RAIRO. Informatique théorique

G. Păun

An operation with languages occurring in the linguistic approach to the management

RAIRO. Informatique théorique, tome 11, nº 4 (1977), p. 303-310

<http://www.numdam.org/item?id=ITA_1977__11_4_303_0>

R.A.I.R.O. Informatique théorique/Theoretical Computer Science
(vol. 11, n° 4, 1977, p. 303 à 310).

AN OPERATION WITH LANGUAGES OCCURRING IN THE LINGUISTIC APPROACH TO THE MANAGEMENT(*)

par G. PĂUN [1]

Communicated by Wilfried BRAUER

Abstract. — *For* $x, y \in V^*$ *we define* $x \sim y$ *iff* x *is a permutation of* y. *For a finite language* $L_0 \subset V^+$ *and for any* $x \in L_0^*$ *define* $Agg_{L_0}(x) = \{ \hat{x}_1 \ldots \hat{x}_p \mid p \geqq 1,\ x = x_1 \ldots x_p,\ x_i \in L_0 \}$, *where* $\hat{x}_i$ *is the equivalence class of* x_i *with respect to* $\sim$. *We show that all the families* $\mathscr{L}_i$, $i = 0, 1, 2, 3$ *in the Chomsky hierarchy are closed under this operation (called aggregation with respect to* L_0*). Moreover,* $\mathrm{Ind}\,(\mathrm{Agg}_{L_0}(L)) \leqq \mathrm{Ind}\,(L)$, *for any* L, *but there is* L_0 *and* $L \subset L_0^*$ *such that* $K(\mathrm{Agg}_{L_0}(L)) > K(L)$ *for any* $K \in \{ \mathrm{Var}, \mathrm{Prod}, \mathrm{Symb}, \mathrm{Lev} \}$ [3]. *As the aggregation occurs in the passing of the management of a system from a smaller working time interval to a larger one, these results are significant with respect to the cost (the algorithmical complexity) of the management.*

NTRODUCTION

In this paper we continue the mathematical-linguistic approach to the economical process from [5, 6]. In this way a further evidence is obtained for the large capabilities of the mathematical linguistics to be applied in practical, nonlinguistic fields (see [4] for many discussions and examples on this subject).

Observing that a given economical system has a finite number of states, its evolution is described by a string of states. Thus the planning of the evolution of this system means the determination of a set of strings of states (allowed trajectories). On the other hand, the follow-up (the control of the system evolution) implies to decide whether or not a given string is an allowed trajectory. Consequently, the planning is a generative activity whereas the follow-up is a parsing activity. In this way, the management becomes a linguistic activity and can be investigated in terms of formal languages.

Although the economical process continuously develop, their evolution is distinguishable (quantifiable) at discrete times. Let t be the length of the interval defined by these times. On the other hand, the management has a given working time interval (monthly, weekly, etc.) — let us denote it by τ-which is greater than the interval t. Generally, τ is a multiple of t, $\tau = kt$.

(*) Received 11. juanuary 1977. Revised 1. june 1977.

[1] Institute of Mathematics, S 1. Bucharest, Romania.

For a given time unit, u, a state of the system will describe the system evolution in one interval of the form $[i, i+1]$, i a positive integer. Therefore, the system states according to t are not perceptible by the management, because it works at larger intervals. The states corresponding to τ are obtained by equalization of some strings of τ/t states corresponding to t, namely of those strings which describe the same evolution of the system. This operation is called *aggregation* (of information).

In what follows, the aggregation is formally defined as an operation on languages and the closure of the four families in the Chomsky hierarchy under this operation is investigated as well as the effect of this operation on the grammatical complexity of languages.

2. THE AGGREGATION AS AN OPERATION ON LANGUAGES

We assume that the notions and results from [1] and [7] on formal languages theory are known. We specify only some notations.

Let V be a vocabulary, and V^* be the free monoid generated by V under the operation of concatenation. We denote by λ the unit element of V^* and by $|x|$ the length of $x \in V^*$.

