JOURNAL OF COMPUTER AND SYSTEM SCIENCES 19, 45–49 (1979)

The Decidability of Equivalence for Deterministic Finite Transducers*

MEERA BLATTNER

Rice University, Houston, Texas, 77001

AND

TOM HEAD

University of Alaska, Fairbanks, Alaska, 99701

Received December 16, 1977

An algorithm is given which will decide, for two given deterministic finite transducers M and M', whether the input-output behaviours of M and M' are identical.

1. INTRODUCTION

A finite transducer is a nondeterministic finite-state machine with final states. State transitions are made on single symbol inputs or on e (empty) input. These transducers differ only superficially from a-transducers as defined in Ginsburg [3]. The a-transducer formalism merely allows the extra flexibility of having single step state transitions made on the input of strings of symbols. Finite transducers have been proposed as formal models for certain translation processes in Aho and Ullman [1].

From a result of Griffiths [4] it follows that equivalence of finite transducers is not decidable. Aho and Ullman [1] define the concept of a deterministic finite transducer. A finite transducer is deterministic if the state transitions are fully determined for each symbol of an input string except that once the last symbol of the string is input a succession of e-moves may be possible. Aho and Ullman [1] propose as open problem 3.1.29: Is it decidable whether two deterministic finite transducers are equivalent? Our purpose is to give an affirmative answer to this question. The demonstration closely parallels the demonstration of the decidability of equivalence for single-valued a-transducers given in Blattner and Head [2]. In particular, both demonstrations make essential use of a result concerning free monoids which we state in section 3 below. Our notation and exposition here are chosen to make the similarities and differences between the demonstrations of these two decidability results transparent.

* This research was partly supported by NSF Grant MCS77-02470.

0022-0000/79/040045-05$02.00/0

2. Preliminaries

DEFINITION. A *finite transducer* M is a 6-tuple $(K, \Sigma, \Delta, H, q_0, F)$ where:

(i) K, Σ, and Δ are finite sets called the set of states, the input alphabet, and the output alphabet, respectively;

(ii) H is a finite set of 4-tuples (q_i, x, y, q_j) where $q_i \in K$, $x \in \Sigma \cup \{e\}$, $y \in \Delta^*$, and $q_j \in K$;

(iii) q_0 is an element of K called the start state;

(iv) F is a subset of K called the set of final states.

For such an M a *path* is a finite sequence of elements of H with the property that, for each pair of successive elements (q_m, x, y, q_n), (q_r, x', y', q_s) in the sequence, $q_n = q_r$. A path $(q_1, x_2, y_2, q_2), \ldots, (q_{i-1}, x_i, y_i, q_i), (q_i, x_{i+1}, y_{i+1}, q_{i+1}), \ldots, (q_{n-1}, x_n, y_n, q_n)$ will be said to be a path *from* q_1 to q_n associated with the *input* $x_2 \cdots x_i x_{i+1} \cdots x_n$ and *output* $y_2 \cdots y_i y_{i+1} \cdots y_n$.

The central notion in terms of which the input-output behaviour of M is specified is that of a transduction: a path from q_0 to a final state having associated input $x(\in \Sigma^*)$ and output $y(\in \Delta^*)$ is called a *transduction of input x into output y via M*. For each $x \in \Sigma^*$, we define $M(x) = \{y \in \Delta^* \mid \text{there exists a transduction } T \text{ of } x \text{ into } y \text{ via } M\}$. The *domain* of M is $\mathrm{Dom}(M) = \{x \in \Sigma^* \mid M(x) \neq \varnothing\}$.

DEFINITION. The finite transducers $M = (K, \Sigma, \Delta, H, q_0, F)$ and $M' = (K', \Sigma, \Delta, H', q_0', F')$ are *equivalent* ($M \equiv M'$) if, for each $x \in \Sigma^*$, $M(x) = M'(x)$.

