

K. CULIK II

H. A. MAURER

On simple representations of language families

RAIRO. Informatique théorique, tome 13, n° 3 (1979), p. 241-250

http://www.numdam.org/item?id=ITA_1979__13_3_241_0

© AFCET, 1979, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Informatique théorique » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques

<http://www.numdam.org/>

ON SIMPLE REPRESENTATIONS OF LANGUAGE FAMILIES (*) (1)

by K. CULIK II (2) and H. A. MAURER (3)

Communiqué par J. BERSTEL

Abstract. — In this paper we establish representation results for families of languages analogous to the Chomsky-Schützenberger theorem for context-free languages and analogous to Greibach's theorem on the hardest context-free language. We show that, using intersection with regular sets and certain simple homomorphisms, the family of recursively enumerable sets, each principal AFL and (under weaker assumptions) each countable family of languages can be generated from one individual language. We then extend Greibach's hardest context-free language theorem to recursively enumerable sets and to the family of context-sensitive languages, the latter result also providing a particularly simple proof that this family of languages is a principal AFL. In contrast to these results we then mention that no such result is possible for the family of regular languages.

Résumé. — Dans cet article, nous établissons des résultats de représentation de familles de langages qui sont analogues au théorème de Chomsky-Schützenberger pour les langages « context-free » (algébriques) et au théorème de Greibach sur le langage algébrique « le plus difficile ». Nous montrons que la famille des ensembles récursivement énumérables, toute AFL principale et (sous des hypothèses plus faibles) toute famille dénombrable de langages peut être engendrée à partir d'un langage unique au moyen d'intersections avec des langages rationnels et de certains homomorphismes simples. Nous étendons le théorème de Greibach sur le langage algébrique le plus difficile aux ensembles récursivement énumérables et à la famille des langages « context-sensitive »; ce dernier résultat donne ainsi une preuve particulièrement simple du fait que cette famille de langages et une AFL principale. En opposition à ces résultats, nous mentionnons alors qu'aucun résultat de ce type n'est possible pour la famille des langages rationnels (réguliers).

0. INTRODUCTION

One of the most important aims of language theory has been the establishment of so-called *representation theorems* for families of languages \mathcal{L} of the following type: There exists a language U , called *generator* (and usually $U \in \mathcal{L}$) such that each $L \in \mathcal{L}$ can be written as $L = f(U)$, where f is a simple combination of simple language operations.

(*) Reçu octobre 1978, et dans sa version définitive en janvier 1979.

(1) Research in this paper was supported by the Natural Sciences and Engineering Research Council of Canada, Grant No. 7403, and by the Austrian Federal Ministry of Science and Research.

(2) Department of Computer Science, University of Waterloo, Ontario, Canada.

(3) Institute of Information Processing, Technical University of Graz, Austria.

One example is the Chomsky-Schützenberger theorem for the family of *CF* languages which asserts that a Dyck language can be chosen as U and that f can be chosen to be the intersection with a regular set followed by a particularly simple type of homomorphism. Another example is Greibach's theorem on the hardest *CF* language establishing for the family of *CF* languages that with a proper choice of U the mapping f can be taken to be a single inverse homomorphism. Still another example is the notion of a full principal *AFL* since for any such full principal *AFL* \mathcal{L} there exists a U such that every $L \in \mathcal{L}$ can be written as $L = f(U)$, where f is a finite (rational) transduction.

A number of other similar results is known in the literature. In particular, a Chomsky-Schützenberger type theorem has recently been proved for the family of *RE* sets. We strengthen this result in our theorem 1, establish a similar result for every full principal *AFL* in theorem 2 and a weaker result (thm. 3) for every countable family \mathcal{L} of languages (weaker in as much as U will, in general, not be in \mathcal{L}). We then prove two results for the family of *RE* sets (thms. 4 and 5) analogous to Greibach's theorem on the hardest *CF* language. Modifying the proof of theorem 5 we obtain that every context-sensitive language can be obtained as inverse homomorphism of a single fixed context sensitive language (thm. 6) thus giving a particularly simple proof that the family of context-sensitive languages is a principal *AFL*, a result originally obtained in [14], cf. also [7], p. 139. We finally mention that no such theorem can hold for the family of regular languages (thm. 7).