A Chomsky grammar is a quadruple $G = (V_N, V_T, S, P)$ where V_N is the nonterminal vocabulary, V_T is the terminal vocabulary, $S \in V_N$ is the starting symbol of G, and P is the set of production rules. The rules are written in the form $\alpha \to \beta$. The language generated by G is denoted by $L(G)$. According to the form of the rules in P, four types of grammars are defined and thus four families of languages are obtained: the recursively enumerable languages, the context-sensitive languages, the context-free and, respectively, the regular languages. We denote these families by $\mathscr{L}_i$, $i = 0, 1, 2, 3$.

DEFINITION 1: Let V be a vocabulary and $x, y \in V^*$. We define $x \sim y$ iff x is obtained by permuting the letters of y.

Obviously, this is an equivalence relation. We denote by $\hat{x}$ the equivalence class of x with respect to $\sim$.

Let $L_0 \subset V^+$ be a finite language (V^+ denotes the set $V^* - \{\lambda\}$). For any $x \in L_0^*$ we define $\mathrm{Agg}_{L_0}(x) \Leftarrow \{\hat{x}_1 \hat{x}_2 \ldots \hat{x}_p \mid p \geqq 1, x = x_1 x_2 \ldots x_p, x_i \in L_0$ for any i $\}$.

If $L \subset L_0^*$ then we define

$$\mathrm{Agg}_{L_0}(L) = \bigcup_{x \in L} \mathrm{Agg}_{L_0}(x).$$

The language $\mathrm{Agg}_{L_0}(L)$ is called the *aggregation of L with respect to L_0*.

REMARK 1: Consider again the management of some system. According to the above definition, the passing of the management from t to τ corresponds to an aggregation with respect to $L_0 = \{x \mid x \in V^*, |x| = \tau/t\}$, where the set of states was denoted by V. It is known that the handling of a language (of the generative devices—grammars or automata—generating this language, and of the parsing algorithms) essentially depends on the type of the language in the Chomsky hierarchy: the greater i is, the easier is the handling of $L \in \mathscr{L}_i$. Consequently, from the point of view of the algorithmic complexity of the the management, it is very important to answer the question: which relation holds between the type of L and that of $\mathrm{Agg}_{L_0}(L)$, as well as between the grammatical complexity of L and that of $\mathrm{Agg}_{L_0}(L)$? (In other words, how the aggregation acts on the complexity of the management algorithms?). This is the purpose of the following sections.

3. THE FAMILIES $\mathscr{L}_i$, $i = 0, 1, 2, 3$ AND THE OPERATION Agg

THEOREM 1: *Let $L_0 \subset V^+$ be a finite language and $L \subset L_0^*$. If $L \in \mathscr{L}_i$, then* $\mathrm{Agg}_{L_0}(L) \in \mathscr{L}_i$, $i = 0, 1, 2, 3$.

Proof (1). Let us suppose that $L_0 = \{ x_1, x_2, \ldots, x_n \}$. We consider the following a-transducer [2]

$$M = (K, V_1, V_2, H, s_0, F)$$

with $K = \{ s_0 \}$ (the set of states), $F = K$ (the set of final states), $V_1 = V$ (the imput vocabulary), $V_2 = \{ \hat{x}_i \mid i = 1, 2, \ldots, n \}$ (the output vocabulary), and $H = \{ (s_0, x_i, \hat{x}_i, s_0) \mid i = 1, 2, \ldots, n \}$ (the set of moves). Obviously, we have $M(L) = \mathrm{Agg}_{L_0}(L)$, for any $L \subset L_0^*$. Since M is a λ-free a-transducer, our theorem follows from Corollary 2 of Theorem 3.2.1 [2].

Therefore the aggregation does not increase the complexity of languages measured by the position in the Chomsky hierarchy. In fact, frequently, by aggregation a simpler language is obtained.

THEOREM 2: 1) *If $L_0 \subset V^+$ such that $x \sim y$ for any $x, y \in L_0$, then for any $L \subset L_0^*$, $L \in \mathscr{L}_2$, the language* $\mathrm{Agg}_{L_0}(L)$ *is regular;* 2) *There are $L_0 \subset V^+$ and $L \subset L_0^*$ such that $L \notin \mathscr{L}_2$ and* $\mathrm{Agg}_{L_0}(L) \in \mathscr{L}_3$.

Proof: Let $L \subset L_0^*$ be a context-free language and let M be the a-transducer constructed in the proof of Theorem 1. Then the language $\mathrm{Agg}_{L_0}(L) = M(L)$ is context-free on the vocabulary with only one element, $\{ \hat{x} \}$, x arbitrary in L_0. Consequently $\mathrm{Agg}_{L_0}(L)$ is regular.

