Formal properties of literal shuffle

B. BERARD

Abstract

We introduce the literal shuffle operation, that is a more constrained form of the well-known shuffle operation. In order to describe concurrent processes, the shuffle operation models the asynchronous case, while the literal shuffle operation corresponds to a synchronous behaviour.

The closure properties of some classical families of languages under literal shuffle are studied and properties of families of languages defined by means of literal shuffle are given.

Introduction

The shuffle operation naturally appears in several problems, like concurrency of processes ([9], [10], [11]), or multi-point communication, where all stations share a single bus ([5]). That is one of the reasons of the large theoretical literature about this operation (see for instance [1], [3], [6], [7]). In the latter example, general shuffle operation models the asynchronous case, where each transmitter uses asynchronously the single communication channel. If the hypothesis of synchronism is made (step-lock transmission), the situation is modeled by what can be named "literal" shuffle. Each transmitter emits, in turn, one elementary signal. The same remark holds for concurrency, where general shuffle corresponds to asynchronism and literal shuffle to synchronism.

There are no specific studies of literal shuffle. One of the reasons is perhaps that, when adding the full trio operations, literal shuffle is as powerful as general shuffle. Nevertheless, when a more precise approach is made, literal shuffle appears as satisfying specific properties. In the present paper, we study the literal shuffle operation, particularly in relation with the classical families of languages: regular sets, context-free languages, context-sensitive languages and recursively enumerable sets. The paper is divided in three sections. The first one contains some specific definitions about shuffle and literal shuffle, and some basic properties of these operations. In the second section, we study the closure properties of the families Rat, $\mathscr{C}f$, $\mathscr{C}\mathscr{S}$ and $\mathscr{R}\mathscr{E}$ under literal shuffle and we show that the family of recursively enumerable sets is the smallest full trio closed under iterated literal shuffle, thus extending a result of M. Jantzen [6] about the shuffle operation.

In the third section, we give some properties of the language families obtained by using literal shuffle, in the same way as the families Shuf and $\mathcal{SE}$ were studied in [6]. The main purpose of this section is to state that the two families obtained that way and $\mathcal{SE}$ are incomparable.

Notations and basic definitions

Let X be an alphabet. X^* is the free monoid generated by X, and ε will denote the empty word in X^*.

Let f be a word in X^*, $|f|$ is the length of f and if f is not the empty word, $f^{(i)}$ is the i^{th} letter of f, $|f|_x$ is the number of occurrences of the letter x in f.

A word g in X^* is a *subword* of f if $f=ugv$, for some, u, v in X^*. If u is the empty word, g is a *prefix* of f.

Fin, Rat, $\mathcal{C}f$, $\mathcal{CS}$, $\mathcal{RE}$ will respectively denote the family of finite sets, regular sets, context-free languages, context-sensitive languages, recursively enumerable sets. Let X and Y be two alphabets. A homomorphism h from X^* into Y^* is said to be: *non erasing* if $h(X)\subseteq Y^+$, where $Y^+=Y^*-\{\varepsilon\}$,
alphabetical if $h(X)\subseteq Y\cup\{\varepsilon\}$,
a coding if $h(X)\subseteq Y$,
an isomorphism if h and h^{-1} are codings. In that case, Y is called a copy of X and if L is a language in X^*, $h(L)$ is called a copy of L.

$\hat{\mathcal{H}}$ is the class of all homomorphism and $\mathcal{H}^{-1}$ is the class of all inverse homomorphisms.

A *full trio* is a family of languages closed under homomorphisms, inverse homomorphisms and intersections with regular sets. $\hat{\mathcal{M}}=(\hat{\mathcal{H}}, \mathcal{H}^{-1}, \wedge\mathcal{R})$ will denote the full trio operations, where $\wedge\mathcal{R}$ is the class of intersections with regular sets. $D_1'^*$ is the resticted Dyck set over the alphabet $\{a, b\}$ generated by the context-free grammar with productions:

$$S\rightarrow aSb+SS+\varepsilon \text{ (see [4] and [3] for details).}$$

Part 1 — Shuffle and literal shuffle

The shuffle operation will be denoted by the symbol ш and is defined for languages L and M in X^* by

$$L\text{ш}M = \{f = u_1 v_1 \dots u_n v_n,\ u_i, v_i \text{ in } X^*,\ u_1 \dots u_n \in L,\ v_1 \dots v_n \in M\}.$$

The iterated shuffle will be denoted by ш*. Let L be a language in X^*, then $L^{\text{ш}*}= \bigcup_{i\geq 0} L_i$, where $L_0=\{\varepsilon\}$ and $L_{i+1}=L_i\text{ш}L$. The families Shuf and $\mathcal{SE}$ were introduced by M. Jantzen [6]: Shuf$=(\cup, \text{ш}, \text{ш}^*)$(Fin) is the least family of languages including Fin and closed under union, shuffle and iterated shuffle. $\mathcal{SE}=(\cup, \cdot, *, \text{ш}, \text{ш}^*)$(Fin) is the least family of languages including Fin and closed under union, product, Kleene star, shuffle and iterated shuffle.

We give now the specific notations of this paper and make the ideas more precise about literal shuffle.