Throughout the paper we assume familiarity with basic formal language theory. For any terminology not explained in this paper [9-12] may be consulted.

Section 1 contains a summary of only such definitions and terminology which are of a more specific nature. Section 2 contains the results, presented in seven theorems.

1. PRELIMINARIES

In this section we summarize some of the definitions and terminology of this paper. A familiarity with basic formal language theory is assumed throughout.

A homomorphism $h : \Sigma^* \rightarrow \Delta^*$ is called an *erasing* if for some subset T of Σ we have $h(a) = a$ if $a \in T$ and $h(a) = \varepsilon$, otherwise. Throughout the paper such an erasing will be denoted by Π_T .

Let h_1, h_2 be two homomorphisms, $h_1, h_2 : \Sigma^* \rightarrow \Delta^*$. The *minimal equality set* of h_1 and h_2 , denoted by $e(h_1, h_2)$ is defined by:

$$e(h_1, h_2) = \{ w \in \Sigma^+ \mid h_1(w) = h_2(w) \text{ and if } w = uv \\ \text{where } u \in \Sigma^+, v \in \Sigma^+, \text{ then } h_1(u) \neq h_2(u) \}.$$

Throughout this paper, if Σ is an alphabet, $\bar{\Sigma}$ will denote an alphabet disjoint from Σ consisting of "barred" symbols, $\bar{\Sigma} = \{\bar{a} \mid a \in \Sigma\}$. For any word $x \in \Sigma^*$, \bar{x} denotes the word obtained from x by barring each symbol.

Let Σ be an alphabet. The *twin-shuffle* over Σ is a language over $(\Sigma \cup \bar{\Sigma})^*$, denoted by $L(\Sigma)$ and defined by:

$$L(\Sigma) = \{x \in (\Sigma \cup \bar{\Sigma})^* \mid \overline{\Pi_{\Sigma}(x)} = \Pi_{\bar{\Sigma}}(x)\}.$$

A transducer t is defined, as is usual, as a 6-tuple $t = (\Sigma, \Delta, \Phi, M, q_0, F)$, where Σ is an alphabet of inputs, Δ an alphabet of outputs, Φ a finite set of states, $q_0 \in \Phi$ a start state, $F \subseteq \Phi$ a set of final or accepting states, and where M is a finite subset of $\Phi \times \Sigma^* \times \Phi \times \Delta^*$, specifying the behaviour of t . A quadruple (p, x, q, y) indicates that t in state p with input x may switch to state q and produce output y . A transducer t as above is called *simple* if $(p, x, q, y) \in M$ implies $|x| \leq 1$ and $|y| \leq 1$.

Finite transducers can be defined by state diagrams in the obvious way.

2. GENERATION OF LANGUAGE FAMILIES FROM A SINGLE LANGUAGE

2.1. Generation using intersection with a regular set followed by a homomorphism

In this subsection we consider the problem of representing each language L of a family of languages \mathcal{L} as the homomorphic image of the intersection of some (presumably simple) language D_L and a regular set.

The historically first and most widely known result of this type is the well known Chomsky-Schützenberger theorem which can be stated as follows:

P_1 : For every *CF* language L there exist a Dyck language D_L , a regular set R and a homomorphism h such that $L = h(D_L \cap R)$.

Indeed, a stronger version, where D_L does not depend on L but only the alphabet of L , and where h is an erasing is also known to hold:

P'_1 : Let T be an arbitrary alphabet. There exists a language D_T such that for every *CF* language $L \subseteq T^*$ there exist a regular set R and an erasing Π_T such that $L = \Pi_T(D_T \cap R)$.

