ELSEVIER

Discrete Applied Mathematics 126 (2003) 129–165

DISCRETE APPLIED MATHEMATICS

A non-ambiguous decomposition of regular languages and factorizing codes ☆

Marcella Anselmo

Dipartimento di Informatica ed Appl., Università di Salerno, I-84081 Baronissi (SA), Italy

Received 5 June 2000; received in revised form 10 August 2001; accepted 24 September 2001

Abstract

Given languages $Z, L \subseteq \Sigma^*$, Z is L-decomposable (finitely L-decomposable, resp.) if there exists a non-trivial pair of languages (finite languages, resp.) (A, B), such that $Z = AL + B$ and the operations are non-ambiguous. We show that it is decidable whether Z is L-decomposable and whether Z is finitely L-decomposable, in the case Z and L are regular languages. The result in the case $Z = L$ allows one to decide whether, given a finite language $S \subseteq \Sigma^*$, there exist finite languages C, P such that $SC^*P = \Sigma^*$ with non-ambiguous operations. This problem is related to Schützenberger's Factorization Conjecture on codes. We also construct an infinite family of factorizing codes.

Keywords: Formal languages; Non-ambiguous factorizations; Codes

1. Introduction

In the theory of formal languages the problem of *decomposing* languages is central. The aim is to simplify the structure of languages of a certain type. Also the reverse problem of *composing* languages is important in order to construct more complicated languages from simple ones, while preserving some particular properties.

There is a wide variety of problems dealing with language decomposition in the literature. We will equivalently call them *factorization problems*. A first factorization problem was given in 1965 by Paz and Peleg. They asked whether every regular language decomposes as a product of a finite number of stars and primes (see [29]).

☆ This work was partially supported by "Progetto Cofinanziato 60% MURST: Modelli di calcolo innovativi: metodi sintattici e combinatori".

E-mail address: anselmo@dia.unisa.it (M. Anselmo)

PII: S0166-218X(02)00193-2

A positive answer was given in [15]. In the same paper it was posed another (still open) question known as the Star Removal Problem. It concerns the decomposition of a regular language L as $L=A_1^*A_2^*\cdots A_n^*B_n$ (see [14]). Other factorization problems are the ones of deciding, given language $L\subseteq\Sigma^*$, whether $L^*=(1\cup L)^n$ [38], as well as whether there exists a language X such that $L=X^n$ [33] or $L=X^{n_1}\cup\cdots\cup X^{n_k}$ [6]. The problem of characterizing languages L such that $LL=\Sigma^*$ is solved in [22], for a particular case. Other problems of decomposition of regular languages have been very recently considered in [36].

Some factorization problems require that factorizations be *non-ambiguous*. Roughly speaking, a factorization is non-ambiguous if any word decomposing in a right way, has only one such decomposition. The concept of non-ambiguity appears many times in theoretical computer science, for example concerning (non-) ambiguity of machines recognizing languages, of grammars generating languages or of operations on languages. Non-ambiguity means uniqueness in relation to the existence of a path, derivation, or decomposition for a word.

In 1991 Perrin proposed some problems [13,8] dealing with the non-ambiguous equation: $\Sigma^*=L_1L_2\cdots L_n$ (see [26] for a partial solution). The problem of factorizing a finite prefix-closed language as a non-ambiguous product of two languages is studied in [12]. Another non-ambiguous factorization problem is to find the square root of a language L, i.e. a language X such that $L=XX$ with a non-ambiguous product [10,28] or solving the polynomial equation $L=A_0+A_1X+\cdots+A_nX^n$ for given $L,A_0,A_1,\ldots,A_n\subseteq\Sigma^*$. In [21] an algorithm for decomposing any recognizable language in terms of unitary and prefix (-free) languages is given.

Some other problems of non-ambiguous factorization of languages are related to *Schützenberger's Factorization Conjecture* [37,8]. This conjecture is very crucial in the theory of codes, in view of structurally characterizing codes. Here *code* means uniquely dechifrable variable-length code, i.e. $C\subseteq\Sigma^*$ is a code if any word in Σ^* has at most one factorization as a concatenation of elements in C. It is said *maximal* if it is not a proper subset of another code. It is said *factorizing* if there exist finite languages S,P such that $SC^*P=\Sigma^*$ with non-ambiguous operations. Schützenberger's Conjecture claims that *every finite maximal code is factorizing*. The solution of this conjecture seems to be hard: it is still open after more than 30 years and the efforts of many researchers (see [37,11,13,18,19,31,34,20,39] for some partial results). In this paper we solve a problem related to the non-ambiguous equation $SC^*P=\Sigma^*$ as above, as a byproduct of the solution of the main problems here investigated.

The main problems in this paper are the following non-ambiguous language factorization problems. Consider two languages $Z,L\subseteq\Sigma^*$. Define pair (A,B) is a *decomposition* of (Z,L) if $Z=AL\cup B$ and the operations are non-ambiguous. Decomposition (A,B) is a *finite decomposition* if A,B are finite. Moreover we define Z is *L-decomposable* (*finitely L-decomposable*, resp.) if (Z,L) has a *non-trivial* decomposition (finite decomposition, resp.). We say (A,B) is non-trivial if $(A,B)\neq(\emptyset,Z)$, $(1,\emptyset)$. Indeed $Z=AL\cup B$ always holds with $(A,B)=(\emptyset,Z)$ and also with $(A,B)=(1,\emptyset)$ when $Z=L$.

Main problems: Given regular languages $Z, L \subseteq \Sigma^*$:

(1) Is Z, L-decomposable?

(2) Is Z finitely L-decomposable?

In this paper we show that the main problems are both decidable and provide non-trivial decompositions, whenever the answer is positive. The initial motivation for studying these problems came from Schützenberger's Conjecture. Even though, the topic might be of interest in its own or have some practical byproducts: for instance, in comparing, manipulating and characterizing the nucleotide sequences in some data base. We emphasize that in this paper we will be more specifically concerned with regular languages. Given regular languages Z and L, deciding whether Z is L-decomposable is not so difficult. More complex is to decide whether Z is finitely L-decomposable. This one will be the problem more extensively studied in this paper. We solve it and provide a set of finite decompositions, in the case when a finite decomposition exists. Such finite decompositions are minimal in length (Proposition 45), and maximal in the sense specified in Proposition 47. Examples of the construction are given at the end of Section 6.

The search of and the construction of a finite decomposition start from the observation (Theorem 8) of the recursive nature of the problem: if Z is finitely L-decomposable and (A, B) is a finite decomposition then also languages $a^{-1}Z \backslash L$ are finitely L-decomposable, for any a in the prefix-part $PP(A)$ of A. We construct (A, B) starting from $PP(A)$. We notice (Theorem 22) that the search of $PP(A)$, for some finite decomposition (A, B) of (Z, L), can be restricted to a finite set, called $MIN(Z, L)$. The set $MIN(Z, L)$ is defined referring to a deterministic automaton recognizing Z. Moreover we find that also the search of finite decompositions can be restricted to a sub-class of them (Proposition 43). Using these considerations (and some others more) an algorithm can be designed for deciding the finite decomposability of a regular language and for eventually constructing some finite decompositions. Observe that, when looking for a finite decomposition (A, B), we focus our attention on A. Language B is step by step constructed dependently to words inserted in A.

We then consider the case when a finite decomposition does not exist. Remark that in Proposition 6 we prove that it is decidable whether Z is L-decomposable, and that an (infinite) decomposition can be easily provided, when Z, L are regular languages. However we generalize the main construction in order to obtain an algorithm that provides special (infinite) decompositions. These decompositions have some property of maximality, as specified in Proposition 58. Examples are given in Sections 7.1 and 7.2.

We also consider the question of how many non-trivial decompositions (finite decompositions, resp.) a given pair (Z, L) can have. We show that if $Z = L$ and (Z, L) has a non-trivial decomposition (finite decomposition, resp.) then (Z, L) has an infinite number of non-trivial decompositions (finite decompositions, resp.) (Corollary 63). On the contrary, when $Z \neq L$, then (Z, L) can admit only a finite number of non-trivial decompositions (Example 65).

Further, Corollary 63 suggests a way to construct an infinite family of non-trivial decompositions (finite decompositions, resp.) of (L, L), starting from one of them. The construction is based on an operation here introduced and called *substitution*. As a consequence, this operation allows us to construct an infinite family of factorizing codes, starting from one of them. Substitution is indeed an operation preserving the property of being a factorizing code, just as composition [8].

A byproduct of the solution of the main problems is the solution of the following problem related to Schützenberger's Conjecture, firstly considered in [18].

Problem SCP: Given finite language $S \subseteq \Sigma^*$, do there exist finite languages C, P, with C maximal code, such that $SC^*P = \Sigma^*$ with non-ambiguous operations?

A language S for which Problem SCP has positive answer is said a *polynomial having solutions* in [18] and a *strong factorizing language* in [3]. Remark that exchanging the roles of S and P would provide a dual problem. Let us recall some known results. First: if $SC^*P = \Sigma^*$ in a non-ambiguous way then C is necessarily a maximal code [37,8]. Further if finite languages S, C, P satisfy the non-ambiguous equation $SC^*P=\Sigma^*$, then $SZ = \Sigma^*$ in a non-ambiguous way with $Z = C^*P$. Then, given regular language S, it is decidable whether there exists a language Z such that $SZ=\Sigma^*$ with non-ambiguous operations [4,5]. Finally, when such Z exists, then Z is unique, regular and we can construct an automaton recognizing Z from an automaton recognizing S [4,5]. Therefore we can solve Problem *SCP* if we prove that it is decidable whether a regular language Z decomposes in a non-ambiguous way as $Z = C^*P$ with C, P finite languages. The decidability of this problem (Theorem 66) follows from the main result of this paper (Theorem 44) and from the observation that $Z = C^*P$ iff $Z = CZ + P$.

Proofs in this paper widely use (*formal power*) *series* theory. Roughly a series is a generalization of a language, where a "number" is associated with each word (see Section 2.2 for more details). Furthermore, we associate labelled trees to decompositions (see Section 5), as a tool for proving results in Sections 6 and 7. We also define (Definitions 39 and 53) labelled trees that represent all recursive calls of the procedure FIND-FIN-DEC (FIND-MAX-DEC, resp.) for computing a given finite decomposition (decomposition, resp.) of a given input. These definitions are useful for proving that the corresponding procedures are correct and always stop.

In this paper we do not consider complexity aspects of presented problems and algorithms.

The paper is organized as follows. Section 2 recalls basic definitions and notations used in the paper. In Section 3 the main problems are stated. Section 4 contains some theoretical results on finite decomposability and Section 5 a graphical representation of a decomposition by means of labelled trees. The main construction of finite decompositions is presented in Section 6. Section 7 contains a generalization of the construction to the case when a finite decomposition does not exist. Section 8 shows how many decompositions pair (Z, L) can have. Last section contains the results concerning strong factorizing languages and factorizing codes.

A preliminary and partial version of this paper is contained in [1].

2. Background and notations

Let us introduce basic definitions and notations used in the sequel. Classical references to formal languages and automata are [8,24], to formal power series are [9,21,35] and to graphs are [23].

2.1. Languages

Given finite alphabet Σ, let $\langle \Sigma^*, \cdot, 1\rangle$ be the free monoid generated by it.

The *union* of languages X, Y is $X \cup Y = \{w \mid w \in X \text{ or } w \in Y\}$; it is said non-ambiguous when $X \cap Y = \emptyset$. The *product* of languages X, Y is $XY = \{w \mid w = xy,\ x \in X,\ y \in Y\}$; it is said non-ambiguous when $w = xy = x'y'$ with $x, x' \in X,\ y, y' \in Y$ implies $x = x'$ and $y = y'$. The *star* of language X is $X^* = 1 \cup X \cup X^2 \cup \cdots$ and it is said non-ambiguous when all unions and products in the definition are non-ambiguous. In this paper $X \subset Y$ means that $X \subseteq Y$ and $X \neq Y$.

Recall that word $x \in \Sigma^*$ is a *prefix* of a word $w \in \Sigma^*$, in symbols $x \leqslant w$, if $w = xy$, with $y \in \Sigma^*$; it is a *proper prefix*, in symbols $x < w$, if $y \in \Sigma^+$. Notation $Pref(X)$ is used for the set of proper prefixes of words in X. Language X is *prefix* (*-free*) if, whenever $w, w' \in X,\ w \neq w'$, then neither $w \leqslant w'$ nor $w' \leqslant w$. Given languages X, Y, pair (X, Y) is *prefix-free* if no word in X is prefix of a word in Y and conversely. Remark that if language X is prefix-free, then for any language L, the union $\bigcup_{x \in X} xL$ is non-ambiguous. The *prefix part* of language A is the language $PP(A) = A \setminus A\Sigma^+$. Given $w \in \Sigma^*,\ X \subseteq \Sigma^*$, we denote $w^{-1}X = \{z \mid wz \in X\}$.

Note 1: For the sake of simplicity, in some cases we write $A = 1$ instead of $A = \{1\}$ and denote a set by additive notation.

2.2. Formal power series

Given alphabet Σ and semi-ring K, a (*formal power*) *series* in non-commuting variables Σ and coefficients in K is a function $s : \Sigma^* \to K$. Let $w \in \Sigma^*$. The value of s on w is denoted by (s, w) and the power series is written as a formal sum $s = \sum_{w \in \Sigma^*} (s, w)w$. The *sum* of series s, s' is defined by $(s + s', w) = (s, w) + (s', w)$ and has 0 as its identity. The *product* of series s, s' is defined by $(ss', w) = \sum_{w=xy} ((s, x)(s', y))$ and has 1 as its identity. The *star* s^* of series s such that $(s, 1) \neq 0$ is defined by $s^* = 1 + s + s^2 + \cdots$. We have that $s^* = (1 - s)^{-1}$. For a word x and a series s, the series $x^{-1}s$ is defined by $(x^{-1}s, w) = (s, xw)$. Notice that $x^{-1}(s + s') = x^{-1}s + x^{-1}s'$.

The *characteristic series* of language $X \subseteq \Sigma^*$, denoted $\underline{X}$, is the power series mapping words belonging to X to 1, and words not belonging to X to 0. Note that $\underline{\{1\}} = 1$, $\underline{\emptyset} = 0$. Using this formalism, we have that union $X \cup Y$ is non-ambiguous iff $\underline{X \cup Y} = \underline{X} + \underline{Y}$; product XY is non-ambiguous iff $\underline{XY} = \underline{X} \cdot \underline{Y}$; star X^* is non-ambiguous iff $\underline{X^*} = (\underline{X})^*$.

Note 2: In this paper an upper case letter denotes a language in formulas with set-symbols (such as $\subseteq, \in, \cap, \setminus, \cdots$); it denotes the characteristic series of the language in formulas with series-symbols (such as $+, \cdot, ^*$).

2.3. Graphs

An (*undirected*) *graph* G is a pair (V, E) where V is the (finite or infinite) set of *vertices* and E consists of unordered pairs of vertices, called *edges*. A (*rooted*) *tree* is a connected, acyclic, undirected graph $T = (V, E)$ in which one of the vertices is distinguished from the other ones and called the *root* of the tree. Tree T can also be given by $T = (V, child)$ where $child : V \to 2^V$ and $E = \{(i, j) \mid i, j \in V, j \in child(i)\}$.

A *labelled tree* with labels in a semi-ring K is a triplet $T=(V, child, lab)$ where $(V, child)$ is a tree and $lab(i,j)\in K$ is the *label* of edge (i,j). Let T be a tree (labelled tree, resp.) and i a vertex. A *path* in T is a sequence $(i_0,i_1)(i_1,i_2)\cdots(i_{n-1},i_n)$ of consecutive edges. Vertex i is a *leaf* if $child(i)=\emptyset$; it is an *internal vertex*, otherwise. An *ancestor* of i is any vertex on the unique path from the root to i. The *depth* of i is the length of the path from the root to i. The depth of T is either the maximum depth of a leaf of T, if it is finite, or ∞ otherwise. The breadth of i is either the cardinality of $child(i)$, if it is finite, or ∞ otherwise. The breadth of T is either the maximum breadth of its vertices, if it is finite or ∞, otherwise.

2.4. Automata

A (*finite*) *automaton* over Σ, is a quadruple $\mathrm{Aut}=(Q,1,\delta,F)$, where Q is a finite set of *states*, $1\in Q$ is the *initial state*, $F\subseteq Q$ is the set of *final states* and $\delta: Q\times\Sigma\to 2^Q$ is the *transition function*. A *transition* is any $(q,\sigma,p)\in Q\times\Sigma\times Q$ such that $\delta(q,\sigma)=p$. The automaton Aut is *deterministic* if $\delta: Q\times\Sigma\to Q$. Note that, in general, we do not require the transition function δ to be total, i.e., to be defined for every pair in $Q\times\Sigma$. If δ is total then we call Aut a *complete* automaton. Remark that any automaton can be *completed*. It is sufficient to add a non-final state s, called a *sink*, and to extend δ to pairs $(q,\sigma)\in Q\times\Sigma$, for which δ was not defined, by: $\delta(q,\sigma)=s$.

A *path* is a sequence $p=(q_1,\sigma_1,q_2)(q_2,\sigma_2,q_3)\cdots(q_n,\sigma_n,q_{n+1})$ of consecutive transitions. The *label* of p is $\sigma_1\sigma_2\cdots\sigma_n$. A *sub-path* of p is any sequence of consecutive transitions $(q_i,\sigma_i,q_{i+1})(q_{i+1},\sigma_{i+1},q_{i+2})\cdots(q_j,\sigma_j,q_{j+1})$, for $1\leqslant i\leqslant j\leqslant n$. Path p is *successful* if $q_1=1$ and $q_{n+1}\in F$. The *language* $L(\mathrm{Aut})$ *recognized* by Aut is the set of the labels of all successful paths in Aut. A language is said *regular* if it is recognized by a finite automaton. It is well-known that emptiness, finiteness of a regular language and inclusion between regular languages are all decidable problems [24].

An automaton is *trim* if every state q is both *accessible* (there exists a path from 1 to q) and *coaccessible* (there exists a path from q to a final state). We *trim* an automaton when we remove all states q that are not both accessible and coaccessible, together with all transitions (p,σ,p') with $p=q$ or $p'=q$.

