ISSN 0032-9460, Problems of Information Transmission, 2015, Vol. 51, No. 3, pp. 267–288. © Pleiades Publishing, Inc., 2015.
Original Russian Text © An.A. Muchnik, K.Yu. Gorbunov, 2015, published in Problemy Peredachi Informatsii, 2015, Vol. 51, No. 3, pp. 70–92.

Algorithmic Aspects of Decomposition and Equivalence of Finite-Valued Transducers

An. A. Muchnik[†] **and K. Yu. Gorbunov**[a,1]

[a]*Kharkevich Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow, Russia*
e-mail: gorbunov@iitp.ru

Received February 12, 2014; in final form, June 3, 2015

Abstract—We study algorithmic issues of the problems of decomposing a finite-valued transducer into a union of single-valued ones and inclusion of an arbitrary transducer in a finite-valued one. We propose algorithms that partially improve efficiency estimates for known analogous algorithms.

DOI: 10.1134/S0032946015030059

1. INTRODUCTION

In the present paper we address algorithmic issues of the problems of decomposing a finite-valued transducer into a union of single-valued ones and inclusion of an arbitrary transducer in a finite-valued one. These questions were studied by Weber in [1–4] and also by Sakarovitch and de Souza in [5–8]. In these works polynomial decidability of finite-valuedness of a transducer was proved, possibility of decomposing a finite-valued transducer into a union of single-valued ones was shown, and an algorithm for testing inclusion of an arbitrary transducer in a finite-valued one was proposed. We propose simpler constructions, which partially improve estimates of the previous authors. The single-valued transducers obtained in the decomposition have sizes of the order of a single exponent of $\mathrm{poly}(n)$, where n is the size of the finite-valued transducer; testing inclusion of an arbitrary transducer in a finite-valued one requires space estimated by a single exponent. Note that in [5,6] the corresponding estimate is exponential not only in n but also in k, where k is valuedness of the transducer (in those works k was assumed to be constant). Taking into account that k can itself be exponential in n, this bound has the order of a double exponent of n. Our estimate does not contain k under the exponent and thus improves the previously known ones. The same concerns the problem of testing inclusion of an arbitrary transducer in a finite-valued one, which in [8] is solved with space estimation of the order of $\exp(\mathrm{poly}(n,k))$, whereas our estimate is of the order of $\exp(\mathrm{poly}(n))$. On the other hand, the number of single-valued transducers in constructions from [4–6] equals the valuedness of a given finite-valued transducer, which does not follow from our construction. Thus, each of the constructions have its own advantages. Note also the work [9], which contains a detailed presentation of automata and transducers theory.

Section 2 contains basic definitions, examples, and facts introducing the reader to the subject of the paper. In Section 3 we present constructions underlying the decomposition of a finite-valued transducer. Section 4 describes the decomposition itself. In Section 5 we address questions of the smallest input and output length on which noninclusion of one finite-valued transducer

[†] Deceased.
[1] The research was carried out at the Institute for Information Transmission Problems of the Russian Academy of Sciences at the expense of the Russian Science Foundation, project no. 14-50-00150.

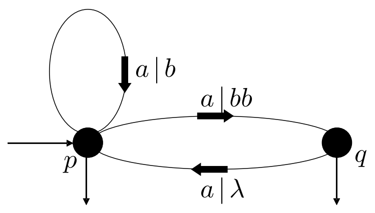

Fig. 1. Two-valued transducer; to the input of i symbols a there correspond two outputs: of i symbols b and of $(i+1)$ symbols b.

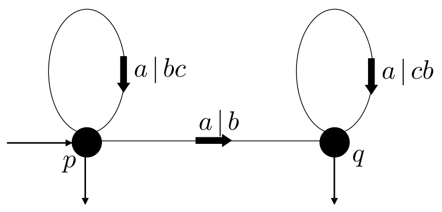

Fig. 2. Two-valued transducer; to the input of i symbols a there correspond two outputs consisting of alternating symbols b and c: both begin with b, but one ends with c and the other ends with b.

in another can be ascertained. Finally, Section 6 presents results on algorithmic testing of such noninclusion.

2. FINITE-VALUED TRANSDUCERS

A finite nondeterministic transducer $\mathfrak{A}$ is a sextuple $\langle A, B, Q, Q_0, F, \delta\rangle$, where A is a (finite) input alphabet, B an output alphabet, Q a finite set of states, $Q_0 \subseteq Q$ a set of initial states, $F \subseteq Q$ a sets of final states, and δ a set of transitions. Each transition is a quadruple $\langle q_1, a, v, q_2\rangle$, where q_1 is a state before transition, q_2 a state after transition, $a \in A \cup \{\lambda\}$ (λ is the empty string) a transition input, and $v \in B^*$ a transition output. We represent $\mathfrak{A}$ as a directed graph whose vertices are states and edges are transitions. Each edge is labeled with the input and output of the corresponding transition. By paths in this graph we mean directed paths. A word obtained by concatenation of inputs along some path l will be referred to as the input of l and denoted by $\mathrm{in}(l)$; a word obtained by concatenation of outputs will be called the output of l and denoted by $\mathrm{out}(l)$. A path starting in an input state and ending in an output state is said to be accepting. The graphic $\Gamma(\mathfrak{A})$ of a transducer $\mathfrak{A}$ is the set of pairs $\langle u, v\rangle$ such that $u = \mathrm{in}(l)$ and $v = \mathrm{out}(l)$ for some accepting path l. We say that a transducer $\mathfrak{A}_1$ is included in a transducer $\mathfrak{A}_2$ if $\Gamma(\mathfrak{A}_1) \subseteq \Gamma(\mathfrak{A}_2)$. Transducers $\mathfrak{A}_1$ and $\mathfrak{A}_2$ are said to be equivalent if $\Gamma(\mathfrak{A}_1) = \Gamma(\mathfrak{A}_2)$. The size $|\mathfrak{A}|$ of a transducer $\mathfrak{A}$ is the sum of the number of its states, transitions, and lengths of their outputs. Clearly, given a transducer $\mathfrak{A}$, in time polynomial in $|\mathfrak{A}|$ one can construct an equivalent transducer in which through each state there passes at least one accepting path. Therefore, throughout what follows we assume that all transducers that we consider possess this property.

A basic definition for further presentation is as follows.

Definition. A transducer $\mathfrak{A}$ is said to be *finite-valued* if there exists a constant c such that for any word u there exists no more than c different words v such that $\langle u, v\rangle \in \Gamma(\mathfrak{A})$. The smallest of such constants c is called the *valuedness* of $\mathfrak{A}$. If the valuedness equals 1, then $\mathfrak{A}$ is *single-valued.*

For example, the transducers shown in Figs. 1 and 2 are two-valued, and the one shown in Fig. 3 is four-valued.

Valuedness of a transducer may be exponential in its size. As the following example shows, even the number of different lengths of outputs for one input can be exponential.

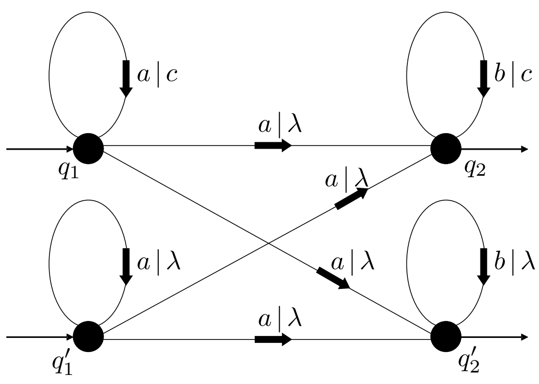

Fig. 3. Four-valued transducer; to the input $aaab$ there correspond four outputs: λ, c, cc, and ccc.

Example 1. Consider a transducer $\mathfrak{B}$ with $m+1$ pairs of states: $(q_1, q_1'), (q_2, q_2'), \ldots, (q_{m+1}, q_{m+1}')$. Initial states are q_1 and q_1'; final states are q_{m+1}, q_{m+1}'. Transitions are as follows:

$$\begin{array}{lll} \langle q_i, u_i, \lambda, q_{i+1}\rangle, & \langle q_i, u_i, \lambda, q_{i+1}'\rangle, & \langle q_i', u_i, \lambda, q_{i+1}\rangle, \\ \langle q_i', u_i, \lambda, q_{i+1}'\rangle, & \langle q_i, u_i, c, q_i\rangle, & \langle q_i', u_i, \lambda, q_i'\rangle, \end{array}$$

where $u_i = a$ for odd i and $u_i = b$ for even i; a, b, and c are symbols (Fig. 3 shows the transducer $\mathfrak{B}$ for $m = 2$). Clearly, $\mathfrak{B}$ is finite-valued. Let $f(k) = 2^{m-k}$. Then, for the input $a^{f(0)} a\, b^{f(1)} b\, a^{f(2)} a\, b^{f(3)} b \ldots a^2 a\, bb$ (let m be even), the transducer $\mathfrak{B}$, obviously, can produce any output c^k where the number k written in binary notation contains at most m binary digits. In total, there are 2^m such numbers, which is exponential in $|\mathfrak{B}|$.

We say that a transducer $\mathfrak{A}$ is *reduced* if has exactly one initial and one final state and all paths with empty input lead from the initial state to the final. In [1], the following statement is proved.

Lemma 1. *Given a transducer $\mathfrak{A}$, in polynomial time one can either construct a reduced transducer equivalent to it or ascertain infinite-valuedness of $\mathfrak{A}$.*

Proof. If the transducer $\mathfrak{A}$ is finite-valued, then it obviously has no cycles with empty input and nonempty output. Clearly, checking the absence of such cycles is performed in polynomial time. If this condition is not fulfilled, infinite-valuedness of $\mathfrak{A}$ is ascertained. Assume that it is fulfilled.

By adding no more than two states and $2|\mathfrak{A}|$ transitions, we can obviously assure that $\mathfrak{A}$ contains exactly one initial state with only outgoing transitions and exactly one final state with only incoming transitions. Next, on states of $\mathfrak{A}$ we define a transitive relation R: $q_1 R q_2$ if there exists a path from q_1 to q_2 with empty input. Assume that $q_1 R q_2$ and $q_2 R q_1$. Then any path from q_1 to q_2 with empty input must have empty output. We can naturally glue each maximal set of states on which R is identically true into a single state. Then we delete all loop transitions with empty input and empty output. We obtain a transducer $\mathfrak{A}'$, which is obviously equivalent to $\mathfrak{A}$. Assume that in $\mathfrak{A}'$ there is a noninitial and nonfinal state q such that there exists a transition with empty input which is either incoming to or outgoing from q. Now we perform the following reduction operation. For each pair of transitions $\langle q_1, a_1, v_1, q\rangle, \langle q, a_2, v_2, q_2\rangle$ such that either $a_1 = \lambda$ or $a_2 = \lambda$, we add the transition $\langle q_1, a_1 a_2, v_1 v_2, q_2\rangle$. After that we delete all transitions with empty input incoming to and outgoing from q. Clearly, the reduction operation does not change the graphic of the transducer. Moreover, if before the reduction for some state q' there were no incident transitions with empty input, there will be no after the reduction as well. Therefore, in no more than $|Q|$ reduction operations we will obtain the desired reduced transducer. $\triangle$

It is easily seen that, for all problems considered below, existence of transitions with empty input in a reduced transducer is not essential. Therefore, we will assume that no such transitions exist.

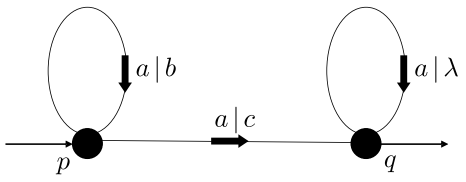

Fig. 4. Infinite-valued transducer: the greater the input length, the more variants for the output.

Remark 1. A reduced transducer can be infinite-valued. For instance, such is the transducer shown in Fig. 4.

Let us state a criterion for finite-valuedness of a transducer (Theorem 1). Note that a similar criterion is stated and proved in [1].

