BULL. MATH. de la Soc. Sci. Math. de la R.S. de Roumanie
Tome 33 (81), nr. 1, 1989

# NUMBER-THEORETIC MAPPINGS COMPUTED BY G.S.M.'S

by

GHEORGHE PĂUN AND LILA SÂNTEAN (ROMÂNIA)

Encoding the natural numbers as strings over some suitable alphabet, the natural mappings can be viewed as string mappings and can be simulated by string transducers. The mappings computed in this way by *gsm*'s are considered here: examples, necessary and sufficient conditions, characterization, infinite hierarchies with respect to various syntactic complexity measures of *gsm*'s etc.

## 1. Introduction

The syntactic treatment of numerical mappings, by encoding them as string mappings, is an already considered topic in formal language theory (see [5], [7], [8], [9], for instance). The reason is twofold: to test the power of various generative mechanisms in formal language theory and to say something about numerical mappings through this unstandard way of studying them.

The present paper continues this path of research, by investigating mappings computed by sequential transducers. Only linearly bounded natural mappings can be obtained in this way, but the study is still worth: infinite complexity hierarchies of such mappings are obtained by considering the syntactic complexity of associated transducers. Please note that the class of linear mappings was not generally splitted into complexity classes in the frame of recursive function theory (see [1]).

## 2. Definitions and examples

A *gsm* is a system (we use the style of [6] in writing its components)

$$g = (I,\ O,\ K,\ s_0,\ F,\ P)$$

where $I$ is the input alphabet, $O$ is the output alphabet, $K$ is the (finite and nonempty) set of states, $s_0 \in K$ is the initial state, $F \subseteq K$ is the set of final states, and $P$ is a finite set of rewriting rules of the form $sa \to ws'$,

$s, s' \in K$, $a \in I$, $w \in O^*$ (for a given alphabet $V$, $V^*$ is the free monoid generated by $V$ under the operation of concatenation and the null element $\lambda$; $|x|$ denotes the length of a string $x$ and $|x|_a$ is the number of occurrences of the symbol $a$ in the string $x$).

For two strings $x, y$, $x = x_1 s a x_2$, $y = x_1 w s' x_2$, $x_1 \in O^*$, $x_2 \in I^*$, $sa \to \to ws' \in P$, we write $x \Rightarrow y$ (in the state $s$, the *gsm* reads the symbol $a$, translates it into the string $w$ and passes to state $s'$). We denote by $\overset{*}{\Rightarrow}$ the reflexive and transitive closure of $\Rightarrow$. For a string $x \in I^*$ we define

$$g(x) = \{y \in O^* \mid x \overset{*}{\Rightarrow} y \text{ according to } g\}$$

For a language $L \subseteq I^*$ we put

$$g(L) = \bigcup_{x \in L} g(x)$$

Given a (partial) mapping $f : \mathbf{N}^k \to \mathbf{N}$ we can consider the (partial string) mapping

$$f_s : \{a_1, \ldots, a_k\}^* \to \{a\}^*$$

defined by

$$f_s(a_1^{n_1} \ldots a_k^{n_k}) = a^m, \; m = f(n_1, \ldots, n_k),$$

for all $(n_1, \ldots, n_k) \in dom\, f$. A device which would compute the mapping $f_s$ be said to compute $f$.

In this way we can say that a mapping $f : \mathbf{N}^k \to \mathbf{N}$ is computed by a *gsm* $g = (\{a_1, \ldots, a_k\}, \{a\}, K, s_0, F, P)$ if $g(a_1^{n_1} \ldots a_k^{n_k}) = a^m$, $m = f(n_1, \ldots, n_k)$ for $(n_1, \ldots, n_k) \in dom\, f - \{(0, \ldots, 0)\}$ and $g(a_1^{n_1} \ldots a_k^{n_k})$ is not defined for $(n_1, \ldots \ldots, n_n) \notin dom\, f$. (We have to omit the argument $(0, \ldots, 0)$ because we have $g(\lambda) = \lambda$, hence the mappings value in origin cannot be computed.) When a *gsm* $g$ computes a mapping $f$ we write $g \equiv f$.

Please note that we are interested only in the behavior of $g$ over $a_1^* \ldots a_k^*$ and we ignore its behavior outside this language.

We denote

$$\mathcal{G}_k = \{f : \mathbf{N}^k \to \mathbf{N} \mid \text{there is a } gsm \; g \text{ such that } g \equiv f\}$$

$$\mathcal{G} = \bigcup_{k \geq 1} \mathcal{G}_k$$

*Examples.* (1) The mapping $f(n_1, \ldots, n_k) = \sum_{i=1}^{k} \alpha_i n_i + \beta$, $\alpha_i \in \mathbf{N}$, $1 \leq i \leq k$, $\beta > 0$, is computed by

$$g = (\{a_1, \ldots, a_k\}, \{a\}, \{s_0, s_1\}, s_0, \{s_1\}, P)$$

$$P = \{s_0 a_i \to a^{\alpha_i + \beta} s_1 \mid 1 \leq i \leq k\} \cup$$

$$\cup \{s_1 a_i \to a^{\alpha_i} s_1 \mid 1 \leq i \leq k\}$$

(2) The mapping $f(n_1, \ldots, n_k) = \sum_{i=1}^{k} n_i \pmod 3$ is computed by

$$g = (\{a_1, \ldots, a_k\}, \{a\}, \{s_0, s_1, s_2, s_3\}, s_0, \{s_3\}, P)$$
$$P = \{s_0 a_i \to s_1 \mid 1 \leqslant i \leqslant k\} \cup \{s_1 a_i \to s_2 \mid 1 \leqslant i \leqslant k\} \cup$$
$$\cup \{s_2 a_i \to s_0 \mid 1 \leqslant i \leqslant k\} \cup \{s_0 a_i \to a s_3 \mid 1 \leqslant i \leqslant k\} \cup$$
$$\cup \{s_1 a_i \to a^2 s_3 \mid 1 \leqslant i \leqslant k\} \cup \{s_2 a_i \to s_3 \mid 1 \leqslant i \leqslant k\}$$

