Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®
Theoretical Computer Science 310 (2004) 147–158
Theoretical Computer Science
www.elsevier.com/locate/tcs

The limitedness problem on distance automata: Hashiguchi's method revisited☆

Hing Leung*, Viktor Podolskiy

Department of Computer Science, New Mexico State University, Box 30001, Dept. MSC CS, Las Cruces, NM 88003, USA

Received 28 May 2002; received in revised form 26 November 2002; accepted 2 July 2003
Communicated by D. Perrin

Abstract

Hashiguchi has studied the limitedness problem of distance automata (DA) in a series of paper [(J. Comput. System Sci. 24 (1982) 233; Theoret. Comput. Sci. 72 (1990) 27; Theoret. Comput. Sci. 233 (2000) 19)]. The distance of a DA can be limited or unbounded. Given that the distance of a DA is limited, Hashiguchi has proved in Hashiguchi (2000) that the distance of the automaton is bounded by $2^{4n^3+n\lg(n+2)+n}$, where n is the number of states. In this paper, we study again Hashiguchi's solution to the limitedness problem. We have made a number of simplification and improvement on Hashiguchi's method. We are able to improve the upper bound to $2^{3n^3+n\lg n+n-1}$.

1. Introduction

A distance automaton (DA) is a nondeterministic finite automaton (NFA) extended with a distance 0 or 1 on each transition. A string in the language of an NFA may be accepted by many different accepting paths. We define the distance of a path to be the sum of the distances on the transitions in the path. The distance of an accepted string is the minimum of the distances over the different accepting paths. The distance of a DA is the supremum over all distances of the accepted strings. The limitedness problem is to determine if a given DA is limited or unbounded in distance.

☆ A preliminary version of this paper was presented in the 3rd workshop on Descriptional Complexity of Automata, Grammars and Related Structures, Vienna, Austria, July 20–22, 2001.

* Corresponding author.
E-mail address: hleung@cs.nmsu.edu (H. Leung).

0304-3975/$ - see front matter
doi:10.1016/S0304-3975(03)00377-3

Hashiguchi studied the star height problem and the representation problems for regular languages in [2,4,5]. His solutions relied heavily on the decidability result of the limitedness problem [3,6] for distance automata. Leung [8,9,10] solved the limitedness problem for distance automata by using semigroup theory. Using factorization forests, Simon [13,14] also solved the limitedness problem. A survey on these results is given by Simon in [12].

It has been shown [9] that the limitedness problem for distance automata is PSPACE-hard. Hashiguchi has proved in [7] that the distance of the automaton, if limited, is bounded by $2^{4n^3+n\lg(n+2)+n}$, where n is the number of states. Based on the single exponential upper bound result, we design a nondeterministic polynomial space algorithm [1] to decide the limitedness problem for distance automata. Next, by Savitch's theorem (Theorem 7.12 of [1]) we can solve the limitedness problem in deterministic polynomial space.

However, we have found that Hashiguchi's work is difficult to understand. In this paper, we study again Hashiguchi's solution to the limitedness problem. We have made a number of simplification and improvement on Hashiguchi's method. We are able to improve the upper bound to $2^{3n^3+n\lg n+n-1}$ (Theorem 3).

The aim of this paper is to present a clean and simple proof of the result. Its aim is not to compute the lowest possible bound even though our bound is lower than the one obtained by Hashiguchi.

2. Limitedness problem on distance automata

Let $\mathbb{N}$ denote the set of nonnegative integer numbers. A distance automaton is a 5-tuple (Q, Σ, d, Q_I, Q_F) where Q is the set of states, Σ is the alphabet set, $d : Q\times\Sigma\times Q\to\{0, 1, \infty\}$ is the distance function, $Q_I \subseteq Q$ is the set of starting states and $Q_F \subseteq Q$ is the set of final states. Let $n = |Q|$ be the number of states.

Given a path $(q_1, a_1, q_2, a_2, \ldots, a_k, q_{k+1})$, the distance of the path is defined as $d(q_1, a_1, q_2) + d(q_2, a_2, q_3) + \cdots + d(q_k, a_k, q_{k+1})$. Consider the processing of a string x from state p to state q. Since a distance automaton is nondeterministic, there could be many different paths. We define the distance to be the minimum distance over the different paths.

We extend the distance function d to a function $Q \times \Sigma^* \times Q \to \mathbb{N} \cup \{\infty\}$ such that $d(p, x, q)$ denotes the distance used by the automaton for going from state p to state q consuming $x \in \Sigma^*$.

The distance of an accepted string x is defined as $d(x) = \min_{p\in Q_I,\, q\in Q_F} d(p, x, q)$. The distance of a DA M is defined to be $\sup_{x\in L(M)} d(x)$.

[1] The algorithm guesses an accepted string w that requires a distance more than $2^{4n^3+n\lg(n+2)+n}$. The algorithm does not hold the complete string w in its memory. The string w is guessed and processed one symbol at a time. At any moment, the program maintains a set of states reached together with the associated distances. To represent a distance that does not exceed $2^{4n^3+n\lg(n+2)+n}$, we need at most $4n^3 + n\lg(n+2) + n$ bits. If the distance exceeds $2^{4n^3+n\lg(n+2)+n}$, we need only to remember it by a special value (that takes a constant amount of space). The total space needed is $n * \mathrm{O}(4n^3) = \mathrm{O}(n^4)$, which is polynomial.

