Rep. Fac. Sci. Kagoshima Univ., (Math., Phys. & Chem.),
No. 13, p. 47–54, 1980

REPRESENTATION SYSTEMS OF AUTOMATA BY THEIR TEXTS

By

Makoto HARAGUCHI* and Sueo IIMORI**

(Received Sept. 30, 1980)

Abstract

We study the possibility to represent finite automata by their positive samples called texts. It is shown that, for the class of all finite automata, such a representation is impossible. However, restricting a proper subclass of automata, called stem automata, we really construct the representation system for that class.

1. Representation systems

An effective numbering of objects (machines, languages, etc.) can be considered as a system which represents the objects by natural numbers. The numbers in the system are called codes or indices. The system has an encoder, which associates the objects with the codes, and a decoder, which reconstructs the objects, with the following equation.

$$decode(code(object)) = object.$$

Certainly, the equation is essential to the representation. Hence, one may represent the objects by their data rather than by numbers so long as the equation is satisfied. We denote this idea by the following diagram.

$$\mathbf{OBJ} \underset{g}{\overset{f}{\rightleftarrows}} \mathbf{DATA},$$

where **OBJ** is a class of objects (called a object space), **DATA** is a class of data (called a data space), and f and g are total recursive functions such that $g(f(x))=x$ and that $f(x)$ is "consistent" with x in OBJ.

In this paper, we are interested in classes of finite automata as the object space. Now, what do we allow as a data space? Biermann [1], and Tanatsugu and Arikawa [6] gave the system

$$\mathbf{REG} \underset{g}{\overset{f}{\rightleftarrows}} D^{+} \times N,$$

where **REG** is the class of all regular sets, D^{+} is the class of all finite sets of (positive) strings, and N is the set of all positive integers. Enomoto and Tomita [2] gave the system

* Department of Mathematics, Faculty of Science, Kagoshima University, Kagoshima, Japan.
** Department of Mathematics, Faculty of Education, Saga University, Saga, Japan.

$$\mathbf{AUT} \underset{g}{\overset{f}{\rightleftarrows}} D^{\pm},$$

where **AUT** is the class of all finite automata, and $D^{\pm}$ is the class of all finite sets of signed strings.

While their results are useful ones, what the authors want to study is whether we can represent automata by their positive samples only. Therefore, our data space is D^+. The elements of D^+ are called "texts" (Gold [3]). Formally, we call $d \in D^+$ a text of $x \in \mathbf{REG}$ [**AUT**] if $d \subseteq x$ [$d \subseteq b(x)$], where $b(x)$ is the regular set recognized by x.

DEFINITION 1. *A representation system* (R.S. in short) of a class L is a diagram

$$L \underset{g}{\overset{f}{\rightleftarrows}} D^+ \qquad \text{such that } g(f(x)) = x \text{ and}$$

$f(x)$ is a text of x for each x in L. We denote the R.S. by a 3-tuple (L, f, g).

In Section 2, we show that **AUT** has no representation system.

In Section 3, we define the subclass **STEM** of stem automata, and consider the relations between the stem automata and their texts. The considerations introduce a text generator G_k and an expansion procedure E, and we show that (**STEM**, $\mathbf{G}_k$, E) is a R.S.. The expansion procedure uses "subtext relations", and this technique is originated with Huzino [5].

Another related works are found in Schubert [7] and Gold [4]. Schubert gave a non-effective method to represent partial recursive functions by their finite functions. The representation is based on the sizes of machines, not on the structural relations. Stating in terms of our definition, Gold gave the system $\mathbf{AUT} \underset{g}{\overset{f}{\rightleftarrows}} \mathbf{D}^{\pm}$ with the property that $g(d)=x$ whenever $f(x) \subseteq d$. Our system for stem automata loses the property, however, this is because the negative samples are not allowed in our system.

2. Fundamental Results

The following theorem is also valid for any class L with $L \supsetneqq D^+$.

THEOREM 1. *There is no R.S. of* **REG**.

