Transforming a Single-Valued Transducer Into a Mealy Machine*

Andreas Weber†

Fachbereich Informatik, Johann Wolfgang Goethe-Universität, D-60054 Frankfurt am Main, Germany

Received October 5, 1995; revised April 14, 1997

This article deals with the transformation of a single-valued finite transducer into a Mealy machine. The following results are obtained: (1) Let $M$ be a single-valued real-time (or "letter-to-word") transducer with $n$ states, input alphabet $\Sigma$, and output alphabet $\Delta$ which is equivalent to some Mealy machine $M'$. Then, $M$ can be effectively transformed into such an $M'$ having at most $2^n + 1$ states. A similar result holds if $M$ is not real time. As an important side effect three "Mealy" properties are obtained which characterize the fact that the given transducer $M$ is equivalent to some Mealy machine. (2) The upper bound in result (1) improves to $2^n - 1$ if $M$ is known to be a letter-to-letter transducer. (3) For each integer $t \geq 2$ and each odd integer $n \geq 3$ there is a single-valued real-time transducer $M$ with $n$ states and input and output alphabets of cardinality $t$ such that $M$ is equivalent to some Mealy machine $M'$ and every such $M'$ has at least $t^{n-1}/2$ states. (4) If $t = 3$, then result (3) holds true with letter-to-letter transducers rather than real-time transducers and with a lower bound of $2^{n-1}/2$. (5) It is a PSPACE-complete problem to decide whether or not a given single-valued transducer $M$ is equivalent to some Mealy machine. The problem remains PSPACE-complete if $M$ is known to be a letter-to-letter transducer. © 1998 Academic Press

1. INTRODUCTION

A transducer is a finite-state machine with an output device. An atomic step (or transition) of a real-time (or "letter-to-word") transducer $M$ causes a change of the state of the machine while an input letter is consumed from the one-way input tape and an output word is produced on the one-way output tape. The machine $M$ is a letter-to-letter transducer if these output words are always letters. A computation of $M$ is a sequence of transitions with matching states. The word consumed (produced) by the computation is the concatenation of the words consumed (produced, respectively) by its transitions. The transduction realized by $M$ is the set of pairs of input/output words being consumed/produced by any computation initiating and terminating at designated initial and final states, respectively. For each such pair $(x, y)$, $y$ is called a value for $x$ in $M$. A transducer is single valued if the transduction realized by it is a partial function; i.e., each input word has at most one value. Two transducers are equivalent if the transductions realized by them coincide; i.e., every input word has the same set of values in both machines.

A Mealy machine is a letter-to-letter transducer where all states are final and the transitions behave in a deterministic way on the input tape. Every Mealy machine is a single-valued real-time transducer. On the other hand, not every single-valued real-time transducer is equivalent to a Mealy machine. The detailed definitions for transducers and some examples can be found in Section 2.

In this article we investigate the transformation of a single-valued transducer into an equivalent Mealy machine if such a machine exists. Our main results are as follows: (1) Let $M$ be a single-valued real-time transducer with $n$ states, input alphabet $\Sigma$, and output alphabet $\Delta$ which is equivalent to some Mealy machine $M'$. Then, $M$ can be effectively transformed into such an $M'$ having at most $\min\{9/4, 2 \cdot \# \Sigma^{n-1}, 2 \cdot (1 + \# \Delta)^{n-1}\}$ states. A similar result holds if $M$ is not real time. As an important side effect three "Mealy" properties are obtained which characterize the fact that the given transducer $M$ is equivalent to some Mealy machine (see Section 3).

(2) Let $M$ be a single-valued letter-to-letter transducer with $n$ states which is equivalent to some Mealy machine $M'$. Then, $M$ can be effectively transformed into such an $M'$ having at most $2^n - 1$ states (see Section 3).

(3) For every integer $t \geq 2$ and every odd integer $n \geq 3$ there is a single-valued real-time transducer $M$ with $n$ states and input and output alphabets of cardinality $t$ such that $M$ is equivalent to some Mealy machine $M'$ and every such $M'$ has at least $t^{n-1}/2$ states (see Section 4).

(4) For every odd integer $n \geq 3$ there is a single-valued letter-to-letter transducer $M$ with $n$ states and ternary input and output alphabets such that $M$ is equivalent to some Mealy machine $M'$ and every such $M'$ has at least $2^{(n-1)/2}$ states (see Section 4).

(5) It is a PSPACE-complete problem to decide whether or not a given single-valued transducer $M$ is equivalent to some Mealy machine. The problem remains...
In result (1) we have \((9/4) \cdot (2 \cdot \# \Sigma)^{n-1} \leq 2^{n+1} \cdot \# \Sigma^{n-1} \) and \(2 \cdot (1 + \# A)^{n-1} \leq 2^n \cdot \# A^{n-1} \). Therefore, we could state that result with a unified upper bound \(2^{n+1} \cdot \min\{ \# \Sigma, \# A\}^{n-1} \). From previous work on the transformation of a single-valued transducer into an equivalent deterministic transducer \([24]\) it can be derived that result (1) holds with the upper bound \(1 + 2^n \cdot \max\{2, \# A\}^{\cdot n-1} \). The two Mealy machine constructions employed in the proof of (1) and (2) are entirely different from the one in \([24]\). The basic idea for the design of a first candidate for the Mealy machine \(M'\) in (1) is to carry out a subset construction on the states of \(M\) and to memorize for certain states \(q\) of \(M\) the last letter of the uniquely determined output word produced so far at \(q\). If \(M\) was a letter-to-letter transducer, then \(M'\) does not memorize any output letters which makes it suitable for proving (2); a direct construction of this machine is also given. The basic idea for the design of a second candidate for the Mealy machine \(M''\) in (1) is to carry out a subset construction on the states of \(M\) with a certain delay on the input. Result (1) is proved by means of that candidate machine having fewer states than the other one. The correctness of the Mealy properties mentioned in result (1) can be also derived from \([24]\) (see Section 5).

The computations of our transducer \(M\) in result (3) temporally produce a certain surplus of output letters. Our transducer \(M\) in result (4) nondeterministically decides whether or not certain factors of its input word belong to a certain regular language. Both phenomena require exponentially many states to be handled by a Mealy machine. Filling the gap between the upper and lower bounds in (1) and (3) (and in (2) and (4), respectively) is an open problem (see Section 4). Our procedures used in result (5) employ, among other things, elementary graph algorithms (see, e.g., \([7, \text{Section 23}]\)) and a procedure presented in \([24]\). Note that it is decidable in polynomial time whether or not a given transducer is single valued \([11]\), (see Section 5).

The Moore machine is a model very similar to the Mealy machine. Every Moore machine can be transformed into an equivalent Mealy machine having the same number of states, and every Mealy machine with \(n\) states and output alphabet \(\Sigma\) can be transformed into an equivalent Moore machine having \(n \cdot \# A\) states \([3, \text{Section 1.1.5}]\). Therefore, results (1)–(5) also hold for Moore machines rather than for Mealy machines, provided that the upper bounds in (1) and (2) are replaced by \(\min\{9/4, 2 \cdot \# \Sigma^{n-1}, 2 \cdot (1 + \# A)^{n-1}\} \cdot \# A\) and by \((2^n - 1) \cdot \# A\), respectively.

Mealy and Moore machines are important abstract tools for the computer-aided design of microelectronic circuits. Specifically, these machines are used at the register-transfer level of the circuit design as a model for the control units of the circuit. A control unit has finitely many states and communicates regularly with other control units and with operational units (or datapaths) of the circuit by means of a number of one-bit signals. The transformation of a Mealy or Moore machine from the register-transfer level to the gate level of the circuit design is known as sequential logic synthesis \([2, 3, 19, \text{Chap. 4}]\). Interestingly, the Mealy and Moore machines considered in \([3]\) stem from graph schemes of algorithms which are at the system level of the circuit design. A general background on high-level synthesis can be found in \([8]\).

Results (1), (2), and (5) suggest modelling a control unit of a circuit by a single-valued transducer rather than a Mealy or Moore machine, to test whether or not the transducer is equivalent to some Mealy (or Moore) machine and, if so, to transform it automatically into an equivalent Mealy or Moore machine. Results (3) and (4) say that in some cases this method leads to an exponential saving in the descriptional complexity of the circuit design. Intuitively, these savings may be achieved in certain cases where a Mealy machine has to delay considerably the output of a single-valued transducer or has to simulate a nondeterministic decision procedure carried out by a single-valued transducer.

