ANNALES ACADEMIAE SCIENTIARUM FENNICAE

Series A

I. MATHEMATICA

525

# ON A HOMOMORPHIC CHARACTERIZATION OF RECURSIVELY ENUMERABLE LANGUAGES

BY

ARTO SALOMAA

HELSINKI 1972
SUOMALAINEN TIEDEAKATEMIA

https://doi.org/10.5186/aasfm.1973.525

ISBN 951-41-0066-2

Communicated 8 September 1972

KESKUSKIRJAPAINO
HELSINKI 1972

## 1. Introduction

According to the well-known theorems of Medvedev and Schützenberger, every regular language is a homomorphic image of a 2-testable language and every context-free language is a homomorphic image of the intersection of a Dyck language and a 2-testable language. Many such homomorphic representations are known also for the family of recursively enumerable languages. Every recursively enumerable language is a homomorphic image of the intersection of two deterministic context-free languages, [3], and a homomorphic image of a language generated by a context-sensitive grammar in linear time, [1], as well as a homomorphic image of a language generated by a $\lambda$-free context-free programmed grammar, [6].

The purpose of this paper is to establish by a direct combinatorial argument the following result. Consider a fixed alphabet $V_T$. Then there exist another alphabet $V'$, deterministic context-free languages $L_1$ and $L_2$ over $V'$ and a homomorphism $h$ of $W(V')$ onto $W(V_T)$ with the following property. For every recursively enumerable language $L$ over $V_T$, there is a regular noncounting language $K$ over $V'$ such that

$$L = h(L_1 \cap L_2 \cap K) .$$

Thus, only the noncounting language $K$ depends on $L$, everything else is determined by the alphabet of $L$ alone. Essentially the same result has been proven by Fisher and Raney, [2]. Their proof, however, is based on a complicated theory of automata on networks.

Homomorphic representation can be used for proving results concerning decidability, nonclosure and generative capacity, [1], [6]—[9]. Some simple applications to decidability will be considered also in this paper. Very few homomorphic representations are known for Lindenmayer systems (cf. [5], [7], [10], [11]). This may be due to the resistance displayed by these systems against closure operations. We would like to mention, finally, that in spite of the many homomorphic representations given for recursively enumerable languages, there still is no satisfactory general theory concerning such representations.

## 2. Definitions and results

The reader is assumed to be familiar with the basic notions concerning automata and formal languages, [12]. As regards locally testable and noncounting languages, cf. [4].

The set of all words, including the empty word $\lambda$, over an alphabet $V$ is denoted by $W(V)$. We use the customary notations $mi$, $^*$, $^+$ for mirror image, catenation closure and $\lambda$-free catenation closure. In the statement of the following theorem,

$$V_T = \{a_1, \ldots, a_{u-4}\}, \quad u \geqq 5,$$

$$V'_T = \{a'_i \mid a_i \in V_T\},$$

$$V' = V'_T \cup \{0, 1\}.$$

**Theorem 1.** *There exist two deterministic context-free languages $L_1$ and $L_2$ over $V'$ and a homomorphism $h$ of $W(V')$ onto $W(V_T)$ with the following property. For every recursively enumerable language $L$ over $V_T$, there is a noncounting regular language $K$ over $V'$ such that*

$$L = h(L_1 \cap L_2 \cap K). \tag{1}$$

All constructions involved in the proof of Theorem 1 are effective. Since the emptiness problem is undecidable for recursively enumerable languages and since a homomorphic image of a language is empty if and only if the language itself is empty, the following theorem is an immediate corollary of Theorem 1.

**Theorem 2.** *There are two deterministic context-free languages $L_1$ and $L_2$ such that there is no algorithm for deciding of an arbitrary noncounting regular language $K$ whether or not the intersection $L_1 \cap L_2 \cap K$ is empty.*

In some sense, Theorem 2 can be considered as an unsolvability result for regular languages since $K$ is the only variable. However, one can also say that it is not a problem dealing »properly» with regular languages.

