

STRING ASSEMBLY IN NETWORKS OF EVOLUTIONARY PROCESSORS

ERZSÉBET CSUHAJ-VARJÚ KATALIN ANNA LÁZÁR

*Department of Algorithms and Their Applications, Faculty of Informatics,
Eötvös Loránd University
Pázmány Péter sétány 1/c, 1117 Budapest, Hungary
csuhaj@inf.elte.hu kati@caesar.elte.hu*

ABSTRACT

In this paper, we introduce a new way of communication in networks of evolutionary processors (NEPs) where the filtered communicated strings are appended according to different variants of the overlap operation. These NEPs aim at modelling self-assembly in complex biological systems. We show that these networks with three nodes are computationally complete devices, furthermore they are able to solve NP-complete problems efficiently.

Keywords: string assembly, networks of evolutionary processors, computational completeness, Hamiltonian path problem

1. Introduction

Self-assembly is the process during which simple objects autonomously aggregate into larger structures. A great deal of effort has been devoted to studying self-assembly as a formal language theoretic operation and developing a formal framework to treat DNA-based information and DNA-based computation. A first example of this trend is the mathematical formalism for biological phenomena of recombinant processes proposed in 1987 by Tom Head. In his seminal paper, he introduced a formalism of splicing systems to describe the recombination of DNA strands under the cut-and-paste action of restriction enzymes and ligases [18]. Systems based on other types of bio-operations include insertion/deletion systems [20, 32, 33], substitution based systems modelling errors occurring in DNA-encoded information [21], systems using hairpin completion [6] and hairpin reduction [27], superposition [4, 28], PAMatching [22] and overlapping concatenation [30].

In this paper we study properties of a formal language theoretic operation that models the linear self-assembly of partially overlapping DNA strands. During the overlap operation, the input strands xy and yz with a nonempty common overlap, y , produce the output xyz . Overlap assembly was introduced in [9] (where it was called (self)-assembly of strings and languages), and later refined in [4, 28] and [13, 14].