FURTHER REMARKS ON PARALLEL COMMUNICATING GRAMMAR SYSTEMS WITHOUT A MASTER [1]

GYÖRGY VASZIL

Computer and Automation Research Institute, Hungarian Academy of Sciences
Kende utca 13–17, H-1111 Budapest, Hungary
e-mail: `vaszil@sztaki.hu`

ABSTRACT

In this paper we continue our investigations concerning the generative power of parallel communicating grammar systems working in the competitive and popular derivation modes, introduced in [11], with Chomsky grammars and extended Lindenmayer systems as components. We show that returning and non-returning parallel communicating systems with right-linear or linear grammars as components have the same generative power in these modes as they have in the usual one, which also holds for returning systems with context-free grammars and extended 0L systems as components.

Keywords: Lindenmayer systems, generative capacity, parallel communicating grammar systems.

1. Introduction

Parallel communicating grammar systems are introduced in [7] as formal language theoretic models of parallel and distributed computation. A parallel communicating grammar system consists of several language generating devices (for example Chomsky type grammars or Lindenmayer systems) which generate one language by using a certain cooperation strategy. In the variant considered in this paper the components work on their own sentential form in a parallel and synchronized manner, each executing one rewriting step at a time. Communication is based on requests, realized by a set of special nonterminals called query symbols, one such symbol denoting each component. Whenever a query symbol appears in a string, the rewriting process is interrupted and one or more communication steps must follow. Each query symbol must be replaced by the current sentential form of the corresponding component before the system can continue rewriting its sentential forms. After communication the system can behave in basically two ways. In returning systems the components which have sent their sentential form during the communication start to generate a new string, while in non-returning systems they continue to process the same string

---