Capítulo 2

Práctica 2. Combinatoria

2.1. Principios elementales

Hemos aprendido en la Práctica 1 cómo definir conjuntos, entre ellos el producto cartesiano y el conjunto potencia, y a calcular el cardinal de un conjunto.

Ejercicio 1. Use algunos comandos de Maxima para resolver los Ejercicios 1, 3 y 4 de la Relación de ejercicios del Tema 2.

Un problema típico que se resuelve aplicando el Principio de Inclusión-Exclusión es el que consiste en calcular cuántos enteros positivos $x \leq 1000$ existen tales que x es múltiplo de 5 ó de 7. La solución es 314. Veamos cómo podemos utilizar Maxima para verificar ésto.

En primer lugar generamos el conjunto A de todos los enteros x tales que $1 \leq x \leq 1000$.

```
(%ixx) A:setify(makelist(k,k,1,1000))$
```

A continuación definimos un predicado o función lógica que al aplicarse a un elemento $a \in A$ valdrá `true` si y sólo si a es múltiplo de 5 ó de 7.

```
(%ixx) f(x):=is(mod(x,5)=0) or is(mod(x,7)=0);
```

Entonces el número de elementos solicitado se obtiene con

```
(%ixx) cardinality(subset(A,f));
```

Como tarea para casa, debería de verificar este resultado aplicando el Principio de Inclusión-Exclusión. Recuerde que dados los números enteros positivos a y b, el número de enteros x tales que $1 \leq x \leq a$ y x es múltiplo de b, es igual al cociente que resulta de dividir a entre b.

Ejercicio 2. Use algunos comandos de Maxima para resolver los Ejercicios 11 y 15 de la Relación de ejercicios del Tema 2.

2.2. Combinaciones

En Maxima tenemos disponible el comando `binomial(n,k)` que nos permite calcular el coeficiente binomial $\binom{n}{k}$. Veamos algunos ejemplos.

```
(%ixx) binomial(7,2);
```

```
(%ixx) binomial(7,0);
```

```
(%ixx) binomial(7,10);
```

Ahora vamos a generar una lista cuyos elementos son los coeficientes binomiales $\binom{10}{k}$, para $k = 0, 1, 2, \ldots, 10$.

```
(%ixx) makelist(binomial(10,k),k,0,10);
```

Observe que el primer elemento generado es igual al último, el segundo es igual al penúltimo, etc. ¿Cómo se denomina esta propiedad de los coeficientes binomiales?

Al invocar el comando `binomial(n,k)`, también podemos dejar alguno de los valores n ó k sin determinar.

```
(%ixx) binomial(n,3);
```

Como vemos, Maxima devuelve su forma "simplificada".

Ejercicio 3. Calcule un entero positivo n sabiendo que el conjunto $\{1, 2, \ldots, n\}$ tiene exactamente 11486765 subconjuntos de cardinal 3. (Se recomienda plantear una ecuación y resolverla con el comando `solve`. Consulte la ayuda de Maxima para más información.)

A continuación comprobamos la identidad siguiente, la cual aparece propuesta en el Ejercicio 20(b) de la Relación de ejercicios del Tema 2:

$$\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n} = 2^n.$$

```
(%ixx) sum(binomial(n,k),k,0,n);
```

Con ésto estamos indicando a Maxima que realice la suma de los números $\binom{n}{k}$ para $k = 0, 1, 2, \ldots, n$.

Como podemos observar, Maxima no simplifica la sumatoria. Para obligar a Maxima que lo haga, escribimos la opción `simpsum`.

```
(%ixx) sum(binomial(n,k),k,0,n), simpsum;
```

Ejercicio 4. Escriba algunos comandos de Maxima para comprobar la igualdad siguiente:

$$\binom{m}{0} + \binom{m+1}{1} + \cdots + \binom{m+\ell}{\ell} = \binom{m+\ell+1}{\ell}.$$

A continuación resolvemos el Ejercicio 23 de la Relación de ejercicios del Tema 2 usando los comandos que hemos aprendido. Recuerde que $\binom{n}{\ell} = 0$ cuando $\ell > n$.

```
(%ixx) sum(binomial(n,2*k),k,0,n), simpsum;

(%ixx) sum(binomial(n,2*k+1),k,0,n), simpsum;
```

Como se puede observar, la simplificación de las dos sumatorias da el mismo resultado.

