Vol. 13 (núm. 2) pp. 85-97
Julio-Diciembre 2022

Ciencia en Desarrollo
Revista de la Facultad de Ciencias
Vicerrectoría en Investigación y Extensión - VIE
Centro de Investigaciones y Extensión de la Facultad de Ciencias - CIEC

https://doi.org/10.19053/01217488.v13.n2.2022.13914

Sobre composiciones de enteros usando SageMath

On Integer Compositions Using SageMath

Jazmín L. Mantilla R.[1] y Wilson Olaya-León[2]

Resumen

SageMath (o simplemente SAGE, por las iniciales de: Software for Algebra and Geometry Experimentation) es un software matemático libre que integra, bajo el moderno entorno de programación de Python, todo el software matemático de código abierto existente, en una interfaz unificada que permite la implementación de una amplia gama de funcionalidades para dar soporte a la investigación y educación en las distintas áreas de las matemáticas.

En este artículo presentamos los códigos de los programas hechos en Sage para obtener los distintos tipos de restricciones y generalizaciones de las composiciones de números enteros más citadas en la literatura. Mostramos cómo dichos programas han servido para obtener conjeturas y nuevos resultados en temas actuales de investigación en esta área de las matemáticas.

***Palabras clave*:** Composiciones, Composiciones palíndromas, Composiciones de Carlitz, Composiciones coloreadas, composiciones superdiagonales.

Abstract

SageMath (or simply SAGE, by the initials of: Software for Algebra and Geometry Experimentation) is a free mathematical software that integrates, under the modern Python programming environment, all the best existing open source mathematical software in a unified interface that includes the implementation of a wide range of functionalities that allow to support research and education in the different areas of mathematics.

In this article we present the program codes made in Sage to obtain the different types of restrictions and generalizations of the compositions of integers most cited in the literature. We show how these programs have served to obtain conjectures and new results in current research topics in this area of mathematics.

***Keywords*:** Compositions, Palindromic compositions, Carlitz compositions, Colored compositions, Superdiagonal compositions.

Recepción: 28-ene-2022
Aceptación: 5-may-2022

[1]Matemática de la Universidad Industrial de Santander, Bucaramanga-Colombia. Dirección electrónica: jazlismaro_@hotmail.com
[2]Doctor en Matemáticas, Profesor de la Escuela de Matemáticas, Universidad Industrial de Santander, Bucaramanga-Colombia. Dirección electrónica: wolaya@uis.edu.co

1 Introducción

La composición de números enteros es un área de investigación en la combinatoria enumerativa y la teoría de números, con ella se busca establecer las formas en las que se puede representar un entero positivo como suma de enteros positivos menores o iguales a él. Aunque los orígenes de la teoría de composiciones se atribuyen a los trabajos realizados por Leonhard Euler en el siglo XVIII, la primera publicación sobre composiciones, [1] *Memoir on the Theory of Compositions of a Number*, fue hecha por Percy A. MacMahon en 1893. A finales de la década de 1960 aparecieron algunos artículos sobre composiciones con algunos tipos de restricciones. Pero sólo hasta este siglo se han hecho la mayoría de las publicaciones sobre composiciones, en las cuales se introducieron nuevos tipos de composiciones y métodos para determinar tales composiciones. En particular los diferentes tipos de composiciones se obtienen haciendo restricciones o generalizaciones al conjunto de las partes.

SageMath es un software matemático gratuito y de código abierto, soportado en el potente lenguaje de programación Python, para programar y realizar cálculos matemáticos complejos en distintos campos de las matemáticas a un nivel similar a los softwares de pago como Magma, Maple, Mathematica y Matlab. A través de la página web sagemath.org se puede:

- trabajar en la nube (portal COCALC),
- descargar la última versión para instalar en computador, o
- consultar su extensa documentación.

En términos generales, el software matemático ha contribuido a la investigación en matemáticas, al proporcionar una gran cantidad de datos para hacer conjeturas; y a la enseñanza de las matemáticas, al permitir la simulación para comprender un fenómeno y proyectar la solución de problemas. Además, la ventaja de los códigos de programas hechos sobre software libre radica en que todos pueden acceder al sistema y a su comprensión, ya que todas sus implementaciones son de código abierto, a diferencia del software de pago que está orientado a un selecto grupo de investigadores y educadores que lo pueden comprar y aun así sin acceso al funcionamiento de sus rutinas.

En este artículo presentamos los códigos de los programas que hicimos en *Sage* para generar los diferentes tipos de composiciones, con el ánimo de contribuir con la enseñanza e investigación de esta área de las matemáticas. Particularmente, mostramos que gracias a dichos códigos se obtuvieron algunos resultados sobre composiciones superdiagonales.

2 Composiciones

Una *composición* de un entero positivo n es una secuencia de enteros positivos $\sigma = [\sigma_1, \sigma_2, \ldots, \sigma_m]$ tales que $\sigma_1 + \sigma_2 + \cdots + \sigma_m = n$. Los σ_i son llamados las partes de la composición y m es el número de partes. Por conveniencia, se define la composición vacía como la única composición de cero.

Sage incluye el comando **Compositions(n)** que permite obtener la lista de las composiciones del entero n.