Many undecidability results similar to Theorem 2 can be obtained. We mention only the following, due to the fact that one can easily (by analyzing the proof in the next section) modify the construction in such a way that $L_1 \cap L_2 \cap K$ is nonempty if and only if it is infinite.

**Theorem 3.** *There are two deterministic context-free languages $L_1$ and $L_2$ such that there is no algorithm for deciding of an arbitrary noncounting regular language $K$ whether or not the intersection $L_1 \cap L_2 \cap K$ is finite.*

## 3. Proof of Theorem 1

Every recursively enumerable language over $V_T$ is generated by a type 0 grammar. Without loss of generality, we assume that the nonterminals form an initial segment of the sequence

$$a_{u+1}\ ,\ a_{u+3}\ ,\ a_{u+5}\ ,\ \ldots\ , \tag{2}$$

$a_{u+1}$ being the initial letter.

In what follows, the letters

$$a_{u-3}\ ,\ a_{u-2}\ ,\ a_{u-1}\ ,\ a_u$$

will play the role of boundary markers, and the letters in the sequence

$$a_{u+2}\ ,\ a_{u+4}\ ,\ a_{u+6}\ ,\ \ldots$$

the role of production indicators.

All letters $a_i$, $i \geqq 1$, will now be encoded in the alphabet $\{0, 1\}$ by defining the homomorphism

$$h_1(a_i) = 10^i1\ ,\ i \geqq 1\ .$$

To make the following definitions more readable, we also use the following abbreviations

$$h_1(a_u) = c\ ,\quad h_1(a_{u-i}) = c_i \text{ for } 1 \leqq i \leqq 3\ ,$$
$$h_1(a_{u+1}) = S\ ,\quad h_1(a_{u+2i}) = d_i \text{ for } i \geqq 1\ .$$

Furthermore, we denote

$$U_T = (h_1(a_1) \cup \ldots \cup h_1(a_{u-4}))^*\ .$$

By $U$ we denote the language consisting of $\lambda$ and of all words of the form $h_1(a_{i_1}) \ldots h_1(a_{i_v})$ where $v \geqq 1$ and, for each $j = 1, \ldots, v$, either $1 \leqq i_j \leqq u-4$ or else $i_j = u + 2r + 1$ for some $r \geqq 0$. (Intuitively, $U_T$ consists of encoded words over the terminal alphabet and $U$ of encoded words in terminal and nonterminal letters.)

We are now in the position to define the homomorphism $h$ and the two languages $L_1$ and $L_2$ over $V'$. By definition

$$h(b) = \begin{cases} a_i \text{ for } b = a_i' \ (a_i' \in V_T')\ , \\ \lambda \text{ for } b \in \{0\ ,\ 1\}\ , \end{cases}$$

$$L_1 = \{P_1c_1\alpha d_iP_2c\ \ mi(P_2)\ \ c_2\ \ mi(\beta)\ \ c_3\ \ mi(P_1)c \mid$$
$$P_1\ ,\ P_2\ ,\ \alpha\ ,\ \beta \in U\ ,\ i \geqq 1\}^+\ (V_T')^*$$

and

$$L_2 = \{c_1 S d_i c \mid i \geqq 1\}\ \{P_1 c_2 P_2 c_3 P_3 c P_4 c_1 P_5 d_j P_6 c \mid$$
$$P_1, P_2, P_3, P_4, P_5, P_6 \in U,\ j \geqq 1,\ P_1 P_2 P_3 = mi(P_4 P_5 P_6)\}^*$$
$$\{Q_1 c_2 Q_2 c_3 Q_3 c Q_4 \mid Q_4 \in W(V_T')\};$$
$$Q_1, Q_2, Q_3 \in U_T,\ Q_1 Q_2 Q_3 = h_1 h(mi(Q_4))\}.$$