Given a complete automaton $\mathrm{Aut}=(Q,1,\delta,F)$, the *complement* of Aut is the automaton $\mathrm{Aut}^c=(Q,1,\delta,Q\setminus F)$. Remark that $L(\mathrm{Aut}^c)=\Sigma^*\setminus L(\mathrm{Aut})$. The *intersection* of two complete deterministic automata $\mathrm{Aut}_A=(Q_A,1_A,\delta_A,F_A)$ and $\mathrm{Aut}_B=(Q_B,1_B,\delta_B,F_B)$ is the automaton $\mathrm{Aut}_A\otimes\mathrm{Aut}_B=(Q,1,\delta,F)$ where $Q=Q_A\times Q_B$, $1=(1_A,1_B)$, $F=F_A\times F_B$ and $\delta((q_A,q_B),\sigma)=(\delta_A(q_A,\sigma),\delta_B(q_B,\sigma))$, for all $q_A\in Q_A$, $q_B\in Q_B$. Remark that $L(\mathrm{Aut}_A\otimes\mathrm{Aut}_B)=L(\mathrm{Aut}_A)\cap L(\mathrm{Aut}_B)$.

Given an automaton $\mathrm{Aut}=(Q,1,\delta,F)$, we say that $p,q\in Q$ are *distinguishable* if $\exists x\in\Sigma^*$ such that $\delta(p,x)\in F$ and $\delta(q,x)\notin F$, or vice versa. A classical minimization algorithm is based on (not) distinguishable states (cf. [24]).

Given word $w\in\Sigma^*$ and deterministic automaton $\mathrm{Aut}=(Q,1,\delta,F)$, we use the following notations:

- $q\cdot w$ is the ending state of the (unique) path from q labelled w
- $q(w)=1\cdot w$

- $p(w)$ is the (unique) path from the initial state labelled w
- $q \cdot L = \{q \cdot w \mid w \in L\}$
- $L_{q,F}$ is the set of the labels of all paths from q to a state in F.

3. Main problems

In this section we introduce the non-ambiguous factorization problems we deal with in this paper.

Let us consider the question: Given languages $Z, L \subseteq \Sigma^*$, do there exist languages $A, B \subseteq \Sigma^*$ such that $Z = AL + B$?

Recall that by notation, $Z = AL + B$ means $\underline{Z} = \underline{A}\,\underline{L} + \underline{B}$. Remark that if we put no restrictions on, a trivial solution to this problem is always $A = \emptyset$, $B = Z$. Further, if $Z = L$ then $A = 1$, $B = \emptyset$ is another trivial solution.

Remark 1. We have that $L = AL + B$ iff $L = (A)^*B$ (see e.g. [9,30]). Indeed $L = AL + B$ implies $1 \notin A$ and $L = AL + B$ iff $(1 - A)L = B$ iff $L = (1 - A)^{-1}B$ iff $L = (A)^*B$. Recall that A^* exists iff $1 \notin A$ and in this case $(A)^* = (1 - A)^{-1}$.

Proposition 2. *If $Z, L, A, B \subseteq \Sigma^*$ are such that $Z = AL + B$ then the following properties hold.*

(1) $B \subseteq Z$
(2) $1 \in L$ *implies* $A \subseteq Z$
(3) $1 \in L$ *implies* $A \cap B = \emptyset$
(4) $1 \in A$ *implies* $L \subseteq Z$
(5) $Z = L$ *and* $1 \in A$ *imply* $(A, B) = (1, \emptyset)$
(6) $1 \in A$ *and* $(A, B) \neq (1, \emptyset)$ *imply* $L \subset Z$
(7) $Z = L$ *and* $(A, B) \neq (1, \emptyset)$ *imply* $(1 \in L$ *iff* $1 \in B)$.

Proof. Items (1), (2), (3), (4) are straightforward. Let us prove item (5). Indeed, $Z = L$ and $1 \in A$ imply $L = (A \setminus 1)L + L + B$ which is an ambiguous equality, unless $(A, B) = (1, \emptyset)$. In order to prove item (6), note that $1 \in A$ implies $L \subseteq Z$ and the case $Z = L$ is forbidden by item (5). Item (7) follows from previous ones. □

Let us state basic definitions.

Definition 3. Given languages $Z, L \subseteq \Sigma^*$, a *decomposition* of (Z, L) is a pair (A, B), $A, B \subseteq \Sigma^*$ such that $Z = AL + B$. A decomposition (A, B) is *trivial* if $(A, B) = (\emptyset, Z)$ or $(1, \emptyset)$. Language Z is *L-decomposable* if (Z, L) has a non-trivial decomposition.

Definition 4. Given languages $Z, L \subseteq \Sigma^*$, a *finite decomposition* of (Z, L) is a decomposition (A, B) of (Z, L) with A, B finite. A finite decomposition (A, B) is *trivial* if $(A, B) = (\emptyset, Z)$ or $(1, \emptyset)$. Language Z is *finitely L-decomposable* if (Z, L) has a non-trivial finite decomposition.

Example 5. Let us consider some pairs of languages (Z,L) and study their decomposability or finite decomposability.

(1) Let $Z = a^* + ba^*$, $L = a^*$. Language Z is L-decomposable: non-trivial (infinite) decompositions are $(1, ba^*)$ and (b, a^*). Language Z is also finitely L-decomposable: finite decompositions are $(a+b, 1)$, $(aa+b, 1+a)$, $(aa+ba, 1+a+b)$, and so on.

(2) Let $Z = L$ be the language recognized by the deterministic automaton shown in Fig. 2. Using algorithm Fin-Dec of Section 6 we will find that L is finitely L-decomposable and that $(a+ab+b^2, 1)$ is a finite decomposition of (L,L) (see Section 6.1). According to Proposition 62 another non-trivial finite decomposition is $(ab+b^2+a^2+a^2b+ab^2, 1+a)$ and an infinite number of other non-trivial finite decompositions exists.

(3) Let $Z = L$ be the language recognized by the deterministic automaton shown in Fig. 3. Using algorithm Fin-Dec of Section 6 we will find that L is not finitely L-decomposable (see Section 6.2). An infinite decomposition (A,B) is computed in Section 7.1 using algorithm Max-Dec of Section 7: $(A,B) = ((a^2)^+ab + (a^2)^+ba + (a^2)^+baab + b + ba + ba^2b, (a^2)^*)$.

(4) Let $Z = a^2b^+$, $L = ab^*$. In Example 65 it is shown that the only non-trivial decompositions of (Z,L) are $(\emptyset, Z)$ and $(a, \emptyset)$. Hence Z is L-decomposable and finitely L-decomposable.

Main problems: Given regular languages $Z, L \subseteq \Sigma^*$:

(1) Is Z, L-decomposable?

(2) Is Z finitely L-decomposable?

We will show that the main problems are both decidable. Furthermore the presented decision procedures also provide some non-trivial decompositions. Let us now show a solution of main problem (1).

Proposition 6. *It is decidable whether, given regular languages Z, $L \subseteq \Sigma^*$, Z is L-decomposable.*

Proof. Language Z is L-decomposable iff there exists a non-empty word $u \in \Sigma^+$ such that $uL \subseteq Z$. In this case $(u, Z \setminus uL)$ is a non-trivial decomposition of (Z,L). The decidability of this property is easy to show. It suffices to consider an automaton recognizing Z and decide whether there exists an accessible state q such that $q = q(w)$ for some $w \in \Sigma^+$ and $L \subseteq L_{q,F}$. The decidability of this property follows from the decidability of inclusion between regular languages. □

Proposition 6 solves main problem (1). Note that the decomposition $(u, Z \setminus uL)$ provided in the proof of Proposition 6, is often not a finite decomposition. Indeed to decide whether a language is *finitely* decomposable is much more complex. This is the problem more extensively considered in the next sections. Also note that in Section 7 we will present another way of solving main problem (1), that yet provides special decompositions.

4. Finite decomposability of regular languages

In this section we consider the second main problem of the paper: Given regular languages $Z, L \subseteq \Sigma^*$, is Z finitely L-decomposable? An algorithm for solving this problem is presented in Section 6. Here we show the theoretical foundations of such algorithm. Theorem 8 shows the recursive nature of the problem: if Z is finitely L-decomposable and (A, B) is a finite decomposition then any language $a^{-1}Z \setminus L$ is finitely L-decomposable, for any a in the prefix-part $PP(A)$ of A. We will construct (A, B) starting from $PP(A)$. Theorem 22 shows that the search of $PP(A)$ can be restricted to a finite set, called $MIN(Z, L)$. The set $MIN(Z, L)$ is defined in terms of a deterministic, trim and complete automaton recognizing Z.

Firstly let us show a result concerning non-trivial decompositions. We will then precise it for non-trivial finite decompositions. Remark that the hypothesis of regularity of languages is not yet necessary.

Theorem 7. *Let $Z, L \subseteq \Sigma^*$ be languages.*

If (A, B) is a non-trivial decomposition of (Z, L) then $PP(A)$ is a prefix-free set and for any $a \in PP(A)$, $(a^{-1}A \setminus 1, a^{-1}B)$ is a decomposition of $(a^{-1}Z \setminus L, L)$. Moreover $(a^{-1}A \setminus 1, a^{-1}B)$ is not trivial if $a^{-1}A \setminus 1 \neq \emptyset$.

Conversely, let A_p be a non-empty prefix-free set such that for any $a \in A_p$, $L \subseteq a^{-1}Z$ and $a^{-1}Z \setminus L = B_a + A_aL$. Then (A, B) is a decomposition of (Z, L), where: $A = A_p \cup (\bigcup_{a\in A_p} aA_a)$ and $B = Z \setminus A_p\Sigma^* \cup (\bigcup_{a\in A_p} aB_a)$. Moreover (A, B) is not trivial if $A_p \neq 1$ or $Z \setminus L \neq \emptyset$.

Proof. Let (A, B) be a non-trivial decomposition of (Z, L).

Language $PP(A)$ is prefix-free by definition. Let $a \in PP(A)$. By hypothesis, $Z = AL + B$ and thus $a^{-1}Z = a^{-1}(AL+B) = a^{-1}AL + a^{-1}B = (a^{-1}A)L + a^{-1}B = L + (a^{-1}A - 1)L + a^{-1}B$. Notice that we have used the additivity of operator a^{-1} on series (see Section 2). Therefore $(a^{-1}A \setminus 1, a^{-1}B)$ is a decomposition of $(a^{-1}Z \setminus L, L)$. Further $a^{-1}A \setminus 1 \neq 1$ and thus $(a^{-1}A \setminus 1, a^{-1}B)$ is not trivial if $a^{-1}A \setminus 1 \neq \emptyset$.

For the converse part, let A_p be as in the hypothesis. Remark that $1 \notin A_a$ since $L \subseteq a^{-1}Z \setminus L$ does not hold (see Proposition 2). Thus $a^{-1}Z = L + B_a + A_aL$. Further Z can be decomposed with respect to A_p in the following non-ambiguous way: $Z = Z \setminus A_p\Sigma^* + Z \cap A_p\Sigma^* = Z \setminus A_p\Sigma^* + \sum_{a\in A_p} a(a^{-1}Z) = Z \setminus A_p\Sigma^* + \sum_{a\in A_p} aL + \sum_{a\in A_p} aB_a + \sum_{a\in A_p} aA_aL = AL + B$, where:

$A = A_p \cup (\bigcup_{a\in A_p} aA_a)$ and

$B = Z \setminus A_p\Sigma^* \cup (\bigcup_{a\in A_p} aB_a)$.

Moreover if $A_p \neq 1$ then $A \neq 1, \emptyset$ and thus (A, B) is not trivial. If $A = 1$ and $Z \setminus L \neq \emptyset$ then $A_a \cup B_a \neq \emptyset$ and finally (A, B) is not trivial. We also have that $1 \notin A$ iff $1 \notin A_p$. □

Let us present a result characterizing finite decompositions. This result is the core of the recursive algorithm for finding finite decompositions presented in Section 6.

Theorem 8. *Let $Z, L \subseteq \Sigma^*$ be languages.*

If (A,B) is a non-trivial finite decomposition of (Z,L) then $PP(A)$ is a prefix-free finite set such that $Z \setminus PP(A)\Sigma^*$ is finite and for any $a \in PP(A)$, $a^{-1}Z \setminus L$ is either finitely L-decomposable or (if not) finite.

Conversely, let A_p be a non-empty prefix-free finite set such that $Z \setminus A_p\Sigma^*$ is finite and for any $a \in A_p$, $L \subseteq a^{-1}Z$ and $a^{-1}Z \setminus L = B_a + A_aL$, with A_a, B_a finite. Then (A,B) is a finite decomposition of (Z,L), where: $A = A_p \cup (\bigcup_{a \in A_p} aA_a)$ and $B = Z \setminus A_p\Sigma^* \cup (\bigcup_{a \in A_p} aB_a)$. Moreover (A,B) is not trivial if $A_p \neq 1$ or $Z \setminus L \neq \emptyset$.

Proof. Let (A,B) be a non-trivial finite decomposition of (Z,L).

Language $PP(A)$ is prefix-free by definition and it is finite since it is a subset of A. Moreover $Z \setminus PP(A)\Sigma^*$ is finite because it is contained in $Z \setminus AL = B$. It remains to show that for any $a \in PP(A)$, then $a^{-1}Z \setminus L$ is either finitely L-decomposable or (if not) finite. Let $a \in PP(A)$. By hypothesis, $Z = AL + B$ and thus $a^{-1}Z = a^{-1}(AL+B) = a^{-1}AL + a^{-1}B = (a^{-1}A)L + a^{-1}B = L + (a^{-1}A - 1)L + a^{-1}B$. Notice that we have used the additivity of operator a^{-1} on series (see Section 2).

If $a^{-1}A \setminus 1 \neq \emptyset$ then $(a^{-1}A \setminus 1, a^{-1}B)$ is a non-trivial finite decomposition of $(a^{-1}Z \setminus L, L)$. Otherwise, $(\emptyset, a^{-1}B)$ is a (trivial) decomposition of $(a^{-1}Z \setminus L, L)$. Therefore $a^{-1}Z \setminus L = a^{-1}B$ and language $a^{-1}Z \setminus L$ is finite since B is finite.

For the converse part, let A_p be as in the hypothesis and for any $a \in A_p$, let $a^{-1}Z \setminus L = B_a + A_aL$, with A_a, B_a finite and possibly empty. Remark that $1 \notin A_a$ since $L \subseteq a^{-1}Z \setminus L$ does not hold (see Proposition 2). Thus $a^{-1}Z = L + B_a + A_aL$. Further Z can be decomposed in a non-ambiguous way as $Z = AL + B$, where: $A = A_p \cup (\bigcup_{a \in A_p} aA_a)$ and $B = Z \setminus A_p\Sigma^* \cup (\bigcup_{a \in A_p} aB_a)$ (see the proof of Theorem 7).

The finiteness of $A_p, Z \setminus A_p\Sigma^*, A_a, B_a$ implies the finiteness of A, B. Moreover if $A_p \neq 1$ then $A \neq 1, \emptyset$ and thus (A,B) is not trivial. If $A = 1$ and $Z \setminus L \neq \emptyset$ then $A_a \cup B_a \neq \emptyset$ and finally (A,B) is not trivial. We also have that $1 \notin A$ iff $1 \notin A_p$. $\square$

Remark 9. Any finite decomposition (A,B) of $(a^{-1}Z \setminus L, L)$ is such that $1 \notin A$ since $L \subseteq a^{-1}Z \setminus L$ does not hold (see Proposition 2).

Remark 10. Let (A,B) be a non-trivial finite decomposition of (Z,L). If $1 \in A$ then $(A \setminus 1, B)$ is a non-trivial finite decomposition of $(Z \setminus L, L)$. Indeed $Z = L + (A \setminus 1)L + B$ and $Z \setminus L = (A \setminus 1)L + B$.

From now on, we consider the case when Z, L are regular languages. We fix some notations used in this section and in the following ones:

- $Z, L \subseteq \Sigma^*$ are regular languages
- $\mathrm{Aut}_Z = (Q, 1, \delta, F)$ is a deterministic, complete and trim automaton recognizing Z
- $Q_y = \{q \in Q \mid L \subseteq L_{q,F}\}$
- A_y is the set of labels of all paths from 1 to a state of Q_y.

Remark 11. The sets Q_y, A_y are constructible from Aut_Z. For any state $q \in Q$ it is decidable whether $q \in Q_y$, that is whether $L \subseteq L_{q,F}$. Indeed $L_{q,F}$ is recognized by the automaton (Q, q, δ, F) and the inclusion between regular languages is decidable. Then A_y is the language recognized by $(Q, 1, \delta, Q_y)$.

Remark 12. For any finite decomposition (A, B) of (Z, L) we have that $A \subseteq A_y$ and $Z \setminus A\Sigma^* \subseteq B$. Indeed $w \in A$ implies $wL \subseteq AL \subseteq Z$ and $Z \setminus A\Sigma^* \subseteq Z \setminus AL = B$.

Theorem 8 does not directly provide a constructive way to decide whether Z is finitely L-decomposable. Even in the case when Z, L are regular languages, the family of all languages A_p's as in Theorem 8 could be infinite. This is the reason why we introduce the set $MIN(Z, L)$. It is a finite family and it can be used as a *test-family* for deciding whether a regular language is finitely decomposable or not (Theorem 22).

Definition 13. Family $MIN(Z, L)$ is the class of all non-empty prefix-free finite languages $A = \{a_1, a_2, \ldots, a_n\}$ such that:

(1) $A \subseteq A_y$

(2) for any $i = 1, 2, \ldots, n$, path $p(a_i) = (q_1, \sigma_1, q_2) \cdots (q_n, \sigma_n, q_{n+1})$ is such that $q_j = q_k$ with $1 \leqslant j < k \leqslant n + 1$ implies $j = 1,\ k = n + 1$ $(q_j = q_k = 1)$.

Remark that $MIN(Z, L)$ is a finite family. In some cases we write MIN instead of $MIN(Z, L)$, if no ambiguity is possible on languages Z, L referred to.

Example 14. If no loop exists on a state of $Q \setminus Q_y$ then $PP(A_y) \in MIN(Z, L)$.

In order to present the main result of this section (Theorem 22), let us give some definitions and preliminary results.

