ARTICLE IN PRESS

Discrete Mathematics ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Discrete Mathematics

journal homepage: www.elsevier.com/locate/disc

A generalized Goulden–Jackson cluster method and lattice path enumeration

Yan Zhuang

Department of Mathematics, Brandeis University, United States

ARTICLE INFO

Article history:
Received 27 September 2016
Received in revised form 16 August 2017
Accepted 4 September 2017
Available online xxxx

Keywords:
Goulden–Jackson cluster method
Free monoids
Lattice paths
Motzkin paths
Generating functions
Statistics

ABSTRACT

The Goulden–Jackson cluster method is a powerful tool for obtaining generating functions counting words in a free monoid by occurrences of a set of subwords. We introduce a generalization of the cluster method for monoid networks, which generalize the combinatorial framework of free monoids. As a sample application of the generalized cluster method, we compute bivariate and multivariate generating functions counting Motzkin paths – both with height bounded and unbounded – by statistics corresponding to the number of occurrences of various subwords, yielding both closed-form and continued fraction formulas.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Given a finite or countably infinite set A, let A^* be the set of all finite sequences of elements of A, including the empty sequence. We call A an *alphabet*, the elements of A *letters*, and the elements of A^* *words*. By defining an associative binary operation on two words by concatenating them, we see that A^* is a monoid under the operation of concatenation (where the empty word is the identity element), and we call A^* the *free monoid* on A. The length $l(\alpha)$ of a word $\alpha \in A^*$ is the number of letters in α. For $\alpha, \beta \in A^*$, we say that β is a *subword* of α if $\alpha = \gamma_1\beta\gamma_2$ for some $\gamma_1, \gamma_2 \in A^*$, and in this case we also say α *contains* β.

More generally, a *free monoid* is a monoid isomorphic to a free monoid on some alphabet. The combinatorial framework of free monoids is useful for the study of combinatorial objects that can be uniquely decomposed into sequences of "prime elements", corresponding to letters in an alphabet. This framework can furthermore be generalized using what are called "monoid networks", which were first introduced by Gessel [9, Chapter 6] in a slightly different yet equivalent form called "G-systems".[1] Roughly speaking, a monoid network consists of a digraph G with each arc assigned a set of letters from an alphabet A, in which the set of sequences of arcs in G is given a monoid structure and is equipped with a monoid homomorphism.

The Goulden–Jackson cluster method allows one to determine the generating function for words in a free monoid A^* by occurrences of words in a set $B \subseteq A^*$ as subwords in terms of the generating function for what are called "clusters" formed by words in B, which is easier to compute. As its name suggests, this celebrated result was first given by Goulden and Jackson in [10]. The cluster method has seen a number of extensions and generalizations [1,5,11,13,15,20,21], and the cluster method itself can be viewed as a generalization of the Carlitz–Scoville–Vaughan theorem, which allows one to count words in a free monoid avoiding a specified set of length 2 subwords.

E-mail address: zhuangy@brandeis.edu.

[1] The term "G-system" was dropped at the request of Ira Gessel, who prefers the name "monoid network" given by the author.

http://dx.doi.org/10.1016/j.disc.2017.09.004
0012-365X/© 2017 Elsevier B.V. All rights reserved.

Please cite this article in press as: Y. Zhuang, A generalized Goulden–Jackson cluster method and lattice path enumeration, Discrete Mathematics (2017), http://dx.doi.org/10.1016/j.disc.2017.09.004.

In this paper, we give a new generalization of the Goulden–Jackson cluster method of a different flavor: we generalize the cluster method to monoid networks, which gives a way of counting words in A^* corresponding to walks between two specified vertices in G (that is, words in a regular language if the alphabet A is finite) by occurrences of subwords in a set B. Then the original version of the cluster method corresponds to the special case in which G consists of a single vertex with a loop to which the entire alphabet A is assigned.

The organization of this paper is as follows. In Section 2, we give an expository account of the original Goulden–Jackson cluster method. In Section 3, we introduce the combinatorial framework of monoid networks and present our generalization of the cluster method for monoid networks. Finally, in Section 4, we demonstrate how our monoid network version of the cluster method can be used to tackle problems in lattice path enumeration.

Although many types of lattice paths can be represented as walks in certain digraphs, in this paper we focus on Motzkin paths, which are paths in $\mathbb{Z}$ beginning and ending at 0 with steps -1, 0, and 1 (also called "down steps", "flat steps", and "up steps", respectively). We consider both regular Motzkin paths and Motzkin paths bounded by height, and our results include bivariate and multivariate generating functions counting these paths by ascents, plateaus, peaks, and valleys – all of which are statistics that are determined by occurrences of various subwords in the underlying word of the Motzkin path – as well as generating functions for Motzkin paths with restrictions on the heights at which these subwords can occur, yielding both closed-form and continued fraction formulas. Several interesting identities are uncovered along the way.

2. The Goulden–Jackson cluster method

We begin this section with a motivating problem: let A be a finite or countably infinite alphabet and suppose that we want to count words in A^* that do not contain a specified set B of forbidden subwords of length at least 2. The Goulden–Jackson cluster method allows us to count this restricted set of words by counting "clusters" formed by words in B, which we shall define shortly.

Given a word $\alpha = a_1a_2\cdots a_n \in A^*$ (where the a_i are letters) and a set $B \subseteq A^*$, we say that (i, β) is a *marked subword* of α if $\beta \in B$ and

$$\beta = a_ia_{i+1}\cdots a_{i+l(\beta)-1},$$

that is, β is a subword of α starting at position i. Moreover, we say that (α, S) is a *marked word* on α if $\alpha \in A^*$ and S is any set of marked subwords of α.

For example, suppose that $A = \{a, b, c\}$ and $B = \{abc, bca\}$. Then

$$\{abcabbcabc, \{(1, abc), (2, bca), (6, bca)\}\}, \tag{1}$$

is a marked word which can also be displayed as

$$(a\,b\,c)\,a\,b\,(b\,c\,a)\,b\,c\ .$$

The concatenation of two marked words is defined in the obvious way. For example, (1) can be obtained by concatenating $\{abca, \{(1, abc), (2, bca)\}\}$ and $\{bbcabc, \{(2, bca)\}\}$, i.e.,

$$(a\,b\,c)\,a \quad \text{and} \quad b\,(b\,c\,a)\,b\,c\ .$$

A marked word on α is called a *cluster* on α if it is not a concatenation of two nonempty marked words. So, (1) is not a cluster, but

$$b\,c\,(a\,b\,c)\,a$$

is a cluster. Two additional examples of clusters, using $A = \{a\}$ and $B = \{aaaa\}$, are

$$(a\,a\,a\,a)\,a\,a$$

and

$$(a\,a\,a\,a\,a\,a)\,,$$

which we include to emphasize the fact that a cluster is not required to be "maximal" in the sense that every possible marked subword must be included. If a word α has only one possible cluster, then there is no need to indicate the positions of the marked subwords and we say by abuse of language that the only cluster on α is itself.

Please cite this article in press as: Y. Zhuang, A generalized Goulden–Jackson cluster method and lattice path enumeration, Discrete Mathematics (2017), http://dx.doi.org/10.1016/j.disc.2017.09.004.

Download English Version:

https://daneshyari.com/en/article/8903111

Download Persian Version:

https://daneshyari.com/article/8903111

Daneshyari.com