So far our definitions are based on the alphabet $V_T$ alone. We now consider an arbitrary but fixed recursively enumerable language $L$ over $V_T$, generated by a type 0 grammar $G = (V_N, V_T, a_{u+1}, F)$. Denote $V = V_N \cup V_T$. We assume that $V_N$ consists of a finite initial segment of (2) and the production set is

$$F = \{\alpha_i \to \beta_i \mid 1 \leqq i \leqq k\}.$$

We now define $K = K_1 \cap K_2$, where

$$K_1 = \{P_1 c_1 h_1(\alpha_i) d_i P_2 c P_3 c_2 h_1(mi(\beta_i)) c_3 P_4 c \mid$$
$$P_1, P_2, P_3, P_4 \in (h_1(V))^*,\ 1 \leqq i \leqq k\}^+ (V_T')^*$$

and

$$K_2 = \{c_1 S d_i c \mid 1 \leqq i \leqq k\}\ \{P_1 c_2 P_2 c_3 P_3 c P_4 c_1 P_5 d_j P_6 c \mid$$
$$P_1, P_2, P_3, P_4, P_5, P_6 \in (h_1(V))^*,\ 1 \leqq j \leqq k\}^*$$
$$\{Q_1 c_2 Q_2 c_3 Q_3 c Q_4 \mid Q_4 \in W(V_T');\ Q_1, Q_2, Q_3 \in U_T\}.$$

It is immediately verified that $L_1$ and $L_2$ are deterministic context-free languages over the alphabet $V'$. It is also obvious that $K_1$ and $K_2$ are denoted by star-free regular expressions (involving intersections and complements) and, consequently, $K$ is a noncounting regular language. We shall now prove that (1) holds. For this purpose, we introduce two auxiliary languages $L_1'$ and $L_2'$ over the alphabet $V'$ as follows:

$$L_1' = \{P_1 h_1(\alpha_i) d_i P_2 c\ mi(P_2)\ h_1(mi(\beta_i))\ mi(P_1) c \mid$$
$$1 \leqq i \leqq k;\ P_1, P_2 \in (h_1(V))^*\}^+ (V_T')^*$$

and

$$L_2' = \{S d_i c \mid 1 \leqq i \leqq k\}\ \{mi(P_1 P_2) c P_1 d_j P_2 c \mid$$
$$P_1, P_2 \in (h_1(V))^*,\ 1 \leqq j \leqq k\}^*\ \{Q_1 c Q_2 \mid$$
$$Q_2 \in W(V_T'),\ Q_1 = h_1 h(mi(Q_2))\}.$$

We claim that

$$L = h(L_1' \cap L_2'). \tag{3}$$

To prove the equation (3), we show first that the left side is included in the right side. Assume that $P \in L$. Since $G$ generates $L$, there is an integer $m$, words $R_{j1}$, $R_{j2} \in W(V)$ and indices $g(j)$ with $1 \leqq g(j) \leqq k$, defined for all $j = 0, \ldots, m$, such that the following conditions are satisfied. For every $j = 0, \ldots, m-1$,

$$R_{j1}\beta_{g(j)}R_{j2} = R_{(j+1)1}\alpha_{g(j+1)}R_{(j+1)2} . \tag{4}$$

Furthermore,

$$R_{01} = R_{02} = \lambda , \ \alpha_{g(0)} = a_{u+1} , \ R_{m1}\beta_{g(m)}R_{m2} = P . \tag{5}$$

In other words, we consider the following derivation according to $G$:

$$a_{u+1} = R_{01}\alpha_{g(0)}R_{02} \Rightarrow R_{01}\beta_{g(0)}R_{02} = R_{11}\alpha_{g(1)}R_{12}$$
$$\Rightarrow R_{11}\beta_{g(1)}R_{12} = R_{21}\alpha_{g(2)}R_{22} \Rightarrow \ldots \Rightarrow$$
$$R_{(m-1)1}\beta_{g(m-1)}R_{(m-1)2} = R_{m1}\alpha_{g(m)}R_{m2} \Rightarrow R_{m1}\beta_{g(m)}R_{m2} = P .$$

