

Sea S una cadena con n paréntesis que abren y n paréntesis que cierran.

Dada una longitud l impar, decimos que es un l -posicionamiento uniforme de S si $s(i)$ es par y al escribir el i -ésimo paréntesis que abre en $s(i)$ y el i -ésimo paréntesis que cierra en $s(i) + l$, $1 \leq i \leq n$, se obtiene una escritura válida de S . Por ejemplo, si $S = (((()))(()))$ y $l = 15$, entonces $s(1) = 0$, $s(2) = 6$, $s(3) = 10$, $s(4) = 16$, $s(5) = 20$, $s(6) = 22$, y $s(7) = 36$ es un 15-posicionamiento uniforme de S .

Definir un SRD que permita resolver el problema de determinar si una cadena dada S tiene un l -posicionamiento uniforme cuando $l > 0$ impar también es dado. El mejor SRD que conocemos tiene $O(n)$ inecuaciones y, por lo tanto, permite resolver el problema en $O(n^2)$ aplicando el algoritmo del ejercicio anterior.

Sea $A = \{a_1, a_2, \dots, a_n\}$ y $B = \{b_1, b_2, \dots, b_n\}$ las posiciones de los paréntesis que en S abren y cierran respectivamente.

Observemos que en la secuencia se debe cumplir que $\forall 1 \leq i < j \leq n, a_i < a_j \wedge b_i < b_j$.

Por lo tanto la función $s(i)$ debe cumplir: $s(i) < s(j) \wedge s(i) + l \leq s(j) + l$, lo cual vale si la función es estrictamente creciente, ya que $l > 0$. Sin embargo, hay que pedir un poco más, ya que si tenemos un paréntesis que cierra antes de un paréntesis que abre, se debe respetar la relación de orden, es decir:

$$b_i < a_j \implies s(i) + l < s(j).$$

$$\text{Análogamente } b_i > a_j \implies s(i) + l > s(j).$$

Podemos plantear las primeras n desigualdades como:

$$\begin{aligned} x_1 - x_2 &\leq -1 \\ x_2 - x_3 &\leq -1 \\ &\vdots \\ x_{n-1} - x_n &\leq -1 \end{aligned}$$

Y luego, para todas las demás instancias en el paréntesis donde tenemos desórdenes del estilo anteriormente mencionado, se plantea como:

$$\begin{aligned} x_i - x_j &\leq l - 1 \\ \text{y } x_j - x_i &\leq l - 1 \end{aligned}$$

según el caso. En el peor caso hay n desórdenes; luego, la mayor cantidad de inecuaciones que podemos tener es $O(n)$. Luego podemos usar el algoritmo anterior, dándonos una complejidad de resolución de $O(n^2)$.