# Decomposable regular languages and the shuffle operator

Ph. Schnoebelen *

This note summarizes what I know and do not know about a class of regular language we call *decomposable languages*. Today a conjecture is that they coincide with union-products of commutative regular languages.

Decomposable languages have been used recently for the analysis of a concurrency model [LS98]. I am not a language-theory expert, and several of the results given below have been submitted by colleagues to whom I mentioned the problem. I hope that submitting these open questions to the EATCS community will prompt some readers to tackle them, and hopefully solve them. Any comment, suggestion, ... , is welcome. Write to `phs@lsv.ens-cachan.fr`.

## Notations

$\Sigma = \{a, b, \ldots\}$ is a finite alphabet and $L, L', M, \ldots$ denote languages over $\Sigma$, i.e. subsets of $\Sigma^*$. $\varepsilon$ denotes the empty word. $L.M$ is the concatenation of two languages. Recall that the *shuffle* $w \sqcup\!\sqcup w'$ of two finite words is the set of all words one can obtain by interleaving $w$ and $w'$ in an arbitary way. E.g. $abc \sqcup\!\sqcup d = \{abcd, abdc, adbc, dabc\}$.

## 1 Sequential and parallel decompositions of a language

Our starting point is

**Definition 1.1. [LS98]** *We say*
• $\{(L_1, L'_1), \ldots, (L_m, L'_m)\}$ *is a (finite)* sequential decomposition *of* $L$ *iff for all* $u, v \in \Sigma^*$ *we have*

$$u.v \in L \quad \textit{iff} \quad (\textit{for some } 1 \le i \le m, u \in L_i \textit{ and } v \in L'_i).$$

• $\{(L_1, L'_2), \ldots, (L_m, L'_m)\}$ *is a (finite)* parallel decomposition *of* $L$ *iff for all* $u, v \in \Sigma^*$ *we have*

$$L \cap (u \sqcup\!\sqcup v) \neq \varnothing \quad \textit{iff} \quad (\textit{for some } 1 \le i \le m, u \in L_i \textit{ and } v \in L'_i).$$

Some remarks may help understand Definition 1.1. Observe that a sequential decomposition $\{(L_i, L'_i) \mid i = \ldots\}$ of $L$ must apply to all possible ways of splitting a word in $L$. It even applies to a decomposition $u.v$ with $u = \varepsilon$ (or $v = \varepsilon$), hence one of the $L_i$'s (and one of the $L'_i$'s) contains $\varepsilon$.

*Lab. Spécification & Vérification, ENS de Cachan & CNRS UMR 8643, 61, av. Pdt. Wilson, 94235 Cachan Cedex France, email: `phs@lsv.ens-cachan.fr`.

Sequential and parallel decompositions look similar, but $w \sqcup\!\sqcup w'$ usually contains several elements: when $w \in L$ can be decomposed as a shuffle of some $u$ and some $v$, there must be a $(L_i, L'_i)$ for $(u, v)$. Reciprocally, when $u \in L_i$ and $v \in L'_i$, there must be some way of shuffling them into some $w \in L$.

Hence, while we have $L_i.L'_i \subseteq L$ in sequential decompositions, we don't ask for $(L_i \sqcup\!\sqcup L'_i) \subseteq L$ in parallel decompositions, and in general it does not hold.

**Example 1.2.** *Write $L_E$ (resp. $L_O$) for the language of all words with even length (resp. odd length). Then $L_E$ admits a finite decomposition into $\{(L_E, L_E), (L_O, L_O)\}$. This is both a sequential and a parallel decomposition.*

**Example 1.3.** *A sequential decomposition of $\Sigma^+$ is $\{(\Sigma^+, \Sigma^+), (\Sigma^+, \{\varepsilon\}), (\{\varepsilon\}, \Sigma^+)\}$. This is also a parallel decomposition.*

**Example 1.4.** *Consider the language $L = \{abc\}$. It contains only one word. A sequential decomposition is*

$$\{(\{\varepsilon\}, \{abc\}),\ (\{a\}, \{bc\}),\ (\{ab\}, \{c\}),\ (\{abc\}, \{\varepsilon\})\}.$$

*A parallel decomposition of $L$ needs more pairs:*

$$\{(\{\varepsilon\}, \{abc\}),\ (\{a\}, \{bc\}),\ (\{ab\}, \{c\}),\ (\{abc\}, \{\varepsilon\}),\ (\{b\}, \{ac\}),\ (\{ac\}, \{b\})\}.$$

Not all $L \subseteq \Sigma^*$ admit finite decompositions, even in the regular case.