Similar results have also been established for other language families. For instance, a result analogous to P'_1 has been proven in [2] for both *EOL* and *ETOL* languages:

P_2 : Let T be an arbitrary alphabet. Let X stand (consistently) for either *EOL*

or *ETOL*. There exists a language $L_T^{(x)}$ such that for every X language $L \subseteq T^*$ there exists a regular set R and an erasing Π_T such that $L = \Pi_T(L_T^{(x)} \cap R)$.

Rather recently, a similar result has also been obtained for *RE* languages in [6]:

P_3 : For every *RE* language L there exist a twin-shuffle $L(\Sigma)$, a regular set R and an erasing Π such that $L = \Pi(L(\Sigma) \cap R)$.

In what follows we first present an alternate proof of P_3 (lemma 1) based on a result in [3]. A modification thereof shows that the language D_L in P_3 can be chosen to depend only on the alphabet of L (thm. 1), a strengthening analogous to P'_1 . This result can also be obtained readily from the ideas in [6] but our proof idea seems to be of independent interest. We then show that a result analogous to P_1 holds for every principal *AFL* and that even an erasing instead of a homomorphism suffices (thm. 2). We finally observe that a similar result holds for any countable family of languages (thm. 3) but that the generator used then will, in general, not be a language of the family of languages at issue.

LEMMA 1: For every *RE* language $L \subseteq T^*$ there exist a twin-shuffle $L(\Gamma)$, a regular set R_L and an erasing Π_T such that $L = \Pi_T(L(\Gamma) \cap R_L)$.

Proof: By theorem 1 of [3] we can write $L = \Pi_T(e(h_1, h_2))$ for some homomorphisms $h_1, h_2: \Sigma^* \rightarrow \Delta^*$ and $T \subseteq \Sigma$. We may assume that $\Delta \cap \Sigma = \emptyset$. Moreover, it follows from the proof of the theorem that we may assume that a symbol $\bar{3}$ is in Σ and $e(h_1, h_2) \subseteq (\Sigma - \{\bar{3}\})^* \{\bar{3}\}$.

Let $\bar{\Sigma} = \{\bar{a} \mid a \in \Sigma\}$, $\bar{\Delta} = \{\bar{b} \mid b \in \Delta\}$, $\Gamma = \Sigma \cup \Delta$, $\bar{\Gamma} = \Sigma \cup \bar{\Delta}$

and let \bar{w} be the word obtained from a word w by barring each symbol, $w \in (\Sigma \cup \Delta)^*$. Let $F = \{ah_1(a)\bar{h}_2(a) \mid a \in \Sigma\}$ and let $R_L = \Pi_\Omega(F^*) \{\bar{3}\}$, where $\Omega = \Gamma \cup \bar{\Delta} \cup \{\bar{3}\}$.

Clearly, $e(h_1, h_2) = \Pi_\Sigma(L(\Gamma) \cap R_L)$. Note in particular that only "minimal solutions" are in $L(\Gamma) \cap R_L$, since symbol $\bar{3}$ acts as an "endmarker". \square

We now strengthen lemma 1 by showing that for each alphabet T we can use the twin-shuffle $L(T \cup \{0, 1\})$ as a fixed generator for every $L \subseteq T^*$.

THEOREM 1: Let $L \subseteq T^*$ be an *RE* language and let $L(T \cup \{0, 1\})$ be the twin-shuffle over the alphabet $T \cup \{0, 1\}$. There exists a regular set $R \subseteq (T \cup \bar{T} \cup \{0, \bar{0}, 1, \bar{1}\})^*$ such that for the erasing Π_T we have $L = \Pi_T(L(T \cup \{0, 1\}) \cap R)$.