In fact, assertion 1 is also true if we replace $\mathscr{L}_2$ by any family of languages, $\mathscr{L}$, for which any language $L \in \mathscr{L}$ on the vocabulary with only one element is regular. Such families are those of matrix languages of finite index and of

(1) In the first version of this paper this theorem was proved in a more complicated way. We are indebted to Professor W. Brauer for the suggestion to use the a-transducers and the results from [2] in order to obtain an essential simplification of the proof.

simple matrix languages. These families strictly contain $\mathscr{L}_2$, therefore assertion 2 follows too.

4. THE OPERATION Agg AND THE GRAMMATICAL COMPLEXITY OF LANGUAGES

Languages in a given family can be compared by means of some measures of grammatical complexity. Such measures were defined and intensively investigated for the family of context-free languages [3].

In what follows we shall show that the aggregation decreases the grammatical complexity of context-free languages from the point of view of the Index [3, 7] but for some other measures this does not hold.

DEFINITION 2: Let $G = (V_N, V_T, S, P)$ be a context-free grammar. We define the following complexity measures for G (see [3]).

$$\mathrm{Var}(G) = \mathrm{card}\, V_N,$$

$$\mathrm{Prod}(G) = \mathrm{card}\, P,$$

$$\mathrm{Symb}(G) = \sum_{r \in P} \mathrm{Symb}(r),$$

where $\mathrm{Symb}(r) = |\alpha| + 2$ for $r : A \to \alpha$.

$$\mathrm{Lev}(G) = \mathrm{card}\,(V_N/\simeq),$$

where for A_1 and A_2 in V_N we define $A_1 \simeq A_2$ iff there are the derivations $A_1 \overset{*}{\underset{G}{\Rightarrow}} z A_2 y$ and $A_2 \overset{*}{\underset{G}{\Rightarrow}} z' A_1 y'$.

For a derivation

$$D : S = w_0 \Rightarrow w_1 \Rightarrow \ldots \Rightarrow w_k = w,\ w \in V_T^*$$

we put $\mathrm{Ind}(D, G) = \max_{i \leq k} |h(w_i)|$ where $h : V_N \cup V_T \to V_N^*$ is defined by $h(A) = A$, $A \in V_N$ and $h(a) = \lambda$, $a \in V_T$. Then $\mathrm{Ind}(w, G) = \min_D \mathrm{Ind}(D, G)$, where D exhausts the set of derivations giving w, and

$$\mathrm{Ind}(G) = \sup\{\mathrm{Ind}(w, G) \mid w \in L(G)\}.$$

For a language $L \in \mathscr{L}_2$ and a complexity measure K we define

$$K(L) = \inf\{K(G) \mid L = L(G)\},$$

and, for any $n \geqq 1$, we put

$$K^{-1}(n) = \{L \in \mathscr{L}_2 \mid K(L) \leqq n\}.$$

We shall prove that $\mathrm{Ind}^{-1}(n)$ is closed under the operation Agg but, for $K \in \{\mathrm{Prod}, \mathrm{Var}, \mathrm{Symb}, \mathrm{Lev}\}$, this does not hold.

LEMMA 1: *For any $L_0 \subset V^+$ (finite) and $L \subset L_0^*$, there are a homomorphism h, a finite substitution s, and a regular language R such that*

$$\mathrm{Agg}_{L_0}(L) = h(s(L) \cap R).$$

Proof: If $L_0 = \{ x_1, \ldots, x_n \}$ then we define

$$s : V \to \mathscr{P}((V \cup \{ \hat{x}_i \mid i = 1, 2, \ldots, n \})^*)$$

by $s(a) = \{ a, a\hat{x}_i \mid i = 1, 2, \ldots, n \}$, $a \in V$,
and

$$h : V \cup \{ \hat{x}_i \mid i = 1, 2, \ldots, n \} \to \{ \hat{x}_i \mid i = 1, 2, \ldots, n \}^*$$

by $h(a) = \lambda$, $a \in V$, $h(\hat{x}_i) = \hat{x}_i$, $i = 1, 2, \ldots, n$.