Recuerde que en *Sage* para obtener ayuda sobre un comando se escribe el nombre del comando seguido por el signo de interrogación "?". Asimísmo, para ver el código interno de un comando se escribe el nombre del comando seguido por el doble signo de interrogación "??".

Ejemplo 1. *Las composiciones de 5.*

```
sage: list(Compositions(5))
[[1, 1, 1, 1, 1], [1, 1, 1, 2], [1, 1, 2, 1], [1, 1, 3],
[1, 2, 1, 1], [1, 2, 2], [1, 3, 1], [1, 4], [2, 1, 1, 1],
[2, 1, 2], [2, 2, 1], [2, 3], [3, 1, 1], [3, 2], [4, 1], [5]]
```

Denotamos por $C(n)$ el total de composiciones del entero n. Observe en el Ejemplo 1 que $C(5) = 16$.

En *Sage* podemos utilizar el comando **len(Compositions(n))** para obtener $C(n)$.

Ejemplo 2. *La secuencia $C(n)$, para $0 \leq n < 20$.*

```
sage: [len(Compositions(n)) for n in range(20)]
[1, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048,
4096, 8192, 16384, 32768, 65536, 131072,]
262144]
```

Observe que los valores de $C(n)$ en la anterior secuencia son siempre una potencia de dos. Este resultado fue demostrado por MacMahon usando diagramas lineales.

Teorema 3 (MacMahon [1])**.** *El número total de composiciones de un entero positivo n es*

$$C(n) = 2^{n-1}.$$

En la literatura se conocen diversas restricciones sobre las composiciones de un entero. Por ejemplo, restringir el número de partes en la composición.

2.1 Composiciones con exactamente m partes

Las composiciones con exactamente m partes se obtienen al restringir a m el número total de partes en la composición.

En *Sage* podemos utilizar el comando **Compositions(n, length = m)** para la lista de las composiciones del entero n con m partes.

Ejemplo 4. *Las composiciones de 5 con 3 partes.*

```
sage: list(Compositions(5,length=3))
[[1, 1, 3], [1, 2, 2], [1, 3, 1], [2, 1, 2], [2, 2, 1],
[3, 1, 1]]
```

Denotamos por $C(n,m)$ el total de composiciones del entero n con exactamente m partes. Observe en el Ejemplo 4 que $C(5,3) = 6$.

El comando **len(Compositions(n, length = m))** en *Sage* nos permite calcular $C(n,m)$.

Ejemplo 5. *La secuencia $C(n,3)$ con $0 \leq n < 20$.*

```
sage: [len(Compositions(n, length = 3)) for
n in range(20)]
[0, 0, 0, 1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 66, 78,
91, 105, 120, 136, 153]
```

Con ayuda del código anterior podemos encontrar los primeros valores del arreglo $C(n,m)$, $n,m \geq 1$.

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 3 & 3 & 1 & 0 & 0 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 \\ 1 & 5 & 10 & 10 & 5 & 1 & 0 & 0 \\ 1 & 6 & 15 & 20 & 15 & 6 & 1 & 0 \\ 1 & 7 & 21 & 35 & 35 & 21 & 7 & 1 \end{pmatrix}$$

Observe que los valores en el arreglo anterior coinciden con el Triángulo de Pascal y con el Teorema 3 ya que la suma en cada fila es $C(n) = 2^{n-1}$.

Teorema 6 (MacMahon [1])**.** *El número total de composiciones de n con exactamente m partes es*

$$C(n,m) = \binom{n-1}{m-1}.$$

2.2 Composiciones con partes en $\{1,2,\ldots,k\}$

Las composiciones del entero n con partes en el conjunto $\{1,2,\ldots,k\}$ son las composiciones de n cuyas partes son únicamente algunos de los enteros $1,2,\ldots,k$.

El comando **Compositions(n,max_part=k)** en *Sage* nos permite obtener la lista de las composiciones de n con partes en $\{1,2,\ldots,k\}$.

Ejemplo 7. *Las composiciones de 5 con partes en* $\{1,2,3\}$.

```
sage: list(Compositions(5,max_part=3))
[[3, 2], [3, 1, 1], [2, 3], [2, 2, 1], [2, 1, 2], [2, 1, 1, 1],
[1, 3, 1], [1, 2, 2], [1, 2, 1, 1], [1, 1, 3], [1, 1, 2, 1],
[1, 1, 1, 2], [1, 1, 1, 1, 1]]
```

Denotamos por $C_{[k]}(n)$ el total de composiciones de n con partes en el conjunto $\{1,2,\ldots,k\}$. Observe en el Ejemplo 7 que $C_{\{1,2,3\}}(5) = 13$.

El comando **len(Compositions(n,max_part=k))** en *Sage* nos permite obtener el valor de $C_{[k]}(n)$.

Ejemplo 8. *La secuencia* $C_{[3]}(n)$ *con* $0 \leq n < 20$.

```
sage: [len(Compositions(n, max_part = 3))
for n in range(20)]
[1,1,2,4,7,13,24,44,81,149,274,504,927,
1705,3136,5768,10609,19513,35890,66012]
```

En 1975, Hoggatt y Bicknell [2], mostraron que los valores de la secuencia $C_{[k]}(n)$ tienen una correspondencia con los números k-generalizados de Fibonacci (o números k-bonacci).

