Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

On the decomposition of prefix codes ☆

Clelia De Felice [a,*], Sabrina Mantaci [b], Antonio Restivo [b]

[a] *Università degli Studi di Salerno, Italy*
[b] *Università degli Studi di Palermo, Italy*

ARTICLE INFO

Article history:
Received 8 April 2015
Received in revised form 30 October 2015
Accepted 4 December 2015
Available online xxxx

Keywords:
Prefix codes
Rational languages
Finite automata
Composition of codes

ABSTRACT

In this paper we focus on the decomposition of rational and maximal prefix codes. We present an effective procedure that allows us to decide whether such a code is decomposable. In this case, the procedure also produces the factors of some of its decompositions. We also give partial results on the problem of deciding whether a rational maximal prefix code decomposes over a finite prefix code.

1. Introduction

The origin of the theory of *(uniquely decipherable) codes* can be found in the early works of Shannon in the 1950's and an algebraic theory of codes was later initiated by Schützenberger, using noncommutative algebra as a tool for studying them [15]. This paper follows Schützenberger's approach according to which a variable-length code X is the base of a free submonoid of A^* [3,15].

The aim of the theory of codes is to give a structural description of codes in a way that allows their construction. This is easily accomplished for *prefix* codes, i.e., codes such that none of their words is a left factor of another, and for the symmetrical class of suffix codes. A classical representation of a finite prefix code X over an alphabet A is as a set of leaves on a tree.

Despite their simplicity, they play an important role in the theory of codes. Most of the interesting problems on codes can be raised for prefix codes or are related with them. In the context of formal language theory, they also intervene in a decomposition algorithm for rational sets (see Theorem 4.4.1 in [6]).

There are some basic problems on prefix codes that are still unsolved. For instance, we still do not know how to decide whether a *rational maximal* prefix code decomposes over a finite prefix code. A prefix code $X \subseteq A^*$ is maximal over A if it is not properly contained in any other prefix code over A.

Composition is a partially binary operation on codes. It stimulated deep interest at the beginning of the theory since it was conjectured that any code could be obtained by composition of prefix and suffix codes. This conjecture was false but other interesting questions related to this operation are still open (see [3,9,10]). The investigation of a class of codes is often

☆ Partially supported by the 60% Project *"Aspetti algebrici e computazionali nella teoria dei codici, degli automi e dei linguaggi formali"* (University of Salerno, 2013 grant ORSA138754), the 60% Project *"Codici e sistemi splicing nella teoria dei linguaggi formali"* (University of Salerno, 2014 grant ORSA145954) and the *MIUR* PRIN 2010–2011 grant H41J12000190001 *"Automata and Formal Languages: Mathematical and Applicative Aspects"*.

* Corresponding author.
E-mail address: defelice@dia.unisa.it (C. De Felice).

http://dx.doi.org/10.1016/j.tcs.2015.12.002
0304-3975/© 2015 Published by Elsevier B.V.

Please cite this article in press as: C. De Felice et al., On the decomposition of prefix codes, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2015.12.002

accompanied with the study of the behavior of the class with respect to this operation and with respect to the converse notion of *decomposition*. Indeed, if a code X decomposes over two codes Y and Z, then these codes are generally simpler. It is worth noting that the terms “composition” and “decomposition” are used with a completely different meaning in [5,7,14], where they are referred to rational operations on rational languages.

In this paper we focus on the decomposition of rational and maximal prefix codes. Of course, there are several known results on this topic, which is presented here systematically for the first time. Our approach takes into account the minimal automaton $\mathcal{A} = (Q, 1, 1)$ recognizing X^*, where X is a rational maximal prefix code. We introduce the definition of *balanced* state and of *unbalanced* state. This definition applies to a state different from 1 and plays a fundamental role in our investigation.

We present an effective procedure that, when applied to $\mathcal{A}$, allows us to decide whether a rational and maximal prefix code X is decomposable. In this case, the procedure also produces the factors of some of the decompositions of X. This procedure applies to special languages associated with states q in $\mathcal{A}$. The definition of such languages and the actions of the procedure depend on whether q is a balanced or an unbalanced state.