Theorem 1. *A transducer $\mathfrak{A}$ of size n is finite-valued if and only if for any two of its states s_1 and s_2 (not necessarily distinct) and any three of its paths p_1, p_2, and p_3 with the same input u such that p_1 starts in s_1 and ends in s_1, p_2 starts in s_1 and ends in s_2, and p_3 starts in s_2 and ends in s_2 the following two conditions hold*:

(1) *If* $\mathrm{out}(p_1) \neq \lambda$, *then* $\mathrm{out}(p_2)$ *is a head (initial segment) of an infinite word* $(\mathrm{out}(p_1))^\infty$;

(2) *For any $u' \subseteq u$ (i.e., u' is a head of u) we have $|d(p_1, p_2, u')| \leq n^4$, where $d(p_1, p_2, u')$ is the difference of output lengths of heads of p_1 and p_2 on input u'.*

Example 2. For the transducer shown in Fig. 4, condition (1) is not fulfilled, though condition (2) holds. If we replace the c in it with λ, condition (2) will be violated, but condition (1) will hold. In both cases the transducer is infinite-valued.

Let us prove the necessity of this criterion. Let $\mathfrak{A}$ be a finite-valued transducer.

First we prove condition (1). Let $k > c$, where c is the valuedness of $\mathfrak{A}$. Consider the set $\{p_1^{i-1} p_2 p_3^{k-i} \mid i = 1, \ldots, k\}$ of paths from s_1 to s_2. It is easily seen that all these paths have the same input u^k. The choice of k guarantees existence of $j > 0$ such that $(\mathrm{out}(p_1))^j \,\mathrm{out}(p_2) = \mathrm{out}(p_2)(\mathrm{out}(p_3))^j$. Hence,

$$\begin{aligned}(\mathrm{out}(p_1))^{2j}\,\mathrm{out}(p_2) &= (\mathrm{out}(p_1))^j\,\mathrm{out}(p_2)(\mathrm{out}(p_3))^j \\ &= \mathrm{out}(p_2)(\mathrm{out}(p_3))^j(\mathrm{out}(p_3))^j \\ &= \mathrm{out}(p_2)(\mathrm{out}(p_3))^{2j},\end{aligned}$$

and similarly $(\mathrm{out}(p_1))^{tj}\,\mathrm{out}(p_2) = \mathrm{out}(p_2)(\mathrm{out}(p_3))^{tj}$ for any t, which shows that $\mathrm{out}(p_2)$ is a head of $(\mathrm{out}(p_1))^\infty$.

Condition (2) is proved by contradiction. Assume that $|d(p_1, p_2, u')| > n^4$ for some $u' \subseteq u$. Let $\lambda = u_0, u_1, u_2, \ldots$ be all initial segments of u. Note that $|d(p_1, p_2, u_i) - d(p_1, p_2, u_{i+1})| \leq n$ for any i and that $d(p_1, p_2, u_0) = 0$; hence, among u_i there are at least $n^3 + 1$ initial segments u_{i_k} with different $d(p_1, p_2, u_{i_k})$. To each u_{i_k} there corresponds a triple of states at which the paths $p_1(u_{i_k})$, $p_2(u_{i_k})$, and $p_3(u_{i_k})$ end. Choose $u_{i'}$ and $u_{i''}$ ($u_{i'} \subset u_{i''}$) to which the same triple corresponds. Thus, $u = v_1 v_2 v_3$, where $v_1 = u_{i'}$ and $v_1 v_2 = u_{i''}$ (see Fig. 5).

By the construction we have $|\mathrm{out}(p_1(v_2))| \neq |\mathrm{out}(p_2(v_2))|$, since

$$\begin{aligned}d(p_1, p_2, v_1) &= |\mathrm{out}(p_1(v_1))| - |\mathrm{out}(p_2(v_1))| \\ &\neq |\mathrm{out}(p_1(v_1))| + |\mathrm{out}(p_1(v_2))| - |\mathrm{out}(p_2(v_1))| - |\mathrm{out}(p_2(v_2))| = d(p_1, p_2, v_1 v_2).\end{aligned}$$

If $|\mathrm{out}(p_1)| \neq |\mathrm{out}(p_3)|$, then the lengths of outputs of the paths $p_1^{i-1} p_2 p_3^{k-i}$ are obviously pairwise distinct, which contradicts the choice of k. If

$$|\mathrm{out}(p_1(v_1))| + |\mathrm{out}(p_1(v_3))| \neq |\mathrm{out}(p_3(v_1))| + |\mathrm{out}(p_3(v_3))|,$$

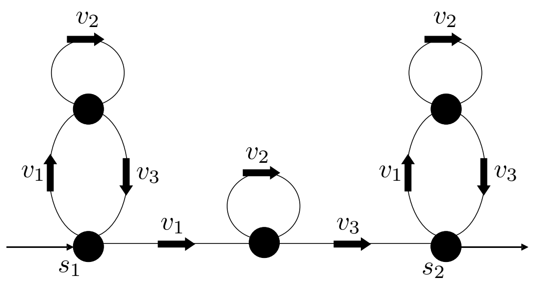

Fig. 5. Scheme of paths in the proof of Theorem 1. Only inputs are shown.

then the same contradiction is obtained for the paths $[p_1(v_1)p_1(v_3)]^{i-1}p_2(v_1)p_2(v_3)[p_3(v_1)p_3(v_3)]^{k-i}$. If equalities hold in both cases, consider different paths from s_1 to s_2 with input $v_1v_2v_3v_1v_2v_2v_3\ldots v_1v_2^kv_3$ passing through p_1, p_2, and p_3. It is easily seen that the length of the output of these paths monotonically depends of which block $v_1v_2^iv_3$ corresponds to the path p_2, which contradicts the choice of k. Condition (2) and the necessity of the criterion are proved. The sufficiency will be proved in Section 4, together with Theorem 3.

The following theorem was proved in [1].

Theorem 2. *There exists a polynomial algorithm which, given an arbitrary transducer, decides whether it is finite-valued or not.*

Proof. This result follows from Lemma 1, finite-valuedness criterion (Theorem 1), and the fact that, given an arbitrary transducer $\mathfrak{A}$ without empty inputs, it is possible to decide in polynomial time whether it satisfies the criterion. Let us show how this can be done. We look through pairs of states $\langle s_1, s_2\rangle$. For each pair we first check whether condition (2) holds. For that, we construct the following nondeterministic automaton A_1. Its states are order quadruples $\langle q_1, q_2, q_3, m\rangle$, where q_1, q_2, and q_3 are states of $\mathfrak{A}$, and m is either an integer such that $|m| \le n^4$ or the symbol $*$. A transition from a state $\langle q_1, q_2, q_3, m\rangle$ to a state $\langle q_1', q_2', q_3', m'\rangle$ with input a exists if and only if, first, for each $i = 1, 2, 3$ there exists a transition in $\mathfrak{A}$ from q_i to q_i' with input a (denote its output by v_i) and, second, if m is an integer and $|m + |v_1| - |v_2|| \le n^4$, then $m' = m + |v_1| - |v_2|$, and otherwise $m' = *$. The initial state of A_1 is $\langle s_1, s_1, s_2, 0\rangle$, and the final state is $\langle s_1, s_2, s_2, *\rangle$. In m we compute the difference between the lengths of inputs of p_1 and p_2, and $*$ indicates that its absolute value exceeds n^4.

Clearly, the existence of an accepting path in A_1 implies existence of paths p_1, p_2, and p_3 from the criterion such that on some initial segment u' of their common input we have $|d(p_1, p_2, u')| > n^4$. Conversely, the existence of such paths obviously implies existence of an accepting path in A_1. Thus, fulfillment of criterion (2) for s_1 and s_2 is equivalent to the nonexistence of an accepting path in A_1, which, clearly, can be verified in polynomial time.

We say that two words are *matching* if one of them is a head of another.

Now let us check condition (1) for s_1 and s_2 provided that condition (2) for them is fulfilled. First note that conditions (1) is equivalent to the following condition (1′): the words $\mathrm{out}(p_1)$ and $\mathrm{out}(p_2)$ are matching. Indeed, if (1′) does not hold, then (1), clearly, does not hold either. Conversely, if (1) is violated for p_1, p_2, and p_3, then condition (1′) is violated for the paths $p_1' = p_1^k$, $p_2' = p_2p_3^{k-1}$, and $p_3' = p_3^k$ with k large enough.

Now we construct the following automaton A_2. Its states are divided into three sets: Q_1, Q_2, and Q_3. States in Q_1 are quintuples $\langle q_1, q_2, q_3, m, a\rangle$, where q_1, q_2, and q_3 (as in A_1) correspond to last states of the guessed paths p_1, p_2, and p_3; in m, the difference of lengths of outputs of p_1 and p_2 is computed. It is required that $|m| \le n^4$; if this condition is violated, the transition does not exist. In a, the last symbol of the path among p_1 and p_2 whose output at the current moment is strictly longer (if $|m| > 0$) is computed in a natural way; if $m = 0$, then $a = \lambda$. The initial state is $\langle s_1, s_1, s_2, 0, \lambda\rangle$, and there are no final states in Q_1.

When being in an arbitrary state $\langle q_1, q_2, q_3, m, a\rangle \in Q_1$ ($m \neq 0$), A_2 may "assume" that precisely in this symbol a a mismatching of $\mathrm{out}(p_1)$ and $\mathrm{out}(p_2)$ will occur (let us call it a failure) and pass (with empty input and empty output) to a state from Q_2 of the form $\langle q_1, q_2, q_3, a, k\rangle$, where k is at first equal to m. In q_1, q_2, and q_3 the usual information is computed; a is the symbol of the assumed failure; k indicates by how many symbols the "short" output must be enlarged to check if a failure will indeed occur in this place. Transitions between states of Q_2 obviously reduce $|k|$, while the "short" output is enlarged, and the sign of k indicates the path with the "short" output. When in the "short" output the symbol occurs in which by the assumption there should be failure, it is checked whether this symbol actually does not equal a. If so, A_2 can pass to a state from Q_3 of the form $\langle q_1, q_2, q_3\rangle$. If, when being in a state $\langle q_1, q_2, q_3, 0, \lambda\rangle \in Q_1$, A_2 detects a mismatching of outputs of p_1 and p_2 in a current transition, it can also pass to $\langle q_1', q_2', q_3'\rangle \in Q_3$. When A_2 is in a state from Q_3, this means that a failure has occurred and it remains to complete the construction of the paths p_1, p_2, and p_3. Transitions in Q_3 are defined in a natural way; the final state is $\langle s_1, s_2, s_2\rangle$. It is clear that condition $(1')$ is satisfied for s_1 and s_2 if and only if there is no accepting state in A_2. $\triangle$

By Theorem 2, the question of inclusion of an arbitrary transducer in a finite-valued one reduces in polynomial time to the question of inclusion of one finite-valued transducer in another. Indeed, check finite-valuedness of the first transducer. If it is infinite-valued, it obviously cannot be included in a finite-valued one.

A particular case of single-valued transducers are automata, i.e., transducers with empty input. Even for them, it is possible that noninclusion can be detected on an exponentially long input only. This immediately follows from the fact that for any n there exists a nondeterministic automaton A of size $\mathrm{poly}(n)$ which does not accept some word w of length $\exp(\mathrm{poly}(n))$ but accepts all words of smaller lengths. Indeed, take for w the concatenation of strings representing all n-digit binary numbers (low-order bits being written on the left) arranged in ascending order, starting from the all-zero string and ending with the all-one string. When operating, the automaton A "guesses" the reason for why the word being read is not equal to w. The reasons can be as follows: the first string is not all-zero; some of subsequent numbers does not equal the preceding number increased by 1 (in this case the position is guessed which does not match w in the next number); the length of the whole word is not a multiple of n; the last string is not all-one. In the states of A (before guessing) there are stored the number of the last read digit, its value, and information on whether in the current number there are zeros to the left of this digit. This information is sufficient to determine the corresponding digit in the next word. Further details are obvious.

3. PATH DIAGRAMS AND THEIR PROPERTIES

For decomposition of a finite-valued transducer, we are going to assign to each accepting path in it a certain information which is not too large, admits not too many variants, but at the same time uniquely determines the output for a given input. Here we describe the corresponding constructions and prove necessary lemmas.