Definition 15. Let $Q' \subseteq Q$ and $p = (q_0, \sigma_0, q_1)(q_1, \sigma_1, q_2) \cdots (q_n, \sigma_n, q_{n+1})$ be a path in Aut_Z. Let i be the lowest index $1 \leqslant i \leqslant n$ such that $\exists i < k \leqslant n + 1$ and $q_k = q_i \in Q'$.

A *Q'-reduction* of p is the path obtained from p by removing the sub-path from the first occurrence of q_i to its last occurrence.

Moreover $red_{Q'}(p)$ is the path obtained from p by repeated Q'-reductions until no other Q'-reduction is possible.

Definition 16. Let $Q' \subseteq Q,\ x \in \Sigma^+,\ p(x) = (1, \sigma_0, q_1)(q_1, \sigma_1, q_2) \cdots (q_n, \sigma_n, q_{n+1})$ and w the label of $red_{Q'}((q_1, \sigma_1, q_2) \cdots (q_n, \sigma_n, q_{n+1}))$. The word $Red_{Q'}(x) \in \Sigma^+$ is either the label of $red_{Q'}(p(x))$ if it is not 1 or $\sigma_0 w$ otherwise.

Definition 17. Let $X \subset \Sigma^*$ be a prefix-free language. Language $Red_{Q'}(X)$ is the set $Red_{Q'}(X) = \{Red_{Q'}(x) \mid x \in X\}$. Moreover $Red(X)$ is the set $Red(X) = Red_Q(X)$.

Remark 18. Let X be a language. Then $\{q(a) \mid a \in Red(X)\} = \{q(x) \mid x \in X\}$.

Remark 19. Let Z, L be regular languages and $X \subseteq A_y$ be a non-empty language. Then $A = PP(Red(X)) \in MIN(Z, L)$. Indeed A is finite, non-empty, prefix-free and for any $a \in A,\ p(a) = (q_1, \sigma_1, q_2) \cdots (q_n, \sigma_n, q_{n+1})$ is such that $q_j = q_k$, with $1 \leqslant j < k \leqslant n + 1$ implies $j = 1,\ k = n + 1$, by definition of Q-reduction.

Remark 20. If $a \in \Sigma^*$ then $L_{q(a),F} = a^{-1}Z$.

Lemma 21. *Let $Z, L \subseteq \Sigma^*$ be regular languages, Z infinite and (A, B) be any finite decomposition of (Z, L).*

For any $x, t \in \Sigma^*$, $y \in \Sigma^+$ such that $xy^*t \subseteq Z$, there exists $k \geqslant 0$ such that $PP(A) \cap Pref(\{xy^k\}) \neq \emptyset$.

Conversely, for any $a \in A$, there exist $k \geqslant 0$, $x, t \in \Sigma^*$, $y \in \Sigma^+$ such that $xy^*t \subseteq Z$ and $a \in Pref(\{xy^k\})$.

Proof. Suppose by the contrary, that $\forall k \geqslant 0$, $PP(A) \cap Pref(\{xy^k\}) = \emptyset$. Then $A \cap Pref(\{xy^k\}) = \emptyset$ holds. Therefore an infinite set of words xy^it_i, with $t_i \leqslant t$, is a subset of $A \cup B$, against the finiteness of A, B.

Conversely, if Z is infinite then L is infinite. Hence for any $a \in A$, language $Z \cap a\Sigma^*$ is infinite because $aL \subseteq Z \cap a\Sigma^*$. Moreover $Z \cap a\Sigma^*$ is regular since it is the intersection of two regular languages. Therefore $\exists x, t \in \Sigma^*$, $y \in \Sigma^+$ such that $xy^*t \subseteq Z \cap a\Sigma^*$. $\square$

We are now able to present the main result of this section.

Theorem 22. *Let $Z, L \subseteq \Sigma^*$ be regular languages. Language Z is finitely L-decomposable iff $MIN(Z, L) \neq \emptyset$ and there exists $A \in MIN(Z, L)$ such that*:
(1) *$Z \setminus A\Sigma^*$ is finite*,
(2) *for any $a \in A$, $a^{-1}Z \setminus L$ is either finitely L-decomposable or (if not) finite*,
(3) *$A \neq 1$ or $Z \setminus L \neq \emptyset$.*

Proof. (*Direct implication*) Let $Z = B_f + A_fL$ where (A_f, B_f) is a non-trivial finite decomposition of (Z, L).

If $PP(A_f) = 1$ then $A = 1 \in MIN(Z, L)$ and it satisfies (1), (2), (3). Indeed $Z \setminus A\Sigma^* = Z \setminus \Sigma^* = \emptyset$ is finite, $Z \setminus L$ is either finitely L-decomposable or (if not) finite (Remark 10) and $Z \setminus L \neq \emptyset$ since otherwise $(A_f, B_f) = (1, \emptyset)$ is trivial.

If $PP(A_f) \neq 1$ then let $A = PP(Red(PP(A_f)))$. We have that $A \in MIN(Z, L)$ (Remark 19) and thus in particular $MIN(Z, L) \neq \emptyset$. Let us now show that A has properties (3), (2), (1).

(3) We have $A \neq 1$ from definition of A.

(2) Remark that for any $a \in A$, $q(a) = q(w)$ for $w \in PP(A_f)$ such that $a = Red(w)$. Then $a^{-1}Z \setminus L = L_{q(a),F} \setminus L = L_{q(w),F} \setminus L$ is either finitely L-decomposable or (if not) finite by Theorem 8.

(1) If Z is finite then $Z \setminus A\Sigma^*$ is trivially finite. We claim that if Z is infinite then $A \subseteq Pref(PP(A_f)) \cup PP(A_f)$ and thus $Z \setminus A\Sigma^* \subseteq Z \setminus PP(A_f)\Sigma^*$ is finite. Indeed for any $w \in A_f$, w is a prefix of xy^k for some $x, t \in \Sigma^*, y \in \Sigma^+, k \geqslant 0$ such that $xy^*t \subseteq Z$ (Lemma 21). Therefore for any $w \in A_f$, there exist in Aut_Z a path from $q(w)$ to a state s labelled x', a loop on s labelled y' and a path from s to a final state labelled t, for some $x' \in \Sigma^*, y' \in \Sigma^+$. Recall that for any $a \in A$, $q(a) = q(w)$ for some $w \in PP(A_f)$ such that $a = Red(w)$. The existence of such x', y', t implies (Lemma 21 again) that for any $a \in A$ there exists $w' \in PP(A_f)$ such that $w' \leqslant axy^h$, for some $h \geqslant 0$. Further it is

not the case $w' < a$, otherwise $Red(w') = w' < a$, against the definition of A. Therefore $a \leqslant w'$ and $w' \in PP(A_f)$.

(*Inverse implication*) Any $A \in MIN(Z, L)$ that has properties (1), (2) and (3), satisfies the hypothesis of the converse part of Theorem 8 and thus the claim. □

Example 23. Let $\Sigma = \{a, b\}$ and $Z = L = \Sigma^*$. Language L is finitely L-decomposable. For instance $L = A_f L + B_f$ with $A_f = aa + ab + b$, $B_f = 1 + a$. Let $\mathrm{Aut_L} = (\{1\}, 1, \delta, \{1\})$ where $\delta(1, a) = \delta(1, b) = 1$. Consider $A = PP(Red(PP(A_f)))$, as in the proof of Theorem 22. We find $A = PP(Red(aa + ab + b)) = PP(a + b) = a + b$. According to Theorem 22, $A \in MIN(L, L)$ and (1) $L \setminus A\Sigma^* = 1$ is finite; (2) $a^{-1}L \setminus L = b^{-1}L \setminus L = \emptyset$ is finite and (3) $A \neq 1$.

Remark 24. Theorem 22 in particular shows that if $MIN(Z, L) = \emptyset$ then the only finite decomposition of (Z, L) is the trivial one $(\emptyset, Z)$ if Z is finite. Observe that if $(1, \emptyset)$ is a finite decomposition of (Z, L) then $Z = L$ and $\{1\} \in MIN(Z, L)$. Also remark that $MIN(Z, L) = \emptyset$ iff $Q_y = \emptyset$.

5. A graphical representation of decompositions

In this section we present a graphical representation of decompositions. We associate a labelled tree to any decomposition. This representation will be useful in the next sections. Algorithm Fin-Dec in Section 6 constructs a finite decomposition following its associated tree, starting from the root and proceeding by increasing depth of the vertices.

Definition 25. Let $Z, L \subseteq \Sigma^*$, (A, B) be a decomposition of (Z, L) with $A = \{a_1, a_2, \ldots, a_n, \ldots\} \neq \emptyset$. The *tree associated to* (A, B) is the labelled tree $T_{A,B} = (V, child, lab)$ where:

- $V = \{0, 1, \ldots, n, \ldots\}$
- the root is 0
- $child(0) = \{j \in V \mid a_j \in PP(A)\}$
- for any $i \in V \setminus \{0\}$, $child(i) = \{j \in V \mid a_i < a_j$ and $\nexists\ a \in A$ s.t. $a_i < a < a_j\}$
- the label of edge (i, j) is $lab(i, j) = x_{i,j}$ s.t. $a_j = a_i x_{i,j}$.

Moreover the *languages associated to* $T_{A,B}$ are:

- $R_0 = Z$, $R_j = x_{i,j}^{-1} R_i \setminus L$, for any $j \in V$, $j \in child(i)$
- $B_{i,j} = \{a_i^{-1} b \in B \mid a_i < b < a_j\}$, for any internal vertex $i \in V$ and $j \in child(i)$
- $B_{i,\infty} = \{a_i^{-1} b \in B \mid a_i < b\}$, for any leaf i.

Example 26. Let (A, B) be a finite decomposition of some pair of languages (Z, L). Suppose that $A = \{a_1, a_2, \ldots, a_9\}$ where $a_1 < a_4, a_5$, $a_2 < a_6, a_7$, $a_5 < a_8, a_9$ and no other prefix relation holds among words in A. Tree $T_{A,B}$ associated to (A, B) is shown in Fig. 1.

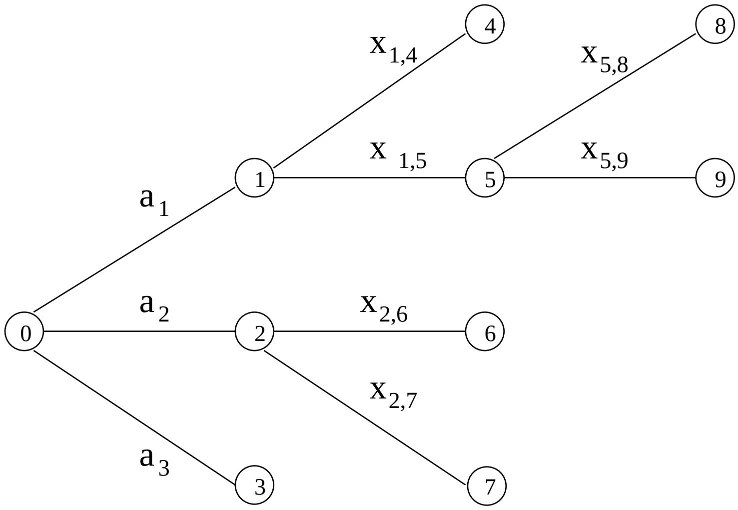

Fig. 1. The tree $T_{A,B}$.

We have that $PP(A)=\{a_1,a_2,a_3\}$, $Pref(PP(A))\cap Z=B_{0,1}\cup B_{0,2}\cup B_{0,3}$ and

$Z=Z\setminus PP(A)\Sigma^*+a_1\Sigma^*\cap Z+a_2\Sigma^*\cap Z+a_3\Sigma^*\cap Z=Z\setminus PP(A)\Sigma^*+a_1L+a_1x_{1,4}L+a_1x_{1,5}L+a_1x_{1,5}x_{5,8}L+a_1x_{1,5}x_{5,9}L+a_1B_{1,4}+a_1B_{1,5}+a_1x_{1,5}B_{5,8}+a_1x_{1,5}B_{5,9}+a_1x_{1,4}B_{4,\infty}+a_1x_{1,5}x_{5,8}B_{8,\infty}+a_1x_{1,5}x_{5,9}B_{9,\infty}+a_2L+a_2x_{2,6}L+a_2x_{2,7}L+a_2B_{2,6}+a_2B_{2,7}+a_2x_{2,6}B_{6,\infty}+a_2x_{2,7}B_{7,\infty}+a_3L+a_3B_{3,\infty}$.

We also have that:

$R_1=a_1^{-1}Z\setminus L=x_{1,4}L+x_{1,5}L+x_{1,5}x_{5,8}L+x_{1,5}x_{5,9}L+B_{1,4}+B_{1,5}+x_{1,5}B_{5,8}+x_{1,5}B_{5,9}+x_{1,4}B_{4,\infty}+x_{1,5}x_{5,8}B_{8,\infty}+x_{1,5}x_{5,9}B_{9,\infty}$;

$R_2=a_2^{-1}Z\setminus L=x_{2,6}L+x_{2,7}L+B_{2,6}+B_{2,7}+x_{2,6}B_{6,\infty}+x_{2,7}B_{7,\infty}$;

$R_3=a_3^{-1}Z\setminus L=B_{3,\infty}$;

$R_4=x_{1,4}^{-1}R_1\setminus L=B_{4,\infty}$;

$R_5=x_{1,5}^{-1}R_1\setminus L=x_{5,8}L+x_{5,9}L+B_{5,8}+B_{5,9}+x_{5,8}B_{8,\infty}+x_{5,9}B_{9,\infty}$;

$R_6=x_{2,6}^{-1}R_2\setminus L=B_{6,\infty}$;

$R_7=x_{2,7}^{-1}R_2\setminus L=B_{7,\infty}$;

$R_8=x_{5,8}^{-1}R_5\setminus L=B_{8,\infty}$;

$R_9=x_{5,9}^{-1}R_5\setminus L=B_{9,\infty}$.

We remark that $PP(A)=\{a_1,a_2,a_3\}$ is a prefix-free finite set such that $Z\setminus PP(A)\Sigma^*\subseteq B$ is finite and $\forall a\in PP(A)$, $a^{-1}Z\setminus L$ is finitely L-decomposable, as in Theorem 8. For any $i=1,2,3$ we have $a_i^{-1}Z\setminus L=R_i$. Moreover $(A_1,B_1)=(x_{1,4}+x_{1,5}+x_{1,5}x_{5,8}+x_{1,5}x_{5,9},B_{1,4}+B_{1,5}+x_{1,5}B_{5,8}+x_{1,5}B_{5,9}+x_{1,4}B_{4,\infty}+x_{1,5}x_{5,8}B_{8,\infty}+x_{1,5}x_{5,9}B_{9,\infty})$ is a finite decomposition of (R_1,L); $(A_2,B_2)=(x_{2,6}+x_{2,7},B_{2,6}+B_{2,7}+x_{2,6}B_{6,\infty}+x_{2,7}B_{7,\infty})$ is a finite decomposition of (R_2,L); $(A_3,B_3)=(\emptyset,B_{3,\infty})$ is a finite decomposition of (R_3,L).

The following Propositions 27 and 28 are a transposition of Theorems 7 and 8 in terms of the trees associated to decompositions.

Proposition 27. *Let $Z, L \subseteq \Sigma^*$, (A, B) be a non-trivial decomposition of (Z, L), $T_{A,B} = (V, child, lab)$ be the associated tree and $i \in V$ be an internal vertex.*

Then the sub-tree rooted in i is the tree associated to a non-trivial decomposition of (R_i, L).

Conversely if (A_i, B_i) is any non-trivial decomposition of (R_i, L) then the tree obtained by replacing the sub-tree rooted in i with T_{A_i,B_i} is the tree associated to a non-trivial decomposition of (Z, L).

Proof. The statement of the proposition trivially holds for $i = 0$. Now we prove it for $i \in child(0)$. The result for any internal vertex i follows by induction on the depth of i.

Observe that when $i \in child(0)$ then $x_{0,i} = lab(0, i) = a_i$ and $\{a_i \mid i \in child(0)\} = PP(A)$. Moreover $child(i) \neq \emptyset$ since i is not a leaf. Following Theorem 8, $(a_i^{-1}A \setminus 1, a_i^{-1}B)$ is a non-trivial decomposition of (R_i, L). The sub-tree of $T_{A,B}$ rooted in i is the tree associated to $(a_i^{-1}A \setminus 1, a_i^{-1}B)$.

For the converse part, let $i \in child(0)$ and (A_i, B_i) be a non-trivial decomposition of (R_i, L). We have that $Z = Z \setminus a_i\Sigma^* + a_iL + a_iR_i = Z \setminus a_i\Sigma^* + a_iL + a_i(A_iL + B_i) = (A \setminus a_i\Sigma^*)L + B \setminus a_i\Sigma^* + a_iL + a_iA_iL + a_iB_i = (A \setminus a_i\Sigma^+ + a_iA_i)L + B \setminus a_i\Sigma^* + a_iB_i$. Define $A' = A \setminus a_i\Sigma^+ \cup a_iA_i$ and $B' = B \setminus a_i\Sigma^* \cup a_iB_i$. Pair (A', B') is thus a decomposition of (Z, L). It is non-trivial since (A_i, B_i) is non-trivial. The associated tree is exactly the tree obtained by replacing the sub-tree rooted in i with T_{A_i,B_i}. □

Proposition 28. *Let $Z, L \subseteq \Sigma^*$, (A, B) be a non-trivial finite decomposition of (Z, L), $T_{A,B} = (V, child, lab)$ be the associated tree and $i \in V$ be an internal vertex.*

Then R_i is finitely L-decomposable and the sub-tree rooted in i is the tree associated to a non-trivial finite decomposition of (R_i, L).

Conversely if (A_i, B_i) is any non-trivial finite decomposition of (R_i, L) then the tree obtained by replacing the sub-tree rooted in i with T_{A_i,B_i} is the tree associated to a non-trivial finite decomposition of (Z, L).

Proof. The statement of the proposition trivially holds for $i = 0$. Now we prove it for $i \in child(0)$. The result for any internal vertex i follows by induction on the depth of i.

