Journal of Automata, Languages and Combinatorics **19** (2014) 1–4, 33–44

NONTERMINAL CONTROLLED STRING ASSEMBLING SYSTEMS

Henning Bordihn
Institut für Informatik, Universität Potsdam,
August-Bebel-Straße 89, 14482 Potsdam, Germany
e-mail: henning@cs.uni-potsdam.de

Martin Kutrib and Matthias Wendlandt
Institut für Informatik, Universität Giessen,
Arndtstraße 2, 35392 Giessen, Germany
e-mail: {kutrib, matthias.wendlandt}@informatik.uni-giessen.de

ABSTRACT

String assembling systems are biologically inspired mechanisms that generate strings by assembling substrings to the upper and the lower strand of double-stranded strings. We consider the variant where, at both ends, auxiliary (nonterminal) symbols may appear as encodings of ordinary (terminal) symbols, controlling the string assembling process. Those bidirectional systems are compared with unidirectional ones, where substrings can only be assembled at the right end. We show that, in contrast to string assembling systems without auxiliary symbols, bi- and unidirectional nonterminal controlled string assembling systems are equally powerful and characterize the family of languages accepted by nondeterministic one-way two-head finite automata. Some further results are derived comparing nonterminal controlled string assembling systems with traditional complexity and formal language classes.

Keywords: String Assembling, Double-Stranded Sequences, Nonterminal Controlled, Two-Head Finite Automata

1. Introduction

String processing and string generation are traditional subjects in computer science. Among the realms of string generating mechanisms there is a bunch of devices inspired by biological processes. Examples are Lindenmayer systems [5], splicing systems and sticker systems [4]. The latter two types of devices model operations on DNA molecules and are therefore based upon double stranded strings as raw material of the string generation process, where corresponding symbols are uniquely related. String assembling systems, introduced in [3], are another string generating mechanism using double strands, where corresponding symbols in the upper and the lower strand have to be identical. This simplification is also encountered when instances of Post's Correspondence Problem are viewed as devices generating strings by completing matching double strands.