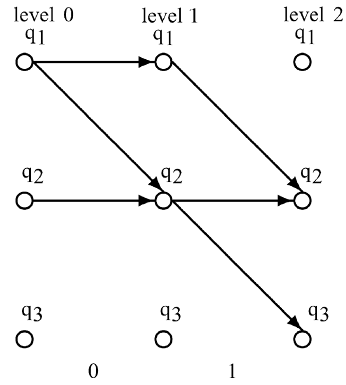

Fig. 1. Highway structure for the string 01.

2.1. Highway and decomposition

Let $Q=\{q_1,q_2,\dots,q_n\}$. Let $x=a_1a_2\dots a_k\in\Sigma^+$ be a string where $a_1,a_2,\dots,a_k\in\Sigma$. We define the 0-distance graph of x to be a directed graph (V,E) where $V=\{q_{i,j}\mid 0\leqslant i\leqslant k, 1\leqslant j\leqslant n\}$ and $E=\{(q_{i-1,j},q_{i,h})\mid d(q_j,a_i,q_h)=0,\ 1\leqslant i\leqslant k\}$. The 0-distance graph is also called the highway associated with x. The highway has $k+1$ levels of vertices. The ith level consists of vertices $q_{i,j}$ where $1\leqslant j\leqslant n$. Each edge going from the $(i-1)$th level to the ith level corresponds to a 0-distance transition that consumes a_i. A path of the highway of length k is called a lane of the highway. We say that two lanes are disjoint if the two paths of length k have disjoint vertices. The width of a highway is the maximum number of disjoint lanes in the highway.

Example 1. Consider a distance automaton with three states $\{q_1,q_2,q_3\}$ over a two-letter alphabet $\{0,1\}$. Suppose the 0-distance transitions consist of moves $\{(q_1,0,q_1)$, $(q_1,0,q_2)$, $(q_2,0,q_2)$, $(q_1,1,q_2)$, $(q_2,1,q_2)$, $(q_2,1,q_3)\}$. That is, $d(q_1,0,q_1)=d(q_1,0,q_2)=d(q_2,0,q_2)=d(q_1,1,q_2)=d(q_2,1,q_2)=d(q_2,1,q_3)=0$. The rest of the transitions are assumed to have distances other than 0. Let $x=01$. We can find three lanes in the highway associated with x. The three lanes are respectively $l_1=(q_1,0,q_1,1,q_2)$, $l_2=(q_1,0,q_2,1,q_2)$ and $l_3=(q_2,0,q_2,1,q_3)$. See Fig. 1. However, the three lanes are not disjoint. The second lane l_2 intersects with the first lane l_1 on the starting state. l_2 also intersects with l_3 on the second state. On the other hand, l_1 and l_3 are disjoint. Thus, the maximum number of disjoint lanes is 2, which is defined to be the width of the highway.

We say that a non-empty string x is primitive if either it has only one symbol or the string x with its last symbol removed has a larger highway width than the width of the highway associated with x.

Example 2. We continue with the previous example. The string 010 has a highway width of one, whereas the string 01 has a highway width of two. The string 11 has

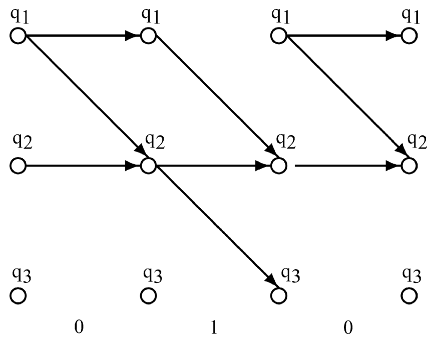

Fig. 2. Highway structure for the string 010.

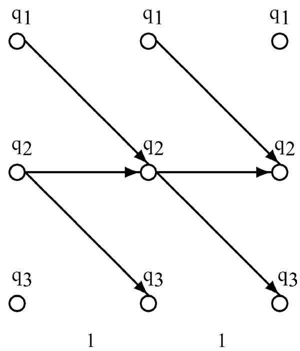

Fig. 3. Highway structure for the string 11.

a highway width of one, whereas the string 1 has a highway width of two. Thus, the two strings 010 and 11 are both primitive. See Figs. 2 and 3. Moreover, the strings 0 and 1 are both primitive since they are of length one.

We decompose a given string x according to highway width. A string x is decomposed into a tuple of non-empty strings $(x_1, x_2, \ldots, x_m)$ such that $x = x_1 x_2 \ldots x_m$ where $x_1, x_2, \ldots, x_{m-1}$ are primitive strings of the same highway width as that of x and for the string x_m, either it is also a primitive string of the same highway width as that of x or it has a larger highway width than that of x. We call each string x_i a component in the decomposition of x.

Example 3. Again we continue with the previous examples. The string $x = 0101100$ has highway width one. It is decomposed into $(010, 11, 00)$ where 010 and 11 are primitive strings of width one, 00 is a string of width two.

2.2. Relatives of a string

Let x be a string. The relatives of x is a family of strings $x(k) \in \Sigma^*$, $k \geqslant 1$, such that $x(1) = x$. In addition, the relatives of x have to satisfy further conditions that we describe below. We call $x(k)$ the kth relative of x. Note that it is possible to define $x(k) = x$ for $k \geqslant 1$.