Let f and g be two words in X^* with the same length p. The interleaving I of the words, f, g is defined by:

$$I(\varepsilon, \varepsilon) = \varepsilon \quad \text{if} \quad p = 0,$$

$$I(f, g) = f^{(1)} g^{(1)} \dots f^{(p)} g^{(p)} \quad \text{if} \quad p > 0.$$

Let L and M be languages in X^*, we define:

1) *The initial literal shuffle* $ш_1$:

$$L ш_1 M = \{I(f_1, f_2)\, g \mid f_1, f_2, g \ \text{ in } \ X^*, \ |f_1| = |f_2|,$$
$$(f_1 g \in L \ \text{ and } \ f_2 \in M) \ \text{ or } \ (f_1 \in L \ \text{ and } \ f_2 g \in M)\}.$$

2) *The literal shuffle* $ш_2$:

$$L ш_2 M = \{f I(g_1, g_2) h \mid f, g_1, g_2, h \ \text{ in } X^*, \ |g_1| = |g_2|,$$
$$(f g_1 h \in L \ \text{ and } \ g_2 \in M) \ \text{ or}$$
$$(g_1 \in L \ \text{ and } \ f g_2 h \in M) \ \text{ or}$$
$$(f g_1 \in L \ \text{ and } \ g_2 h \in M) \ \text{ or}$$
$$(g_1 h \in L \ \text{ and } \ f g_2 \in M)\}.$$

Example: $L = a^*$ and $M = b^*$

$$L ш_1 M = (ab)^*(a^* \cup b^*),$$

$$L ш_2 M = (a^* \cup b^*)(ab)^*(a^* \cup b^*).$$

3) *The iterated initial literal shuffle* $ш_1^*$ and the *iterated literal shuffle* $ш_2^*$:

$$L^{ш_1^*} = \bigcup_{i \geq 0} L_i, \quad \text{where} \quad L_0 = \{\varepsilon\} \quad \text{and} \quad L_{i+1} = L_i ш_1 L,$$

$$L^{ш_2^*} = \bigcup_{i \geq 0} L_i, \quad \text{where} \quad L_0 = \{\varepsilon\} \quad \text{and} \quad L_{i+1} = L_i ш_2 L.$$

We then define four families of languages:

$$\mathscr{L}_1 \mathscr{S} h = (\cup, ш_1, ш_1^*)(\mathrm{Fin})$$

$$\mathscr{L} \mathscr{S} h = (\cup, ш_2, ш_2^*)(\mathrm{Fin})$$

$$\mathscr{L}_1 \mathscr{S} \mathscr{E} = (\cup, \cdot, *, ш_1, ш_1^*)(\mathrm{Fin})$$

$$\mathscr{L} \mathscr{S} \mathscr{E} = (\cup, \cdot, *, ш_2, ш_2^*)(\mathrm{Fin}).$$

At the end, we summarize some basic properties of the initial literal shuffle and the literal shuffle.

Proposition 1.1. Let X be an alphabet and A, B languages in X^*.

a) The initial literal shuffle and the literal shuffle are not associative operations.
b) The literal shuffle is commutative but the initial literal shuffle is not commutative.

c) $AB \subseteqq A\shuffle_2 B$, $A\shuffle_1 B \subseteqq A\shuffle_2 B \subseteqq A\shuffle B$.

d) $X^* = X^{\shuffle_1^*} = X^{\shuffle_2^*}$.

e) Let f, g, h be words in X^* such that $h=f\shuffle_1 g$ or $h\in f\shuffle_2 g$, then $|h|=|f|+|g|$.

Recall ([1]) that $D_1'^*=(ab)^{\shuffle *}$; we have:

Proposition 1.2.

a) $(ab)^{\shuffle_1^*} = \{\varepsilon, ab\}\cup a^2(ab)^* b^2$,

b) $(ab)^{\shuffle_2^*} = D_1'^*$.

The initial literal shuffle seems then to be less powerful than both shuffle and literal shuffle. However, we will see that even a very simple language like $((ab)^{\shuffle_1^*})^{\shuffle_1^*}$ is not context-free. Furthermore, the three families $\mathscr{S}\mathscr{E}$, $\mathscr{L}_1\mathscr{S}\mathscr{E}$ and $\mathscr{L}\mathscr{S}\mathscr{E}$ are pairwise incomparable.

Proof.

a) The proof is straightforward.

b) From the definition, we can write $(ab)^{\shuffle_2^*} = \bigcup_{p\geqq 0} L_p$, where

$$L_0 = \{\varepsilon\} \quad \text{and} \quad L_{p+1} = L_p \shuffle_2 \{ab\}.$$

Since $A\shuffle_2 B\subseteqq A\shuffle B$ (Proposition 1.1.c), it is easy to verify that

$$A^{\shuffle_2^*} \subseteqq A^{\shuffle^*}, \quad \text{thus} \quad (ab)^{\shuffle_2^*} \subseteqq D_1'^*.$$

For the converse inclusion let f be in $D_1'^*$ with $|f|=2p$. An induction argument proves that f is in L_p.

The basis when $p=1$ is trivial.

Induction step. Assume the result for words of length $2p$ and consider a word f in $D_1'^*$ of length $2(p+1)$. There are two possibilities:

Case 1. $f=(ab)^{p+1}$. By the induction hypothesis, $(ab)^p$ is in L_p, thus f belongs to $L_p\{ab\}$. Since $L_p\{ab\}\subseteqq L_p\shuffle_2\{ab\}$ (Prop. 1.1.c), $f\in L_{p+1}$.

Case 2. $f=f_1 f_2 f_3$, where f_2 is a word of D_1', the set of restricted Dyck primes, with $|f_2|\geqq 4$.

Let $u_0=\varepsilon, u_1, \dots, u_{2k}=f_2$, be the sequence of prefixes of f_2, $k\geqq 2$, and let $\|u_j\|=|u_j|_a-|u_j|_b$ be the height of the word u_j. If i is the greatest integer such that $\|u_i\|$ is maximum, then there exists a letter x in $\{a, b\}$ and a word v in $\{a, b\}^*$ with $f_2= =u_{i-2}xabbv$. We define $g=f_1 u_{i-2}$, $v_1=xb$, $v_2=ab$, $h=vf_3$. $f= gl(v_1, v_2)h$, thus f is in $gv_1 h\shuffle_2 ab$. Since $gv_1 h$ is a word in $D_1'^*$ of length $2p$, $gv_1 h$ is in L_p by induction hypothesis. Consequently, f is in L_{p+1}.