(3) The so-called Collatz mapping ([3], pages 121–122),

$$f(n) = \begin{cases} \frac{n}{2} & \text{if } n \text{ is even} \\ 3\frac{n-1}{2} + 2 & \text{if } n \text{ is odd} \end{cases}$$

is also in $\mathscr{G}$, as being computed by

$$g = (\{a_1\}, \{a\}, \{s_0, s_1, s_2, s_3, s_4\}, s_0, \{s_2, s_3\}, P)$$
$$P = \{s_0 a_1 \to s_1,\ s_1 a_1 \to a s_2,\ s_2 a_1 \to s_1,$$
$$s_0 a_1 \to a^2 s_3,\ s_3 a_1 \to s_4,\ s_4 a_1 \to a^3 s_3\}$$

The relation $f \equiv g$ is obvious: on the path $s_0, s_1, s_2$ one computes $f(n)$ for even $n$ and on the path $s_0, s_3, s_4$ one computes $f(n)$ for odd $n$.

## 3. Necessary and sufficient conditions

The aim of this section is to characterize the class $\mathscr{G}$. First, two sufficient conditions are given.

THEOREM 1. *The family $\mathscr{G}$ is closed under composition.*

Proof. Let $f_i : \mathbf{N}^{k_i} \to \mathbf{N}$, $1 \leqslant i \leqslant r$, $f : \mathbf{N}^r \to \mathbf{N}$ be in $\mathscr{G}$ and let $g_i = (\{a_1, \ldots, a_{k_i}\}, \{a\}, K_i, s_{0,i}, F_i, P_i)$, $g = (\{a_1, \ldots, a_k\}, \{a\}, K, s_0, F, P)$ be *gsm*'s such that $f_i \equiv g_i$, $f \equiv g$. We construct the *gsm*

$$g' = (\{a_1, \ldots, a_t\}, \{a\}, K', q_0, \{s_f\}, P'),\ t = \sum_{i=1}^{r} k_i$$

with

$$K' = \left(\bigcup_{i=1}^{r} K_i\right) \times K \times \{0, 1\} \cup \{q_0, s_f\}$$

For $j$, $1 \leqslant j \leqslant t$, denote $\gamma(j) = j - \sum_{i=1}^{\alpha(j)-1} k_i$, where

$$\alpha(j) = \begin{cases} 1 \text{ if } j \leqslant k_1 \\ u \text{ if } \sum_{i=1}^{u-1} k_i + 1 \leqslant i < \sum_{i=1}^{u} k_i,\ u \geqslant 2 \end{cases}$$

Then, $P'$ contains the following rules:

1) $$q_0 a_j \to a^h(s_{\alpha(j)}, s, \beta), \text{ for}$$
$$1 \leqslant j \leqslant t, \ \beta \in \{0, 1\},$$
$$s_{0,\alpha(j)} a_{\gamma(j)} \to a^u s_{\alpha(j)} \text{ is a rule in } P_{\alpha(j)},$$
$$s_0 w a^u_{\alpha(j)} \overset{*}{\Rightarrow} a^h s \text{ in } g, \ w = a_1^{f_1(0,\ldots,0)} \ldots a_{\alpha(j)-1}^{f_{\alpha(j)-1}(0,\ldots,0)}$$
$$\beta = 0 \text{ iff } u = 0 \text{ (therefore } s = s_0);$$

2) $$(s_{\alpha(j)}, s, \beta) a_j \to a^h(s'_{\alpha(j)}, s', \beta'), \text{ for}$$
$$1 \leqslant j \leqslant t, \quad \beta, \beta' \in \{0, 1\},$$
$$s_{\alpha(j)} a_{\gamma(j)} \to a^u s'_{\alpha(j)} \text{ is a rule in } P_{\alpha(j)},$$
$$s a^u_{\alpha(j)} \overset{*}{\Rightarrow} a^h s' \text{ in } g,$$
$$\beta' = 0 \text{ iff } \beta = 0 \text{ and } u = 0 \text{ (therefore } s = s' = s_0);$$

3) $$(s_{\alpha(j)}, s, \beta)\, a_{j+v} \to a^h(s_{\alpha(j+v)}, s', \beta'), \ \alpha(j+v) > \alpha(j), \text{ for}$$
$$1 \leqslant j < j + v \leqslant t, \ \beta, \ \beta' \in \{0, 1\},$$
$$s_{\alpha(j)} \in F_{\alpha(j)}, \ s_{0\ \alpha(j+v)} a_{\gamma(j+v)} \to a^u s_{\alpha(j+v)} \text{ is in } P_{\alpha(j+v)},$$
$$s w a^u_{\alpha(\ +v)} \overset{*}{\Rightarrow} a^h s' \text{ in } g, \text{ where}$$
$$w = a_{\alpha(j)+1}^{f_{\alpha(j)+1}(0,\ldots,0)} \ldots a_{\alpha(j+v)-1}^{f_{\alpha(j+v)-1}(0,\ldots,0)}$$
$$\beta' = 0 \text{ iff } \beta = 0 \text{ and } u = 0 \text{ (therefore } s' = s = s_0);$$

4) For each rule $q a_j \to a^h(s_{\alpha(j)}, s, \beta)$ as above, with $s_{\alpha(j)} \in F_{\alpha(j)}$, we also introduce in $P'$ the rules

$$q a_j \to a^{h+v} s_f, \text{ for}$$

$$v = \begin{cases} 0 \text{ if } \alpha(j) = r \text{ and } s \in F \\ p \text{ if } \alpha(j) < r, \text{ where } s a_{\alpha(j)+1}^{f_{\alpha(j)+1}(0,\ldots,0)} \ldots a_r^{f_r(0,\ldots,0)} \overset{*}{\Rightarrow} a^p s' \text{ in } g, \text{ and } s' \in F. \end{cases}$$

