

m-compositions and *m*-partitions: exhaustive generation and Gray code

ELISABETTA GRAZZINI

Dipartimento di Sistemi e Informatica,
Viale G. B. Morgagni 65, 50134 Firenze, Italy
e-mail: grazzini@dsi.unifi.it

and

EMANUELE MUNARINI

Dipartimento di Matematica, Politecnico di Milano
Piazza Leonardo da Vinci 32, 20133 Milano, Italy
e-mail: munarini@mate.polimi.it

and

MADDALENA PONETI

Dipartimento di Sistemi e Informatica,
Viale G. B. Morgagni 65, 50134 Firenze, Italy
Phone: +39 055 4237454; Fax: +39 055 4237436
e-mail: poneti@dsi.unifi.it

and

SIMONE RINALDI

Dipartimento di Scienze Matematiche e Informatiche R. Magari,
Pian dei Mantellini 44, 53100 Siena, Italy
e-mail: rinaldi@unisi.it

(Received: October 31, 2006)

Abstract. In this paper we give an exhaustive generation algorithm for the class of *m*-compositions of integers with fixed *m*. Moreover we define a Gray code to list these combinatorial objects according to a particular order. Finally, we define *m*-partitions, which are an *m*-dimensional generalization of integer partitions, and an exhaustive generation algorithm for *m*-partitions, based on the same ideas used for *m*-compositions, is given.

Mathematics Subject Classifications (2000). 68R05

1 Introduction

A *composition* of a natural number n is any k -tuple (x_1, \dots, x_k) of positive integers such that $x_1 + \dots + x_k = n$. In [8] the authors generalized the notion of compositions introducing the class of *2-compositions*, in order to enumerate L -convex polyominoes. Then in [14] the authors extended this definition to the m -dimensional case, and determined combinatorial properties and enumerative results on the so-called *m-compositions*.

Let m be a positive integer. An m -composition of an integer $n > 0$ is an $m \times k$ matrix M with nonnegative integer entries,

$$M = \begin{bmatrix} x_{11} & \cdots & x_{1k} \\ \vdots & & \vdots \\ x_{m1} & \cdots & x_{mk} \end{bmatrix}$$

with columns different from the zero vector, and such that the sum of all its entries is equal to n . The *length* of M is the number k of columns.

We point out that the given definition of m -composition is similar to the *compositions in form of vectors* defined by P. A. MacMahon [1, p.57] and studied in [2, 3, 4].

The aim of this paper is to present a simple algorithm for the generation of all of the m -compositions of a given integer n . Then we show that this algorithm can be applied also for the generation of the class of the m -partitions, which are a particular kind of compositions extending the classical notion of integer partitions.

If the generated list of objects is such that two consecutive elements differ only by a constant way, then the list is a *Gray code*. The classic example is the *binary reflected Gray code* [10], which is a scheme for listing all n -bit binary numbers so that successive numbers differ by exactly one bit. The first Gray code for (ordinary) compositions is that of Knuth-Klingsberg [12] and it was refined later and looplessly implemented by Walsh in [18]. The term *combinatorial Gray code* was introduced in [11] to refer to any method for generating combinatorial objects so that successive objects differ by some pre-specified, small way. The area of combinatorial Gray codes was popularized by Herbert Wilf [20]; a survey of Gray codes is given by C. Savage in [17].

The advantage of such an approach is that generation of successive objects might be faster. Although for many combinatorial families a straightforward lexicographic listing algorithm requires only constant average time per element, for other families, like linear extensions of a poset [9, 15, 16] such performance has only been achieved by a Gray code approach.