Finally, we want to point out that results (1)–(4) may be seen as part of a rich literature on economy of description for transducers \([24]\) and for similar models of computation \([13–15, 17, 18, 20–22]\). A further background on transducers and Mealy and Moore machines may be obtained from sources as \([1, 4–6, 10, 12]\). For a general framework on descriptional (or Kolmogorov) complexity the reader is referred to \([16, 23]\).

2. PRELIMINARIES

We use the following notations. Let \(\Sigma\) and \(\Delta\) be nonempty finite sets. For every \(y \in \Delta^*\) and \(j \in [1, ..., |y|]\) the \(j\)th letter of the word \(y\) is denoted by \(y(j)\). For any rational number \(l\) the set \([y \in \Delta^* \colon |y| \leq l]\) is denoted by \(\Delta^{|l|}\). Let \(\Delta\) and \(\Sigma\) be sets. A relation \(\rho \subseteq \Delta \times \Sigma\) is identified with a function \(\rho : \Delta \to \Sigma^*\), where \(y \in \rho(x)\) whenever \((x, y) \in \rho\). If \# \(\rho(x)\) \(\leq 1\) for every \(x \in \Delta\), then \(\rho\) is a partial function \(\rho : \Delta \rightharpoonup \Sigma^*\). If \# \(\rho(x)\) \(= 1\) for every \(x \in \Delta\), then \(\rho\) is a function \(\rho : \Delta \to \Sigma^*\). A relation \(\rho : \Sigma^* \times \Delta^*\) is said to preserve prefixes if, for every \(x, x' \in \Sigma^*\) and \(y, y' \in \Delta^*\) such that \((x, y), (x', y') \in \rho\), \(x'\) is a prefix of \(x\), we have that \(y\) is a prefix of \(y'\).

A finite transducer is a 6-tuple \(M = (Q, \Sigma, \Delta, \theta, Q_0, Q_f)\), where \(Q\), \(\Sigma\), and \(\Delta\) denote nonempty finite sets of states, input symbols, and output symbols, respectively;
We say that $p$ transducer is, of course, isomorphic to the usual one of a $Qm$ such that $(q_2, x, y) \in \delta$ if there is a word $x \in \Sigma^*$ such that $y \in y(x)$ is represented as an edge leading from $x$ to the output. Intuitively, an output word of the Mealy machine is obtained from the input word by doubling the letters in the odd positions and deleting the letters in the even positions. If the input word has odd length, its last letter is just copied to the output.

By definition, every Mealy machine is a deterministic real-time transducer and every deterministic transducer is single valued. In order to avoid trivial cases we assume from now on that in a single-valued transducer the empty word $\varepsilon$ has no value other than $\varepsilon$. The following proposition shows that the real-time transducer is a normal form for a single-valued or deterministic transducer.

**Proposition 2.1** [24, Proposition 1.1]. Let $M$ be a single-valued transducer. There is an equivalent real-time transducer $M'\varepsilon$ having at most $|M|$ states and size polynomial in $|M|$. The transducer $M'$ inherits from $M$ the property of being deterministic. It can be computed in time polynomial in $|M|$.

![FIG. 1. A Mealy machine.](image-url)
Having in mind Proposition 2.1, a relation \( \rho \subseteq \Sigma^* \times A^* \) is called a Mealy relation (a deterministic rational relation, a rational function, a real-time function, a rational relation, a rational relation) if it is realized by some Mealy machine (deterministic real-time transducer, single-valued real-time transducer, real-time transducer, transducer, respectively). The following examples of relations may illustrate the above-mentioned types of transducers. For all these relations it is straightforward to construct a transducer of the respective type. For each relation we briefly explain why it cannot be realized by any transducer of the more special type mentioned before.

The set \( \rho_1 = \{ (x,y) : x \in \{0,1\}^*, |x| \text{ even} \} \) is a deterministic rational relation, but not a Mealy relation, since its domain is a proper subset of \( \{0,1\}^* \). The sets \( \rho_2 = \{ (x,0) : x \in \{0,1\}^+, |x| \text{ even} \} \cup \{ (x,1) : x \in \{0,1\}^+, |x| \text{ odd} \} \) and \( \rho_3 = \{ (x,0^{(i)} : x \in \{0,1\}^*, |x| \text{ even} \} \cup \{ (x,1^{(i)} : x \in \{0,1\}^*, |x| \text{ odd} \} \) are rational functions but not deterministic rational relations. If there was a deterministic real-time transducer realizing \( \rho_2 \), then this transducer must contain accepting paths of the form \((q_1,0,y_1),(q_2,0,y_2),(q_3)\) such that \( y_1,y_2=0 \) and \( y_1=1 \), a contradiction. In the same fashion it can be seen that there is no deterministic real-time transducer realizing \( \rho_3 \). The set \( \rho_4 = \{ (x,e) : x \in \{0,1\}^* \} \cup \{ (x,x) : x \in \{0,1\}^* \} \) is a real-time rational relation, but not a rational function, since every word \( x \in \{0,1\}^+ \) has two distinct values. Finally, the set \( \rho_5 = \{ (a,a^i) : a \in \{0,1\}, j \geq 1 \} \) is a rational relation, but not a real-time rational relation, since each of the words 0 and 1 has infinitely many values.

3. UPPER BOUNDS

In this section we prove upper bounds for the transformation of a single-valued transducer into an equivalent Mealy machine if such a machine exists. The outcome is stated in Theorems 3.1 and 3.2. The first theorem also gives a testable necessary and sufficient condition for a rational function to be a Mealy relation. In order to formulate this condition we define the “Mealy” properties (M1), (M2), and (M3) of a relation \( \rho \subseteq \Sigma^* \times A^* \), where \( \Sigma \) and \( A \) are nonempty finite sets:

\( \text{(M1)} \) For every \( (x,y) \in \rho \), \( |x| = |y| \).

\( \text{(M2)} \) For every \( x \in \Sigma^* \) there is a \( y \in A^* \) such that \( |x| = |y| \) and \( \rho(\{x\} \times \Sigma^*) \subseteq \{y\} \times A^* \).

\( \text{(M3)} \) For every \( x \in \Sigma^* \), \( \rho(x) \) is nonempty.

Let us remark that properties (M1) and (M3) simply mean that the relation \( \rho \) preserves lengths and that its domain is the full set \( \Sigma^* \), respectively.

Theorem 3.1. Let \( M \) be a single-valued real-time transducer with \( n \) states, input alphabet \( \Sigma \), and output alphabet \( A \).

The transduction realized by \( M \) is a Mealy relation if and only if it has properties (M1), (M2), and (M3). If \( T(M) \) is a Mealy relation, then there are Mealy machines \( M' \) and \( M'' \) realizing \( T(M) \) and having at most \( 2 \cdot (1 + \# A)^{n-1} \) states and at most \( (9/4) \cdot (2 \cdot \# \Sigma)^{n-1} \) states, respectively.

Theorem 3.2. Let \( M \) be a single-valued letter-to-letter transducer with \( n \) states. If \( T(M) \) is a Mealy relation, then there is a Mealy machine \( M' \) realizing \( T(M) \) and having at most \( 2^n - 1 \) states. The machine \( M' \) can be determined in time polynomial in \( 2^n \cdot |M| \).

It is easy to see that every relation \( \rho \subseteq \Sigma^* \times A^* \) with properties (M1) and (M2) is a partial function \( \rho : \Sigma^* \rightarrow A^*. \) Therefore, the condition that \( M \) is single valued can be omitted in Theorems 3.1 and 3.2. Note that if in Theorem 3.1 the alphabet \( \Sigma \) is a singleton set then \( M' \) (\( M'' \)) has at most \( 2^n \) states (at most \( (9/4) \cdot 2^n \) states, respectively). Proposition 2.1 implies that Theorem 3.1 also holds if \( M \) is not real time provided that in the upper bounds \( n \) is replaced by \( |M| \).

The remainder of this section is devoted to the proof of Theorems 3.1 and 3.2. Let us first discuss in an informal way what can be derived from previous work on the same topic.