PROOF. Assume to the contrary that (**REG**, f, g) is a R.S. Since there exists an infinite language in **REG**, f is not an identity function on the subdomain $D^+ \subsetneqq \mathbf{REG}$. Therefore, there exists a set $x_0 \in D^+$ such that $f(x_0) \subsetneqq x_0$ and $f(x_0)$ is not empty. Now assume that $f(fx_0))=f(x_0)$. Then $f(x_0)=g(f(f(x_0))))=g(f(x_0))=x_0$ holds. This contradicts to $f(x_0) \neq x_0$. Hence, $f(f(x_0)) \subsetneqq f(x_0)$ holds and $f(f(x_0))$ is not empty. Similarly, we obtain the infinite sequence of the finite sets $\{f^{(n)}(x_0)\}$ such that

$$f^{(n+1)}(x_0) \subsetneqq f^{(n)}(x_0),$$

where $f^{(n)}(x_0)$ is an abbreviation of $\underbrace{f(f(\cdots f(x_0)\cdots))}_{n \text{ times}}$. Clearly, this is a contradiction.

COROLLARY. *There is no R.S. of* **AUT**.

PROOF. If otherwise, composition of **AUT** $\rightleftarrows$ **D**$^+$ and **REG** $\underset{r}{\overset{b}{\rightleftarrows}}$ **AUT** becomes a *R.S.* of *REG*, where r is a "realization".

In other words, the theorem asserts that it is impossible to represent both infinite and finite sets by texts. On the other hand, finite regular sets are of no concern. Hence, it is natural to restrict our considerations to **REG***, the class of all infinite regular languages.

THEOREM 2. *There is a R.S.* (**REG***, f, g)

PROOF. Let $x_1, x_2, x_3, \cdots$ be an effective enumeration of **REG*** with no repetitions. We define the encoder f by

$$f(x) = \mathit{if}\ \ x = x_1 \text{ then } \min_w [w \in x]$$
$$\text{else (let } x \text{ be } x_k) \ \min_w [w \in x \text{ and } w \notin \{f(x_1), \cdots, f(x_{k-1})\}].$$

The decoder g is defined as

$$g(d) = \mathit{if}\ \ d = \phi \text{ then } \phi \text{ else flag } (\min_w [w \in d])$$
$$\text{flag}(w) = x_{\min_i [f(i) = w]}.$$

It is easy to verify that the functions are totally defined, f is one-to-one, and $g(f(x)) = x$. The consistency is trivial from the definition of f.

The proof of the theorem uses a coding of **REG*** to **N**, hence a text $f(x)$ of x dose not reveals the structure of x. In the next section, we investigate the structural relations between the stem automata and their texts. In counter to the theorem 2, the relations reveals the structures of automata.

3. Stem Automata

A stem automaton has a very simple structure, however, is not trivial. First we list up the necessary definitions.

DEFINITIONS. A *stem* over a finite alphabet Σ is a linear tree such that
1. the leaf is specified by the reserved name "*end*,"
2. the arc to the leaf is labelled by $\# \notin \Sigma$, an end marker, and
3. each arc except 2. is labelled by a letter $\sigma \in \Sigma$.

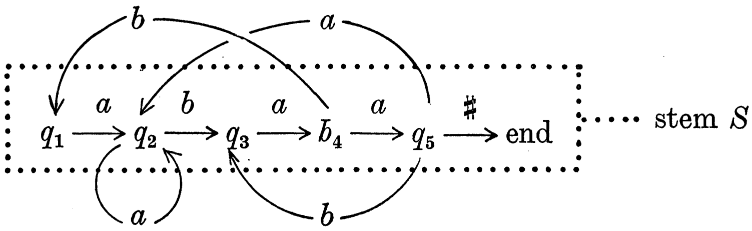
The nodes except leaf are called states. The order on the set of states is defined as

$$q_1 < q_2 \text{ iff } q_2 \text{ is on the path form } q_1.$$

The unique path from q to q' is denoted by $(q \to q')$.

