Information and Computation 253 (2017) 143–154

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Further remarks on DNA overlap assembly ☆

Srujan Kumar Enaganti [a], Oscar H. Ibarra [b], Lila Kari [a,c,*], Steffen Kopecki [a]

[a] Department of Computer Science, University of Western Ontario, London, ON, N6A 5B7, Canada
[b] Department of Computer Science, University of California, Santa Barbara, CA, 93106, USA
[c] School of Computer Science, University of Waterloo, Waterloo, ON, N2L 3G1, Canada

ARTICLE INFO

Article history:
Received 21 July 2015
Received in revised form 8 November 2016
Available online 19 January 2017

Keywords:
DNA computing
Bio-operations
DNA self-assembly
Overlap assembly

ABSTRACT

The operation of *overlap assembly* was defined by Csuhaj-Varjú, Petre, and Vaszil as a formal model of the linear self-assembly of DNA strands: The overlap assembly of two strings, xy and yz, which share an "overlap" y, results in the string xyz. This paper continues the exploration of the properties of the overlap assembly operation by investigating closure of various language classes under iterated overlap assembly, and the decidability of the completeness of a language. It also investigates the problem of deciding whether a given string is terminal with respect to a language, and the problem of deciding if a given language can be generated by an overlap assembly operation of two given others.

1. Introduction

The word and language operation *overlap assembly* was first introduced by Csuhaj-Varjú, Petre, and Vaszil – under the name (self-)assembly – in [1], and later studied in [2], as a formal model of the linear self-assembly of DNA strands. Formally, the overlap assembly is a binary operation which, when applied to two input strings xy and yz (where y is their nonempty *overlap*), produces the output xyz.

The study of overlap assembly as a formal language operation is part of ongoing efforts to provide a formal framework and rigorous treatment of DNA-based information and DNA-based computation. More specifically, this study can be placed in the context of studies of DNA bio-operations enabled by the action of the DNA polymerase enzyme, such as hairpin completion and its inverse operation, hairpin reduction [3–6], overlapping concatenation [7], and directed extension [8].

The activity of DNA Polymerase presupposes the existence of a DNA single strand, called *template*, and of a second short DNA strand, called *primer*, that is Watson–Crick complementary to the template and binds to it. Given a supply of individual nucleotides, DNA polymerase then extends the primer, at one of its ends only, by adding individual nucleotides complementary to the template nucleotides, one by one, until the end of the template is reached. In the wet lab, the iteration of this process is used to obtain an exponential replication of DNA strands, in a protocol called *Polymerase Chain Reaction (PCR)*. Experimentally, (parallel) overlap assembly of DNA strands under the action of the DNA Polymerase enzyme was used for gene shuffling in, e.g., [9]. In the context of experimental DNA computing, overlap assembly was used in, e.g., [10–13] for the formation of combinatorial DNA or RNA libraries. Overlap assembly can also be viewed as modeling a special case of an experimental procedure called cross-pairing PCR, introduced in [14] and studied in, e.g., [15–18].

☆ This research was supported by Natural Science and Engineering Council of Canada (NSERC) Discovery Grant R2824A01 and a University of Western Ontario Grant to L.K., and US NSF Grant CCF-1117708 to O.H.I.

* Corresponding author at: Department of Computer Science, University of Western Ontario, London, ON, N6A 5B7, Canada.
E-mail address: lila.kari@uwo.ca (L. Kari).

http://dx.doi.org/10.1016/j.ic.2017.01.009
0890-5401/

This paper is a continuation of the theoretical analysis of overlap assembly as a formal language operation, that was started in [1] and [2]. In [1], the authors proposed a formal language operation called "self-assembly", inspired by the linear self-assembly of DNA single strands via Watson–Crick complementarity. The authors obtained closure properties of various language classes under iterated overlap assembly, and studied the question of whether or not a given language can be generated through assembly and, if so, what is the minimal generator. In [2], the aforementioned formal operation of self-assembly was renamed overlap assembly, to avoid confusion with other usages of the syntagm self-assembly, such as in the context of DNA computing by two-dimensional self-assembly of DNA rectangular tiles, [19–21]. The paper [2] explored closure properties of basic language families under overlap assembly, decision problems, as well as the potential use of iterated overlap assembly to generate combinatorial DNA libraries.

In this paper, following Section 2 comprising definitions, notations and basic properties, in Section 3 we correlate the overlap assembly operation with the superposition operation introduced in [22] and determine additional closure properties of some language classes under iterated overlap assembly. A string w is terminal with respect to a language L if $w \in L$ and the result of the overlap assembly between w and L equals $\{w\}$; the terminal set $T(L)$ contains all words that are terminal with respect to L. In Section 4 we investigate the closure properties of terminal sets of complete languages (languages closed under overlap assembly). In Section 5 we study three decision problems: deciding the completeness of an arbitrary language, deciding whether a string is terminal with respect to a language, and deciding whether a language is generated by an overlap assembly operation between two given languages. Section 6 contains concluding remarks.