We say that (p,q) is a free pair of states for the relatives of x when $d(p,x(k),q) = 0$ for $k \geqslant 1$. We say that (p,q) is an unreachable pair of states for the relatives of x when $d(p,x(k),q) = \infty$ for $k \geqslant 1$. We say that (p,q) is a bounded pair of states for the relatives of x when there exists $b \in \mathbb{N}$ such that $1 \leqslant d(p,x(k),q) < b$ for $k \geqslant 1$. We say that (p,q) is an unbounded pair of states for the relatives of x when $0 \neq d(p,x(k),q) \in \mathbb{N}$ for $k \geqslant 1$ and $\lim_{k\to\infty} d(p,x(k),q) = \omega$.

We say that (p,q) is a good pair of states if (p,q) is either a free pair or a bounded pair. We say that (p,q) is a bad pair of states if (p,q) is either an unreachable pair or an unbounded pair.

Definition 1. The relatives of a string x is a family of strings $x(k) \in \Sigma^*$, $k \geqslant 1$, such that $x(1) = x$ and for each pair of states (p,q), either it is a free pair, a bounded pair, an unbounded pair or an unreachable pair.

We say that the limited distance behaviors of the relatives of x are bounded by B if for all bounded pair of states (p,q) and for all $k \geqslant 1$, we have $d(p,x(k),q) \leqslant B$. We say that the limited distance behaviors of the relatives of x are bounded above by B if for all bounded pair of states (p,q) and for all $k \geqslant 1$, we have $d(p,x(k),q) < B$.

Given the relatives of x and the relatives of y, we say that the two families of relatives have the same 'structure' if the two relative families have the same set of free state pairs, the same set of bounded state pairs, the same set of unbounded state pairs and the same set of unreachable state pairs. Since there are n^2 possible state pairs, there could be at most $4^{n^2} = 2^{2n^2}$ possible different structures of relative families.

Given the relatives $x(k)$ of x and the relatives $y(k)$ of y, suppose we define the relatives of xy such that $xy(k) = x(k)y(k)$ for $k \geqslant 1$. That is, the kth relative of xy is the concatenation of the kth relative of x and the kth relative of y. Note that $xy(1) = x(1)y(1) = xy$. One can verify that the relatives of xy defined in this way satisfy all requirements of relatives given above.

2.3. Main lemma

Let $f(n) = n$, $f(t) = (2n-1)2^{4n^2} f(t+1)$ for $0 < t < n$ and $f(0) = 2^{4n^2} f(1)$.

Lemma 1. *Given any string x of highway width h, there exist relatives of x, $x(k)$ for $k \geqslant 1$, such that the limited distance behaviors of the relatives of x are bounded above by $f(h)$.*

As an immediate consequence of the main lemma, the distance of an automaton, if limited, is bounded by $f(0)$ which is $n(2n-1)^{n-1}(2^{4n^2})^n < n(2n)^{n-1}(2^{4n^2})^n = 2^{4n^3 + n\lg n + n - 1}$.

Theorem 2. *The distance of an automaton, if limited, is bounded by* $2^{4n^3+n\lg n+n-1}$.

2.4. *Proof of main lemma*

We prove the main lemma by an induction on the highway width.

2.4.1. *Base case (highway width is n)*

Consider the base case when the highway width is n. We define the relatives of x, $x(k)$, to be x for all $k \geqslant 1$. Since the highway is of width n, every state is part of the highway system at any stage of the processing for x. Given a state pair (p, q), either we cannot reach q from p consuming x or q can be reached from p by switching lanes for at most $n-1$ times. Thus, the distance $d(p, x, q)$ is either ∞ or less than n.

2.4.2. *General case (highway width is between 0 and n)*

We consider the general case when the highway width t of x is between 0 and n. Let $(x_1, x_2, \ldots, x_m)$ be the decomposition of x, where m is the number of components in the decomposition of x.

Case 1: $m = 1$.

x must be primitive; otherwise, it has more than one component in its decomposition. Let $x = ya$ where $y \in \Sigma^*$ and $a \in \Sigma$ is a single symbol. We define the relatives $x(k) = y(k)\ a$ for $k \geqslant 1$. Note that y may be an empty string ε. In that case, $y(k) = \varepsilon$ for $k \geqslant 1$. Let us first consider the general case when $y \neq \varepsilon$. Since by the induction hypothesis the limited distance behaviors of the relatives of y are bounded *above* by $f(h)$ where h is the highway width of y and $h \geqslant t+1$, the limited distance behaviors of the relatives of x are bounded by $f(h) \leqslant f(t+1)$. It is easy to see that the limited distance behaviors of the relatives of x are bounded above by $f(t)$ since $f(t) > f(t+1)$. For the special case when $y = \varepsilon$, the limited distance behaviors of the relatives of x, where $x(k) = a$ for all $k \geqslant 1$, are bounded by the distance one, which is easily seen to be bounded above by $f(t)$. Since $y(1) = y$, we have $x(1) = y(1)\ a = ya = x$.

Before we continue with cases 2 and 3 where $m > 1$, we define the relatives of a component x_i in the same way as discussed in case 1 if the component x_i is primitive and has highway width t. If x_m has a larger highway width than t, then by the induction hypothesis the relatives of x_m have already been defined. In both cases, the limited distance behaviors of a component x_i are bounded by $f(t+1)$.

Case 2: $1 < m < 2^{4n^2}$.

We define the relatives of x, $x(k)$, to be $x_1(k)x_2(k)\ldots x_m(k)$. It is immediate that $x(1) = x$ by the assumption that $x_i(1) = x_i$ for $1 \leqslant i \leqslant m$. The distance behaviors of the relatives of x are computed by the composition of the distance behaviors of the relatives of its components. Thus, the limited distance behaviors of the relatives of x are bounded above by $2^{4n^2} f(t+1)$ since the limited distance behaviors of the relatives of x_i are bounded by $f(t+1)$.