Proof: Let $L \subseteq T^*$ be an arbitrary *RE* language. By lemma 1 there exist an alphabet Γ , $T \subseteq \Gamma$ and a regular set $Q \subseteq \Gamma^*$ so that $L = \Pi_T(L(\Gamma) \cap Q)$. Let $\Gamma - T = \{c_1, c_2, \dots, c_m\}$ and let $g: (\Gamma \cup \bar{\Gamma})^* \rightarrow T \cup \bar{T} \cup \{0, 1, \bar{0}, \bar{1}\}$ be the

homomorphism defined by:

$$g(a) = a \quad \text{for } a \in T \cup \bar{T},$$

$$\left. \begin{aligned} g(c_i) &= 01^i g \\ g(\bar{c}_i) &= \bar{01}^i \end{aligned} \right\} \quad \text{for } i = 1, 2, \dots, m.$$

Finally, let $R = g(Q)$. Since g is a one-to-one mapping, since $g(L(\Gamma)) = L(T \cup \{0, 1\}) \cap (g(\Gamma \cup \bar{\Gamma}))^*$ and since $\Pi_T(x) = \Pi_T(g(x))$ for every $x \in (\Gamma \cup \bar{\Gamma})^*$, we have

$$L = \Pi_T(L(\Gamma) \cap Q) = \Pi_T(g(L(\Gamma) \cap g(Q))) = \Pi_T(L(T \cup \{0, 1\}) \cap R). \quad \square$$

We establish that representation theorems such as theorem 1 are not restricted to a few special language families but hold for a variety of "natural" families of languages.

THEOREM 2: *Let T be an alphabet and \mathcal{L} a full principal AFL. There exists a language L_T in \mathcal{L} such that for each L in \mathcal{L} , $L \subseteq T^*$, there exists a regular set R_L such that $L = \Pi_T(L_T \cap R_L)$ (Π_T is an erasing).*

Proof: Since \mathcal{L} is a full principal AFL, there is L_0 in \mathcal{L} such that \mathcal{L} is the closure of $\{L_0\}$ under finite transducers (rational relations), cf. [7], that is for each L in \mathcal{L} , $L \subseteq T^*$, we can write $L = t_L(L_0)$ where t_L is a finite transducer. We may assume that $L_0 \subseteq \Sigma^*$, $\Sigma \cap T = \emptyset$. Using [5], thm. 2.5, there is $R \subseteq (T \cup \Sigma)^*$ such that $L = \Pi_T(\Pi_\Sigma^{-1}(L_0) \cap R)$. Now, let $L_T = \Pi_\Sigma^{-1}(L_0)$. Then, for all L in \mathcal{L} , $L \subseteq T^*$, there is $R \subseteq (\Sigma \cup T)^*$ such that $L = \Pi_T(L_T \cap R)$. Note, that L_T and Π_T depends only on T and not on L . \square

We conclude this subsection by observing that a result akin to theorem 2 holds for any countable family of languages \mathcal{L} , provided we do not insist that the generator is element of \mathcal{L} :

THEOREM 3: *Let Σ be an alphabet and \mathcal{L} a countable family of languages over (subsets of) Σ . Then there exists a language U such that for each $L \in \mathcal{L}$ there exist an erasing Π_T and a regular set R such that $L = \Pi_T(U \cap R)$.*

Proof: Let $\mathcal{L} = \{L_1, L_2, L_3, \dots\}$ and let c, d be new symbols. Let $U = \bigcup_{i=1}^{\infty} c^i d L_i$. Suppose $L = L_i \in \mathcal{L}$, $L \subseteq T^*$. Define $R = c^i d T^*$. Clearly, $L = \Pi_T(U \cap R)$. \square

Note that by restricting the choice of T in the erasing Π_T we can get a precise characterization, i. e., only languages in \mathcal{L} , even if U is not in \mathcal{L} .