Let us be given a transducer $\mathfrak{A}$ satisfying the finite-valuedness criterion of Theorem 1. Let q_1 and q_2 be states of $\mathfrak{A}$. We say that $q_1 \geq q_2$ if there exists a path from q_1 to q_2. We say that states q_1 and q_2 are equivalent if $q_1 \geq q_2$ and $q_2 \geq q_1$. Then the state set Q is split into equivalence classes of states. A transition $\langle q_1, a, v, q_2\rangle$ will be called a *state transition* if q_1 is not equivalent to q_2.

Lemma 2. *Let two states q_1 and q_2 in $\mathfrak{A}$ be equivalent. Then for any two paths l_1 and l_2 from q_1 to q_2 such that* $\mathrm{in}(l_1) = \mathrm{in}(l_2)$ *we have* $\mathrm{out}(l_1) = \mathrm{out}(l_2)$. *For any head u' of the word $u = \mathrm{in}(l_1)$ we have $|d(l_1, l_2, u')| \leq n^4$.*

Proof. Denote $w_1 = \mathrm{out}(l_1)$ and $w_2 = \mathrm{out}(l_2)$. Since q_1 and q_2 are equivalent, there is a path l from q_2 to q_1. If $|w_1| = |w_2|$, apply condition (1) of Theorem 1 with $s_1 = s_2 = q_1$, $p_1 = l_1 l$, and $p_2 = p_3 = l_2 l$. We obtain $\mathrm{out}(l_1 l) = \mathrm{out}(l_2 l)$, which implies $w_1 = w_2$. Let $|w_1| \neq |w_2|$. Let us assume that $|w_1| > 0$. Then we obtain a contradiction with condition (2) by letting $s_1 = s_2 = q_1$, $p_1 = (l_1 l)^k$, and $p_2 = p_3 = (l_2 l)^k$ for $k > n^4$. The last claim of the lemma follows from condition (2) of Theorem 1. △

Let Q_1 and Q_2 be subsets of Q. We say that Q_2 is accessible from Q_1 on a word w if there exists a set L of paths from Q_1 to Q_2 with input w such that for any $q_1 \in Q_1$ there exists a path in L starting from q_1 and for any $q_2 \in Q_2$ there exists a path in L ending in q_2. We say that $Q_1 \geq Q_2$ if there exists a word on which Q_2 is accessible from Q_1. Clearly, the introduced relation is transitive and extends the relation already introduced on words: $q_1 \geq q_2$ if and only if $\{q_1\} \geq \{q_2\}$. We say that sets Q_1 and Q_2 are equivalent if $Q_1 \geq Q_2$ and $Q_2 \geq Q_1$. Then the set Q of subsets is split into equivalence classes of subsets.

Lemma 3. *Given two subsets Q_1 and Q_2, it is decidable in time* $\exp(\mathrm{poly}(n))$ *whether $Q_1 \geq Q_2$ or not.*

Proof. Let us construct an automaton that checks whether Q_2 is accessible from Q_1. Its states are tuples of subsets of Q of length $m = |Q_1|$. Each subset corresponds to "its own" element of Q_1; the initial state is the tuple of one-element subsets of "their own" elements. A transition with input a leads from a tuple $\langle P_1, P_2, \ldots, P_m\rangle$ to a tuple $\langle R_1, R_2, \ldots, R_m\rangle$ if for each i all transitions from P_i with input a lead to R_i and for each state in R_i there exists a transition from P_i with input a leading to it. Final states of the automaton are tuples $\langle R_1, R_2, \ldots, R_m\rangle$ such that each R_i has a nonempty intersection with Q_2, and Q_2 is contained in the union of all the R_i. Clearly, $Q_1 \geq Q_2$ if and only if there exists an accepting path in the automaton. It is easily seen that constructing this automaton and checking this condition requires no more than exponential time. △

Let us have a word u, and let $u' \subseteq u$. The *double-sided accessibility* set $M_u(u')$ is the set of all states q such that there exists an accepting path l with $\mathrm{in}(l) = u$ and the path $l(u')$ ends in q. It is easily seen that if $u_1 \subseteq u_2 \subseteq u$, then $M_u(u_1) \geq M_u(u_2)$. Let there be a transition p on an accepting path l; let l_1 be the head of l ending directly before p, and let l_2 be the head of l ending immediately after p. We say that p is a *set transition* on l if $M_u(u_1)$ is not equivalent to $M_u(u_2)$, where $u_1 = \mathrm{in}(l_1)$ and $u_2 = \mathrm{in}(l_2)$.

Lemma 4. *Let M_1 and M_2 be arbitrary equivalent sets of states, $q_1 \in M_1$, $q_2 \in M_2$, and let l' be a path from q_1 to q_2 such that M_2 is accessible from M_1 on the word $v = \mathrm{in}(l')$ and $|\mathrm{out}(l')| > n^4$. Then for any two paths l_1 and l_2 from q_1 to q_2 with input v the words* $\mathrm{out}(l_1)$ *and* $\mathrm{out}(l_2)$ *are matching, and for any $v' \subseteq v$ we have $|d(l_1, l_2, v')| \leq 2n^4$.*

Proof. let us assume that q_1 is not equivalent to q_2, since otherwise the claim immediately follows from Lemma 2. Since M_1 and M_2 are equivalent, M_1 is accessible from M_2 on some word v_1. Then M_1 is accessible from itself on the word vv_1. Therefore, there exists an infinite sequence $q_0, q_{-1}, q_{-2} \ldots$ of states in M_1 such that for any $i \leq 0$ there is a path γ_i from q_i to q_{i+1} with input vv_1. Fix a state $b_1 = q_i = q_j$ for some $i < j \leq 1$. We obtain a closed path from b_1 to b_1; denote it by p (Fig. 6). Furthermore, there exists an infinite sequence $q_3, q_4, \ldots$ of states in M_1 such that there is a path γ_2 from q_2 to q_3 with input v_1 and for any $i \geq 3$ there is a path γ_i from q_i to q_{i+1} with input vv_1. Here we have $q_i \neq q_j$ for all $i \leq 1$ and $j \geq 2$, since q_1 and q_2 are nonequivalent. By fixing $b_2 = q_i = q_j$ for some $3 \leq i < j$, we obtain a closed path from b_2 to b_2; denote it by p' (Fig. 6).

Denote by γ the path from b_1 to b_2 consisting of three segments: from b_1 to q_1 through paths γ_i ($i \leq 0$), from q_1 to q_2 through l', and from q_2 to b_2 through paths γ_i ($i \geq 2$). Clearly, $|\mathrm{in}(p)| = kn_1$, $|\mathrm{in}(\gamma)| = kn_2$, and $|\mathrm{in}(p')| = kn_3$, where $k = |vv_1|$ and n_1, n_2, n_3 are natural numbers. Choose a

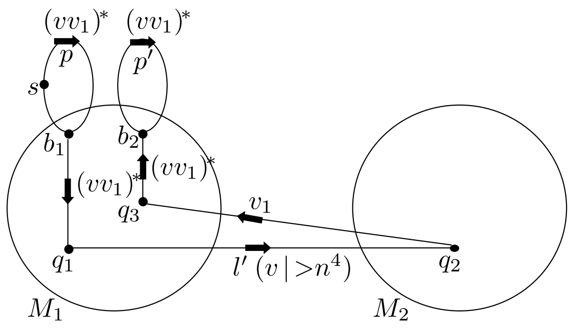

Fig. 6. Scheme of paths in the proof of Lemma 4. Symbol $*$ stands for natural numbers (which may be different in different places).

state s on p such that the path starting at s, going along the cycle p all the time, and having the length $k(n_1n_2n_3 - n_2)$ ends in b_1; denote this path by γ'. Consider three paths: path p_1 from s to s making n_2n_3 turns around the cycle p, path p_2 from s to b_2 equal to $\gamma'\gamma$, and path p_3 from b_2 to b_2 making n_1n_2 turns around p'. Clearly, we have $\mathrm{in}(p_1) = \mathrm{in}(p_2) = \mathrm{in}(p_3) = kn_1n_2n_3$. Since $|\mathrm{out}(p_2)| \geq |\mathrm{out}(l')| > n^4$, by condition (2) of Theorem 1 we have $|\mathrm{out}(p_1)| > 0$. The path p_2 can be drawn along l_1 or l_2 instead of l'. By condition (1), outputs of any of these three variants of p_2 are heads of the word $(\mathrm{out}(p_1))^\infty$. Hence it follows that the words $\mathrm{out}(l_1)$ and $\mathrm{out}(l_2)$ are matching. It is easily seen from condition (2) that $|d(l_1, l_2, v')| \leq 2n^4$. $\triangle$

If l is an accepting path in $\mathfrak{A}$ with input u and l_1 is its head with input u_1, we will call $M_u(u_1)$ the two-sided accessibility set of the path l with respect to l_1 and denote it by $M(l, l_1)$. An obvious consequence of Lemma 4 is as follows.

Corollary. *Let l be an accepting path in $\mathfrak{A}$, let l_1 and l_2 be two its heads (l_1 being shorter), and let $M(l, l_1)$ be equivalent to $M(l, l_2)$. Denote by l' the segment of l complementing l_1 to l_2, and denote by q_1 and q_2 the initial and final states of l', respectively. Let $|\mathrm{out}(l')| > n^4$. Then for any two paths p_1 and p_2 from q_1 to q_2 with input $v = \mathrm{in}(l')$ the words* $\mathrm{out}(p_1)$ *and* $\mathrm{out}(p_2)$ *are matching, and for any $v' \subseteq v$ we have $|d(p_1, p_2, v')| \leq 2n^4$.*

Let l be an accepting path in $\mathfrak{A}$. To each head l' of l there corresponds a pair (q, M), where $q \in M$ is the state in which l' ends and $M = M(l, l')$. Let us mark pairs corresponding to some heads of l. First we mark the pairs corresponding to the empty head and to the whole path. For each state transition on l that is not a set transition on l, take the nearest from the left and right set transitions on l (we assume that l is directed rightwards), if they exist. For each of these two transitions, we mark two pairs corresponding to the heads of l ending directly before the transition and immediately after it. For each state transition that is also a set transition on l, we mark two such pairs corresponding to this transition.

Denote by D the sequence of all marked pairs arranged in ascending order of the corresponding heads. One can easily see that for any two neighboring pairs (q_1, M_1) and (q_2, M_2) in D the following three cases are possible:

1. q_1 is equivalent to q_2;
2. Heads of l corresponding to these pairs differ by one transition of l which is both a state and set transition;
3. M_1 is equivalent to M_2, but q_1 is not equivalent to q_2.

If the ith case takes place, we say that the segment $[(q_1, M_1), (q_2, M_2)]$ is of the ith type. To a first-type segment, no information is assigned except for its type number. To a second-type segment, a word is assigned which is the output of the corresponding transition. For a third-type segment, let l' be a part of path l between the positions (q_1, M_1) and (q_2, M_2), $v = \mathrm{in}(l')$, and $w = \mathrm{out}(l')$.

Two cases are possible. If $|w| \leq n^4$, we say that the segment is of the first subtype and assign to it the word w itself. If $|w| > n^4$, we say that the segment is of the second subtype and assign to it the number k equal to the difference between $|w|$ and the minimum output length among all outputs of all paths from q_1 to q_2 with input v. By the corollary, $k \leq 2n^4$. Furthermore, among all segments we select those for which the pairs (q_1, M_1) and (q_2, M_2) correspond to two heads of l differing by one transition. Such segments will be referred to as *short*.

A sequence D with this information assigned to segments will be called the *diagram* of the path l and will be denoted by $D(l)$. The information includes also the type number of a segment and (for the third type) the subtype number, and also an indicator of whether the segment is short. Since the number of state transitions is at most n, diagrams of all accepting paths in $\mathfrak{A}$ are of length at most $4n$.