2. Basic definitions and notations

An alphabet Σ is a finite nonempty set of symbols. Σ^* denotes the set of all words over Σ, including the empty word λ. Σ^+ is the set of all nonempty words over Σ. For words $w, x, y, z \in \Sigma^*$ such that $w = xyz$ we call the subwords x, y, and z *prefix*, *infix*, and *suffix* of w, respectively. The sets $\mathrm{pref}(w)$, $\mathrm{inf}(w)$, and $\mathrm{suff}(w)$ contain, respectively, all prefixes, infixes, and suffixes of w. A prefix (resp., infix or suffix) x of w is *proper* if $x \neq w$. We employ the following notation: $\mathrm{Pref}(w) = \mathrm{pref}(w) \setminus \{w\}$, $\mathrm{Inf}(w) = \mathrm{inf}(w) \setminus \{w\}$, and $\mathrm{Suff}(w) = \mathrm{suff}(w) \setminus \{w\}$. This notation is naturally extended to languages; for example, $\mathrm{Suff}(L) = \bigcup_{w \in L} \mathrm{Suff}(w)$. The complement of a language $L \subseteq \Sigma^*$ is $L^c = \Sigma^* \setminus L$.

Let $\mathbb{N}$ be the set of non-negative integers and k be a positive integer. A subset Q of $\mathbb{N}^k$ is a *linear set* if there exist vectors $\vec{v}_0, \vec{v}_1, \ldots, \vec{v}_n \in \mathbb{N}^k$ such that $Q = \{\vec{v}_0 + i_1\vec{v}_1 + \cdots + i_n\vec{v}_n \mid i_1, \ldots, i_n \in \mathbb{N}\}$. A finite union of linear sets is called a *semilinear set*.

Let $\Sigma = \{a_1, \ldots, a_k\}$. The *Parikh map* of a language $L \subseteq \Sigma^*$, denoted $\Psi(L)$, is defined as

$$\Psi(L) = \{(|w|_{a_1}, \ldots, |w|_{a_k}) \mid w \in L\},$$

where $|w|_{a_i}$ is the number of a_i's in w.

2.1. The overlap assembly

An *involution* is a function $\theta : \Sigma^* \to \Sigma^*$ with the property that θ^2 is the identity. θ is called an *antimorphism* if $\theta(uv) = \theta(v)\theta(u)$. Traditionally, the Watson–Crick complementarity of DNA strands has been modeled as an antimorphic involution over the DNA alphabet $\Delta = \{A, C, G, T\}$, [23,24].

Using the convention that a word x over this alphabet represents the DNA single strand x in the $5'$ to $3'$ direction, the overlap assembly of a strand uv with a strand $\theta(w)\theta(v)$ first forms a partially double-stranded DNA molecule with v in uv and $\theta(v)$ in $\theta(w)\theta(v)$ attaching to each other, see Fig. 1(a). The DNA polymerase enzyme will extend the $3'$ end of uv with the strand w, see Fig. 1(b). Similarly, the $3'$ end of $\theta(w)\theta(v)$ will be extended, resulting in a full double strand whose upper strand is uvw, see Fig. 1(c). Formally, the overlap assembly between uv and $\theta(w)\theta(v)$ is uvw. Assuming that all involved DNA strands are initially double-stranded, that is, whenever the strand x is available, its Watson–Crick complement $\theta(x)$ is also available, one can further simplify this model and, given two words x, y over an alphabet Σ, define the *overlap assembly of x with y*, [1], as:

$$x \overline{\odot} y = \{z \in \Sigma^+ \mid \exists u, w \in \Sigma^*, \exists v \in \Sigma^+ : x = uv, y = vw, z = uvw\}.$$

The definition of overlap assembly can be extended to languages in the natural way.

The *iterated overlap assembly* $\mu_*(L)$ of a language L is defined as:

$$\mu_0(L) = L, \qquad \mu_{i+1}(L) = \mu_i(L) \overline{\odot} \mu_i(L), \qquad \mu_*(L) = \bigcup_{i \geq 0} \mu_i(L).$$

Since $w \in w \overline{\odot} w$ for any nonempty word w, it easily follows that $\mu_i(L) \subseteq \mu_{i+1}(L)$ for $L \in \Sigma^+$.

A string $w \in L$ is said to be *terminal* with respect to the language L if $w \overline{\odot} L = L \overline{\odot} w = \{w\}$. A set of strings $T(L) \subseteq L$ is said to be the *(maximal) terminal set* of L if every $w \in T(L)$ is terminal with respect to L and for all $w \in L \setminus T(L)$, w is not terminal with respect to L, that is,

$$T(L) = \{w \in L \mid w \overline{\odot} L = L \overline{\odot} w = \{w\}\}.$$