ON SIMPLE REPRESENTATIONS
OF LANGUAGE FAMILIES*

by
K. Culik II[1]
and
H.A. Maurer[2]

Research Report CS-78-41

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

Faculty

of

Mathematics

University of Waterloo
Waterloo, Ontario, Canada

Abstract

In this paper we establish representation results for families of languages analogous to the Chomsky-Schützenberger theorem for CF languages and analogous to Greibach's theorem on the hardest CF language. We show that, using intersection with regular sets and certain simple homomorphisms, the family of RE 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 CF language theorem to RE 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 establish that no such result is possible for the family of regular languages.

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.

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 (Theorem 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 (Theorems 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 (Theorem 6). Not only is this the strongest representation theorem for context-sensitive languages known to date, it also gives a particularly simple proof that the family of context-sensitive languages is a principal AFL, a result originally obtained in [12], c.f. also [5, p. 139]. We finally establish that no such theorem can hold for the family of regular languages (Theorem 7).

Throughout the paper we assume familiarity with basic formal language theory. For any terminology not explained in this paper [7-10] 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 $h : \Sigma^* \to \Delta^*$ is called an erasing if for some subset T of Σ we have $h(a) = a$ iff $a \in T$ 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, $h_1, h_2 : \Sigma^* \to \Delta^*$. 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, $\overline{\Sigma}$ will denote an alphabet disjoint from Σ consisting of "barred" symbols, $\overline{\Sigma} = \{\overline{a} \mid a \in \Sigma\}$. For any word $x \in \Sigma$, $\overline{x}$ denotes the word obtained from x by barring each symbol.

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

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

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 $\Phi \times \Sigma^* \times \Phi \times \Sigma^*$, 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 <u>simple</u> 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:

> $\underline{P_1}$: 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:

> $\underline{P_1'}$: Let T be an arbitrary alphabet. There exists a language L_T such that for every CF language L 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 [1] for both EOL and ETOL languages:

$\underline{P_2}$: 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 [4]:

$\underline{P_3}$: For every RE language L there exist a twinshuffle D_L , a regular set R and an erasing Π such that $L = \Pi(D_L \cap R)$.

In what follows we first present an alternate proof of P_3 (Lemma 1) based on a result in [2]. A modification thereof shows that the language D_L in P_3 can be chosen to depend only on the alphabet of L (Theorem 1), a strengthening analogous to P_1' . 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 (Theorem 2). We finally observe that a similar result holds for any countable family of languages (Theorem 3) but that the generator used then will, in general, not be a language of the family of languages at issue.

<u>Lemma 1</u>

For every RE language $L \subseteq T^*$ there exist a twinshuffle L_Γ , a regular set R and an erasing Π_T such that $L = \Pi_T(L_\Gamma \cap R)$.

<u>Proof</u>: By Theorem 1 of [2] we can write $L = \Pi_T(e(h_1, h_2))$ for some homomorphisms $h_1, h_2 : \Sigma^* \to \Delta^*$ and $T \subseteq \Sigma$. We may assume that

$\Delta \cap \Sigma = \phi$. Moreover, it follows from the proof of the theorem that we may assume that a symbol 3 is element of Σ and $e(h_1, h_2) \subseteq (\Sigma - \{3\})^*\{3\}$.

Let $\overline{\Sigma} = \{\overline{a} \mid a \in \Sigma\}$, $\overline{\Delta} = \{\overline{b} \mid b \in \Delta\}$, $\Gamma = \Sigma \cup \Delta$, $\overline{\Gamma} = \overline{\Sigma} \cup \overline{\Delta}$ and let $\overline{w}$ be the word obtained from a word w by barring each symbol, $w \in (\Sigma \cup \Delta)^*$. Let $F = \{ah_1(a)h_2(a) \mid a \in \Sigma\}$ and let $R = (F \cup \overline{\Sigma} - \{\overline{3}\})^*\{\overline{3}\}$.

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

□

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 $R \subseteq (T \cup \overline{T} \cup \{0, \overline{0}, 1, \overline{1}\})^*$ 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 $\Gamma - T = \{c_1, c_2, \ldots, c_m\}$ and let $g : (\Gamma \cup \overline{\Gamma})^* \to T \cup \overline{T} \cup \{0, 1, \overline{0}, \overline{1}\}$ be the homomorphism defined by:

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

$$\left.\begin{array}{l} g(c_i) = 01^i \\ g(\overline{c}_i) = \overline{0}\overline{1}^i \end{array}\right\} \quad \text{for } i = 1,2,\ldots,m \ .$$

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

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

□

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 Σ be an alphabet and $\mathcal{L}$ a full principal AFL. There exists a language $L_\Sigma \in \mathcal{L}$ such that for each $L \in \mathcal{L}$, L over some alphabet $T \subseteq \Sigma$, there exist an erasing Π_T and a regular set R such that $L = \Pi_T(L_\Sigma \cap R)$.

