Computación Simbólica

Ángel Ríos San Nicolás

20 de noviembre de 2020

Ejercicio 1. *En el caso $|f| = |g| = n$, si tomamos $k = \lceil \log_2(n) \rceil$, podemos añadir ceros a la representación de f y g hasta llegar a una representación $|f| = |g| = 2^k$. Para esta representación, Karatsuba necesita hacer $\mathcal{O}((2^k)^{\log_2(3)})$. Compare, asintóticamente, las cantidades*

$$n^{\log_2(3)}, \qquad (2^k)^{\log_2(3)}$$

Solución. Vamos a ver que $\mathcal{O}\left(n^{\log_2(3)}\right) = \mathcal{O}\left((2^k)^{\log_2(3)}\right)$ acotando los cocientes.

Como $\lceil \log_2(n) \rceil \leq \log_2(n) + 1$ para todo $n \in \mathbb{N}$,

$$\frac{(2^k)^{\log_2(3)}}{n^{\log_2(3)}} = \frac{\left(2^{\lceil \log_2(n) \rceil}\right)^{\log_2(3)}}{n^{\log_2(3)}} = \frac{2^{\log_2(3)\lceil \log_2(n) \rceil}}{n^{\log_2(3)}} \leq \frac{2^{\log_2(3)(1+\log_2(n))}}{n^{\log_2(3)}} = \frac{2^{\log_2(3)+\log_2(3)\log_2(n)}}{n^{\log_2(3)}} =$$

$$= \frac{2^{\log_2(3)} 2^{\log_2\left(n^{\log_2(3)}\right)}}{n^{\log_2(3)}} = 3\frac{n^{\log_2(3)}}{n^{\log_2(3)}} = 3.$$

Por tanto, existe un $N \in \mathbb{N}$ tal que para todo $n > N$, $(2^k)^{\log_2(3)} \leq 3n^{\log_2(3)}$ con lo que

$$(2^k)^{\log_2(3)} \in \mathcal{O}\left(n^{\log_2(3)}\right).$$

Como $\log_2(n) \leq \lceil \log_2(n) \rceil$ para todo $n \in \mathbb{N}$,

$$\frac{n^{\log_2(3)}}{\left(2^{\lceil \log_2(n) \rceil}\right)^{\log_2(3)}} = \frac{n^{\log_2(3)}}{2^{\log_2(3)\lceil \log_2(n) \rceil}} \leq \frac{n^{\log_2(3)}}{2^{\log_2(3)\log_2(n)}} = \frac{n^{\log_2(3)}}{2^{\log_2\left(n^{\log_2(3)}\right)}} = \frac{n^{\log_2(3)}}{n^{\log_2(3)}} = 1.$$

Por tanto, existe un $N \in \mathbb{N}$ tal que para todo $n > N$, $n^{\log_2(3)} \leq (2^k)^{\log_2(3)}$ y también

$$n^{\log_2(3)} \in \mathcal{O}\left((2^k)^{\log_2(3)}\right).$$

Con esto hemos probado que

$$\mathcal{O}\left((2^k)^{\log_2(3)}\right) = \mathcal{O}\left(n^{\log_2(3)}\right),$$

es decir, que las cantidades son iguales asintóticamente en términos de $\mathcal{O}$.

Ejercicio 2. *Describa el método de multiplicación de Karatsuba para la multiplicación de enteros.*

Solución. Sean $f, g \in \mathbb{Z}$. Podemos expresar f, g en base 2 de la forma

$$f = a_0 + a_1 2 + a_3 2^2 + \cdots + a_{n-1} 2^{n-1} = [a_{n-1} \dots a_0]$$
$$g = b_0 + b_1 2 + b_3 2^2 + \cdots + b_{n-1} 2^{n-1} = [b_{n-1} \dots b_0]$$

con $a_i, b_i \in \{0, 1\}$ para todo $i \in \{1, \dots, n\}$ donde estamos suponiendo que $n = 2^k$ con $k \in \mathbb{N}$ añadiendo ceros si es necesario.

A partir de

$$f = a_0 + a_1 2 + \cdots + a_{\frac{n}{2}-1} 2^{\frac{n}{2}-1} + 2^{\frac{n}{2}} \left(a_{\frac{n}{2}} + a_{\frac{n}{2}+1} 2 + \cdots + a_n 2^{\frac{n}{2}}\right)$$
$$g = b_0 + b_1 2 + \cdots + b_{\frac{n}{2}-1} 2^{\frac{n}{2}-1} + 2^{\frac{n}{2}} \left(b_{\frac{n}{2}} + b_{\frac{n}{2}+1} 2 + \cdots + b_n 2^{\frac{n}{2}}\right),$$