By a diagram (not related to any path previously specified) we call a sequence of pairs of the form (q, M) in which to each two neighboring pairs there is assigned one of the three segment types (for the third type, also a subtype) with the corresponding information and with an indicator of whether the segment is short. A diagram D is said to be *correct* if

1. Its first pair is $\langle q_0, \{q_0\}\rangle$, where q_0 is the initial state of $\mathfrak{A}$; its last pair is $\langle f, \{f\}\rangle$, where f is the final state of $\mathfrak{A}$. For each pair (q, M) in D we have $q \in M$;

2. The length of D is at most $4n$;

3. For each first-type segment $[(q_1, M_1), (q_2, M_2)]$, the states q_1 and q_2 are equivalent;

4. Each second-type segment $[(q_1, M_1), (q_2, M_2)]$ is marked as short, q_1 is not equivalent to q_2, M_1 is not equivalent to M_2, the word assigned to the segment is an output of some transition from q_1 to q_2;

5. For each third-type segment $[(q_1, M_1), (q_2, M_2)]$, the sets M_1 and M_2 are equivalent, and the states q_1 and q_2 are not equivalent. For the first subtype, the assigned word w is of length at most n^4; for the second subtype, the assigned number is not greater than $2n^4$;

6. Each pair (q, M) (except for, maybe, the first and the last) is an endpoint of a short segment $[(q_1, M_1), (q_2, M_2)]$ such that the sets M_1 and M_2 are not equivalent (this property guarantees, in particular, that segments of the first and third types are always separated by a second-type segment in D).

Lemma 5. *The diagram of any accepting path is correct. In total, there are no more than* $\exp(\mathrm{poly}(n))$ *correct diagrams. Given a diagram, its correctness can be checked using time* $\exp(\mathrm{poly}(n))$.

Proof. The first two claims of the lemma are obvious. In the third claim, only the algorithm for checking equivalence of two sets is nontrivial. It easily follows from Lemma 3. △

We say that an accepting path l satisfies a correct diagram $D = (q_1, M_1), (q_2, M_2), \ldots, (q_m, M_m)$ if an ascending sequence $l_1, l_2, \ldots, l_m$ of heads can be selected in it such that l_i ends in state q_i for each i, $M(l, l_i) = M_i$, all short segments correspond to the same transition, on each segment of the second type or of the first subtype of the third type the output coincides with that indicated in D, on each segment of the second subtype of the third type the output is of length greater than n^4 and is longer than the minimum output (among outputs of all paths from q_1 to q_2 with the same input) by the number indicated in D. Clearly, any accepting path l satisfies the diagram $D(l)$.

Lemma 6. *Let l_1 and l_2 be accepting paths in $\mathfrak{A}$ with the same input and satisfying the same correct diagram D. Then* $\mathrm{out}(l_1) = \mathrm{out}(l_2)$.

Proof. By property 6, any inner pair in D corresponds to a position in a path satisfying D directly before or after a set transition, and it is seen from D which of these two cases takes place. Any two-sided accessibility set M which is known to correspond to a position in the path directly to the left (or right) of a set transition uniquely determines a head u' of the word $u = \text{in}(l_1) = \text{in}(l_2)$ such that $M = M_u(u')$. Therefore, pairs in D uniquely divide the input u into corresponding parts, and it suffices to prove the equality of the outputs l_1 and l_2 on each part of the input. If the segment of the diagram is of the first type, coincidence of outputs on the corresponding part of the input follows from Lemma 2. For a segment of the second type or of the first subtype of the third type, the output is explicitly given in the diagram. On a segment of the second subtype of the third type, the outputs l_1 and l_2 are matching (see the corollary), and the same difference of the length with the same number ensures their coincidence. $\triangle$

4. DECOMPOSITION OF A FINITE-VALUED TRANSDUCER

We say that a finite-valued transducer $\mathfrak{A}$ is *decomposed* into single-valued transducers $\mathfrak{A}_1, \mathfrak{A}_2, \ldots, \mathfrak{A}_K$ if $\Gamma(\mathfrak{A}) = \bigcup\limits_{i=1}^{K} \Gamma(\mathfrak{A}_i)$. It is proved in [2] that any finite-valued transducer $\mathfrak{A}$ can be effectively decomposed into exactly k single-valued transducers in time $\exp(\exp(\text{poly}(n)))$, where k is the valuedness of $\mathfrak{A}$. Moreover, the size of each single-valued transducer is at most $\exp(\exp(\text{poly}(n)))$. In [5], the time and component size estimates are reduced to a single exponent, but the argument of the exponent contains k, which itself can be exponential in n (see Example 1 in Section 2). In the next theorem we propose a decomposition method with "purely" exponential estimates, which, however, does not guarantee that the number of single-valued transducers will be equal to k.

Theorem 3. *There exists an algorithm which, given any finite-valued transducer* $\mathfrak{A}$ *of size* n, *computes in time at most* $\exp(\text{poly}(n))$ *its decomposition into at most* $\exp(\text{poly}(n))$ *single-valued transducers of sizes at most* $\exp(\text{poly}(n))$.

Proof. As in Section 3, instead of finite-valuedness of $\mathfrak{A}$ we will assume that the finite-valuedness criterion (see Theorem 1) is satisfied. Since the possibility for decomposing a transducer implies its finite-valuedness, along with Theorem 3 we will prove the sufficiency of the criterion.

By Lemma 5, to prove Theorem 3 it suffices, for each correct diagram D, to construct in exponential time a transducer $\mathfrak{A}(D)$ whose graphic consists of only the elements of $\Gamma(\mathfrak{A})$ that are realized by paths satisfying the diagram D. By Lemma 6, $\mathfrak{A}(D)$ will be a finite-valued transducer.

By the left-sided accessibility set for a word u, we will call the set of states q for which there exists a path from the initial state to q with input u. If a diagram D is given, then by the local two-sided accessibility set on the a segment $[(q_1, M_1), (q_2, M_2)]$ of the diagram D for an input u and its head u' we call the set of states q for which there exists a path l from q_1 to q_2 such that $\text{in}(l) = u$ and the path $l(u')$ ends in q. By the local left-sided accessibility set on this segment for an input u, we call the set of states q for which there exists a path from q_1 to q with input u.

Let us describe $\mathfrak{A}(D)$. States of $\mathfrak{A}(D)$ are tuples of the form $\langle Q_1, Q_2, q, L\rangle$. In Q_1 there is computed the left-sided accessibility set for the head of the input already read, in $Q_2 \subseteq Q_1$ the two-sided accessibility set for all the input and the current head of the input, in $q \in Q_2$ the current state of the path in $\mathfrak{A}$ being guessed. Clearly, Q_2 uniquely determines the current first- or third-type segment $[(q_1, M_1), (q_2, M_2)]$ of D such that $M_1 \geq Q_2 \geq M_2$, if it exists. L is nonempty if and only if this segment exists and is of the third type. In the case of the first subtype, L is the number m, which can take all values from 0 to the length of the output indicated in D (in m, the length of the output on the segment is computed). In the case of the second subtype, L is a tuple $\langle Q_1', Q_2', P\rangle$. In Q_1' there is computed the local left-sided accessibility set on the current segment, and in Q_2' the local two-sided accessibility set, $q \in Q_2' \subseteq Q_1'$. P is a set of pairs $\langle q', m\rangle$, one pair for each state $q' \in Q_2'$, where $0 \leq m \leq 2n^4$, plus one current pair with q' equal to the current state q. Denote by u_t the part of the input that has been read by the current time in $\mathfrak{A}(D)$ on the considered segment. In the current pair $\langle q, m\rangle$, in m there is computed the difference between the length of the output of the guessed path on the input part u_t and the minimum output length among the outputs of all paths with input u_t from q_1 to the set Q'_2. To keep trace of this minimum length, in each noncurrent pair $\langle q', m\rangle$ in m there is computed the difference between the minimum output length among the outputs of all paths from q_1 to q' with input u_t and the minimum output length among the outputs of all paths with input u_t from q_1 to the set Q'_2. Initial states of $\mathfrak{A}(D)$ are those with $Q_1 = Q_2 = \{q_0\}$ and $q = q_0$; if the first segment of the diagram is of the third type, then L is nonempty: for the first subtype, $L = 0$, and for the second, $Q'_1 = Q'_2 = \{q_0\}$ and both pairs are $\langle q_0, 0\rangle$. The current pair is $\langle q, 0\rangle$. Final states of $\mathfrak{A}(D)$ are those with $Q_2 = \{f\}$ and $q = f$; if the last segment in D is of the third type, then L is nonempty: for the first subtype, m equals the length of the output indicated in D, for the second subtype, $Q'_2 = \{f\}$, and the current pair is $\langle f, m\rangle$, where m is the number indicated in D.

Now let us describe transitions of $\mathfrak{A}(D)$. We say that a state q'' from $\mathfrak{A}$ is a successor of q' by symbol a if there is a transition from q' to q'' with input a. A transition from state s_1 to state s_2 with input symbol a and output word v exists if and only if all the following conditions are satisfied:

1. $Q_1(s_2)$ (this is the notation for the component Q_1 in state s_2) is the set of all successors of states from $Q_1(s_1)$ by a. Thus, the component Q_1 is computed deterministically;

2. In $\mathfrak{A}$ there exists a transition from $q(s_1)$ to $q(s_2)$ with input a and output v;

3. For each state q' in $Q_2(s_1)$, at least one of its successors by a belongs to $Q_2(s_2)$. For each state q' in $Q_1(s_1) \setminus Q_2(s_1)$, all its successors by a do not belong to $Q_2(s_2)$;

4. In D there exists a segment R: $[(q_1, M_1), (q_2, M_2)]$ either of type 1 or 3 such that $M_1 \geq Q_2(s_1) \geq M_2$ and $M_1 \geq Q_2(s_2) \geq M_2$ (clearly, there can be only one such segment) or of type 2 such that $M_1 = Q_2(s_1)$, $M_2 = Q_2(s_2)$, $q_1 = q(s_1)$, $q_2 = q(s_2)$, and v equals the output indicated for R. In this latter case, if the preceding segment of D (before (q_1, M_1)) is of the second type, then we must have the following: $Q'_2(s_1) = \{q_1\}$, m in $L(s_1)$ equals the length of the output indicated in D (for the first subtype) or m in the current pair equals the number indicated in D (for the second subtype). Similarly, if the succeeding segment in D (from (q_2, M_2)) is of the third type, obvious initial conditions must hold. In the case where R is of the third type, the conditions given in the next paragraph must hold.

The set $Q'_1(s_2)$ consists of all successors of states from $Q'_1(s_1)$ by a. For each state q' in $Q'_2(s_1)$, at least one of its successors by a belongs to $Q'_2(s_2)$. For each state q' in $Q'_1(s_1) \setminus Q'_2(s_1)$, all its successors by a do not belong to $Q'_2(s_2)$. In the case of the first subtype, the output v extends the head of length $m(s_1)$ of the output indicated in D and $m(s_2) = m(s_1)+|v|$. In the case of the second subtype, $P(s_2)$ is obtained according to the following rules. First, to each $q' \in Q'_2(s_2)$ we assign the minimum number among all sums $m+|w|$ such that for some q'' we have $\langle q'', m\rangle \in P(s_1)$ and there exists a transition from q'' to q' with input a and output w. Among all thus assigned numbers, take the smallest number k. If $k \neq 0$, reduce all these numbers by k. All the obtained numbers do not exceed $2n^4$. These numbers in pairs with the states corresponding to them form $P(s_2)$. In the current pair we have $m(s_2) = m(s_1) + |v| - k \leq 2n^4$.

Finally, if the segment R is short, then $M_1 = Q_2(s_1)$, $M_2 = Q_2(s_2)$, $q_1 = q(s_1)$, and $q_2 = q(s_2)$.