Case 3: $m \geqslant 2^{4n^2}$.

We need to define the relatives of x, which involve 'pumping'. We assume that there are some bounded pairs of states for the relatives of x. Otherwise, it is trivially true that the limited distance behaviors of the relatives of x are bounded above by $f(t)$.

Let c denote the number 2^{4n^2}. We divide the m components of x into groups of c components each. That is, the first group has components $x_1, x_2, \ldots, x_c$. The second group has components $x_{c+1}, x_{c+2}, \ldots, x_{2c}$. The third group has components x_{2c+1}, $x_{2c+2}, \ldots, x_{3c}$. If c does not divide m, the last group would have less than c components.

We need to introduce some notations. We write $x_{i,j}$ to denote the string $x_i x_{i+1} x_{i+2} \ldots x_j$. We write $x_{i,j}(k)$ to denote the kth relative of the string $x_{i,j}$ which is defined as $x_i(k)\ x_{i+1}(k)\ x_{i+2}(k) \ldots x_j(k)$ for $k \geqslant 1$.

Consider the $(r+1)$th group of c components $(x_{rc+1}, x_{rc+2}, \ldots, x_{rc+c})$. Let $1 < i \leqslant c$. Consider the ith partition of the group into $x_{rc+1} x_{rc+2} \ldots x_{rc+i-1}$ and $x_{rc+i} x_{rc+i+1} \ldots x_{rc+c}$. We define the structure of the ith partition as the ordered pair (`structure of` $x_{rc+1,rc+i-1}(k)$, `structure of` $x_{rc+i,rc+c}(k)$). We want to argue that there are two different partitions of the $(r+1)$th group with the same structure.

Since it is assumed that some bounded pairs must occur, the number of possible structures of relative families is at most $2^{2n^2} - 1$. Thus, the number of possible structures of a partition is at most $(2^{2n^2} - 1)(2^{2n^2} - 1)$. On the other hand, there are $c - 1$ different partitions where $c - 1 = 2^{4n^2} - 1$ which is larger than $(2^{2n^2} - 1)(2^{2n^2} - 1)$. Thus, by the pigeonhole principle, there are two different partitions of the $(r+1)$th group with the same structure.

Suppose the structure of the ith partition is the same as the structure of the jth partition, where $1 < i < j \leqslant c$. We introduce pumping to create a family of relatives associated with the $(r+1)$th group of c components. We define the relatives for the $(r+1)$th group as $g_{r+1}(k) = x_{rc+1,rc+i-1}(k)(x_{rc+i,rc+j-1}(k))^k x_{rc+j,rc+c}(k)$. Note that the kth relative of $x_{rc+i,rc+j-1}$ is pumped by exactly k times.

We also denote $x_{rc+1,rc+i-1}(k)$ by $\alpha_{r+1}(k)$, $x_{rc+i,rc+j-1}(k)$ by $\beta_{r+1}(k)$ and $x_{rc+j,rc+c}(k)$ by $\gamma_{r+1}(k)$. Thus, $g_{r+1}(k) = \alpha_{r+1}(k)(\beta_{r+1}(k))^k \gamma_{r+1}(k)$.

Suppose there are d groups of c components, where $d = \lfloor m/c \rfloor$. Let the last group be the $(d+1)$th group which does not have c components. Note that the last group is empty if c divides m.

The relatives for the $(d+1)$th group is defined as $g_{d+1}(k) = x_{dc+1}(k) x_{dc+2}(k) \ldots x_m(k)$. That is, there is no pumping involved. If c divides m, then $g_d(k) = \varepsilon$.

We define the relatives of x, $x(k)$, to be $g_1(k) g_2(k) \ldots g_d(k) g_{d+1}(k)$. By the inductive assumption that $x_i(1) = x_i$ for $1 \leqslant i \leqslant m$, it is easy to verify that $x(1) = x$.

By the way the pumping construction is defined, the structure of $\gamma_{r+1}(k)$ has the same set of free and unreachable state pairs as the structure of $\beta_{r+1}(k)\gamma_{r+1}(k)$. By the composition of the structures, we deduce that the structure of $(\beta_{r+1}(k))^2 \gamma_{r+1}(k)$ has the same set of free and unreachable state pairs as the structure of $\beta_{r+1}(k)\gamma_{r+1}(k)$. Similarly, the structure of $(\beta_{r+1}(k))^k \gamma_{r+1}(k)$ also has the same set of free and unreachable state pairs as the structure of $\beta_{r+1}(k)\gamma_{r+1}(k)$. Thus, the structure of $g_{r+1}(k) = \alpha_{r+1}(k)(\beta_{r+1}(k))^k$ $\gamma_{r+1}(k)$ has the same set of free and unreachable state pairs as the structure of $\alpha_{r+1}(k)$ $\beta_{r+1}(k)\gamma_{r+1}(k) = x_{rc+1,rc+c}(k)$. Therefore, both structures for $x(k)$ and $x_1(k) x_2(k) \ldots$ $x_m(k)$ have the same set of free and unreachable state pairs.

In order to show that $x(k)$, $k \geqslant 1$, satisfy the definition of a relative family, it remains to show that if a state pair (p,q) is not free or unreachable, then it is either bounded or unbounded.