**Example 1.5.** *$L = (ab)^*$ is not decomposable.*

*Proof.* Assume $\{(L_1, L'_1), \ldots, (L_m, L'_m)\}$ is a parallel decomposition of $L$. Then for every $k \in \mathbb{N}$, there is a shuffling of $a^k$ and $b^k$ in $L$. Hence there must be a $i_k$ s.t. $a^k \in L_{i_k}$ and $b^k \in L'_{i_k}$. Now, because the $i_k$'s can only take a finite number of distinct values, there must be some $i_k = i_{k'}$ with $k \neq k'$, and then there must exist a shuffling of $a^k$ and $b^{k'}$ in $L$, contradicting $L = (ab)^*$. □

## 2 Finite decomposition systems

In [LS98] we use finite decompositions in a recursive way. Given some $L$, we decompose it into some $(L_i, L'_i)$'s, but then we also want to decompose the $L_i$'s and the $L'_i$'s, and so on. Hence the following

**Definition 2.1.** **[LS98]** *Consider a* finite *family $\mathbb{L} = \{L_1, \ldots, L_n\}$ of languages over $\Sigma$.*
*- $\mathbb{L}$ is a* sequential decomposition system *iff every $L \in \mathbb{L}$ admits a sequential decomposition only using $L_i$'s from $\mathbb{L}$,*
*- $\mathbb{L}$ is a* parallel decomposition system *iff every $L \in \mathbb{L}$ admits a parallel decomposition only using $L_i$'s from $\mathbb{L}$,*
*- $\mathbb{L}$ is a* finite decomposition system *iff it is both a sequential and a parallel decomposition system.*

This ensures that it is possible to only use a finite number of different languages, and still decompose recursively *ad infinitum.*

We are interested into the languages that appear into such finite decomposition systems. We call them *decomposable languages.*

**Open problem: Which languages are decomposable ?**

Some **partial results** are known (we prove them in the following sections):

1. all decomposable languages are regular but not all regular languages are decomposable.
2. finite languages, cofinite languages and commutative regular languages are decomposable.
3. the family of decomposable languages is closed by union, concatenation, shuffle.
4. it is not closed by complementation or Kleene star.
5. the commutative closure of a decomposable language is decomposable (hence regular).

Some **open questions/conjectures** can be useful starting points:

1. Is the class of decomposable languages closed by intersection ?
2. Are decomposable languages closed under some family of (inverse-) morphisms ?
3. Do decomposable languages coincide with union-products of commutative regular languages ?

# 3 Basic necessary and sufficient conditions

To begin with, simply being a *sequential* decomposition system entails regularity. Recall that the *syntactic congruence* $\equiv_L$ associated to a language $L \subseteq \Sigma^*$ is given by

$$w_1 \equiv_L w_2 \overset{\text{def}}{\Leftrightarrow} \text{for all } u, v \in \Sigma^*,\ uw_1v \in L \text{ iff } uw_2v \in L.$$

A standard result states that $L$ is regular iff $\equiv_L$ has finite index.

**Lemma 3.1.** (Indicated by O. Carton) *All decomposable languages are regular.*

*Proof.* Assume $\mathbb{L}$ is a sequential decomposition system and write $u \equiv_{\mathbb{L}} v$ when for any $L_i \in \mathbb{L}$, $u \in L_i \Leftrightarrow v \in L_i$. Clearly, $\equiv_{\mathbb{L}}$ is an equivalence with finite index.

Now assume $u \equiv_{\mathbb{L}} v$. Then, for any $w$ and any $L \in \mathbb{L}$, $uw \in L$ implies $vw \in L$ as a consequence of the existence of a sequential decomposition of $L$. Hence $u \equiv_{\mathbb{L}} v$ implies $uw \equiv_{\mathbb{L}} vw$ for all $w$ (and, by a similar argument, $wu \equiv_{\mathbb{L}} wv$).

Hence $\equiv_{\mathbb{L}}$ coincides with $\bigcap_{L\in\mathbb{L}} \equiv_L$. The corollary is that the syntactic congruences of the $L_i$'s in $\mathbb{L}$ all have finite index. Hence all $L_i$'s are regular. ☐

We already saw that not all regular languages are decomposable (e.g. $(ab)^*$ is not). Still, we can display families of decomposable languages.

Two words $u$ and $v$ are *commutatively equivalent* (also, Parikh equivalent), written $u \sim_P v$, if $v$ is a permutation of $u$. E.g. $abcd \sim_P bdca$. We write $c(u)$ for $\{v \mid u \sim_P v\}$ and $c(L) \overset{\text{def}}{=} \bigcup_{u\in L} c(u)$ for the commutative closure of $L$. We say a language is *commutative* if it is closed w.r.t. commutative equivalence, i.e. if $L = c(L)$.