Part 2 — Closure properties of the families Rat, $\mathscr{C}f$, $\mathscr{C}\mathscr{S}$ and $\mathscr{R}\mathscr{E}$ under literal shuffle

We first show that, when adding the full trio operations, literal shuffle is a powerful as shuffle.

Recall ([3]) that a full trio is closed under shuffle if and only if it is closed under intersection.

Proposition 2.1. Let $\mathscr{L}$ be a full trio. The following properties are equivalent:

a) $\mathscr{L}$ is closed under shuffle.
b) $\mathscr{L}$ is closed under literal shuffle.
c) $\mathscr{L}$ is closed under initial literal shuffle.

Proof. The result is easily obtained from the two following facts. Let L and M be languages respectively in X^* and Y^*.

Fact 1. Assume that X and Y are disjoint alphabets; we define regular languages in $(X\cup Y)^*$ by:

$$R_1 = (XY)^*(X^*\cup Y^*) \quad \text{and} \quad R_2 = (X^*\cup Y^*)(XY)^*(X^*\cup Y^*).$$

Then

$$L\sqcup\!\sqcup_1 M = (L\sqcup\!\sqcup M)\cap R_1 \quad \text{and} \quad L\sqcup\!\sqcup_2 M = (L\sqcup\!\sqcup M)\cap R_2.$$

Fact 2. If $\$$ is a new letter and if h is the homomorphism from $(X\cup Y\cup\{\$\})^*$ onto $(X\cup Y)^*$ defined by:

$$h(z)=z, \text{ for each } z \text{ in } X\cup Y, \text{ and } h(\$)=\varepsilon,$$

then

$$L\sqcup\!\sqcup M = h[h^{-1}(L)\sqcup\!\sqcup_1 h^{-1}(M)] = h[h^{-1}(L)\sqcup\!\sqcup_2 h^{-1}(M)].$$

Proposition 2.2. Let L be a language in X^*, let $\$$ be a letter not in X and let h be the homomorphism from $(X\cup\{\$\})^*$ onto X^* defined by: $h(x)=x$ if x is in X, $h(\$)=\varepsilon$. Then,

$$L^{\sqcup\!\sqcup^*} = h[(h^{-1}(L))^{\sqcup\!\sqcup_1^*}] = h[(h^{-1}(L))^{\sqcup\!\sqcup_2^*}].$$

Proof. Using Proposition 1.1.c, we can get

$$h[(h^{-1}(L))^{\sqcup\!\sqcup_1^*}] \subseteq h[(h^{-1}(L))^{\sqcup\!\sqcup_2^*}].$$

Furthermore, if φ is an arbitrary homomorphism and if A, B are languages, then

$$\varphi(A\sqcup\!\sqcup B) \subseteq (A)\sqcup\!\sqcup\varphi(B).$$

Therefore, we have the following inclusions:

$$h[(h^{-1}(L))^{\sqcup\!\sqcup_1^*}] \subseteq h[(h^{-1}(L))^{\sqcup\!\sqcup_2^*}] \subseteq L^{\sqcup\!\sqcup^*}.$$

Conversely, we use the definition of iterated shuffle and initial literal shuffle:

$$L^{\sqcup\!\sqcup^*} = \bigcup_{n\geqq 0} L_n, \quad L_0 = \{\varepsilon\}, \quad L_{n+1} = L_n\sqcup\!\sqcup L$$

and

$$(h^{-1}(L))^{\sqcup\!\sqcup_1^*} = \bigcup_{n\geqq 0} M_n, \quad M_0 = \{\varepsilon\}, \quad M_{n+1} = M_n\sqcup\!\sqcup_1 h^{-1}(L).$$

We prove that for each integer $n \geqq 0$, $L_n \subseteqq h(M_n)$. If $n=0$ or $n=1$, the result is immediate. Assume $n \geqq 2$ and let u be a word in $u_1 \sqcup\!\sqcup \ldots \sqcup\!\sqcup u_n$, where $u_i \in L$. There exists an integer $p \geqq 1$ such that

$$u = \prod_{j=1}^{p} (u_{1,j} \ldots u_{n,j}), \quad u_{i,j} \text{ in } X^*, \quad u_i = u_{i,1} \ldots u_{i,p}.$$

We define a sequence of words f_i, $1 \leqq i \leqq n$, by:

$$f_i = f_{i,1} \cdots f_{i,p}, \quad f_{i,j} = \$^{r_{i,j}} u_{i,j} \$^{s_{i,j}}, \quad \text{with:}$$

$$r_{1,j} = 0$$

$$r_{i,j} = 2^{i-2}(|u_{1,j}| + \ldots + |u_{i-1,j}|), \quad i \geqq 2$$

$$s_{1,j} = |u_{2,j}| + \ldots + |u_{n,j}|$$

$$s_{i,j} = (2^{i-2}-1)\,|u_{i,j}| + 2^{i-2}(|u_{i+1,j}| + \ldots + |u_{n,j}|), \quad i \geqq 2.$$

Clearly, f_i belongs to $h^{-1}(L)$, $1 \leqq i \leqq n$,

$$|f_1| = |u| \quad \text{and} \quad |f_i| = 2^{i-2}|u| \quad \text{for each} \quad i \geqq 2.$$

Define:

$$g_1 = f_1 \quad \text{and for} \quad 1 \leqq i \leqq n-1, \quad g_{i+1} = g_i \sqcup\!\sqcup_1 f_{i+1}.$$

Obviously, g_i is in M_i, $1 \leqq i \leqq n$. Further, $|g_i| = |f_{i+1}| = 2^{i-1}|n|$ for $1 \leqq i \leqq n$, and $|g_n| = 2^{n-1}|u|$.