Suppose (p,q) is not an unbounded pair of states for the relatives of x. Also, suppose that (p,q) is not a free or unreachable pair of states. We want to show that (p,q) is a bounded pair of states for the relatives of x.

Recall that $x(k)=\alpha_1(k)(\beta_1(k))^k\gamma_1(k)\ldots\alpha_d(k)(\beta_d(k))^k\gamma_d(k)g_{d+1}(k)$. Since (p,q) is not an unbounded pair, $\lim_{k\to\infty} d(p,x(k),q)\neq\omega$. That is, $\exists B$, $\forall k$, $\exists K>k$, $d(p,x(K),q) <B$. Let $k=nB$. We choose a large enough value K such that $K>nB$ and moreover for all $1\leqslant i\leqslant d$, $\alpha_i(K)(r,s)>B$ for any unbounded pair (r,s) for the relatives $\alpha_i(k)$, $\beta_i(K)(r,s)>B$ for any unbounded pair (r,s) for the relatives $\beta_i(k)$, $\gamma_i(K)(r,s)>B$ for any unbounded pair (r,s) for the relatives $\gamma_i(k)$, and $g_{d+1}(K)(r,s)>B$ for any unbounded pair (r,s) for the relatives $g_{d+1}(k)$. Since $d(p,x(K),q)<B$, there is a path P from p to q of distance less than B consuming $x(K)$. Let $P=(p,\alpha_1(K),p_1,(\beta_1(K))^K, q_1, \gamma_1(K),s_2,\alpha_2(K),p_2,(\beta_2(K))^K,q_2,\gamma_2(K),\ldots,s_d,\alpha_d(K),p_d,(\beta_d(K))^K,q_d,\gamma_d(K),s_{d+1}, g_{d+1}(K),q)$ that visit states $p_1,q_1,s_2,p_2,q_2,\ldots,s_d,p_d,q_d,s_{d+1}$ such that (s_i,p_i) is a good pair for the relatives $\alpha_i(k)$, $d(p_i,(\beta_i(K))^K,q_i)<B$, (q_i,s_{i+1}) is a good pair for the relatives $\gamma_i(k)$, where $1\leqslant i\leqslant d$ and $s_1=p$, and (s_{d+1},q) is a good pair for the relatives $g_{d+1}(k)$. Thus, there is a path of cost less than B going from p_i to q_i consuming $(\beta_i(K))^K$. Let the path be $(z_{i_0},\beta_i(K),z_{i_1},\beta_i(K),z_{i_2},\ldots,\beta_i(K),z_{i_K})$ where $z_{i_0}=p_i$, $z_{i_K}=q_i$ and $z_{i_1},\ldots,z_{i_{K-1}}$ are the intermediate states reached after each copy of $\beta_i(K)$ is processed. Since the distance $d(p_i,(\beta_i(K))^K,q_i)$ is less than B and K is chosen such that $K>nB$, by the pigeonhole principle there exists two indices $0\leqslant e<f\leqslant K$ such that $z_{i_e}=z_{i_f}$ and $d(z_{i_e},(\beta_i(K))^{f-e},z_{i_f})=0$. Let $z_{i_e}=z_{i_f}$ be denoted by z_i.

Consider the processing of the kth relative of x from state p to state q, where k is *any* positive integer. We want to argue that there is a path $(p,\alpha_1(k),u_1,(\beta_1(k))^k,v_1,\gamma_1(k), s_2,\alpha_2(k),u_2,(\beta_2(k))^k,v_2,\gamma_2(k),\ldots,s_d,\alpha_d(k),u_d,(\beta_d(k))^k,v_d,\gamma_d(k),s_{d+1},g_{d+1}(k),q)$ from p to q consuming $x(k)$ that visit states $u_1,v_1,s_2,u_2,v_2,\ldots,s_d,u_d,v_d,s_{d+1}$ such that (s_i,u_i) is a good pair for the relatives $\alpha_i(k)$, $d(u_i,(\beta_i(k))^k,v_i)=0$, (v_i,s_{i+1}) is a good pair for the relatives $\gamma_i(k)$, where $1\leqslant i\leqslant d$ and $s_1=p$ and (s_{d+1},q) is a good pair for the relatives $g_{d+1}(k)$.

We need to explain how to select the values for u_i and v_i where $1\leqslant i\leqslant d$ for the processing of any kth relative of x. Note that s_i's are selected as in the processing of $x(K)$. We select u_i to be z_i as explained above for the processing of $\beta_i(K)^K$. By the condition for pumping, we deduce that (s_i,u_i) is a good pair of $\alpha_i(k)$. To process k copies of $\beta_i(k)$, the path is $(z_{i_e},\beta_i(k),z_{i_{e+1}},\beta_i(k),z_{i_{e+2}},\beta_i(k),\ldots,\beta_i(k),z_{i_f},\beta_i(k),z_{i_{e+1}},\beta_i(k),z_{i_{e+2}}, \ldots)$. The last state reached after processing k copies of $\beta_i(k)$ is defined as the state v_i. Again by the condition for pumping, we deduce that (v_i,s_{i+1}) is a good pair of $\gamma_i(k)$.

We claim that the 0-distance path from u_i to v_i consuming $(\beta_i(k))^k$ intersects with one of t 0-distance paths for $x(k)$, where t is the highway width for x. Note that we are not claiming that the highway width for $x(k)$ is t. The t 0-distance paths for $x(k)$ may not be disjoint. To prove the claim, we need to construct t 0-distance paths for $x(k)$ based on t disjoint highway lanes for x.