Para generar las k-combinaciones de un conjunto X, es decir, los subconjuntos de X con cardinal k, Maxima incorpora el comando `powerset(X,k)`.

```
(%ixx) powerset({a,b,c,d,e},1);
(%ixx) powerset({a,b,c,d,e},2);
(%ixx) powerset({a,b,c,d,e},3);
(%ixx) powerset({a,b,c,d,e},0);
(%ixx) powerset({a,b,c,d,e},8);
```

Podemos definir una función que hace esto mismo de manera artesanal como sigue.

```
(%ixx) combi(X,k):=subset(powerset(X),lambda([A],is(cardinality(A)=k)));

(%ixx) combi({a,b,c,d,e},1);
(%ixx) combi({a,b,c,d,e},2);
(%ixx) combi({a,b,c,d,e},3);
(%ixx) combi({a,b,c,d,e},0);
(%ixx) combi({a,b,c,d,e},8);
```

En nuestra definición hemos usado una función anónima ó función `lambda`, es decir, aquella que se define dentro de un comando ó de otra función sin darle un nombre. El alumno puede encontrar más información sobre este tema en la ayuda de Maxima.

Trabajamos ahora con la fórmula del binomio de Newton. Comenzamos expandiendo la expresión algebraica $(x+y)^8$:

```
(%ixx) expand((x+y)^8);
```

Según el Teorema del binomio de Newton, obtenemos este mismo resultado escribiendo lo siguiente:

```
(%ixx) sum(binomial(8,k)*x^k*y^(8-k),k,0,8);
```

Ejercicio 5. Utilice algunos comandos de Maxima para resolver el Ejercicio 21 de la Relación de ejercicios del Tema 2.

Seguidamente definimos una función `CR(n,k)` que nos permite calcular el número de combinaciones con repetición de `n` elementos tomados de `k` en `k`.

```
(%ixx) CR(n,k):=binomial(n+k-1,n-1);
```

Recuerde que `CR(n,k)` representa el número de soluciones de la ecuación

$$x_1 + x_2 + \cdots + x_n = k$$

donde cada incógnita se toma en $\mathbb{N} = \{0, 1, 2, \ldots\}$.

Por ejemplo, las soluciones de la ecuación $x_1 + x_2 = 10$ en $\mathbb{N}$ son $(0, 10), (1, 9), \ldots, (9, 1), (10, 0)$, y por tanto hay exactamente 11. Podemos obtener este mismo resultado evaluando

```
(%ixx) CR(2,10);
(%oxx) 11
```

Ejercicio 6. Utilice algunos comandos de Maxima para resolver los Ejercicios 26 y 27 de la Relación de ejercicios del Tema 2.

Ejercicio 7. Calcule un número natural d sabiendo que al lanzar simultáneamente d dados idénticos pueden obtenerse 80730 resultados posibles.

Hemos resuelto en clase el Ejercicio 28(d) de la Relación de ejercicios del Tema 2, según el cual tenemos que determinar el número de soluciones en $\mathbb{N}$ de la ecuación $x_1 + x_2 + x_3 = 20$ sujetas a la restricción $0 \leq x_1 < 7$. Vamos a obtener el mismo resultado construyendo el conjunto de soluciones mediante comandos apropiados de Maxima.

```
(%ixx) A:setify(makelist(k,k,0,20))$
(%ixx) X:cartesian_product(A,A,A)$
(%ixx) f(li):=is((li[1]+li[2]+li[3])=20) and is(li[1] < 7);
(%ixx) cardinality(subset(X,f));
```

La respuesta obtenida es 126 que coincide con el valor de $\binom{22}{2} - \binom{15}{2}$, tal y como obtuvimos en clase.

```
(%ixx) binomial(22,2)-binomial(15,2);
(%oxx) 126
```

Ejercicio 8. Aplique algunos comandos de Maxima para resolver los Ejercicios 28(e) y 29 de la Relación de ejercicios del Tema 2. Como tarea para casa, también sería conveniente que resolviera estos ejercicios aplicando la teoría.