Then, we can write $g_i = g_{i,1} \ldots g_{i,p}$, where $|g_{i,j}| = |f_{i+1,j}|$, $1 \leqq i < n$, and $|g_{n,j}| = 2|f_{n,j}|$. It is easy to prove by induction on $i \geqq 2$ that:

$$g_{i,j} = g'_{i,j}\, \$^{t_{i,j}}, \quad \text{where} \quad t_{i,j} = s_{i,j} + |u_{i+1,j}|$$

and

$$h(g_{i,j}) = u_{1,j} \ldots u_{i,j}.$$

For $i=n$, we obtain:

$$g_n = g_{n,1} \cdots g_{n,p}, \quad h(g_{n,j}) = u_{1,j} \cdots u_{n,j},$$

hence $h(g)=u$ and u is in $h(M_n)$.

From $L_n \subseteqq h(M_n)$, we have $L^{\sqcup\!\sqcup^*} \subseteqq h[(h^{-1}(L))^{\sqcup\!\sqcup_1^*}]$, and the proof is complete.

We now state the closure properties of the families Rat, $\mathscr{CS}$ and $\mathscr{RE}$ under literal shuffle. They can be obtained by easy machine constructions.

Proposition 2.3.

a) The families Rat, $\mathscr{CS}$ and $\mathscr{RE}$ are closed under $\sqcup\!\sqcup_1$ and $\sqcup\!\sqcup_2$.

b) Moreover, the families $\mathscr{CS}$ and $\mathscr{RE}$ are closed under $\sqcup\!\sqcup_1^*$ and $\sqcup\!\sqcup_2^*$.

Corollary 1. The families $\mathscr{L}_1\mathscr{SE}$ and $\mathscr{LSE}$ are both contained in the family of context-sensitive languages.

We will see in the next section that there are, in fact, proper containments.

Using Propositions 2.2 and 2.3 together with a result of M. Jantzen ([6]): $\mathscr{RE} = (\mathscr{M}\ \sqcup\!\sqcup^*)(\text{Fin})$, we can show:

Corollary 2. The family of recursively enumerable sets is the least family of languages including the finite sets and closed under the full trio operations and the iterated literal shuffle.

The same result holds with the iterated initial literal shuffle:

$$\mathcal{RE} = (\hat{\mathcal{M}}, \sqcup\!\sqcup_1^*)\,(\text{Fin}) = (\hat{\mathcal{M}}, \sqcup\!\sqcup_2^*)\,(\text{Fin}).$$

Property 2.3.a) does not remain true for context-free languages: let L and M be two different copies of the restricted Dyck set over the disjoint alphabets $\{a, b\}$ and $\{c, d\}$, respectively. Then, neither $L \sqcup\!\sqcup_1 M$ nor $L \sqcup\!\sqcup_2 M$ are context-free languages.

We mention a strong result of M. Latteux about the shuffle operation:

Proposition 2.4. ([7]) Let L and M be two languages over disjoint alphabets X and Y respectively. $L \sqcup\!\sqcup M$ is a context-free language if and only if either L or M is a regular language.

This result does not extend to the initial literal shuffle: Let G be the context-free, non regular language over the alphabet $\{a, b\}$ defined by:

$$G = \{a^{n_1} b \dots a^{n_k} b \mid k \geqq 1, \quad n_i \geqq 0, \quad \exists i \neq n_i\}.$$

(G is known as the Goldstine's language.) If $\bar{G}$ is a copy of G over the alphabet $\{\bar{a}, \bar{b}\}$, we have:

Proposition 2.5. $G \sqcup\!\sqcup_1 \bar{G}$ is a context-free language.

Scheme of the proof. Let $\$$ be a new letter and let $\hat{G}$ be the following language in $(\{a, b, \$\} \times \{a, b, \$\})^*$:

$$\hat{G} = \left\{ \begin{bmatrix} f\$^p \\ g\$^q \end{bmatrix}, \quad f \in G, \quad g \in G \quad \text{and} \quad p + |f| = q + |g| \right\}^{1)}.$$

Let h be the homomorphism from $(\{a, b, \$\} \times \{a, b, \$\})^*$ into $\{a, b, \bar{a}, \bar{b}\}^*$ defined by:

$$h\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = x\bar{y} \quad \text{if} \quad x, y \in \{a, b\}, \quad h\left(\begin{bmatrix} \$ \\ \$ \end{bmatrix}\right) = \varepsilon,$$

$$h\left(\begin{bmatrix} x \\ \$ \end{bmatrix}\right) = x \quad \text{if} \quad x \in \{a, b\} \quad \text{and} \quad h\left(\begin{bmatrix} \$ \\ y \end{bmatrix}\right) = \bar{y} \quad \text{if} \quad y \in \{a, b\}.$$

Clearly enough, $h(\hat{G}) = G \sqcup\!\sqcup_1 \bar{G}$. Then, it suffices to prove the context-freeness of $\hat{G}$, and we build a pushdown automaton recognizing $\hat{G}$. We will use two different versions of non-deterministic pushdown automata recognizing G (by final states).