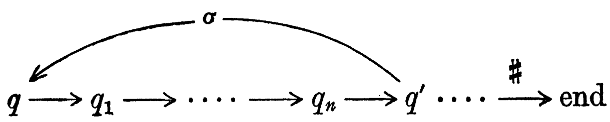
A *stem automaton* consists of a stem S and a set of arcs R, and is denoted by (S, R). Each arc in R is of the form $q_i \xrightarrow{\sigma} q_j$ with $q_i \geq q_j$, and is called "*return*" arc. Moreover, for each letter, there is exactly one transition out of each state. Note that a stem automaton is a determinstic finite automaton under the interpretation that the initial state is the root of the stem and that the state q with $q \xrightarrow{\#} \mathit{end}$ is the unique final state.

Example.



$$R=\{b_4 \xrightarrow{b} q_1,\ q_5 \xrightarrow{b} q_3,\ q_2 \xrightarrow{a} q_2,\ q_5 \xrightarrow{a} q_2\}\ .$$

Now consider the following loop structure to find systematic relations between the stem automata and their texts.



stem path $w=(q \longrightarrow q')$

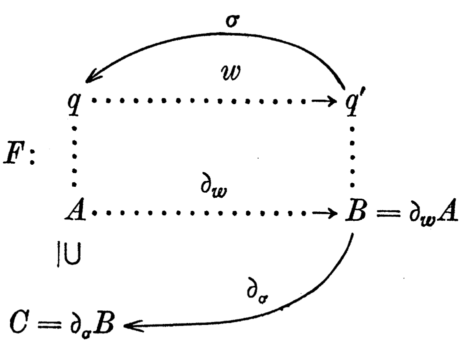
For each $n \geq 0$, $(w\sigma)^n wy \in b(q)$ whenever $y \in b(q')$, where $b(q)$ is the set of all path from q to *end*. Let A be a set

$A=\{wy,\ (w\sigma)wy,\ (w\sigma)^2wy,\ (w\sigma)^3wy, \cdots, (w\sigma)^k wy\}$, a text of q. After scanning w, A becomes

$B=\{y,\ \sigma wy,\ \sigma(w\sigma)wy,\ \sigma(w\sigma)^2 wy, \cdots, \sigma(w\sigma)^{k-1} wy)$, a text of q'. Moreover, after scanning σ, B becomes

$C=(wy,\ (w\sigma)wy,\ (w\sigma)^2wy, \cdots, (w\sigma)^{k-1} wy\}$, a subtext of A.

F:



The figure F roughly shows the relations of texts and transitions, where $\partial_w L$ denotes the derivative of the language L with respect to the string w.

Thus if we generate each text for each state by moving backward through the stem path and by iterating loops continuatively using fixed "loop parameter", then the structure of the automaton corresponds to the expansion of the text using the derivatives, especially, the subtext relations reveal the structure of return arcs. The text generation is done by the text generator G_k and the text expansion by the expansion procedure E.

TEXT GENERATOR G_k

For a stem automaton $x=(S, R)$, a text D_q for each node q is inductively constructed:

Base: for the leaf node, $D_{end}=\{\lambda\}$.

$$\text{Steps:}\quad D_q^{self} = \begin{cases} \sum_{\sigma\in SA_q} \sum_{j=0}^{k} \sigma^j & \text{if } SA_q \neq \phi \\ \{\lambda\} & \text{if otherwise} \end{cases}$$

where $SA_q \ni \sigma$ iff σ is a "self arc" $q \xrightarrow{\sigma} q$.

$$D_q^{ret} = \begin{cases} \sum_{q'\in R_q} [\sum_{j=1}^{k} (wr)^j w]\, D_q & \text{if } R_q \neq \phi \\ \phi & \text{if otherwise,} \end{cases}$$

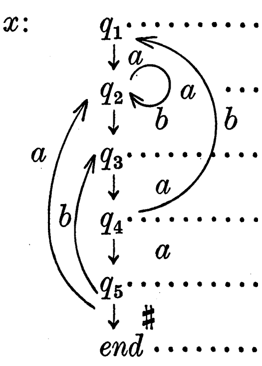
where $R_q \ni q'$ iff q' is a state from which a return arc $q' \xrightarrow{r} q$ exists, and $w=(q\rightarrow q')\neq\lambda$. Finally

$$D_q = D_q^{self}[\tau D_{q'} + D_q^{ret}],$$

where q' is the direct successor of q in the stem, that is, $q \xrightarrow{\tau} q'$ is a stem arc.