We now define, for any $P_1$, $P_2 \in W(V)$ and $1 \leqq i \leqq k$,

$$t(P_1 , i , P_2) = h_1(P_1)h_1(\alpha_i)d_ih_1(P_2)c \; mi(h_1(P_2)) \; mi(h_1(\beta_i)) \quad mi(h_1(P_1))c$$

and consider the word

$$R = t(R_{01} , g(0) , R_{02})t(R_{11} , g(1) , R_{12}) \ldots t(R_{m1} , g(m) , R_{m2})P' , \tag{6}$$

where $P'$ is obtained from $P$ by replacing every letter with the corresponding primed one. (Thus, $h(P') = P$.) By the definition of $L_1'$, we have $R \in L_1'$. (Note that the operators $mi$ and $h_1$ commute). Using the notation

$$s(P_1 , i , P_2) = mi(h_1(P_1P_2))ch_1(P_1)d_ih_1(P_2)c ,$$

we may also write (by (4) and (5))

$$\begin{aligned} R &= Sd_{g(0)}c \; mi(h_1(R_{01}\beta_{g(0)}R_{02})) \; ch_1(R_{11})h_1(\alpha_{g(1)})d_{g(1)} \\ &\quad h_1(R_{12})c \ldots mi(h_1(R_{(m-1)1}\beta_{g(m-1)}R_{(m-1)2}))c \\ &\quad h_1(R_{m1})h_1(\alpha_{g(m)})d_{g(m)}h_1(R_{m2})c \; mi(h_1(R_{m1}\beta_{g(m)}R_{m2}))cP' \\ &= Sd_{g(0)}c \; mi(h_1(R_{11}\alpha_{g(1)}R_{12}))ch_1(R_{11})h_1(\alpha_{g(1)})d_{g(1)} \\ &\quad h_1(R_{12})c \ldots mi(h_1(R_{m1}\alpha_{g(m)}R_{m2}))c \\ &\quad h_1(R_{m1})h_1(\alpha_{g(m)})d_{g(m)}h_1(R_{m2})c \; mi(h_1(R_{m1}\beta_{g(m)}R_{m2}))cP' \\ &= Sd_{g(0)}cs(R_{11}\alpha_{g(1)} , g(1) , R_{12}) \\ &\quad \ldots s(R_{m1}\alpha_{g(m)} , g(m) , R_{m2}) \; mi(h_1(P))cP' . \end{aligned}$$

From the last expression we see that $R \in L_2'$ and, hence, $R \in L_1' \cap L_2'$. On the other hand, $h(R) = P$. This implies that $P \in h(L_1' \cap L_2')$.

Having established that the left side of (3) is included in the right side, we now prove the reverse inclusion. Assume that $P \in h(L_1' \cap L_2')$. Consequently, there is a word $R \in L_1' \cap L_2'$ such that $P = h(R)$. Since $R \in L_1'$, it can be expressed in the form (6), for some numbers $m$, $g(i)$ and words $R_{ij}$. (Recall that $P'$ is obtained from $P$ by replacing every letter with the corresponding primed one.) Since $R \in L_2'$, the word $R$ can be expressed in the form

$$R = Sd_{f(0)}cs(Q_{11}, f(1), Q_{12}) \ldots s(Q_{n1}, f(n), Q_{n2})\ mi(h_1(P))cP', \tag{7}$$

for some numbers $n$, $f(i)$ and words $Q_{ij}$. Comparing the number of occurrences of the boundary marker $c$ in (6) and (7), we see that $m = n$. It is also clear that $g(i) = f(i)$, for $i = 0, \ldots, m$. A further comparison between (6) and (7) gives the equations

$$\begin{aligned} &R_{01} = R_{02} = \lambda, \ \alpha_{g(0)} = a_{u+1}, \ R_{i1}\alpha_{g(i)} = Q_{i1}, \\ &R_{i2} = Q_{i2}, \ R_{(i-1)1}\beta_{g(i-1)}R_{(i-1)2} = Q_{i1}Q_{i2} = R_{i1}\alpha_{g(i)}R_{i2}, \end{aligned} \tag{8}$$

