Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Outfix-guided insertion ☆

Da-Jung Cho [a], Yo-Sub Han [a], Timothy Ng [b], Kai Salomaa [b,*]

[a] Department of Computer Science, Yonsei University, 50, Yonsei-Ro, Seodaemum-Gu, Seoul 120-749, Republic of Korea
[b] School of Computing, Queen's University, Kingston, Ontario K7L 3N6, Canada

ARTICLE INFO

Article history:
Received 10 October 2016
Received in revised form 27 February 2017
Accepted 10 March 2017
Available online xxxx

Keywords:
Language operations
Closure properties
Regular languages

ABSTRACT

Motivated by work on bio-operations on DNA strings, we consider an outfix-guided insertion operation that can be viewed as a generalization of the overlap assembly operation on strings studied previously. As the main result we construct a finite language L such that the outfix-guided insertion closure of L is non-regular. We consider also the closure properties of regular and (deterministic) context-free languages under the outfix-guided insertion operation and decision problems related to outfix-guided insertion. Deciding whether a language recognized by a deterministic finite automaton is closed under outfix-guided insertion can be done in polynomial time. The complexity of the corresponding question for nondeterministic finite automata remains open.

1. Introduction

Gene insertion and deletion are basic operations occurring in DNA recombination in molecular biology. Recombination creates a new DNA strand by cutting, substituting, inserting, deleting or combining other strands. Possible errors in this process impair the function of genes. Errors in DNA recombination cause mutation that plays a part in normal and abnormal biological processes such as cancer, the immune system, protein synthesis and evolution [1]. Since mutational damage may or may not be easily identifiable, researchers deliberately generate mutations so that the structure and biological activity of genes can be examined in detail. *Site-directed mutagenesis* is one of the most important techniques in laboratory for generating mutations on specific sites of DNA using PCR (polymerase chain reaction) based methods [2,3]. For a site-directed insertion mutagenesis by PCR, the mutagenic primers are typically designed to include the desired change, which could be base addition [4,5]. This enzymatic reaction occurs in the test tube with a DNA strand and predesigned primers in which the DNA strand includes a target region, and a predesigned primer includes a complementary region of the target region. The complementary region of primers leads it to hybridize the target DNA region and generate a desired insertion on a specific site as a mutation. Fig. 1 illustrates the procedure of site-directed insertion mutagenesis by PCR.

In formal language theory, the insertion of a string means adding a substring to a given string and deletion of a string means removing a substring. The insertions occurring in DNA strands are in some sense context-sensitive and Kari and Thierrin [6] modeled such bio-operations using *contextual insertions and deletions* [7,8]. A finite set of insertion–deletion rules, together with a finite set of axioms, can be viewed as a language generating device. Contextual insertion–deletion systems in the study of molecular computing have been used e.g. by Daley et al. [9], Enaganti et al. [10], Krassovitskiy

☆ An extended abstract of this paper appeared in the *Proceedings of the 20th International Conference Developments in Language Theory*, DLT 2016, Lect. Notes Comput. Sci. 9840, Springer-Verlag, 2016, pp. 102–113.

* Corresponding author.
E-mail addresses: dajungcho@yonsei.ac.kr (D.-J. Cho), emmous@yonsei.ac.kr (Y.-S. Han), ng@cs.queensu.ca (T. Ng), ksalomaa@cs.queensu.ca (K. Salomaa).