**Lemma 3.2.** *If $\{(L_i, L'_i) \mid i = \dots\}$ is a parallel decomposition of some $L$ then $\{(c(L_i), c(L'_i)) \mid i = \dots\}$ is a sequential decomposition of $c(L)$.*

*Proof.* First observe that $c(uv) = c(u) \sqcup c(v)$. Then $uv \in c(L)$ iff $c(uv) \cap L \neq \emptyset$ iff $c(u) \sqcup c(v) \cap L \neq \emptyset$ iff $\exists u' \in c(u), v' \in c(v)$ s.t. $u' \sqcup v' \cap L \neq \emptyset$ iff $\exists u' \in c(u), v' \in c(v)$ s.t. for some $i$, $u' \in L_i, v' \in L'_i$, iff for some $i$, $u \in c(L_i), v \in c(L'_i)$. $\square$

**Proposition 3.3.** *All commutative regular languages are decomposable.*

*Proof.* Assume $L$ is a regular language. $\equiv_L$, its syntactic congruence, partitions $\Sigma^*$ into a finite number of languages: $\Sigma^* = L_1 + \cdots + L_k$ and $L$ (and any $L_j$) admits a sequential decomposition using only these $L_i$'s so that $\mathbb{L} \stackrel{\text{def}}{=} \{L, L_1, \ldots, L_k\}$ is a sequential decomposition system.

Now we only have to notice (1) that if $L$ is commutative, then $\equiv_L$ contains $\sim_P$, so that the $L_i$'s are commutative too, and (2) that a sequential decomposition system containing only commutative languages is also a parallel decomposition system (lemma 3.2). $\square$

$L_E$ and $L_O$ from example 1.2 are commutative regular languages. Their definitions rely on *lengths* of words so that closure w.r.t. $\sim_P$ is guaranteed.

More generally, commutative regular languages have simple "letter-counting" definitions: Presburger formulas over their Parikh image. Indeed, assume $|\Sigma| = k$ and associate to any $w \in \Sigma^*$ its Parikh's vector $P(w)$, a $k$-tuple of integers recording how many $a_1$'s occur in $w$, how many $a_2$'s, up to how many $a_k$. E.g. $P(a_1a_2a_3a_2a_2a_5) = \langle 1, 3, 1, 0, 1 \rangle$. For a language $L$, $P(L)$ is a subset of $\mathbb{N}^k$. A classic result is

**Proposition 3.4.** *$L$ is a commutative regular language iff it can be written as $P^{-1}(K)$ for some semi-linear subset $K$ of $\mathbb{N}^k$.*

However, begin a commutative regular language is not a necessary condition for decomposability. Our example 1.4 is not commutative. More generally

**Proposition 3.5.** *All finite languages are decomposable.*

## 4 Closure properties

The family of decomposable languages enjoys some closure properties:

**Proposition 4.1.** *If $L$ and $M$ are decomposable then $L \cup M$ is.*

*Proof.* This is quite easy. A sequential (resp. parallel) decomposition of $L \cup M$ is obtained by taking the union of a sequential (resp. parallel) decomposition of $L$ and one of $M$. Hence if $L$ belongs to a finite decomposition system $\mathbb{L}$, and $M$ belongs to some $\mathbb{M}$, $\mathbb{L} \cup \mathbb{M} \cup \{L \cup M\}$ is a finite decomposition system. $\square$

**Proposition 4.2.** *If $L$ and $M$ are decomposable then $L.M$ is.*

*Proof.* Assume $L$ belongs to the finite decomposition system $\mathbb{L}$, and $M$ belongs to $\mathbb{M}$. Define

$$\mathbb{L} \otimes \mathbb{M} \stackrel{\text{def}}{=} \mathbb{L} \cup \mathbb{M} \cup \{L_i.M_j \mid L_i \in \mathbb{L}, M_j \in \mathbb{M}\}$$

This is a finite decomposition system (containing $L.M$). We let the reader check that if some $L \in \mathbb{L}$ (resp. some $M \in \mathbb{M}$) has a parallel decomposition of the form $\{(L_i, L'_i) \mid i = \ldots\}$ (resp. $\{(M_j, M'_j) \mid j = \ldots\}$) then a parallel decomposition of $L.M$ is simply $\{(L_i.M_j, L'_i.M'_j) \mid i = \ldots, j = \ldots\}$.