Los números k-bonacci se definen recursivamente como

$$\mathscr{F}_n^{(k)} = \mathscr{F}_{n-1}^{(k)} + \mathscr{F}_{n-2}^{(k)} + \cdots + \mathscr{F}_{n-k}^{(k)}, \text{ para } n \geq k,$$

con valores iniciales,

$$\mathscr{F}_0^{(k)} = \mathscr{F}_1^{(k)} = \cdots = \mathscr{F}_{k-2}^{(k)} = 0 \text{ y } \mathscr{F}_{k-1}^{(k)} = 1.$$

Por ejemplo, la secuencia de números Tribonacci es (OEIS-A000073)*:

$$0,0,1,1,2,4,7,13,24,44,81,149,274,\ldots$$

Observe en el Ejemplo 8 que los valores de la secuencia $C_{[3]}(n)$ corresponden con los números Tribonacci con $n \geq 2$. Esto es cierto en general.

Teorema 9 (Hoggatt y Bicknell [2])**.** *El número total de composiciones de n con partes en el conjunto* $\{1,2,\ldots,k\}$ *es*

$$C_{[k]}(n) = \mathscr{F}_{n+k-1}^{(k)}.$$

Corolario 10. *El número total de composiciones de n con partes en el conjunto* $\{1,2\}$ *es*

$$C_{[2]}(n) = F_{n+1}$$

donde F_{n+1} *es el número de Fibonacci* $n+1$.

3 Composiciones palíndromas

Las composiciones *palíndromas* de un entero n son aquellas composiciones de n en las que las partes se leen de la misma forma de izquierda a derecha que de derecha a izquierda. Denotaremos por $P(n)$ el total de composiciones palíndromas de n.

La tabla 1 muestra el código del programa en *Sage* para obtener las composiciones palíndromas de un entero positivo n.

```
sage: def Palin(n):
....:     C = list(Compositions(n))
....:     k = len(Compositions(n))
....:     P = [ ]
....:     for i in range(k):
....:         r = Composition(C[i]).reversed()
....:         if r == C[i]:
....:             P.append(r)
....:     return (P)
```

Tabla 1. Código en *Sage* para las composiciones palíndromas.

Ejemplo 11. *Las composiciones palíndromas de 6.*

```
sage: Palin(6)
[[1,1,1,1,1,1],[1,1,2,1,1],[1,2,2,1],[1,4,1],
[2,1,1,2],[2,2,2],[3,3],[6]]
```

La secuencia $P(n)$ *con* $0 \leq n < 20$.

```
sage: [len(Palin(n)) for n in range(20)]
[1,1,2,2,4,4,8,8,16,16,32,32,64,64,128,
128,256,256,512,512,1024]
```

La propiedad observada en el Ejemplo 11 para la secuencia $P(n)$, fue demostrada por Hoggatt y Bicknell utilizando diagramas de barras.

Teorema 12 (Hoggatt y Bicknell [2])**.** *El número total de composiciones palíndromas de n es*

$$P(n) = 2^{\lfloor n/2 \rfloor}$$

donde $\lfloor \cdot \rfloor$ *es la usual función parte entera.*

3.1 Composiciones palíndromas con m partes

Denotamos por $P(n,m)$ el total de composiciones palíndromas con exactamente m partes.

La Tabla 2 muestra el código del programa en *Sage* para obtener las composiciones palíndromas de n con exactamente m partes.

*Código en On-Line Encyclopedia of Integer Sequences, ver *https:oeis.org*

```
sage: def Palin(n,m):
....:     C = list(Compositions(n, length = m))
....:     k = len(Compositions(n, length = m))
....:     P = [ ]
....:     for i in range(k):
....:         r = Composition(C[i]).reversed()
....:         if r == C[i]:
....:             P.append(r)
....:     return (P)
```

Tabla 2. Código en *Sage* para las composiciones palíndromas con m partes.

```
sage: def Palin[2](n):
....:     C = list(Compositions(n, max_part = 2))
....:     k = len(Compositions(n, max_part = 2))
....:     P = [ ]
....:     for i in range(k):
....:         r = Composition(C[i]).reversed()
....:         if r == C[i]:
....:             P.append(r)
....:     return (P)
```

Tabla 3. Código en *Sage* para las composiciones palíndromas con partes en $\{1,2\}$.

Ejemplo 13. *Las composiciones palíndromas de 21 con 3 partes.*

```
sage: Palin(21,3)
[[10, 1, 10], [9, 3, 9], [8, 5, 8], [7, 7, 7], [6, 9, 6],
[5, 11, 5], [4, 13, 4], [3, 15, 3], [2, 17, 2], [1, 19, 1]]
```

La secuencia $P(n,3)$ para $0 \leq n < 20$.

```
sage: [len(Palin(n,3)) for n in range(20)]
[0, 0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9,
10, 10, 11, 11, 12, 12, 13, 13, 14].
```

En [2] Hoggatt y Bicknell también mostraron que el total de composiciones palíndromas de n con m partes depende de la paridad del número de partes. Note que no hay composiciones palíndromas de un número impar con un número par de partes.

Teorema 14 (Hoggatt y Bicknell, [2])**.** *El total de composiciones palíndromas de n con m partes es*

- $P(2n,2k) = \binom{n-1}{k-1}$.
- $P(2n,2k+1) = P(2n-1,2k+1) = \binom{n-1}{k}$.