2 m -compositions and regular languages

As a simple extension of the coding used in [7] for the ordinary compositions, in [14] the authors showed that an m -composition can be represented in terms of a word on the alphabet $\mathcal{A}_m = \{a_1, \dots, a_m, b_1, \dots, b_m\}$. Let M be an m -composition of length k , let C_1, \dots, C_k be its vector columns, and let $\ell(M)$ be the word obtained as the concatenation of $\ell(C_1) \dots \ell(C_k)$, where:

$$\begin{aligned} \ell(C_1) &= a_1^{x_{11}} \dots a_m^{x_{m1}}, \\ \ell(C_i) &= b_j a_j^{x_{ji}-1} a_{j+1}^{x_{(j+1)i}} \dots a_m^{x_{mi}}, \quad i > 1, \end{aligned}$$

where $j = \min\{s \geq 1 : x_{si} > 0\}$ and, as usual, the power notation stands for concatenation of symbols. See the Example 1 below. M can be retrieved

from $\ell(M)$ uniquely: the first column of M is obtained from the longest prefix of a 's in $\ell(M)$ and each b marks the beginning of a new column. Let \mathcal{L}^m be the language of the words associated with the m -compositions, and let \mathcal{L}_n^m be the set of words in \mathcal{L}^m having length equal to n . The words in \mathcal{L}_n^m satisfy the following conditions:

1. they begin with a symbol in $\{a_1, \dots, a_m\}$;
2. every symbol a_i or b_i can be followed by an arbitrary b_j , and it can be followed by a symbol a_j only if $i \leq j$.

The two properties suggest that the words in \mathcal{L}_n^m have a unique factorization of the form xy where:

1. x is a nonempty word $a_1^{i_1} \dots a_m^{i_m}$, with $i_1, \dots, i_m \geq 0$;
2. y is a (possibly empty) word $y = y_1 \dots y_k$, with $y_r = b_j a_j^{q_j} \dots a_m^{q_m}$, with $q_j, \dots, q_m \geq 0$.

EXAMPLE 1 Consider the following 3-compositions of 12

$$M = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 3 & 0 & 1 & 2 \end{bmatrix} = 0 + 2 + 1 + 0.$$

According to the described procedure $\ell(M) = a_1 a_3 a_3 a_3 \ b_1 a_1 a_2 a_2 \ b_2 a_3 \ b_3 a_3$.

Following the above characterization we easily obtain that \mathcal{L}^m is a regular language defined by the unambiguous regular expression:

$$(a_1 a_1^* a_2^* \dots a_m^* + a_2 a_2^* a_3^* \dots a_m^* + \dots + a_m a_m^*) \times (b_1 a_1^* a_2^* \dots a_m^* + b_2 a_2^* \dots a_m^* + \dots + b_m a_m^*)^*$$

3 Generation of m -compositions

In this section we present a simple algorithm for the generation of m -compositions. We start by introducing the *index representation* of an m -composition. We consider the alphabets $\Sigma^1 = \{1, 2, \dots, m\}$, $\Sigma^2 = \{\bar{1}, \bar{2} \dots, \bar{m}\}$, and $\Sigma = \Sigma^1 \cup \Sigma^2$. Every word $\ell(M)$ associated with an m -composition M can be conveniently represented by means of a word $\tilde{\ell}(M) \in \Sigma^*$, obtained by substituting, in $\ell(M)$, every occurrence of a symbol a_i with $i \in \Sigma^1$, and every occurrence of b_j with $\bar{j} \in \Sigma^2$. For instance, the 3-composition of 12 of Example 1 can be represented by means of the word $\tilde{\ell}(M) = 1333\bar{1}22\bar{2}\bar{3}\bar{3}\bar{3}$.

We denote by $\tilde{\mathcal{L}}_n^m$ the set of the index representations of all the m -compositions of n and with M an m -composition of n . In order to generate the m -compositions of $n + 1$ we may apply one of the following operations to M :

- (i) increase by 1 any entry of M ;

- (ii) add a column with only an entry equal to 1 in the k -th position ($1 \leq k \leq m$) to the left of a column of M ;
- (iii) add a column with only an entry equal to 1 in the k -th position ($1 \leq k \leq m$) to the right of the last column of M .