Let \( M = (Q, \Sigma, \Lambda, \delta, Q_0, Q_F) \) be a single-valued real-time transducer. Assume for a moment that \( M \) is a Mealy machine. Since \( M \) is a letter-to-letter transducer, \( T(M) \) has property (M1). Since furthermore \( M \) is deterministic, \( T(M) \) has property (M2). Finally, since \( M \) is completely specified and real time and all its states are final, \( T(M) \) has property (M3). We therefore showed that if \( T(M) \) is a Mealy relation then it has properties (M1), (M2), and (M3). On the other hand, let us assume that \( T(M) \) has properties (M1), (M2), and (M3). Two states \( q_1, q_2 \in Q \) are said to be twinned if, for all states \( p_1, p_2 \in Q \) and all words \( u, v \in \Sigma^* \) and \( y_1, y_2, z_1, z_2 \in A^* \) such that \( (p_1, u, y_1, q_1), (q_1, v, z_1, q_1) \in \delta \) and \( (p_2, u, y_2, q_2) \in \delta \) it follows that either \( z_1 = z_2 = e \) or \( |z_1| = |z_2| \neq 0 \) and \( y_1 \) is a prefix of \( y_2 \) or \( y_2 \) is a prefix of \( y_1 \). It is easy to see that, because of properties (M1) and (M2), any two useful states of \( M \) are twinned and \( T(M) \) preserves prefixes. According to [24, Theorems 3.2 and 3.3], \( T(M) \) is a deterministic rational relation. Consequently, there is a trim deterministic real-time transducer, say, \( M' \) equivalent to \( M \). Since \( T(M') \) has properties (M1) and (M3), \( M' \) must be a completely specified letter-to-letter transducer whose states are all final states; i.e., \( T(M) \) is a Mealy relation. In summary, we derived from [24] that the equivalence stated in Theorem 3.1 is valid.

Let us next assume that \( M \) is a single-valued real-time transducer with \( n \) states and output alphabet \( A \) such that \( T(M) \) is a Mealy relation. Then, this relation is deterministic rational and has properties (M1), (M2), and (M3),
as we saw above. According to [24, Theorem 3.3], $M$ can be effectively transformed into an equivalent trim deterministic real-time transducer $M'$ having at most $1 + 2^n \cdot \max\{2, A\}^{2^{2^n}}$ states, where $l$ is the maximal length of an output word produced by any transition of $M$. Since $T(M)$ has properties (M1) and (M3), the integer $l$ must be positive. If $M$ is a letter-to-letter transducer, then $l = 1$. As above, properties (M1) and (M3) imply that $M$ must be a Mealy machine. Hence, we obtained from [24] upper bounds for the number of states of the Mealy machine $M'$ in Theorems 3.1 and 3.2 which are much larger than the ones stated in the theorems.

In order to prove the full Theorems 3.1 and 3.2 we present constructions entirely different from the one in [24]. Our proof is independent of [24] and does not use most of the above discussion.

**Proof of Theorems 3.1 and 3.2.** Let $M = (Q, \Sigma, \Delta, \delta, Q_I, Q_F)$ be a single-valued real-time transducer with $n$ states. As we saw above, if $T(M)$ is a Mealy relation then it has properties (M1), (M2), and (M3). Let us conversely assume that $T(M)$ satisfies (M1), (M2), and (M3). We are going to construct a Mealy machine $M'$ and a deterministic letter-to-letter transducer $M''$ both realizing $T(M)$ and having the properties stated in Theorem 3.1. If $M$ is a letter-to-letter transducer, then $M'$ also satisfies the demands of Theorem 3.2. For the construction of $M''$ property (M3) is not required. If $T(M)$ has this property, then $M'$ is a Mealy machine. This will establish the two theorems. The basic ideas for the design of $M'$ and $M''$ will be explained just before these machines are constructed. In order to demonstrate that Theorem 3.2 is much easier than Theorem 3.1, we also present a direct construction and a simple verification of the machine $M'$ in the case that $M$ is a letter-to-letter transducer.

We may assume that $M$ is trim. The construction of the Mealy machines $M'$ and $M''$ is prepared by Facts 3.3 and 3.4 and by some definitions derived from these facts.

**FACT 3.3.** Assume that $T(M)$ has property (M1). Let $p, p' \in Q$, $r \in Q$, $u, u' \in \Sigma^*$, and $v, v' \in \Delta^*$ such that $(p, u, v, r) \in \delta$ and $(p', u', v', r) \in \delta$. Then, $|u| - |v| = |u'| - |v'|$.  

**Proof.** Since $M$ is trim, there is a state $q \in Q_F$ and there are words $x, y \in \Sigma^*$ and $z, z' \in \Delta^*$ such that $(r, x, y, q) \in \delta$. Therefore, $(x, y) \in T(M)$. Since $T(M)$ has property (M1), $|ux| = |xy|$ and $|ux'| = |xy'|$. Consequently, $|u| - |v| = |y| - |x| = |u'| - |v'|$.  

**FACT 3.4.** Assume that $T(M)$ has property (M1). Let $p, p' \in Q$, $r \in Q$, $u \in \Sigma^*$, and $v, v' \in \Delta^*$ such that $(p, u, v, r) \in \delta$. Then, $|u| - |v| \leq n$.  

**Proof.** Since $M$ is trim, there is a state $q \in Q_F$ and there are words $x \in \Sigma^*$ and $y \in \Delta^*$ such that $(r, x, y, q) \in \delta$. Using pumping arguments, inductions on the length of $u$ and on the length of $x$, respectively, show the existence of words $u', x' \in \Sigma^*$ and $v', y' \in \Delta^*$ such that $(p, u', x', v', y, r) \in \delta$, $(r, x', y', q) \in \delta$, and $|u'|, |x'| < n$. Using Fact 3.3, $|u| - |v| = |u'| - |v'| \leq |u'| < n$. Since $T(M)$ has property (M1), $|ux'| = |xy'|$. Thus, $|u| - |v| = |x'| - |y'| \leq |x'| < n$. Altogether, $|u| - |v| < n$.  

Assume that $T(M)$ has property (M1). Recall that $M$ is trim. For every state $q \in Q$ we define $\text{diff}(r)$ as the uniquely determined integer of the form $|u| - |v|$, where $(p, u, v, r) \in \delta$ for some $p \in Q$, $u \in \Sigma^*$, and $v \in \Delta^*$. The uniqueness of this integer is guaranteed by Fact 3.3. We further set $d = \text{diff}(M) = \max\{|\text{diff}(r)| : r \in Q\}$. Using Fact 3.4, $d < n$.

Note that $d = 0$ if and only if $M$ is a letter-to-letter transducer. We set $Q_1 = \{r \in Q : \text{diff}(r) \geq 0\}$ and $Q_2 = Q \setminus Q_1$. Note that $Q_1 \cup Q_2 = Q$; i.e., $|Q_1| \geq 1$ and $|Q_2| \leq n - 1$. If $M$ is a letter-to-letter transducer, then $Q_1 = Q$ and $Q_2$ is empty. For every $u \in \Sigma^*$ define the set $A(u) = \{q \in Q : \text{for some } p \in Q_2, v \in \Delta^*, (p, u, v, q) \in \delta\}$.  

Assume that $T(M)$ has properties (M1), (M2), and (M3). The basic idea for the design of the Mealy machine $M'$ is to carry out a subset construction on the states of $M$ and to memorize for every state $q \in Q$ the last letter of the (uniquely determined) output word produced so far at $q$.

In order to prepare the definition of the state set of $M'$ let us define, for any $u \in \Sigma^*$, a partial mapping $\varphi_u : \Sigma^* \rightarrow \Sigma \cup \Delta$. The domain of $\varphi_u$ is $A(u)$. For every $q \in A(u) \cap Q_2$ set $\varphi_u(q) = \varepsilon$. Finally, consider any $q \in A(u) \cap Q_1$. Let $r \in \Delta^*$ be the uniquely determined word such that $(p, u, v, q) \in \delta$ for some $p \in Q_1$. The uniqueness of this word is guaranteed by the fact that $M$ is trim and single valued. Since $|u| - |v| = \text{diff}(q) < 0$, the word $v$ must be nonempty. Define $\varphi_u(q)$ as the last letter of $v$. Because of property (M3) we know that $L(M) = \Sigma^*$. Consequently, the word $u$ belongs to $L(M)$; i.e., $A(u) \cap Q_1$ is nonempty.