DEFINITION. A finite transducer $M = (K, \Sigma, \Delta, H, q_0, F)$ is *deterministic* if the following condition holds for all $q \in K$: Either

(i) H contains no 4-tuple of the form (q, e, y, q_j), $y \in \Delta^*$, $q_j \in K$, and for each $a \in \Sigma$, H contains at most one 4-tuple of the form (q, a, y, q_j), $y \in \Delta^*$, $q_j \in K$, or

(ii) H contains only one 4-tuple of the form (q, e, y, q_j), $y \in \Delta^*$, $q_j \in K$, and no other 4-tuples having q as a first coordinate.

See Aho and Ullman [1] for examples and further general discussion of finite transducers. Our definition of a finite transducer is equivalent to, although not identical with, the definition given in Aho and Ullman. Our formulation allows our proofs to be given in a slightly clearer form. The concept of a finite transducer is identical with that of a prepared a-transducer as given in Blatter and Head [2].

3. A Theorem on Free Monoids

In Blatter and Head [2] a theorem concerning free monoids was proved, but not stated. This result is a fundamental tool for proving the decidability of equivalence of deterministic finite transducers as it was also in proving all decidability results in the earlier paper.

THEOREM 1. *Let* $s_1, s_2, s_3, s_4, s_5, s_1', s_2', s_3', s_4', s_5'$, be *elements of a free monoid* Σ^*. *If*

$$s_1 s_3 s_5 = s_1' s_3' s_5', \tag{3.1}$$

$$s_1 s_2 s_3 s_5 = s_1' s_2' s_3' s_5', \quad \textit{and} \tag{3.2}$$

$$s_1 s_3 s_4 s_5 = s_1' s_3' s_4' s_5' \tag{3.3}$$

then

$$s_1 s_2 s_3 s_4 s_5 = s_1' s_2' s_3' s_4' s_5'. \tag{3.4}$$

Proof. Lines 1 through 25 of page 314 of Blatter and Head [2] constitute this proof.

4. DECIDABILITY OF EQUIVALENCE

The deterministic condition is so strong that it forces the output set $M(w)$ of an input string w to take a particularly transparent form. The following lemma elucidates this form only to the degree required for the proof of decidability.

LEMMA. *For a deterministic finite transducer* M, *and an input string* w, *either* $M(w)$ *is empty or* $M(w)$ *contains a unique string* s *of minimal length and* $M(w) = sR$ *for a regular set* R.

Proof. Assume that $M(w)$ is not empty. For $w \neq e$, there is a unique shortest path $(q_0, u_1, s_1, q_1), (q_1, u_2, s_2, q_2), \ldots, (q_{n-1}, u_n, s_n, q_n)$ such that $u_1 u_2 \cdots u_n = w$. Note that some u_i may be e, but $u_n \neq e$ by the choice of a *shortest* path. If $w = e$ and $M(e)$ is not empty we use the 'empty' path q_0 to q_0 as our shortest path and here $q_n = q_0$.

Consider those states accessible from q_n by sequences of e-moves. The deterministic condition requires that these states consist of either (i) q_n alone, or (ii) the states appearing in a linear path $(q_n, e, s_{n+1}, q_{n+1}), \ldots, (q_{n+m-1}, e, s_{n+m}, q_{n+m})$, or (iii) the states appearing in a 'figure 6' path: $(q_n, e, s_{n+1}, q_{n+1}), \ldots, (q_{n+m-1}, e, s_{n+m}, q_{n+m}), (q_{n+m}, e, s_{n+m+1}, q_{n+m+1}), \ldots, (q_{n+m+k-1}, e, s_{n+m+k}, q_{n+m})$.

Since $M(w)$ is not empty there must be at least one final state accessible from q_n by a sequence of e-moves. In each of the three cases described above the meaning and truth of the following assertion are clear: When $n + p$ $(p \geqslant 0)$ is chosen to be the least subscript for which q_{n+p} is a final state, $s_1 \cdots s_n \cdots s_{n+p}$ is the (unique) shortest string in $M(w)$ and $M(w) = s_1 \cdots s_n \cdots s_{n+p} R$ for a regular set R.