Observe that when $i \in child(0)$ then $x_{0,i} = lab(0, i) = a_i$ and $\{a_i \mid i \in child(0)\} = PP(A)$. Moreover $child(i) \neq \emptyset$ since i is not a leaf. The language $R_i = a_i^{-1}Z \setminus L$ is not finite since $x_{i,j}L \subseteq R_i$ for any $j \in child(i)$. Therefore R_i is finitely L-decomposable by Theorem 8. Following the proof of Theorem 8, $(a_i^{-1}A \setminus 1, a_i^{-1}B)$ is a non-trivial finite decomposition of (R_i, L). The sub-tree of $T_{A,B}$ rooted in i is the tree associated to $(a_i^{-1}A \setminus 1, a_i^{-1}B)$.

The proof of the converse part is the same as the one of Proposition 27. □

Remark 29. If (A, B) is a finite decomposition of (Z, L) and i is a leaf in $T_{A,B}$, then R_i is finite. Indeed if i is a leaf then $\nexists k \in V$ such that $a_i < a_k$. Hence $R_i = a_i^{-1}Z \setminus L = (L \cup a_i^{-1}B) \setminus L = B_{i,\infty}$ is finite since $a_iB_{i,\infty} \subseteq B$ is finite.

Example 30. Consider $Z, L, A, B, T_{A,B}$, as in Example 26. We have $child(0) = \{1, 2, 3\}$. For any $i \in \{1, 2, 3\}$, $R_i = a_i^{-1}Z \setminus L$ is finitely L-decomposable. A non-trivial finite decomposition of (R_i, L) is (A_i, B_i) as defined in Example 26. Further for any $i \in \{1, 2, 3\}$, the sub-tree rooted in i is the tree associated to (A_i, B_i). Consider now vertex 4; it is a leaf and $R_4 = B_{4,\infty}$ is finite.

The following results allow us to restrict the search of finite decompositions to a sub-class of them: if a finite decomposition exists then a finite decomposition of a special kind exists.

Definition 31. Let $T_{A,B}$ be the tree associated to decomposition (A, B) of (Z, L). Tree $T_{A,B}$ is *R-simple* if for any path from the root $0 \to i_1 \to \cdots \to i_n$ we have $R_{i_j} \neq R_{i_k}$ for any $1 \leqslant j < k \leqslant n$. Tree $T_{A,B}$ is *normal* if it is R-simple and for any internal vertex i of $T_{A,B}$, $Lab(i) \in MIN(R_i, L)$ and $R_i \setminus Lab(i)\Sigma^*$ is finite, where $Lab(i)$ denotes set $\{x_{i,j} \mid j \in child(i)\}$.

Remark 32. If $T_{A,B}$ is R-simple then any sub-tree of $T_{A,B}$ is R-simple; if $T_{A,B}$ is normal then any sub-tree of $T_{A,B}$ is normal.

Proposition 33. *Let $Z, L \subseteq \Sigma^*$. If Z is finitely L-decomposable then (Z, L) has a non-trivial finite decomposition (A, B) whose associated tree $T_{A,B}$ is R-simple.*

Proof. Suppose that the statement does not hold for some (Z, L). Let $T_{A,B}$ be the tree associated to a finite decomposition (A, B) of (Z, L) and $0 \to i_1 \to \cdots \to i_n$ be a path from the root such that $R_{i_j} = R_{i_k}$ for some $1 \leqslant j < k \leqslant n$. Suppose without loss of generality, that i_n is a leaf. If $k = n$ then $R_{i_j} = R_{i_k}$ is finite (Remark 29). We can thus remove the sub-tree rooted in i_j and obtain a tree associated to a finite decomposition of (Z, L). If $k < n$ then the sub-tree rooted in i_k is the tree associated to a finite decomposition of $(R_{i_k}, L) = (R_{i_j}, L)$ (Proposition 28). Therefore we can replace the sub-tree rooted in i_j by the one rooted in i_k and obtain a tree associated to a finite decomposition of (Z, L) (Proposition 28). Repeating such removals or replacement until there is no path $0 \to h_1 \to \cdots \to h_m$ such that $R_{h_j} = R_{h_k}$ for some $1 \leqslant j < k \leqslant m$, we obtain a tree associated to a finite decomposition of (Z, L) as required. This contradicts the initial hypothesis. $\square$

Proposition 34. *Let $Z, L \subseteq \Sigma^*$. If Z is finitely L-decomposable then (Z, L) has a non-trivial finite decomposition (A, B) whose associated tree $T_{A,B}$ is normal.*

Proof. Let Z be a finitely L-decomposable language. We show that (Z, L) has a non-trivial finite decomposition (X, Y) such that $T_{X,Y}$ is R-simple, and for $i=0$, $PP(X) = Lab(i) \in MIN(R_i, L)$, $R_i \setminus Lab(i)\Sigma^*$ is finite and for every $h \in child(i)$, the sub-tree rooted in h is R-simple. The general statement will follow by induction on depth of i.

If Z is finitely L-decomposable then (Z, L) has a non-trivial finite decomposition (X', Y') such that $T_{X',Y'}$ is R-simple (Proposition 33). Let $PP(X') = \{a_1, \ldots, a_k\}$. For any $j = 1, \ldots, k$, language $R_j = L_{q(a_j),F} \setminus L$ is finitely L-decomposable (Proposition 28).

Further the sub-tree rooted in j is R-simple (Remark 32) and is the tree associated to a finite decomposition of (R_j, L) (Proposition 28). Consider $A = PP(Red(PP(X')))$. We have $A \in MIN(Z,L)$ (Remark 19) and for any $a \in A$, $a^{-1}Z \setminus L = L_{q(a),F} \setminus L = R_j$ for some $j \in \{1,\ldots,k\}$ (Remark 18). Applying Theorem 8, we obtain a finite decomposition (X,Y) of (Z,L) where $T_{X,Y}$ is R-simple, $PP(X) = A \in MIN(R_0, L)$, $R_0 \setminus Lab(0)\Sigma^* \subseteq Y$ is finite and $\forall h \in child(0)$ the sub-tree rooted in h is R-simple. Indeed the sub-tree rooted in h is the sub-tree rooted in some $j \in \{1,\ldots,k\}$ of $T_{X',Y'}$. The decomposition (X,Y) is not trivial since $A \neq 1, \emptyset$, by definition of Red. $\square$

Next lemmas will be used in Section 7. Note that they hold for any decomposition.

Lemma 35. *Let $Z, L \subseteq \Sigma^*$, $T_{A,B} = (V, child, lab)$ be the tree associated to decomposition (A,B) of (Z,L) and i,j two vertices in $T_{A,B}$. If i is an ancestor of j in $T_{A,B}$ then $R_i = wR_j + A'L + B'$, where w is the label of the path from vertex i to vertex j and $A', B' \subseteq \Sigma^*$.*

Proof. Let us suppose, without loss of generality, that the path from i to j is $i \to i+1 \to \cdots \to j-1 \to j$. Remark that for any $h = i,\ldots,j-1$, R_h is either $R_h = w_{h,h+1}L + w_{h,h+1}R_{h+1} + R_h \setminus w_{h,h+1}\Sigma^*$, if $h \neq 0$ or $R_h = w_{h,h+1}L + w_{h,h+1}R_{h+1} + R_h \setminus w_{h,h+1}\Sigma^* + Z \setminus A\Sigma^*$, if $h = 0$. Consider the tree obtained from the sub-tree rooted in h by removing $h+1$ from $child(h)$. Using a proof similar to the one of Proposition 27, it can be shown that $(R_h \setminus w_{h,h+1}\Sigma^*, L)$ has a finite decomposition (A_h, B_h). Let us denote for any $h = i, \cdots, j, w_h = w_{i,i+1} \cdots w_{h-1,h}$. Suppose $i \neq 0$. Then $R_i = w_j R_j + \sum_{h=i+1,\ldots,j} w_h L + \sum_{h=i+1,\ldots,j-1} w_h(A_h L + B_h)$. Equality $R_i = wR_j + A'L + B'$ thus holds $w = w_j$, $A' = \sum_{h=i+1,\ldots,j} w_h + \sum_{h=i+1,\ldots,j-1} w_h A_h$ and $B' = \sum_{h=i+1,\ldots,j-1} w_h B_h$. If $i = 0$ then the equality holds with $B' = \sum_{h=i+1,\ldots,j-1} w_h B_h + Z \setminus A\Sigma^*$. $\square$

Lemma 36. *Let $Z, L \subseteq \Sigma^*$, $T_{A,B} = (V, child, lab)$ be the tree associated to decomposition (A,B) of (Z,L). Let $i_0 \in V$ and I a set of vertices in the sub-tree rooted in i_0 such that $\forall i \in I$, $R_i = R_{i_0}$. Then $\exists W, A, B \subseteq \Sigma^*$ such that $R_{i_0} = WR_{i_0} + AL + B$.*

Proof. By a proof similar to the one of Lemma 35, one can prove that $R_{i_0} = \sum_{i \in I} w_i R_i + AL + B$ for some $w_i \in \Sigma^*$, $A, B \subseteq \Sigma^*$. Since $\forall i \in I$, $R_i = R_{i_0}$, we have $R_{i_0} = WR_{i_0} + AL + B$, with $W = \sum_{i \in I} w_i$. $\square$

6. Main construction

Theorem 22 in Section 4 suggests a recursive way to construct a finite decomposition of (Z,L), when Z, L are regular languages. Indeed a finite decomposition of (Z,L) can be obtained from finite decompositions of $(a^{-1}Z \setminus L, L)$ for $a \in A$ and $A \in MIN(Z,L)$. The basis of this recursion is the case when $MIN(Z,L) = \emptyset$: if Z is finite then a finite decomposition is simply obtained, otherwise no finite decomposition exists at all. Based on these results, we design an algorithm for deciding whether (Z,L) has a non-trivial finite decomposition and for eventually providing some ones. Proposition 42 ensures that the algorithm always stops in a finite number of steps and Proposition 43 states its correctness.

We fix some notations:

- Z, L are regular languages
- Aut_Z is a deterministic, complete and trim automaton recognizing Z
- Aut_L is a deterministic, complete and trim automaton recognizing L.

We are going to sketch algorithm FIN-DEC working on input $(Z, L, \mathrm{Aut}_Z, \mathrm{Aut}_L)$. The algorithm calls procedure FIND-FIN-DEC. In procedure FIND-FIN-DEC, $\mathrm{Aut}_X = (Q, 1, \delta, F)$ denotes a deterministic, trim automaton recognizing X and Aut_a is the automaton recognizing $L_{q(a),F} \setminus L = L_{q(a),F} \cap (\Sigma^* \setminus L)$ obtained by intersection of $(Q, q(a), \delta, F)$ and the complement of Aut_L (see Section 2). We emphasize that the algorithm is written in an informal way.

FIN-DEC($Z, L, \mathrm{Aut}_Z, \mathrm{Aut}_L$)

FIND-FIN-DEC($Z, L, \mathrm{Aut}_Z, \mathrm{Aut}_L, \{Z\}$)
if FD(Z, L) $\neq \emptyset$
  then return FD(Z, L)
  else return "$\nexists$ finite decompositions"

FIND-FIN-DEC($X, L, \mathrm{Aut}_X, \mathrm{Aut}_L$, TRACK)

FD(X, L) $\leftarrow \emptyset$
if X is finite
  then $FD(X,L) \leftarrow FD(X,L) \cup \{(\emptyset, X)\}$
if $MIN(X,L) \neq \emptyset$
  then for any $A \in MIN(X,L)$ s.t. $X \setminus A\Sigma^*$ finite and
      $\forall a \in A,\ L_{q(a),F} \setminus L \notin TRACK$
    do for any $a \in A$
      do $I \leftarrow (L_{q(a),F} \setminus L, L, \mathrm{Aut}_a, \mathrm{Aut}_L,\ TRACK \cup \{L_{q(a),F} \setminus L\})$
      FIND-FIN-DEC(I)
    if $\forall a \in A, FD(L_{q(a),F} \setminus L) \neq \emptyset$
      then for any $a \in A$ and $(A_a, B_a) \in FD(L_{q(a),F} \setminus L)$
        do $A_M \leftarrow A \cup (\bigcup_{a \in A} aA_a)$
          $B_M \leftarrow X \setminus A\Sigma^* \cup (\bigcup_{a \in A} aB_a)$
          $FD(X,L) \leftarrow FD(X,L) \cup \{(A_M, B_M)\}$

Note that in the sequel, $FD(Z,L)$ denotes the set returned by FIND-FIN-DEC(Z, L, Aut_Z, $\mathrm{Aut}_L, \{Z\}$).

Remark 37. When looking for a finite decomposition (A,B) in algorithm FIN-DEC we focus on A. Language B is step by step constructed dependently to words inserted in A, in such a way it is finite too.

Our goal is now to prove that algorithm FIN-DEC always stops. Let us state a preliminary lemma and then define a labelled tree associated to (Z, L, A_M, B_M) where $(A_M, B_M) \in FD(Z,L)$. The structure of such a tree exactly mirrors the structure of recursive calls of the procedure FIND-FIN-DEC when it adds pair (A_M,B_M) to $FD(Z,L)$ with initial call to $(Z,L,\mathrm{Aut}_Z,\mathrm{Aut}_L,\{Z\})$.

Lemma 38. *Let* Aut_A, Aut_B *be deterministic, complete and trim automata. Let* $\mathrm{Aut}_0=\mathrm{Aut}_A$ *and* $\forall i \geqslant 1$, $\mathrm{Aut}_i=(Q_i,q_i,\delta_i,F_i)$ *where* $(Q_i,q_i^0,\delta_i,F_i)=\mathrm{Aut}_{i-1}\otimes\mathrm{Aut}_B$ *and* $q_i\in Q_i$. *Then* $\{L(\mathrm{Aut}_i)\mid i\geqslant 0\}$ *is finite.*

Proof. Let $\mathrm{Aut}_A=(Q_A,q_A,\delta_A,F_A)$ and $\mathrm{Aut}_B=(Q_B,q_B,\delta_B,F_B)$. We associate to each Aut_i an automaton Aut'_i with states in $Q_A\times 2^{Q_B}$ and show that for every $i\geqslant 0$, $L(\mathrm{Aut}_i)=L(\mathrm{Aut}'_i)$. Since Q_A,Q_B are finite, then $Q_A\times 2^{Q_B}$ is finite too. Thus the family $\{\mathrm{Aut}'_i\mid i\geqslant 0\}$ is finite and the goal is achieved.

Let $p,p'\in Q_i$, $p=(\dots(p_0,p_1),\dots,p_i)$, $p'=(\dots(p'_0,p'_1),\dots,p'_i)$, with $p_0,p'_0\in Q_A$, and $\forall j=1,\dots,i$, $p_j,p'_j\in Q_B$. We claim that if $\{p_1,\dots,p_i\}=\{p'_1,\dots,p'_i\}$ then p,p' are not distinguishable, i.e. $L_{p,F_i}=L_{p',F_i}$. Denote $P=\{p_1,\dots,p_i\}$. For any word $w\in\Sigma^*$, $w\in L_{p,F_i}$ iff $(p_0\cdot w\in F_A$ and $\forall j=1,\dots,i, p_j\cdot w\in F_B)$ iff $(p_0\cdot w\in F_A$ and $\forall p\in P, p\cdot w\in F_B)$ iff $(p_0\cdot w\in F_A$ and $\forall j=1,\dots,i, p'_j\cdot w\in F_B)$ iff $w\in L_{p',F_i}$.

From results of automata theory, $L(\mathrm{Aut}_i)=L(\mathrm{Aut}'_i)$, where Aut'_i is obtained by identifying (not distinguishable) states $p=(\dots(p_0,p_1),\dots,p_i)$, $p'=(\dots(p'_0,p'_1),\dots,p'_i)$, $\dots$, $p^{(n)}=(\dots(p_0^{(n)},p_1^{(n)}),\dots,p_i^{(n)})$ such that $\{p_1,\dots,p_i\}=\{p'_1,\dots,p'_i\}=\dots=\{p_1^{(n)},\dots,p_i^{(n)}\}$ in a unique state renamed $(p_0,\{p_1,\dots,p_i\})$. □

Definition 39. Let $Z,L\subseteq\Sigma^*$ and $(A_M,B_M)\in FD(Z,L)$.

The tree $F(Z,L,A_M,B_M)$ is the following labelled tree $F(Z,L,A_M,B_M)=(V,child,lab)$ where vertices are languages on Σ and labels are words on Σ.

The root is Z. The set $child(Z)$ is the set of all languages Z_i such that, when considering $A=PP(A_M)$ in line 5 of FIND-FIN-DEC, then FIND-FIN-DEC calls FIND-FIN-DEC$(Z_i,L,\mathrm{Aut}_{Z_i},\mathrm{Aut}_L,\{Z\cup Z_i\})$ in line 8.

The label $lab(Z,Z_i)=a_i$, where $PP(A_M)=\{a_1,\dots,a_k\}$ and $Z_i=L_{q(a_i),F}\setminus L$.

For any $i=1,\dots,k$, the sub-tree rooted in i is $F(Z_i,L,a_i^{-1}A_M\setminus 1,a_i^{-1}B_M)$ if $MIN(Z_i,L)\neq\emptyset$ or the tree composed of the only vertex Z_i, otherwise.

An example of tree $F(Z,L,A_M,B_M)$ is given in Section 6.1, below.

Remark 40. Suppose that FIN-DEC applies to input $(Z,L,\mathrm{Aut}_Z,\mathrm{Aut}_L)$, it returns $FD(Z,L)\neq\emptyset$ and $(A_M,B_M)\in FD(Z,L)$.

If during the computation of (A_M,B_M), the procedure FIND-FIN-DEC is called on $(X,L,\mathrm{Aut}_X,\mathrm{Aut}_L, \mathrm{T}RACK)$ then TRACK contains all ancestors of X in $F(Z,L,A_M,B_M)$.

Remark 41. Let (A,B) be a finite decomposition of (Z,L).

The trees $F(Z,L,A,B)$ and $T_{A,B}$ are related by a one-to-one correspondence mapping vertices of $F(Z,L,A,B)$ to vertices of $T_{A,B}$. The correspondence maps root Z of $F(Z,L,A,B)$ to root 0 of $T_{A,B}$. Further if the correspondence maps vertex Z_i of $F(Z,L,A,B)$ to vertex i of $T_{A,B}$ then it maps $child(Z_i)$ to $child(i)$. Observe that if the correspondence maps Z_i to vertex i then $R_i=Z_i$.