It is clear that if in $\mathfrak{A}$ there is an accepting path l satisfying the diagram D, then in $\mathfrak{A}(D)$ there exists an accepting path l' such that $\mathrm{in}(l) = \mathrm{in}(l')$ and $\mathrm{out}(l) = \mathrm{out}(l')$. Conversely, let in $\mathfrak{A}(D)$ there be an accepting path l'. Let us show that the corresponding path l in $\mathfrak{A}$ satisfies D. The component Q_1 is computed correctly, since it is deterministic. Let us show that Q_2 is also computed correctly. Assume the contrary. Let Q_2 at some time has a superfluous state. Then, since elements of Q_2 have at least one successor in Q_2 at each step, at the end the component Q_2 cannot consists of the superfluous state only, which yields a contradiction. Now assume that Q_2 does not contain some state which actually belongs to the two-sided accessibility set. Since all successors of states from $Q_1 \setminus Q_2$ do not belong to Q_2, then at the end the final state f should not belong to Q_2, which is impossible. Correctness of computation of Q'_1 and Q'_2 on segments of the third type is proved

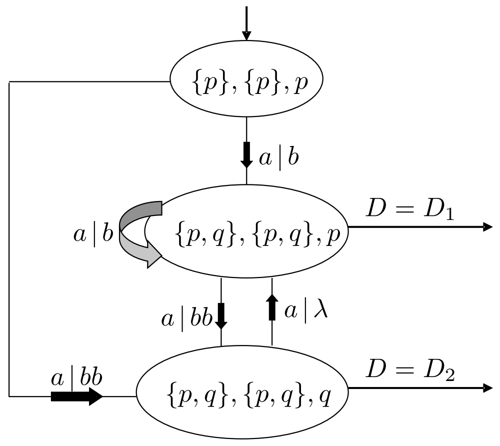

Fig. 7. Single-valued transducers obtained by decomposing the two-valued transducer shown in Fig. 1 differ in their final states only. Components of states are given in the order Q_1, Q_2, q. The middle state is final for $\mathfrak{A}(D_1)$, and the lower for $\mathfrak{A}(D_2)$, where $D_1 = \langle p,\{p\}\rangle, \langle p,\{p,q\}\rangle$ and $D_2 = \langle p,\{p\}\rangle, \langle q,\{p,q\}\rangle$.

similarly. The transitions of $\mathfrak{A}(D)$ are such that, being in D on some segment of the first or third type, it is possible to leave it only trough a transition corresponding to the next segment of the second type and only when the information on the first segment corresponds to D. Correctness of computing this information is obvious. Clearly, $\mathfrak{A}(D)$ is constructed in polynomial time. Thus, Theorem 3 and correctness of the finite-valuedness criterion are proved. $\triangle$

Remark 2. The presented construction can be applied to finite-valued transducers without empty inputs which need not necessarily have one initial and one final state (i.e., they need not be reduced). The only changes concern the form of an initial pair of a correct diagram (now the set M in it is a subset of the set Q_0 of initial states), the form of a final pair of a correct diagram (M in it is a subset of the set Q_f of final states), and the definition of initial and final states of the transducer $\mathfrak{A}(D)$. In its initial states we have $Q_1 = Q_0$, and in its final states the set $Q_1 \setminus Q_2$ does not contain states from Q_f.

Remark 3. If for some reasons it is known that on any segment of the third type all paths with the same input have matching outputs (for instance, outputs of all transitions in $\mathfrak{A}$ are words in an alphabet of one symbol), then, clearly, we can consider only diagrams where all third-type segments are of the second subtype.

Remark 4. After constructing all transducers $\mathfrak{A}(D)$ as above, one should delete all states in them through which an accepting path does not pass. Then both the sizes of these transducers and their number can often be reduced.

Example 3. In the transducer shown in Fig. 1, the states p and q are equivalent, and the possible two-sided accessibility sets $\{p\}$ and $\{p,q\}$ are also equivalent. One can easily see that only two diagrams of accepting paths are possible, $\langle p,\{p\}\rangle, \langle p,\{p,q\}\rangle$ and $\langle p,\{p\}\rangle, \langle q,\{p,q\}\rangle$, which consists of a single segment of the first type. Taking into account Remarks 2 and 4, we see that the finally obtained decomposition consists of two transducers differing in their final state only (see Fig. 7).

Example 4. In the transducer shown in Fig. 2, the states p and q are not equivalent, and the possible two-sided accessibility sets $\{p\}$ and $\{p,q\}$ are equivalent. Taking into account Remark 3, one can easily see that only two diagrams of accepting paths are possible, $\langle p,\{p\}\rangle, \langle p,\{p,q\}\rangle$ and $\langle p,\{p\}\rangle, \langle q,\{p,q\}\rangle$, which consists of a single third-type segment of the second subtype with number $k = 0$ assigned to it. Taking into account Remarks 2 and 4, we see that the finally obtained decomposition consists of two transducers shown in Figs. 8 and 9.

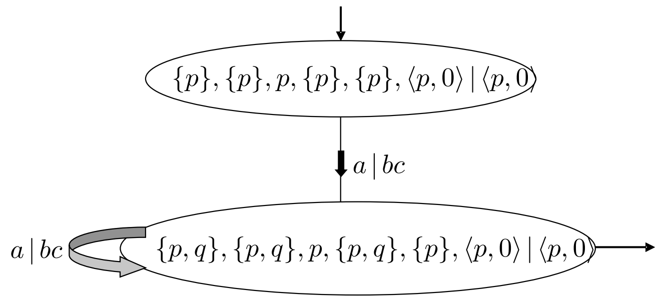
$\{p\},\{p\},p,\{p\},\{p\},\langle p,0\rangle \mid \langle p,0\rangle$
$a \mid bc$
$a \mid bc$
$\{p,q\},\{p,q\},p,\{p,q\},\{p\},\langle p,0\rangle \mid \langle p,0\rangle$

Fig. 8. Single-valued transducer corresponding to the diagram $\langle p,\{p\}\rangle,\langle p,\{p,q\}\rangle$ $(k=0)$ in the decomposition of the two-valued transducer shown in Fig. 2. Components of its states are given in the order Q_1,Q_2,q,Q_1',Q_2',P; the current pair is separated by a vertical line.

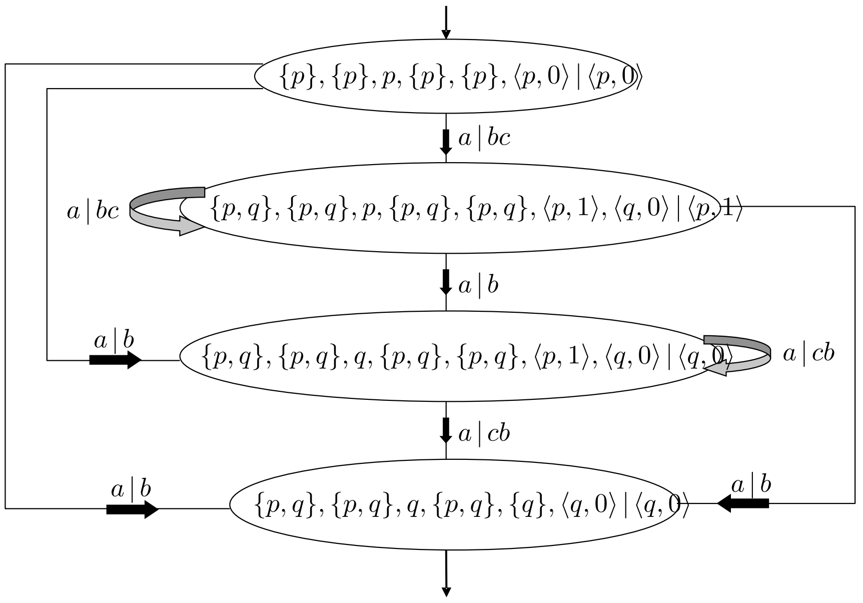
$\{p\},\{p\},p,\{p\},\{p\},\langle p,0\rangle \mid \langle p,0\rangle$
$a \mid bc$
$a \mid bc$
$\{p,q\},\{p,q\},p,\{p,q\},\{p,q\},\langle p,1\rangle,\langle q,0\rangle \mid \langle p,1\rangle$
$a \mid b$
$a \mid b$
$\{p,q\},\{p,q\},q,\{p,q\},\{p,q\},\langle p,1\rangle,\langle q,0\rangle \mid \langle q,0\rangle$
$a \mid cb$
$a \mid cb$
$a \mid b$
$\{p,q\},\{p,q\},q,\{p,q\},\{q\},\langle q,0\rangle \mid \langle q,0\rangle$
$a \mid b$

Fig. 9. Single-valued transducer corresponding to the diagram $\langle p,\{p\}\rangle,\langle q,\{p,q\}\rangle$ $(k=0)$ in the decomposition of the two-valued transducer shown in Fig. 2.

Example 5. In the transducer shown in Fig. 3, all states are not equivalent, and possible two-sided accessibility sets $\{q_1,q_1'\}$ and $\{q_2,q_2'\}$ are also not equivalent. One can easily see that only four diagrams of accepting paths are possible, $\langle q_1,\{q_1,q_1'\}\rangle,\langle q_2,\{q_2,q_2'\}\rangle$, $\langle q_1,\{q_1,q_1'\}\rangle,\langle q_2',\{q_2,q_2'\}\rangle$, $\langle q_1',\{q_1,q_1'\}\rangle,\langle q_2,\{q_2,q_2'\}\rangle$, and $\langle q_1',\{q_1,q_1'\}\rangle,\langle q_2',\{q_2,q_2'\}\rangle$, which consist of a single second-type segment. Taking into account Remarks 2 and 4, we see that the finally obtained decomposition consists of four transducers differing in a pair of current states only. The transducer for the first diagram is shown in Fig. 10.

Let us state one consequence of our constructions.

Theorem 4. *For any word u in $\mathfrak{A}$ there exists a set $M(u)$ consisting of* $\exp(\mathrm{poly}(n))$ *accepting paths with input u such that for any accepting path l with input u there exists a path $l' \in M(u)$ such that* $\mathrm{out}(l')=\mathrm{out}(l)$ *and for any $u' \subseteq u$ we have $|d(l,l',u')| \le 2n^4$.*

In particular, as is shown in [1], valuedness of a finite-valued transducer of size n is not greater than $\exp(\mathrm{poly}(n))$.

To prove Theorem 4, take for $M(u)$ one path from each possible diagram. The statement easily follows from Lemmas 2 and 6 and the corollary of Lemma 4.

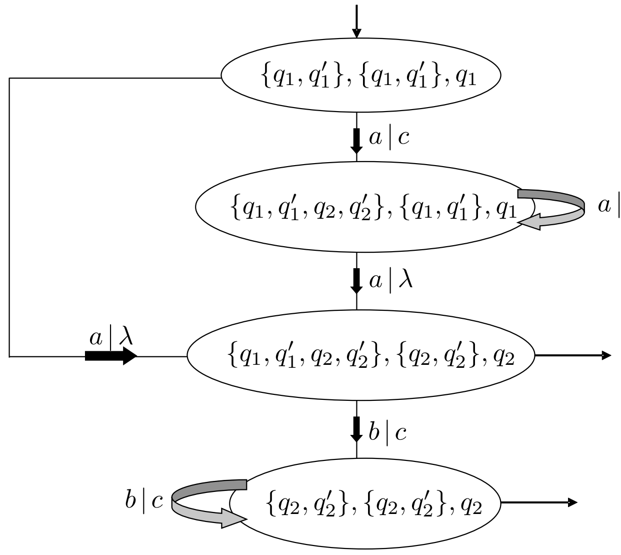
$\{q_1, q_1'\}, \{q_1, q_1'\}, q_1$
$a \mid c$
$\{q_1, q_1', q_2, q_2'\}, \{q_1, q_1'\}, q_1$
$a \mid c$
$a \mid \lambda$
$a \mid \lambda$
$\{q_1, q_1', q_2, q_2'\}, \{q_2, q_2'\}, q_2$
$b \mid c$
$b \mid c$
$\{q_2, q_2'\}, \{q_2, q_2'\}, q_2$

Fig. 10. Single-valued transducer obtained by decomposing the four-valued transducer shown in Fig. 3 for the diagram $\langle q_1, \{q_1, q_1'\}\rangle, \langle q_2, \{q_2, q_2'\}\rangle$ (components of states are given in the order Q_1, Q_2, q). The other three transducers are obtained from it by replacing the pair (q_1, q_2) of current states with one of the other three pairs.

5. TESTING THE INCLUSION

Now we consider the questions of testing the inclusion, i.e., of what input and output length it is sufficient to check to ascertain that an arbitrary transducer $\mathfrak{A}_1$ is included in a finite-valued transducer $\mathfrak{A}_2$. By Lemma 1, we may assume that $\mathfrak{A}_1$ and $\mathfrak{A}_2$ do not have empty inputs. The following lemma states that if $\mathfrak{A}_1$ is not included in $\mathfrak{A}_2$, this can be detected on inputs of exponential length.