The previous operations can easily be translated into operations on a word $w = w_1 w_2 \dots w_n$, with $w_i \in \Sigma$, representing an m -composition of n . Operations (i) and (ii) add a symbol, say ν , between two symbols of w or to the left of w_1 ; in these cases we obtain a word

$$\alpha_1 = w_1 w_2 \dots w_i \nu w_{i+1} \dots w_{n-1} w_n$$

or

$$\alpha_2 = \nu w_1 w_2 \dots w_{n-1} w_n.$$

The words:

$$\beta_1 = w_1 w_2 \dots w_i \nu w_{i+1} \dots w_{n-1} \quad \text{and} \quad \beta_2 = \nu w_1 w_2 \dots w_{n-1}$$

are m -compositions of n . Moreover, since if $w_n \in \Sigma^1$ then $w_n \geq w_{n-1}$, we have that $\alpha_1 = \beta_1 w_n$ and $\alpha_2 = \beta_2 w_n$ represent m -compositions of $n + 1$. The words obtained applying operation (iii) have the form $\alpha_3 = w_1 w_2 \dots w_{n-1} w_n \bar{\nu}$ with $\bar{\nu} \in \Sigma^2$, $1 \leq \bar{\nu} \leq m$; since $\beta_3 = w_1 w_2 \dots w_n$ belongs to $\tilde{\mathcal{L}}_n^m$, then α_3 belongs to $\tilde{\mathcal{L}}_{n+1}^m$.

Therefore, from every word $w = w_1 \dots w_n$ representing an m -composition of n we can obtain an m -composition of $n + 1$ simply concatenating to w the symbol $w_{n+1} \in \Sigma^1$ such that $w_{n+1} \geq w_n$ or a symbol $\bar{w}_{n+1} \in \Sigma^2$.

So, for any $n > 0$, the following recursive definition gives a way for generate the set $\tilde{\mathcal{L}}_n^m$:

$$\begin{aligned} \tilde{\mathcal{L}}_1^m &= \{1, 2, \dots, m\} \\ \tilde{\mathcal{L}}_n^m &= \left\{ w i \mid w = w_1 w_2 \dots w_{n-1} \in \tilde{\mathcal{L}}_{n-1}^m, i \in \Sigma^1, i \geq w_{n-1} \right\} \\ &\quad \cup \left\{ w i \mid w = w_1 w_2 \dots w_{n-1} \in \tilde{\mathcal{L}}_{n-1}^m, i \in \Sigma^2 \right\}. \end{aligned} \quad (1)$$

For $m = 2$ the first sets generated according to the above definition are:

$$\begin{aligned} \tilde{\mathcal{L}}_1^2 &= \{1, 2\} \\ \tilde{\mathcal{L}}_2^2 &= \{11, 12, 22, 1\bar{1}, 1\bar{2}, 2\bar{1}, 2\bar{2}\} \\ \tilde{\mathcal{L}}_3^2 &= \{111, 112, 122, 222, 1\bar{1}1, 1\bar{1}2, 1\bar{2}2, 2\bar{1}1, 2\bar{1}2, 2\bar{2}2, 11\bar{1}, 11\bar{2}, 12\bar{1}, 12\bar{2}, 22\bar{1}, \\ &\quad 22\bar{2}, 1\bar{1}\bar{1}, 1\bar{1}\bar{2}, 1\bar{2}\bar{1}, 1\bar{2}\bar{2}, 2\bar{1}\bar{1}, 2\bar{1}\bar{2}, 2\bar{2}\bar{1}, 2\bar{2}\bar{2}\}. \end{aligned}$$

4 A Gray code for the m -compositions

Using the recursive construction for the words of $\tilde{\mathcal{L}}_n^m$ given in Section 3 it is possible to define a listing of elements of the class such that two consecutive elements differ by one digit, i.e., to define a Gray code for m -compositions.