5) For each rule $q a_j \to (s_{\alpha(j)}, s, 0)$ as above, with $s = s_0$, $s_{\alpha(j)} \in F_{\alpha(j)}$, $f_{\alpha(j)+1}(0,\ldots,0) = 0, \ldots, f_r(0,\ldots,0) = 0$, we introduce in $P'$ the rule

$$q a_j \to a^{f(0,\ldots,0)} s_f$$

From the above definition one can check that $g'(a_1^{n_1} \ldots a_t^{n_t}) = g(a_1^{u_1} \ldots \ldots a_r^{u_r})$, where $u_1 = f_1(a_1^{n_1} \ldots a_{k_1}^{n_{k_1}})$, $u_i = f_i(a_1^{n_{v_i}+1} \ldots a_{k_i}^{n_{v_i}+k_i})$, $v_i = \sum_{j+1}^{i-1} k_t$, $2 \leqslant i \leqslant r$. As $g(a_1^{u_1} \ldots a_r^{u_r}) = a^{f(u_1,\ldots,u_r)}$, we have $g' \equiv h$, where $h : \mathbf{N}^t \to \mathbf{N}$ is the mapping $h(n_1, \ldots, n_{k_1}, n_{k_1+1}, \ldots, n_t) = f(f_1(n_1, \ldots, n_{k_1}), f_2(n_{k_1+1}, \ldots, n_{k_1+k_2}), \ldots, f_r(n_{v_r+1}, \ldots, n_t))$. The parameter $\beta$ in states of $K'$ and the rules in group 5 ensure also the computation of $h(n_1, \ldots, n_r)$ for $u_i = 0$, $1 \leqslant i \leqslant r$, $u_i$ as above, that is they solve the case when we have to deal with $f(0,\ldots,0)$, each zero being the value of some $f(n_1,\ldots,n_{k_i})$, $\sum_{j=1}^{k_i} n_i \neq 0$ for at least one $i$.

COROLLARY. *If $f_1, f_2 \in \mathcal{G}$, then also $f_1 + f_2 \in \mathcal{G}$.*

Proof. Consider the mapping $f: \mathbf{N}^2 \to \mathbf{N}$, $f(n, m) = n + m$. According to Example 1, $f \in \mathcal{G}$. In view of the previous theorem, also $f(f_1(\ldots), f_2(\ldots)) = f_1(\ldots) + f_2(\ldots)$ is in $\mathcal{G}$.

THEOREM 2. *Let $f: \mathbf{N}^k \to \mathbf{N}$ be a mapping such that there are the regular languages $L_i$, $1 \leqslant i \leqslant r$, for given $r$ such that:*

i) $L_i \cap L_j \cap a_1^* \ldots a_k^* = \emptyset$, $1 \leqslant i < j \leqslant r$,

ii) $\bigcup_{i=1}^{r} (L_i \cap a_1^* \ldots a_k^*) = \{a_1^{n_1} \ldots a_k^{n_k} | (n_1,\ldots,n_k) \in dom\, f\}$,

iii) *for each $i$, $1 \leqslant i \leqslant r$, there are $\alpha_{i,j} \in \mathbf{N}$, $1 \leqslant j \leqslant k$, and $\beta_i \in \mathbf{N}$ such that $a_1^{n_1} \ldots a_k^{n_k} \in L_i$ iff $f(n_1,\ldots,n_k) = \sum_{j=1}^{k} \alpha_{i,j} n_j + \beta_i$.*

*Then, $f \in \mathcal{G}_k$ (when a mapping is lineary defined on finitely many disjoint regular parts of its domain, then the mapping is in $\mathcal{G}$).*

Proof. Let $A = (\{a_1,\ldots,a_k\}, K_i, s_{0,i}, F_i, P_i)$, $1 \leqslant i \leqslant r$, be finite automata for languages $L_i$, and assume $K_i \cap K_j = \emptyset$ for all $i \neq j$. We construct the *gsm* $g = (\{a_1,\ldots,a_k\}, \{a\}, K, s_0, \{s_f\}, P)$ with

$$K = \bigcup_{i=1}^{r} K_i \cup \{s, s_f\}$$

$$P = \{s_0 a_j \to a^{\alpha_{i,j}+\beta_i} s \mid 1 \leqslant j \leqslant k, 1 \leqslant i \leqslant r, s_{0,i} a_j \to s \in P_i\} \cup$$

$$\cup \{s_0 a_j \to a^{\alpha_{i,j}+\beta_i} s \mid 1 \leqslant j \leqslant k, 1 \leqslant i \leqslant r, s_{0,i} a_j \to s \in P_i, s \in F_i\} \cup$$

$$\cup \{s a_j \to a^{\alpha_{i,j}} s' \mid s a_j \to s' \in P_i\} \cup \{s a_j \to a^{\alpha_{ij}} s_f \mid s a_j \to s' \in P_i, s' \in F_i\}$$

Each translation $s_0 a^{n_1} \ldots a_k^{n_k} \overset{*}{\Rightarrow} a^u s_f$ in $g$ corresponds to a rewriting $s_{0,i}\, a_1^{n_1} \ldots a_k^{n_k} \overset{*}{\Rightarrow} s$, $s \in F_i$, according to some $A_i$, and has $u = \sum_{j=1}^{k} n_i \alpha_{i,j} + \beta_i$, therefore $f \in \mathcal{G}_k$.

Consider now some necessary conditions for a mapping to be in $\mathcal{G}$.

LEMMA 1. *If $f \in \mathcal{G}_k$ then there exists a natural number $\alpha$ such that $f(n_1,\ldots,n_k) \leqslant \alpha \sum_{i=1}^{k} n_n$ for all $n_1,\ldots,n_k$ with $\sum_{i=1}^{k} n_i > 0$.*