2.3. Permutaciones

En Maxima cada permutación de un conjunto se representa como una lista de longitud igual al cardinal de dicho conjunto. Tenemos el comando `permutations(X)` que nos permite obtener el conjunto de las permutaciones de un conjunto X. Veamos algunos ejemplos.

```
(%ixx) permutations({a,b});
```

```
(%ixx) permutations({a,b,c});
```

```
(%ixx) permutations({a,b,c,d});
```

Nótese que las 24 permutaciones del conjunto `{a,b,c,d}` se presentan en pantalla **ordenadas lexicográficamente**. Si numeramos la primera de ellas `[a,b,c,d]` con el número 0, la segunda `[a,b,d,c]` con el número 1, y así hasta la última con el número 23, nos podemos plantear cuestiones del tipo siguiente. ¿Qué permutación se numera con el 16? ¿Con qué valor se numera la permutación `[b,c,d,a]`?

Ejercicio 9. Consideramos las permutaciones del conjunto `{1,2,3,4,5,6,7}` ordenadas lexicográficamente y las numeramos comenzando en cero. ¿Qué permutación se numera con el 2014? ¿Con qué valor se numera la permutación `[5,2,4,6,3,7,1]`?

Ejercicio 10. Resuelva, de forma teórica, el Ejercicio 45 de la Relación de ejercicios del Tema 2. A continuación use Maxima para confirmar el resultado obtenido.

El comando `permutations` también nos permite calcular las permutaciones de los elementos de una lista, donde, como sabemos pueden haber elementos repetidos.

```
(%ixx) permutations([a,b,c,d]);
```

En este ejemplo el resultado es idéntico al del comando `permutations({a,b,c,d})`.

Como consecuencia de lo anterior, podemos generar **permutaciones con repetición**.

Obtenemos todas las formas de permutar las letras de la palabra RELEER.

```
(%ixx) permutations([R,E,L,E,E,R]);
```

Calculamos el número de tales permutaciones:

```
(%ixx) cardinality(%);
```

De forma alternativa, el número de permutaciones de dicha palabra viene dado por el denominado **coeficiente multinomial**:

$$\binom{6}{3,2,1} = \frac{6!}{3!2!1!} = 60.$$

Repase el Ejercicio 47 de la Relación de Ejercicios del Tema 2.

En Maxima existe el comando `multinomial_coeff`$(r_1, ..., r_t)$ que permite calcular el coeficiente multinomial

$$\binom{n}{r_1, \ldots, r_t} = \frac{n!}{r_1! \cdot \cdots \cdot r_t!},$$

con $n = r_1 + \cdots + r_t$.

Nótese que en dicho comando sólo escribimos la lista de números que van en la segunda fila del coeficiente multinomial.

Obtenemos la respuesta del ejemplo anterior (es decir, las permutaciones de la palabra `RELEER`) usando este comando.

```
(%ixx) multinomial_coeff(3, 2, 1);
```

Los coeficientes multinomiales son una generalización de los coeficientes binomiales. Si hacemos $t = 2$ y $r_1 + r_2 = n$, resulta que

$$\binom{n}{r_1, r_2} = \frac{n!}{r_1!r_2!} = \frac{n!}{r_1!(n-r_1)!} = \binom{n}{r_1} = \binom{n}{r_2}.$$

Ejercicio 11. ¿De cuántas formas podemos permutar todas las letras de la palabra CONCURRENCIA? ¿En cuántas de dichas permutaciones aparecen las dos letras R consecutivas?

El Teorema del binomio de Newton se generaliza de la forma siguiente.

Teorema multinomial.
Sean $x_1, \dots, x_t$ elementos de un anillo tales que $x_i \cdot x_j = x_j \cdot x_i$ para cualesquiera $i, j \in \{1, 2, \dots, t\}$ y sea $n \in \mathbb{N}$. Entonces

$$(x_1 + \cdots + x_t)^n = \sum_{\substack{0 \leq r_1, r_2, \dots, r_t \leq n \\ r_1 + r_2 + \cdots + r_t = n}} \binom{n}{r_1, r_2, \dots, r_t} \cdot x_1^{r_1} \cdot x_2^{r_2} \cdots x_t^{r_t}.$$

□

Recuerde que el grado de un término $x_1^{s_1} \cdot x_2^{s_2} \cdots x_t^{s_t}$ es $s_1 + s_2 + \cdots + s_t$. Por tanto todos los términos $x_1^{r_1} \cdot x_2^{r_2} \cdots x_t^{r_t}$ que aparecen en el desarrollo del polinomio $(x_1 + \cdots + x_t)^n$ tienen grado n, pues $r_1 + r_2 + \cdots + r_t = n$. Además hay tantos términos distintos como t-uplas de números naturales $(r_1, r_2, \dots, r_t)$ tales que

$$r_1 + r_2 + \cdots + r_t = n.$$

Dicho número, como ya sabemos, viene dado precisamente por `CR(t,n)`.