From now on, for sake of simplicity, we represent the set $\tilde{\mathcal{L}}_n^m$ by $\tilde{\mathcal{L}}_n$ and an m -composition of n by means of its corresponding word $w = w_1 w_2 \dots w_{n-1} w_n \in \tilde{\mathcal{L}}_n$; we define *successors* of w_n the digits that can be concatenated to w to obtain an element of $\tilde{\mathcal{L}}_{n+1}$, according to the generation algorithm. So the successors of $w_n = i \in \Sigma^1 \cup \Sigma^2$ are:

$$\{i, i + 1, \dots, m, \bar{1}, \bar{2}, \dots, \bar{m}\}.$$

Since among the successors of any element i there are always both m and \bar{m} , we can consider two possible arrangements of this set, given by the two lists $s(i, m)$ and $s(i, \bar{m})$:

$$\begin{aligned} s(i, m) &= \langle m, m - 1, \dots, i + 1, i, \bar{1}, \bar{2}, \dots, \bar{m} \rangle \\ s(i, \bar{m}) &= \langle \bar{m}, \bar{m} - \bar{1}, \dots, \bar{2}, \bar{1}, i, i + 1, \dots, m \rangle. \end{aligned}$$

Thus the list of the successors of a word $w \in \tilde{\mathcal{L}}_n$ always ends with m or \bar{m} (alternatively).

4.1 Definition of the Gray code for the m -compositions

Our aim in this section is to define a recursive construction of a Gray code on the set of words $\tilde{\mathcal{L}}_n$. We remark that the construction of such a Gray code, denoted by \mathbb{L}_n , relies on the same arguments applied in [6], and it is based on the generation of lists of elements of $\tilde{\mathcal{L}}_n$ with increasing lengths.

Let us give the following definitions:

- \mathbb{L}_n is the list of words of $\tilde{\mathcal{L}}_n$ that we are going to define;
- l_i^k is the i -th element of \mathbb{L}_k ;
- $|\mathbb{L}_k|$ is the cardinality of \mathbb{L}_k ;
- if x is a sequence of symbols, \vec{x} indicates the rightmost symbol of x ;
- given a list L ,
 - $first(L)$ is the first element of the list L ;
 - $last(L)$ is the last element of the list L ;
 - $x \circ L$ is the list obtained by the concatenation of every element of L to x .

4.1.1 Construction of the list \mathbb{L}_n

The list \mathbb{L}_n of the words of $\tilde{\mathcal{L}}_n$ is obtained by the concatenation of $q = |\tilde{\mathcal{L}}_{n-1}|$ lists L_n^i defined as follows:

$$\begin{cases} L_n^1 = l_1^{n-1} \circ s(\overrightarrow{l_1^{n-1}}, m) \\ L_n^i = l_i^{n-1} \circ s(\overrightarrow{l_i^{n-1}}, \overrightarrow{last(L_n^{i-1})}) \quad 1 < i \leq q. \end{cases}$$

Thus we set

$$\begin{cases} \mathbb{L}_1 = \langle 1, 2, \dots, m \rangle \\ \mathbb{L}_n = \Theta_{i=1}^q L_n^i & n > 1, \end{cases}$$

where, as usual, $\Theta_{i=1}^q L_n^i$ denotes the concatenation of the lists L_n^1, \dots, L_n^q .

EXAMPLE 2 *Let $m = 2$; we apply the previous definition to construct the list \mathbb{L}_n of the elements of $\tilde{\mathcal{L}}_n$, for $n = 1, 2, 3$. In this case a word always ends with an element $i \in \{1, 2\} \cup \{\bar{1}, \bar{2}\}$.*

We start by determining the sets $s(i, j)$ with $i, j \in \{1, 2\} \cup \{\bar{1}, \bar{2}\}$:

$$\begin{aligned} s(1, 2) &= s(\bar{1}, 2) = \langle 2, 1, \bar{1}, \bar{2} \rangle \\ s(1, \bar{2}) &= s(\bar{1}, \bar{2}) = \langle \bar{2}, \bar{1}, 1, 2 \rangle \\ s(2, 2) &= s(\bar{2}, 2) = \langle 2, \bar{1}, \bar{2} \rangle \\ s(2, \bar{2}) &= s(\bar{2}, \bar{2}) = \langle \bar{2}, \bar{1}, 2 \rangle. \end{aligned}$$