Proposition 42. *The algorithm* FIN-DEC *always stops.*

Proof. Let us suppose FIN-DEC applies to input $(Z, L, \mathrm{Aut}_Z, \mathrm{Aut}_L)$. Let $C(\mathrm{Aut}_Z, \mathrm{Aut}_L)$ denote the family of all languages X's such that FIND-FIN-DEC is recursively called on $(X, L, \mathrm{Aut}_X, \mathrm{Aut}_L,\ \mathrm{T}RACK)$ during the execution of FIN-DEC. We have that $C(\mathrm{Aut}_Z, \mathrm{Aut}_L)$ is finite by Lemma 38. Indeed Aut_Z and Aut_L are finite automata, each language X is $X = L_{q(a),F} \setminus L$ for some a, and the automaton Aut_a is obtained by intersection of $(Q, q(a), \delta, F)$ and the complement of Aut_L. Consider now the set $\{F(Z, L, A, B) \mid (A, B) \in FD(Z, L)\}$. Note that such a set shows all recursive calls to FIND-FIN-DEC necessary to compute set $FD(Z, L)$. This set is finite since $MIN(X, L)$ is finite for any regular language $X \subseteq \Sigma^*$ and any $X = L_{q(a),F} \setminus L$ is a regular language. Moreover the breadth of any vertex in $F(Z, L, A, B)$ is finite since any $A \in MIN$ is finite. Finally any $F(Z, L, A, B)$ has finite depth because $\mathrm{C}(\mathrm{Aut}_Z, \mathrm{A}_L)$ is finite and because of the condition $L_{q(a),F} \setminus L \notin \mathrm{T}RACK$ in line 5 of FIND-FIN-DEC and the meaning of the set TRACK in Remark 40. □

Let us now prove that algorithm FIN-DEC computes in $FD(Z, L)$ a set of special finite decompositions of (Z, L).

Proposition 43. *Let $Z, L \subseteq \Sigma^*$ be regular languages.*

Then $FD(Z, L) \setminus \{(\emptyset, Z)\} = NORM(Z, L)$, where $NORM(Z, L)$ is the set of all finite decompositions (A, B) of (Z, L) such that $T_{A,B}$ is normal.

Proof. First we show the inclusion $FD(Z, L) \setminus \{(\emptyset, Z)\} \subseteq NORM(Z, L)$. Let $(A_M, B_M) \in FD(Z, L) \setminus \{(\emptyset, Z)\}$. We prove that (A_M, B_M) is a finite decomposition and that T_{A_M, B_M} is normal by induction on the depth d of $F(Z, L, A_M, B_M) = (V, child, lab)$. Note that if $d = 0$ then $(A_M, B_M) = (\emptyset, Z)$.

Suppose $d = 1$. For any leaf Z_i of $F(Z, L, A_M, B_M)$ we have $FD(Z_i, L)$ is either $\{(\emptyset, Z_i)\}$ if Z_i is finite or $\emptyset$. From the construction of A_M (line 11 of FIND-FIN-DEC), we necessarily have that $A_M = PP(A_M)$. Moreover the condition for entering the **for** loop of line 5 implies that $A_M \in MIN(Z, L)$, $Z \setminus A_M \Sigma^*$ is finite and $\forall a \in A_M,\ L_{q(a),F} \setminus L \neq Z$. Therefore A_M satisfies all hypothesis of Theorem 8 (converse part). Then (A_M, B_M) is a finite decomposition since it is constructed (lines 11–12) according to Theorem 8. Further $T_{A,B}$ is R-simple because of the condition $L_{q(a),F} \setminus L \notin \mathrm{T}RACK$ for entering **for** loop in line 5 and because of Remark 40. Moreover it is normal. The only internal node is $i = 0$. For $i = 0$ we have $Lab(0) = A_M \in MIN(Z, L)$ and $R_0 \setminus Lab(0)\Sigma^* = Z \setminus A_M \Sigma^*$ is finite.

Suppose now $d > 1$. The construction of A_M (line 11) implies that FIND-FIN-DEC applied to $(Z, L, \mathrm{Aut}_Z, \mathrm{Aut}_L)$ executes the **for** loop in line 5 with $A = PP(A_M)$. Therefore $PP(A_M) \in MIN(Z, L)$, $Z \setminus PP(A_M)\Sigma^*$ is finite and $\forall a \in PP(A_M)$, $L_{q(a),F} \setminus L \neq Z$. Moreover the procedure FIND-FIN-DEC calls itself on $(L_{q(a),F}, L, \mathrm{Aut}_a, \mathrm{Aut}_L)$ for any $a \in PP(A_M)$. For any $a \in L_{q(a),F} \setminus L$ and $(A, B) \in FD(L_{q(a),F} \setminus L, L)$, the depth of $F(L_{q(a),F} \setminus L, L, A, B)$ is less than d. By the inductive hypothesis $FD(L_{q(a),F} \setminus L, L) \setminus \{(\emptyset, L_{q(a),F} \setminus L)\} = NORM(L_{q(a),F} \setminus L, L)$. Therefore (A_M, B_M) is a finite decomposition since it is constructed (line 11) according to Theorem 8. Further T_{A_M, B_M} is normal. Indeed it is

R-simple because of the condition $L_{q(a),F} \setminus L \notin \mathrm{T}RACK$ each time necessary for entering the **for** loop of line 5 and because of Remark 40. Moreover let i be any internal vertex of T_{A_M,B_M}. If $i=0$ then $Lab(0)=PP(A_M)$ and $R_0 \setminus Lab(0)\Sigma^* = Z \setminus PP(A_M)\Sigma^*$ are both finite languages because of the condition for entering the **for** loop of line 5. If $i \neq 0$ then i is an internal vertex of the sub-tree T_k of T_{A_M,B_M} rooted in some $k \in child(0)$. Further R_i is the language associated to some vertex of T_k because of the definition of languages associated to T_{A_M,B_M}. Therefore $Lab(i)$ and $R_i \setminus Lab(i)\Sigma^*$ are both finite by inductive hypothesis.

Let us now show that $NORM(Z,L) \subseteq FD(Z,L) \setminus \{(\emptyset,Z)\}$. Let (A,B) be a finite decomposition of (Z,L) such that $T_{A,B}=(V,child,lab)$ is normal. We have $(A,B) \neq (\emptyset,Z)$ since the definition of $T_{A,B}$ forbids that $A=\emptyset$. We prove that $(A,B) \in FD(Z,L)$ by induction on the depth d of $T_{A,B}$.

If $d=0$ then $A=1$ and $B=Z \setminus L$. Hence $A \in MIN(Z,L)$, $Z \setminus A\Sigma^* = \emptyset$ is finite and $L_{q(1),F} \setminus L = Z \setminus L = B$ is finite. Therefore A is processed in line 5 of Find-Fin-Dec $(Z,L,\mathrm{Aut}_Z,\mathrm{Aut}_L)$ and $(A,B) \in FD(Z,L)$.

Suppose $d>0$. We have $PP(A)=Lab(0)$, $Z \setminus PP(A) = R_0 \setminus Lab(0)$ and $\forall a \in PP(A)$, $L_{q(a),F} \setminus L = R_{i(a)}$ for some $i(a) \in child(0)$. Therefore when executing Find-Fin-Dec $(Z,L,\mathrm{Aut}_Z,\mathrm{Aut}_L,\{Z\})$ we find $PP(A) \in MIN(Z,L)$, $Z \setminus PP(A)$ is finite and $L_{q(a),F} \setminus L \notin$ Track(=$\{Z\}$). Hence Find-Fin-Dec is called on $(L_{q(a),F} \setminus L, L, \mathrm{Aut}_a, \mathrm{Aut}_L, \{Z, L_{q(a),F} \setminus L\})$ for any $a \in PP(A)$. Consider now for any $a \in PP(A)$ the sub-tree T_a of $T_{A,B}$ rooted in $i(a)$. If T_a is empty then $L_{q(a),F} \setminus L$ is finite. Otherwise T_a is the tree associated to some finite decomposition (A_a,B_a) of $(L_{q(a),F} \setminus L, L)$ (Proposition 28). Further it is normal since it is the sub-tree of a normal tree (Remark 32). By the inductive hypothesis $FD(L_{q(a),F} \setminus L, L) \setminus \{(\emptyset, L_{q(a),F} \setminus L)\} = NORM(L_{q(a),F} \setminus L, L)$. Hence $(A_a,B_a) \in FD(L_{q(a),F} \setminus L, L)$. Observe now that when calling Fin-Dec$(L_{q(a),F} \setminus L, L, \mathrm{Aut}_a, \mathrm{Aut}_L)$ we find $Z_i \notin \mathrm{T}RACK$ for any vertex Z_i of $F(L_{q(a),F} \setminus L, L, A_a, B_a)$. Moreover, this situation holds also if the initial call is Fin-Dec$(Z,L,\mathrm{Aut}_Z,\mathrm{Aut}_L)$, because $T_{A,B}$ is R-simple. Finally $(A,B) \in FD(Z,L)$ since its construction (line 11) follows Theorem 8. $\square$

We are now able to state the main result of this section.

Theorem 44. *It is decidable* (*in a constructive way*) *whether*, *given regular languages* $Z,L \subseteq \Sigma^*$, *Z is finitely L-decomposable.*

Proof. Let $Z,L \subseteq \Sigma^*$ be regular languages. In order to decide whether Z is finitely L-decomposable or not, we choose a deterministic, complete and trim automaton Aut_Z recognizing Z and a deterministic, complete and trim automaton Aut_L recognizing L. Then we use algorithm Fin-Dec and the decidability of inclusion, emptiness and finiteness for regular languages. From Proposition 34, we have that (Z,L) has a finite decomposition not $(\emptyset,Z)$ iff $NORM(Z,L) \neq \emptyset$. Moreover consider the set $FD(Z,L)$ returned by Find-Fin-Dec$(Z,L,\mathrm{Aut}_Z,\mathrm{Aut}_L,\{Z\})$. We have (Proposition 43) that $FD(Z,L) \setminus \{(\emptyset,Z)\} = NORM(Z,L)$. Hence algorithm Fin-Dec applied to $(Z,L,\mathrm{Aut}_Z,\mathrm{Aut}_L)$ returns in $FD(Z,L)$ either a non-empty set of finite decompositions of (Z,L), if (Z,L) has some finite decomposition not $(\emptyset,Z)$, or the message "$\nexists$ finite decompositions", otherwise. In order to decide whether Z is finitely L-decomposable or not, we can thus apply FIN-DEC to $(Z, L, \mathrm{Aut}_Z, \mathrm{Aut}_L)$ and decide that Z is finitely L-decomposable iff $FD(Z,L) \setminus \{(1,\emptyset)),(\emptyset,Z)\} \neq \emptyset$. $\square$

We show now that the finite decompositions provided by FIND-FIN-DEC, if any, are minimal with respect to the length and maximal in the sense specified here below. We denote by $|w|$ the length of word w and by $l(X)$ the *length* of finite language X, that is $l(X) = \sum_{x\in X} |x|$.

Proposition 45. *Let $Z, L \subseteq \Sigma^*$ be regular languages.*

If Z is finitely L-decomposable then for every non-trivial finite decomposition (X,Y) of (Z,L) such that $(X,Y) \notin FD(Z,L)$, then there exists a finite decomposition $(A,B) \in FD(Z,L)$ such that $l(A) < l(X)$ and $l(B) < l(Y)$.

Proof. Let (X,Y) be a non-trivial finite decomposition of (Z,L) such that $(X,Y) \notin FD(Z,L)$. Let $A' = PP(Red(PP(X)))$. As shown in Theorem 22, $A' \in MIN(Z,L)$, $Z \setminus A'\Sigma^*$ is finite and for any $a \in A'$, $L_{q(a),F} \setminus L$ is either finitely L-decomposable or (if not) finite. Therefore, we can apply Theorem 8 and obtain a finite decomposition (A,B) of (Z,L) with $PP(A) = A'$. Indeed $A = A' \cup (\bigcup_{a\in A'} aA_a)$ and $B = Z \setminus A'\Sigma^* \cup (\bigcup_{a\in A'} aB_a)$, where $L_{q(a),F} \setminus L = B_a + A_aL$ and $(A_a,B_a) \in FD(L_{q(a),F} \setminus L, L)$. From Proposition 43, $(A,B) \in FD(Z,L)$.

Moreover, in the proof of Theorem 22, we have shown that $Z \setminus A'\Sigma^* \subseteq Z \setminus PP(X)\Sigma^*$. Hence $l(Z \setminus A'\Sigma^*) \leqslant l(Z \setminus PP(X)\Sigma^*)$. We are now able to show that $l(A) < l(X)$ and $l(B) < l(Y)$. Indeed, for any $x \in PP(X)$ it holds $|Red(x)| \leqslant |x|$ and since $(X,Y) \notin FD(Z,L)$ implies $PP(X) \notin MIN$ then there exists $x \in PP(X)$ such that $|Red(x)| < |x|$. Therefore:

$$l(A) = l(A') + \sum_{a\in A'} |a| l(A_a) < l(PP(X)) + \sum_{x \text{ s.t. } Red(x)=a} |x| l(A_a) = l(X)$$

and

$$\begin{aligned} l(B) &= \sum_{a\in A'} |\, a \,|\, l(B_a) + l(Z \setminus A'\Sigma^*) \\ &< \sum_{x \text{ s.t. } Red(x)=a} |\, x \,|\, l(B_a) + l(Z \setminus PP(X)\Sigma^*) = l(Y). \qquad \square \end{aligned}$$

Remark 46. Let $(A,B),(A',B')$ two finite decompositions of (Z,L). If L is infinite then neither $A' \subset A$ nor $A \subset A'$. Indeed if $A \subset A'$ then there exists $a \in A' \setminus A$ and $aL \subseteq Z$. Hence $Z \setminus A\Sigma^*$ would contain the infinite language aL, against $Z \setminus A\Sigma^* \subseteq B$ is finite.

Proposition 47. *Let $Z, L \subseteq \Sigma^*$ be regular languages.*

If Z is finitely L-decomposable then for every non-trivial finite decomposition (X,Y) of (Z,L), $(X,Y) \notin FD(Z,L)$, such that $A \subseteq X$ for some $(A,B) \in FD(Z,L)$, we have $A = X$.

Proof. Let (X,Y), (A,B) as in the hypothesis. Note that (A,B) is a finite decomposition of (Z,L) (Theorem 43). If L is infinite then $A \subset X$ cannot hold (Remark 46). Hence $A = X$. If L is finite then Z is finite too, since $Z = AL + B$ and A,B,L are finite.

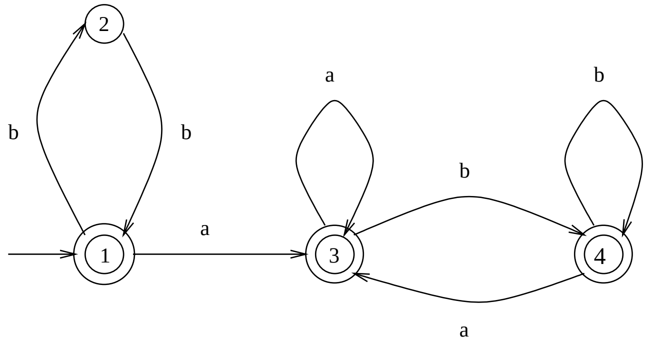

Fig. 2. The automaton Aut_L of Section 6.1.

Therefore no loop exists on a state of Aut_Z. Hence $MIN(Z,L)=\{X \mid X \subseteq A_y, X$ prefix-free$\}$. Moreover for any $(A_M,B_M)\in FD(Z,L)$, any vertex Z_i of $F(Z,L,A_M,B_M)$ is finite. Finally $FD(Z,L)$ is exactly the set of all finite decompositions of (Z,L). In other words no finite decomposition (X,Y) exists such that $(X,Y)\notin FD(Z,L)$. $\square$

6.1. *A first example of execution of* FIN-DEC

Let $Z=L$ be the language recognized by deterministic automaton $\mathrm{Aut}_L=(Q,1,\delta,F)$ shown in Fig. 2. Let us apply algorithm FIN-DEC to input $(L,L,\mathrm{Aut}_L,\mathrm{Aut}_L)$ and follow the computation of a pair (A_M,B_M) in $FD(Z,L)$.

The algorithm calls the procedure FIND-FIN-DEC on $(L,L,\mathrm{Aut}_L,\mathrm{Aut}_L,\{L\})$. Since L is infinite, it looks for $MIN(L,L)$. We have that $Q_y=\{1,3,4\}$. Further $MIN(L,L)\neq\emptyset$ and $A=\{b^2,a\}\in MIN(L,L)$. Language A satisfies the conditions for entering the **for** loop in line 5. Indeed $L\setminus A\Sigma^*=\{1\}$ is finite. Consider $L_{q(b^2),F}\setminus L$ and $L_{q(a),F}\setminus L$. We find $L_{q(b^2),F}\setminus L=L_{1,F}\setminus L=L\setminus L=\emptyset\notin TRACK(=\{L\})$. The language $L_{q(a),F}\setminus L=L_{3,F}\setminus L$ is the language recognized by the automaton $\mathrm{Aut}_3=(Q,3,\delta,F)\otimes(Q,1,\delta,Q\setminus F)$, where $(Q,3,\delta,F)$ and $(Q,1,Q\setminus\delta,F)$ are both canonically completed (Section 1). We find $L_{3,F}\setminus L\notin TRACK(=\{L\})$. Thus the **for** loop in line 6 is executed for any element of A and FIND-FIN-DEC is called on input $I_{b^2}=(\emptyset,L,\mathrm{Aut}_{b^2},\mathrm{Aut}_L,\{L,\emptyset\})$ and $I_a=(L_{3,F}\setminus L,L,\mathrm{Aut}_a,\mathrm{Aut}_L,\{L,L_{3,F}\setminus L\})$.

Procedure FIND-FIN-DEC(I_{b^2}) returns $FD(\emptyset,L)=\{(\emptyset,\emptyset)\}$. Consider now I_a. We find $MIN(L_{3,F}\setminus L,L)\neq\emptyset$ and $A'=\{b\}\in MIN(L_{3,F}\setminus L,L)$. Looking at Aut_3, we find $L_{(4,2),(F,Q\setminus F)}=L$. Hence FIND-FIN-DEC $(L_{3,F}\setminus L,L)$ adds $(b,\emptyset)$ to $FD(L_{3,F}\setminus L)$.