Proof. If $f \equiv g$ for $g = (\{a_1,\ldots,a_k\}, \{a\}, K, s_0, F, P)$, then the needed $\alpha$ is

$$\alpha = \max \{t \mid s a_i \to a^t s' \in P, 1 \leqslant i \leqslant k\}.$$

COROLLARY. *The non-linear bounded mappings are not in $\mathcal{G}$.*

Generally, for a given *gsm* $g = (I, O, K, s_0, F, P)$, it is known that the language

$$L_g = \{x \in I^* \mid g(x) \text{ is defined}\}$$

is regular. Therefore

$$L_{dom\,f} = \{a_1^{n_1}\ldots a_k^{n_k} \mid (n_1,\ldots,n_k) \in dom\,f\}$$

is regular for all $f \in \mathcal{G}$, because, for $g \equiv f$, we have

$$L_{dom\,f} = L_g \cap a_1^* \ldots a_k^*$$

A more precise condition is true, namely

LEMMA 2. *If $f \in \mathcal{G}_k$, then for all $m \in \mathbf{N}$, the language*

$$L_m = \{a_1^{n_1}\ldots a_k^{n_k} \mid f(n_1,\ldots,n_k) = m\}$$

*is regular.*

Proof. Let $g = (\{a_1,\ldots,a_k\}, \{a\}, K, s_0, F, P)$ be a *gsm* computing $f$. We construct the *gsm*

$$g' = (\{a_1,\ldots,a_k\}, \{a_1,\ldots,a_k, a\}, K, s_0, F, P')$$

$$P' = \{sa_i \to a_i a^r s' \mid 1 \leqslant i \leqslant k,\ sa_i \to a^r s' \in P\}$$

and consider the regular language

$$L'_m = Shuf\ (\{a^m\},\ a_1^* \ldots a_k^*),\ m \geqslant 0,$$

where *Shuf* is the operation defined by

$$Shuf\ (x, y) = \{x_1y_1x_2y_2\ldots x_ty_t \mid t \geqslant 1,\ x = x_1\ldots x_t,$$

$$y = y_1\ldots y_t,\ x_i,\ y_i \in V^*,\ 1 \leqslant i \leqslant t\},$$

$$Shuf\ (L_1, L_2) = \{w \mid w \in Shuf(x, y),\ x \in L_1,\ y \in L_2\}.$$

Then we have

$$L_m = h(g'(\{a_1^{n_1}\ldots a_k^{n_k} \mid n_i \in \mathbf{N},\ 1 \leqslant i \leqslant k\} \cap L'_m)),\ m \geqslant 0,$$

where $h: \{a_1,\ldots,a_l, a\}^* \to \{a_1,\ldots,a_k\}^*$ is the homomorphism defined by $h(a_i) = a_i$, $1 \leqslant i \leqslant k$, $h(a) = \lambda$. All the above operations (*Shuf*, $h$, $g'$, $\cap$) preserve the regular languages, therefore $L_m$ is regular.

COROLLARY 1. *The mapping* $f(n, m) = n \dot{-} m = \begin{cases} n - m \text{ for } n \geqslant m \\ 0 \text{ otherwise} \end{cases}$ *is not in* $\mathcal{G}$.

Proof. The language $L_0 = \{a_1^n a_2^m \mid n \leqslant m\}$ is not a regular one.

COROLLARY 2. *The mapping* $f(n, m) = \left[\frac{n}{m}\right]$ *is not in* $\mathcal{G}$.

Proof. Clearly, $f(n, m) = 0$ iff $n < m$, therefore $L_0 = \{a_1^n a_2^m \mid n < m\}$ and this languages is not regular.

Please note that both these mappings satisfy the condition in Lemma 1, therefore Lemma 1 does not give a sufficient condition for a mapping to be in $\mathcal{G}$. Neither the condition in Lemma 2 is sufficient: the mapping $f(n) = n^2$ is not linear, hence it is not in $\mathcal{G}$, but $L_m$ is regular for all $m$ (either $L_m = \emptyset$ or $L_m$ is a singleton). The next condition will reject such a mapping.

LEMMA 3. *If* $f \in \mathscr{G}$, *then the language*

$$L_f = \{a_1^{n_1} \ldots a_k a^{f(n_1, \ldots, n_k)} \mid (n_1, \ldots, n_k) \in dom\ f\}$$

*is linear* (*$L_f$ is the graph of f, written as language* [5]).

Proof. Let $g = (\{a_1, \ldots, a_k\}, \{a\}, K, s_0, F, P)$ be a *gsm* which computes $f$ and construct the linear grammar $G = (K, \{a_1, \ldots, a_k\}, s_0, P')$ where

$$P' = \{s \to a_i s' a^r \mid 1 \leqslant i \leqslant k,\ s a_i \to a^r s' \in P\} \cup$$
$$\cup \{s \to a_i a^r \mid s a_i \to a^r s_f \in P,\ i \leqslant i \leqslant k,\ s_f \in F\}$$

Clearly,

$$L_f = L(G) \cap a_1^* \ldots a_k^* a^*$$

As the linear languages family is closed under intersection by regular sets, the lemma is proved.

Remark. Neither this condition is sufficient for a mapping to be in $\mathscr{G}$ : the mapping $f(n, m) = n \dot{-} m$ is not in $\mathscr{G}$, but

$$L_f = \{a_1^n a_2^m a^{n-m} \mid n \geqslant m\} \cup \{a_1^n a_2^m \mid n < m\}$$

is the union of two linear languages, therefore also $L_f$ is linear. (Two grammars generating these languages are

$$G_1 = (\{S, A\}, \{a_1, a_2, a_3\}, S, \{S \to a_1 S a,\ S \to A,\ A \to a_1 A a_2,\ A \to \lambda\}),$$

$$G_2 = (\{S, A\}, \{a_1, a_2, a_3\}, S, \{S \to a_1 S a_2,\ S \to A,\ A \to A a_2,\ A \to a_2\}).)$$

This shows that the inclusion $\mathscr{G} \subset \mathscr{F}_{lin}$, $\mathscr{F}_{lin}$ being the family of mappings with linear graph in [5], is proper.

Clearly, $f(n) = n^2$ does not fulfil the condition in Lemma 3. This condition also rejects mappings which cannot be rejected by both conditions in Lemmas 1 and 2. Here is an example.

COROLLARY. *The mapping* $f(n) = [\sqrt{n}]$ *is not in* $\mathscr{G}$.

