Cheatsheet - Combinatoria básica

Emanuel G. Mompó

7 de febrero de 2017

Las armas principales con las que atacar los problemas esenciales de combinatoria son las que siguen. Son los "*building blocks*". Es decir, se tendrá que reducir los problemas a cuestiones sencillas abordables con ellas.

1. Principio de la suma

Se utiliza cuando las formas de resolver un problema son **incompatibles** entre sí. Entonces se suman las posibilidades de cada forma de resolución.

Ejemplo 1 *¿Cuántas cadenas de* 5 *letras se pueden formar si no se pueden combinar vocales y consonantes?*

El enunciado ya establece que tener al menos una vocal es incompatible con tener consonantes. Entonces calculamos, por separado, cuántas hay que tengan sólo vocales, y cuántas que tengan sólo consonantes. Finalmente sumamos.

Observación 1 *A veces, los casos incompatibles surgen en el desarrollo de la solución del problema. Incluso los podemos introducir artificialmente.*

2. Principio del producto

Se utiliza cuando un problema se puede resolver como una serie de pasos **independientes**. En tal caso, se calculan las posibilidades de cada paso, y se multiplican.

Una forma intuitiva de ver este tipo de problemas es usando un arbol de posibilidades.

Ejemplo 2 *¿Cuántas series de longitud* 3 *se pueden escribir si usamos números del* 1 *al* 7*?*

Lo podemos ver como una tarea que requiere 3 intrucciones: la elección en la primera posición, la segunda y, finalmente, la tercera. En la primera posición tenemos 7 posibles candidatos, al igual que en la segunda y la tercera. Por tanto la solución es $7 \cdot 7 \cdot 7 = 7^3$.

2.1. Ordenando un conjunto (permutaciones)

Supongamos que tenemos una serie de instrucciones, todas **diferentes** e **independientes**, y queremos saber de cuántas maneras podemos organizarlas. Al ser independientes, quiere decir que da igual cuál vaya antes o después (el resultado será el mismo). Sin embargo, cada vez que realizamos una de las tareas, no podemos repetirla, por lo que introducimos una dependencia a las instrucciones realizadas con anterioridad.

Para hacerlo más sencillo, ponemos una etiqueta numérica a cada una de las instrucciones, empezando desde el 1 y acabando en el número n de instrucciones. Así, nos estamos preguntando por el número de posibles ordenaciones del conjunto $\{1, 2, \dots, n\}$. A esto se le llaman **permutaciones**.

¿Cuántas permutaciones hay si tenemos que realizar n tareas? Se puede hacer un arbol de posibilidades, y ver que, primero, hay n posibles elecciones. Entonces quedan $n-1$ tareas (¡que eran independientes de la anterior!). Luego podemos elegir entre $n-2$. Así, hasta que queda una sola tarea y ya no podemos elegir. Por tanto, hay

$$n! = n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1$$

posibles formas de organizar las tareas (o permutaciones de n elementos).

Ejemplo 3 *Si queremos hacer la compra, fregar los platos, y mirar una serie en Netflix, nos podemos organizar de* $3! = 6$ *maneras diferentes. Usando las etiquetas numéricas:*

$$\begin{array}{ccc}
(1, & 2, & 3) \\
(1, & 3, & 2) \\
(2, & 1, & 3) \\
(2, & 3, & 1) \\
(3, & 1, & 2) \\
(3, & 2, & 1)
\end{array}$$

2.2. Orden con "elementos repetidos"

Si alguna de las tareas se tiene que realizar varias veces (digamos, k veces), lo que hacemos es considerar que, a pesar de ser la misma tarea, le ponemos varias etiquetas (k etiquetas diferentes). Entonces calculamos el número total de ordenaciones posibles de todas las tareas (seguimos pensando que tenemos un total de n tareas, donde k en realidad son la misma tarea), y quitamos (dividiendo) las permutaciones posibles de la tarea que se hace varias veces. En este caso sería

$$\frac{n!}{k!}$$

Si hubiesen más tareas que se repitan, seguimos dividiendo por las permutaciones posibles de esas otras tareas.

2.3. Extracciones ordenadas

Si de muchas tareas (n tareas), independientes y diferentes, elegimos realizar unas pocas (por ejemplo, r), de nuevo, haciendo un arbol de posibilidades se puede ver que hay

$$n \cdot (n-1) \cdot \ldots \cdot (n-r+1) = \frac{n!}{(n-r)!}$$

posibles *configuraciones* de tareas.