$G_k(x)=D_q$, where q is the root of the stem.

EXAMPLE. In the following figure, dotted lines show the correspondences of states and texts, and the loop parameter k is 1.



$$D_{q1} = D_{q2} + abababaD_{q4} = G_1(x)$$
$$D_{q2} = \{\lambda + a\}\,[bD_{q3} + baaabaaD_{q5}]$$
$$D_{q3} = aD_{q4} + aabaaD_{q5}$$
$$D_{q4} = aD_{q5}$$
$$D_{q5} = \{\#\}$$
$$D_{end} = \{\lambda\}$$

The expansion procedure E takes $D \in D^+$ and returns $E(D) \in \textbf{STEM}$. Each state of $E(D)$ is specified by a string w and is denoted by s_w.

EXPANSION PROCEDURE E.

E is specified by giving the following "expansion rules".

R1: the initial state is s_λ

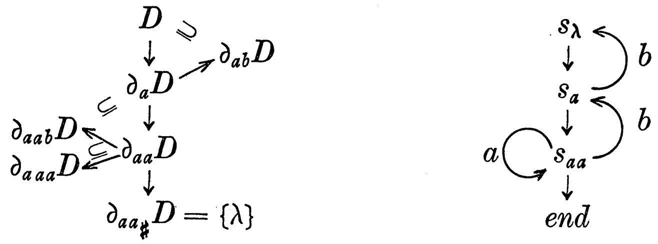
R2: Assume that $s_\lambda, s_{\sigma_1}, s_{\sigma_1\sigma_2}, \cdots, s_{\sigma_1\cdots\sigma_n}$ are already created as states, and let w be $\sigma_1\sigma_2\cdots\sigma_n$.

R2–1. if $\phi \neq \partial_{w\sigma} D \subseteq \partial_{\sigma_1\sigma_2\cdots\sigma_j} D$ for some $0 \leq j \leq n$, create the return arc $s_w \xrightarrow{\sigma} s_{\sigma_1\sigma_2\cdots\sigma_j}$.

R2–2. if there is exactly one $\partial_{w\sigma} D \neq \phi$ such that $\partial_{w\sigma} D \subsetneqq \partial_{\sigma_1\cdots\sigma_j} D$ for any $0 \leq j \leq n$, create the new state $s_{w\sigma}$ and the stem arc $s_w \xrightarrow{\sigma} s_{w\sigma}$. if otherwise, stop the expansion.

R3. If the expansion is stopped at s_w with $\partial_w D = \{\lambda\}$, define $E(D)$ by the automata expanded. Note that s_w is the end node. Otherwise, $E(D)$ is an arbitrary stem automaton.

EXAMPLE. For a text $D = \{a^2\#, a^3\#, a^2ba\#, a^2ba^2\#, aba^2\#, aba^3\#, aba^2ba\#, aba^2ba^2\#\}$, $E(D)$ is



Now let us prove that (**STEM**, G_k, E) is a R.S. From the definition of the text generator, we have

FACT. *For each $k \geq 1$, G_k satisfies the consistency condition, that is, $G_k(x) \subseteq b(x)$ for each x.*

LEMMA. *Let Δ_w be $\partial_w G_k(x)$ for a stem automaton x. Then, for each state q of x, we have*

$$\Delta_w = D_q + \sum_{(q', q'', \sigma) \in C_q} \Big[\sum_{i=1}^{k} (q \to q'') \{\sigma(q' \to q'')\}^i\Big] D_{q''},$$

where w is the stem path $(q_0 \to q)$, q_0 is the initial state, and a 3-tuple $(q', q'', \sigma)) \in C_q$ iff $q'' \xrightarrow{\sigma} q'$ is a return arc with $q'' \geq q$ and $q > q'$.