2.2. Generation using inverse homomorphism, possibly followed by homomorphism

Greibach's result on the "hardest" CF language, *see* [8], asserts that every CF language can be obtained as inverse homomorphic image of one fixed CF language:

P_4 : There exists a CF language U such that for each CF language L there exists a homomorphism h such that $L = h^{-1}(U)$,

We show that a result akin to P_4 holds for RE languages: every RE language L can be obtained from some fixed "simple" RE language U by some inverse homomorphism followed by an erasing (thm. 4). Indeed, every RE set L can be generated from some fixed RE set U by just an inverse homomorphism by using as U an encoding of all possible RE languages (thm. 5). We then modify the proof of theorem 5 and obtain (thm. 6) that the family of context-sensitive languages can be obtained from a single context-sensitive language in the same way. This theorem also provides an alternative simple proof that the family of context-sensitive languages is a principal AFL. We conclude the paper by mentioning that such purely homomorphic characterizations are impossible for the class of regular languages (thm. 6).

THEOREM 4: *There exists a fixed RE language $U \subseteq \{0, 1\}^*$ such that for every RE language L there is a homomorphism h and an erasing Π_T such that $L = \Pi_T(h^{-1}(U))$.*

Proof: Assume $L \subseteq T^*$. By theorem 1 in [3] we can write $L = \Pi_T(e(h_1, h_2))$ for some homomorphisms $h_1, h_2: \Sigma^* \rightarrow \Delta^*$ and $T \subseteq \Sigma$.

Let $\Delta = \{c_1, c_2, \dots, c_m\}$ and $g_1, g_2: \Delta^* \rightarrow \{0, 1\}^*$ be homomorphisms defined by $g_1(c_i) = 01^i, g_2(c_i) = 001^i$, for $i = 1, 2, \dots, m$. That is, g_1 and g_2 encode an arbitrary alphabet Δ into the binary alphabet $\{0, 1\}$. Observe that both g_1 and g_2 are one-to-one functions.

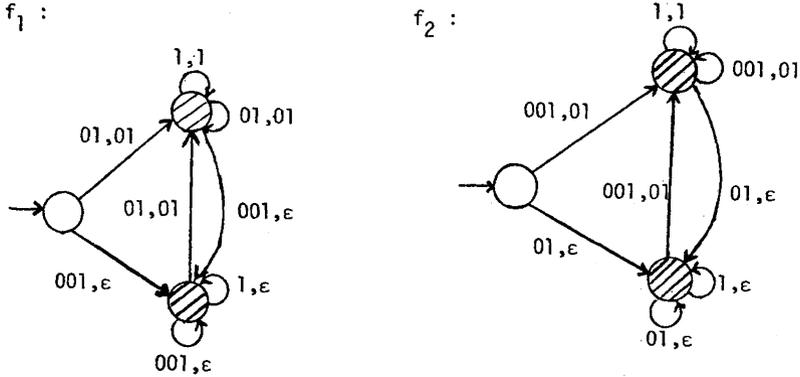
Let f_1, f_2 be finite transducers defined by their diagrams in the figure (shaded circles indicate final states).

Using f_1 and f_2 we now define our generator U . Let

$$U = \{w \in \{0, 1\}^* \mid f_1(w) = f_2(w) \neq \emptyset \\ \text{and } f_1(v) \neq f_2(v) \text{ for each proper prefix } v \text{ of } w\}.$$

Note that U is defined independently of Δ . However, for each Δ we have:

$$U \cap (g_1(\Delta) \cup g_2(\Delta))^* = \{w \in (g_1(\Delta) \cup g_2(\Delta))^* \mid g_1^{-1}(w) = g_2^{-1}(w) \\ \text{and } g_1^{-1}(v) \neq g_2^{-1}(v) \text{ for each proper prefix } v \text{ of } w\}. \quad (1)$$



Finally, let $h: \Sigma^* \rightarrow \{0, 1\}^*$ be the homomorphism defined by $h(a) = g_1(h_1(a))$ for each $a \in \Sigma$. It follows from (1) that $e(h_1, h_2) = h^{-1}(U \cap (g_1(\Delta) \cup g_2(\Delta))^*) = h^{-1}(U)$. Hence $L = \Pi_T(h^{-1}(U))$ as desired. \square

By coding all RE languages into one (complicated) RE language U , every RE language L can be obtained from U by a single inverse homomorphism h^{-1} , h^{-1} in essence “retrieving” L from U . We present this method already mentioned in [1] for completeness sake and as a basis for theorem 6.