Proof: Since $\mathcal{L}$ is a full principal AFL, $\mathcal{L}$ is the transducer closure of some language $L_0 \in \mathcal{L}$, c.f. [5]. Thus, every $L \in \mathcal{L}$, L over some alphabet $T \subseteq \Sigma$, can be written as $L = t(L_0)$, where t is a finite transducer and may be assumed to be simple. Equivalently, L can be written as $L = f(Q \cap g^{-1}(L_0))$, where $Q \subseteq \Phi^*$ is a regular set, g is a homomorphism $g : \Phi^* \to \Delta^*$, f is a homomorphism $f : \Phi^* \to T^*$, and Δ is the alphabet of L_0 .

For our further observations it is of crucial importance that the alphabet Φ can be assumed to depend only on T and Δ . (Applying the "quadruple-construction", as is often done when proving results on transducers, would result in Φ also depending on the number of states of the transduction involved.) We show that $\Phi = (T \cup \{\varepsilon\}) \times (\Delta \cup \{\varepsilon\})$ can be assumed by using the following approach. Let $t = (\Delta, T, \Gamma, M, q_0, F)$ be a simple finite transducer with states Γ , input alphabet Δ , output alphabet T , $q_0 \in \Gamma$ start state, $F \subseteq \Gamma$ final states and $M \subseteq \Gamma \times (\Delta \cup \{\varepsilon\}) \times \Gamma \times (T \cup \{\varepsilon\})$ describing the behaviour of T .

Let $\Phi = \{[\alpha, \beta] \mid \alpha \in \Delta \cup \{\varepsilon\} , \beta \in T \cup \{\varepsilon\}\}$ and let $A = (\Phi, \Gamma, \delta, q_0, F)$ be the nondeterministic finite automaton with states Γ and input alphabet Φ whose transition function δ is defined by $\delta(p, [\alpha, \beta]) = \{q \mid (p, \alpha, q, \beta) \in \Gamma\}$. Let Q be the regular set defined by A . Defining $g : \Phi^* \to \Delta^*$ by $g([\alpha, \beta]) = \alpha$ for $[\alpha, \beta] \in \Phi$ and $f([\alpha, \beta]) = \beta$ for $[\alpha, \beta] \in \Phi$ it is easy to see that $L = f(Q \cap g^{-1}(L_0))$, as desired.

Since $|f(a)| \leq 1$ for $a \in \Phi$ and $|g(a)| \leq 1$ for $a \in \Phi$, since Φ depends only on T and Δ , $T \subseteq \Sigma$, and since Σ and Δ are fixed alphabets, every $L \in \mathcal{L}$ with $L \subseteq \Sigma^*$ can be written in above form with $f = f_i$ in F and $g = g_j$ in G , where $F = \{f_1, f_2, \ldots, f_m\}$ and $G = \{g_1, g_2, \ldots, g_n\}$ are finite sets of homomorphisms.

Let $\overline{\Phi} = \{\overline{a} \mid a \in \Phi\}$, $\overline{G} = \{\overline{g}_1, \overline{g}_2, \ldots, \overline{g}_n\}$ and $\overline{F} = \{\overline{f}_1, \overline{f}_2, \ldots, \overline{f}_m\}$ be sets of new symbols. For each k $(1 \leq k \leq m)$

define $h_k(a) = \bar{a}f_k(a)$ for all $a \in \Phi$. Let $L_\Sigma = \bigcup_{\substack{1 \le k \le m \\ 1 \le \ell \le n}} \bar{f}_k \cdot \bar{g}_\ell \cdot h_k(g_\ell^{-1}(L_o))$ and $R = \bar{f}_i \bar{g}_j h_i(Q)$.

Then $L = f_i(g_j^{-1}(L_o) \cap Q)$ clearly implies $L = \Pi_T(L_\Sigma \cap R)$, where Π_T is an erasing. Since L_Σ has been obtained from L_o by AFL operations, $L_\Sigma \in \mathcal{L}$ holds, as desired.

□

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 $\mathcal{L}$:

<u>Theorem 3</u>

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 $L = \Pi_T(U \cap R)$.