Proof. We shall prove that the language $L_f$ is not linear; in act, it is neither context-free. For, suppose $L_f$ satisfies Bar-Hillel lemma, and let $p$, $q$ be the involved pumping constants. Let $z = a_1^n a^{f(n)}$, $n > p$, be a string in $L_f$ and let $z = uvwxy$ be a splitting such that $|vx| > 0$, $uv^i w x^i y \in L_f$ for all $i \geqslant 0$. Clearly, we must have $v = a_1^t$, $x = a^s$, $t$, $s > 0$ (otherwise we cam pump only $a_1$ or only $a$, and the relation between $n$ and $f(n)$ is lost). We have $z = a_1^{r_1} a_1^t a_1^{r_2} a^{r_3} a^s a^{r_4}$, $r_1 + r_2 + t = n$, $r_3 + r_4 + s = f(n)$ and $a_1^{r_1} a_1^{it} a_1^{r_2} a^{r_3} a^{is} a^{r_4} \in L_f$ for all $i \geqslant 0$, that is $r_3 + r_4 + is = f(r_1 + r_2 + it)$, for all $i \geqslant 0$. Let $i_0$ be an integer such that $t < \sqrt{r_1 + r_2 + i_0 t}$ ($t$ is non-null) and consider the strings $a_1^{r_1} a_1^{i_0 t} a_1^{r_2} a^{r_3} a^{i_0 s} a^{r_4}$, $a_1^{r_1} a_1^{i_0 t + 2t} a_1^{r_2} a^{r_3} a^{i_0 s + 2s} a^{r_4}$. Both these strings are in $L_f$, therefor $r_3 + i_0 s + r_4 = [\sqrt{r_1 + i_0 t + r_2}]$, $r_3 + i_0 s + 2s + r_4 = [\sqrt{r_1 + i_0 t + 2t + r_2}]$. However, from the choosing of $i_0$, $\sqrt{r_1 + i_0 t + 2t + r_2} < \sqrt{r_1 + i_0 t + r_2 + 2\sqrt{r_1 + i_0 t + r_2}} < \sqrt{(\sqrt{r_1 + i_0 t + r_2} + 1)^2} = \sqrt{r_1 + i_0 t + r_2} + 1$.

For all $x \in \mathbf{R}_+$ we have $[x+1] = [x]+1$, therefore $[\sqrt{r_1+i_0t+2t+r_2}] \leqslant \leqslant [\sqrt{r_1+i_0t+r_2+1}] = [\sqrt{r_1+i_0t+r_2}]+1$. This means that $r_3 + i_0s + 2s + r_4 \leqslant r_3 + i_0s + r_4 + 1$, a contradiction, because $s > 0$. In conclusion, $L$ cannot be a context-free language, hence $f$ is not in $\mathcal{G}$.

No one of the above theree conditions is sufficient. However, the last two of them together give such a condition, hence a characterization of mappings in $\mathcal{G}$.

THEOREM 3. *A mapping* $f: \mathbf{N}^k \to \mathbf{N}$ *is in* $\mathcal{G}$ *if and only if* $L_m$ *is regular for all* $m \geqslant 0$ *and* $L_f$ *is linear* ($L_m$ *and* $L_f$ *are defined as in above Lemmas 2 and 3*).

Proof. Let $G = (V_N, \{a_1,\ldots,a_k, a\}, S, P)$ be a linear grammar for $L_f$. We assume $G$ completely reduced, that is without $\lambda$-rules, without chain rules $A \to B$, and with each $A \in V_N$ involved in a derivation $S \overset{*}{\Rightarrow} x_1Ax_2 \overset{*}{\Rightarrow} x_1x_3x_2 \in V_T^*$. For $A \in V_N$, denote $L_A = \{x \in V_T^* | A \overset{*}{\Rightarrow} x$ in the grammar $G\}$. We successively modify $G$ in the following way :

1) If a rule $A \to x_1aBx_2$ is in $P$, then card $L_B = 1$. Indeed, we have card $L_B \geqslant 1$ ($G$ is completely reduced); suppose card $L_B > 1$, $y_1, y_2 \in L_B$, $y_1 \neq y_2$. Clearly, $y_1 = a^t$, $y_2 = a^s$. Consider a derivation $S \overset{*}{\Rightarrow} w_1Aw_2 \Rightarrow w_1x_1aBx_2w_2$. Both strings $z_1 = w_1x_1ay_1x_2w_2$ and $z_2 = w_1x_1ay_2x_2w_2$ must be in $L(G)$. However $z_1 = a_1^{n_1}\ldots a_k^{n_k}a^u$, $u = |w_1x_1ay_1x_2w_2|_a$ and $z_2 = a_1^{n_1}\ldots a_k^{n_k}a^v$, $v = |w_1x_1ay_2x_2w_2|_a$, because $ax_2w_2 \in a^*$. Clearly $u \neq v$, because $y_1 \neq y_2$, which implies either $z_1$ or $z_2$ is not in $L_f$ (only one of $u$, $v$ can be the image of $(n_1,\ldots,n_k)$ by $f$). Contradiction.

As $L_B$ is a singleton, we can replace $B$ by $z \in L_B$, hence the rule becomes $A \to x_1azx_2$. In conclusion, we can assume that $G$ contains no rule of the form $A \to x_1aBx_2$.

2) Rules of the form $A \to Bax_2$ can be in $P$, but no cycle of the form $A \overset{*}{\Rightarrow} Aax_2$ may exist. For, take a derivation $S \overset{*}{\Rightarrow} w_1Aw_2 \overset{*}{\Rightarrow} w_1zw_2$ and consider the corect derivations

$$S \overset{*}{\Rightarrow} w_1Aw_2 \overset{*}{\Rightarrow} w_1Aax_2w_2 \overset{*}{\Rightarrow} w_1zax_2w_2$$

$$S \overset{*}{\Rightarrow} w_1Aw_2 \Rightarrow w_1Aax_2w_2 \Rightarrow w_1Aax_2ax_2w_2 \overset{*}{\Rightarrow} w_1zax_2ax_2w_2$$

Again, only one of the strings $w_1zax_2w_2$, $w_1zax_2ax_2w_2$ can be in $L_f$, otherwise one contradicts the fact that $f$ is a mapping.

In conclusion, we can assume that only finitely many derivations of the form $A \overset{*}{\Rightarrow} Ba^t$ or $A \overset{*}{\Rightarrow} a^t$ are possible in $G$.