With each transduction T via a finite transducer M we will associate a sequence of states, Seq T. To define Seq T we let T be denoted:

$$T\colon (q_0, x_1, y_1, q_1), (q_1, x_2, y_2, q_2), \ldots, (q_{i-1}, x_i, y_i, q_i),$$
$$(q_i, x_{i+1}, y_{i+1}, q_{i+1}), \ldots, (q_{n-1}, x_n, y_n, q_n).$$

Then Seq T is the subsequence of $q_0, q_1, q_2, \dots, q_{i-1}, q_i, q_{i+1}, \dots, q_{n-1}, q_n$ defined as follows: Seq T will start with q_0 and end with q_n, but will contain q_i $(0 < i < n)$ only when the following two conditions hold

(i) $x_i \neq e$, and
(ii) there is a j such that $i < j \leqslant n$ and $x_j \neq e$.

We illustrate: For a transduction T: (q_0, e, aa, q_1), (q_1, a, a, q_2), (q_2, a, e, q_1), (q_1, e, a, q_3), (q_3, a, a, q_4), (q_4, e, a, q_5) via some unspecified M, we have Seq T: q_0, q_2, q_1, q_5. (Note that the number of terms in Seq T is $1 +$ length $(x_1 \cdots x_n)$.)

THEOREM 2. *Let M and M' be deterministic finite transducers having the same domain and having k and k' states respectively. If $M(x) = M(x')$ for every input string x of length at most $2kk' - 1$, then M and M' are equivalent.*

Proof. Assume that the theorem is false. Then there are M, M' with k, k' states such that $M(x) = M'(x)$ for all input strings of length at most $2kk' - 1$ and yet M and M' are not equivalent. Then there is an input string w of least length subject to the condition $M(w) \neq M'(w)$. The length of w must be at least $2kk'$. Since M and M' have the same domain and $M(w) \neq M'(w)$, it follows that neither $M(w)$ nor $M'(w)$ is empty.

By the lemma above, for strings s, s' of least length in $M(w)$, $M'(w)$, respectively, $M(w) = sR$ and $M'(w) = s'R'$ for regular sets R, R'.

Let T and T' be transductions of w into s via M and w into s' via M' respectively. Consider

$$\text{Seq } T\colon q_0, \dots, q_i, \dots, q_n$$

$$\text{Seq } T'\colon q_0', \dots, q_i', \dots, q_n'$$

where n is the length of w. Since $n \geqslant 2kk'$, the number of terms in each sequence is $n + 1 \geqslant 2kk' + 1$. Regard, for a moment, the two sequences as a single sequence of (vertically) ordered pairs (q_i, q_i'). Since the number of distinct ordere pairs in $K \times K'$ is kk' and since there are at least $2kk' + 1$ pairs in the sequence we conclude that repetitions of ordered pairs occur as follows: There are subscripts $i(1)$, $i(2)$, $i(3)$, $i(4)$ for which $0 \leqslant i(1) < i(2) \leqslant kk' \leqslant i(3) < i(4) \leqslant n$ and $q_{i(1)} = q_{i(2)}$, $q_{i(1)}' = q_{i(2)}'$, $q_{i(3)} = q_{i(4)}$, $q_{i(3)}' = q_{i(4)}'$. We use this choice of states to define factorizations $w = w_1w_2w_3w_4w_5$, $s = s_1s_2s_3s_4s_5$, and $s' = s_1's_2's_3's_4's_5'$ as follows: w_1 is that portion of w that corresponds to the change of state from q_0 to $q_{i(1)}$ via T and from q_0' to $q_{i(1)}'$ via T' and s_1 and s_1' are the associated output strings; w_2 is that portion of w that corresponds to the changes $q_{i(1)}$ to $q_{i(2)}$ via T, $q_{i(1)}'$ to $q_{i(2)}'$ via T' with s_2, s_2' the output strings;...;...; w_5 is that portion of w that corresponds to the changes $q_{i(4)}$ to q_n via T, $q_{i(4)}'$ to q_n' via T' with s_5, s_5' the output strings.