We first consider the consequence of the claim, and defer proving the claim until later. Suppose the 0-distance path from u_i to v_i and the 0-distance path from u_j to

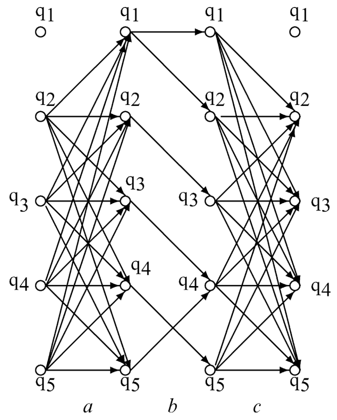

Fig. 4. Highway structure for the string *abc*.

v_j both intersect with the same 0-distance path for $x(k)$, where $i<j$. It is immediate that $d(u_i,(\beta_i(k))^k\gamma_i(k)g_{i+1}(k)g_{i+2}(k)\dots g_{j-1}(k)\alpha_j(k)(\beta_j(k))^k,v_j)=0$. Since $t<n$, there are less than n good pairs of (s_i,u_i) that require non-zero distance to process $\alpha_i(k)$. Similarly, there are less than n good pairs of (v_i,s_{i+1}) that require non-zero distance to process $\gamma_i(k)$. Finally, the good pair (s_{d+1},q) may require non-zero distance to process $g_{d+1}(k)$. Since $\alpha_i(k)$, $\gamma_i(k)$ and $g_{d+1}(k)$ all have less than 2^{4n^2} components, the total distance required to process $x(k)$ is less than $(2(n-1)+1)2^{4n^2}f(t+1)=(2n-1)2^{4n^2}f(t+1)$. Note that k is an arbitrary positive integer. We have proved that, for all $k\geqslant 1$, the distance of $x(k)$ is bounded above by $(2n-1)2^{4n^2}f(t+1)$.

Before we prove the claim, we demonstrate by an example that the highway width of $x(k)$ may be smaller than that of x when $k>1$.

Example 4. Consider a distance automaton with 5 states $\{q_1,q_2,q_3,q_4,q_5\}$ over a three-letter alphabet $\{a,b,c\}$. Suppose the 0-distance transitions consist of moves $\{(q_i,a,q_j)\mid 2\leqslant i\leqslant 5, 1\leqslant j\leqslant 5\}\cup\{(q_1,b,q_1),(q_1,b,q_2),(q_2,b,q_3),(q_3,b,q_4),(q_4,b,q_5),(q_5,b,q_4)\}\cup\{(q_i,c,q_j)\mid 1\leqslant i\leqslant 5, 2\leqslant j\leqslant 5\}$. Assume that all other moves are unreachable. That is, all moves are either of 0-distance or unreachable. Fig. 4 illustrates the highway structure for the string $x=abc$. The string x is decomposed into primitive strings a, b and c, respectively, since each symbol has highway width 4, which is also the highway width of x. We observe that `(structure of` a`, structure of` bc`)` = `(structure of` ab`, structure of` c`)`. Let $x(k)=ab^kc$. One can verify that the highway width of $x(k)$ is 3 for $k>1$.

It remains to prove the claim.

Let us name the 0-distance path from u_i to v_i consuming $(\beta_i(k))^k$ as H. We switch our attention back to $x=x(1)$ instead of $x(k)$. Since $(z_{i_e},\beta_i(k),z_{i_{e+1}})$ is a 0-distance path, there is also a 0-distance path $(z_{i_e},\beta_i,z_{i_{e+1}})$, where $\beta_i=\beta_i(1)$. Next, there is a 0-distance path H_1 for x that intersects with the 0-distance path $(z_{i_e},\beta_i,z_{i_{e+1}})$. Otherwise, the highway width for β_i will be larger than that for x, a contradiction to the way that x is decomposed.

Recall that there is a 0-distance path from z_{i_e} to z_{i_f} consuming $f-e$ copies of $\beta_i(k)$, where $z_{i_e}=z_{i_f}$. Let k' be an integer such that $k'=\eta(f-e)+1>k$ for some integer η. Thus, there is a 0-distance path from z_{i_e} to $z_{i_{e+1}}$ consuming k' copies of $\beta_i(k)$. Similarly, there is a 0-distance path (called H_2) from z_{i_e} to $z_{i_{e+1}}$ consuming k' copies of β_i.

Consider the string $y_i^{k'}=\alpha_1\beta_1\gamma_1\alpha_2\beta_2\gamma_2\ldots\alpha_i{\beta_i}^{k'}\gamma_i\ldots\alpha_d\beta_d\gamma_d g_{d+1}$. That is, $y_i^{k'}$ is similar to x except that β_i is pumped for k' times. All other β_j are not pumped for $j\neq i$. We combine H_1 and H_2 to construct a 0-distance path H_3 for $y_i^{k'}$. The construction is as follows. First, we use H_1 until it intersects H_2 in the traversal of the first copy of β_i. We continue using H_2 until the intersection with H_1 again in the traversal of the last copy of β_i. Then the 0-distance path switches back to using H_1. The construction is possible since both the first and last copy (which is the $(\eta(f-e)+1)$th copy) of β_i are traversed by H_2 using the same 0-distance path (z_e,β_i,z_{e+1}).