The set of states of $M'$ is $Q' = \{\varphi_u : u \in \Sigma^*\}$. The initial state of $M'$ is $q'_0 = \varphi_\varepsilon \in Q'$. By definition, the mapping $\varphi_\varepsilon$ has domain $A(\varepsilon) = Q$ and maps every $q \in Q$ to $\varepsilon$. The transition relation of $M'$ will be the function $\delta' : Q' \times \Sigma \times \Delta \times \varepsilon$. Let $\varphi \in Q'$ and $a \in \Sigma^*$. We are going to define $\delta'(\varphi, a)$ as $(\beta, \psi) \in Q' \times \Delta'$. Let $a \in \Sigma^*$ be arbitrary such that $\varphi = \varphi_a$. Consider the set $A = A(u)$ which is the domain of $\varphi$. Define $B = \{q \in Q' : \text{for some } p \in A, z \in \Delta^*, (p, a, z, q) \in \delta\}$.  

Note that $B = A(u|a)$. As we saw above, $B \cap Q_F$ is nonempty. The set $B$ will be the domain of $\psi$.

Let us first determine $b \in A$. Select any $q \in B \cap Q_F$, $p \in A$, and $z \in \Delta^*$ such that $(p, a, z, q) \in \delta$. If $z = \varepsilon$, then $\text{diff}(p) = \text{diff}(q) - 1 = -1$; i.e., $p \notin Q_2$. According to the definition of $\varphi$, this implies that $\varphi(p) \in A$. Consequently, $\varphi(p) = z$ is always in $A$. Define $b$ as the last letter of $\varphi(p)$ $z$. We have to show that the definition of $b$ is independent of
Let us determine ψ in Q which is a partial mapping ψ: Q =⇒ {e} ∪ A. The domain of ψ is B. For every x ∈ B ∩ Q, set ψ(x) = ε. Finally, consider any q ∈ B ∩ Q. Let p ∈ A and z ∈ A* such that (p, a, z, q) ∈ δ. If z = ε, then diff(p) = |diff(q)| − 1 < 0; i.e., p ∈ Q2. According to the definition of ψ, this implies that ψ(p) ∈ A. Consequently, ψ(p) z is always in A+. Define ψ(p) as the last letter of ψ(p) z. We have to show that the definition of ψ(p) is independent of u, p, and z; i.e., it only depends on ψ(p) z. Let u ∈ Σ* such that ψ(p) z = ψ(u) z. Note that A = A(u). Let p ∈ A and z ∈ A* such that (p, a, z, q) ∈ δ. As above, ψ(p) z is in A+. Let ψ(q') be the last letter of ψ(p') z'. Select any r, r' ∈ Q and v, v' ∈ A* such that (r, u, v, r'), (r, u, v, r') ∈ δ. By definition of ψ = ψ(a) ψ(p) is a suffix of ψ and ψ(p') is a suffix of v'. Since M is trim and single valued, v and v' coincide. Hence, ψ(p) z and ψ(p') z' are suffixes of v; i.e., b and b' coincide.

Let us thereby construct the Mealy machine M' = (Q', Σ, A, δ', {q1'}). By definition of δ', for every u ∈ Σ* and every a ∈ Σ there is a b ∈ A such that (q, a, b, q) ∈ δ'. Consequently, it can be verified by induction on the length of u that every state q ∈ Q' is attainable from q1'. Employing property (M2) we are going to show in Fact 3.5 that M' and M are equivalent. Since every q ∈ Q' has nonempty domain, we can estimate

\[ |Q'| \leq 2^{2|Q|} |1 + |A| |\Sigma| - 1 < 2 \cdot (1 + |A|)^{|\Sigma| - 1}. \]

Having in mind the fact that all states in M' are attainable from q1', the Mealy machine M' can be computed in a straightforward way in time polynomial in (1 + |A|)^{|\Sigma|}. If M is a letter-to-letter transducer, then |Q'| \leq 2^{|\Sigma|} − 1 and M' can be computed in time polynomial in 2^{|\Sigma|} |M|. Now Fact 3.5 remains to be verified.

**Fact 3.5. M' and M are equivalent.**

**Proof.** We first prove that T(M') ⊆ T(M). Let (x, y) ∈ T(M'). Let x = x1 · · · x_m and y = y1 · · · y_m, where x1, · · · , x_m ∈ Σ and y1, · · · , y_m ∈ A. Let q1, · · · , q_m+1 ∈ Q such that q1 = q'1 and (q_i, x_i, y_i, q_{i+1}) ∈ δ for every i = 1, · · · , m. According to the above remark, for every i = 1, · · · , m there is a y_i ∈ A such that (q_{i−1}, x_{i−1}, y_i, q_i, q_{i+1}) ∈ δ'. Since \( q_0 = q_1 \) and \( M' \) is deterministic, we conclude by induction on i that \( q_{n−i−1} = q_i \) for every \( i = 1, · · · , m+1 \) and that \( y_i = y_i \) for every \( i = 1, · · · , m_1 \). Let \( m = |(x, y) | \). By definition of δ' and of \( \varphi \) there are \( s_i \in A(x_1 · · · x_{i−1}) \cap Q_{r_i}, r_i \in Q_{r_i}, \) and \( v_i \in A^* \) such that \( (r_i, x_i · · · x_{i−1}, v_i, s_i, \delta) \) and \( y_i \) is the last letter of \( v_i \). Since \( T(M') \) has properties (M1) and (M2), the identity \( v_i = v_{i−1} y_i \) holds for every \( i = 1, · · · , m \). By induction on i this implies that \( y_1 = y_2 · · · y_m \); i.e., \( y_m = y \). Consequently, (x, y) \in T(M).

It remains to be shown that \( T(M') \subseteq T(M) \). Since \( M' \) is a Mealy machine, it recognizes Σ*. Let (x, y) \in T(M). As \( x \in L(M') \), there is a y' \in A* such that \( (x, y') \in T(M') \subseteq T(M) \). Since M is single valued, y and y' coincide; i.e., \( (x, y) \in T(M) \).}

Let us now turn to the construction of the deterministic letter-to-letter transducer \( M'' \). For that purpose it is sufficient to assume that \( T(M) \) has properties (M1) and (M2). The basic idea for the design of \( M'' \) is to carry out a subset construction on the states of \( M \) with delay \( d = |diff(M)| \) on the input. If \( d = 0 \), then \( M \) is a letter-to-letter transducer and \( M'' \) may be chosen as the machine \( M' \) constructed above (provided that \( T(M) \) has property (M3)) or in the direct proof of Theorem 3.2 given below. Therefore, we may assume that \( d > 0 \).

The set of states of \( M'' \) is \( Q'' = Q''_0 \cup Q''_1 \), where

\[ Q''_0 = \{(Q_1, u_2) : u_2 \in \Sigma^{|\delta|−1}, A(u_2) \neq \emptyset \} \]

and

\[ Q''_1 = \{(A(u_1), u_2) : u_1 \in \Sigma^*, u_2 \in \Sigma^{|\delta|}, \delta \cap A(u_1) \neq \emptyset \} \]

The initial state of \( M'' \) is \( q''_0 = (Q_1, \varepsilon) \in Q''_0 \). The set of final states of \( M'' \) is

\[ Q''_f = \{(A, u_2) \in Q'' : \delta \cap A \times \{u_2\} \neq \emptyset \times Q \neq \emptyset \}. \]

The transition relation of \( M'' \) will be the partial function \( \delta' : Q'' \times \Sigma \rightarrow A \times Q''. \) Recall that \( Q'' \subseteq Q''_0 \cup Q''_1 \). Let \( (Q_2, u_2) \in Q''_0 \) and \( a \in \Sigma \). If \( A(u_2 \varepsilon) = \emptyset \), then \( \delta''((Q_2, u_2 \varepsilon), a) \) is undefined. Otherwise, we are going to define \( \delta''((Q_2, u_2), a) \) as \((b, (Q_2, u_2, a)\) \in \emptyset \times Q'' \). In order to determine \( b \in A \) select any \( q \in A(u_2) \). Let \( p \in Q_2 \) and \( v_2 \in A^* \) such that \((p, u_2, v_2, q) \in \emptyset \). Since M is trim, there are \( x_{Q_2} \in Q_2, u_2 \in \Sigma^* \), and \( v_2 \in A^* \) such that \((q, u_3, x_1) \in \emptyset \). Since \((u_2, a v_2, v_2) \in T(M) \) and \( T(M) \) has property (M1), the word \( v_2 \) has length \( |u_2| \). Define \( b \) as \( (v_2, v_1) |u_2| \in A \). We have to show that the definition of \( b \) is independent of \( q, p, v_2, s, u_2, \) and \( v_2 \); i.e., it only depends on \((Q_2, u_2)\) and \( a \). Let \( p \in Q_2 \),
Having in mind the definition of $\delta^*$ and the fact that all states in $M^*$ are attainable from $q_0^*$, the transducer $M^*$ can be computed in a straightforward way in time polynomial in $(2 \# \Sigma^* \cdot \# M)$. Now Fact 3.6 remains to be verified.