Considering the regular language $R = \{ x_i \hat{x}_i \mid i = 1, 2, \ldots, n \}^*$, the equality $\mathrm{Agg}_{L_0}(L) = h(s(L) \cap R)$ obviously follows.

THEOREM 3: *For any finite $L_0 \subset V^+$ and context-free $L \subset L_0^*$ we have* $\mathrm{Ind}(\mathrm{Agg}_{L_0}(L)) \leqq \mathrm{Ind}(L)$.

Proof: From Lemma 1, it is sufficient to prove that for any $n \geqq 1$ the family $\mathrm{Ind}^{-1}(n)$ is closed under (arbitrary) finite substitutions and under intersection with regular languages.

Let $G = (V_N, V_T, S, P)$ be a context-free grammar and $s : V_T \to \mathscr{P}(U^*)$ a finite substitution. We extend s to $V_N \cup V_T$ by $s(A) = \{ A \}$ for $A \in V_N$. Then take the grammar

$$G' = (V_N, U, S, \{ A \to w \mid \text{there is } A \to x \text{ in } P \text{ such that } w \in s(x) \}.$$

Obviously, $s(L(G)) = L(G')$ and $\mathrm{Ind}(G) \geqq \mathrm{Ind}(G')$. Thus,

$$\mathrm{Ind}(s(L(G)) \leqq \mathrm{Ind}(L(G)).$$

Let now A be finite automaton

$$A = (V, K, s_0, F, P)$$

(with V the vocabulary, K the set of states, s_0 the initial state, F the set of final states, and P the set of rules of the form $s_i a \to s_j$, $s_i, s_j \in K$, $a \in V$), and $L(A)$ the language accepted by A.

For a context-free grammar $G = (V_N, V, S, P)$ we construct the grammar

$$G' = (\{ S_0 \} \cup (K \times V_N \times K), V, S_0, P')$$

where

$$\begin{aligned} P' = \{ S_0 \to (s_0, S, s) \mid s \in F \} \\ \cup \{ (s, C, s') \to x_1 (s_1, B_1, s_1') x_2 (s_2, B_2, s_2') x_3 \ldots x_k (s_k, B_k, s_k') x_{k+1} \mid s, s', \\ s_i, s_i' \in K \text{ for all } i, C \to x_1 B_1 x_2 B_2 x_3 \ldots x_k B_k x_{k+1} \in P, \\ x_i \in V^*, B_i \in V_N \text{ for all } i, \\ \text{and } s x_1 \overset{*}{\underset{A}{\Rightarrow}} s_1, s_i' x_{i+1} \overset{*}{\underset{A}{\Rightarrow}} s_{i+1}, i = 1, 2, \ldots, k-1, s_k' x_{k+1} \overset{*}{\underset{A}{\Rightarrow}} s' \} \\ \cup \{ (s, B, s') \to x \mid s, s' \in K \text{ and } B \to x \in P, x \in V^* \text{ and } s x \overset{*}{\underset{A}{\Rightarrow}} s' \}. \end{aligned}$$

It is easy to see that $L(G') = L(G) \cap L(A)$ and $\operatorname{Ind}(G') \leqq \operatorname{Ind}(G)$, hence $\operatorname{Ind}(L(G) \cap L(A)) \leqq \operatorname{Ind} L(G)$.

REMARK 2: We can show that $\operatorname{Ind}^{-1}(n)$ is closed under union and substitution with regular sets. According to the Corollary of Proposition 3.7.1 [2], it follows that $\operatorname{Ind}^{-1}(n)$ is closed also under inverse homomorphisms, hence it is a trio. Thus, Theorem 3 can be obtained from Theorem 1 and from Corollary 2 of Theorem 3.2.1 [2].

The aggregation does not increase the grammatical complexity of languages according to Ind. For the other measures previously defined the situation is different.