In what follows, for q', q'', q, and σ with $q' \leq q \leq q''$ and $q'' \xrightarrow{\sigma} q' \in R$, we denote the expression $[\sum_{i=n}^{m} (q \to q'') \{\sigma(q' \to q'')\}^i] Dq''$ by $I(q', q, q''; \sigma: n, m)$. The meaning of this expression is as follows:

"*Iterate the loop $q' \to q \to q''$ i times at the state q, where $n \leq i \leq m$*".

PROOF OF THE LEMMA. By the definition of G_k, we have
$(q_0 \to q)\, D_q \subseteq D_{q_0} = G_k(x)$, and hence

(1) $D_q \subseteq \Delta_w$ *holds.*

For $(q', q'', \sigma) \in C_q$, if any, the definition of $D_{q'}^{ret}$ implies $D_{q'} \supseteq I(q'\, q', q'' \colon \sigma \colon 1, k)$. Thus we have

(2) $$\Delta_w \supseteq \partial_{(q' \to q)}\, D_{q'} \supseteq I(q', q, q''; \sigma; 1, k)\,.$$

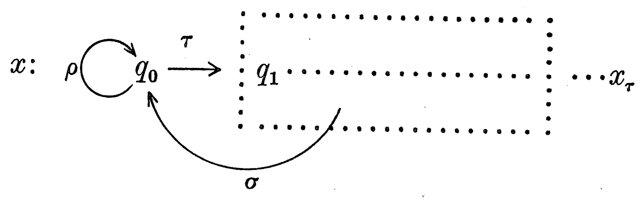
By (1) and (2), we have $\Delta_w \supseteq D_q + \sum_{(q', q'', \sigma) \in C_q} I(q', q, q''; \sigma; 1, k)$. The converse is proved by induction on the number of states. Let $q_0 \xrightarrow{\tau} q_1$ is a stem arc. Then by the definition, we have

(3) $$\partial_\tau D_{q_0} = D_{q_1} + \sum_{(q_0, q'', \sigma) \in C_{q_1}} I(q_0, q_1, q''; \sigma; 1, k)\,.$$

Thus, for each state q except q_0, we have

(4) $$\Delta_{(q_0 \to q)} = \partial_{(q_1 \to q)}\, D_{q_1} + \sum_{\substack{(q_0, q'', \sigma) \in C_{q_1} \\ \varepsilon\, q_1 \leq q \leq q''}} I(q_0, q, q''; \sigma; 1, k)$$

Let x_τ be a subautomaton of x with its root q_1.



By the induction hypothesis for x_τ, and by the fact that $D_q = D_q^\tau$, a text assigned to q by $G_k(x_\tau)$, we have

(5) $$\partial_{(q_1 \to q)} D_{q_1} = D_q + \sum_{\substack{(q', q'', \sigma) \in C_q \\ \varepsilon\, q' \neq q_0}} I(q', q, q''; \sigma; 1, k)$$

Note that $D_{q_1} = G_k(x_\tau)$.
By (4) and (5), we have

$$\begin{aligned} \Delta_{(q_0 \to q)} &= D_q + \sum_{(q', q'', \sigma) \in C_q\ \varepsilon\, q' \neq q_0} I(q', q, q''; \sigma; 1, k) \\ &\quad + \sum_{(q_0, q'', \sigma) \in C_q\ \varepsilon\, q_1 \leq q \leq q''} I(q_0, q, q''; \sigma; 1, k) \\ &\subseteq D_q + \sum_{(q', q'', \sigma) \in C_q} I(q', q, q''; \sigma; 1, k)\,. \end{aligned}$$

The base of our induction is trivial because $D_q = \partial_\lambda D_q = \Delta_\lambda$ and C_q is the empty set.