First version. The underlying idea of how this automaton works is the following: let w be a word in $\{a, b\}^*$. Non-deterministically, a block of a's is chosen. The b's preceding this block are pushed into the stack. Then, each a in the chosen block makes

[1] If $x, y \in \{a, b, \$\}^*$ with $|x| = |y| = n$, we write $\begin{bmatrix} x \\ y \end{bmatrix}$ for $\begin{bmatrix} x^{(1)} \\ y^{(1)} \end{bmatrix} \dots \begin{bmatrix} x^{(n)} \\ y^{(n)} \end{bmatrix}$.

a b to be popped from the stack. The word w is accepted if the number of a's in the chosen block does not match the number of b's in the stack. (Initially, the stack contains a single b.)

Second version. It allows to keep in the stack, after checking, the rank of the chosen block of a's. It is based upon the fact that G is also defined by:

$$G=\{a^{n_1}b\ldots a^{n_p}b \mid n_1\neq 1 \quad \text{or there exists a } k, \quad 1\leqq k\leqq p-1,$$

$$\text{such that } n_{k+1}\neq n_k+1\}.$$

The automaton first checks wether or not $n_1=1$ or chooses a block of a's. (Let $k+2$ be its rank.) In the second case, the first k b's are pushed into the stack, then the a's of the following block (their number is n_{k+1}) are also pushed into the stack. The b is skipped and it is then checked if the number of a's in the following block is different of $n_{k+1}+1$ (by using the n_{k+1} a's on the top of the stack). If this test is positive, the word is accepted and the rank of the current block can be retrieved from the stack (number of b's plus 2).

Now we can describe a non-deterministic pushdown automaton recognizing $\hat{G}$. As long as couples of letters $\begin{bmatrix}a\\a\end{bmatrix}$ or $\begin{bmatrix}b\\b\end{bmatrix}$ are encountered, the automaton works as in the second version. As soon as a couple $\begin{bmatrix}a\\b\end{bmatrix}$ or $\begin{bmatrix}b\\a\end{bmatrix}$ is encountered (involving that one of the upper and lower words is then known to be in G), the automaton uses the a's at the top of the stack for determining which word is in G (say the upper word). Using the b's in the stack and switching for first version, the automaton checks then that the other word (here the lower one) does belong to G.

Clearly, no problem appears if the first encountered couple of different letters is $\begin{bmatrix}x\\\$\end{bmatrix}$ or $\begin{bmatrix}\$\\y\end{bmatrix}$, $x,y\in\{a,b\}$.

Open question: Do there exist two non regular languages L and M over disjoint alphabets, such that $L\text{ш}_2M$ is context-free?

Property 2.3b) does not hold for Rat or $\mathscr{C}f$. We use Proposition 2.2 with the language $L=\{abc\}$. It is easy to see that $L^{\text{ш}*}$ is not context-free. $M=h^{-1}(L)=$ $=\$^*a\$^*b\$^*c\$^*$ is a regular language and since $L^{\text{ш}*}=h(M^{\text{ш}_1^*})=h(M^{\text{ш}_2^*})$ is not in $\mathscr{C}f$, neither $M^{\text{ш}_1^*}$ nor $M^{\text{ш}_2^*}$ is a context-free language.

However, regular languages or context-free languages can be obtained in some very particular cases:

Proposition 2.6. Let F be a finite set. $F^{\text{ш}_1^*}$ is a regular language.

Proposition 2.7. Let F be a finite set such that for any word f in F, the length of f is less than or equal to 2. Then, $F^{\text{ш}_2^*}$ is a context-free language.

Proof of Prop. 2.6. The proof consists of a construction of a left linear grammar such that $L(G)=F^{\text{ш}_1^*}$.

Since $\emptyset^{\text{ш}_1^*}=\{\varepsilon\}$ and for any language A, $(A\cup\{\varepsilon\})^{\text{ш}_1^*}=A^{\text{ш}_1^*}$, we may assume that F is not the empty set and does not contain the empty word; $F=\{f_1, \ldots, f_k\}$, $k\geqq 1$. If X is the alphabet of F, we set $p=\operatorname{card}(X)$, $t=\max\{|f_j|, 1\leqq j\leqq k\}$ and we consider the set X^t of words in X^* with length t: $X^t=\{g_1, \ldots, g_m\}$ where $m=p^t$. We can write

$$F^{\text{ш}_1^*} = \bigcup_{i\geqq 0} L_i, \quad L_0 = \{\varepsilon\}, \quad L_{i+1} = L_i \,\text{ш}_i\, F.$$

Let n_0 be the smallest integer greater than or equal to k, such that for each word f in L_{n_0}, $|f|\geqq t$.

Since $\varepsilon\notin F$, the words in L_i are strictly shorter than the words in L_{i+1} and such an integer n_0 can be found.

We define: $R=\bigcup\limits_{i\leqq n_0-1} L_i$, R is a finite set,

$$J(i) = \{f\in L_{n_0} | g_i \text{ is a prefix of } f\}, \quad 1\leqq i\leqq m,$$

$$I=\{i\in\{1, \ldots, m\} | J(i)\neq\emptyset\}$$

and for each $i\in I$, $q_i=\operatorname{card}(J(i))$, so that

$$J(i) = \{h_{i,1}, \ldots, h_{i,q_i}\} \text{ with } h_{i,r} = g_i u_{i,r} \text{ for some } u_{i,r} \text{ in } X^*,$$

$1\leqq r\leqq q_i$,

$$L_{n_0} = \bigcup_{i\in I} J(i).$$

For each (i,j), $1\leqq i\leqq m$, $1\leqq j\leqq k$, there exists a unique integer $s(i,j)$ in $\{1, \ldots, m\}$ and a unique word $v_{i,j}$ in X^* such that:

$$g_i \,\text{ш}_1\, f_j = g_{s(i,j)} v_{i,j}.$$

Now we can finish the proof by constructing a grammar $G=(X, N, S, P)$; $N=\{S, D_1, \ldots, D_m\}$, where $S, D_1, \ldots, D_m$ are new letters. The rules of P are the following:

(i) $S\to w$ for each word w in R;

(ii) $S\to D_i u_{i,r}$ for each r, $1\leqq r\leqq q_i$, for each i in I;

(iii) $D_i\to g_i$, $1\leqq i\leqq m$;

(iv) $D_i\to D_{s(i,j)} v_{i,j}$, $1\leqq j\leqq k$, $1\leqq i\leqq m$.