Finally the pair $(A_M,B_M)=(\{b^2,a,ab\},\{1\})$ is constructed (lines 11–12) and added to $FD(L,L)$ (line 13). Indeed $L=1+b^2L_{1,F}+aL_{3,F}=1+b^2L+a(L+bL)$.

Consider tree $F(L,L,A_M,B_M)$. Its root is L; $child(L)=\{\emptyset,L_{3,F}\setminus L\}$; the vertex $\emptyset$ is a leaf and $child(L_{3,F}\setminus L)=\{\emptyset\}$. Moreover $lab(L,\emptyset)=b^2$, $lab(L,L_{3,F}\setminus L)=a$ and $lab(L_{3,F}\setminus L,\emptyset)=b$.

The tree associated to (A_M,B_M) is $T_{A_M,B_M}=(V,child,lab)$, where $V=\{0,1,2,3\}$; the root is 0; $child(0)=\{1,2\}$, $child(2)=\{3\}$ and 1,3 are leaves. Further

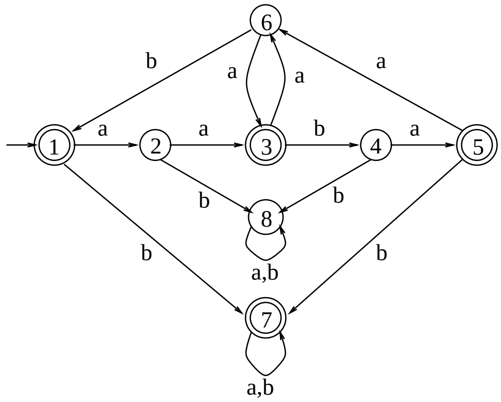

Fig. 3. The automaton Aut_L of Section 6.2.

$lab(0,1) = b^2$, $lab(0,2) = a$ and $lab(2,3) = b$. Note the correspondence between $F(L,L,A_M,B_M)$ and T_{A_M,B_M}, as pointed out in Remark 41.

6.2. *A second example of execution of* FIN-DEC

Let $Z = L$ be the language recognized by deterministic automaton $\mathrm{Aut}_L = (Q,1,\delta,F)$ shown in Fig. 3. Let us apply algorithm FIN-DEC to input $(L,L,\mathrm{Aut}_L,\mathrm{Aut}_L)$ and follow the computation of a pair (A_M,B_M) in $FD(Z,L)$. The procedure FIND-FIN-DEC is called on $(L,L,\mathrm{Aut}_L,\mathrm{Aut}_L,\{L\})$. We have $Q_y = \{1,5,7\}$. For $A = \{1\} \in MIN(L,L)$ we find $L \setminus A\Sigma^* = \emptyset$ is finite and $L_{1,F} \setminus L = \emptyset \notin TRACK(=\{L\})$. Therefore the pair $(1,\emptyset)$ is added to $FD(L,L)$. For any other $A \in MIN(L,L)$, $A \neq \{1\}$, we find that $L \setminus A\Sigma^*$ is infinite since it necessarily contains $(a^2)^+$. Indeed for every $i \geqslant 1$, word $a^{2i} \in L$ and $p(a^{2i})$ never passes through a state of Q_y. Procedure FIN-DEC$(L,L,\mathrm{Aut}_L,\mathrm{Aut}_L)$ thus returns $FD(L,L) = \{(1,\emptyset)\}$. According to Theorem 44, (L,L) has no non-trivial finite decomposition.

7. A generalization to infinite decompositions

In the previous sections, we have looked for finite decompositions of a pair (Z,L) of regular languages. Assume now that (Z,L) has no non-trivial finite decomposition. We sketch here a construction generalizing the one in Section 6. It provides non-trivial (infinite) decompositions of (Z,L) having some property of maximality (Proposition 58). Observe that we have already proved (Proposition 6) that it is decidable whether Z is L-decomposable, for given regular languages Z,L. However the decompositions there provided had not in general the property here mentioned.

We fix some notations:

- $Z,L \subseteq \Sigma^*$ are regular languages
- $\mathrm{Aut}_Z = (Q,1,\delta,F)$ is a deterministic, trim and complete automaton recognizing Z

- $\mathrm{Aut}_L = (Q_L, 1_L, \delta_L, F_L)$ is a deterministic, trim and complete automaton recognizing L
- $Q_y = \{q \in Q \mid L \subseteq L_{q,F}\}$
- A_y is the set of labels of all paths from 1 to a state of Q_y.

Generalizing the definition of $MIN(Z,L)$ in Section 6, let us define the family $MIN^{\infty}(Z,L)$. It turns out that $MIN^{\infty}(Z,L)$ can be used for constructing (infinite) non-trivial decompositions of (Z,L) having some maximality property.

Definition 48. Family $MIN^{\infty}(Z,L)$ is the class of all non-empty prefix-free languages $A = \{a_1, a_2, \ldots, a_n, \ldots\}$ such that:

(1) $A \subseteq A_y$,

(2) for any $i = 1, 2, \ldots n, \ldots,$ path $p(a_i) = (q_1, \sigma_1, q_2) \cdots (q_n, \sigma_n, q_{n+1})$ is such that $q_j = q_k$, with $1 \leqslant j < k \leqslant n+1$ implies either $j = 1$ and $k = n+1$ or $q_j \notin Q_y$.

We will simply write MIN^{∞} when no ambiguity is possible on languages Z, L referred to.

Theorem 49. *Let $Z, L \subseteq \Sigma^*$ be regular languages.*

Language Z is L-decomposable iff $MIN^{\infty}(Z,L) \neq \emptyset$ and $\exists A \in MIN^{\infty}(Z,L)$ such that $A \neq 1$ or $Z \setminus L \neq \emptyset$.

Proof. The proof uses Theorem 7 and mimics the proof of Theorem 22.

Suppose that Z is L-decomposable and (A,B) is a non-trivial decomposition of (Z,L). If $A = 1$ then $B \neq \emptyset$ and A satisfies the requirements of the theorem. Let us suppose that $A \neq 1$. Let $A' = PP(Red_{Q_y}(PP(A)))$. One can easily show that $A' \neq 1, \emptyset$ and $A' \in MIN^{\infty}(Z,L)$. In particular $MIN^{\infty}(Z,L) \neq \emptyset$.

For the converse part, let us suppose $MIN^{\infty}(Z,L) \neq \emptyset$ and $A \in MIN^{\infty}(Z,L)$ is such that either $A \neq 1$ or $Z \setminus L \neq \emptyset$. For any $a \in A$ let $a^{-1}Z \setminus L = B_a + A_a L$, where (A_a, B_a) is eventually trivial. Remark that $1 \notin A_a$ because $L \subset a^{-1}Z \setminus L$ does not hold (Proposition 2). Thus $a^{-1}Z = L + B_a + A_a L$ and Z can be decomposed with respect to A as $Z = A'L + B'$ where: $A' = A \cup (\bigcup_{a \in A} aA_a)$ and $B' = Z \setminus A\Sigma^* \cup (\bigcup_{a \in A} aB_a)$ (for details see the proof of the vice versa in Theorem 7). The decomposition (A', B') is not trivial since either $A \neq 1$ or $A_a \cup B_a \neq \emptyset$, for $a = 1$. $\square$

Lemma 50. *Let $Z, L \subseteq \Sigma^*$ be regular languages. Any $A \in MIN^{\infty}(Z,L)$ maximal in $MIN^{\infty}(Z,L)$ with respect to inclusion can be decomposed as $A = \bigcup_{q \in Q(A)} X_q$ for some $Q(A) \subseteq Q_y$ and $X_q \subseteq \Sigma^*$. Moreover the set of all $A \in MIN^{\infty}(Z,L)$ that are maximal in $MIN^{\infty}(Z,L)$ with respect to inclusion is finite.*

Proof. Languages $A \in MIN^{\infty}(Z,L)$ that are maximal in $MIN^{\infty}(Z,L)$ with respect to inclusion, can be decomposed as $A = \bigcup_{q \in Q(A)} X_q$ for some $Q(A) \subseteq Q_y$ and X_q is the language of labels of *all* paths from 1 to q without loops on states of Q_y. The finiteness of the set of all $A \in MIN^{\infty}(Z,L)$ that are maximal in $MIN^{\infty}(Z,L)$ with respect to inclusion, follows from the finiteness of Q_y. $\square$

Lemma 51. *Let $Z, L \subseteq \Sigma^*$ be regular languages. Let $A \in MIN^\infty(Z,L)$ maximal in $MIN^\infty(Z,L)$ with respect to inclusion, $A = \bigcup_{q\in Q(A)} X_q$ and $L_{q,F} \setminus L = A_q L + B_q$. Then $Z = A_M L + B_M$ where $A_M = A \cup (\bigcup_{q\in Q(A)} X_q A_q)$, $B_M = Z \setminus A\Sigma^* \cup (\bigcup_{q\in Q(A)} X_q B_q)$.*

Proof. From Theorem 7, we have that $Z = A'L + B'$ where $A' = A \cup (\bigcup_{a\in A} aA_a)$, $B' = Z \setminus A\Sigma^* \cup (\bigcup_{a\in A} aB_a)$ and $L_{q(a),F} \setminus L = A_a L + B_a$. Remark that $L_{q(a),F} \setminus L = A_{q(a)} L + B_{q(a)}$ and $X_q = \{a \mid q(a) = q\}$. Hence $A' = A \cup (\bigcup_{a\in A} aA_a) = A \cup (\bigcup_{a\in A} aA_{q(a)}) = A \cup (\bigcup_{q\in Q(A)} X_q A_q)$ and $B' = Z \setminus \Sigma^* \cup (\bigcup_{a\in A} aB_a) = Z \setminus \Sigma^* \cup (\bigcup_{a\in A} aB_{q(a)}xs) = Z \setminus \Sigma^* \cup (\bigcup_{q\in Q(A)} X_q B_q)$. □

We sketch now an algorithm that applied to input $(Z, L, \mathrm{Aut}_Z, \mathrm{Aut}_L)$, where Z is not finitely L-decomposable, tests whether Z is L-decomposable and eventually returns some non-trivial decompositions of (Z,L). Such decompositions are maximal in the sense specified in Proposition 58.

Note that in the following procedure FIND-MAX-DEC, for any set A considered in line 3, $Q(A)$ and X_q denote the sets such that $A = \bigcup_{q\in Q(A)} X_q$, as in Lemma 50. We emphasize that the algorithm is written in an informal way.

MAX-DEC$(Z, L, \mathrm{Aut}_Z, \mathrm{Aut}_L)$

FIND-MAX-DEC$(Z, L, \mathrm{Aut}_Z, \mathrm{Aut}_L, \{Z\})$
if $MD(Z,L) \setminus \{(\emptyset, Z), (1, \emptyset)\} \neq \emptyset$
  then return "Z is L-decomposable"
  else return "Z is not L-decomposable"

FIND-MAX-DEC$(X, L, \mathrm{Aut}_X, \mathrm{Aut}_L, \mathrm{T}RACK)$

1 **if** $MIN^\infty(X,L) = \emptyset$
2   **then** $MD(X,L) \leftarrow \{(\emptyset, X)\}$
3   **else for** any $A \in MIN^\infty(X,L)$ maximal w.r.t. inclusion
4   **do for** any $q \in Q(A)$ s.t. $L_{q,F} \setminus L \in TRACK$
      and $L_{q,F} \setminus L = W(L_{q,F} \setminus L) + A'L + B'$
5   **do** $MD(L_{q,F} \setminus L, L) \leftarrow MD(L_{q,F} \setminus L, L) \cup \{(W^*A', W^*B')\}$
6   **for** any $q \in Q(A)$ s.t. $L_{q,F} \setminus L \notin TRACK$
7   **do** $I \leftarrow (L_{q,F} \setminus L, L, \mathrm{Aut}_a, \mathrm{Aut}_L, TRACK \cup \{L_{q,F} \setminus L\})$
8   FIND-MAX-DEC(I)
9   **for** any $q \in Q(A)$ and $(A_q, B_q) \in MD(L_{q,F} \setminus L, L)$
10   **do** $A_M \leftarrow A \cup (\bigcup_{q\in Q(A)} X_q A_q)$
11         $B_M \leftarrow X \setminus A\Sigma^* \cup (\bigcup_{q\in Q(A)} X_q B_q)$
12         $MD(X,L) \leftarrow MD(X,L) \cup \{(A_M, B_M)\}$

Note that in the sequel $MD(Z,L)$ denotes the set returned by FIND-MAX-DEC$(Z, L, \mathrm{Aut}_Z, \mathrm{Aut}_L, \{Z\})$.

Remark 52. Set MD(Z,L) is always non-empty. If $MIN^\infty(Z,L) = \emptyset$ then $MD(Z,L)$ contains the trivial and infinite decomposition $(\emptyset, Z)$, else it contains all pairs (A_M, B_M) constructed as in lines 10–11.

During the execution of algorithm Max-Dec with a given input, a language A is considered in line 3, even if it is infinite. In the case when A in line 3 is infinite, we say that the resulting decomposition (A_M, B_M) is *breadth-infinite*. Further in the case when the **for** loop of line 4 is executed, we say that the resulting decomposition (A_M, B_M) is *depth-infinite*. The terms breadth-infinite and depth-infinite are in relation to tree T_{A_M,B_M} associated to decomposition (A_M, B_M) (see Section 5).

Suppose now that Max-Dec applies to input $(Z, L, \mathrm{Aut}_Z, \mathrm{Aut}_L)$. Let us recursively define for any $(A_M, B_M) \in MD(Z, L)$, a labelled tree. Its structure exactly mirrors the structure of recursive calls of the procedure Max-Dec when it computes (A_M, B_M) and adds it to $MD(Z, L)$. This definition will be useful in proving that algorithm Max-Dec always stops.

Definition 53. Let $Z, L \subseteq \Sigma^*$ and $(A_M, B_M) \in MD(Z, L)$.

The tree $M(Z, L, A_M, B_M)$ is the following labelled tree $M(Z, L, A_M, B_M) = (V, child, lab)$ whose vertices and labels are languages on Σ.

The root is Z. The set $child(Z)$ is the set of all languages Z_i such that, when considering $A = PP(A_M)$ in line 3 of Find-Max-Dec, then Find-Max-Dec calls Find-Max-Dec $(Z_i, L, \mathrm{Aut}_{Z_i}, \mathrm{Aut}_L, \{Z \cup Z_i\})$ in line 8, where $A = \bigcup_{q \in Q(A)} X_q$, $Q(A) = \{q_1, \ldots, q_k\}$, and $Z_i = L_{q_i,F} \setminus L$, for any $i = 1, \ldots, k$.

The label $lab(Z, Z_i) = X_{q_i}$.

For any $i=1, \ldots, k$, the sub-tree rooted in i, is $M(Z_i, L, \bigcup_{a \in A} a^{-1}A_M \setminus 1, \bigcup_{a \in A} a^{-1}B_M)$ if $MIN^\infty(Z_i, L) \neq \emptyset$ or the tree composed of the only vertex Z_i, otherwise.

Some examples of trees $M(Z, L, A, B)$ are given in Sections 7.1 and 7.2.

Remark 54. Let $(A, B) \in MD(Z, L)$. As pointed out in the proof of Proposition 56, tree $M(Z, L, A, B)$ is always a finite tree: it has finite breadth and finite depth. On the other hand, if (A, B) is not a finite decomposition then $T_{A,B}$ is an infinite tree: it has either infinite breadth or infinite depth, following that (A, B) is either a breadth-infinite or a depth-infinite decomposition, respectively.

Remark 55. Suppose that Max-Dec applies to input $(Z, L, \mathrm{Aut}_Z, \mathrm{Aut}_L)$ and $(A_M, B_M) \in MD(Z, L)$. If during the computation of (A_M, B_M), the procedure Find-Max-Dec is called on $(X, L, \mathrm{Aut}_X, \mathrm{Aut}_L, TRACK)$ then Track contains all ancestors of X in $M(Z, L, A_M, B_M)$.

Proposition 56. *The algorithm* Max-Dec *always stops.*

Proof. The proof mimics the one of Proposition 42. Let us suppose that Max-Dec applies to input $(Z, L, \mathrm{Aut}_Z, \mathrm{Aut}_L)$. Let $C_{\max}(\mathrm{Aut}_Z, \mathrm{Aut}_L)$ denote the family of all languages X's such that Find-Max-Dec is recursively called on $(X, L, \mathrm{Aut}_X, \mathrm{Aut}_L, TRACK)$ during the execution of Max-Dec. We have that $C_{\max}(\mathrm{Aut}_Z, \mathrm{Aut}_L)$ is finite by Lemma 38. Indeed Aut_Z and Aut_L are finite automata, each set X is $X = L_{q,F} \setminus L$ for some $q \in Q$, and the automaton Aut_a is obtained by intersection of (Q, q, δ, F) and the complement of Aut_L. Consider now the set $\{M(Z, L, A, B) \mid (A, B) \in FD(Z, L)\}$. Note that such a set shows all recursive calls to Find-Max-Dec necessary to compute $MD(Z, L)$. This set is finite since the set of all $A \in MIN^{\infty}(X,L)$ maximal with respect to inclusion is finite for any regular language X (Lemma 50) and $X = L_{q,F} \setminus L$ is a regular language. Moreover the breadth of any vertex in $M(Z,L,A,B)$ is finite since $Q(A)$ is finite. Finally any $M(Z,L,A,B)$ has finite depth since $C_{\max}(\mathrm{Aut}_Z, \mathrm{Aut}_L)$ is finite and FIND-MAX-DEC calls itself only if $L_{q(a),F} \setminus L \notin TRACK$ (see line 8 and Remark 55). □

Let us now prove that algorithm MAX-DEC computes in $MD(Z,L)$ a set of special decompositions of (Z,L). Observe that Proposition 57 can be used to prove that it is decidable whether Z is L-decomposable, for given regular languages Z,L (by a proof similar to that of Theorem 44). This decidability result was already proved in Proposition 6. However the decompositions provided in Proposition 6 have not in general the maximality property of the decompositions constructed by algorithm MAX-DEC (Proposition 58).

Proposition 57. *Let $Z,L \subseteq \Sigma^*$ be regular languages.*
The language Z is L-decomposable iff $MD(Z,L) \setminus \{(1,\emptyset),(\emptyset,Z)\} \neq \emptyset$. Moreover any $(A,B) \in MD(Z,L)$ is a decomposition of (Z,L).