Now we can state the representation theorem for **STEM**. From the Fact, it suffices to show $E(G_k(x)) = x$.

THEOREM 3. *For each* $k \geq 1$, (**STEM**, G_k, E) *is a R.S.*

PROOF. Assume that the expansion procedure E has already created $s_\lambda, s_{\sigma_1}, \cdots, s_w = s_{\sigma_1 \cdots \sigma_n}$ from the text $G_k(x)$ of x and that s_w corresponds a state q of x. If q has a self arc $q \overset{\rho}{\to} q$, then

$$D_q^{self} \supseteq \{\lambda, \rho, \cdots \rho^k\}$$

$$(1) \qquad \Delta_q = D_q^{self}[\tau D_{q'} + D_q^{ret}] + \sum_{(q', q'', \sigma) \in C_q} I(q', q, q''; \sigma; 1, k)$$

holds. Since stem automata are deterministic, we have

$$\partial_\rho \Delta_q = \partial_{w\rho} G_k(x) = \{\lambda, \rho, \cdots, \rho^{k-1}\} [\tau D_{q'} + D_q^{ret}] \subseteq \Delta_q .$$

Thus, the self arc $s_w \overset{\rho}{\to} s_w$ is formed by the expansion rule R2–1.
Similarly, for a return arc $q \overset{\sigma}{\to} q$, the equation (1) implies

$$\partial_\sigma \Delta_w = I(q_1, q_1, q; \sigma; 0, k-1) = (q_1 \to q)\, D_q + I(q_1, q_1, q; \sigma; 1, k-1) .$$

Note that, among the elements of C_q, only (q_1, q, σ) is remained by ∂_σ.
Since $(q_1 \to q)\, D_q \subseteq D_{q_1}$ and
$I(q_1, q_1, q; \sigma; 1, k\text{-}1) \subseteq I(q_1, q_1, q; \sigma; i, k) \subseteq D_{q_1}^{ret} \subseteq D_{q_1}$ holds, we have

$$\partial_\sigma \Delta_w \subseteq D_{q_1} \subseteq \Delta_{(q_0 \to q_1)} .$$

Hence, the return arc $s_w \overset{\sigma}{\to} s_{(q_0 \to q_1)}$ is created by R2–1. Finally, we verify that the stem is exactly expanded. Let w_q be $(q \to end)$, then w_q is the minimal length string in $b(q)$. Since $|w_q| < |w_{q'}|$ holds for each q' with $q' < q$, we have

$$w_q \in D_q \subseteq \Delta_q \text{ and } w_q \notin \Delta_{q'} \subseteq b(q') .$$

Thus, if $q \overset{\tau}{\to} q_2$ is a stem arc, we have
$\Delta_{(q_0 \to q)\tau} \subsetneq \Delta_{(q_0 \to q')}$ for each q' with $q' < q$. That is, the new state $s_{(q_0 \to q)\tau}$ and the transtion $s_{(q_0 \to q)} \overset{\tau}{—} s_{(q_0 \to q)\tau}$ are created by the rule R2–2.

References

[1] A.W., BIERMANN, *An interactive finite state language learner*, 1-st USA-Japan Computer Conf. Proc., 13–23 (1972).

[2] H. ENOMOTO and E. TOMITA, *A representative set of strings for a deterministic finite-state automaton*, UDC, J-59, 660–667 (1976).

[3] E.M. GOLD, *Language identification in the limit*, Information and Control 10, 447–474 (1967).

[4] E.M. GOLD, *Complexity of automation identification from given data*, Information and Control 37, 302–320 (1978).

[5] S. HUZINO, *On inferring dynamics of automata from finite samples*, Mem. Fac. Sci. Kyushu Univ. Ser A 32, 291–299 (1978).

[6] K. TANATSUGU and S. ARIKAWA, *On characteristic sets and degrees of finite automata*, International Journal of Computer and Information Sciene 6, 83–93 (1977).

[7] L.K. SCHUBERT, *Representative samples of programmable functions*, Information and Control 25, 30–44 (1974).