**Fact 3.6.** $M^*$ and $M$ are equivalent.

The proof of Fact 3.6 is based on Facts 3.7 and 3.8.

**Fact 3.7.** Let $x \in \Sigma^*$ such that $|x| \leq d$.

(i) If $y \in A^*$ and $(A, u_2) \in Q^*$ such that $((Q_I, e), x, y, (A, u_2), \delta)$, then $(A, u_2) = (Q_I, x)$.

(ii) If $u_1, u_2, u_3 \in \Sigma^*$ and $y \in A^*$ such that $(xu_3, y) \in T(M)$, then $|y| = |x| + |u_1|$ and $((Q_I, e), x, y(1) \ldots y(|x|), (A, u_2), \delta)$.

**Fact 3.8.** Let $x \in \Sigma^*$ such that $|x| \geq d$.

(i) If $y \in A^*$ and $(A, u_2) \in Q^*$ such that $((Q_I, e), x, y, (A, u_2), \delta)$, then $(A, u_2) = d$, $x = u_1 u_2$, and for some $u_1 \in \Sigma^*$, and $A = A(u_1)$.

(ii) If $u_1, u_2, u_3 \in \Sigma^*$ and $y \in A^*$ such that $(xu_3, y) \in T(M)$, then $|y| = |x| + |u_1|$ and $((Q_I, e), x, y(1) \ldots y(|x|), (A, u_2), \delta)$, where $A = A(u_1)$.

Fact 3.7 can be easily proved by induction on the length of $x$. Fact 3.7 implies Fact 3.8 for $|x| = d$. This establishes the base of induction ($|x| = d$) in a proof of Fact 3.8 by induction on $|x| - d$. The inductive step is straightforward.

**Proof of Fact 3.6.** We first prove that $L(M^*) \subseteq L(M)$. Let $x \in L(M^*)$. Let $y \in A^*$ and $(A, u_2) \in Q^*$ such that $((Q_I, e), x, y, (A, u_2), \delta)$. Since $(A, u_2) \in Q^*$, there are $p \in A$, $q \in Q_I$, and $v_2 \in A^*$ such that $(p, u_2, v_2, q) \in \delta$. If $|x| \leq d$, then by Fact 3.7(i) $(A, u_2) = (Q_I, x)$. Thus, $p \in Q_I$ and $(x, v_2) = (u_2, v_2) \in T(M)$, i.e., $x \in L(M)$. If $|x| > d + 1$, then by Fact 3.8(ii), $(A, u_2) = d$, $x = u_1 u_2$, for some $u_1 \in \Sigma^*$, and $A = A(u_1)$. Since $p \in A = A(u_1)$, there are $r \in Q_I$ and $v_2 \in A^*$ such that $(r, u_1, v_1, p) \in \delta$. Hence, $(x, v_2, v_3) = (u_2, v_2, v_3) \in T(M)$, i.e., $x \in L(M)$.

We next prove that $T(M^*) \subseteq T(M^*)$. Let $(x, y) \in T(M)$. If $|x| \leq d$, then by Fact 3.7(ii) $((Q_I, e), x, y, (Q_I, x), \delta)$. Since $x \in L(M)$, $(Q_I, x)$ belongs to $Q^*$. Hence, $(x, y) \in T(M^*)$. If $|x| > d + 1$, let $u_1, u_2 \in \Sigma^*$ such that $|u_1| = d$ and $x = u_1 u_2$. Let $r \in Q_I, p \in Q, q \in Q_I$, and $v_2 \in A^*$ such that $y = x v_2$ and $(r, u_1, v_1, p) \in \delta$. Using Fact 3.8(ii), $(Q_I, e), x, y, (A, u_2), \delta) \in \delta^* r$. Since $p \in A = A(u_1)$, $(A, u_2) \in T(M^*)$. Hence, $(x, y) \in T(M^*)$.

It remains to be shown that $T(M^*) \subseteq T(M)$. Let $(x, y) \in T(M^*)$. Since $x \in L(M^*) \subseteq L(M)$, there is a $y' \in A^*$ such that $(x, y') \in T(M) \subseteq T(M^*)$. Since $M^*$ is deterministic.

---

File: DISTIL 151707. By CV, Date:27/02/98, Time:10:07 LOP9M. V8.B Page 01:01
Codes: 7085 Sign: 4828. Length: 56 pic 0 pts, 236 mm
This completes the proof of Theorems 3.1 and 3.2.

As an example let us consider the real-time transducer $M$ whose graph representation is displayed in Fig. 2. It can be verified that $T(M)$ has properties (M1), (M2), and (M3). However, $M$ is neither deterministic nor a letter-to-letter transducer. Note that $\text{diff}(q_0) = \text{diff}(q_s) = 0$, $\text{diff}(q_1) = 1$, and $\text{diff}(M) = 1$. The Mealy machines $M'$ and $M''$ equivalent to $M$ which are obtained by applying Theorem 3.1 to $M$ can be also seen in Fig. 2. Note that $M'$ has three states and $M''$ has five states. The states $(A_0, \varepsilon)$, $(A_1, 0)$, and $(A_1, 1)$ of $M''$ can be identified which yield a Mealy machine isomorphic to $M'$. Note that $M'$ is isomorphic to the Mealy machine shown in Fig. 1.

Direct proof of Theorem 3.2. Let $M = (Q, \Sigma, A, \delta, Q', Q_f)$ be a single-valued letter-to-letter transducer with $n$ states such that $T(M)$ is a Mealy relation. As we saw before the proof of Theorem 3.1, $T(M)$ has properties (M1), (M2), and (M3). In the present proof we generally only need to assume that $T(M)$ satisfies (M1) and (M2). We are going to construct a deterministic letter-to-letter transducer $M''$ realizing $T(M)$ and having the properties stated in Theorem 3.2. If $T(M)$ has property (M3), then $M'$ is a Mealy machine, which is isomorphic to the Mealy machine $M''$ constructed in the above proof of Theorems 3.1 and 3.2.

By this, Theorem 3.2 will be established. The basic feature of the present approach is that the machine $M''$ also has a meaning if property (M3) is not present and that the proof of its correctness is very simple. The construction of $M'$ is derived from the Mealy machine $M''$ constructed in the proof of Theorem 3.1 when setting $d = 0$.

We may assume that $M$ is trim. From the above proof of Theorems 3.1 and 3.2 we recall, for every $u \in \Sigma^*$, the definition of the set

$$A(u) = \{ q \in Q : \text{for some } p \in Q, v \in A^*, (p, u, v, q) \in \delta \}.
$$

Let us construct the letter-to-letter transducer $M' = (Q', \Sigma, A, \delta', \{ q'_0 \}, \{ Q'_f \})$ by setting

$$Q' = \{ A(u) : u \in \Sigma^*, A(u) \neq \emptyset \},$$

$$\delta' = \{ (A, a, b, B) \in Q' \times \Sigma \times A \times Q' :$$

$$B = \{ q \in Q : \text{for some } p \in A \text{ and } b' \in A, (p, a, b', q) \in \delta \},$$

there are $p \in A$ and $q \in B$ such that $(p, a, b, q) \in \delta$, \}

$q'_0 = \emptyset$, and $\{ Q'_f \} = \{ A \in Q' : A \cap Q_f \neq \emptyset \}$.