Este teorema nos permite calcular el coeficiente de un término en un polinomio sin tener que desarrollarlo. Por ejemplo, si desarrollamos el polinomio $(x + y + z)^4$,

```
(%ixx) expand((x+y+z)^4);
```

podemos constatar que el coeficiente del término xy^2z es 12, y el coeficiente del término x^3z es 4. Obtenemos esto mismo usando el Teorema multinomial.

```
(%ixx) multinomial_coeff(1, 2, 1);
(%oxx) 12
(%ixx) multinomial_coeff(3, 0, 1);
(%oxx) 4
```

Además observe que el número de términos obtenidos en el desarrollo de $(x + y + z)^4$ es igual a 15. Según lo anterior, podemos obtener ésto mismo, sin tener que desarrollar el polinomio, escribiendo

```
(%ixx) CR(3, 4);
(%oxx) 15
```

Hemos escrito 3 porque hay tres variables y 4 por el exponente.

Ejercicio 12. Calcule el coeficiente del término $x^7y^5z^4$ en el polinomio $(x - 2y - 3z)^{16}$.

Ejercicio 13. ¿Cuántos términos aparecen al desarrollar el polinomio $(11x + y - 9z + 15t)^{20}$?

Maxima también permite generar permutaciones aleatorias de una lista o conjunto mediante el comando `random_permutation`.

```
(%ixx) random_permutation({1,2,3,4});
(%ixx) random_permutation({1,2,3,4});
```

Incluso podemos crear una lista de permutaciones generadas aleatoriamente.

```
(%ixx) makelist(random_permutation({1,2,3,4}),i,1,10);
```

Si lo ejecuta nuevamente, seguro que piensa que se va a repetir la lista, pero no es así.

```
(%ixx) makelist(random_permutation({1,2,3,4}),i,1,10);
```

2.4. Variaciones ordinarias

¿Cómo podemos generar las variaciones ordinarias de n elementos tomados de k en k?

Supongamos el conjunto $X = \{x_1, \ldots, x_n\}$. Generamos en primer lugar las k-combinaciones de X y para cada una de ellas calculamos las permutaciones de sus elementos.

Todo este proceso se implementa en la función siguiente:

```
(%ixx) variordi(X,k):=apply(union,makelist(permutations(li),li,listify(powerset(X,k))));
```

La probamos con algunos ejemplos.

```
(%ixx) variordi({1,2,3,4,5},2);
(%ixx) variordi({1,2,3,4,5},3);
(%ixx) variordi({1,2,3,4,5},5);
(%ixx) variordi({1,2,3,4,5},1);
(%ixx) variordi({1,2,3,4,5},8);
```

Ahora definimos la función siguiente.

```
(%ixx) p(n,k):=prod(i,i,n-k+1,n);
```

Estamos calculando el producto de los números $n-k+1, n-k+2, \ldots, n-1, n$, lo cual es el número de variaciones ordinarias de n elementos tomados de k en k. Evaluemos la función `p(n,k)` para algunos valores.

```
(%ixx) p(5,2);
```

Puede comprobar que el resultado obtenido es el número de elementos del conjunto `variordi({1,2,3,4,5},2)` previamente calculado.

Observe que la simple definición anterior de la función `p(n,k)` cubre todos los casos particulares.

```
(%ixx) p(5,0);
(%ixx) p(5,1);
(%ixx) p(5,5);
(%ixx) p(5,8);
```

2.5. Desordenes

Un desorden del conjunto $\{1, 2, \ldots, n\}$ es una permutación de los elementos de dicho conjunto, ninguno de los cuales aparece en su posición natural. Por ejemplo $2, 1$ es el único desorden del conjunto $\{1, 2\}$. Los desórdenes del conjunto $\{1, 2, 3\}$ son $2, 3, 1$ y $3, 1, 2$.