Lemma 7. *If a transducer $\mathfrak{A}_1$ is not included in a finite-valued transducer $\mathfrak{A}_2$, then there exists a pair $\langle u, v\rangle$ such that $\langle u, v\rangle \in \Gamma(\mathfrak{A}_1)$, $\langle u, v\rangle \notin \Gamma(\mathfrak{A}_2)$, and $|v| \le \exp(p_1(n))$, where $p_1(n)$ is a polynomial.*

Proof. Consider an accepting path l_1 in $\mathfrak{A}_1$ with a minimum-length output such that $\langle u, v_1\rangle \in \Gamma(\mathfrak{A}_1)$ and $\langle u, v_1\rangle \notin \Gamma(\mathfrak{A}_2)$, where $u = \mathrm{in}(l_1)$ and $v_1 = \mathrm{out}(l_1)$. Assume that $|v_1| > \exp(p_1(n))$, where $p_1(n)$ is sufficiently large. Hereinafter, by expressions like "sufficiently large" we mean the magnitude of $\exp(\mathrm{poly}(n))$ where the degree of the polynomial is large enough to make all the described operations possible. By a "fixed exponent" we mean the magnitude of $\exp(p(n))$, where $p(n)$ is a polynomial whose existence is either obvious or has been proved before. If $\langle u, v_2\rangle \in \Gamma(\mathfrak{A}_2)$, $|v_1| = |v_2|$, but $v_1 \ne v_2$, then the first on the left (respectively, right) pair of distinct symbols of the words v_1 and v_2 equally distant from the beginning (respectively, end) will be called the left (respectively, right) *failure* between v_1 and v_2. By Theorem 4, the number of different v_2 such that $\langle u, v_2\rangle \in \Gamma(\mathfrak{A}_2)$ is a fixed exponent; therefore, the number of symbols in v_1 in which there occurs a failure between v_1 and some of such v_2 is also small. Take in v_1 a sufficiently large subword r in which no failure occurs and which is located at distance sufficiently many times greater than $|r|$ from the nearest failure.

Select sufficiently many heads u' of the input u (and thus also heads $l_1(u')$ of the path l_1) so that the following three conditions hold:

1. The words $\mathrm{out}(l_1(u'))$ for all the selected words u' end inside r and are pairwise distinct;

2. The state in which the path $l_1(u')$ ends is the same for all the selected u';

3. For all the selected words u', the left- and right-sided accessibility sets in $\mathfrak{A}_2$ are the same (the right-sided accessibility set for u' is the set of states from which there is a path to the final state with input u'', where $u'u'' = u$).

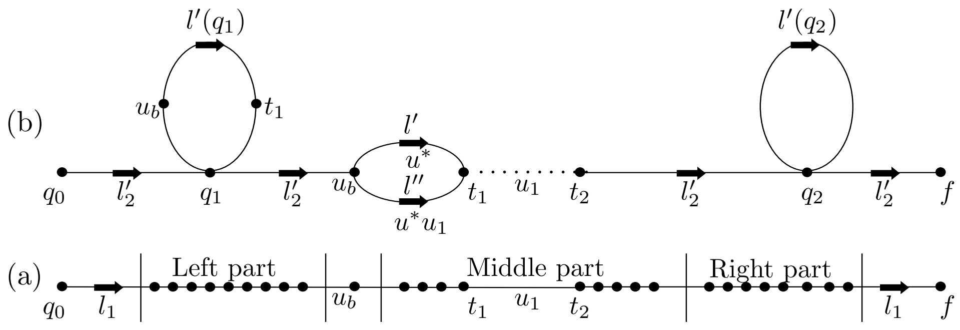

Fig. 11. Controlling the length of the output of a reconstructed path. (a) Structure of the path l_1 in transducer $\mathfrak{A}_1$. All states corresponding to the dots (bold circles) are the same. (b) Scheme of reconstructing the path l_2 from l'_2 in transducer $\mathfrak{A}_2$. A deleted segment u_1 is inserted so that the points t_1 and t_2 correspond to the same state.

The plan of the proof is the following. Since the path l_1 passes many times through one and the same state, one can delete from it (moreover, in many ways) some of its segments to obtain an accepting path with a shorter input and output. Then in $\mathfrak{A}_2$ there exists an accepting path with the same input and output. Property 3 makes it possible to "extend" this path to a path with input u (in what follows, the obtained path will be called *reconstructed*). The difficulty is to obtain also an output equal to v_1, thus arriving at an obvious contradiction. To this end, we first provide some "reserve" of accepting paths in $\mathfrak{A}_2$ with input u in order to find such paths among them to which (or to parts of which) it is convenient to apply the finite-valuedness criterion.

We will reduce step by step the set of selected words and at the same time construct in $\mathfrak{A}_2$ a set of marked states in the following way. At each step, for each unmarked state q in $\mathfrak{A}_2$ we check whether there exists an accepting path in $\mathfrak{A}_2$ with input u for which the heads $l(u')$ end in q for at least $m/\exp(n)$ selected words u', where m is the current number of selected words. If it exists, we mark q, assign to it one of the described paths, and make the set of selected words $1/\exp(n)$ as large, so that to make all the marked heads end in q. The path assigned to q will be denoted by $l(q)$ and will be called marked. When, at some step, no unmarked state can be marked, the process terminates. Since there are no more than n steps, the number of marked words after the process termination is sufficiently large. Each path $l(q)$ in all selected inputs is in state q.

Now we divide all the selected words into three approximately equally large parts consisting of words arranged in ascending order and fix one of the selected words u_b between the left and middle parts; this word will be referred to as boundary (see Fig. 11a). To each selected word u' in the middle part, we assign the set consisting of all tuples $\langle q, q_1, q_2, d\rangle$ such that in $\mathfrak{A}_2$ there exists a path l from q_1 to q_2 with input u'', where $u' = u_b u''$; q is a marked state; and the difference between $|\mathrm{out}(l)|$ and the length of the output of $l(q)$ on u'' is d, where $|d| \le 2n^4$. The number of the described sets is a fixed exponent; therefore, in the middle part there are many words to which one and the same set corresponds. In this part, we regard only these words as selected.

Since all selected heads of l_1 end in the same state, we can delete any set of segments between them to obtain a shortened accepting path l'_1 in $\mathfrak{A}_1$ with some input $\bar{u}$. By the choice of l_1, there exists an accepting path l'_2 in $\mathfrak{A}_2$ such that $\mathrm{in}(l'_2) = \mathrm{in}(l'_1) = \bar{u}$ and $\mathrm{out}(l'_2) = \mathrm{out}(l'_1)$. Let the segments be deleted in the middle part. By selected heads $\bar{u}'$ of the input $\bar{u}$ we will mean its heads obtained from the heads of the original input u by these deletions. Let us show that for at least one selected $\bar{u}'$ from the left and right parts the path $l'_2(\bar{u}')$ ends in a selected state. We denote by $q(t)$ the state in which the path $l'_2(t)$ ends. Consider the head t in $\mathrm{in}(l'_2)$ corresponding to the rightmost deleted segment. Since the right-sided accessibility sets for all the selected words coincide, there exists a path from $q(t)$ to the final state with input $u'u''$, where u' is the input of the rightmost deleted segment and u'' is the input of l'_2 from t to the end. By joining the old head with the new tail, we obtain a new accepting path in $\mathfrak{A}_2$ whose input is now smaller by one deleted segment. In the same way we insert the other deleted segments and obtain an accepting path with input u. In the left part of selected heads it passes through some state q_1 which occurs in approximately $1/(3n)$ of all selected heads. If q_1 were not marked, this would contradict the termination of the process of constructing the marked states. In a similar way (by inserting segments from left to right) one can show that in the right part there exists a selected head of the path l'_2 ending in a marked state q_2.

Thus, we have found in $\mathfrak{A}_2$ the paths $l(q_1)$ and $l(q_2)$ with input u which are in a convenient (for applying the finite-valuedness criterion) configuration with the path l'_2, which we can reconstruct to a path with input u but still cannot provide the output v_1. First let us make the length of the output of the reconstructed path to become definite in a sense.

Since the paths $l(q_1)$ and $l(q_2)$ in all selected heads pass through q_1 and q_2, in these paths we can make deletions on the same segments of the input as for l_1. Let us denote the paths with such deletions by $l'(q_1)$ and $l'(q_2)$. Condition (2) of Theorem 1 (with $s_1 = q_1$ and $s_2 = q_2$; p_1, p_2, and p_3 being parts of the paths $l'(q_1)$, l'_2, and $l'(q_2)$, respectively) implies that the difference of lengths of outputs of the path l'_2 on the segment from u_b to any selected middle head u' and of the path $l'(q_1)$ on the same part of the input is not greater in absolute value than $2n^4$. If there are no deletions in this part, then $l'(q_1)$ coincides with $l(q_1)$ on it. Taking this into account, consider the following process of reconstruction l'_2 to a path with input u (see Fig. 11b). Let the leftmost deleted segment be located between heads t_1 and t_2, and let its input be u_1. Denote by l' the part of the path l'_2 from u_b to t_1, and denote its input by u^*. By the construction (coincidence of the sets of tuples $\langle q, q_1, q_2, d\rangle$ for middle selected heads), there exists a path l'' from $q(u_b)$ to $q(t_1)$ such that $\mathrm{in}(l'') = u^*u_1$ and the difference $|\mathrm{out}(l'')| - |\mathrm{out}(l')|$ equals the length of the output of $l(q_1)$ on the input segment u_1. The input of the new accepting path obtained by replacing the part l' with l'' in l'_2 contains one less deleted segment. We may say that we have inserted a segment in the input, and the output became longer by the length of the output of $l(q_1)$ on the inserted input segment. After that, we in the same way insert the second to the leftmost segment, etc. Finally, we obtain a reconstructed path with input u.

Thus, we are controlling the length of the output of the reconstructed path. Now let us make it be equal to $|v_1|$.

By a *removal* we will call deleting a set of segments between selected heads from u together with deleting the corresponding segments of the output of l_1 from v_1. For a removal α we will denote by $l'_2(\alpha)$ some accepting path with removal α in $\mathfrak{A}_2$, by $l_2(\alpha)$ some reconstructed path with input u constructed by the above-described process, by $q_1(\alpha)$ some marked state through which the path $l'_2(\alpha)$ passes in the left part of selected heads, by $t_1(\alpha)$ some selected head in the left part in which $l_2(\alpha)$ is in $q_1(\alpha)$, by $q_2(\alpha)$ and $t_2(\alpha)$ the same in the right part, by $l'_\alpha(q_1)$ the path $l(q_1)$ with removal α, and by $|\alpha|$ the sum of lengths of outputs of the segments deleted from l_1.

Any segment between two neighboring selected heads in the middle part has one of the three types with respect to each marked state q: the length of the output of $l(q)$ on this segment can be greater than the length of the output of l_1 on it (positive type), less than this length (negative type), or equal to it (zero type). Thus, to each such segment there corresponds a collection of pairs $\langle q, \mathrm{type}\rangle$. The total number of such collections is a fixed exponent. Let us choose sufficiently many disjoint segments to which one and the same collection corresponds. Let as order them from left to right and consider a sequence S of removals in which the mth removal consists of the first m segments. To each removal α in S we assign a pair $\langle q, v\rangle$ with $q = q_1(\alpha)$ and $v = \mathrm{out}(l_2(\alpha))$. By Theorem 4, the number of such pairs is a fixed exponent. Let α_1 and α_2 be two distinct removals in S to which the same pair $\langle q, v\rangle$ is assigned. If the type of all segments with respect to q were nonzero, then, clearly, the differences between the length of outputs of the reconstructed paths

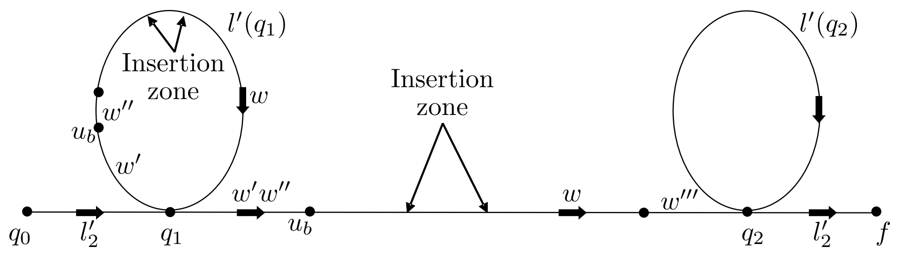

Fig. 12. Assuring the "insertion" character of input reconstruction and explicating the structure of insertions. A case is shown where outputs of the path $l'_2(\alpha)$ on inputs from $t_1(\alpha)$ to u_b and from $t_1(\alpha)$ to $t_2(\alpha)$ are longer than the corresponding outputs of $l'_2(\alpha)$. Here $w'w''$ and w''' are heads of the output w with $|w''| < n^4$ and $|w'''| < n^4$.

and $|v_1|$ would be different for α_1 and α_2, and hence these outputs would be different too. The obtained contradiction shows that the type of all segments with respect to q is zero, which, taking into account the insertion construction, yields for α_1 (and also for α_2) the equality $|v| = |v_1|$, where $v = \text{out}(l_2(\alpha_1))$.