Next, consider the string $y_i^k=\alpha_1\beta_1\gamma_1\alpha_2\beta_2\gamma_2\ldots\alpha_i{\beta_i}^k\gamma_i\ldots\alpha_d\beta_d\gamma_d g_{d+1}$. That is, y_i^k is similar to x except that β_i is pumped for k times instead of k' times for $y_i^{k'}$. We want to modify H_3 for $y_i^{k'}$ to form a 0-distance path H_4 for y_i^k. Note that $k'>k$. Let $y_1=\alpha_1\beta_1\gamma_1\alpha_2\beta_2\gamma_2\ldots\alpha_{i-1}\beta_{i-1}\gamma_{i-1}$ and $y_3=\alpha_{i+1}\beta_{i+1}\gamma_{i+1}\ldots\alpha_d\beta_d\gamma_d g_{d+1}$. Let the 0-distance path H_3 for $y_i^{k'}$ be denoted as $(h,y_1,h',\alpha_i,h'',\beta_i,h_1,\beta_i,h_2,\ldots,\beta_i,h_{k'},\gamma_i,h''',y_3,h'''')$. Note that $h_1=z_{e+1},h_2=z_{e+2},\ldots$ etc. Also note that h'' is not necessarily z_e. By the definition that the structure of $\beta_i(k)\gamma_i(k)$ is the same as the structure of $\gamma_i(k)$, the set of free state pairs are the same for $\beta_i(k)\gamma_i(k)$ and $\gamma_i(k)$. Hence, the set of free state pairs are the same for $\beta_i\gamma_i$ and γ_i. Thus, we can deduce a 0-distance path H_4 for y_i^k as $(h,y_1,h',\alpha_i,h'',\beta_i,h_1,\beta_i,h_2,\ldots,\beta_i,h_k,\gamma_i,h''',y_3,h'''')$.

Let $y_i(k)=\alpha_1\beta_1\gamma_1\alpha_2\beta_2\gamma_2\ldots\alpha_i(k)\beta_i(k)^k\gamma_i(k)\ldots\alpha_d\beta_d\gamma_d g_{d+1}$. That is, $y_i(k)$ is similar to y_i^k except that α_i, β_i and γ_i are substituted by $\alpha_i(k)$, $\beta_i(k)$ and $\gamma_i(k)$ respectively. We can deduce a 0-distance path H_5 for $y_i(k)$ as $(h,y_1,h',\alpha_i(k),h'',\beta_i(k),h_1,\beta_i(k),h_2,\ldots,\beta_i(k),h_k,\gamma_i(k),h''',y_3,h'''')$. This is because α_i and $\alpha_i(k)$ (respectively, β_i and $\beta_i(k)$, γ_i and $\gamma_i(k)$) have the same set of free pairs of states. Observe that H intersects with the 0-distance sub-path $(h'',\beta_i(k),h_1,\beta_i(k),h_2,\ldots,\beta_i(k),h_k)$ of H_5.

Note that the 0-distance path H_5 for $y_i(k)$ is very similar to the 0-distance path H_1 for x. The two paths are the same for processing g_{d+1} and $\alpha_j\beta_j\gamma_j$ where $j\neq i$.

Let us summarize the construction so far. From the 0-distance path from u_i to v_i consuming $(\beta_i(k))^k$, we consider a 0-distance path H_1 for x from which we derive a 0-distance path for $y_i(k)$. We call the 0-distance path H_1 the base of the 0-distance path for $y_i(k)$.

If we consider the 0-distance path from u_j to v_j consuming $(\beta_j(k))^k$, it is possible that it may give rise to the same base H_1 for the 0-distance path for $y_j(k)$. Of course, it is also possible that the base may not be H_1.

There are t possible 0-distance path that we may consider as base 0-distance paths since the highway width for x is t.

With respect to H_1, we want to explain how to derive a 0-distance path for $x(k)=\alpha_1(k)\beta_1(k)^k\gamma_1(k)\alpha_2(k)\beta_2(k)^k\gamma_2(k)\ldots\alpha_d(k)\beta_d(k)^k\gamma_d(k)g_{d+1}(k)$.

Let $\iota \in \{1, \ldots, d\}$. If H_1 is the base of the 0-distance path for $y_\iota(k)$, we define the 0-distance sub-path of $x(k)$ for the substring $\alpha_\iota(k)\beta_\iota(k)^k\gamma_\iota(k)$ to be the same as the 0-distance sub-path of $y_\iota(k)$ for $\alpha_\iota(k)\beta_\iota(k)^k\gamma_\iota(k)$. If H_1 is not the base of the 0-distance path for $y_\iota(k)$, let $(\phi_\iota, \alpha_\iota, \varphi_\iota, \beta_\iota, \chi_\iota, \gamma_\iota, \psi_\iota)$ be the 0-distance sub-path of H_1 for the substring $\alpha_\iota\beta_\iota\gamma_\iota$ of x. Since α_ι and $\alpha_\iota(k)$ (respectively, β_ι and $\beta_\iota(k)$, γ_ι and $\gamma_\iota(k)$) have the same set of free pairs of states, we can deduce a 0-distance path $(\phi_\iota, \alpha_\iota(k), \varphi_\iota, \beta_\iota(k), \chi_\iota, \gamma_\iota(k), \psi_\iota)$. Since the structure of $\beta_\iota(k)\gamma_\iota(k)$ is the same as the structure of $\gamma_\iota(k)$, there exists a state χ_ι^2 that gives rise to a 0-distance path $(\phi_\iota, \alpha_\iota(k), \varphi_\iota, \beta_\iota(k), \chi_\iota, \beta_\iota(k), \chi_\iota^2, \gamma_\iota(k), \psi_\iota)$. Similarly, we can define $\chi_\iota^3, \chi_\iota^4, \ldots, \chi_\iota^k$ such that $(\phi_\iota, \alpha_\iota(k), \varphi_\iota, \beta_\iota(k), \chi_\iota, \beta_\iota(k), \chi_\iota^2, \beta_\iota(k), \chi_\iota^3, \ldots, \beta_\iota(k), \chi_\iota^k, \gamma_\iota(k), \psi_\iota)$ is a 0-distance path.

