Esteban Antón Sellés - 48788593G

Complejidad temporal casos especiales

Mejor caso

El mejor caso en cuanto a la complejidad temporal para el cálculo de casos especiales es:

$$\Omega(n)$$

Siendo *n* el tamaño de la cadena a tokenizar y *d* el tamaño de la cadena de delimitadores. Esto se podría dar en el caso de que se encontrase un caracter que no es delimitador en la primera posición que se busca y el primer delimitador seguidamente, además de no pertenecer a ningún caso especial. Esta complejidad vendría dada principalmente por el while que va moviendo las posiciones para saber la siguiente cadena la cual tokenizar.

Caso peor

El caso peor en cuanto a la complejidad temporal para el cálculo de casos especiales es:

$$O(2*(pos - lastPos) + (pos - lastPos)*(n*d)) =$$
$$= O(2*n + n*(n*d)) = O(n^2*d + 2*n) =$$
$$= O(n^2)$$

Perteneciendo *n* y *d* las mismas medidas que en el *Mejor caso* y, *lastPos* a la posición donde se encuentra el primer caracter que no es un delimitador y *pos* a la posición del primer delimitador a partir de lastPos; esta complejidad se ve afectada principalmente, y siendo pos - lastPos = n, por el caso en que se estudie el caso especial de ser un decimal, el cual valida primero con la función **esNumero** de coste $O(2*(pos - lastPos))$ y después a la hora de tokenizar el número decimal se haría uso de la función **tokenizarDecimal** de coste $O((pos - lastPos)*(n*d))$.