3.2 Composiciones palíndromas con partes en el conjunto $\{1,2\}$

Denotamos por $P_{[2]}(n)$ el total de composiciones palíndromas con partes en $\{1,2\}$.

La Tabla 3 muestra el código del programa en *Sage* para obtener las composiciones palíndromas con partes en $\{1,2\}$.

Ejemplo 15. *Las composiciones palíndromas de 6 con partes en $\{1,2\}$.*

```
sage: Palin[2](6)
[[2, 2, 2], [2, 1, 1, 2], [1, 2, 2, 1], [1, 1, 2, 1, 1],
[1, 1, 1, 1, 1, 1]]
```

La secuencia $P_{[2]}(n)$ con $0 \leq n < 20$.

```
sage: [len(Palin[2](n)) for n in range(20)]
[1, 1, 2, 1, 3, 2, 5, 3, 8, 5, 13, 8, 21, 13, 34, 21, 55,
34, 89, 55]
```

En [3] Alladi y Hoggatt mostraron que existe una correspondencia de las composiciones palíndromas con partes en $\{1,2\}$ y los números de Fibonacci.

Teorema 16 (Alladi y Hoggatt [3])**.** *El número total de composiciones palíndromas de un entero n con partes en el conjunto $\{1,2\}$ es*

- $P_{[2]}(2n+1) = F_{n+1}$
- $P_{[2]}(2n) = F_{n+2}$

donde F_{n+1} y F_{n+2} son los números de Fibonacci $n+1$ y $n+2$, respectivamente.

4 Composiciones de Carlitz

Las composiciones de Carlitz de un entero n son las composiciones de n sin partes adyacentes iguales. Denotamos por $CC(n)$ el total de composiciones de Carlitz de n.

La Tabla 4 muestra el código del programa en *Sage* para obtener las composiciones de Carlitz de n.

```
sage: def Carlitz(n):
....:     C = list(Compositions(n))
....:     k = len(Compositions(n))
....:     M = [ ]
....:     for i in range(k):
....:         p = len(C[i])
....:         W = [ ]
....:         J = [ ]
....:         for j in range(1,p):
....:             W.append(C[i][j])
....:         for j in range(0,p-1):
....:             if C[i][j] != W[j]:
....:                 J.append(C[i][j])
....:         if len(J) == p - 1:
....:             M.append(C[i])
....:     return (M)
```

Tabla 4. Código en *Sage* para las composiciones de Carlitz.

Ejemplo 17. *Las composiciones de Carlitz de 6.*

```
sage: Carlitz(6)
[[1,2,1,2],[1,2,3],[1,3,2],[1,4,1],[1,5],
[2,1,2,1],[2,1,3],[2,3,1],[2,4],[3,1,2],
[3,2,1],[4,2],[5,1],[6]]
```

La secuencia $CC(n)$ con $0 \leq n < 20$.

```
sage: [len(Carlitz(n)) for n in range(20)]
[1,1,1,3,4,7,14,23,39,71,124,214,378,661,
1152,2024,3542,6189,10843,18978]
```

En [4] Carlitz presentó una formula recursiva para $CC(n)$ en función de la diferencia del número de divisores pares e impares de los enteros no negativos menores o iguales a n.

Teorema 18 (Carlitz [4])**.** *El total de composiciones de Carlitz de n es*

$$CC(n) = \sum_{l=1}^{n} d'(l)CC(n-l),$$

donde $d'(n)$ denota la diferencia entre el número de divisores impares y pares de n.

4.1 Composiciones de Carlitz palíndromas

Denotamos por $CP(n)$ el total de composiciones de Carlitz palíndromas del entero n.

La Tabla 5 muestra el código del programa en *Sage* para obtener las composiciones de Carlitz palíndromas de n.

```
sage: def CarlitzP(n):
....:     C = Carlitz(n)
....:     p=len(C)
....:     M = [ ]
....:     for i in range(p):
....:         m=C[i].reversed()
....:         if m==C[i]:
....:             M.append(C[i])
....:     return(M)
```

Tabla 5. Código en *Sage* para las composiciones de Carlitz palíndromas.

Ejemplo 19. *Las composiciones de Carlitz palíndromas de 7.*

```
sage: CarlitzP(7)
[[1,2,1,2,1],[1,5,1],[2,3,2],[3,1,3],[7]]
```

La secuencia $CP(n)$, para $0 \leq n < 20$.

```
sage: [len(CarlitzP(n)) for n in range(20)]
[0,1,1,1,2,3,2,5,5,7,10,14,14,25,26,42,48,
75,79,132]
```

La secuencia de $CP(n)$ es (OEIS-A239327):

$$1,1,1,1,2,3,2,5,5,7,10,14,14,25,26,42,48,\ldots$$

El siguiente teorema, demostrado por Carlitz en [4], establece la función generatriz $CP(x) = \sum_{n\geq 0} CP(n)x^n$. Que puede utilizarse para determinar los valores de la secuencia $CP(n)$.