¿Cómo podemos calcular los desordenes del conjunto $\{1, 2, \ldots, n\}$? Una forma simple (aunque no la más óptima) que aprovecha los comandos ya conocidos de Maxima es la siguiente.

```
(%ixx) desordenes(n):=block([ini:makelist(i,i,1,n)],
       subset(permutations(ini),lambda([d],not equal(apply("*",d-ini),0))));
```

Lo aplicamos a algunos ejemplos.

```
(%ixx) desordenes(1);
(%ixx) desordenes(2);
(%ixx) desordenes(3);
(%ixx) desordenes(4);
(%ixx) desordenes(5);
```

¿Entiende cómo actúa la función `desordenes` que acabamos de definir?

Si denotamos el número de desordenes del conjunto $\{1, 2, \ldots, n\}$ por $d(n)$ y aplicamos el Principio de Inclusión-Exclusión, llegaremos a que

$$d(n) = n! \cdot \left(1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \cdots + (-1)^n \frac{1}{n!}\right) \quad \text{para todo } n \geq 1.$$

Observe que con ésto hemos dado la solución del Ejercicio 12 de la Relación de Ejercicios del Tema 2.

Ejercicio 14. Implemente la función anterior `d(n)` en Maxima y utilícela para calcular el número de desordenes para $n = 10$. (Puede serle útil el comando `sum`.)

En la fórmula anterior, cuando n se hace cada vez más grande, el sumando entre paréntesis se va aproximando al número $\frac{1}{e} \approx 0{,}36787944117144$.

Por tanto, para n suficientemente grande, podemos afirmar que aproximadamente un tercio de las permutaciones del conjunto $\{1, 2, \ldots, n\}$ son desordenes. Ésto explica que el número de desordenes vaya aumentando muy rápidamente conforme aumenta n.

2.6. Distribuciones de objetos en cajas

Una partición de un entero positivo n es una descomposición de n como suma de enteros positivos, sin tener en cuenta el orden de los sumandos. Por ejemplo 3, $2 + 1$ y $1 + 1 + 1$ son todas las particiones de 3.

Maxima incluye el comando `integer_partitions(n)` que nos devuelve todas las particiones de `n`. Vea también el comando `integer_partitions(n,long)`.

```
(%ixx) integer_partitions(1);
(%ixx) integer_partitions(2);
(%ixx) integer_partitions(5);
```

El número de Stirling de segunda especie, $S(n, k)$, es el número de formas de repartir n objetos diferentes en k cajas iguales, de forma que ninguna caja se quede vacía. En Maxima estos números están disponibles escribiendo `stirling2(n, k)`.

```
(%ixx) stirling2(5,1);
(%ixx) stirling2(5,2);
(%ixx) stirling2(5,6);
(%ixx) stirling2(5,7);
(%ixx) makelist(stirling2(n+1,n),n,1,8);
```

En vista del resultado del último comando, ¿cuál sería una fórmula para $S(n + 1, n)$? Si ejecuta el comando `stirling2(n+1, n)` verá que Maxima no es tan listo como parece.

Ejercicio 15. En todos los apartados de este ejercicio se supone que todos los objetos mencionados caben en cualquiera de las cajas mencionadas.

1. ¿De cuántas formas podemos colocar 20 objetos distintos en 10 cajas distintas, si se permite que algunas cajas puedan quedar vacías?

2. ¿De cuántas formas podemos colocar 20 objetos distintos en 10 cajas distintas, si ninguna caja puede quedar vacía?

3. ¿De cuántas formas podemos colocar 20 objetos idénticos en 10 cajas distintas, si se permite que algunas cajas puedan quedar vacías?

4. ¿De cuántas formas podemos colocar 20 objetos idénticos en 10 cajas distintas, si ninguna caja puede quedar vacía?

5. ¿De cuántas formas podemos colocar 20 objetos distintos en 10 cajas idénticas, si ninguna caja puede quedar vacía?
6. ¿De cuántas formas podemos colocar 20 objetos distintos en 10 cajas idénticas, si se permite que algunas cajas puedan quedar vacías?
7. ¿De cuántas formas podemos colocar 20 objetos idénticos en 10 cajas idénticas, si permitimos que algunas cajas puedan quedar vacías?