We have explained how to construct a 0-distance path for $x(k)$ for a base 0-distance path H_1 of x. For a different base 0-distance path of x, the same construction allows us to define another 0-distance path for $x(k)$.

Remark. In fact, one can prove a slightly stronger statement which states that the 0-distance path from u_i to v_i consuming $(\beta_i(k))^k$ can be part of one 0-distance path for $x(k)$. That is, one can extend the 0-distance path from u_i to v_i consuming $(\beta_i(k))^k$ on both sides to form one 0-distance path for $x(k)$.

2.4.3. *Special case (highway width is 0)*

The special case when the highway width for x is 0 is similar to the general case. The difference is that when the string x is long enough for pumping (that is, when the number of components in the decomposition of x exceeds 2^{4n^2}), there will be no bounded pairs for the relatives of x. When pumping is not considered in order that we can discuss the limited distance behaviors of x, the string x must have less than 2^{4n^2} components, where each component is of highway width 0. Thus, it follows that $f(0) = 2^{4n^2} f(1)$.

2.4.4. *Tighter bound*

Given the relatives of x and the relatives of y, we say that the two families of relatives have the same 'reduced structure' if the two relative families have the same set of good state pairs and the same set of bad state pairs. Since there are n^2 possible state pairs, there could be at most 2^{n^2} possible different reduced structures of relative families.

By inspecting the proof carefully, we see that the structure of the ith partition can be defined in a weaker way as the ordered pair (`reduced structure of` $x_{rc+1,rc+i-1}(k)$, `structure of` $x_{rc+i,rc+c}(k)$). All the arguments in the proof still work. Only the remark made at the end of subsection 2.4.2 no longer holds. The consequence is that we can obtain a stronger result.

Theorem 3. *The distance of an automaton, if limited, is bounded by* $2^{3n^3+n\lg n+n-1}$.

2.4.5. *Comparisons with Hashiguchi's original proof*

The technique presented in this paper is originated by Hashiguchi [7]. We re-organize the proof by introducing concepts (lanes of a highway, relatives) that are easier to understand. Many definitions in Hashiguchi's paper are omitted. We have committed ourselves to present the most direct proof for showing the upper bound on the distance when the automaton is assumed to be limited in distance. We did not prove the several equivalent formulations of the result found in Hashiguchi's paper. Moreover, the second half of the arguments given in case 3 of Section 2.4.2 are new. We have included a discussion of how the highway structure of the kth relative of x relates to the highway structure of x, which we believe is crucial in the correctness of the proof. In addition, we are able to improve the upper bound found by Hashiguchi.

Acknowledgements

We would like to thank a referee for pointing out a mistake in an earlier version of the paper. This research was supported in part by NSF MII grant EIA-9810732.

References

[1] M.R. Garey, D.S. Johnson, Computers and Intractability, Freeman, San Francisco, CA, 1979.
[2] K. Hashiguchi, Regular languages of star height one, Inform. Control 53 (1982) 199–210.
[3] K. Hashiguchi, Limitedness theorem on finite automata with distance functions, J. Comput. Syst. Sci. 24 (1982) 233–244.
[4] K. Hashiguchi, Representation theorems on regular languages, J. Comput. System Sci. 27 (1983) 101–115.
[5] K. Hashiguchi, Algorithms for determining relative star height and star height, Inform. Comput. 78 (1988) 124–169.
[6] K. Hashiguchi, Improved limitedness theorem on finite automata with distance functions, Theoret. Comput. Sci. 72 (1990) 27–38.
[7] K. Hashiguchi, New upper bounds to the limitedness of distance automata, Theoret. Comput. Sci. 233 (2000) 19–32.
[8] H. Leung, On the topological structure of a finitely generated semigroup of matrices, Semigroup Forum 37 (1988) 273–287.
[9] H. Leung, Limitedness theorem on finite automata with distance functions: an algebraic proof, Theoret. Comput. Sci. 81 (1991) 137–145.
[10] H. Leung, The topological approach to the limitedness problem on distance automata, in: J. Gunawardena (Ed.), Idempotency, Publications of the Newton Institute, Cambridge University Press, Cambridge, UK, 1998, pp. 88–111.
[11] H. Leung, V. Podolskiy, The limitedness problem on distance automata, Preproc. of the 3rd Workshop on Descriptional Complexity of Automata, Grammars and Related Structures, Vienna, Austria, July 20–22, 2001, pp. 115–121.
[12] I. Simon, Recognizable sets with multiplicities in the tropical semiring, Lect. Notes Comput. Sci. 324 (1988) 107–120.
[13] I. Simon, Factorization forests of finite height, Theoret. Comput. Sci. 72 (1990) 65–94.
[14] I. Simon, On semigroups of matrices over the tropical semiring, RAIRO ITA 28 (1994) 277–294.