Teorema 20 (Carlitz [4])**.** *La función generatriz para el número de composiciones de Carlitz palíndroma es*

$$CP(x) = 1 + \frac{\sum_{k\geq 1} \frac{x^k}{1+x^{2k}}}{1 - \sum_{k\geq 1} \frac{x^{2k}}{1+x^{2k}}}.$$

5 Composiciones coloreadas

Las siguientes composiciones, a diferencia de las anteriores, representan una generalización de las composiciones clásicas al permitir que cada parte de la composición se pueda colorear.

Las composiciones coloreadas de un entero n son son las composición de n en la que cada parte de tamaño i se puede colorear en i diferentes colores. Usualmente se utilizan subíndices $i_1, i_2, \ldots, i_i$ para indicar las coloraciones de una parte de tamaño i.

Denotaremos por $CO(n)$ el número total de composiciones coloreadas de n.

La tabla 6 muestra el código del programa en *Sage* para obtener las composiciones coloreadas de n. Este programa imprime las coloraciones de cada parte de tamaño i como $'i\_1'$, $'i\_2'$, ... , $'i\_i'$.

Ejemplo 21. *Las composiciones coloreadas de 4.*

```
sage: Color(4)
[['1_1','1_1','1_1','1_1'],['1_1','1_1','2_1'],
['1_1','1_1','2_2'],['1_1','2_1','1_1'],
['1_1','2_2','1_1'],['1_1','3_1'],['1_1','3_2'],
['1_1','3_3'],['2_1','1_1','1_1'],['2_1','2_1'],
['2_1','2_2'],['2_2','1_1','1_1'],['2_2','2_1'],
['2_2','2_2'],['3_1','1_1'],['3_2','1_1'],
['3_3','1_1'],['4_1'],['4_2'],['4_3'],['4_4']]
```

La secuencia $CO(n)$ con $0 \leq n < 20$.

```
sage: [len(Color(n)) for n in range(20)]
[1, 1, 3, 8, 21, 55, 144, 377, 987, 2584, 6765,
17711, 46368, 121393, 317811, 832040,
2178309, 5702887, 14930352, 39088169]
```

```
sage: def Color(n):
....:     C=list(Compositions(n))
....:     k=len(Compositions(n))
....:     W=[]
....:     for i in range(k):
....:         M=[]
....:         p=len(C[i])
....:         for j in range(p):
....:             J=[]
....:             for k in range(1,C[i][j]+1):
....:                 J.append(str(C[i][j])
                                  +'_'+str(k))
....:             M.append(J)
....:         W.extend(list(cartesian
                              _product(M)))
....:     t=len(W)
....:     N=[]
....:     for i in range(t):
....:         m=len(W[i])
....:         L=[]
....:         for j in range(m):
....:             L.append(W[i][j])
....:         N.append(L)
....:     return(N)
```

Tabla 6. Código en *Sage* para las composiciones coloreadas.

En [5] Agarwal introduce y demuestra que el total de composiciones de n en términos de los números pares de Fibonacci.

Teorema 22 (Agarwal [5])**.** *El número total de composiciones coloreadas de n es*

$$CO(n) = F_{2n},$$

donde F_{2n} es el número de Fibonacci 2*n*.

5.1 Composiciones coloreadas con m partes

Denotaremos por $CO(n,m)$ para indicar el total de composiciones coloreadas de un entero n con exactamente m partes.

La tabla 7 muestra el código del programa en *Sage* para obtener las composiciones coloreadas con exactamente m partes.

Ejemplo 23. *Las composiciones coloreadas de 4 con 3 partes.*

```
sage: def Color(n,m):
....:     C=list(Compositions(n,length = m))
....:     k=len(Compositions(n,length = m))
....:     W=[]
....:     for i in range(k):
....:         M=[]
....:         p=len(C[i])
....:         for j in range(p):
....:             J=[]
....:             for k in range(1,C[i][j]+1):
....:                 J.append(str(C[i][j])+'_'
                                        +str(k))
....:             M.append(J)
....:         W.extend(list(cartesian
                                    _product(M)))
....:     t=len(W);
....:     N=[]
....:     for i in range(t):
....:         m=len(W[i])
....:         L=[]
....:         for j in range(m):
....:             L.append(W[i][j])
....:         N.append(L)
....:     return(N)
```

Tabla 7. Código en *Sage* para las composiciones coloreadas de n con m partes.

```
sage: Color(4,3)
[['2_1','1_1','1_1'],['2_2','1_1','1_1'],
['1_1','2_1','1_1'],['1_1','2_2','1_1'],
['1_1','1_1','2_1'],['1_1','1_1','2_2']]
```

La secuencia $CO(n,3)$ *con* $0 \leq n < 20$.

```
sage: [len(Color(n)) for n in range(20)]
[0,0,0,1,6,21,56,126,252,462,792,1287,
2002,3003,4368,6188,8568,11628,15504,
20349]
```

Con ayuda del código del programa de la tabla 7 podemos encontrar los primeros valores del arreglo $CO(n,m)$, $n,m \geq 1$.

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 4 & 10 & 6 & 1 & 0 & 0 & 0 & 0 \\ 5 & 20 & 21 & 8 & 1 & 0 & 0 & 0 \\ 6 & 35 & 56 & 36 & 10 & 1 & 0 & 0 \\ 7 & 56 & 126 & 120 & 55 & 12 & 1 & 0 \\ 8 & 84 & 252 & 330 & 220 & 78 & 14 & 1 \end{pmatrix}$$

En [5] Agarwal también presenta y demuestra la formula para obtener $CO(n,m)$.