THEOREM 5: *There exists an RE language $U \subseteq \{0, 1\}^*$ such that every RE language L can be written as $L = h_L^{-1}(U)$ for some homomorphism h_L .*

Proof: We assume that each RE language is over some finite subset of an infinite alphabet $\Sigma = \{a_1, a_2, a_3, \dots\}$. RE languages are generated by type 0 grammars. Consider a fixed encoding of type 0 grammars (similar as described for CS grammars in [10], p. 118) such that a_i is encoded as 01^i and all other symbols (including nonterminals) are encoded as 001^i for $i = 1, 2, \dots$. Let G_1, G_2, G_3, \dots be an effective enumeration of encodings of all type zero grammars (we will identify a grammar with its encoding), $G_i \in \{0, 1\}^*$ for $i = 1, 2, \dots$. Let for each $i = 1, 2, \dots$ T_i be the terminal alphabet (subset of Σ) of G_i and let h_i be the homomorphism from T_i^* to $\{0, 1\}^*$ defined by $h_i(a_j) = 00G_i0001^j$ for each $a_j \in T_i$. Finally, we define our generator U as $U = \bigcup_{i=1}^{\infty} h_i(L(G_i))$. Informally, U is the union of all the languages generated by type 0 grammars G_1, G_2, \dots where in every string from $L(G_i)$ every symbol is preceded by the encoding of G_i .

U is an RE set by showing that U can be generated by a type 0 grammar G . Roughly speaking, G works in 4 stages. In stage 1, G generates an arbitrary word which, if meeting certain format restrictions, will be interpreted as the encoding of some grammar H . Stage 2 checks whether the word generated in the

first stage is indeed the encoding of a type 0 grammar. In stage 3, derivations of H are simulated. In stage 4, a "signature" $00H00$ of the grammar H is generated and it is inserted before each terminal symbol.

It is easy to see that for each $i=1, 2, \dots$ $L(G_i)=h_i^{-1}(U)$. The inverse homomorphism h_i selects from U exactly the words of $h_i(L(G_i))$ and decodes them into $L(G_i)$. Since every *RE* language is generated by some G_j we have completed the proof.

We do not know whether for a much simpler U (such as the U of theorem 4) theorem 5 also holds \square .

A theorem analogous to theorem 5 also holds for context sensitive languages. This can be deduced from [14] but can also be obtained by a simple modification of the above proof as follows.

THEOREM 6: *There exists a context sensitive language $U \subseteq \{0, 1\}^*$ such that every context sensitive language L can be written as $L=h_L^{-1}(U)$, for some homomorphism h_L .*

Proof: We construct a generator U as in the proof of theorem 5 except that we have to use monotonic grammars rather than type 0 grammars. This can be done as follows:

For each context sensitive grammar G_i the encoding of terminals and nonterminals as words over $\{0, 1\}$ is done using words of the same length. Suppose the terminal alphabet of G_i is $\Sigma_G = \{a_1, a_2, \dots, a_m\}$ with $j = \max \{i_t \mid t=1, 2, \dots, m\}$. Suppose further that G_i has the nonterminal alphabet $\Sigma_N = \{A_1, A_2, \dots, A_k\}$. Rather than encoding a_i as 01^i and A_t as 001^t as we have done for type 0 grammars in the proof of theorem 5, we now encode a_i as $01^i 00001^{q-i}$ and A_t as $001^t 00001^{q-t-1}$, where $q=2+\max \{j, k\}$. Note that in this way the encoding of each symbol of G_i is a word of length $q+5$, hence the encoding of monotonic productions gives rise to monotonic productions. Instead of defining $h_i(a_i)=00 G_i 0001^i$ for each terminal a_i of G_i we now define $h_i(a_i)=00 G_i 0001^i 00001^{q-i}$, and again define