2) All rules of the form $A \to Ba^r$ can be removed from $P$ in the following way. Take all the derivations of the form $A \overset{*}{\Rightarrow} Ba^t$, $A \overset{*}{\Rightarrow} a^t$ (their set is finite). For given $A \overset{*}{\Rightarrow} Ba^t$ consider all rules of the form $C \to x_1a\,Ax_2$ in $P_2$. Clearly, we must have $x_2 = a^s$. Introduce in $P$ the rules $C \to x_1a\,Ba^{t+s}$. Consider also the rules $B \to x_1Dx_2$, $x_1x_2$ containing at least a symbol $a_i$ and introduce in $P$ the rules $A \to x_1Dx_2a^t$. Finally, consider the rules $B \to x$, $x \in V_T^*$, containing at least a symbol $a_i$, and introduce in $P$ the rules $A \to xa^t$. For given $A \overset{*}{\Rightarrow} a^t$ and for each rule $C \to x_1a_iAax_2$ (we must have $x_2 = a^s$) we introduce in $P$ all rules $C \to x_1a_ia^{s+t}$.

In conclusion, we can assume that no rule of the form $A \to Ba^r$ is in $P$.

Synthetising the above arguments, we can assume that all rules in $P$ are of the forms:

(i) $A \to x_1 a_i B a_j x_2$
(ii) $A \to x_1 a_i B a^t$
(iii) $A \to x_1 a_i$
(iv) $A \to x_1 a_i a^t$

4) Each rule $r : A \to x_1 a_i B a^t$ of type (ii) can be replaced by

$$A \to \alpha_1[\alpha_1, r]a^t$$

$$[\alpha_1 \ldots \alpha_j, r] \to \alpha_{j+1}[\alpha_1 \ldots \alpha_{j+1}, r], 1 \leqslant j \leqslant |x_1| - 1$$

$$[x_1, r] \to a_i B,$$

where $x_1 = \alpha_1 \ldots \alpha_{|x_1|}$, $\alpha_j \in \{a_1, \ldots, a_i\}$;

Each rule $r : A \to x_1 a_i$ of type (ii) can be replaced by

$$A \to \alpha_1[\alpha_1, r]$$

$$[\alpha_1 \ldots \alpha_j, r] \to \alpha_{j+1}[\alpha_1 \ldots \alpha_{j+1}, r], 1 \leqslant j \leqslant |x_1| - 1$$

$$[x_1, r] \to a_i,$$

where $x_1 = \alpha_1 \ldots \alpha_{|x_1|}$, $\alpha_j \in \{a_1, \ldots, a_i\}$;

Each rule $r : A \to x_1 a_i a^t$ of type (iv) can be replaced by

$$A \to \alpha_1[\alpha_1, r]a^t$$

$$[\alpha_1 \ldots \alpha_j, r] \to \alpha_{j+1}[\alpha_1 \ldots \alpha_{j+1}, r], 1 \leqslant j \leqslant |x_1| - 1$$

$$[x_1, r] \to a_i,$$

where $x_1 = \alpha_1 \ldots \alpha_{|x_1|}$, $\alpha_j \in \{a_1, \ldots, a_i\}$.

5) Take now a rule $r : A \to x_1 a_i B a_j x_2$ of type (i) and assume $x_2 = x_2' a_m a^t$, $t \leqslant 0$. Clearly, $L_B \subseteq a_i^* \ldots a_j^*$. Consider an arbitrary given derivation $S \overset{*}{\Rightarrow} w_1 A w_2 \Rightarrow w_1 x_1 a_i B a_j x_2' a_m a^t w_2$. We must have $w_2 = a^s$. The language $L_{t+s}$ is regular. Consider the language

$$L_B' = \{w_1 x_1 a_i\} \backslash (L_{t+s} / \{a_j x_2' a_m\})$$

This is a regular language in $a_i^* \ldots a_j^*$ which includes $L_B$ and for each $z \in L_B'$, the string $w_1 x_1 a_i z a_j x_2' a_m a^t w_2$ is in $L_f$, hence in $L(G)$. Therefore

$$L_A' = \{x_1 a_i\} L_B' \{a_j x_2' a_m\}$$

is regular and each string $w_1 z a^t w_2$, with $z \in L_A'$, is in $L_f$, hence in $L(G)$. Let $G_r$ be a regular grammar for the language $L_A'$, $G_r = (V_N', \{a_1, \ldots, a_m\}, A, P')$, assume $V_N' \cap V_M = \{A\}$ introduce in $P$ all nonterminal rules in $P'$ as well as the rule

$$X \to a_m a^t$$

instead of $X \to a_m \in P'$. Remove from $P$ the rule $r$. In this way, the generated language does not change.

Using this procedure for all rules of type (i), we can lead $G$ to a grammar without such rules.

In conclusion, we can assume that $G$ contains only rules of the following forms:

$$A \to a_i B a^t, \; 1 \leqslant i \leqslant k, \; t \geqslant 0,$$

$$A \to a_i a^t, \; 1 \leqslant i \leqslant k, \; t \geqslant 0.$$

Given such a grammar for $L_f$, we construct a *gsm* $g$, $g = (\{a_1, \ldots, a_k\}, \{a\}, V_N \cup \{s_f\}, S, \{s_f\}, P'')$ with

$$P'' = \{Aa_i \to a^t B \mid A \to a_i B a^t \in P\} \cup$$
$$\cup \{Aa_i \to a^t s_f \mid A \to a_i a^t \in P\}$$

Clearly, $g$ computes $f$, hence $f \in \mathcal{G}$ and the proof is over.