G is left linear and it is easy to see that $L(G, S)=F^{\text{ш}_1^*}$.

Proof of Prop. 2.7. Let F be a finite set in X^* and $L=F^{\text{ш}_2^*}$. If every word in F is of length less than or equal to 1 and if X is of minimal cardinality, then $L=X^*$ is a regular language. Since $(A\cup\{\varepsilon\})^{\text{ш}_2^*}=A^{\text{ш}_2^*}$ for any language A, we may assume that F does not contain the empty word.

We define a sequence of languages F_n, $n\geqq 1$, inductively by: $F_1=F$,

$F_{n+1}=\{f\in X^* |$there exists a word g in F_n such that:

either $g=g_1g_2$, $g_2\neq\varepsilon$ and $f=g_1yg_2$, where y is a word of length 1 in F, or $g=g_1xg_2$; for some x in X and $f=g_1y_1xy_2g_2$, where y_1y_2 is a word of length 2 in F.}

For each $n \geqq 1$, the set F_n is contained in L, therefore the language M defined by $M=\bigcup_{n\geqq 1} F_n$ is also contained in L. It is straightforward to verify that L is a submonoid of X^*; it follows that $M^* \subseteqq L$. The converse inclusion also holds; the argument is an induction on the length of a word in L.

Since $L=M^*$, it suffices to show that M is a context-free language. Thus, we construct a context-free grammar $G=(X, N, S, P)$ such that $L(G, S)=M$.

We consider the fixed alphabet $X=\{a_1, \ldots, a_p\}$ and we define: $N=\{S, T_1, \ldots, T_p\}$, where $S, T_1, \ldots, T_p$ are new letters; the homomorphism h from X^* into N^* such that $h(a_i)=T_i$, $1 \leqq i \leqq p$;

$$I = \{i \in \{1, \ldots, p\} | a_i \in F\}$$

and $w_j=a_{j_1}a_{j_2}$, $1 \leqq j \leqq k$; the words of length 2 in F.

The productions of P are the following:

(i) $S \to T_i, \quad i \in I$

$S \to T_{j_1}T_{j_2}, \quad 1 \leqq j \leqq k$

(ii) $\left.\begin{array}{ll} T \to T_iT; & i \in I, \\ T \to T_{j_1}TT_{j_2}, & 1 \leqq j \leqq k, \end{array}\right\}$ for any variable $T \in \{T_1, \ldots, T_p\}$

(iii) $T_i \to a_i, \quad 1 \leqq i \leqq p.$

Clearly, this grammar generates M.

Part 3 — Properties of the families $\mathscr{L}_1\mathscr{S}\mathscr{E}$ and $\mathscr{L}\mathscr{S}\mathscr{E}$

We do not mention in this part specific properties of the families $\mathscr{L}_1\mathscr{S}h$ and $\mathscr{L}\mathscr{S}h$; however, we state two useful results about some particular languages in these families.

Proposition 3.1. The language $N=((ab)^{Ш_1^*})^{Ш_1^*}$ (in $\mathscr{L}_1\mathscr{S}h$) is not context-free.

Proof. (the details are omitted)

a) Let f be a word in $\{a, b\}^*$. The height of f is $\|f\|=|f|_a-|f|_b$ and $\mathrm{PR}(n)$ denotes the set of all prefixes g of the words in the language N, satisfying: $|g| \leqq n$.

We define, for each integer $n \geqq 0$, $H(n)=\mathrm{Max}\,\{\|g\|,\ g \in \mathrm{PR}(n)\}$. By induction on $n \geqq 2$, we can obtain the following inequality:

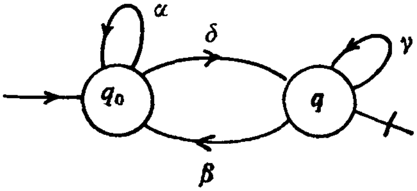
$$H(n) \leqq 6 \log_2(n).$$

b) A sequence f_k; $k \geqq 1$, of words in N can be constructed, such that: $f_k=g_kb^{3k+4}$, for some word g_k in $\{a, b\}^*$.

c) We suppose now that the language N is context-free and, using the Iteration Theorem, [4], we will obtain a contradiction. Let N_0 be the integer from the Iteration Theorem and let $h=f_{N_0}$ be the word of N, obtained as in b): $h=g_{N_0}b^{3N_0+4}$, where the last $3N_0+4$ b's are distinguished. There exists a factorization $h=\alpha u\beta v\gamma$, such that $h_p=\alpha u^p\beta v^p\gamma\in N$, for any $p\geqq 0$. The height of h_p is 0, for any $p\geqq 0$, and v is a subword of b^{3N_0+4}. Thus, $\|u\|>0$ and, using a), we obtain a sequence αu of prefixes of N, such that $\|\alpha\|+p\|u\|\leqq 6\log_2(|\alpha|+p|u|)$, which is impossible. Hence, N is not context-free.

Proposition 3.2. The language $P=\{ab, cd\}^{\text{ш}_s^*}$ (in $\mathscr{L}\mathscr{S}h$) is a generator of the family of context-free languages.