The list of elements of length 1 is simply $\mathbb{L}_1 = \langle 1, 2 \rangle$, so we can generate the lists L_2^i :

$$\begin{aligned} L_2^1 &= 1 \circ s(1, 2) = \langle 12, 11, 1\bar{1}, 1\bar{2} \rangle \\ L_2^2 &= 2 \circ s(2, \bar{2}) = \langle 2\bar{2}, 2\bar{1}, 22 \rangle, \end{aligned}$$

hence \mathbb{L}_2 is given by the concatenation of L_2^1 and L_2^2 ,

$$\mathbb{L}_2 = \langle 12, 11, 1\bar{1}, 1\bar{2}, 2\bar{2}, 2\bar{1}, 22 \rangle.$$

Therefore, for $n = 3$ we have:

$$\begin{aligned} L_3^1 &= 12 \circ s(2, 2) = \langle 122, 12\bar{1}, 12\bar{2} \rangle, \\ L_3^2 &= 11 \circ s(1, \bar{2}) = \langle 11\bar{2}, 11\bar{1}, 111, 112 \rangle, \\ L_3^3 &= 1\bar{1} \circ s(\bar{1}, 2) = \langle 1\bar{1}\bar{2}, 1\bar{1}\bar{1}, 1\bar{1}\bar{1}, 1\bar{1}\bar{2} \rangle, \\ L_3^4 &= 1\bar{2} \circ s(\bar{2}, \bar{2}) = \langle 1\bar{2}\bar{2}, 1\bar{2}\bar{1}, 1\bar{2}2 \rangle, \\ L_3^5 &= 2\bar{2} \circ s(\bar{2}, 2) = \langle 2\bar{2}\bar{2}, 2\bar{2}\bar{1}, 2\bar{2}2 \rangle, \\ L_3^6 &= 2\bar{1} \circ s(\bar{1}, \bar{2}) = \langle 2\bar{1}\bar{2}, 2\bar{1}\bar{1}, 2\bar{1}1, 2\bar{1}2 \rangle, \\ L_3^7 &= 22 \circ s(2, 2) = \langle 222, 22\bar{1}, 22\bar{2} \rangle. \end{aligned}$$

$$\begin{aligned} \mathbb{L}_3 &= \langle 122, 12\bar{1}, 12\bar{2}, 11\bar{2}, 11\bar{1}, 111, 112, 1\bar{1}\bar{2}, 1\bar{1}\bar{1}, 1\bar{1}\bar{1}, 1\bar{1}\bar{2}, 1\bar{2}\bar{2}, \\ &\quad 1\bar{2}\bar{1}, 1\bar{2}2, 2\bar{2}\bar{2}, 2\bar{2}\bar{1}, 2\bar{2}2, 2\bar{1}\bar{2}, 2\bar{1}\bar{1}, 2\bar{1}1, 2\bar{1}2, 222, 22\bar{1}, 22\bar{2} \rangle. \end{aligned}$$

THEOREM 1 *Two consecutive words of the list \mathbb{L}_n differ by one symbol.*

Proof. By induction on n .

BASE. For $n = 1$, $\tilde{\mathcal{L}}_1$ is formed, by the definition, of words with only a symbol, so the theorem is obviously true.

STEP $n \rightarrow n + 1$. We assume that any two consecutive elements of \mathbb{L}_{n-1} only differ by an element and prove that it is true also for \mathbb{L}_n . Since, by construction, the elements of the lists L_n^i differ by one symbol, it is enough to prove that $last(L_n^i)$ and $first(L_n^{i+1})$, with $1 \leq i \leq |\tilde{\mathcal{L}}_{n-1}| - 1$, differ only by one symbol.

