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

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Computing generalized de Bruijn sequences

F. Blanchet-Sadri [a,*], Sinziana Munteanu [b]

[a] Department of Computer Science, University of North Carolina, P.O. Box 26170, Greensboro, NC 27402–6170, USA
[b] Department of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213–3891, USA

A R T I C L E I N F O

Article history:
Received 12 April 2016
Received in revised form 24 August 2017
Accepted 11 September 2017
Available online xxxx
Communicated by D. Perrin

Keywords:
Algorithms on words
Combinatorics on words
Graph theory
Partial words
Subwords
Representable sets

A B S T R A C T

De Bruijn sequences of order n *represent* the set A^n of all words of length n over a given alphabet A in the sense that they contain occurrences of each of these words (they actually contain exactly one occurrence of each of these words). Recently, the computational problem of representing subsets of A^n by *partial words*, which are sequences that may have holes or don't-care symbols that match each letter of A, was considered and shown to be in $\mathcal{NP}$. However, membership in $\mathcal{P}$ remained open. In this paper, we show that deciding if a subset S of A^n is representable by a partial word can be done in polynomial time with respect to the size $n|S|$ of the input. We also describe a polynomial-time algorithm that determines all integers h for representation by partial words with exactly h holes. Moreover, our algorithms construct representation words. Our approach is graph theoretical.

1. Introduction

A de Bruijn sequence of order n is a cyclic sequence over an alphabet A where each of the words of length n over A occurs as a subword exactly once [8,9] (the sequences studied in this paper are not cyclic). The de Bruijn sequences can be efficiently constructed by taking an Eulerian cycle of a de Bruijn graph where every word of length $n-1$ corresponds to a vertex and every word of length n corresponds to an edge (for alternative constructions see, e.g., [11,18]). De Bruijn sequences are useful and appear in a variety of contexts, e.g., combinatorics on words [1], modern public-key cryptographic schemes, pseudo-random number generation [23], digital fault testing, position sensing schemes [22], non-linear shift registers [10], coding [17], data compression, etc. A vast literature on the de Bruijn sequences exists and generalizations have been explored (e.g., [6,7,12,14–16,19,24]).

Algorithmic combinatorics on *partial words* has been developing in the past several years (e.g., [2]). Partial words over an alphabet A are sequences from $A_\diamond = A \cup \{\diamond\}$, where $\diamond \notin A$ is the hole symbol which matches every letter in A (*total words* are sequences without holes). If w is a partial word over A, then a *factor* of w is a block of consecutive symbols of w and a *subword* of w is a total word over A that can be obtained by replacing the holes in a factor of w by symbols from the alphabet. For instance, if we consider the partial word $01\diamond1000$ with one hole over $\{0,1\}$, the total words $101, 111$ are the subwords corresponding to the factor $1\diamond1$. For any partial word w and integer $n \geq 0$, denote by $\mathrm{sub}_w(n)$ the set of subwords of w of length n.

Let S be a set of total words of length n and let $h \geq 0$ be an integer. A partial word w (respectively, partial word w with h holes) such that $\mathrm{sub}_w(n) = S$ is a *representation word* (respectively, *h-representation word*) for S. The set S is *representable*

* Corresponding author.
E-mail addresses: blanchet@uncg.edu (F. Blanchet-Sadri), smuntean@cs.cmu.edu (S. Munteanu).

http://dx.doi.org/10.1016/j.tcs.2017.09.010
0304-3975/

(respectively, *h-representable*) if there exists a representation word (respectively, h-representation word) for S. If we consider $S = \{000, 001, 010, 100, 101\}$, then S can be 0-represented by $w = 00010100$, 1-represented by $\diamond 00101$, and 2-represented by $0\diamond 0\diamond$. As we allow more holes, the representation word shrinks.

Let Rep be the problem of deciding whether S is representable and h-Rep be the problem of deciding whether S is h-representable. Blanchet-Sadri and Simmons [5] showed that Rep is in $\mathcal{NP}$. Moreover, they showed that a certain subproblem of Rep is in $\mathcal{P}$, namely the problem of deciding whether a set S of words of length n can be represented by a partial word w, such that every subword of length $n-1$ of the words in S occurs exactly once in w or in other words, every such subword can be obtained from exactly one factor of w of length $n-1$ in which the holes are replaced by symbols from the alphabet of S. However, whether or not Rep is in $\mathcal{P}$ remained an open problem. They also gave a polynomial-time algorithm (polynomial in the input size $n|S|$) for deciding h-Rep, thus showed that h-Rep is in $\mathcal{P}$, and their algorithm actually constructs an h-representation word. However, the actual exponent grows quickly with h. Tan and Shallit [20] recently studied representable sets of words of equal length and focused on (circular) representation by total words, and Blanchet-Sadri and Lohr [3] showed how to compute minimum length representations by total words.