THEOREM 4: There are finite $L_0 \subset V^+$ and context-free $L \subset L_0^*$ such that for any $K \in \{ \text{Var, Prod, Symb, Lev} \}$ we have

$$K(\operatorname{Agg}_{L_0}(L)) > K(L).$$

Proof: Let $V = \{ a, b \}$, $L_0 = \{ ab, baba, aba \}$, and

$$L = \{ (ab)^{n+1} a (ab)^n \mid n \geqq 0 \}.$$

The grammar $G = (\{ S \}, V, S, \{ S \to ab\, S\, ab,\ S \to aba \})$ generates the language L, therefore, $L \subset L_0^*$, $\operatorname{Var}(L) = 1$, $\operatorname{Lev}(L) = 1$, $\operatorname{Prod}(L) = 2$ and, it is easy to see, $\operatorname{Symb}(L) = 12$. On the other hand

$$\operatorname{Agg}_{L_0}(L) = \{ \hat{x}_1^n \hat{x}_3 \hat{x}_2^m \hat{x}_1^p \mid n, m, p \geqq 0, n+2m = p \}$$

where

$$(x_1, x_2, x_3) = (ab, baba, aba).$$

The inclusion $\supseteq$ is obvious. Now, take $w \in L$ and $\hat{w} \in \operatorname{Agg}_{L_0}(L)$ such that w is obtained from $\hat{w}$ by replacing all symbols $\hat{x}_i$ by the corresponding strings x_i. Assume that $\hat{w}$ contains k_i symbols $\hat{x}_i$, $i = 1, 2, 3$. Then w must contains $k_1 + 2k_2 + 2k_3$ symbols a and $k_1 + 2k_2 + k_3$ symbols b. Since w contains exactly one more a than b's, this implies $k_3 = 1$. Obviously $\hat{w}$ must begin with $\hat{x}_1$ or $\hat{x}_3$ and end with $\hat{x}_1$. No substring $\hat{x}_1 \hat{x}_2$ is allowed in $\hat{w}$. Therefore no substring $\hat{x}_2 \hat{x}_3$ may occur in $\hat{w}$. Consequently, $\hat{w}$ must have the form $\hat{x}_1^n \hat{x}_3 \hat{x}_2^m \hat{x}_1^p$, $n, m, p \geqq 0$. A simple counting argument gives $n + 2m = p$.

Let $G = (V_N, V_T, S, P)$ be a context-free reduced grammar generating $\operatorname{Agg}_{L_0}(L)$, and define

$$[S] = \{ A \in V_N \mid A \simeq S \}, \ \simeq \text{ as in Definition 2,}$$

$$G([S]) = ([S], V_T \cup V_N - [S], S, P \cap ([S] \times (V_N \cup V_T)^*)),$$

(i) Assume that $L(G([S]))$ is infinite. Then for each $T \in [S]$ there is a rule $T \to u\, S\, v$ in P. Therefore only rules of the form $T \to w$ with $T \in [S]$ and $w \in L(G([S]))$ may produce $\hat{x}_3$. Moreover all the rules in $G([S])$ must be linear, since otherwise strings with more than one $\hat{x}_3$ could be derived. Consequently each terminal rule in $G([S])$ must produce x_3. Hence P cannot contain a rule $A \to \hat{x}_2^i\, B\, \hat{x}_1^j$ with $A, B \in [S]$ and $i > 0$.

(ii) Assume moreover that P contains a rule $A \longrightarrow u\, B\, \hat{x}_2^i$ with $A,\ B \in [S]$ and $i > 0$. Then there is no rule of the form $C \longrightarrow v\, D\, \hat{x}_1^j$ with $C,\ D \in [S]$ and $j > 0$ in P. Therefore for some sufficiently large k and arbitrary $n,\ m$ the strings $\hat{x}_1^n\, \hat{x}_3\, \hat{x}_2^m\, \hat{x}_1^k$ cannot be generated. Consequently all rules in P involving a nonterminal from $[S]$ on their right-hand side must have the form $A \longrightarrow \hat{x}_1^i\, B\, \hat{x}_1^j,\ A,\ B \in [S],\ i, j > 0$, for if $i = 0$, or $j = 0$, then strings $\hat{x}_1^n\, \hat{x}_3\, \hat{x}_2^m\, \hat{x}_1^p$ such hat $p \neq n+2\,m$ could be generated.

(iii) Lev $(G) = 1$ would imply $[S] = V_N$, i. e. $L\,(G\,([S])) = L\,(G)$, and (i), (ii) would therefore lead to a contradiction. Consequently Lev $(\mathrm{Agg}_{L_0}\,(L)) > 1$. Hence Var $(\mathrm{Agg}_{L_0}\,(L)) > 1$, Prod $(\mathrm{Agg}_{L_0}\,(L)) > 2$.