Proof. We define the words $\alpha=a^{m+n}$, $\beta=b^n(ac)^p$, $\gamma=(bd)^pb^m$ and $\delta=ab$, where m, n and p are integers, $m\geqq 2$, $p\geqq 2$, $n\geqq p+1$. We then define a regular set K recognized by the transition system ([4]) of the figure below:



At the end, we introduce the context-free language A, generated by the grammar with productions: $T\to\alpha T\beta T\gamma+\delta$.

We shall prove that $P\cap K=A$. Since $\{\alpha, \beta, \gamma, \delta\}$ is a code[2], it proves that P is a generator of f ([2]).

We will say that a word f is directly reduced in a word g if $f=f'axbf''$ or $f=f'cxdf''$ and $g=f'xf''$, for some letter x in X and some words f', f'' in X^*. We will write $f\to g$ and $\xrightarrow{*}$ will denote the reflexive and transitive closure of $\to$. If $f\xrightarrow{*}g$, we say that f is reduced in g.

a) It can be shown by induction on $k\geqq 2$ that, if f is a word in A, $|f|\leqq k$, then f is reduced in ab. This gives the inclusion $A\subseteqq P$.

b) It is easy to see, by induction on the length of a word in A, that $A\subseteqq K$.

c) So far, we have obtained the inclusion $A\subseteqq P\cap K$. To get the converse inclusion, we need two facts:

Fact 1: Let f be a word in P, neither $\alpha\delta\gamma$ nor $\beta\delta\beta$ is a subword of f.

Fact 2: Let $f=f'\alpha\delta\beta\delta\gamma f''$ be a word in P. Then f is reduced in $f'\delta f''$, and this reduction is the only one which can concern the subword $\alpha\delta\beta\delta\gamma$ of f. Let f be a word in $P\cap K$. The argument is again an induction on the length of f.

Case 1. α is not a subword of f. Since f is in K, it can be written as: $f=(\delta\gamma^{r_1}\beta)\dots\dots(\delta\gamma^{r_k}\beta)\delta\gamma^{r_{k+1}}$. Since f is in P, $|f|_a=|f|_b$, therefore $k=0$ and $r_{k+1}=0$; hence $f=ab$ is in A.

Case 2. α is a subword of f. We then consider the last occurrence of α in f, so that $f=f'\alpha f''$, α is not a subword of f''. Since f is in K and in P, using Fact 1, we obtain:

[2] A subset C in X^+ is a code if C^* is a free monoid with base C.

$f''=\delta\beta\delta\gamma h$, $f=f'\alpha\delta\beta\delta\gamma h$ is reduced in $g=f'\delta h$. Obviously, g is in K and, using Fact 2, it turns out that g is in P, too. By induction hypothesis, g belongs to A, and we consider the place where the rule $T\to\delta$ has been applied in a derivation for g: $T\overset{*}{\Rightarrow}m'Tm''\overset{*}{\Rightarrow}m'\delta m''\overset{*}{\Rightarrow}\ g$, where $m'\overset{*}{\Rightarrow}f'$ and $m''\overset{*}{\Rightarrow}h$. Since $T\overset{*}{\Rightarrow}m'Tm''\overset{*}{\Rightarrow}$ $\overset{*}{\Rightarrow}m'\alpha T\beta T\gamma m''\overset{*}{\Rightarrow}f$, f belongs to A. At the end, we have $A=P\cap K$ and the proof is complete.

Before comparing the families $\mathscr{L}_1\mathscr{S}\mathscr{E}$, $\mathscr{L}\mathscr{S}\mathscr{E}$ and $\mathscr{S}\mathscr{E}$, we provide some necessary conditions for a language to belong to one of them. Recall ([6]) that every infinite language in $\mathscr{S}\mathscr{E}$ contains an infinite regular set. Using Proposition 2.6 and an inductive proof, we can extend this property:

Lemma 3.1. Every infinite language in $\mathscr{L}_1\mathscr{S}\mathscr{E}$ or in $\mathscr{L}\mathscr{S}\mathscr{E}$ contains an infinite regular set.

Proposition 3.3.

a) The language $\{a^nb^n|n\geqq 0\}$ is not in $\mathscr{L}_1\mathscr{S}\mathscr{E}\cup\mathscr{L}\mathscr{S}\mathscr{E}$,
b) the language $\{a^{2^n}|n\geqq 0\}$ is not in $\mathscr{L}_1\mathscr{S}\mathscr{E}\cup\mathscr{L}\mathscr{S}\mathscr{E}$.

Proposition 3.3 b) gives the proper inclusions:

Corollary 1. $\mathscr{L}_1\mathscr{S}\mathscr{E}\subsetneqq\mathscr{C}\mathscr{S}$, $\mathscr{L}\mathscr{S}\mathscr{E}\subsetneqq\mathscr{C}\mathscr{S}$.

Using proposition 3.3 a) and the preceding results, we have:

Corollary 2. Each of the families $\mathscr{L}_1\mathscr{S}h$, $\mathscr{L}\mathscr{S}h$, $\mathscr{L}_1\mathscr{S}\mathscr{E}$ and $\mathscr{L}\mathscr{S}\mathscr{E}$ is incomparable with the family of context-free languages.

Proposition 3.4. The families $\mathscr{L}\mathscr{S}\mathscr{E}$ and $\mathscr{L}$ (EDTOL) are incomparable.

Proof. The language $\{a^{2^n}|n\geqq 0\}$ is in $\mathscr{L}$ (EDTOL) ([11]) and does not belong to $\mathscr{L}\mathscr{S}\mathscr{E}$. The language $P=\{ab, cd\}^{\text{Ш}^*_2}$ is in $\mathscr{L}\mathscr{S}\mathscr{E}$ but it does not belong to $\mathscr{L}$ (EDTOL), since it is context-free generator ([8], Proposition 3.2).