Teorema 24 (Agarwal [5])**.** *El número total de las composiciones coloreadas de n con m partes es*

$$CO(n,m) = \binom{n+m-1}{2m-1}.$$

5.2 Composiciones coloreadas palíndromas

Denotaremos por $COP(n)$ para indicar el total de composiciones coloreadas palíndromas de n.

La tabla 8 muestra el código del programa en *Sage* para obtener las composiciones coloreadas palíndromas de un entero positivo n.

```
sage: def ColorP(n):
....:     C=Colored(n)
....:     k=len(C)
....:     W=[]
....:     for i in range(k):
....:         p=len(C[i])
....:         J=[]
....:         M=[]
....:         for j in range(p):
....:             t=p-1-j
....:             J.append(C[i][t])
....:             M.append(C[i][j])
....:         if M==J:
....:             W.append(J)
....:     return(W)
```

Tabla 8. Código en *Sage* para las composiciones coloreadas palíndromas.

Ejemplo 25. *Las composiciones coloreadas palíndromas de 4.*

```
sage: ColorP(4)
[['1_1','1_1','1_1','1_1'],['1_1','2_1','1_1'],
['1_1','2_2','1_1'],['2_1','2_1'],['2_2','2_2'],
['4_1'],['4_2'],['4_3'],['4_4']]
```

La secuencia $COP(n)$ con $0 \leq n < 20$.

```
sage: [len(ColorP(n)) for n in range(20)]
[1,1,3,4,9,11,24,29,63,76,165,199,432,
521,1131,1364,2961,3571,7752,]
```

En [5] Agarwal demuestra el siguiente teorema que establece la formula para obtener $COP(n)$ en términos de los números de Fibonacci y que depende de la paridad del entero positivo n.

Teorema 26 (Agarwal [5])**.** *El número total de las composiciones coloreadas palíndromas de un entero positivo n es*

$$COP(n) = \begin{cases} F_n + 2F_{n-1} & \text{si } n \text{ es impar} \\ 3F_n & \text{si } n \text{ es par} \end{cases}$$

donde F_n y F_{n-1} son los números de Fibonacci n y $n-1$, respectivamente.

Las siguientes composiciones fueron introducidas por Deutsch, Munarini y Rinaldi en [6].

6 Composiciones superdiagonales

Las composiciones superdiagonales del entero n son las composiciones de n en las que cada parte es mayor o igual que la posición de la parte. Es decir, son las composiciones $\sigma = [\sigma_1, \sigma_2, \ldots, \sigma_k]$ de n tales que $\sigma_l \geq l$ para todo $1 \leq l \leq k$.

Denotaremos por $S(n)$ para indicar el número total de composiciones superdiagonales de n.

La tabla 9 muestra el código del programa en *Sage* para obtener las composiciones superdiagonales.

Ejemplo 27. *Las composiciones superdiagonales de 7.*

```
sage: Sdiagonal(7)
[[1,2,4],[1,3,3],[1,6],[2,2,3],[2,5],[3,4],
[4,3],[5,2],[7]]
```

```
sage: def Sdiagonal(n):
....:     C = list(Compositions(n))
....:     k=len(C)
....:     M=[]
....:     for i in range(k):
....:         l = []
....:         p=len(C[i])
....:         for j in range(p):
....:             if j < C[i][j]:
....:                 l.append(C[i][j])
....:         if l == C[i]:
....:             M.append(C[i])
....:     return(M)
```

Tabla 9. Código en *Sage* para las composiciones superdiagonales.

La secuencia $S(n)$ con $0 \leq n < 20$.

```
sage: [len(Sdiagonal(n)) for n in range(20)]
[1,1,1,2,3,4,6,9,13,18,25,35,49,68,93,126,
170,229,308,413].
```

En [6] Deutsch, Munarini y Rinaldi dan la formula para obtener el número de composiciones superdiagonales de un entero positivo n.

Teorema 28 (Deutsch et al. [6])**.** *El número total de composiciones superdiagonales de n es*

$$S(n) = \sum_{k \geq 1} \binom{n - \binom{k}{2} - 1}{k-1}.$$

6.1 Composiciones superdiagonales de n con m partes

Denotaremos por $S(n,m)$ para indicar el total de composiciones superdiagonales de n con m partes.

La tabla 10 muestra el código del programa en *Sage* para obtener las composiciones superdiagonales de n con m partes.

Ejemplo 29. *Las composiciones superdiagonales de 9 con 3 partes.*

```
sage: Sdiagonal(9,3)
[[4,2,3],[3,3,3],[3,2,4],[2,4,3],[2,3,4],
[2,2,5],[1,5,3],[1,4,4],[1,3,5],[1,2,6]]
```

```
sage: def Sdiagonal(n,m):
....:     C = list(Compositions(n,length=m))
....:     k=len(C)
....:     M=[]
....:     for i in range(k):
....:         l = []
....:         p=len(C[i])
....:         for j in range(p):
....:             if j < C[i][j]:
....:                 l.append(C[i][j])
....:         if l == C[i]:
....:             M.append(C[i])
....:     return(M)
```

Tabla 10. Código en *Sage* para las composiciones superdiagonales con m partes.