Sequential decompositions are more involved. From $\{(L_i, L'_i) \mid i = \ldots\}$ (resp. $\{(M_j, M'_j) \mid j = \ldots\}$) for $L$ and $M$, we take $\{(L_i, L'_i.M) \mid i = \ldots\} \cup \{(L.M_j, M'_j) \mid j = \ldots\}$ as the sequential decomposition of $L.M$ in $\mathbb{L} \otimes \mathbb{M}$. $\square$

We can now see that

$$L \stackrel{\text{def}}{=} b.\Sigma^* \;\cup\; \Sigma^*.a \;\cup\; \Sigma^*.(aa + bb).\Sigma^*$$

is decomposable since it is a union of concatenations of finite or commutative regular languages. $L$ is (essentially) the complement of $(ab)^*$ hence

**Proposition 4.3.** *Decomposable languages are not closed under complementation, or Kleene star.*

Another application of the closure properties is

**Proposition 4.4.** *All cofinite languages are decomposable.*

*Proof.* $L = \Sigma^* \backslash \{u_1, \ldots, u_n\}$ can be written as $a_1.(\Sigma^* \backslash \{v_1^1, \ldots, v_1^{k_1}\}) + \cdots + a_m.(\Sigma^* \backslash \{v_m^1, \ldots, v_m^{k_m}\})$ where the $v_j^k$'s are all residuals of the $u_i$'s by letter $a_j$. Hence an inductive construction of $L$ can be given, using unions, concatenations and singletons. The base of the induction requires $\Sigma^*$ and $\Sigma^+$, which are decomposable (example 1.3). $\square$

A morphism $\varphi$ associates a language $L_a$ to every $a \in \Sigma$. $\varphi(L)$ is defined in the obvious way. Decomposable languages are not closed under morphisms associating a decomposable $L_a$: $a^*$ and $L_a \stackrel{\text{def}}{=} b_1 b_2$ are decomposable but $(b_1 b_2)^*$ is not. When we further assume that $L_a$ is commutative, a counter-example is given by $L_a \stackrel{\text{def}}{=} b_1 b_2 + b_2 b_1$. Then $\varphi(a^*)$ is $(b_1 b_2 + b_2 b_1)^*$ which is not decomposable (a corollary of Proposition 5.5).

It is not known whether decomposable languages are closed under intersection. Assume $\mathbb{L} = \{L_i \mid i\}$ and $\mathbb{M} = \{M_j \mid j\}$ are decomposition systems. In general $\{L_i \cap M_j \mid i, j\}$ needs not be a decomposition system. The crucial point here is in the "$\Leftarrow$" direction of the parallel decomposition case. Assume $u \in L_i, v \in L'_i$ entails $(u \sqcup\!\sqcup v) \cap L \neq \emptyset$ and $u \in M_j, v \in M'_j$ entails $(u \sqcup\!\sqcup v) \cap M \neq \emptyset$. This means that $L$ contains some shuffling $w$ of $u$ and $v$, and $M$ contains some possibly distinct shuffling $w'$. Hence we cannot conclude that $L \cap M$ contains a shuffling of $u$ and $v$.

However, if $M$ is commutative, then containing one shuffling means containing all of them. Hence if $\mathbb{M}$ only contains commutative languages, $\{L_i \cap M_j \mid i, j\}$ is a decomposition system. Thus we have

**Proposition 4.5.** (Indicated by A. Arnold) *The intersection of a decomposable language and a commutative regular language is decomposable.*

# 5 Decomposability and commutativity

(All the results in this section have been submitted by A. Arnold who answered some of our earlier conjectures.)

**Lemma 5.1.** *If* $\{(L_i, L'_i) \mid i = \ldots\}$ *and* $\{(M_j, M'_j) \mid j = \ldots\}$ *are sequential (resp. parallel) decompositions of* $L$ *and* $M$*, then* $\{(L_i \sqcup\!\sqcup M_j, L'_i \sqcup\!\sqcup M'_j) \mid i, j = \ldots\}$ *is a sequential (resp. parallel) decomposition of* $L \sqcup\!\sqcup M$*.*

*Proof.* Consider $x \in L$ and $y \in M$. For the sequential case, we relies on $uv \in x \sqcup\!\sqcup y$ iff $x = x'x'', y = y'y'', u \in x' \sqcup\!\sqcup y', v \in x'' \sqcup\!\sqcup y''$.