Thus, we have assured the required length of the output v of the reconstructed path. It remains to make v coincide with v_1. If the input change in the reconstruction always occurred in the same segment r where by the construction the removals of the output of l_1 are located, this would be easy. Indeed, in this case (if the outputs do not coincide) we would consider the left failure between v and v_1. Then changes of the outputs in $\mathfrak{A}_1$ and $\mathfrak{A}_2$ under removals would occur on the same side from the failure and therefore the outputs could not become equal after removals. However, the correspondence between the input and output on the path l'_2 can be rather "twisted" as compared to this correspondence on the path l_1, and then the changes in the output can be located, for example, in the zone of possible failures. This problem can be resolved as follows. First we assure that changes of outputs in path reconstruction are precisely insertions, and then clarify the structure of these insertions.

We say that a removal α is of the zero type if the type of all segments of α with respect to $q_1(\alpha)$ is zero. Previously we have shown that in any piece of the middle part with sufficiently many selected heads there exist sufficiently many removals $\alpha_1, \alpha_2, \ldots, \alpha_k$ of the zero type, where all the $|\alpha_i|$ are distinct. Therefore, we can take sufficiently many removals on the middle part so that the following conditions are satisfied:

1. The removals are ordered, i.e., each segment of one of any two removals is located strictly to the left of each segment of the other and is disjoint with it;
2. All removals have the zero type;
3. For any two removals α_i and α_j we have $|\alpha_i| \neq |\alpha_j|$;
4. For all removals α the states $q_1(\alpha)$ and $q_2(\alpha)$ are the same (denote them by q_1 and q_2);
5. For all α, the differences between the length of the output of $l'_2(\alpha)$ on the segment from $t_1(\alpha)$ to u_b and the length of the output of $l'_\alpha(q_1)$ on the same segment are the same (it follows from condition (2) of Theorem 1 that this difference is not greater in absolute value than n^4);
6. All the $l_2(\alpha)$ (and therefore all the $l'_2(\alpha)$) have the same output from the beginning to u_b;
7. All the $\text{out}(l_2(\alpha))$ are the same (denote this output by v_2).

Condition 2 implies that the output of $l(q_1)$ is nonempty on any segment included in the removals. Then conditions (1) and (2) of Theorem 1 and the fact that the difference of outputs of the paths $l'_2(\alpha)$ and $l'_\alpha(q_1)$ on the segment from $t_1(\alpha)$ to $t_2(\alpha)$ remains unchanged after insertion imply that for all α (except for, maybe, the n^4 leftmost and rightmost ones) $\text{out}(l_2(\alpha))$ is obtained from $\text{out}(l'_2(\alpha))$ by inserting outputs of $l(q_1)$ on segments of α (see Fig. 12). Indeed, if w and w_1 are outputs of the paths $l'_\alpha(q_1)$ and $l(q_1)$ on the segment from $t_1(\alpha)$ to $t_2(\alpha)$, then the output of $l'_2(\alpha)$

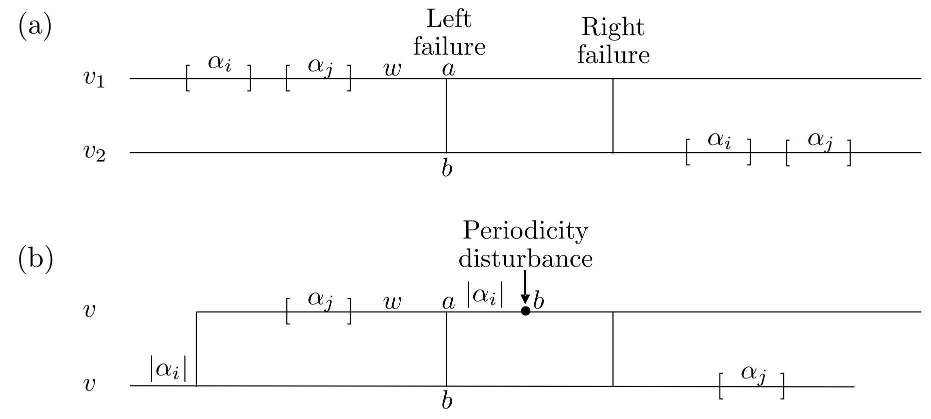

Fig. 13. Scheme of the proof of equality of outputs of the path l_1 and the reconstructed path. (a) Location of the removals α_i and α_j with respect to failures ($a \neq b$) in the outputs v_1 and v_2. (b) After performing the removal α_i, the words v_1 and v_2 turned into the same word v with period of length $|\alpha_i|$ from the beginning to $b \neq a$.

on this segment is a head of the word w^∞ ending in an n^4-vicinity of the end of the first w, and the output of $l_2(\alpha)$ on the same segment is a head of the word w_1^∞ ending in the similar vicinity, where w_1 is obtained from w by insertions made outside the n^4-vicinities of the beginning and end.

Thus, we have clarified the structure of insertions in the output. Now let us clarify the position of these insertions to guarantee their location outside the area of failures. After that we use condition 3 to obtain a contradiction to $v_1 \neq v_2$.

Condition 5 implies that for all α (except for, maybe, the n^4 leftmost ones) the outputs of all $l_2'(\alpha)$ after the head u_b repeat the outputs of the paths $l_\alpha'(q_1)$ starting from the same place on the common head of the outputs of $l_\alpha'(q_1)$. Taking into account condition 6, we conclude that if a removal α_i is located strictly to the left of α_j, then the insertion in the output for α_i is located strictly to the left (counting over the total length of the output) of the insertion for α_j. Let us call this the monotonicity property. Recall that $|v_1| = |v_2|$ (by condition 2), and for any removal, if the corresponding segments are deleted from v_1 and v_2, they become equal. By the monotonicity, there exist many removals for which the removal zone of v_2 contains neither left nor right failure between v_1 and v_2. Take two such removals, α_i and α_j, where α_i is to the left of α_j. Recall that the removal zone of v_1 does not contain failures by the construction of the segment r. It is easily seen that, to make it possible that the words v_1 and v_2 become equal after the removal α_i, it is necessary that removals from v_1 and v_2 lie on different sides from both failures. Let the removals from v_1 be located to the left of the left failure; then removals from v_2 lie to the right of the right failure (see Fig. 13a; the symmetric case is treated similarly). Make the removal α_i, shifting the beginning of v_1 by $|\alpha_i|$ to the right and the end of v_2 by $|\alpha_i|$ to the left (see Fig. 13b). Denote by a the symbol of the left failure in v_1 and by b the symbol of this failure in v_2.

It is clear that the head of the word v into which both words v_1 and v_2 have turned after performing the removal α_i is periodic with period $|\alpha_i|$. This periodicity is disturbed exactly at distance $|\alpha_i|$ to the right of a in a symbol a' (see Fig. 13b). Indeed, if it were not violated in this symbol, the failure would not disappear, and if it had been violated before, the failure would occur to the left of this position. In particular, the same about the period is valid for the suffix w of v beginning from the right endpoint of α_2. Indeed, by the construction of the subword r, the length of the head of w up to the failure a is much greater than $|\alpha_i|$, $|\alpha_j|$, and even their product. By the same arguments for the removal α_j, we obtain that the head of w is periodic with period $|\alpha_j|$, and this periodicity is disturbed precisely at distance $|\alpha_j|$ to the right of a. But then the head of w is periodic with period of length $|\alpha_i||\alpha_j|$. Since $|\alpha_i| \neq |\alpha_j|$ (condition 3), we obtain a contradiction to the fact that the disturbances of these two periods occurred in different positions. △

For completeness, let us also consider the estimate for $|u|$ under the conditions of the lemma. In [2] it is proved that if a transducer $\mathfrak{A}_1$ is not included in a finite-valued transducer $\mathfrak{A}_2$, then there exists a pair $\langle u, v\rangle \in \Gamma(\mathfrak{A}_1)$, $\langle u, v\rangle \notin \Gamma(\mathfrak{A}_2)$, with $|u| \le \exp(\exp(\operatorname{poly}(n)))$. Let us show how this fact can be deduced from Lemma 7. Consider the pair $\langle u, v\rangle$ whose existence is proved in Lemma 7 with the smallest $|u|$ for a given v. Assume that $|u| > \exp(\exp(\operatorname{poly}(n)))$, where the degree of the polynomial is sufficiently large. Since $|v| \le \exp(p_1(n))$, on an accepting path l_1 in $\mathfrak{A}_1$ with input u and output v there exists a sufficiently long (double exponential) segment with empty input. On this segment, let us select many heads of u in which l_1 is in the same state. To each selected head u' we assign a set of pairs of the form $\langle q, v'\rangle$, where $v' \subseteq v$ and q is a state in $\mathfrak{A}_2$ such that in $\mathfrak{A}_2$ there exists a path from the initial state q_0 to q with input u' and output v'. Theorem 4 implies that the number of such sets is a fixed double exponent; therefore, we can choose two heads u_1 and u_2 to which the same set is assigned. By the construction, in $\mathfrak{A}_2$ there is an accepting path l_2 with an input obtained from u by deleting the segment from u_1 to u_2 and with output v. Let q be the state in which $l_2(u_1)$ ends. Since the corresponding sets coincide, in $\mathfrak{A}_2$ there exists a path l_2' from q_0 to q such that $\operatorname{in}(l_2') = u_2$ and $\operatorname{out}(l_2') = \operatorname{out}(l_2(u_1))$. By joining the path l_2' with an extension of the path l_2, we obtain an accepting path in $\mathfrak{A}_2$ with input u and output v. This contradicts the condition that $\langle u, v\rangle \notin \Gamma(\mathfrak{A}_2)$. The estimate for $|u|$ is proved.

The lower estimate for the length of the output v in the statement of Lemma 7 is exponential (i.e., noninclusion can be detected on outputs of exponential length only). This estimate easily follows from the existence of automaton A described at the end of Section 2, which accepts not all words but all "exponentially short." It suffices to equip all transitions of A having nonempty inputs with a fixed single-symbol output, and the same for an automaton that accepts all words.

Remark 5. The authors are unaware of whether there exists a double exponential lower estimate for the length of the input u in Lemma 7.

6. DECIDABILITY OF INCLUSION OF TRANSDUCERS

Now we pass to question concerning decidability of inclusion of one transducer in another. In [10], decidability of inclusion of an arbitrary transducer $\mathfrak{A}_1$ in a finite-valued transducer $\mathfrak{A}_2$ was proved without estimating the running time of the algorithm. In [2], the decidability in time $\exp(\exp(\operatorname{poly}(n)))$ was proved, where n is the sum of sizes of $\mathfrak{A}_1$ and $\mathfrak{A}_2$. The space required for this algorithm also is of the order of a double exponent. In [8], these bounds were improved to a single exponent whose argument involves k, the valuedness of $\mathfrak{A}_2$. As is well known, k can itself be exponential in n. We prove a theorem which improves this result by constructing an algorithm with "purely" exponential space usage.

Theorem 5. *There exists a deterministic algorithm with space usage* $\exp(\operatorname{poly}(n))$ *which, given an arbitrary transducer* $\mathfrak{A}_1$ *and a finite-valued transducer* $\mathfrak{A}_2$, *decides the inclusion of* $\mathfrak{A}_1$ *in* $\mathfrak{A}_2$.