$$U = \bigcup_{i=1}^{\infty} h_i(L(G_i)).$$

That U is a context sensitive language is seen by considering the grammar G generating U as described in the proof of theorem 5. Clearly, stages 1, 2 and 4 can be carried out by monotonic productions. Stage 3 can also be carried out using monotonic productions since the productions of H to be simulated are monotonic by the above. Thus G is a context sensitive grammar as desired. \square

We conclude this paper by mentioning that a strictly homomorphic characterization of regular sets is not possible. An auxiliary result turns out to be useful.

LEMMA 2: Let R be a regular set, $R = T(A)$, A a finite automaton with n states. Let h be a homomorphism. Then $R' = h^{-1}(R)$ can be accepted by a finite automaton A' with n states.

Proof: Let $R \subseteq \Sigma^*$, $A = (\Phi, \Sigma, \delta, q_0, F)$, Φ the set of states, Σ the input alphabet, $\delta \subseteq \Phi \times \Sigma \times \Phi$ the transition function, $q_0 \in \Phi$ the start state, $F \subseteq \Phi$ the set of final states. Let $h: \Sigma' \rightarrow \Sigma^*$. Define $A' = (\Phi, \Sigma', \delta', q_0, F)$ as follows: $\delta'(q, a') = \delta(q, h(a'))$. We maintain: $R' = T(A')$. Let $x = a'_1 a'_2 \dots a'_m$ ($a'_i \in \Sigma'$ for $1 \leq i \leq m$) be an arbitrary word over Σ' .

Part 1: ($R' \subseteq T(A')$).

Suppose $x \in R'$. Then $h(x) = h(a'_1) h(a'_2) \dots h(a'_m) \in R$. Define $q_i = \delta(q_0, h(a'_1) \dots h(a'_i))$. Then $q_m \in F$. We now show $\delta'(q_0, a'_1 \dots a'_i) = q_i$, hence $\delta'(q_0, a_1 \dots a_m) \in F$, i.e. $x \in T(A')$:

$$\begin{aligned} \delta'(q_0, a'_1) &= \delta(q_0, h(a_1)) = q_1, \\ \delta'(q_0, a'_1 \dots a'_{i+1}) &= \delta'(q_i, a'_{i+1}) = \delta(q_i, h(a'_{i+1})) \\ &= \delta(q_0, h(a_1) \dots h(a'_{i+1})) = q_{i+1}. \end{aligned}$$

Part 2: ($T(A') \subseteq R'$).

Suppose $x \in T(A')$. Then $\delta'(q_0, a'_1 \dots a'_m) \in F$. Define $\delta'(q_0, a'_1 \dots a'_i) = q_i$. We will show (inductively) that $\delta(q_0, h(a'_1) \dots h(a'_i)) = q_i$. Hence $\delta(q_0, h(a'_1) \dots h(a'_m)) \in F$, i.e. $h(x) \in T(A)$, i.e. $h(x) \in R$ and thus $x \in R' = h^{-1}(R)$. Clearly, $\delta(q_0, (a'_1)) = \delta'(q_0, a'_1) = q_1$. Further,

$$\begin{aligned} \delta(q_0, h(a'_1) \dots h(a'_{i+1})) &= \delta(q_i, h(a'_{i+1})) = \delta'(q_i, a'_{i+1}) \\ &= \delta'(q_0, a'_1 \dots a'_i a'_{i+1}) = q_{i+1}. \quad \square \end{aligned}$$

THEOREM 7: For every regular set R there exists a regular set R' such that $R' \neq g(h^{-1}(R))$ holds for all homomorphisms g and h .