We introduce two extremal classes of rational maximal prefix codes, the class of *balanced codes* (all states different from 1 are balanced) and the class of *strongly unbalanced codes* (all states different from 1 are unbalanced). The former is a rather large class, including rational maximal *bifix* codes (codes which are both prefix and suffix) and indecomposable rational maximal prefix codes. The latter turns out to coincide with the class of $(1, 0)$-*limited codes* [3]. We give a sufficient condition for a balanced code to decompose over a finite maximal prefix code. We prove that we may decide whether a rational and maximal bifix code decomposes over a finite (bifix) maximal code. Finally, we prove that any strongly unbalanced code decomposes over a finite maximal prefix code.

This paper is the sequel of an investigation initiated in [13]. A part of the material underlying our constructions is already in the above paper but it is here thoroughly developed. Some of our complementary results are already in [13] but they are stated here with different and easier proofs.

The paper is organized as follows. In Section 2, we set up the basic definitions and known results we need. Section 3 contains a brief presentation of the problems we deal with in this paper. In Section 4 we give the definitions of balanced state and unbalanced state. Our procedure is described in Section 4.1 for balanced states and in Section 4.2 for unbalanced states. Section 5 deals with the family of balanced codes, whereas strongly unbalanced codes are investigated in Section 6. We end this paper with Section 7, where we discuss some main issues that follow on from the results.

2. Basics

2.1. Codes and words

A main reference for this section is [3]. Let A^* be the *free monoid* generated by an alphabet A and let $A^+ = A^* \setminus 1$ where 1 is the empty word. For a word $w \in A^*$ and a letter $a \in A$, we denote by $|w|$ the *length* of w. A word $x \in A^*$ is a *factor* of $w \in A^*$ if there are $u_1, u_2 \in A^*$ such that $w = u_1 x u_2$ and x is a *proper factor* of w if $u_1 u_2 \neq 1$. Furthermore, x is a *prefix* (resp. *suffix*) of w if $u_1 = 1$ (resp. $u_2 = 1$) and x is a *proper prefix* (resp. *proper suffix*) of w if $u_2 \neq 1 = u_1$ (resp. $u_1 \neq 1 = u_2$). In the latter case, we say that x and w are prefix-comparable (resp. suffix-comparable). Then, x is *prefix-comparable* with y if x is a proper prefix of y or y is a proper prefix of x.

A set $X \subseteq A^*$ is *rational* if it is accepted by a finite automaton. It is *thin* if there is a word of A^* which is not a factor of X. Rational languages will be also represented at times by means of rational expressions.

A *code* X is a subset of A^* such that, for all $h, k \geq 0$ and $c_1, \ldots, c_h, c'_1, \ldots, c'_k \in X$, we have

$$c_1 \cdots c_h = c'_1 \cdots c'_k \quad \Rightarrow \quad h = k \quad \text{and} \quad c_i = c'_i \quad \text{for} \quad i = 1, \ldots, h.$$

Any rational code is thin. A set $X \subseteq A^+$ is a *prefix* code if no word in X is prefix-comparable with another word in X, that is, $X \cap XA^+ = \emptyset$. X is a *suffix* code if $X \cap A^+ X = \emptyset$ and X is a *bifix* code when X is both a suffix and a prefix code. A code X is a *maximal* code over A if for each code Y over A such that $X \subseteq Y$ we have $X = Y$. A prefix (resp. bifix) code is maximal over A if it is not properly contained in any other prefix (resp. bifix) subset of A^*. For a thin prefix (resp. bifix) code, the two above conditions are equivalent, thus thin maximal prefix (resp. bifix) codes are maximal codes. A submonoid M of A^* is *right unitary* if for all $u, v \in A^*$

$$u, uv \in M \Rightarrow v \in M.$$

Left unitary monoids are symmetrically defined. A submonoid M of A^* is right (resp. left) unitary if and only if its minimal set of generators is a prefix (resp. suffix) code. We recall that, given a set $Z \subseteq A^+$, there exists a *smallest right unitary submonoid* $\mathcal{S}(Z)$ of A^* containing Z [2]. It can be constructed as follows. Define a sequence $(\mathcal{S}_n)_{n \geq 0}$ of subsets of A^* by setting

$$\mathcal{S}_0 = Z^*, \qquad \mathcal{S}_{n+1} = (\mathcal{S}_n^{-1} \mathcal{S}_n)^*$$

Then, the following result has been stated in [2].