Let J be the last element of $s(\overrightarrow{l_i^{n-1}}, \overrightarrow{last(L_n^{i-1})})$; then we have:

$$last(L_n^i) = l_i^{n-1} J$$

and

$$L_n^{i+1} = l_{i+1}^{n-1} \circ s(\overrightarrow{l_{i+1}^{n-1}}, \overrightarrow{last(L_n^i)}) = l_{i+1}^{n-1} \circ s(\overrightarrow{l_{i+1}^{n-1}}, J).$$

The first element of a list $s(i, k)$ is always k ; then:

$$first(L_n^{i+1}) = l_{i+1}^{n-1} J.$$

Then $last(L_n^i)$ and $first(L_n^{i+1})$ differ only by a symbol since this statement holds for l_i^{n-1} and l_{i+1}^{n-1} by the inductive hypothesis. \square

We may achieve a loop free implementation of the generating algorithm based on the following ideas:

- the first word of $\tilde{\mathcal{L}}_n$ ($n \geq m$) is $w = 1m \dots m$, where $w_1 = 1$ and $w_i = m$, $i = 2, \dots, n$, because the construction begins with the word $w = 1$ in $\tilde{\mathcal{L}}_1$; so the first word in $\tilde{\mathcal{L}}_2$ is the main word concatenated with the first digit of the set of the successors of $s(1, m)$ that is m and so on;
- the digit w_i to be modified at each step can be easily determined using the algorithm of Walsh for generating Gray codes in $O(1)$ worst-case time per word [19];
- w_i is modified taking into account the lists $s(i, m)$ and $s(i, \overline{m})$.

Clearly, the use of successors lists $s(i, m)$ and $s(i, \overline{m})$ does not alter the complexity of Walsh's procedure, so his efficient algorithm remains the kernel of his implementation.

5 Generation of m -partitions

A remarkable class of m -compositions is that of m -partitions, introduced in [14], and successively studied in [13]. To deal with this class, it is better to use an alternative graphical representation of m -compositions, which uses the definition of *labelled bargraphs*.

A *bargraph* is a column-convex polyomino, such that the lower edge lies on the horizontal axis. It is uniquely defined by the heights of its columns. A *labelled bargraph* is a bargraph whose cells are all labelled with positive integer numbers, and such that, in each column, the label of a cell is less than or equal

to the label of the cell immediately above (if any). The *degree* is the maximal label of the bargraph.

In [14] the authors show that every m -composition of an integer n can be represented as a labelled bargraph of degree $j \leq m$ having n cells, as follows. Let M be an m -composition of $n \geq 1$, having length k , and let $C_j = (a_{1j}, \dots, a_{mj})$ be the j -th column of M . We build a bargraph with k columns, of degree m at most, where the j -th column has exactly $a_{1j} + \dots + a_{mj}$ cells, and a_{ij} is the number of cells with label i in the j -th column, which are placed, according to the definition of labelled bargraph, just above the cells with label $i - 1$ (if any).

For instance, the following 3-composition of 9:

$$M = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 1 & 1 \\ 0 & 3 & 0 \end{bmatrix}$$

where the degree is 3 and

$$\begin{aligned} C_1 &= (1, 2, 0) & a_{11} &= 1 & a_{21} &= 2 & a_{31} &= 0 \\ C_2 &= (0, 1, 3) & a_{12} &= 0 & a_{22} &= 1 & a_{32} &= 3 \\ C_3 &= (1, 1, 0) & a_{13} &= 1 & a_{23} &= 1 & a_{33} &= 0 \end{aligned}$$

can be represented by means of the labelled bargraph in Figure 1.

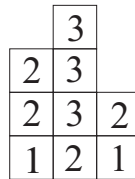


Figure 1: Bargraph associated with 3-composition of 9.

We remark that an m -composition corresponds to a bargraph with columns of variable height. Among labelled bargraphs we may consider those such that each column has height greater than or equal to the height of the column on its right (if any); see, for example, Figure 2. Those objects correspond to m -compositions such that the sum of entries in each column is greater than or equal to the sum of entries in the column on its right. More precisely, if s_i is the sum of the elements in the i -th column, then the sequence $\{s_1, \dots, s_k\}$ is weakly decreasing, i.e.

$$s_1 \geq s_2 \geq \dots \geq s_k, \tag{2}$$

hence the name *m-partitions*.