La secuencia $S(n,2)$ *para* $0 \leq n < 20$.

```
sage: [len(Sdiagonal(n,2)) for n in range(20)]
[0,0,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,
15,16,17].
```

La secuencia $S(n,3)$ *para* $0 \leq n < 20$.

```
sage: [len(Sdiagonal(n,3)) for n in range(20)]
[0,0,0,0,0,0,1,3,6,10,15,21,28,36,45,55,
66,78,91,105].
```

En [6] se demuestra la formula para obtener el total de composiciones superdiagonales de n con exactamente m partes.

Teorema 30 (Deutsch et al. [6])**.** *El número total de composiciones superdiagonales de n con m partes es*

$$S(n,m) = \binom{n-\binom{m}{2}-1}{m-1}.$$

6.2 Composiciones superdiagonales palíndromas

Denotaremos por $SP(n)$ para indicar el total de composiciones superdiagonales palíndromas de n.

La tabla 11 muestra el código del programa en *Sage* para obtener las composiciones superdiagonales palíndromas.

```
sage: def SPalin(n):
....:     C = Superdiagonal(n)
....:     k = len(C)
....:     P = [ ]
....:     for i in range(0,k):
....:         r = C[i].reversed()
....:         if r == C[i]:
....:             P.append(r)
....:     return (P)
```

Tabla 11. Código en *Sage* para las composiciones superdiagonales palíndromas.

Ejemplo 31. *Las composiciones superdiagonales palíndromas de 15.*

```
sage: SPalin(15)
[[3,9,3],[4,7,4],[5,5,5],[6,3,6],[15]]
```

La secuencia $SP(n)$ *con* $0 \leq n < 20$.

```
sage: [len(SPalin(n)) for n in range(20)]
[1,1,1,1,2,1,2,1,3,2,4,3,5,4,7,5,9,6,11,7].
```

Denotaremos por $SP(n,m)$ para indicar el total de composiciones superdiagonales palíndromas de n con m partes.

La tabla 12 muestra el código del programa en *Sage* para obtener las composiciones superdiagonales palíndromas de n con m partes.

Ejemplo 32. *Las composiciones superdiagonales palíndromas de 17 con 3 partes.*

```
sage: SPalin(17,3)
[[7,3,7],[6,5,6],[5,7,5],[4,9,4],[3,11,3]]
```

La secuencia $SP(n,m)$ *con* $0 \leq n < 20$ *y* $m = 2,3$.

```
sage: [len(SPalin(n,2)) for n in range(20)]
[0,0,0,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0].
sage: [len(SPalin(n,3)) for n in range(20)]
[0,0,0,0,0,0,0,0,1,1,2,2,3,3,4,4,5,5,6,6].
```

La secuencia $SP(n,5)$ *con* $20 \leq n < 40$.

```
sage: [len(SPalin(n,5)) for n in range(20,40)]
[0, 1, 1, 3, 3, 6, 6, 10, 10, 15, 15, 21, 21, 28, 28, 36,
36, 45, 45, 55].
```

```
sage: def SPalin(n,m):
....:     C = Superdiagonal(n,m)
....:     k = len(C)
....:     P = [ ]
....:     for i in range(k):
....:         r = C[i].reversed()
....:         if r == C[i]:
....:             P.append(r)
....:     return (P)
```

Tabla 12. Código en *Sage* para las composiciones superdiagonales palíndromas con m partes.

Observe en el Ejemplo 32 que el número total de composiciones superdiagonales palíndromas de n con m partes depende de la paridad del número de partes. Además, note que no hay composiciones superdiagonales palíndromas de un número impar con un número par de partes.

Teorema 33 (M., O-L y Ramírez [7])**.**

1. *El total de las composiciones palíndromas de un entero par con un número par de partes es*

$$P(2n,2k) = \binom{n-\binom{k+1}{2}-2\binom{k}{2}-1}{k-1}.$$

2. *El total de las composiciones palíndromas de un entero par o impar con un número impar de partes es*
$SP(2n,2k-1) = SP(2n-1,2k-1) =$

$$= \binom{\lfloor \frac{n-3k^2}{2} \rfloor + 2k - 1}{k-1}.$$

6.3 Composiciones superdiagonales coloreadas

Denotamos por $SCO(n)$ para indicar el total de composiciones superdiagonales coloreadas de n.

La tabla 13 muestra el código del programa en *Sage* para obtener las composiciones superdiagonales coloreadas.

```
sage: def SuperColored(n):
....:     C=list(Superdiagonal(n))
....:     k=len(C)
....:     W=[]
....:     for i in range(k):
....:         M=[]
....:         p=len(C[i])
....:         for j in range(p):
....:             J=[]
....:             for k in range(1,C[i][j]+1):
....:             M.append(J)
....:         W.extend(list(cartesian_product(M)))
....:     t=len(W)
....:     N=[]
....:     for i in range(t):
....:         m=len(W[i])
....:         L=[]
....:         for j in range(m):
....:             L.append(W[i][j])
....:         N.append(L)
....:     return(N)
```

Tabla 13. Código en *Sage* para las composiciones superdiagonales coloreadas.