For the parallel case $(u \sqcup\!\sqcup v) \cap (x \sqcup\!\sqcup y) \neq \emptyset$ iff there are some $w_1, w_2, w_3, w_4$ s.t. $x \in w_1 \sqcup\!\sqcup w_2, y \in w_3 \sqcup\!\sqcup w_4, u \in w_1 \sqcup\!\sqcup w_3, v \in w_2 \sqcup\!\sqcup w_4$. □

**Proposition 5.2. (A. Arnold)** *If* $L$ *and* $M$ *are decomposable, then their shuffle* $L \sqcup\!\sqcup M$ *is.*

*Proof.* Lemma 5.1 entails that if $\mathbb{L} = \{L_i \mid i = \ldots\}$ and $\mathbb{M} = \{M_j \mid j = \ldots\}$ are two decomposition systems, then $\mathbb{L} \sqcup\!\sqcup \mathbb{M} \stackrel{\text{def}}{=} \{L_i \sqcup\!\sqcup M_j \mid i, j = \ldots\}$ is one. □

**Lemma 5.3.** *If* $L$ *is commutative then* $\{(L_i, L'_i) \mid i = \ldots\}$ *is a parallel decomposition of* $L$ *iff it is a sequential decomposition of* $L$*.*

*Proof.* If $L$ is commutative, then for all $u, v$ we have $uv \in L$ iff $(u \sqcup\!\sqcup v) \cap L \neq \emptyset$. □

**Corollary 5.4.** *If* $\mathbb{L} = \{L_i \mid i = \ldots\}$ *is a parallel decomposition system, then* $c(\mathbb{L}) \stackrel{def}{=} \{c(L_i) \mid i = \ldots\}$ *is a finite decomposition system.*

since every $c(L) \in c(\mathbb{L})$ has a sequential decomposition in $c(\mathbb{L})$ (Lemma 5.3) and this is also a parallel decomposition (Lemma 3.2).

This entails

**Proposition 5.5. (A. Arnold)** *If* $L$ *is decomposable then* $c(L)$ *is.*

This last result can be used to prove

**Example 5.6.** $L = (ab)^*(a^* + b^*)$ *is not decomposable.*

*Proof.* Assume $L$ belongs to a system $\mathbb{L} = \{L_i \mid i = \ldots\}$. Then all $c(L_i)$ are decomposable (Prop. 5.5) and hence regular.

Since all $(ab)^n a$ are in $L$, there is a pair $(L', L'')$ in the decomposition of $L$ s.t. $L'$ contains an infinite number of $(ab)^n$'s (and $a \in L''$). Because $a \in L''$, $L'$ is a subset of $(ab)^*$. But an infinite subset of $(ab)^*$ does not have a regular commutative closure, contradicting our earlier observation that $c(L')$ must be regular. □

Since $c((ab)^*(a^* + b^*)) = (ab)^*$, this last example shows that for a regular $L$, having a regular $c(L)$ does not entail decomposability.

# 6 Union-products of commutative regular languages

An *union-product of commutative regular languages*, shortly a upc, is any language finitely obtained from commutative regular languages using only union and concatenation ("product"). Thanks to distributivity, they can be written as $\bigcup_i C_i^1 \ldots C_i^{k_i}$ where all $C_i^j$'s are commutative regular.

Because a singleton letter $\{a\}$ is a commutative regular language, finite languages are upc's. All upc's are decomposable and we have no example of a decomposable $L$ that is not a upc. Hence the following

**Conjecture 6.1.** *Decomposable languages are exactly the upc languages.*

Notice that

**Proposition 6.2.** (Indicated by A. Arnold) *Upc's are closed by intersection.*

*Proof.* It is sufficient to consider the case of the intersection of two products $(C_1 \ldots C_n) \cap (D_i \ldots D_m)$ where the $C_i$ and $D_j$'s are commutative. The proof is by induction on $n+m$. Assume $n, m > 1$ and a sequential decomposition of $C_1$ leads to $C_1 = \bigcup_i L_i.L'_i$. Note that the $L_i$'s and the $L'_i$'s are commutative. Then

$$(C_1 \ldots C_n) \cap (D_1 \ldots D_m) = \bigcup_i (L_i \cap D_1).\big((L'_i.C_2 \ldots C_n) \cap (D_2 \ldots D_m)\big)$$

where we can see the right-hand side is a upc thanks to the induction hypothesis. □

## References


[LS98] D. Lugiez and Ph. Schnoebelen. The regular viewpoint on PA-processes. In *Proc. 9th Int. Conf. Concurrency Theory (CONCUR'98), Nice, France, Sep. 1998*, volume 1466 of *Lecture Notes in Computer Science*, pages 50–66. Springer, 1998.