(iv) If $L\,(G\,([S]))$ is infinite, then (i)-(iii) imply that P contains at least the following rules

$$A \rightarrow \hat{x}_1^i\, B\, \hat{x}_1^j,\ A,\ B \in [S],\ i, j > 0,$$

$$C \rightarrow \hat{x}_r^i\, D\, \hat{x}_k^j,\ C,\ D \in V_N - [S],\ \{r,\ s\} = \{1,\ 2\},\ i, j > 0,$$

$$A \rightarrow w\, C\, w',\ A \in [S],\ w\, C\, w' \in L(G([S])),$$

$$D \rightarrow v,\ D \in V_N - [S],\ w \in V_T^*,$$

where vww' must contain $\hat{x}_3$.

Therefore Symb $(G) \geqq 16$.

(v) Assume that $L\,(G\,([S]))$ is finite. It is easy to see that there must not exist a derivation $S \underset{G}{\overset{*}{\Rightarrow}} u\, A\, v\, B\, w,\ A,\ B \in V_N$ such that A and B both generate infinite sets of strings. Then for each $T \in V_N$ such that there is a rule $S \longrightarrow u\, T\, u'$ in P, the grammar $G_T = (V_N - \{S\},\ V_T,\ T,\ P)$ can be analysed using similar arguments as in (i)-(v). This proves that Symb $(G) \geqq 16$ in any case. Hence Symb $(\mathrm{Agg}_{L_0}\,(L)) \geqq 16$.

REMARK 3: There are context-free languages, L, such that $K(\mathrm{Agg}_{L_0}(L)) < K(L)$, for any $K \in \{$ Var, Prod, Symb, Lev $\}$. Indeed, let $V = \{a, b\}$, $L_0 = \{ab, ba\}$ and $L = \{(ab)^n\,(ab)^m\,(ba)^m \mid n \geqq 1,\ \ m \geqq 0\}$. We have Var $(L) > 1$, Prod $(L) > 2$, Symb $(L) > 9$, Lev $(L) > 1$, but $\mathrm{Agg}_{L_0}\,(L) = \{\widehat{ab}^{\,2n} \mid n \geqq 1\}$ and Var $(\mathrm{Agg}_{L_0}\,(L)) = 1 =$ Lev $(\mathrm{Agg}_{L_0}\,(L))$, Prod $(\mathrm{Agg}_{L_0}\,(L)) = 2$, Symb $(\mathrm{Agg}_{L_0}\,(L)) = 9$.

CONJECTURE: *For any $L \subset L_0^*$ with $L_0 \subset V^+$ such that $|x| = |y|$ for all $x,\ y \in L_0$ (this is the case for the aggregation occurring in the above approach to the management), we have, $K\,(\mathrm{Agg}_{L_0}\,(L)) \leqq K\,(L)$, $K \in \{$*Var, Prod, Symb, Lev*$\}$.*

Note: Many helpful discussions with Professor S. Marcus are acknowledged, as well as very useful remarks by Professor W. Brauer who helped me to give a clearer and more elegant form to the second part of the proof of Theorem 4.

REFERENCES

1. M. Gross et A. Lentin, *Notions sur les grammaires formelles*, Gauthier-Villars, Paris, 1970.
2. S. Ginsburg, *Algebraic and Automata Theoretic Properties of Formal Languages* North-Holland Pub. Comp. Amsterdam, Oxford, 1975.
3. J. Gruska, *Descriptional Complexity of Context-Free Languages*, Proc. of Symp. and Summ. School. Math. Found. of Computer Sc. High Tatras, 1975.
4. S. Marcus, *Linguistics as a Pilot Science*, In *Current Trends in Linguistics*, Th. Sebeok (Ed.), vol. 12, Mouton, The Hague, Paris, 1974.
5. G. Păun, *A Formal Linguistic Approach to the Production Process*, Foundations of Control Engineering, Poznan, **1**, 3 (1976).
6. G. Păun, *Generative Grammars for some Economical Activities* (*Scheduling and Marshalling Problems*), Foundations of Control Engineering, **2**, 1 (1977).
7. A. Salomaa, *Formal Languages*, Academic Press, New York and London, 1973.