for $1 \leqq i \leqq m$, and also the equation

$$P = R_{m1}\beta_{g(m)}R_{m2}.$$

Thus, we obtain the following derivation according to $G$:

$$\begin{aligned} &a_{u+1} = R_{01}\alpha_{g(0)}R_{02} \Rightarrow R_{01}\beta_{g(0)}R_{02} = R_{11}\alpha_{g(1)}R_{12} \\ &\Rightarrow \ldots \Rightarrow R_{(m-1)1}\beta_{g(m-1)}R_{(m-1)2} = R_{m1}\alpha_{g(m)}R_{m2} \\ &\Rightarrow R_{m1}\beta_{g(m)}R_{m2} = P. \end{aligned}$$

Therefore, $P \in L$. Thus, we have shown that the equation (3) is correct.

For a language $L'$, let $M(L')$ be the language obtained from $L'$ by erasing from all words all occurrences of $c_1$, $c_2$ and $c_3$. (Note that $M$ is not a homomorphism since $c_i$ is a sequence of 0's and 1's. Note also that, for any language $L'$, $h(L') = h(M(L'))$.) Comparing the positions of the boundary markers $c_i$ and the production indicators $d_j$, we obtain the equations

$$M(L_1 \cap K_1) = L_1', \ M(L_2 \cap K_2) = L_2'. \tag{9}$$

The inclusion

$$L_1' \cap L_2' \subseteq M(L_1 \cap L_2 \cap K_1 \cap K_2) \tag{10}$$

is established by (i) considering an arbitrary word $R$ belonging to the left side, (ii) noting that $R$ can be expressed in the forms (6) and (7) from which (8) can be inferred, (iii) inserting the markers $c_1$, $c_2$, $c_3$ in $R$ at proper places which are immediately seen from the $t$-expressions, and (iv)

noting that the resulting word belongs to all four languages in the intersection on the right side. By (9) and (10),

$$M(L_1 \cap L_2 \cap K_1 \cap K_2) = L_1' \cap L_2' .$$

Hence,

$$h(L_1 \cap L_2 \cap K) = h(M(L_1 \cap L_2 \cap K)) \tag{11}$$
$$= h(M(L_1 \cap L_2 \cap K_1 \cap K_2)) = h(L_1' \cap L_2') .$$

The equation (1) is now an immediate consequence of (3) and (11). This completes the proof.

We note that $K$ is not, in general, locally testable. It does not seem likely that the construction could be modified to yield a locally testable language.

Mathematics Department
University of Turku, Finland

## References


[1] Book, R. V.: Grammars with time functions. Aiken Computation Laboratory, Harvard University, Report No. NSF-23 (1969).

[2] Fisher, G. and Raney, G.: On the representation of formal languages using automata on networks. IEEE Conf. Record of Tenth Annual Symp. on Switching and Automata Theory (1969) 157–165.

[3] Ginsburg, S., Greibach, S. and Harrison, M.: One-way stack automata. J. Assoc. Comput. Mach. 14 (1967) 389–418.

[4] McNaughton, R. and Papert, S.: Counter-Free Automata. Research Monograph No. 65, M.I.T. Press (1971) 163 pp.

[5] Paz, A. and Salomaa, A.: Integral sequential word functions and growth equivalence of Lindenmayer systems. Information and Control, to appear.

[6] Rosenkrantz, D. I.: Programmed grammars and classes of formal languages. J. Assoc. Comput. Mach. 16 (1969) 107–131.

[7] Salomaa, A.: Formal Languages. Academic Press (in press).

[8] —»— Matrix grammars with a leftmost restriction. Information and Control 20 (1972) 143–149.

[9] —»— The generative capacity of transformational grammars of Ginsburg and Partee. Ibid. 18 (1971) 227–232.

[10] —»— On sentential forms of context-free grammars. Acta Informatica, to appear.

[11] —»— On exponential growth in Lindenmayer systems. Indagationes Mathematicae, to appear.

[12] —»— Theory of Automata. Pergamon Press (1969) 276 pp.


---


Printed November 1972