Since $M$ is a trim letter-to-letter transducer and $T(M)$ has property (M2), $M'$ must be deterministic. For every $(A, a, b, B) \in \delta'$ and every $q \in B$ there are $p \in A$ and $b' \in A$ such that $(p, a, b', q) \in \delta$. Since $M'$ is deterministic, $b$ and $b'$ coincide; i.e., $(p, a, b, q) \in \delta$. On the other hand, for every...
(p, a, b, q) ∈ δ and every A ∈ Q' such that p ∈ A there is a B ∈ Q' such that q ∈ B and (A, a, b, h) ∈ δ'. Now it is easy to verify that M' and M are equivalent.

It can be seen by induction on the length of u, that every state A(u1) ∈ Q' is attainable from q1'. Since M' and M are equivalent, T(M') inherits property (M3) from T(M). Therefore, if T(M) has property (M3) then M' must be completely specified and all its states must be final; i.e., M' is a Mealy machine. Note that # Q' ≤ 2^n − 1. Having in mind the fact that all states in M' are attainable from q1', the transducer M' can be computed in a straightforward way in time polynomial in 2^n . |M|.

4. LOWER BOUNDS

In this section we prove lower bounds for the transformation of a single-valued transducer into an equivalent Mealy machine if such a machine exists. The outcome is stated in Theorems 4.1 and 4.2.

**Theorem 4.1.** For every integer t ≥ 2 and every odd integer n ≥ 3 there is a single-valued real-time transducer M with n states and input and output alphabets of cardinality t such that M is equivalent to some Mealy machine M' and every such M' has at least t(n−1)/2 states.

**Theorem 4.2.** For every odd integer n ≥ 3 there is a single-valued-letter-to-letter transducer M with n states and ternary input and output alphabets such that M is equivalent to some Mealy machine M' and every such M' has at least 2(n−1)/2 states.

Theorems 4.1 and 3.1 leave a gap between a lower bound # 2^(n−1)/2 (or # 4^(n−1)/2) and an upper bound min{|9/4|): 2, 2−|Σ|^−1, 2−(1+|δ|)^−1} and Theorems 4.2 and 3.2 leave a gap between a lower bound 2^(n−1)/2 and an upper bound 2^n−1. We have no idea how to diminish these gaps significantly. It should be, however, an interesting problem to find a lower bound greater than 2^n for Theorem 4.1, provided that # Σ = # A = 2. We want to point out that [24, Proposition 3.1] does not help to prove Theorems 4.1 and 4.2, since the transducers employed there are in fact automata.

In order to prove Theorems 4.1 and 4.2 we need a necessary and sufficient condition for a Mealy machine to be minimal. Let M = (Q, Σ, A, δ, {q1}, Q) be a Mealy machine. Recall from Section 2 that a state q ∈ Q is attainable from q1 if there are words x ∈ Σ* and y ∈ A* such that (q1, x, y, q) ∈ δ. We also say that q is attainable from q1, by means of the input word x. Two states p1, p2 ∈ Q are distinguishable if there are states q1, q2 ∈ Q and words x ∈ Σ* and y1, y2 ∈ A* such that (p1, x, y1, q1) ∈ δ, (p2, y2, q2) ∈ δ, and y1 and y2 are distinct. We also say that the input word x distinguishes p1 and p2 by means of the output words y1 and y2. The machine M is minimal if every equivalent Mealy machine has at least as many states as M. According to Proposition A.1 in Appendix A, M is minimal if and only if all states in Q are attainable from q1, and any two distinct states in Q are distinguishable.

The remainder of this section is devoted to the proofs of Theorems 4.1 and 4.2.

**Proof of Theorem 4.1.** Let t and n be as in the theorem. Set Σ = A = {a1, ..., a2} and v = (n−1)/2 ≥ 1. Define the homomorphism h : Σ* → A* by setting h(a1) = a1 for every a ∈ Σ. Define the relation ρ = ρ1 ∪ ρ2 ∪ ρ3 = Σ* × A* by setting

ρ1 = \{(w1w3, h(w1)) : w1, w3 ∈ Σ*, i ∈ {0, ..., v}\},
ρ2 = \{(w1aw3, h(w1)a) : a ∈ Σ, w1, w3 ∈ Σ*, i ∈ {0, ..., v−1}\},
ρ3 = \{(w1w2w3, h(w1)w2) : w1, w3 ∈ Σ*, w2 ∈ Σ+\}.

Given a word w ∈ Σ*, we may imagine that ρ factors w into w1w2w3 such that |w1| = min{|v, |w1|/2|}, |w2| = |w| − 2 − |w1|, and |w3| = |w1| and then maps it to h(w1)w2.

The relation ρ is realized by the real-time transducer M = (Q, Σ, A, δ, Q1, Q2), where Q = {q0, ..., q2n−1}, δ = δ1 ∪ ... ∪ δ5, Q1 = {q0}, Q2 = {q0, q2}, and

δ1 = \{(qi, ai, a2i, q2i+1) : i ∈ {0, ..., v−1}, a ∈ Σ\},
δ2 = \{(qi, ai, e, qi+1) : i ∈ {v, ..., 2v−1}, a ∈ Σ\},
δ3 = \{(qi, ai, a, q2i−i) : i ∈ {0, ..., v−1}, a ∈ Σ\},
δ4 = \{(qi, ai, e, q2i−1−i) : i ∈ {1, ..., v−1}, a ∈ Σ\},
δ5 = \{(qi, ai, a, q)e : a ∈ Σ\}.

The transducer M is neither deterministic nor a letter-to-letter transducer. It has n states and 2 − (n−1)/2 · |Σ| transitions. Note that diff(M) = (n−1)/2. For t = 2, a1 = 0, a2 = 1, and n = 7 a graph representation of M is displayed in Fig. 3.

It is easy to see that the relation ρ = T(M) has properties (M1), (M2), and (M3). According to Theorem 3.1 this means that ρ is a Mealy relation. Indeed, it is realized by the Mealy machine M' = (Q', Σ, A, δ', {q'1}, Q'), where

Q' = \{qh(w) : w ∈ Σ ≤ (v−1)/2\}
∪ \{qad(a) : a ∈ Σ, w ∈ Σ ≤ (v−2)/2\}
∪ \{q e : u ∈ Σ\},
δ' = \{(q, a, a, q) : a ∈ Σ\}
∪ \{(qb, a, b, qad(a)) ∈ Q' × A × Σ : u ∈ Σ ≤ v−2\}
∪ \{(qb, a, b, qm) ∈ Q' × Σ × A : u ∈ Σ ≤ v−1\},
q'1 = q1.
We remark that the computational problem of $M'$ is to delay the output of $M$. The output of $M$ which still has to be produced by $M'$ makes up the index of the current state of $M$. Note that $\# Q' \geq \# \Sigma^n = \# \Sigma^{n-1}/2$ for $t = 2, a_1 = 0, a_2 = 1$, and $n = 7$ a graph representation of the Mealy machine $M'$ equivalent to $M$ is displayed in Fig. 3.

Following the characterization stated in Proposition A.1, we now show that $M'$ is a minimal Mealy machine. Select any $a \# 7$. Let $q_{a}$. If $u = b_{1}h(w)$ for some $b_{1} \in \Sigma$ and $w \in \Sigma^{\leq |w|^{2}}$, then $q_{a}$ is attainable from $q_{b}$. If $u = b_{1}h(w)$ for some $b_{1} \in \Sigma$ and $w \in \Sigma^{\leq |w|^{2}}$, then $q_{a}$ is attainable from $q_{b}$ by means of the input word $a_{b_{1}}w$. Finally, if $u \in \Sigma^{*}$, then $q_{a}$ is attainable from $q_{b}$ by means of the input word $a_{b_{1}}w$. Hence, every state of $M'$ is attainable from $q_{a}$. Let $q_{u_{1}}, q_{u_{2}} \in Q'$ be distinct. If $u_{1}$ and $u_{2}$ differ at some position, then the input word $a_{b}$ distinguishes $q_{u_{1}}$ and $q_{u_{2}}$ by means of the output words $u_{1}a_{b}^{-|u_{2}|}$ and $u_{2}a_{b}^{-|u_{1}|}$. If $u_{1}$ is a proper prefix of $u_{2}$, i.e., $u_{1}/u_{2} = u$, for some $b_{1} \in \Sigma$ and $u \in \Sigma^{*}$, then, for any $b_{2} \in \Sigma \backslash \{b_{1}\}$, the input word $a_{b_{2}}$ distinguishes $q_{u_{1}}$ and $q_{u_{2}}$ by means of the output words $u_{1}b_{2}^{-|u_{2}|}$ and $u_{1}b_{2}^{-|u_{1}|}$. The case that $u_{2}$ is a proper prefix of $u_{1}$ can be treated symmetrically. Hence, any two distinct states of $M'$ are distinguishable.

