Journal of Automata, Languages and Combinatorics **25** (2020) 1, 55 – 59

EVERY REGULAR BIFIX CODE IS A FINITE UNION OF REGULAR INFIX CODES

Stavros Konstantinidis[(A)]

Department of Mathematics and Computing Science, Saint Mary's University 923 Robie Str., Halifax, Nova Scotia, B3H 3C3, Canada
s.konstantinidis@smu.ca

ABSTRACT

We show constructively that every regular bifix code is a finite union of regular infix codes. This problem was considered by Jürgen Dassow in the context of studying the generative power of conditional tabled Lindenmayer systems. Our proof is based on some language operators related to independent languages and a lemma of Liang Zhang and Zhonghui Shen about the index of the principal congruence of bifix codes.

Keywords: regular language, bifix code, infix code

1. Introduction

We consider the question of whether a given regular bifix code is equal to a finite union of regular infix codes. This question was raised by Jürgen Dassow in the context of studying the generative power of conditional tabled Lindenmayer systems, when the regular languages associated with tables are subclasses of codes rather than arbitrary, which resulted in [3], here in the same journal issue. In particular, if $\mathcal{X}$ and $\mathcal{Y}$ are language classes such that $\mathcal{Y} \subseteq \mathcal{X}$ and each language in $\mathcal{X}$ is a finite union of languages in $\mathcal{Y}$ then the generative power of certain conditional tabled Lindenmayer systems based on $\mathcal{X}$ is equal to that of systems based on $\mathcal{Y}$.

We show that indeed every regular bifix code is equal to a finite union of regular infix codes. Our proof is constructive and is based on some language operators related to independent languages as well as a beautiful lemma of [15] about the index of the principal congruence of bifix codes.

In the literature, one can find several instances of the general language decomposition problem, that is, the problem of whether a given type $\mathcal{X}$ regular language is a finite union, or concatenation, of type $\mathcal{Y}$ regular languages. In [10], it is shown that every regular language is a finite union of union-free regular languages, and in [1] that the minimum number of terms in the union is computable. A related problem is the representation of a regular language as a union of single-loop languages—these are of

[(A)] Research supported by NSERC.