This paper continues investigating representability of sets of words of equal length by partial words. Its contents are as follows: In Section 2, we review some background material on partial words. In Section 3, we discuss our graph theoretical approach to Rep. Given a set S of words of equal length n, we describe a decomposition of the Rauzy graph of order $n-1$ associated with S into subgraphs, called blocks, that play a central role in our paper. Rauzy graphs are useful for studying subwords and are closely related to the de Bruijn graphs. A Rauzy graph is a subgraph of a de Bruijn graph, while a de Bruijn graph is a Rauzy graph associated with the set of all words of a given length. In Section 4, we describe polynomial-time algorithms for generating the factor set, S^i, and its extension, $\mathrm{Ext}(S^i)$, related to each block i. In Section 5, using the factor sets and their extensions, we give a polynomial-time algorithm (in the size $n|S|$ of the input) for deciding Rep, settling the question "Is Rep in $\mathcal{P}$?". Our algorithm constructs a representation word if S is representable. In Section 6, we introduce AllRep as the problem of determining all integers h for which S is h-representable and show that this problem is in $\mathcal{P}$. In Section 7, we describe a more efficient algorithm than the one in [5] to solve h-Rep. Finally in Section 8, we conclude with some remarks. Part of this paper was presented at IWOCA'13 (the membership of Rep in $\mathcal{P}$ appeared in [4], while the membership of AllRep in $\mathcal{P}$ and the more efficient algorithm for h-Rep are new in this paper).

2. Preliminaries

We review some basic concepts.

A *partial word* w over a non-empty finite alphabet A is a sequence of symbols from $A_\diamond = A \cup \{\diamond\}$, where $\diamond \notin A$ is the hole symbol (a *total word* over A is just a sequence of letters from A). The *length* of w, denoted by $|w|$, is the number of symbols in w. The symbol at position i is denoted by $w[i]$ and the factor $w[i] \cdots w[j-1]$ by $w[i..j)$. A position i is *defined* if $w[i] \in A$ and it is a *hole* if $w[i] = \diamond$. The number of holes in w is denoted by h_w.

For two partial words w and w' of equal length, w' is *contained in* w, denoted by $w' \subset w$, or w *contains* w', denoted by $w \supset w'$, if $w[i] = w'[i]$ for all defined positions i in w'. For example, $\diamond ab\diamond a \subset \diamond abba$. Say that w and w' are *compatible*, denoted by $w \uparrow w'$, if $w[i] = w'[i]$ for all positions i that are defined in both w and w'. For example, $\diamond ab\diamond a \uparrow \diamond a\diamond ba$. A *completion* of a given partial word is a total word compatible with it. Another way to define compatibility is that w and w' share a common completion, e.g., $01\diamond 0\diamond$ and $\diamond 1\diamond 01$ share the completion 01001 and are thus compatible.

The *greatest lower bound* of two partial words u and v of equal length is the partial word w defined by $w \subset u$, $w \subset v$, and if $w' \subset u$ and $w' \subset v$ then $w' \subset w$. For example, the greatest lower bound of $01\diamond 11$ and $\diamond 1\diamond 0\diamond$ is $\diamond 1\diamond\diamond\diamond$.

A *factor* of a partial word w over A is a block of consecutive symbols of w and a *subword* of w is a total word over A compatible with a factor of w. For any integer $n \geq 0$, denote by $\mathrm{sub}_w(n)$ the set of subwords of w of length n and write $\mathrm{sub}(w) = \cup_{0 \leq n} \mathrm{sub}_w(n)$. For any set of partial words S and any $n \geq 0$, denote by $\mathrm{sub}_S(n) = \cup_{s \in S} \mathrm{sub}_s(n)$ the set of subwords of length n of the words in S, and write $\mathrm{sub}(S) = \cup_{s \in S} \mathrm{sub}(s)$.

3. Graph theoretical approach to Rep

For any graph G, we denote by $V(G)$ the set of vertices of G and by $E(G)$ the set of edges of G. For any $V' \subseteq V(G)$, we denote by $G[V']$ the subgraph of G induced by V'. A digraph G is *strongly connected* if, for every pair of vertices u and v, there exists a path from u to v. Computing the strongly connected components of G can be done in $O(|V(G)| + |E(G)|)$ time by using Tarjan's algorithm [21]. For background material on graph theory, we refer the reader to [13].

Let S be a finite set of total words of length n. Define the *Rauzy graph* of order $n-1$, associated with S, with its set of vertices consisting of $\mathrm{sub}_S(n-1)$ and its set of edges consisting of $\{(s[0..n-1), s[1..n)) \mid s \in S\}$. For every $s \in S$, we label edge $(s[0..n-1), s[1..n))$ by s. Note that a 0-representation word for S, if any exists, is a path in the Rauzy graph of order $n-1$ associated with S that visits every edge at least once. For the remainder of this paper, S will denote a finite set of total words of length n and G will denote the Rauzy graph of order $n-1$ associated with S.

Definition 1 *(Decomposition into blocks).* A partition $G = \bigcup_{i=0}^{p} G^i$ is the *decomposition of G into blocks* $G^0, \ldots, G^p$, called blocks $0, \ldots, p$ of G, if