Proof. We prove that
(1) any $(A_M,B_M) \in MD(Z,L)$ is a decomposition of (Z,L)
(2) if (Z,L) has a non-trivial decomposition then $MD(Z,L) \setminus \{(1,\emptyset),(\emptyset,Z)\} \neq \emptyset$.

(1) Let $(A_M,B_M) \in MD(Z,L)$. We prove that (A_M,B_M) is a decomposition of (Z,L) by induction on the depth d of $M(Z,L,A_M,B_M)$.

If $d = 0$ then either $(A_M,B_M) = (\emptyset,Z)$ (and it is a decomposition of (Z,L)) or (A_M,B_M) is constructed in lines 10–11 as $A_M = A \cup (\bigcup_{q \in Q(A)} X_q A_q), B_M = Z \setminus A\Sigma^* \cup (\bigcup_{q \in Q(A)} X_q B_q)$, where $A = \bigcup_{q \in Q(A)} X_q$, $L_{q,F} \setminus L = W(L_{q,F} \setminus L) + A'L + B'$ and $A_q = W^*A'$, $B_q = W^*B'$. Using formal power series theory, we have that $L_{q,F} \setminus L = W(L_{q,F} \setminus L) + A'L + B'$ iff $(1-W)(L_{q,F} \setminus L) = A'L + B'$ iff $L_{q,F} \setminus L = W^*(A'L + B') = W^*A'L + W^*B'$. Therefore (A_M,B_M) is a decomposition of (Z,L) applying Theorem 7 and Lemma 51.

If $d \geqslant 1$ then (A_M,B_M) is constructed in lines 10–11 as $A_M = A \cup (\bigcup_{q \in Q(A)} X_q A_q)$, $B_M = Z \setminus A\Sigma^* \cup (\bigcup_{q \in Q(A)} X_q B_q)$, where $(A_q,B_q) \in MD(L_{q,F} \setminus L, L)$. By inductive hypothesis (A_q,B_q) is a decomposition of $(L_{q,F} \setminus L, L)$ since $M(L_{q,F} \setminus L, L, A_q, B_q)$ has depth less than d. The goal thus follows from Theorem 7 and Lemma 51.

(2) If (Z,L) has a non-trivial decomposition then (Theorem 49) there exists $X \in MIN^{\infty}(Z,L)$, such that $X \neq 1$ or $Z \setminus L \neq \emptyset$. If there exists $A_{\max} \in MIN^{\infty}(Z,L)$ maximal with respect to inclusion and $A_{\max} \neq 1$ then the pair (A_M,B_M) constructed in lines 10–11 starting from $A = A_{\max}$ is added to $MD(Z,L)$ (line 12). Moreover $(A_M,B_M) \in MD(Z,L) \setminus \{(1,\emptyset),(\emptyset,Z)\}$ since $A_M \neq 1, \emptyset$. If the only $A_{\max} \in MIN^{\infty}(Z,L)$ maximal with respect to inclusion is $A_{\max} = 1$ then $Z \setminus L \neq \emptyset$. Hence the pair (A_M,B_M) constructed in lines 10–11 starting from $A = 1$ is $(A_M,B_M) = (1, Z \setminus L)$. Therefore $(1, Z \setminus L) \in MD(Z,L)$. Moreover $(1, Z \setminus L) \in MD(Z,L) \setminus \{(1,\emptyset),(\emptyset,Z)\}$ since $Z \setminus L \neq \emptyset$. □

In dealing with infinite decompositions we cannot talk about minimality in length, as it was the case for finite decompositions (Proposition 45). What characterizes decompositions returned by algorithm MAX-DEC is the maximality property, hereafter stated.

Proposition 58. *Let $Z, L \subseteq \Sigma^*$ be regular languages. If Z is L-decomposable then for every non-trivial decomposition (X, Y) of (Z, L), $(X, Y) \notin MD(Z, L)$ such that $A \subseteq X$, for some $(A, B) \in MD(Z, L)$ we have $A = X$.*

Proof. Let Z, L be infinite languages and (X, Y), (A, B) as in the hypothesis. Let $T_{A,B} = (V_A, child_A, lab_A)$ be the tree associated to (A, B) and $T_{X,Y} = (V_X, child_X, lab_X)$ be the tree associated to (X, Y). For any vertex $i \in V_A$ (V_X, resp.), let $R_i(A, B)$ ($R_i(X, Y)$, resp.) be the language associated to i in $T_{A,B}$ ($T_{X,Y}$, resp.). As pointed out in Proposition 28, for any vertex i in $T_{A,B}$ ($T_{X,Y}$, resp.) language $R_i(A, B)$ ($R_i(X, Y)$, resp.) is finitely L-decomposable and the sub-tree rooted in i is the tree associated to some finite decomposition (A_i, B_i) ((X_i, Y_i), resp.) of $(R_i(A, B), L)$ ($R_i(X, Y)$, resp.). Define a numbering $n : V \to \mathbb{N} \cup \{0\}$ on $(V, child, lab)$, where $(V, child, lab)$ is $(V_A, child_A, lab_A)$ or $(V_X, child_X, lab_X)$, in such a way that for any $i, j \in V$:

(1) $n(i) = 0$ if i is the root,
(2) $n(j) > n(i)$ for any $j \in child(i)$,
(3) $i < j$ implies $n(h) < n(k)$ for any $h \in child(i)$, $k \in child(j)$,
(4) if $lab(i, j)$ is less than $lab(i, k)$ in the lexicographical order then $n(j) < n(k)$,
(5) n is surjective on $\{0, 1, \ldots, card(V)\}$.

Suppose now $A \subset X$ and let i be the first vertex in V_A (following numbering n) such that $PP(A_i) \subset PP(X_i)$. Observe that $R_i(A, B) = R_i(X, Y)$ because $R_0(A, B) = R_0(X, Y) = Z$ and the labels of the paths from the root to i in $T_{A,B}$ and $T_{X,Y}$ are equal. Let $R_i = R_i(A, B)$.

Firstly, consider the case when $A_i = PP(A_i) = \emptyset$. The set $MIN^\infty(R_i(A, B), L) = \emptyset$ (this is the only case Find-Fin-Dec returns $A_i = \emptyset$). By Remark 24 any other finite decomposition of $(R_i(A, B), L) = (R_i(X, Y), L)$ is trivial. If i is the root then (R_i, L) has no non-trivial finite decomposition. If i is not the root then any trivial finite decomposition of $(R_i(X, Y), L)$ is $(\emptyset, R_i(X, Y))$. Pair $(1, \emptyset)$ cannot be a finite decomposition of $(R_i(X, Y), L)$ since $R_i \neq L$. Therefore $A_i = X_i$ and this contradicts $PP(A_i) \subset PP(X_i)$.

Consider now the case when $PP(A_i) \subset PP(X_i)$ and $PP(A_i) \neq \emptyset$. Note that $MIN^\infty(R_i, L) \neq \emptyset$ and $PP(A_i) \in MIN^\infty(R_i, L)$, by construction. Let $x \in PP(X_i) \setminus PP(A_i)$. We have $\{x\}$ and $PP(A_i)$ are prefix-free. This implies $PP(A_i) \cup \{x\} \in MIN^\infty$, against the maximality with respect to inclusion of $PP(A_i)$, as required in line 3 of Find-Max-Dec. □

Remark 59. Proposition 58 in particular shows that if (A, B), $(A', B') \in MD(Z, L)$ then neither $A' \subset A$ nor $A \subset A'$.

Proposition 60. *Let $Z, L \subseteq \Sigma^*$ be regular languages. If (A_M, B_M) is any decomposition in $MD(Z, L)$ then A_M, B_M are regular languages.*

Proof. The proof uses induction on the number of calls of procedure Find-Max-Dec. The basic cases are when the procedure adds to $MD(Z, L)$ the pair $(A_M, B_M) = (\emptyset, Z)$ (line 2) or the pair (W^*A', W^*B') (line 5). In the first case A_M, B_M are trivially regular. In the second case A_M, B_M are regular since W, A', B' can be obtained from finite languages using a finite number of union, product and star (see the proof of Lemma 36).

Consider the case when (A_M, B_M) is constructed after entering the **for** loop in line 6. As observed in Lemma 50, languages $A \in MIN^{\infty}$ maximal with respect to inclusion, can be decomposed as a finite union $A = \bigcup_{q\in Q(A)} X_q$, where X_q is the language of labels of all paths from 1 to q without loops on states of Q_y. Note that any X_q is a regular language. Languages A_M, B_M are constructed as $A_M = \bigcup_{q\in Q(A)} X_q \cup (\bigcup_{q\in Q(A)} X_q A_q)$ and $B_M = Z \setminus A\Sigma^* \cup (\bigcup_{q\in Q(A)} X_q B_q)$, where (A_q, B_q) is a decomposition of $(L_{q,F} \setminus L, L)$. Therefore A_M, B_M are obtained by a finite number of unions of some X_q's, A_q's and B_q's, where A_q's and B_q's are regular by inductive hypothesis. Hence A_M, B_M are regular languages. $\square$

7.1. An example of a breadth-infinite decomposition

Let $Z = L$ be the language recognized by the automaton in Fig. 3, as considered in Section 6.2. In Section 6.2 we noticed that (Z, L) has no non-trivial finite decomposition. We want now to construct a non-trivial infinite decomposition of (Z, L), using algorithm Max-Dec. Let us apply Find-Max-Dec to $(Z, L, \mathrm{Aut}_Z, \mathrm{Aut}_L, \{Z\})$. Looking for $MIN^{\infty}(Z, L)$, we find $Q_y = \{1, 5, 7\}$. Furthermore any A in $MIN^{\infty}(Z, L)$, $A \neq 1$, is infinite. Therefore, when considering any A, $A \neq 1$ (line 3), we will construct a breadth-infinite decomposition. Consider for example $A = \{(a^2)^+ab, (a^2)^+ba, b\}$. We have $A \in MIN^{\infty}(Z, L)$ and A maximal with respect to inclusion. Since $Q(A) = \{1, 5, 7\}$, Find-Max-Dec is called on $(L_{i,F} \setminus L, L, \mathrm{Aut}_i, \mathrm{Aut}_L, \{Z, L_{i,F} \setminus L\})$, where $i = 1, 5, 7$ and $\mathrm{Aut}_i = (Q, i, \delta, F) \otimes (Q, 1, \delta, Q \setminus F)$.

We find that $L_{1,F} \setminus L = L \setminus L = \emptyset$ and thus Find-Max-Dec returns $MD(L_{1,F} \setminus L, L) = \{(\emptyset, \emptyset)\}$. Further $L_{5,F} \setminus L = abL$. Consider now state 7. We have that $L_{7,F} \setminus L = \Sigma^* \setminus L$ and $\mathrm{Aut}_7 = (Q, 1, \delta, Q \setminus F)$. Further $A' = \{a\} \in MIN^{\infty}(L_{7,F} \setminus L, L)$. It is maximal with respect to inclusion, since $q(a) = 2$ and $L \subseteq L_{2,Q\setminus F}$. Recursively calling the procedure with $X = L_{2,Q\setminus F} \setminus L$, we find $L_{2,Q\setminus F} \setminus L = abL_{(4,s),(Q\setminus F, Q\setminus F)}$. After a last step we find $L_{(4,s),(Q\setminus F, Q\setminus F)} \setminus L = \emptyset$.

Therefore $(\{ab\}, \emptyset)$ is a decomposition in $MD(L_{2,Q\setminus F} \setminus L, L)$ and $(\{a, a^2b\}, \emptyset)$ is a decomposition in $MD(L_{7,F} \setminus L, L)$. Remarking that $Z \setminus A\Sigma^* = (a^2)^*$, we finally have that pair (A_M, B_M), where A_M, B_M are as follows, is constructed (lines 10–11) and added to $MD(Z, L)$ (line 12):

$A_M = \{(a^2)^+ab, (a^2)^+ba, (a^2)^+baab, b, ba, ba^2b\}$,

$B_M = (a^2)^*$.

Indeed

$$
\begin{aligned}
Z &= (a^2)^* + (a^2)^+abL_{1,F} + (a^2)^+baL_{5,F} + bL_{7,F} \\
&= (a^2)^* + (a^2)^+abL + (a^2)^+ba(L + abL) + b(L + aL + a^2bL).
\end{aligned}
$$

Tree $M(Z, L, A_M, B_M)$ is given in Fig. 4. On the other hand, the tree associated to decomposition (A_M, B_M) is infinite since the root Z has a child for any word in A. This is the reason why we call the decomposition (A_M, B_M) breadth-infinite.

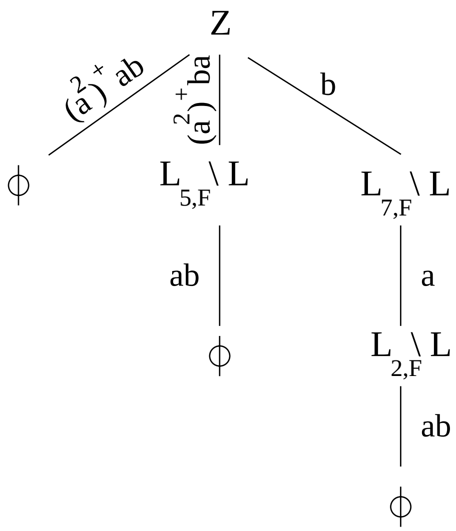

Fig. 4. The tree $M(Z, L, A_M, B_M)$ of Section 7.1.

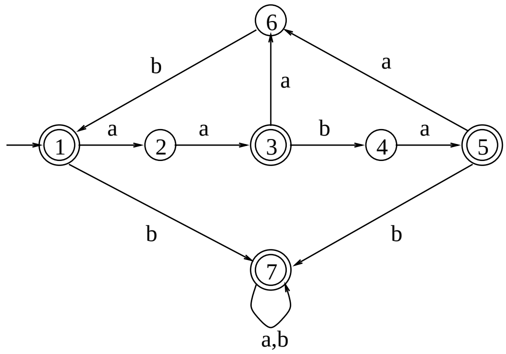

Fig. 5. The automaton Aut_Z of Section 7.2.

7.2. An example of a depth-infinite decomposition

Let $Z = L$ be the language recognized by the deterministic automaton $\mathrm{Aut}_Z = (Q, 1, \delta, F)$ shown in Fig. 5. The automaton is completed by adding a sink state s (not shown in the figure). Let us apply MAX-DEC to $(Z, L, \mathrm{Aut}_Z, \mathrm{Aut}_L)$. Looking for $MIN^\infty(Z, L)$, we find $Q_y = \{1, 7\}$. Further we have that $A = \{a^3b, a^2ba^2b, a^2bab, b\} \in MIN^\infty(Z, L)$ and A is maximal with respect to inclusion. The procedure FIND-MAX-DEC calls itself on input $I_q = (L_{q,F} \setminus L, L, \mathrm{Aut}_q, \mathrm{Aut}_L, \{Z, L_{q,F} \setminus L\})$ for $q \in \{1, 7\}$ and $\mathrm{Aut}_q = (Q, q, \delta, F) \otimes (Q, 1, \delta, Q \setminus F)$.

Note that $L_{1,F} \setminus L = L \setminus L = \emptyset$, which is finite. Thus $MD(L_{1,F} \setminus L, L) = \{(\emptyset,\emptyset)\}$. Consider now state 7. We have that $L_{7,F} \setminus L = \Sigma^* \setminus L$ and $\mathrm{Aut}_7 = (Q, 1, \delta, Q \setminus F)$. Further $A' = \{a\} \in MIN^\infty(\Sigma^* \setminus L, L)$ and it is maximal with respect to inclusion. We have $q(a) = 2$ and it can be shown that $L \subseteq L_{2,Q\setminus F}$. Recursively calling the procedure with $X = L_{2,Q\setminus F} \setminus L$, we find that $A'' = \{ab, a^3, a^2ba^3\} \in MIN^\infty(L_{2,Q\setminus F} \setminus L, L)$ and is maximal with respect to inclusion. Considering the automaton $\mathrm{Aut}_2 = (Q, 2, \delta, Q\setminus F) \otimes (Q, 1, \delta, Q \setminus F)$, we have $L_{2,Q\setminus F} \setminus L = abL_{(4,s),(Q\setminus F, Q\setminus F)} + (a^3 + a^2ba^3)L_{(s,6),(Q\setminus F,Q\setminus F)}$. It can also be shown that $L_{4,F} = \Sigma^* \setminus L$ and then $L_{(4,s),(Q\setminus F,Q\setminus F)} = \Sigma^* \setminus L_{4,F} = L$. Let us denote now $L_6 = L_{(s,6),(Q\setminus F,Q\setminus F)}$. Constructing, in the usual way, an automaton recognizing $L_6 \setminus L$, we find that $L_6 \setminus L = aL_{2,Q\setminus F}$. Further $\{a\} \in MIN^\infty(L_6 \setminus L, L)$ and it is maximal with respect to inclusion, since $L \subseteq L_{2,Q\setminus F}$. Then the procedure FIND-MAX-DEC is called on $(L_{2,Q\setminus F} \setminus L, L, \mathrm{Aut}_2, \mathrm{Aut}_L, \mathrm{T}RACK)$ with $\mathrm{T}RACK = \{Z, L_{7,F} \setminus L, L_{2,Q\setminus F} \setminus L, L_6 \setminus L\}$. Since $L_{2,Q\setminus F} \setminus L \in \mathrm{T}RACK$, we are in a depth-infinite case. Indeed denoting $X = L_{2,Q\setminus F} \setminus L$, we have $X = abL + (a^3 + a^2ba^3)L + (a^3 + a^2ba^3)aL + (a^3 + a^2ba^3)aX$ and then $X = [(a^3 + a^2ba^3)a]^*[ab + (a^3 + a^2ba^3) + (a^3 + a^2ba^3)a]L$. Moreover $Z = (a^3b + a^2ba^2b)L + (a^2bab + b)L + (a^2bab + b)aX$.

Finally (Z, L) has no finite decomposition. An infinite decomposition of (Z, L) is (A_M, B_M) where:

$A_M = \{a^3b, a^2ba^2b, a^2bab, b, \{a^2bab, b\}a\{a^3a, a^2ba^3a\}^*\{ab, a^3, a^2ba^3, a^4, a^2ba^4\}\}$,

$B_M = \{1\}$.