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

□

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 [6], asserts that every CF language can be obtained as inverse homomorphic image of one fixed CF language:

$\underline{P_4}$: 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 (Theorem 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 (Theorem 5). We then modify the proof of Theorem 5 and obtain (Theorem 6) that the family of context-sensitive languages can be obtained from a single context-sensitive language in the same way. This new representation theorem also provides an alternative simple proof that the family of context-sensitive languages is a principal AFL. We conclude the paper by showing that such purely homomorphic characterizations are impossible for the class of regular languages (Theorem 6).

Theorem 4

There exists a fixed 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 [2] we can write $L = \Pi_T(e(h_1, h_2))$ for some homomorphisms $h_1, h_2 : \Sigma^* \to \Delta^*$ and $T \subseteq \Sigma$.

Let $\Delta = \{c_1, c_2, \ldots, c_m\}$ and $g_1, g_2 : \Delta^* \to \{0, 1\}^*$ be homomorphisms defined by $g_1(c_i) = 01^i$, $g_2(c_i) = 001^i$, for $i = 1,2,\ldots,m$. 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 Figure 1. (Shaded circles indicate final states.)

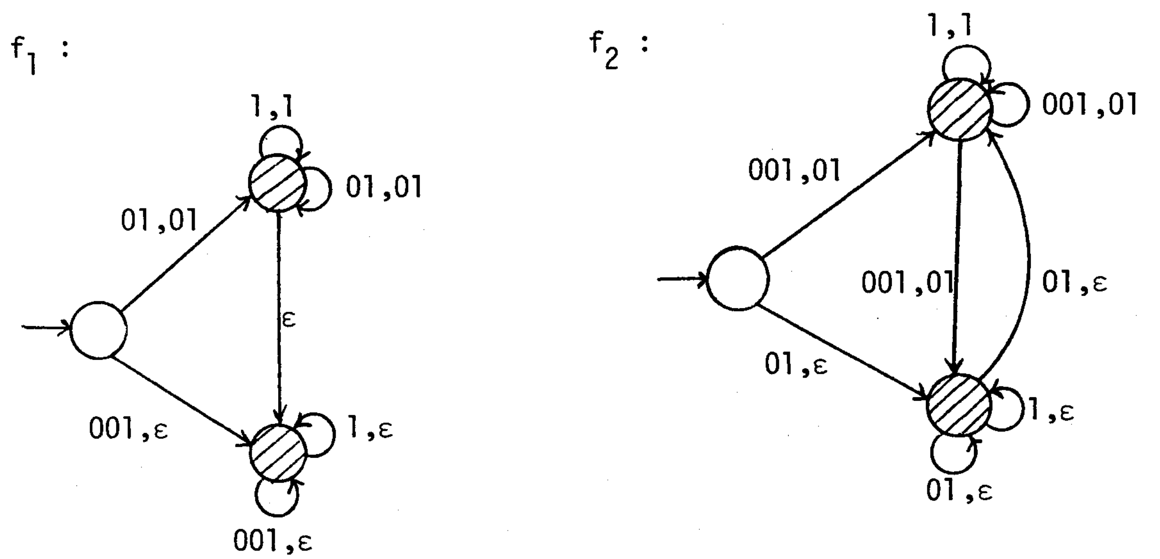

Figure 1

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

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

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

$$U \cap (g_1(\Delta) \cup g_2(\Delta))^*$$

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

Finally, let $h : \Sigma^* \to \{0, 1\}^*$ be the homomorphism defined by $h(a) = g_1(h_1(a))g_2(h_2(a))$ 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.

□

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.

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 $\Sigma = \{a_1, a_2, a_3, \ldots\}$. RE languages are generated by type 0 grammars. Consider a fixed encoding of type 0 grammars (similar as described for CS grammars in [8, p. 118]) such that a_i is encoded as 01^i and all other symbols (including nonterminals) are encoded as 001^i for $i = 1,2,\ldots$. Let $G_1, G_2, G_3, \ldots$ 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 $i = 1,2,\ldots$. Let for each $i = 1,2,\ldots$ 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 $G_1, G_2, \ldots$ where in every string from $L(G_i)$ every symbol is preceeded 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 $i = 1,2,\ldots$ $L(G_i) = h_i^{-1}(U)$. 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.

□

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 the generator U as in the proof of Theorem 5 except that we use monotonic grammars rather than type 0 grammars and we modify the encodings of grammars so that in the encoding of each terminal and nonterminal symbol (grammar dependent) we add from the right side a string 00001^k where k's are chosen so that all symbols of a given grammar have encodings of equal length. We correspondingly modify the homomorphism h_i for each i. This modification assures that a monotonic grammar can be constructed for the generator U. This grammar is constructed in the same way as outlined in the proof of Theorem 5.