Proof: Suppose R is accepted by a finite automaton with n states. Then $h^{-1}(R)$ is also accepted by a finite automaton of n states by lemma 2. Hence $h^{-1}(R)$ is a regular language of star height $\leq n$. Choose R' to be any regular language of star height $> n$. (Such R' is known to exist, cf. [13].) Since homomorphisms do evidently not increase the star height, $g(h^{-1}(R))$ is of star height $\leq n$. Hence $R' \neq g(h^{-1}(R))$. \square

We have shown that both the class of *RE* languages and of *CF* languages can be generated by a single fixed *RE* language, *CF* language respectively, by just using inverse homomorphisms. The family of regular languages (as a subclass of the family of *CF* languages) can certainly be generated under inverse homomorphisms from a *CF* language L (by P_4), but L must be nonregular by theorem 7.

The question arises whether other language families, for instance the family of *ETOL* languages, do have inverse homomorphic representations. We feel that Greibach's proof of P_4 can be carried over to *ETOL* languages, if the following normal form theorem holds for *ETOL* languages:

A_1 : For every *ETOL* language L there exists an *ETOL* system G generating L such that each production is either of the form:

- (i) $\alpha \rightarrow x$, where x is a terminal word, or
- (ii) $\alpha \rightarrow A_a Y$, where A_a is a nonterminal whose only productions are $A_a \rightarrow A_a$ and $A_a \rightarrow a$ (Y is arbitrary) or
- (iii) $a \rightarrow N$, $N \rightarrow N$ where a is a terminal, N is a "blocking" nonterminal.

We do not know whether assertion A_1 holds. In view of the difficulty of proving a somewhat similar normal form result in [4], a proof of A_1 does not seem to be easy.

REFERENCES

1. R. V. BOOK, *Comparing Complexity Classes*, J. Comput. System Sc., Vol. 9, 1974, pp. 213-229.
2. K. CULIK II, *On Some Families of Languages Related to Developmental Systems*, Intern. J. Comput. Math., Vol. 4, 1974, pp. 31-42.
3. K. CULIK II, *A Purely Homomorphic Characterization of Recursive Enumerable Sets*, J. Assoc. Comput. Mach., Vol. 26, 1979, pp. 345-350.
4. K. CULIK II and H. A. MAURER, *Propagating Chain-Free Normal Forms for EOL Systems*, Information and Control, Vol. 36, 1978, pp. 309-319.
5. S. EILENBERG, *Automata, Languages and Machines*, Vol. A, Academic Press, 1974.
6. J. ENGELFRIET and G. ROZENBERG, *Fixed Point Languages, Equality Languages and Representations of Recursively Enumerable Languages*, 19th Annual Symposium on Foundations of Computer Science, 1978, pp. 123-126.
7. S. GINSBURG, *Algebraic and Automata-Theoretic Properties of Formal Languages*, North Holland, Amsterdam, 1975.
8. S. GREIBACH, *The Hardest CF Language*, S.I.A.M. J. Comput., Vol. 2, 1973, pp. 304-310.
9. M. A. HARRISON, *Introduction to Formal Language Theory*, Addison-Wesley, Reading, Mass., 1978.
10. J. E. HOPCROFT and J. D. ULLMAN, *Formal Languages and Their Relation to Automata*, Addison-Wesley, Reading, Mass., 1969.
11. H. A. MAURER, *Theoretische Grundlagen der Programmiersprachen*, BI, Mannheim, 1969.
12. A. SALOMAA, *Formal Languages*, Academic Press, New York, 1973.
13. A. SALOMAA, *Theory of Automata*, Pergamon Press, London, 1969.
14. B. WEGBREIT, *A Generator of Context Sensitive Languages*, J. Comput. System Sc., Vol. 3, 1969, pp. 456-461.