This completes the proof of Theorem 4.1.

Proof of Theorem 4.2. Let $n$ be as in the theorem. Set $r = (n - 1)/2 \geq 1$ and $\Sigma = \Delta = \{0, 1, \#\}$. Define the language $L = \{w_{1}w_{2} : w_{1}, w_{2} \in \Sigma^{*}, |w_{1}| = r - 1\}$. Define the relation $\rho = \{(w_{1} \# \cdots \# w_{k} \# w_{k+1}, w_{1} \# \cdots \# w_{k} \# w_{k+1}) : k \geq 0, w_{1}, \ldots, w_{k+1} \in \{0, 1\}^{*}, x_{1}, \ldots, x_{k} \in \{0, 1\}\}$.

Note that $\rho \leq \{0, 1\}^{*}$. Define the relation $\rho = \{(w_{1} \# \cdots \# w_{k} \# w_{k+1}, w_{1} \# \cdots \# w_{k} \# w_{k+1}) : k \geq 0, w_{1}, \ldots, w_{k+1} \in \{0, 1\}^{*}, x_{1}, \ldots, x_{k} \in \{0, 1\}\}$.

Note that $\rho \leq \Sigma^{*} \times \Delta^{*}$. Given a word $w \in \Sigma^{*}$, we may imagine that $\rho$ replaces every occurrence of $\#$ in $w$ by $1$ or $0$, depending on whether or not the longest factor $u \in \{0, 1\}^{*}$ of $w$ which appears directly left of this occurrence of $\#$ belongs to $L$.
The relation $\rho$ is realized by the letter-to-letter transducer $M = (Q, \Sigma, \Delta, \delta, Q_0, Q_f)$, where $Q = \{q_0\} \cup \{q_i; i \in \{0, 1\}\}$, $j \in \{1, ..., v\}$, $\delta = \delta_1 \cup \cdots \cup \delta_5$, $Q_f = \{q_0\} \cup \{q_i; j \in \{1, ..., v\}\}$, $Q = \{q_0\}$, and

\[
\delta_1 = \{(q_0, a, a, q_0) : a \in \{0, 1\}\}, \\
\delta_2 = \{(q_0, a, \alpha, q_0) : a \in \{0, 1\}\}, \\
\delta_3 = \{(q_{i,v}, a, a, q_{i,v}) : i \in \{0, 1\}, j \in \{1, ..., v-1\}, a \in \{0, 1\}\}, \\
\delta_4 = \{(q_{i,v}, \#, a, q_0) : a \in \{0, 1\}\}, \\
\delta_5 = \{(q_{i,v}, \#, a, q_0) : a \in \{0, 1\}, j \in \{1, ..., v\}\}.
\]

The transducer $M$ is nondeterministic. It has $n$ states and $3n - 1$ transitions. For $n = 7$ a graph representation of $M$ is displayed in Fig. 4.

It is easy to see that the relation $\rho = T(M)$ has properties (M1), (M2), and (M3). According to Theorem 3.1 this means that $\rho$ is a Mealy relation. In fact, it is realized by the Mealy machine $M' = (Q', \Sigma, \Delta', \delta', \{q_0'\}, Q')$, where

\[
Q' = \{q_w : w \in \{0, 1\}^*\}, \\
\delta' = \{q_{bw}, a, a, q_{bw} : a, b \in \{0, 1\}, w \in \{0, 1\}^{r-1}\} \\
\cup \{q_{bw}, \#, b, q_0 : b \in \{0, 1\}, w \in \{0, 1\}^{r-1}\}, \\
q_0' = q_0.
\]

We remark that the computational problem of $M'$ is to queue the—at most $v$—last input symbols in $\{0, 1\}$ successively consumed by $M$. These input symbols, possibly filled up with leading zeros, make up the index of the current state of $M'$. Note that $\# Q' = 2^r = 2^{n-1/2}$. For $n = 7$ a graph representation of the Mealy machine $M'$ equivalent to $M$ is displayed in Fig. 4.

Following the characterization stated in Proposition A.1, we now show that $M'$ is a minimal Mealy machine. Any state $q_u \in Q'$ is attainable from $q_f$ by means of the input word $u$. Hence, every state of $M'$ is attainable from $q_f$. Let $q_{uw}, q_{uw} \in Q'$ be distinct. Let $\mu \in \{1, ..., v\}$ such that $u_{\mu}(\mu)$ and $u_{\mu}(\mu)$ are distinct. Then the input word $0^n \#$ distinguishes $q_{uw}$ and $q_{uw}$ by means of the output words $0^n - 1u_{\mu}(\mu)$ and $0^n - 1u_{\mu}(\mu)$. Hence, any two distinct states of $M'$ are distinguishable.

This completes the proof of Theorem 4.2.  

5. ALGORITHMS

In this section we prove the following theorem.

**Theorem 5.1.** It is a $\text{PSPACE}$-complete problem to decide whether or not a given single-valued real-time transducer $M$ is equivalent to some Mealy machine. The problem remains $\text{PSPACE}$-complete if $M$ is known to be a letter-to-letter transducer.
Proposition 5.1 implies that Theorem 5.1 also holds if \( M \) is not real-time. The proof of Theorem 5.1 is given in the remainder of this section. Let us first discuss what is easy and what can be derived from previous work.

Let \( M \) be a single-valued real-time transducer. As discussed at the beginning of Section 3, \( T(M) \) is a Mealy relation if and only if it is a deterministic rational relation having properties (M1) and (M3). According to [24, Theorem 4.3(ii)] it is decidable in time polynomial in \( |M| \) whether or not \( T(M) \) is a deterministic rational relation. There is a straightforward nondeterministic polynomial-space algorithm for deciding whether \( T(M) \) does not have property (M3); i.e., the problem to decide whether \( T(M) \) has property (M3) belongs to PSPACE. Therefore, in order to complete the proof of Theorem 5.1 it remains to establish Lemmas 5.2 and 5.3.

**Lemma 5.2.** It is decidable in time polynomial in \( |M| \) whether or not the transduction realized by a given single-valued real-time transducer \( M \) has property (M1).

**Lemma 5.3.** The problem to decide whether or not a given single-valued letter-to-letter transducer is equivalent to some Mealy machine is PSPACE-hard.

Our proof of Lemma 5.2 employs elementary graph algorithms (see, e.g., [7, Section 23]). Note that a similar proof shows that this lemma also holds if (M1) is replaced by (M2). Therefore, Theorem 5.1 can be proved by simply combining Theorem 3.1, the above-mentioned tests for properties (M1), (M2), and (M3), and Lemma 5.3.

**Proof of Lemma 5.2.** Let \( M = (Q, \Sigma, A, \delta, Q_I, Q_F) \) be a single-valued real-time transducer. By means of a breadth-first search of the graph representing \( M \) starting from the vertices in \( Q_F \) and of the reversal of this graph starting from the vertices in \( Q_I \) one can remove all useless states from \( M \) in time polynomial in \( |M| \). We may therefore assume that \( M \) is trim. If there are \( p, q \in Q I, a_1, a_2 \in \Sigma \), and \( z_1, z_2 \in A^* \) such that \((p, a_1, z_1, q), (p, a_2, z_2, q) \in \delta \) and \( z_1 \) and \( z_2 \) have distinct lengths, then it can be seen that \( T(M) \) does not satisfy (M1). Since the former property can be tested in time polynomial in \( |M| \), we may assume that for every \((p, q) \in Q^2 \) there is at most one nonnegative integer \( l \) such that \( \delta \cap \{p\} \times \Sigma \times A^l \times \{q\} \) is nonempty.

