_j^\nu, c_j^\nu)),\ 1 \leq j \leq m,\ 1 \leq \nu \leq K_j.$$

(Recall our way of indicating the alphabet of a Wood language.) Similarly, let $B_j^\nu(n)$ be the language over Σ consisting of all words w such that the number of occurrences of the letter a_j^ν in w is divisible by n. Finally, for each n-tuple $(x_1, \ldots, x_m)$ of nonnegative integers, let $D_i(x_1, \ldots, x_m)$ be the regular language, resulting from $C(E_i^{(l)})$ as follows. If n_j is the value assigned for x_j, $1 \leq j \leq m$, substitute every occurrence of $mi(L_j^\nu)$ (resp. L_j^ν) with $A_j^\nu(n_j)$ (resp. $B_j^\nu(n_j)$).

Consider now the example

$$(mi(L_1^1) \cap mi(L_1^2) \cap L_1^1 \cap L_3^1) \ddagger (L_1^1 \cap L_2^1 \cap L_2^2 \cap L_3^1)$$
$$\ddagger (L_1^1 \cap L_2^1 \cap L_2^2 \cap L_3^1) \ddagger (mi(L_2^1)).$$

from the preceding section, as well as the ordered triple $(8, 6, 5)$ as values of the three variables. Then the associated D_i-languages is

$$\begin{aligned}(mi(W(8)(b_1^1, c_1^1)) \cap mi(W(8)(b_1^2, c_1^2)) \cap B_2^1(6) \cap B_3^1(5))\\ \ddagger(B_1^1(8) \cap B_2^1(6) \cap B_2^2(6) \cap B_3^1(5))\\ \ddagger(B_1^1(8) \cap B_2^1(6) \cap B_2^2(6) \cap B_3^1(5))\ddagger(mi(W(6)(b_2^1, c_2^1))).\end{aligned}$$

The proof of the following lemma is straightforward, along the lines of Theorems 1 and 6. The only additional technicality needed is to take care of the possible sinks appearing in automata for the mirror images of the languages of Wood automata. Since the alphabets are pairwise disjoint, we can introduce transitions in such a way that sinks can never be combined, and there is only one sink where the wrong transitions, including the ones involving ‡, are leading.

Lemma 2. *For all values of the variables, the state complexity of the language* $D_i(x_1, \ldots, x_m)$ *equals* $E_i^{(l)}(x_1, \ldots, x_m)$.

Our undecidability result now follows by Theorems 5 and 7 and Lemma 2.

Theorem 8. *For the sequence of E-polynomials* $E_i^{(r)}$, $i = 1, 2, \ldots$ *and three-compositions* $C(E_i^{(l)})$, $i = 1, 2, \ldots$, *as constructed above, it is undecidable whether or not* $E_i^{(r)}$ *is a state complexity function of* $C(E_i^{(l)})$.

7 Conclusion

We have investigated the operation *mirror image*, in particular, the cases where the state complexity of the language $mi(L)$ is maximal in comparison with the state complexity of L. This gives also the maximal increase in state complexity in the transition from a nondeterministic automaton to the equivalent deterministic automaton. Wood automata $W(n)$ with n states constitute good examples. If n is divisible by 4, the state complexity of $mi(W(n))$ is, for certain choices of the final state set, maximal but sometimes only $2^n - 4$. Several open problems remain in connection with mirror images, in particular, the construction of automata for languages L such that the state complexity of $mi(L)$ is close to the maximal, or close to the minimal one.

In our result concerning the undecidability of the state complexity of compositions of regular languages, we were able to use reduction to exponential polynomials, instead of polynomials. The three operations in the compositions were mirror image, intersection and marked catenation. The Davis-Putnam-Robinson Theorem provided the undecidable problem used as the basis of reduction. Other undecidable problems will in general lead to other operations. The undecidability of the state complexity will then concern composition sequences in terms of these operations. It is an interesting open problem to study the possibilities in this direction. For instance, is it possible to use the undecidability of the Post Correspondence Problem and, if this is the case, which are the regularity-preserving operations involved?

In our undecidability result above the state complexities of the languages appearing as components of the marked catenations depended on the values for the variables in a rather complicated way. On the other hand, we can study simpler cases. If the state complexity of each of the component languages L_j equals directly the value of one of the variables x_j, then the state complexity of the marked catenation is a linear function of the variables, and our problem is clearly decidable, provided an E-polynomial is the proposed state complexity function. This result can possibly be extended to the case where the state complexity of each component language is of the form x_j^t, where t is a positive integer. Each pair $((\mathcal{C}, \mathcal{F}))$, where $\mathcal{C}$ (resp. $\mathcal{F}$) is a class in compositions (resp. functions) defines in the natural way a decision problem. A general task is to find interesting pairs for which this problem is decidable.

References

1. Esik, Z., Gao, Y., Liu, G., Yu, S.: Estimation of State Complexity of Combined Operations. Theoretical Computer Science 410, 3272–3280 (2009)
2. Gao, Y., Salomaa, K., Yu, S.: State complexity of catenation and reversal combined with star. In: Descriptional Complexity of Formal Systems (DCFS 2006), pp. 153–164 (2006)
3. Rozenberg, G., Salomaa, A.: Cornerstones of Undecidability. Prentice Hall, New York (1994)
4. Rozenberg, G., Salomaa, A. (eds.): Handbook of Formal Languages, vol. 1-3. Springer, Heidelberg (1997)
5. Salomaa, A.: Composition sequences for functions over a finite domain. Theoretical Computer Science 292, 263–281 (2003)
6. Salomaa, A.: Mirror images and schemes for the maximal complexity of nondeterminism. Fundamenta Informaticae (to appear)
7. Salomaa, A., Salomaa, K., Yu, S.: State Complexity of Combined Operations. Theoretical Computer Science 383, 140–152 (2007)
8. Salomaa, A., Salomaa, K., Yu, S.: Undecidability of the State Complexity of Composed Regular Operations. In: Dediu, A.-H., Inenaga, S., Martín-Vide, C. (eds.) LATA 2011. LNCS, vol. 6638, pp. 489–498. Springer, Heidelberg (2011)
9. Salomaa, A., Wood, D., Yu, S.: On the state complexity of reversals of regular languages. Theoretical Computer Science 320, 293–313 (2004)
10. Salomaa, K., Yu, S.: On the state complexity of combined operations and their estimation. International Journal of Foundations of Computer Science 18, 683–698 (2007)
11. Stanković, R.S., Astola, J.T. (eds.): Reprints from the Early Days of Information Sciences: On the Contributions of Arto Salomaa to Multiple-Valued Logic. Tampere International Center for Signal Processing, TICSP Series 50 (2009)
12. Yu, S.: On the State Complexity of Combined Operations. In: Ibarra, O.H., Yen, H.-C. (eds.) CIAA 2006. LNCS, vol. 4094, pp. 11–22. Springer, Heidelberg (2006)
13. Yu, S., Zhuang, Q., Salomaa, K.: The state complexities of some basic operations on regular languages. Theoretical Computer Science 125, 315–328 (1994)