Ejemplo 4 *Tenemos las mismas tareas que en el Ejemplo 3, pero sólo vamos a hacer 2 de ellas. Entonces tenemos* $3!/1! = 6$ *posibles configuraciones:*

$$\begin{array}{cc} (1, \ 2) & (2, \ 3) \\ (1, \ 3) & (3, \ 1) \\ (2, \ 1) & (3, \ 2) \end{array}$$

Observación 2 *Como se ve en el ejemplo, se está teniendo en cuenta el orden de realización de las tareas a la hora de contar, aunque las tareas sean independientes.*

3. Elección de subconjuntos (números combinatorios)

De un conjunto de n elementos queremos escoger r. **Nos da igual el orden**. Sólo nos preocupa la **elección**. Para esto se usan los números combinatorios, que se denotan:

$$\binom{n}{r} = \frac{n!}{r!(n-r)!}$$

Esto en inglés suele tener un nombre bastante descriptivo: "n *choose* r". Es decir, dándole algo de civismo a la frase: de n candidatos, escoger r.

Ejemplo 5 *La cantidad de posibilidades del bingo es un claro ejemplo de* n choose r.

Observación 3 *Si después de elegir los* r *elementos de un conjunto de* n*, también queremos tener en cuenta las permutaciones de esos* r *elementos (es decir, queremos contar también el orden), deberiamos multiplicar por* $r!$ *(principio del producto). ¿Qué nos queda en ese caso?*

$$\binom{n}{r} \cdot r! = \frac{n!r!}{r!(n-r)!} = \frac{n!}{(n-r)!}$$

¡que es la expresión de las extracciones ordenadas! Con esto se quiere decir que un mismo problema se puede enfocar desde diferentes ángulos.

4. Principio de Inclusión-Exclusión

Esto se puede ver como una generalización del principio de la suma. Si de entre todas las posibles configuraciones de un problema, queremos contar las que cumplen bien una propiedad A, bien una propiedad B, si ambas propiedades son incompatibles, usamos el principio de la suma. Es decir, contamos las que cumplen A por un lado, las que cumplen B por otro, las sumamos y ya está.

Pero si hay configuraciones que cumplen ambas propiedades (lo que hace que las propiedades no sean incompatibles), al contar las que cumplen A por un lado, y las que cumplen B por otro, habríamos contado dos veces las que cumplen tanto A como B. Por ello restamos una vez la cantidad de configuraciones que cumplen ambas propiedades. En un abuso se notación, haríamos:

$$|A \cup B| = |A| + |B| - |A \cap B|$$

Ejemplo 6 *Entre los números del bingo (que van del 1 al 75), ¿cuántos hay que sean pares o que sean primos?*

Ser par y ser primo no es independiente. Al menos, no lo es siempre que consideremos el 2*, que es el único número primo par.*

La cantidad de primos hasta el 75 es $\pi(75) = 21$*. La cantidad de pares entre 1 y 75 es* $\lfloor 75/2 \rfloor = 37$*.*

Como 2 cumple ambas propiedades, lo estamos teniendo en cuenta en ambos casos. Como es el único número que lo cumple, sólo habrá que restar uno.

Por tanto la solución es $21 + 37 - 1 = 57$*.*

5. Dar vuelta el problema

Cuando queremos contar la cantidad de configuraciones que cumplen una propiedad A, a veces es más fácil contar las configuraciones que **no** la cumplen y restarlas del total de configuraciones posibles. Lo que quedará no puede ser ni más, ni menos, que las configuraciones que cumplen la propiedad.

Ejemplo 7 *Si consideramos que el alfabeto tiene 26 caracteres, ¿Cuántas cadenas de 10 letras hay que contengan la letra L?*

Si nos ponemos a ver todas las posibilidades nos hacemos la picha un lío: si tiene una sóla L, dónde colocarla... si tiene dos, dónde se pueden colocar (recordatorio: de 10 posiciones, escoger 2)... cómo rellenar el resto de huecos... Mucho trabajo.

Si lo damos vuelta, queremos contar todas las palabras que NO contengan la L. Usando el principio del producto, tenemos que serán 25^{10} *(así de fácil). Pero se las tenemos que restar al total, que es* 26^{10}*. O sea, la respuesta al problema es* $26^{10} - 25^{10} = 45\ 799\ 664\ 012\ 751 \approx 4.58 \cdot 10^{13}$*.*