## 4. The syntactic complexity of mappings in $\mathcal{G}$

For a given *gsm* $g = (I, O, K, s_0, F, P)$ we define

$State\,(g) = \text{card}\; K$

$Fin(g) = \text{card}\; F$

$Prod\,(g) = \text{card}\; P$

$Length\,(g) = \max\, \{|x| \mid sa \to xs' \in P\}$

$Symb(g) = \sum_{r \in P} Symb(r)$, where $Symb(r) = |x| + 4$ for

$$r : sa \to xs' \in P.$$

For a mapping $f \in \mathcal{G}$ and a measure $M \in \{State, Fin, Prod, Length, Symb\}$ we define

$$M(f) = \inf\{M(g) \mid g \equiv f\}$$

In the style of [2], we say that $M$ is *nontrivial* if for all $n \geqslant n_0$ there is $f_n \in \mathcal{G}$ such that $M(f_n) > n$; $M$ is called *connected* if for al $n \geqslant n_0$ there is $f_n \in \mathcal{G}$ such that $M(f_n) = n$.

THEOREM 4. *All above defined measures are connected, excepting* $Fin$, *which is trivial.*

Proof. If $g = (I, O, K, s_0, F, P)$ has card $F > 1$ then we consider the *gsm* $g' = (I, O, K \cup \{s_f\}, s_0, \{s_f\}, P')$ where $s_f$ is a new state and

$$P' = P \cup \{sa \to xs_f \mid sa \to xs' \in P, \; s' \in F\}$$

Clearly, $g'$ is equivalent to $g$, hence $Fin$ is a trivial measure.

Consider now the mapping $f_n : \mathbf{N} \to \mathbf{N}$, $f_n(m) = \begin{cases} 0 & \text{if } m \leqslant n \\ m - n & \text{if } m > n. \end{cases}$

It is computed by $g = (\{a_1\}, \{a\}, \{s_0, s_1, \ldots, s_n\}, s_0, \{s_1, \ldots, s_n\}, P)$, with

$$P = \{s_i a_1 \to s_{i+1} \mid 0 \leqslant i \leqslant n - 1\} \cup$$
$$\cup \{s_n a_1 \to a s_n\}$$

Therefore $State\,(f_n) \leqslant n + 1$, $Prod\,(f_n) \leqslant n + 1$.

Let us suppose that a *gsm* $g' = (\{a_1\}, \{a\}, K, s_0, F, P)$ there is such that $g' \equiv f$ and *State* $(g') < n + 1$. Take a translation $s_0 a_1^n \overset{*}{\Rightarrow} s'$. It is of the form $s_0 a_1^n \Rightarrow s_1 a_1^{n-1} \Rightarrow \ldots \Rightarrow s_n = s'$, $s' \in F$. Clearly, all states $s_0, s_1, \ldots, s_n$ must be different, otherwise a cycle $s_i a_1^{j-i} \overset{*}{\Rightarrow} s_j = s_i$ is possible, which leads to translations of the form $s_0 a_1^{n+j-i} \overset{*}{\Rightarrow} s_i a_1^{n-i+j-i} \overset{*}{\Rightarrow} s_j a_1^{n-i} = s_i a_1^{n-i} \overset{*}{\Rightarrow} s_j a_1^{n-i-j+i} \overset{*}{\Rightarrow} s_n$ which implies $f_n(n + j - i) = 0$, a contradiction. Consequently, *State* $(g') \geqslant n + 1$, hence *State* $(f_n) = n + 1$.

The $n + 1$ states are linked by at least $n$ productions and at least a cycle there exists, hence for a state $s$ we need two rules of the form $s_i a_1 \to xs$, $sa_1 \to xs_j$. This implies *Prod* $(g') \geqslant n + 1$, hence *Prod* $(f_n) = n + 1$.

Consider now the mapping $f_n : \mathbf{N} \to \mathbf{N}$ defined by $f_n(m) = n$. It is computed by $g = (\{a_1\}, \{a\}, \{s_0, s_1\}, s_0, \{s_1\}, P)$, with

$$P = \{s_0 a_1 \to a^n s_1,\ s_1 a_1 \to s_1\}.$$

Therefore, *Length*$(f_n) \leqslant n$, *Symb*$(f_n) \geqslant n + 8$. As the translation $s_0 a_1 \overset{*}{\Rightarrow} a^n s_f$ must contain only one step, we find that the rule $s_0 a_1 \to a^n s_f$ is in all *gsm's* for $f_n$, hence *Length* $(f_n) = n$. Such a rule $s_0 a_1 \to a^n s_f$ cannot be used in a cycle therefore there is also a further rule $s_0 a_1 \to xs'$. This means that each $g' \equiv f_n$ has *Symb* $(g') \leqslant n + 8$, therefore *Symb*$(f_n) = n + 8$ and the proof is over.

Another important problem in the theory of syntactic complexity of languages is the minimization. The problem remains open for the above measures other than *Fin*. We deal here only with the related compatibility problem: For a measure $M : \mathscr{G} \to \mathbf{N}$ we define

$$M^{-1}(f) = \{g \mid g \equiv f \text{ and } M(g) = M(f)\},\ f \in \mathscr{G}.$$

Two measures $M_1$, $M_2$ are said to be *incompatible* when there is $f \in \mathscr{G}$ such that

$$M_1^{-1}(f) \cap M_2^{1-}(f) = \emptyset.$$

(the two measures cannot be simultaneously minimized).

LEMMA 4. *The measures Length and Fin are incompatible with Prod and Symb.*

Proof. Consider the mapping $f : \mathbf{N} \to \mathbf{N}$ defined by $f(n) = \begin{cases} 1 \text{ if } n = 1 \\ 4 \text{ if } n \geqslant 2 \end{cases}$. It is computed by $g_1 = (\{a_1\}, \{a\}, \{s_0, s_1, s_2\}, s_0, \{s_1, s_2\}, \{s_0 a_1 \to as_1,\ s_1 a_1 \to a^3 s_2,\ s_2 a_1 \to s_2\})$, hence *Prod*$(f) \leqslant 3$, *Symb*$(f) \leqslant 16$, as well as by $g_2 = (\{a_1\}, \{a\}, \{s_0, s_1, s_2\}, s_0, \{s_2\}, \{s_0 a_1 \to as_2, s_0 a_1 \to a^2 s_1, s_1 a_1 \to s_1, s_1 a_1 \to a^2 s_2\})$, hence *Length*$(f) \leqslant 2$.