Since $i(1) < i(2)$ and $i(3) < i(4)$, $w_2 \neq e$ and $w_4 \neq e$. Thus the strings $w_1w_3w_5$, $w_1w_2w_3w_5$, and $w_1w_3w_4w_5$ are strictly shorter than w. Moreover:

$$M(w_1w_3w_5) = s_1s_3s_5R, \qquad M'(w_1w_3w_5) = s_1's_3's_5'R',$$

$$M(w_1w_2w_3w_5) = s_1s_2s_3s_5R, \qquad M'(w_1w_2w_3w_5) = s_1's_2's_3's_5'R',$$

$$M(w_1w_3w_4w_5) = s_1s_3s_4s_5R, \quad \text{and} \quad M'(w_1w_3w_4w_5) = s_1's_3's_4's_5'R'.$$

Since $w_1w_3w_5$, $w_1w_2w_3w_5$, and $w_1w_3w_4w_5$ are shorter than w:

$$s_1s_3s_5R = s_1's_3's_5'R'$$
$$s_1s_2s_3s_5R = s_1's_2's_3's_5'R',$$
$$s_1s_3s_4s_5R = s_1's_3's_4's_5'R'.$$

Since $s_1s_3s_5$ and $s_1's_3's_5'$ are the shortest strings in $s_1s_3s_5R$ and $s_1's_3's_5'R'$ respectively, it follows that $s_1s_3s_5 = s_1's_3's_5'$ and then that $R = R'$. Similarly, $s_1s_2s_3s_5 = s_1's_2's_3's_5'$ and $s_1s_3s_4s_5 = s_1's_3's_4's_5'$. Theorem 3.1 now applies and yields the conclusion that $s_1s_2s_3s_4s_5 = s_1's_2's_3's_4's_5'$, i.e., $s = s'$. Since $R = R'$ we have arrived at the contradiction $M(w) = sR = s'R' = M'(w)$. We conclude that the theorem is true.

Theorem 2 assures the validity of the following:

Algorithm. To decide whether two deterministic finite state transducers M and M' are equivalent, (1) decide whether they have the same domain, and (2) if they have the same domain, decide for each string w in their common domain having length at most $2kk' - 1$ (where k, k' are the number of states in M, M' respectively) whether the regular sets $M(w)$ and $M'(w)$ are equal. Then M and M' are equivalent if and only if they have the same domain and $M(w) = M'(w)$ for the finite number of strings of length less than $2kk' - 1$.

Note added in proof. We thank J. Berstel for communicating to us the following short proof of Thm. 3.1: Substitute $s' = s_1x$ into (3.1) and cancel to produce $s_3s_5 = xs_3's_5'$. Replace the latter in (3.2) and cancel to produce $s_1s_2x = s_1's_2'$. Substitute $s' = s_1x$ into (3.3) and cancel to produce $s_3s_4s_5 = xs_3's_4's_5'$. Then $s_1's_2's_3's_4's_5' = s_1s_2xs_3's_4's_5' = s_1s_2s_3s_4s_5$.

References

1. A. Aho and J. Ullman, "The Theory of Parsing, Translation and Compiling," Vol. I, Prentice-Hall, Englewood Cliffs, N. J., 1972.
2. M. Blattner and T. Head, Single-valued a-Transducers, *J. Comput. System Sci.* **15** (1977), 310–327.
3. S. Ginsburg, "Formal Languages," North-Holland/American Elsevier, New York, 1975.
4. T. V. Griffiths, The unsolvability of the equivalence problem for Λ-free nondeterministic generalized machines, *J. Assoc. Comput. Mach.* **15** (1968), 409–413.