EXAMPLE 3 *Let us consider the following 3-partition of 11*

$$M = \begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \\ 1 & 0 & 2 & 0 \end{bmatrix}.$$

The associated bargraph is depicted in Figure 2. We have $(s_1 = 4) > (s_2 = 3) = (s_3 = 3) > (s_4 = 1)$.

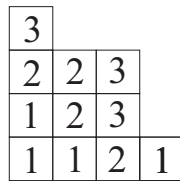


Figure 2: Bargraph associated with the 3-partition of 11 $w = 1123\bar{1}22\bar{2}33\bar{1}$.

The recursive method for generating the m -compositions explained in Section 3 can be suitably applied also to the generation of the m -partitions: it is sufficient to use an additional control on the procedure of generation, in order to check that every generated m -composition satisfies condition (2).

In a word $\tilde{\ell}(M) \in \tilde{\mathcal{L}}_n$ containing at least one symbol of Σ^2 , let d_1 be the number of symbols of Σ^1 placed between the last two elements of Σ^2 (or the number of symbols in Σ^1 placed between w_1 and the last symbol in Σ^2 , if there is an unique symbol of Σ^2), and let d_2 be the number of symbols in Σ^1 placed after the last symbol of Σ^2 in $\tilde{\ell}(M)$; for the relationship between the matrix and the bargraph representations of M , we have d_1 and d_2 cells above the base of the last two columns respectively. If $\tilde{\ell}(M) \in \mathcal{L}_n$ does not contain any symbol of Σ^2 , the corresponding bargraph has only one column whose height is n .

We consider a bargraph with n cells, representing an m -partition of n . In order to generate a bargraph with $(n + 1)$ cells, where the columns' heights are not increasing, we can perform the following two operations:

1. add a cell on the top of the rightmost column, provided that its height remains less than or equal to the one of previous column; the label l of the new cell must satisfy $t \leq l \leq m$, t being the label of the cell immediately below;
2. add a cell beside the rightmost column, with label $l \leq m$.

Therefore, the generation algorithm for the language of m -partitions is substantially the same as the algorithm for the generation of $\tilde{\mathcal{L}}_n$, with an additional control. We represent an m -partition by the word

$$w = w_1 \dots w_{h-j} \dots w_h \dots w_n,$$

where w_{h-j}, w_h are the last two elements of Σ^2 contained in w , and $w_{h+1}, \dots, w_n \in \Sigma^1$. If there is a unique symbol of Σ^2 , then

$$d_1 = h - 2, \quad d_2 = n - h$$

otherwise

$$d_1 = j - 1 = |w_{h-j+1} \dots w_{h-1}|, \quad d_2 = n - h = |w_{h+1} \dots w_n|.$$

Finally, when no symbol of Σ^2 is present in w , have that

$$d_1 = n - 1, \quad d_2 = 0.$$

In order to simplify the necessary tests on w we consider a “new word” $w' = d_1 d_2 w_1 \dots w_n$ with length $(n + 2)$. In this way, for each step of the generating algorithm, we have only to test, and then update, the first two components of w' . So:

- if $d_1 > d_2$ then
 - (i) $w = w_1 \dots w_n i$ with $i \in \Sigma^1$ and $i \geq w_n$, then set $d_2 = d_2 + 1$;
 - (ii) $w = w_1 \dots w_n i$ with $i \in \Sigma^2$, then set $d_1 = d_2$ and $d_2 = 0$;
- otherwise ($d_1 = d_2$), add a new column (operation 2.).

6 Concluding remarks

In this paper we deal with exhaustive generation algorithms for the m -compositions of integers. After introducing m -compositions, we showed that an m -composition M can be represented in terms of a word $\ell(M)$ on the alphabet $\mathcal{A}_m = \{a_1, \dots, a_m, b_1, \dots, b_m\}$. Then we associated to a word $\ell(M)$ a new word $\tilde{\ell}(M) \in \Sigma^1 \cup \Sigma^2$, obtained by substituting, in $\ell(M)$, every occurrence of a symbol a_i with $i \in \Sigma^1$, and every occurrence of b_j with $\bar{j} \in \Sigma^2$. A recursive algorithm for generating all the words $\tilde{\ell}(M)$ is then given.