http://dx.doi.org/10.1016/j.tcs.2017.03.040
0304-3975/

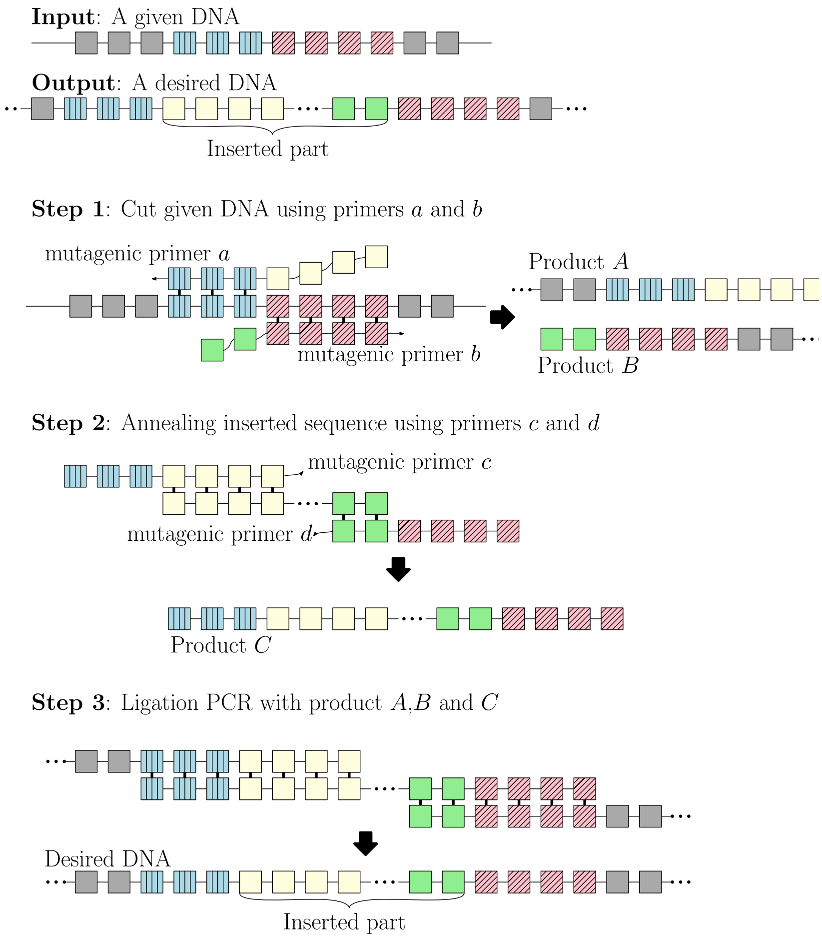

Fig. 1. An example of site-directed insertion mutagenesis by PCR. Given a DNA sequence and four predesigned primers a, b, c and d, two primers a and b lead the DNA sequence to break and extend into two products A and B under enzymatic reaction (Step 1). Two primers c and d complementarily bind to desired insertion region according to the overlapping region and extend into product C (Step 2). Then, the products A, B and C join together to create recombinant DNA that include the desired insertion (Step 3).

et al. [11] and Takahara and Yokomori [12]. Further theoretical studies on the computational power of insertion–deletion systems were done e.g. by Margenstern et al. [13] and Păun et al. [14]. Enaganti et al. [10] have studied related operations to model the action of DNA polymeraze enzymes.

We formalize site-directed insertion mutagenesis by PCR and define a new operation *outfix-guided insertion* that *partially* inserts a string y into a string x when two non-empty substrings of x match with an outfix of y, see Fig. 2(b). We will consider also variants where only a prefix or a suffix of y must match with a non-empty substring of x at the position where the insertion occurs. The outfix-guided insertion is an overlapping variant of the ordinary insertion operation, analogously as the overlap assembly [15–17], cf. Fig. 2(a), is a variant of the ordinary string concatenation operation. An operation equivalent to overlap assembly has been considered under the name chop of languages by Holzer et al. [18]. Holzer and Jacobi [19] have given tight state complexity bounds for a variant of the operation where the overlapping string always has length one. Furthermore, Cărăuşu and Păun [20] have considered another related operation called short concatenation.

This paper investigates the language theoretic closure properties of outfix-guided insertion and iterated outfix-guided insertion. Note that since outfix-guided insertion, similarly as overlap assembly, is not associative, there are more than one way to define the iteration of the operation. We consider a general outfix-guided insertion closure of a language which is defined analogously as the iterated overlap assembly by Enaganti et al. [16]. Iterated (overlap) assembly is defined by Csuhaj-Varju et al. [15] in a different way, which we call right one-sided iteration of an operation.

It is fairly easy to see that regular languages are closed under outfix-guided insertion. Closure of regular languages under iterated outfix-guided insertion turns out to be less obvious. It is well known that regular languages are not closed under the iteration of the ordinary (non-overlapping) insertion operation [21] and it is also fairly easy to establish that iterated prefix-guided (or suffix-guided) insertion does not preserve regularity. However, the known counter-examples, nor their variants, do not work for iterated outfix-guided insertion. Here using a more involved construction we show that there