Tree $M(Z, L, A_M, B_M)$ is given in Fig. 6. Tree T_{A_M,B_M} is given in Fig. 7, where T_5 is the sub-tree rooted in vertex 5. The tree T_{A_M,B_M} has finite breadth, but infinite depth.

8. How many decompositions?

In this section we consider the problem of how many non-trivial decompositions (finite decompositions, resp.) a pair (Z, L) of languages can have. We show that if $Z = L$ and L is L-decomposable (finitely L-decomposable, resp.) then it has an infinite number of non-trivial decompositions (finite decompositions, resp.). This is not the case when $Z \neq L$. We introduce an operation called *substitution*. It allows to construct an infinite family of non-trivial decompositions of (L, L), starting from some known ones.

Definition 61. Given languages $A, B, A', B' \subseteq \Sigma^*$, with $A \neq \emptyset$, a *substitution* of (A', B') in (A, B) with respect to $a \in A$ is the pair $(A - a + aA', B + aB')$.

Proposition 62. *Let $L \subseteq \Sigma^*$ and (A', B'), (A, B) be two decompositions of (L, L). If $A, A' \neq \emptyset$ and $(A', B'), (A, B) \neq (1, \emptyset)$ then for all $a \in A$ the substitution (A'', B'') of (A', B') in (A, B) with respect to a is a non-trivial decomposition of (L, L). Moreover if A, B, A', B' are finite languages then A'', B'' are finite and $l(A \cup B), l(A' \cup B') < l(A'' \cup B'')$.*

Proof. $L = AL + B = (A - a)L + aL + B = (A - a)L + a(B' + A'L) + B = (A - a + aA')L + (B + aB')$. Notice that all equalities are unambiguous. Remark that since L denotes a characteristic series then also $A - a + aA'$ and $B + aB'$ are characteristic series. In particular $A \setminus \{a\} \cap aA' = \emptyset$ and $B \cap aB' = \emptyset$. $\square$

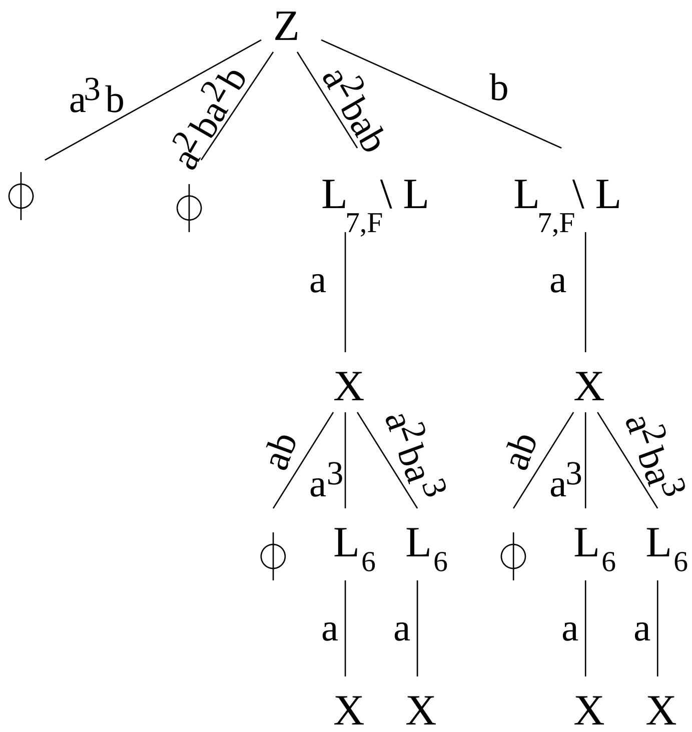

Fig. 6. The tree $M(Z, L, A_M, B_M)$ of Section 7.2.

Corollary 63. *Let $L \subseteq \Sigma^*$. If L is L-decomposable (finitely L-decomposable, resp.), then (L, L) has an infinite number of non-trivial decompositions (finite decompositions, resp.).*

Example 64. Let (A, B) be a non-trivial decomposition of (L, L) with $A = \{a_1, a_2, \ldots, a_n\}$. Consider the substitution of (A, B) in (A, B) with respect to a_1, say (A_1, B_1), then the substitution of (A, B) in (A_1, B_1) with respect to a_2 and so on. We obtain that $(A^2, B + AB)$ is a decomposition of (L, L). Indeed, $L = AL + B = A(B + AL) + B = A^2L + B + AB$. Another example is Example 5.

Consider now the case when $Z \neq L$. We show that (Z, L) can have only a finite number of non-trivial decompositions by the following example. Remark that this is not always the case. Let $Z = a^* + ba^*$, $L = A^*$, as in Example 5. We have that for any $n \geqslant 1$, pair $(a^n + b, 1 + a + \cdots + a^{n-1})$ is a finite decomposition of (Z, L).

Example 65. Let $Z = a^2b^*$, $L = ab^*$. The equality $Z = a^2b^* = Aab^* + B$ implies $B = aab^* - Aab^* = (a - A)ab^* = (a - A)a(1 - b)^{-1}$, that implies $B(1 - b) = (a - A)a = a^2 - Aa$. If $B = \emptyset$ then $(a, \emptyset)$ is a non-trivial decomposition. If $B \neq \emptyset$ let w be the shortest word in B. Then $(B - Bb, w) = 1$. Since $(-Aa, w) \leqslant 0$ then $(a^2, w) = 1$ and $(-Aa, w) = 0$.

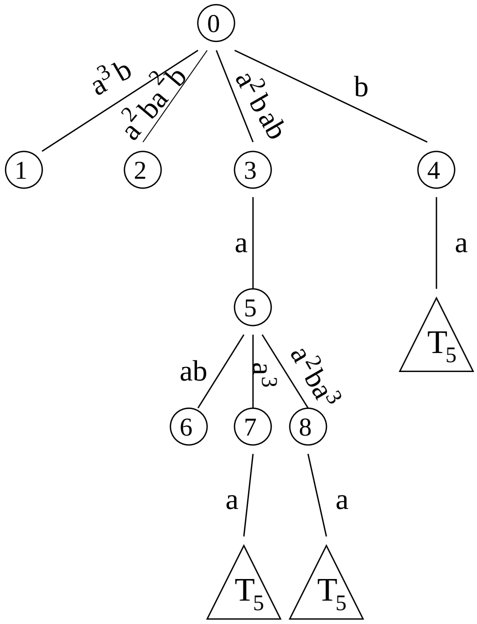

Fig. 7. The tree T_{A_M,B_M} of Section 7.2.

This means $w=a^2$ and $(A,a)=0$, i.e. $a\notin A$. On the other hand, for every $i\geqslant 1$ we have $a^2b^i\in B$ since $a^2b^i\in Z$ and $a^2b^i\notin Aab^*$. Therefore $B=a^2b^*=Z$, $A=\emptyset$. Finally the only decompositions of (Z,L) are $(a,\emptyset)$ and $(\emptyset,Z)$.

9. An application to the Factorization Conjecture

We present here some results in the theory of codes, as developed by Schützenberger and his school. Many deep results about codes have been proved (see [8] for a complete survey on this topic and [13] for a list of open problems in this area). Nevertheless, the structure of these objects is not yet completely investigated. In particular, more than thirty years ago Schützenberger gave the following Factorization Conjecture which is still open (see [37,11,13,18,19,31,34,20,39] for some partial results). We show here some results related to this conjecture. They are based on results shown in previous sections.

We say that $C\subseteq\Sigma^*$ is a *code* (over Σ) if for any $c_1,\dots,c_h,c'_1,\dots,c'_k\in C$, $c_1\cdots c_h=c'_1\cdots c'_k$ implies $h=k$ and for every $i\in\{1,\dots,h\}$, $c_i=c'_i$. In terms of series C is a code iff $\underline{C^*}=(\underline{C})^*$. A code C is *maximal* (over Σ) if for any code C' over Σ then $C\subseteq C'$ implies that $C=C'$. A code C is *factorizing* (over Σ) if there exist finite subsets S, P of Σ^* such that $\underline{S}\ \underline{C}^*\underline{P}=\underline{\Sigma}^*$. Pair (S,P) is called a *factorization* of C. As an example, maximal prefix codes C are factorizing, by taking $S=1$ and $P=Pref(C)$.

Factorization Conjecture (Schützenberger [37]).

Any finite maximal code is factorizing.

In this paper we consider the following problem related to Factorization Conjecture.

Problem SCP: Given finite language S, do there exist finite languages C, P, with C maximal code, such that $SC^*P = \Sigma^*$ with non-ambiguous operations?

Problem SCP was first proposed in [18]. A language S for which Problem SCP has a positive answer is said a *polynomial having solutions* in [18] and a *strong factorizing language* in [5]. Remark that exchanging the roles of S and P gives raise to a dual problem.

Theorem 66. *It is decidable* (*in a constructive way*) *whether given a finite language $S \subseteq \Sigma^*$ there exist finite languages C, P, with C maximal code, such that $\underline{S}\ \underline{C}^*\underline{P} = \underline{\Sigma}^*$.*

Proof. Let $S \subseteq \Sigma^*$ be a finite language. From results in [3,4] we know that it is decidable whether there exists an (infinite) language Z such that $\underline{S}\ \underline{Z} = \underline{\Sigma}^*$, that Z is regular and that an automaton recognizing Z can be constructed starting from one recognizing S. Because $\underline{S}\ \underline{C}^*\underline{P} = \underline{\Sigma}^*$ implies $\underline{S}\ \underline{C^*P} = \underline{\Sigma}^*$, then, if there does not exist a language Z such that $\underline{S}\ \underline{Z} = \underline{\Sigma}^*$, then also Problem SCP has no solution. Otherwise, let Z such a solution. Recall that if finite languages S, C, P satisfy the non-ambiguous equation $SC^*P = \Sigma^*$, then C is necessarily a maximal code [37,8]. The problem reduces to the problem of factorizing in a non-ambiguous way the regular language Z as $Z = C^*P$ with C, P finite languages. In other words, we have to decide whether Z is finitely Z-decomposable and eventually provide a finite decomposition. This problem is solved in previous sections. $\square$

Example 67. Let $S = 1 + a + a^2b$. We know [4,5] that there exists a language Z such that $\underline{S}\ \underline{Z} = \underline{\Sigma}^*$. Further, applying techniques shown in [2], we have that Z is recognized by the automaton in Fig. 3. As shown in Section 6.2, language Z is not finitely Z-decomposable, i.e. there do not exist finite languages C, P such that $\underline{Z} = \underline{C}^*\underline{P}$. We can thus claim that there do not exist finite languages C, P such that $\underline{S}\ \underline{C}^*\underline{P} = \underline{\Sigma}^*$.

We show now how to construct an infinite family of factorizing codes, starting from one of them, as a byproduct of results in previous sections. We define the operation of *substitution* on languages and show that it preserves the property of being a factorizing code.

Definition 68. Given languages $C, C' \neq \emptyset$ a *substitution* of C' in C with respect to $c \in C$ is the language $C \setminus \{c\} \cup cC'$.

Proposition 69. *If C and C' are factorizing codes with factorizations (S, P) and (S, P'), respectively, then $C'' = C \setminus c \cup cC'$ is a factorizing code, $C'' \neq C, C'$. Moreover $(S, P \cup cP')$ is a factorization of C'' and $l(C), l(C') < l(C'')$.*

Proof. By hypothesis, $\underline{S}\ \underline{C}^*\underline{P} = \underline{S}\ \underline{C'}^*\underline{P'} = \underline{\Sigma}^*$. By the uniqueness of language Z such that $\underline{S}\ \underline{Z} = \underline{\Sigma^*}$ [3,4], we have $Z = C^*P = C'^*P'$. Moreover (C, P) and (C', P') are two non-trivial finite decompositions of (Z,Z). The goal thus follows from Proposition 62. □

Corollary 70. *Given a factorizing code* C, *we can construct an infinite family* $\{C_i \mid i \in N\}$ *of factorizing codes.*

Proof. Consider for example $C_0 = C$, and $C_i = C_{i-1} \setminus \{c\} \cup cC_{i-1}$, for some $c \in C_{i-1}$. □

10. Uncited References

[7,16,17,25,27,32]

Acknowledgements

The author wishes to thank the anonymous referees for helpful remarks and suggestions.

References

[1] M. Anselmo, A non-ambiguous language factorization problem, in: G. Rozenberg, W. Thomas (Eds.), Proceedings of the Development on Language Theory 1999, World Scientific, Singapore, 2000, pp. 141–152.
[2] M. Anselmo, Constructing finite maximal codes from Schützenberger Conjecture, in: A. Restivo, S. Ronchi Della Rocca, L. Roversi (Eds.), ICTCS 2001, Lecture Notes in Computer Science, Vol. 2202, Springer, Berlin, 2001, pp. 197–214.
[3] M. Anselmo, A. Restivo, Factorizing languages, in: B. Pehrson, I. Simon (Eds.), Proceedings of the 13th World Computer Congress IFIP 94, Vol. 1, Elsevier, North-Holland, Amsterdam, 1994, pp. 445–450.
[4] M. Anselmo, A. Restivo, On languages factorizing the free monoid, Internat. J. Algebra Comput. 6 (4) (1996) 413–427.
[5] M. Anselmo, C. De Felice, A. Restivo, On some factorization problems, Bull. Belgian Math. Soc. 4 (1997) 25–43.
[6] J.M. Autebert, L. Boasson, M. Latteux, Motifs et bases de langages, RAIRO Inform. Theor. Appl. 23 (4) (1989) 379–393.
[7] G. Bergman, Centralizers in free associative algebras, Trans. Amer. Math. Soc. 137 (1969) 327–344.
[8] J. Berstel, D. Perrin, Theory of Codes, Academic Press, New York, 1985.
[9] J. Berstel, Ch. Reutenauer, Rational series and their languages, EATCS Monographs, Vol. 12, Springer, Berlin, 1988.
[10] A. Bertoni, P. Massazza, On the square root of regular languages, Proceedings of the FPSAC'00, Springer, Berlin, 2000, pp. 125–134.
[11] J.M. Boë, Factorisation par excés du monoïde libre, LIRMM Report 94-005, 1994.
[12] V. Bruyère, Factorisation des Ensembles Préfixiels, RAIRO Inform. Théor. Appl. 23 (3) (1989) 295–315.
[13] V. Bruyère, Research topics in the theory of codes, Bull. EATCS 48 (1992) 412–424.
[14] J. Brzozowsky, Open problems about regular languages, in: R.V. Book (Ed.), Formal Language Theory, Perspectives and Open Problems, Academic Press, New York, 1980.

[15] J. Brzozowsky, R. Cohen, On decomposition of regular events, J. ACM 16 (1969) 132–144.
[16] C. Choffrut, J. Karhumäki, N. Ollinger, The commutation of finite sets: a challenging problem, TUCS Technical Report No. 303, 1999, http://www.tucs.fi/.
[17] J.H. Conway, Regular Algebra and Finite Machines, Chapman & Hall, London, 1971.
[18] C. De Felice, Construction et complétion de codes finis, Thèse de 3ème cycle, Rapport LITP, Vol. 85, No. 3, 1985.
[19] C. De Felice, Construction of a family of finite maximal codes, Theoret. Comput. Sci. 63 (1989) 157–184.
[20] C. De Felice, A partial result about the factorization conjecture for finite variable-length codes, Discrete Math. 122 (1993) 137–152.
[21] S. Eilenberg, Automata, Languages, and Machines, Vol. A, Academic Press, New York, 1974.
[22] P. Enflo, A. Granville, J. Shallit, S. Yu, On sparse languages L such that $LL=\Sigma^*$.
[23] F. Harary, Graph Theory, Addison-Wesley, Reading, MA, 1969.
[24] J.E. Hopcroft, R. Motwani, J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, Reading, MA, 2001.
[25] J. Karhumäki, I. Petre, On the centralizer of a finite set, Proceedings of the ICALP 2000, Lecture Notes in Computer Science, Vol. 1853, Springer, Berlin, 2000, pp. 536–546.
[26] D. Krob, Codes limités et factorisations finies du monoïde libre, RAIRO Inform. Théor. 21 (1987) 437–467.
[27] N.H. Lam, On codes having no finite completion, Proceedings of the STACS 94, Lecture Notes in Computer Science, Vol. 775, Springer, Berlin, 1994, pp. 691–698.
[28] P. Massazza, On the root of Languages, Internat. J. Algebra Comput. 11 (2001) 549–564.
[29] A. Paz, B. Peleg, On concatenative decompositions of regular events, IEEE Trans. Electron. Comput. EC17 (1968) 229–237.
[30] D. Perrin, Finite automata, in: J. van Leeuwen (Ed.), Handbook of Theoretical Computer Science, Vol. B, Elsevier, North-Holland, Amsterdam, 1990.
[31] D. Perrin, M.P. Schützenberger, Un problème élémentaire de la théorie de l'information, Théorie Inform.—Colloq. Internat. CNRS 276 (1977) 249–260.
[32] B. Ratoandromanana, Codes et motifs, RAIRO Inform. Theory 23 (4) (1989) 425–444.
[33] A. Restivo, Some decision results for recognizable sets in arbitrary monoids, Proceedings of ICALP 78, Lecture Notes in Computer Science, Vol. 62, Springer, Berlin, 1978.
[34] C. Reutenauer, Non commutative factorization of variable-length codes, J. Pure Appl. Algebra 36 (1985) 167–186.
[35] A. Salomaa, M. Soittola, Automata-Theoretic Aspects of Formal Power Series, Springer, Berlin, Heidelberg, New York, 1978.
[36] A. Salomaa, S. Yu, On the decomposition of finite languages, in: G. Rozenberg, W. Thomas (Eds.), Proceedings of the Development on Language Theory 1999, World Scientific, Singapore, 2000, pp. 22–31.
[37] M.P. Schützenberger, Sur certains sous-monoïdes libres, Bull. Soc. Math. France 93 (1965) 209–223.
[38] I. Simon, Locally finite semigroups and limited subsets of a free monoid, Universidade de Sao Paulo, 1978, preprint.
[39] L. Zang, C.K. Gu, Two classes of factorizing codes-(p,p)-codes and (4,4)-codes, in: M. Ito, H. Juergensen (Eds.), Words, Languages and Combinatorics II, World Scientific, Singapore, 1994, pp. 477–483.