□

We conclude this paper by showing 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 $h : \Sigma' \to \Sigma^*$. Define $A' = (\Phi, \Sigma', \delta', q_o, F)$ as follows: $\delta'(q, a') = \delta(q, h(a'))$. We maintain: $R' = T(A')$. Let $x = a_1'a_2'...a_m'$ $(a_i' \in \Sigma'$ for $1 \leq i \leq m)$ be an arbitrary word over Σ' .

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

Suppose $x \in R'$. Then $h(x) = h(a_1')h(a_2')...h(a_m') \in R$. Define $q_i = \delta(q_o, h(a_1')...h(a_i'))$. Then $q_m \in F$. We now show $\delta'(q_o, a_1'...a_i') = q_i$, hence $\delta'(q_o, a_1...a_m) \in F$, i.e. $x \in T(A')$:

$$\delta'(q_o, a_1') = \delta(q_o, h(a_i)) = q_1 .$$

$$\delta'(q_o, a_1'...a_{i+1}') = \delta'(q_i, a_{i+1}') = \delta(q_i, h(a_{i+1}'))$$
$$= \delta(q_o, h(a_1)...h(a_{i+1}')) = q_{i+1} .$$

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

Suppose $x \in T(A')$. Then $\delta'(q_o, a_1'...a_m') \in F$. Define $\delta'(q_o, a_1'...a_i') = q_i$. We will show (inductively) that $\delta(q_o, h(a_1')...h(a_i')) = q_i$. Hence $\delta(q_o, h(a_1')...h(a_m')) \in F$, i.e. $h(x) \in T(A)$, i.e. $h(x) \in R$ and thus $x \in R' = h^{-1}(R)$. Clearly, $\delta(q_o, h(a_1')) = \delta'(q_o, a_1') = q_1$. Further,

$$\delta(q_o, h(a_1')...h(a_{i+1}')) = \delta(q_i, h(a_{i+1}')) = \delta'(q_i, a_{i+1}')$$
$$= \delta'(q_o, a_1'...a_i'a_{i+1}')$$
$$= q_{i+1} .$$

□

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 $\leq n$. Chose R' to be any regular language of star height $> n$. (Such R' is known to exist, c.f. [11].) Since homomorphisms do evidently not increase the star height, $g(h^{-1}(R))$ is of star height $\leq n$. Hence $R' \neq g(h^{-1}(R))$.

□

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 6.

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:

$\underline{A_1}$: For every ETOL language L there exists an ETOL system G generating L such that each production is either of the form:

(i) $\alpha \to x$, where x is a terminal word, or

(ii) $\alpha \to A_a Y$, where A_a is a nonterminal whose only productions are $A_a \to A_a$ and $A_a \to a$ (Y is arbitrary) or

(iii) $a \to N$, $N \to N$ 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 [3],a proof of A_1 does not seem to be easy.

References:

1. K. Culik II, On some families of languages related to developmental systems, Intern. J. Computer Math 4, 1974, pp. 31- 42.

2. K. Culik II, A purely homomorphic characterization of recursive enumerable sets, J. ACM (to appear).

3. K. Culik II and H.A. Maurer, Propagating chain-free normal forms for EOL systems, Information and Control 36, 1978, 309-319.

4. J. Engelfriet, G. Rozenberg, Fixed point languages, equality languages and representations of recursively enumerable languages, 19th Annual Symposium on Foundations of Computer Science, 1978.

5. S. Ginsburg, Algebraic and automata - theoretic properties of formal languages, North Holland, Amsterdam, 1975.

6. S. Greibach, The hardest CF language, SIAM Journal of Computing 2, 1973, 304-310.

7. M.A. Harrison, Introduction to formal language theory, Addison-Wesley, Reading, Mass, 1978.

8. J.E. Hopcroft and J.D. Ullman, Formal languages and their relation to automata. Addison-Wesley, Reading, Mass., 1969.

9. H.A. Maurer, Theoretische grundlagen der programmiersprachen, BI Mannheim, 1969.

10. A. Salomaa, Formal languages, Academic Press, New York, 1973.

11. A. Salomaa, Theory of automata, Pergamon Press, London, 1969.

12. B. Wegbreit, A generator of context sensitive languages, J. of Computers and Systems Sciences 3, 1969, 456-461.