Proof. Let us describe a nondeterministic algorithm which ascertains noniclusion of $\mathfrak{A}_1$ in $\mathfrak{A}_2$ requiring exponential space. First the algorithm guesses an output v, $|v| \le \exp(p_1(n))$, and writes v on the tape. After that, it step by step guesses an input and a path in $\mathfrak{A}_1$. At each time moment the following information is written on the tape: the output $v_1 \subseteq v$ of the guessed head of the path in $\mathfrak{A}_1$ and the set of pairs $\langle v', q\rangle$, where $v' \subseteq v$ and q is a state in $\mathfrak{A}_2$ such that there exists a path from the initial state to q with the input guessed by this moment and with output v'. At each next step, the next symbol a of the input is guessed and the next transition in $\mathfrak{A}_1$ with input a. The algorithm checks whether the output of this transition extends v_1 along v and appends it to v_1. Then, for each pair $\langle v', q\rangle$ and each transition in $\mathfrak{A}_2$ from q with input a and an output which extends v' along v and does not go beyond it, a new pair is constructed in a natural way. Repeated pairs are deleted from the newly obtained set. The algorithm operates while v_1 is a head of v and the set of pairs is nonempty. If $v_1 = v$ and the set of pairs contains no pair $\langle v, f\rangle$, where f is the final state, the algorithm ascertains that $\mathfrak{A}_1$ is not included in $\mathfrak{A}_2$. Correctness of the algorithm and its exponential space usage are evident. By Savitch's theorem (see [11, pp. 477–478]), a nondeterministic algorithm using space S can be simulated efficiently by a deterministic algorithm recognizing the same language and and requiring space S^2. This implies the existence of the desired algorithm. $\triangle$

The removal construction used in the proof of Theorem 5 yields the following result.

Theorem 6. *Let $\mathfrak{A}_1$ and $\mathfrak{A}_2$ be finite-valued transducers without empty inputs, and let $\mathfrak{A}_1$ be included in $\mathfrak{A}_2$. Then for any accepting path l_1 in $\mathfrak{A}_1$ with input u and output v there exists an accepting path l_2 in $\mathfrak{A}_2$ with the same input and output and such that for any $u' \subseteq u$ we have $|d(l_1, l_2, u')| \le \exp(\mathrm{poly}(n))$.*

Proof. Denote by $M_1(u, v)$ and $M_2(u, v)$ the sets of accepting paths with input u and output v in $\mathfrak{A}_1$ and $\mathfrak{A}_2$, respectively. Contrary to the claim of the theorem, assume that there exists a path $l_1 \in M_1(u, v)$ such that for any path $l_2 \in M_2(u, v)$ there exists $u' \subseteq u$ with $|d(l_1, l_2, u')| > k \ge \exp(\mathrm{poly}(n))$, where the degree of the polynomial is sufficiently large. By Theorem 4, in $\mathfrak{A}_2$ there is a no more than exponential set M of paths from $M_2(u, v)$ such that for any path $l \in M_2(u, v)$ there exists $l' \in M$ such that for any $u' \subseteq u$ we have $|d(l, l', u')| \le 2n^4$. Consider the set P of accepting paths l in $\mathfrak{A}_1$ possessing the following property: on the input l there exists a set T of no more than $|M|$ heads such that for any path $l' \in M_2(\mathrm{in}(l), \mathrm{out}(l))$ there exists a head $u' \in T$ with $|d(l, l', u')| > k_1 = k - 2n^4$. One can easily see that $l_1 \in P$, and therefore P is nonempty. Let l_0 be a path in P with the smallest output length; denote $u_0 = \mathrm{in}(l_0)$ and $v_0 = \mathrm{out}(l_0)$. By T_0, denote the set of heads corresponding to it. Our assumptions imply that $|v_0| > k_1$. Therefore, there exists a segment r of l_0 with sufficiently large output that does not contain heads from T_0. For r, we repeat all the removal construction described in the proof of Lemma 7 (for l_1, u, and v_1 in Lemma 7 we now take l_0, u_0, and v_0). The only difference is that now as a path l_2' in $\mathfrak{A}_2$ for the path l_0' in $\mathfrak{A}_1$ with removals we now take not an arbitrary path but such path that for any head $u' \in T_0$ we have $|d(l_0', l_2', u')| \le k_1$. Such a path exists by the condition of the choice of l_0 and the fact that $|\mathrm{out}(l_0')| < |\mathrm{out}(l_0)|$. Repeating the corresponding arguments, one can easily prove that the exists a zero-type removal from l_0 such that for the reconstructed path l_2 in $\mathfrak{A}_2$ we have $\mathrm{in}(l_2) = u_0$ and $\mathrm{out}(l_2) = v_0$. The latter equality follows from the fact that in the proof of Lemma 7 we have obtained a contradiction when assuming that for all removals α we have $\mathrm{out}(l_2(\alpha)) \ne v_0$. It is easily seen that, because of the zero type and by the insertion construction, when a segment is inserted, the deviation $d(l_0', l_2', u')$ of the path in $\mathfrak{A}_1$ from the path in $\mathfrak{A}_2$ can be changed only for u' lying between the boundary head u_b and the segment inserted in the input (more precisely, $d(l_0'', l_2'', u'') = d(l_0', l_2', u')$, where l_0'' and l_2'' are paths before the insertion, l_0' and l_2' are paths after the insertion, $u' = u''$ if u' ends to the left of the insertion, and $u' = (u''$ with the insertion) if u' ends to the right of it). Hence, outside the segment r, and in particular on all heads $u' \in T_0$, we have $|d(l_0, l_2, u')| \le k_1$. This contradicts the fact that $l_0 \in P$. $\triangle$

In one particular case, the result of Theorem 5 can be improved. We say that a transducer $\mathfrak{A}$ *has finite delay* if there exists a natural number c such that for any path l the condition $|\mathrm{in}(l)| \ge c$ implies $|\mathrm{out}(l)| > 0$. Clearly, $c \le \mathrm{poly}(n)$, where $n = |\mathfrak{A}|$ (if the transducer is finite-valued and reduced, then its finite delay is equivalent to the nonexistence of cycles with empty input).

Theorem 7. *There exists a nondeterministic algorithm which ascertains noniclusion of an arbitrary transducer $\mathfrak{A}_1$ in a finite-valued transducer $\mathfrak{A}_2$ in nondeterministic time* $\exp(\mathrm{poly}(n))$, *where $\mathfrak{A}_2$ has finite delay.*

For the proof of the theorem, we need the following lemma, which improves Lemma 7 in this particular case.

Lemma 8. *If a transducer $\mathfrak{A}_1$ is not included in a finite-valued transducer $\mathfrak{A}_2$ with finite delay, then there exists a pair $\langle u, v\rangle \in \Gamma(\mathfrak{A}_1)$, $\langle u, v\rangle \notin \Gamma(\mathfrak{A}_2)$, with $|u| \le \exp(p_2(n))$, where $p_2(n)$ is a polynomial.*

Proof. Let l_1 be an accepting path in $\mathfrak{A}_1$ of the minimum length such that $\langle \mathrm{in}(l_1), \mathrm{out}(l_1)\rangle \notin \Gamma(\mathfrak{A}_2)$. Denote $u = \mathrm{in}(l_1)$ and $v_1 = \mathrm{out}(l_1)$. Assume that $|u|$ is sufficiently large. Two cases are possible.

Case 1. $|v_1| > \exp(p_1(n))$, where $p_1(n)$ is the polynomial from Lemma 7. In this case this assumption leads to a contradiction in the same way as in Lemma 7.

Case 2. $|v_1| \le \exp(p_1(n))$. In this case there exists a sufficiently large part r of the path l_1 with empty output. We will make removals on it in the same way as in Lemma 7, but instead of the requirement that the output of the deleted segments is nonempty we require that the length of the input of each deleted segment is greater than c. Repeating the corresponding arguments, we prove the existence of a zero-type removal α. However, due to the finite delay property, the output of any marked path in $\mathfrak{A}_2$ is nonempty on the deleted segments of the input. This contradicts the fact that the output of r is empty. △

Proof of Theorem 7. Let us describe the desired algorithm. It guesses an input u and an output v with $|u| \le \exp(p_2(n))$ and $|v| \le n|u|$. After that, it deterministically ascertains whether in $\mathfrak{A}_1$ and in $\mathfrak{A}_2$ there is a path with input u and output v. To this end, for each $u' \subseteq u$, where u' is extended symbol by symbol, it finds the set of pairs $\langle v', q\rangle$ where $v' \subseteq v$ and q is a state such that there exists a path from the initial state to q with input u' and output v'. Details are obvious. The algorithm detects noniclusion of $\mathfrak{A}_1$ in $\mathfrak{A}_2$ if $\langle u, v\rangle \in \Gamma(\mathfrak{A}_1)$, $\langle u, v\rangle \notin \Gamma(\mathfrak{A}_2)$. Clearly, the running time of the algorithm is approximately $n|u||v|$. △

The second author is grateful to Yu.L. Pritykin and an anonymous reviewer for careful reading of the text and numerous valuable remarks leading to a considerable improvement of the paper.

REFERENCES

1. Weber, A., Über die Mehrdeutigkeit und Wertigkeit von endlichen Automaten und Transducern, *Dissertation*, Goethe-Universität Frankfurt am Main, Germany, 1987.
2. Weber, A., A Decomposition Theorem for Finite Valued Transducers and an Application to the Equivalence Problem, *Proc. 13th Int. Sympos. on Mathematical Foundations of Computer Science (MFCS'88), Carlsbad, Czechoslovakia, Aug. 29 – Sept. 2, 1988*, Chytil, M., Janiga, L., and Koubek, V., Eds., Lect. Notes Comput. Sci., vol. 324, Berlin: Springer, 1988, pp. 552–562.
3. Weber, A., On the Valuedness of Finite Transducers, *Acta Inform.*, 1990, vol. 27, no. 8, pp. 749–780.
4. Weber, A., Decomposing a k-Valued Transducer into k Unambiguous Ones, *RAIRO Inform. Théor. Appl.*, 1996, vol. 30, no. 5, pp. 379–413.
5. Sakarovitch, J. and de Souza, R., On the Decomposition of k-Valued Rational Relations, *Proc. 25th Int. Sympos. on Theoretical Aspects of Computer Science (STACS'2008), Bordeaux, France, Feb. 21–23, 2008*, Albers, S. and Weil, P., Eds., Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2008, pp. 621–632.
6. Sakarovitch, J. and de Souza, R., Lexicographic Decomposition of k-Valued Transducers, *Theory Comput. Syst.*, 2010, vol. 47, no. 3, pp. 758–785.
7. Sakarovitch, J. and de Souza, R., On the Decidability of Bounded Valuedness for Transducers, *Proc. 33rd Int. Sympos. on Mathematical Foundations of Computer Science (MFCS'2008), Toruń, Poland, Aug. 25–29, 2008*, Ochmanski, E. and Tyszkiewicz, J., Eds., Lect. Notes Comput. Sci., vol. 5162, Berlin: Springer, 2008, pp. 588–600.

8. de Souza, R., On the Decidability of the Equivalence for k-Valued Transducers, *Proc. 12th Int. Conf. on Developments in Language Theory (DLT'2008), Kyoto, Japan, Sept. 16–19, 2008*, Ito, M. and Toyama, M., Eds., Lect. Notes Comput. Sci., vol. 5257, Berlin: Springer, 2008, pp. 252–263.

9. Sakarovitch, J., *Elements of Automata Theory*, Cambridge: Cambridge Univ. Press, 2009.

10. Culik, K., II and Karhumäki, J., The Equivalence of Finite Valued Transducers (on HDT0L Languages) is Decidable, *Theoret. Comput. Sci.*, 1986, vol. 47, no. 1, pp. 71–84.

11. Hopcroft, J.E., Motwani, R., and Ullman, J.D., *Introduction to Automata Theory, Languages, and Computation*, Boston: Addison-Wesley, 2001, 2nd ed. Translated under the title *Vvedenie v teoriyu avtomatov, yazykov i vychislenii*, Moscow: Williams, 2002.