Define the partial mapping \( f: Q^2 \to \mathcal{X} \) by setting, for every \((p, q) \in Q^2, f(p, q) = 1 - |q| \) if \((p, a, z, q) \in \delta \) for some \( a \in \Sigma \) and \( z \in A^* \); \( f(p, q) \) is undefined if such \( a \) and \( z \) do not exist. Denote the domain of \( f \) by \( D \). Consider the directed graph \( G \) with vertex set \( Q \) and edge set \( D \). Note that \( G \) has at most \#A edges. By means of a breadth-first search of \( G \) starting from \( Q_I \) one can compute a spanning forest, say, \( S \) of \( G \) rooted at the vertices in \( Q_I \). For every vertex \( q \) of \( G \), \( g(q) \) is defined as the sum of values under \( f \) along the edges of the unique path in \( S \) initiating at some vertex in \( Q_I \) and terminating at \( q \). Note that \( g(q) = 0 \) for all vertices \( q \) belonging to \( Q_I \). Now, it is easy to see that \( T(M) \) does not satisfy (M1) if and only if the graph \( G \) has property (*).

(*) There is a vertex \( q \) of \( G \) such that \( q \) belongs to \( Q_I \) and \( g(q) \neq 0 \) or there is an edge \( e = (p, q) \) of \( G \) such that \( g(q) \neq g(p) + f(e) \).

The mapping \( f \) and the graph \( G \) can be determined in time polynomial in \( |M| \). The spanning forest \( S \) and the mapping \( g \) can be computed in time linear in the number of vertices and edges of \( G \). Property (\#) of \( G \) can be tested within the same time bound. Altogether it is decidable in time polynomial in \( |M| \) whether or not the transduction realized by \( M \) has property (M1).

**Proof of Lemma 5.3.** Let us consider the problem to decide whether or not a given automaton with input alphabet \( \Sigma \) recognizes \( \Sigma^* \). This problem is known to be PSPACE-complete [9, Section A10.1]. We therefore wish to establish a polynomial-time reduction of this problem to the problem of Lemma 5.3. Let \( M \) be some given automaton with input alphabet \( \Sigma \). Replacing every transition \((p, a, e, q) \) of \( M \) by \((p, a, a, q) \) we get a single-valued letter-to-letter transducer, say, \( M' \) realizing the transduction \( \{(x, x): x \in L(M)\} \). Evidently, \( T(M') \) has properties (M1) and (M2). Therefore, using Theorem 3.1, \( T(M') \) is a Mealy relation if and only if it has property (M3); i.e., the automaton \( M \) recognizes \( \Sigma^* \). Finally observe that \( M' \) has size at most \( 2 \cdot |M| \) and can be computed in time linear in \( |M| \).

**APPENDIX A: MINIMAL MEALY MACHINE**

Let \( M = (Q, \Sigma, A, \delta, \{q_I\}, Q) \) be a Mealy machine. Recall the notions for \( M \) introduced before the proofs of Theorems 4.1 and 4.2. The following result is very similar to a well-known result for deterministic automata.

**Proposition A.1.** The Mealy machine \( M \) is minimal if and only if all states in \( Q \) are attainable from \( q_I \) and any two distinct states in \( Q \) are distinguishable.

Since we did not find an explicit proof of Proposition A.1 in the literature, we present it in this appendix for the reader's convenience. The approach is similar to the usual one for the minimization of a deterministic automaton (see, e.g., [12, Section 3.4]). Let \( \Sigma \) and \( A \) be nonempty finite sets. Consider any relation \( \rho \subseteq \Sigma^* \times A^* \) having properties (M1), (M2), and (M3) introduced at the beginning of Section 3. It can be seen that \( \rho \) is a function \( \rho: \Sigma^* \to A^* \) which preserves prefixes. For any words \( u_1, u_2 \in \Sigma^* \) we write \( u_1 \sim_\rho u_2 \) if for every word...
\[ x \in \Sigma^* \text{ the words } y_1, y_2 \in A^* \text{ coincide, where } \rho(u_1) y_1 = \rho(u_2) y_2 = \rho(u_3) x, \] 

We thereby defined an equivalence relation \( \sim_\rho \) on \( \Sigma^* \).

Now, let \( \rho \) be the transduction realized by any Mealy machine \( M = (\Sigma, \Pi, \delta, \{q_I\}, Q) \). According to Theorem 3.1, the relation \( \rho \) has properties (M1), (M2), and (M3). Note that if \( (q_I, x, y, q) \in \delta \) for some \( x \in \Sigma^* \), \( y \in A^* \), and \( q \in Q \) then \( y = \rho(x) \). In order to prove Proposition A.1 we need one more lemma.

**Lemma A.2.** The relation \( \sim_\rho \) has index at most \( \# Q \). If all states in \( Q \) are attainable from \( q_I \) and any two distinct states in \( Q \) are distinguishable, then the index of \( \sim_\rho \) is exactly \( \# Q \).

**Proof.** For any \( u_1, u_2 \in \Sigma^* \) we write \( u_1 \sim_M u_2 \) if the states \( q_1, q_2 \in Q \) coincide, where \( (q_1, u_1, p(u_1), q_1), (q_2, u_2, p(u_2), q_2) \in \delta \). We thereby defined an equivalence relation \( \sim_\rho \) on \( \Sigma^* \). Let \( u_1, u_2 \in \Sigma^* \) such that \( u_1 \sim_M u_2 \). Consider the state \( q \in Q \) for which \( (q_1, u_1, p(u_1), q), (q_2, u_2, p(u_2), q) \in \delta \). For any \( x \in \Sigma^* \), let \( y \in A^* \) and \( r \in Q \) such that \( (q, x, y, r) \in \delta \). Then, \( \rho(u_1 x) = \rho(u_1) y \) and \( \rho(u_2 x) = \rho(u_2) y \). Since \( x \) was arbitrary, it holds that \( u_1 \sim_\rho u_2 \). We therefore showed that the partition of \( \Sigma^* \) induced by \( \sim_\rho \) refines the partition induced by \( \sim_M \). By definition, the index of \( \sim_\rho \) coincides with the number of states in \( Q \) which are attainable from \( q_I \). As a first consequence, the index of \( \sim_\rho \) is at most \( \# Q \).

Assume that the relations \( \sim_\rho \) and \( \sim_M \) are distinct. Since \( \sim_\rho \) is a subset of \( \sim_M \), there must be words \( u_1, u_2 \in \Sigma^* \) such that \( u_1 \sim_M u_2 \) but not \( u_1 \sim_\rho u_2 \). Let \( p_1, p_2 \in Q \) such that \( (q_1, u_1, p(u_1), q_1), (q_2, u_2, p(u_2), q_2) \in \delta \). Since \( u_1 \sim_M u_2 \) does not hold, the states \( p_1 \) and \( p_2 \) must be distinct. For any \( x \in \Sigma^* \), let \( y_1, y_2 \in A^* \) such that \( (p_1, x, y_1), (p_2, x, y_2) \in Q \). Then, \( \rho(u_1 x) = \rho(u_1) y_1 \) and \( \rho(u_2 x) = \rho(u_2) y_2 \). As \( u_1 \sim_\rho u_2 \), the words \( y_1 \) and \( y_2 \) coincide. Since \( x \) was arbitrary, the states \( p_1, p_2 \in Q \) are indistinguishable. Consequently, if all states in \( Q \) are attainable from \( q_I \) and any two distinct states in \( Q \) are distinguishable, then the relations \( \sim_\rho \) and \( \sim_M \) coincide and the index of \( \sim_\rho \) is exactly \( \# Q \).

**Proof of Proposition A.1.** (Only if) Assume that some state \( q \) of \( M \) is not attainable from \( q_I \). Then we can remove from \( M \) all states which are not attainable from \( q_I \), including \( q \), and all transitions initiating at them. By this we obtain a Mealy machine being equivalent to \( M \) and having fewer states; i.e., the machine \( M \) was not minimal.

(If) Assume that all states in \( Q \) are attainable from \( q_I \) and that any two distinct states in \( Q \) are distinguishable. Let \( M' = (\Sigma', \Pi, \delta', \{q'_I\}, Q') \) be any Mealy machine equivalent to \( M \) according to Lemma A.2 applied to \( M' \) and \( \rho \), the relation \( \sim_\rho \) has index exactly \( \# Q \). Let \( \rho = \tau(M) \). From Lemma A.2 applied to \( M' \) and \( \rho \), the index of \( \sim_\rho \) is at most \( \# Q' \); i.e., \( \# Q \leq \# Q' \). Consequently, the machine \( M \) is minimal.

**Acknowledgments**

I thank Reinhard Klemm for his comments and suggestions on several versions of this article, Christoph Grimm for pointing out relations to circuit design problems, and the referee for his or her remarks concerning the presentation.

**References**