Let $g$ be a *gsm* for $f$ with *Length* $(g) \leqslant 2$. As we need a translation $s_0 a_1^2 \Rightarrow a^t s_1 a_1 \Rightarrow a^4 s_2$, we must have $t \leqslant 2$, $4 - t \leqslant 2$, hence $t = 2$. This implies that a further rule $s_0 a_1 \to as_3$ must exists. No one of these rules can be used in a cycle ($f(n) \leqslant 4$ for all $n$), therefore also a rule $sa_1 \to s'$ must exist, used in a cycle. In conclusion, *Prod*$(g) \geqslant 4$, hence *Prod* and *Length* are incompatible. The above rules have the total *Symb* equal to $4 \cdot 4 + 5 = 21$, therefore also *Symb* and *Length* are incompatible.

If $g$ has $Fin(g) = 1$, then we have two translations $s_0a_1 \Rightarrow as_f$, $s_0a_1a_1 \Rightarrow a^ts_1a_1 \Rightarrow a^4s_f$. When the rule $s_0a_1 \to as_f$ is used in $s_0a_1a_1 \Rightarrow a^ts_1a_1$, this implies that $t = 1$, $s_f = s_1$ and the rule $s_fa_1 \to a^3s_f$ is in $g$, a contradiction, because illegal translations can be obtained. Therefore, $s_f \neq s_1$ and we have three rules involved in these translations. No one can be used in a cycle, hence again $Prod(g) \geqslant 4$, which implies $Symb(g) \geqslant 21$, hence also $Fin$ is incompatible with $Prod$ and $Symb$.

LEMMA 5. *The measure Fin is incompatible with State, but it is compatible with Length.*

Proof. The compatibility of $Fin$ and $Length$ easily follows from the construct which shows the triviality of $Fin$ (Theorem 4 : that construction does not change the length of rules in $g$).

Consider now the mapping $f : \mathbf{N}^2 \to \mathbf{N}$ defined by $f(n, m) = \begin{cases} n + m & \text{if } n \neq 0 \\ 2m & \text{if } n = 0. \end{cases}$

It is computed by $g = (\{a_1, a_2\}, \{a\}, \{s_0, s_1\}, s_0, \{s_0, s_1\}, \{s_0a_1 \to as_1, s_1a_1 \to as_1, s_1a_2 \to as_1, s_0a_2 \to a^2s_0\})$ therefore $State(f) \leqslant 2$. Suppose that a $gsm$ $g$ there exists, $g \equiv f$, with $Fin(g) = 1$, $State(g) = 2$. If $s_0$ is the final state, then we must have the rules $s_0a_1 \to s_0$, $s_0a_2 \to a^2s_0$, which lead to $s_0a_1a_2 \Rightarrow as_0a_2 \Rightarrow a^3s_0$, a contradiction. Therefore the final state $s_f$ is different from $s_0$. We need a cycle of the form $sa_k^2 \overset{*}{\Rightarrow} a^{2k} s$, as well as one of the form $s'a_1^k \overset{*}{\Rightarrow} a^ks'$. We cannot have $s = s'$, neither $s' = s_0$, $s = s_f$ (in both cases we can obtain $s_0a_1^ua_2^v \overset{*}{\Rightarrow} a^ws_f$, with $u \neq 0, w \neq u + v$). Therefore $s = s_0$, $s' = s_f$. In order to obtain $s_0a_1a_2^k \overset{*}{\Rightarrow} a^{k+1}s_f$ for arbitrary $k$, we also need a cycle $s''a_2^u \overset{*}{\Rightarrow} a^us''$, If $s'' = s_0$, then $s_0a_2^{r(k+u)+1} \overset{*}{\Rightarrow} a^{2kr}s_0a_2^{ru+1} \overset{*}{\Rightarrow} a^{2kr+ur}s_0a_2 \overset{*}{\Rightarrow} a^{2kr+ur+v}s_f$ and for large enough $r$ we have $2kr + ru + v \neq 2(rk + ru + 1)$. If $s'' = s_f$, then $s_0a_2^{rk+ru+1} \overset{*}{\Rightarrow} a^{2kr}s_0a_2^{ru+1} \overset{*}{\Rightarrow} a^{2kr+v}s_fa_2^{ru} \overset{*}{\Rightarrow} a^{2kr+v+ru}s_f$. Again, for large enough $r$ we have $2kr + v + ru \neq 2(rk + ru + 1)$. Both cases are contradictory, the existence of $g$ is impossible, hence either $Fin(g) \geqslant 2$, or $State(g) \geqslant 3$.

Open problems. Which of the next pairs contain incompatible measures : ($Prod$, $Symb$), ($Prod$, $State$), ($State$, $Symb$), ($State$, $Length$).

Note. Useful remarks by dr. Cristian Calude are acknowledged.

*Received 13 V 1987*

REFERENCES

[1] C. Calude, *Theories of computational complexity*, North-Holland, Amsterdam, 1987.
[2] J. Gruska, *Descriptional complexity of context-free languages*, Proc. Symp. Math. Found. Computer Sci., High Tatras, 1973.
[3] R. K. Guy, *Unsolved problems in number theory*, vol. 1 in Ser. Unsolved problems in intuitive mathematics, Springer-Verlag, New York, 1981.
[4] Gh. Păun, *The complexity of languages translation by gsm's Rev. Roum. Lingv. CLTA*, **25**, 1 (1988), 49—58.
[5] Gh. Păun, M. Tătărâm, *Classes of mappings having context sensitive graphs*, Rev. Roum. Math. Pures Appl., 30, 4 (1985), 273—288.
[6] A. Salomaa, *Formal languages*, Academic Press, New York, London, 1973.
[7] D Simovici, S. Istrail, *Computing grammars and context sensitive languages*, RAIRO/Inform. Theorique, 12, 1 (1978), 33—48.
[8] M. Tătărâm, *Ackermann-Peter's function has primitive recursive graph and range*, Found. Control Engineering, 9, 4 (1984), 177—180.
[9] J. Van Leeuwen, *Rule-labeled programs*, Ph. D. Dissertation, Dept. of Math., Univ. of Utrecht, 1972.