Successively, we listed the m -compositions determining a Gray code on such a class.

Then, using substantially the same procedure, we gave an exhaustive generation algorithm for the m -partitions, which are a natural generalization of the ordinary integer partitions, and a special type of m -compositions. It is then reasonable to think that, in some future work, we will adapt the strategy used for m -compositions in order to obtain a Gray code on m -partitions as well.

References

- [1] G.E. ANDREWS, *The Theory of Partitions*, Encyclopedia of Mathematics and its Applications, Addison-Wesley Publishing Company, Reading, Massachusetts (1976).

-
- [2] G.E. ANDREWS, *The Theory of Compositions, II: Simon Newcom's problem*, *Utilitas Math.*, **7** (1975), 33–54.
 - [3] G.E. ANDREWS, *The Theory of Compositions, I: The ordered factorizations of n and a conjecture of C. Long*, *Canadian Math. Bull.*, **18** (1976), 479–484.
 - [4] G.E. ANDREWS, *The Theory of Compositions, III: The MacMahon formula and the Stanton-Cowan numbers*, *Utilitas Math.*, **9** (1976), 283–290.
 - [5] E. BARCUCCI, A. DEL LUNGO, E. PERGOLA and R. PINZANI, *ECO: a methodology for the enumeration of combinatorial objects*, *J. Difference Equ. Appl.*, **5** (1999), 435–490.
 - [6] A. BERNINI, E. GRAZZINI, E. PERGOLA and R. PINZANI, *A general exhaustive generation algorithm for Gray structures*, accepted on *Acta Informatica*.
 - [7] A. BJÖRNER and R. STANLEY, *An analogue of Young's lattice for compositions*, preprint (2005).
 - [8] G. CASTIGLIONE, A. FROSINI, E. MUNARINI, A. RESTIVO and S. RINALDI, *Enumeration of L -convex polyominoes II. Bijection and area*, *Proceedings di FPSAC'05, Taormina*, #49.
 - [9] E.R. CANFIELD and S.G. WILLIAMSON, *A loop-free algorithm for generating the linear extensions of a poset*, *Order Journal*, **12** (1994), 57–75.
 - [10] F. GRAY, *Pulse code communication*, U. S. Patent 2632058 (1953).
 - [11] M.T. JOICHI, D.E. WHITE and S.G. WILLIAMSON, *Combinatorial Gray codes*, *SIAM J. Comput.*, **9** (1980), 130–141.
 - [12] P. KLINGSBERG, *A Gray code for compositions*, *J. Algorithms*, **3** (1982), 41–44.
 - [13] E. MUNARINI, *Culnum enriched stacks*, preprint.
 - [14] E. MUNARINI, M. PONETI and S. RINALDI, *Matrix compositions*, accepted in FPSAC 2006.
 - [15] G. PRUESSE and F. RUSKEY, *Generating linear extensions fast*, *SIAM J. Comput.*, **23** (1994), 373–386.
 - [16] F. RUSKEY, *Generating linear extensions of posets by transpositions*, *J. Comb. Theory Ser. B*, **54** (1992), 77–101.
 - [17] C. SAVAGE, *A survey of combinatorial Gray codes*, *SIAM Review*, **39** (4) (1997), 605–629.
 - [18] T. WALSH, *Loop-free sequencing of bounded integer compositions*, *J. Combin. Math. Combin. Comput.*, **33** (2000), 323–345.
 - [19] T. WALSH, *Generating Gray codes in $O(1)$ worst-case time per word*, *LNCS 2731* (2003), 73–88.
 - [20] H.S. WILF, *Combinatorial algorithms: An update*, *CBNS, SIAM*, **55** (1989).