Ejemplo 34. *Las composiciones superdiagonales coloreadas de 4.*

```
sage: SuperColored(4)
[['1_1','3_1'],['1_1','3_2'],['1_1','3_3'],
['2_1','2_1'],['2_1','2_2'],['2_2','2_1'],
['2_2','2_2'],['4_1'],['4_2'],['4_3'],['4_4']]
```

La secuencia $SCO(n)$ con $0 \leq n < 20$.

```
sage: [len(SuperColored(n)) for n in range(20)]
[1, 1, 2, 5, 11, 21, 42, 86, 171, 322, 596, 1109,
2072, 3837, 6954, 12360, 21735, 38097, 66722,
116490].
```

Teorema 35 (M., O-L y Ramírez [7])**.** *El total de composiciones superdiagonales coloreadas de n es SCO(n)=*

$$= \sum_{m,l\geq 0} \binom{2m+l-1}{l} T\left(m, n-\binom{m+1}{2}-l\right),$$

donde

$$T(m,k)=\sum_{i=0}^{m}(-1)^{m+k+i}\binom{i}{m-k}\begin{bmatrix}m+1\\ m+1-i\end{bmatrix}$$

y $\begin{bmatrix}n\\ k\end{bmatrix}$ *son los números de Stirling de la primera clase.*

6.4 Composiciones superdiagonales de Carlitz

Denotaremos por $SC(n)$ para indicar el total de composiciones superdiagonales de Carlitz de n.

La tabla 14 muestra el código del programa en *Sage* para obtener las composiciones superdiagonales de Carlitz para un entero positivo n.

```
sage: def SuperdCarlitz(n):
....:     C = list(Carliz(n))
....:     k=len(C)
....:     M=[]
....:     for i in range(k):
....:         l = []
....:         p=len(C[i])
....:         for j in range(p):
....:             if j < C[i][j]:
....:                 l.append(C[i][j])
....:         if l == C[i]:
....:             M.append(C[i])
....:     return(M)
```

Tabla 14. Código en *Sage* para las composiciones superdiagonales de Carlitz.

Ejemplo 36. *Las composiciones superdiagonales de Carlitz de 7.*

```
sage: SuperdCarlitz(7)
[[1,2,4],[1,6],[2,5],[3,4],[4,3],[5,2],[7]]
```

La secuencia $SC(n)$ *con* $0 \leq n < 20$.

```
sage: [len(superdCarlitz(n)) for n in range(20)]
[0,1,1,2,2,4,5,7,10,15,20,26,38,50,70,92,125,164,222,291].
```

Problema abierto: encontrar una fórmula para el total de composiciones superdiagonales de Carlitz de un entero positivo n.

7 Conclusiones

Presentamos los códigos de los programas hechos en *SageMath* para generar las distintas restricciones y generalizaciones de la composiciones más citadas en la literatura. Esperamos que sirvan de soporte al estudio, investigación y enseñanza de la teoría de composiciones, rama de investigación activa en Combinatoria y Teoría de Números.

La ventaja de *SageMath* está en su gratuidad, a diferencia del software de pago que esta orientado al selecto grupo de investigadores y profesores de matemáticas que lo pueden comprar. Iniciativa que ayuda a contribuir a la apertura de la ciencia para beneficio de toda la sociedad.

Hemos incluido los códigos presentados en este artículo en el repositorio Noesis de la Universidad Industrial de Santander. https://noesis.uis.edu.co/handle/20.500.14071/4523

Agradecimientos

Los autores deseamos agradecer al profesor José Luis Ramírez de la Universidad Nacional de Colombia por las muchas reuniones y discusiones que motivaron este trabajo. También queremos agradecer a los jurados evaluadores por su cuidadosa lectura del manuscrito y sus valiosos comentarios y sugerencias, que mejoraron en gran medida este documento. El segundo autor agradece a la Vicerrectoría de Investigación y Extensión de la Universidad Industrial de Santander, Proyecto 3702.

Referencias

[1] P. A. Macmahon. “Memoir on the theory of the compositions of numbers.”, *Phil. Trans. RoyalSociety London*, A184, pp. 835-901, 1893.

[2] V. E. JR. Hoggatt y M. Bicknell, “Palindromic compositions”. *Fibonacci Quart.* vol. 13, pp. 350-356, 1975.

[3] K. Alladi y V. E. Hoggatt, “Compositions with ones and twos”. *Fibonacci Quart.* vol. 13, no. 3, pp. 1021-1031N, 1975.

[4] L. Carlitz, “Restricted compositions”. *J. Combin. Theory Ser. A*, vol 14, no. 3, pp. 254-264, 1976.

[5] A. K. Agarwal, “n-colour compositions”. *Indian J. Pure Appl. Math.*, vol. 31, no. 11, pp. 1421-1427, 2000.

[6] E. Deutsch, E. Munarini y S. Rinaldi, “Skew Dyck paths, area, and superdiagonal bargraphs”. *Journal of Statistical Planning and Inference*, vol. 140, pp. 1550-1562. 2009.

[7] J. Mantilla, W. Olaya-León y J. L. Ramírez, “Palindromic and colored superdiagonal compositions”. *Preprint.* Available online arXiv 2101.07733. 2021.

[8] W. A. Stein et al., “Sage Mathematics Software (v. 9.3)”. *The Sage Development Team* 2021.