Lemma 3.2. Let L be a language in X^*, where X is of minimal cardinality, $L\in\mathscr{S}\mathscr{E}$, then
either L is regular,
or for each letter x in X, for each integer $p\geqq 0$, there exists a word f in L, such that x^p is a subword of f.

Lemma 3.3. Let L be a language in X^*, $L\in\mathscr{L}\mathscr{S}\mathscr{E}$. Then, either L is regular or the two conditions hold:

(i) there exists a letter x in X and an integer n_0 such that, for each integer $p\geqq 0$, a word $f=gh$ can be found in L, where $|g|\leqq n_0+p$ and x^p is a subword of g.

(ii) there exists a letter y in X such that, for each integer $p\geqq 0$, a word $f=gh$ can be found in L, where $|g|\geqq p$ and y^p is a subword of h.

We now consider languages over a fixed ordered alphabet $X=\{a_1, \ldots, a_n\}$ with $n=2$, satisfying:

(*) There exist integers $k_1, \ldots, k_{n-1}$ in $\mathbf{Z}$, such that for each word f in L, $|f|_{a_i}-|f|_{a_{i+1}}=k_i$, $1\leqq i\leqq n-1$.

Lemma 3.4. Let L be a language in X^*, satisfying the property $(*)$ above.

a) if L is in $\mathscr{L}_1\mathscr{S}\mathscr{E}$, then $L\cap a_1^* a_2^* \ldots a_n^*$ is a finite set,
b) If L is in $\mathscr{L}\mathscr{S}\mathscr{E}$ and $n \geqq 3$, then $L\cap a_1^* a_2^* \ldots a_n^*$ is a finite set.

We can now state the main result of this section:

Proposition 3.5.

The families $\mathscr{L}_1\mathscr{L}\mathscr{E}$, $\mathscr{L}\mathscr{S}\mathscr{E}$ and $\mathscr{S}\mathscr{E}$ are pairwise incomparable.
The families $\mathscr{L}_1\mathscr{S}h$, $\mathscr{L}\mathscr{S}h$ and Shuf are pairwise incomparable.

Proof.

— The language $L=(abc)^{\sqcup\!\sqcup_2^*}$ is in $\mathscr{L}\mathscr{S}h$ and it is easy to see that L is not regular. Moreover, if b^p is a subword of a word in L, then $p \leqq 3$. Using Lemma 3.2, we obtain: $L \notin \mathscr{S}\mathscr{E}$.

— The language $M=(abc)^{\sqcup\!\sqcup^*}$ is in Shuf and $M\cap a^*b^*c^*$ is equal to $\{a^n b^n c^n \mid n \geqq 0\}$. Since M has property $(*)$, we can use Lemma 3.4 a) and b). Thus M is neither in $\mathscr{L}_1\mathscr{S}\mathscr{E}$ nor in $\mathscr{L}\mathscr{S}\mathscr{E}$.

— The restricted Dyck set $D_1'^*$ is in the families Shuf and $\mathscr{L}\mathscr{S}h$, (Proposition 1.2. b)), and $D_1'^*$ has the property $(*)$. By Lemma 3.4. a), we have: $D_1'^*$ does not belong to the family $\mathscr{L}_1\mathscr{S}\mathscr{E}$.

— The language $N=((ab)^{\sqcup\!\sqcup_1^*})^{\sqcup\!\sqcup_1^*}$ is in $\mathscr{L}_1\mathscr{S}h$ and is not regular (Proposition 3.1). Using Lemma 3.2 and Lemma 3.3 we can show that N is not in $\mathscr{L}\mathscr{S}\mathscr{E}$ and N is not in $\mathscr{S}\mathscr{E}$.

8, BD DE l'HÔPITAL
75 005 PARIS — FRANCE

References

[1] Araki, T. and N. Tokura, Flow languages equal recursively enumerable languages, Acta Informatica 15, 209—217, (1981).
[2] Beauquier, J., Générateurs algébriques et systèmes de paires itérantes, Theoretical Computer Science 8, 293—323, (1979).
[3] Ginsburg, S., Algebraic and Automata-Theoretic Properties of Formal Languages, North-Holland (1975).
[4] Harrison, M. A., Introduction to Formal Language Theory, Addison Wesley (1978).
[5] Iwama, K., Unique decomposability of shuffled strings; a fromal treatment of asynchronous time-multiplexed communication, 5th ACM Symp. on Theory of Comput., 374—381, (1983).
[6] Jantzen, M., The power of synchronizing operations on strings, Theoretical Computer Science 14, 127—154, (1981).
[7] Latteux, M., Cônes rationnels commutatifs, Journal of Computer and System Sciences 18, 307—333, (1979).
[8] Latteux, M., Sur les générateurs algébriques et linéaires, Publication du Laboratoire de Calcul de l'Université de Lille I, n° I. T. 11—79, (1979).
[9] Nivat, M., Behaviours of synchronized systems of processes, L.I.T.P. Report n° 81—64, Université de Paris 7, (1981).
[10] Ogden, W. F., W. E. Riddle and W. C. Rounds, Complexity of expressions allowing concurrency, 5th ACM Symp. on Principles of Programming Languages, 185—194, (1978).
[11] Rozenberg, G. and A. Salomaa, The Mathematical Theory of L Systems, Academic Press, (1980).
[12] Shaw, A. C., Software descriptions with flow expressions, IEEE Trans. Engrg., SE-14, 242—254, (1978).

(Received Dec. 27, 1985)