Completion of finite codes with finite deciphering delay

Véronique BRUYERE

Faculté des Sciences, Avenue Maistriau, 15, 7000 MONS (Belgium)

Abstract - We show how to embed any finite code with finite deciphering delay in a rational maximal code with the same deciphering delay. This answers a question in [1].

1 Introduction

A code \( X \subseteq A^* \) is a set of words over an alphabet \( A \) such that when the letters of a message are coded with the words of the code, the deciphering is unique. A code \( X \) is said to be maximal in \( A^* \) if it is not properly included in any code \( Y \subseteq A^* \). It is not very difficult to see that any code is included in a maximal code over the same alphabet. However, only few constructions for obtaining such an embedding are known.

For instance, the embedding is easy for prefix codes defined as codes with none of their words appearing in the beginning of another one: any finite (resp. rational) prefix code is included in a maximal code which is still prefix and finite (resp. rational) [1]. On the other hand, Restivo [9] has shown that there exist finite codes which are not included in any finite maximal code. He conjectured that every finite code is included in a rational maximal code. Ehrenfeucht and Rozenberg [5] proved and generalized this statement by describing a simple algorithm to embed a rational code into a rational maximal one.

Perrin has investigated the family of biprefix codes and he proved in [8] that every finite biprefix code is included in a rational maximal biprefix one.

In this paper, we are interested in codes with finite deciphering delay. When reading from left to right messages coded according to such codes, the deciphering can already begin after a bounded "delay", without waiting for the end of the message. This property is not always verified, as for the coded message \( ab \) using the code \( \{a, ab, bb\} \). In that case, we say that the code has an infinite deciphering delay. On the other hand, prefix codes are called instantaneous because they have the smallest deciphering delay.

The notion of deciphering delay appeared at the very beginning of the theory of codes: Schützenberger [12] has solved a conjecture of Gilbert and Moore [6] where he proved that all finite maximal codes with finite deciphering delay are necessarily prefix codes. The family of codes with finite deciphering delay has some other remarkable properties: Choffrut [3] has obtained a characterization of their decoding function; elementary codes which are particular codes with finite deciphering delay were introduced
by Ehrenfeucht and Rozenberg to successfully solve the DOL sequence equivalence problem [4]; more recently, the notion of deciphering with bounded delay has been investigated in the problem of coding arbitrary sequences into a constrained system of sequences [7].

By Schützenberger result [12] (mentioned before), finite maximal codes are either instantaneous or have an infinite deciphering delay, thus a finite code with finite deciphering delay cannot be embedded in a maximal code with the same properties of finiteness and finite deciphering delay, except if it is a prefix code. Naturally a question raises [1, p.129]: does there a (simple) construction to embed a finite code with finite deciphering delay in a maximal code which is rational and again has finite deciphering delay exist? We can go further: is it possible to keep the same deciphering delay for the maximal code?

We present in this paper a simple algorithm to embed a finite code with finite deciphering delay in a rational maximal code with the same deciphering delay. This construction is completely different from Ehrenfeucht and Rozenberg algorithm [5], because in most cases the latter gives rise to a maximal code with an infinite deciphering delay.

The proposed method consists in associating with the finite code \( X \) a forest \( F(X) \) of finite trees such that every element of the code is described by some path in the forest and such that for each node the set of all its sons is a prefix code. To include the code \( X \) in a maximal one, these prefix codes are completed by new nodes to obtain prefix codes as maximal as possible with respect to the required properties of code and finite deciphering delay. This operation creates new paths in the forest \( F(X) \) that indicate the new words to add to \( X \). We perform the completion of the forest by means of Sardinas and Patterson algorithm [10]. Then we extract from the completed forest \( F(X) \) the rational maximal code which includes \( X \) and has the same deciphering delay.

In this paper, we only give sketches of proof; more detailed proofs will appear in a forthcoming paper [2].

2 Preliminaries

\( A \) is a finite alphabet of letters and \( A^* \) is the free monoid generated by \( A \). Elements of \( A^* \) are called words, including the empty word \( 1 \). \( A^+ \) is the free semigroup \( A^* \{1\} \) generated by \( A \). We denote by \( |w| \) the length of any word \( w \in A^* \).

The basic operations on subsets \( X, Y \) of \( A^* \) are denoted \( + \) for union, \( - \) for difference, \( \cdot \) for concatenation product and \( * \) for star. We will often write \( w \) instead of \( \{w\} \). Let also \( X^+ \) be the set \( X^* - 1 \) and

\[
X^{-1}Y = \{ z \in A^* | \exists x \in X, \exists y \in Y : xz = y \}.
\]

The subsets \( X \) of \( A^* \) obtained from the letters of \( A \) by a finite number of unions, products and stars, are called rational.

Let \( u, v \) be two words of \( A^* \), \( u \) is a (proper) left factor of \( v \) if \( \exists w \in A^* (\in A^+) : uw = v \). This relation defines a partial order on \( A^* \); we write \( u \leq v \) (\( u < v \)). For \( X \subset A^* \), \( LF(X) \) is defined as the set of the nonempty proper left factors of words \( x \in X \). In the same