definimos

$$f_0 = a_0 + a_1 2 + \cdots + a_{\frac{n}{2}-1} 2^{\frac{n}{2}-1} \qquad f_1 = a_{\frac{n}{2}} + a_{\frac{n}{2}+1} 2 + \cdots + a_n 2^{\frac{n}{2}}$$
$$g_0 = b_0 + b_1 2 + \cdots + b_{\frac{n}{2}-1} 2^{\frac{n}{2}-1} \qquad g_1 = b_{\frac{n}{2}} + b_{\frac{n}{2}+1} 2 + \cdots + b_n 2^{\frac{n}{2}},$$

que son enteros de tamaño la mitad que los originales f y g y con los que se tiene $f = f_0 + 2^{\frac{n}{2}} f_1$ y $g = g_0 + 2^{\frac{n}{2}} g_1$. El producto es

$$fg = f_0 g_0 + 2^{\frac{n}{2}} (f_0 g_1 + f_1 g_0) + 2^n f_1 g_1.$$

Tomamos $u = f_0 g_0$, $v = f_1 g_1$ y $w = f_0 g_0 - f_0 g_1 - f_1 g_0 + f_1 g_1 = (f_0 - f_1)(g_0 - g_1)$, de manera que $f_0 g_1 + f_1 g_0 = w + u - v$ y podemos escribir

$$fg = u + 2^{\frac{n}{2}} (u + v - w) + 2^n v.$$

Sabemos que en las restas $f_0 - f_1$ y $g_0 - g_1$ no tenemos ningún dígito extra porque en valor absoluto $|f_0 - f_1| \leq \max\{|f_0|, |f_1|\}$ con lo que el número de dígitos del resultado se puede tomar igual que el de las entradas. Podremos obtener resultados intermedios negativos, pero esto no afectará a la complejidad porque el número de dígitos es el mismo, solo hay que controlar el signo. Con esto sustiuimos el producto de dos enteros de $n > 1$ dígitos en tres productos de enteros de $\frac{n}{2}$ dígitos y cuatro sumas de enteros de n dígitos (porque al multiplicar dos números de $\frac{n}{2}$ dígitos obtenemos un número de a lo sumo n dígitos). No contamos los productos por potencias de 2 porque en base 2 se corresponden simplemente a añadir ceros al final.

El algoritmo de Karatsuba consiste en aplicar este esquema de manera recursiva a los tres productos $f_0 g_0$, $f_1 g_1$ y $(f_0 - f_1)(g_0 - g_1)$ hasta que llegamos a productos de enteros de un dígito que sucede siempre porque n es potencia de 2.

Calculamos ahora el coste del algoritmo. Si $T(n)$ es el coste del algortimo de Karatsuba en número de operaciones bit para tamaño de la entrada en base 2 igual a $n = 2^k$, lo que acabamos de decir implica que $T(n) \leq 3T(\frac{n}{2}) + 4n$. Aplicando la recursión, obtenemos

$$\begin{aligned}
T(n) &\leq 3T\left(\frac{n}{2}\right) + 4n \leq \\
&\leq 3\left(3T\left(\frac{n}{4}\right) + 4\frac{n}{2}\right) + 4n = 3^2 T\left(\frac{n}{4}\right) + 4n\left(\frac{3}{2} + 1\right) \leq \\
&\leq 3^2\left(3T\left(\frac{n}{8}\right) + 4\frac{n}{4}\right) + 4n\frac{3}{2} = 3^3 T\left(\frac{n}{8}\right) + 4n\left(\left(\frac{3}{2}\right)^2 + \frac{3}{2} + 1\right) \leq \\
&\vdots \ (k \text{ pasos}) \\
&\leq 3^k T(1) + 4n \sum_{i=0}^{k-1} \left(\frac{3}{2}\right)^i = 3^k T(1) + 4n \frac{\left(\frac{3}{2}\right)^k - 1}{\frac{3}{2} - 1} = \\
&= 3^k T(1) + 8 \cdot 3^k - 8n = (T(1) + 8)3^k - 8n \leq (T(1) + 8)3^k.
\end{aligned}$$

Por tanto, $T(n) \leq (T(1) + 8)3^k$ donde $T(1) + 8$ es una constante con lo que $T(n) \in \mathcal{O}\left(3^k\right)$, pero como

$$3^k = 3^{\log_2(n)} = 2^{\log_2(3)\log_2(n)} = 2^{\log_2\left(n^{\log_2(3)}\right)} = n^{\log_2(3)},$$

el coste del algoritmo de Karatsuba es $\mathcal{O}\left(n^{\log_2(3)}\right)$ en el número de operaciones bit.

Ejercicio 3. *Busque en qué consiste el método de Horner y calcule el número de operaciones que realice. Compare el resultado con el lema anterior.*

Solución. El método de Horner es un algoritmo para evaluar polinomios univariados.

Si $n \in \mathbb{N}$, $f \in \mathbb{K}[X]$, $f = a_0 + a_1 X + a_2 X^2 + \cdots + a_n X^n$ y $x \in \mathbb{K}$. Podemos ir sacando factor común a X progresivamente manteniendo la igualdad de la siguiente manera:

$$\begin{aligned}
f &= a_0 + a_1 X + a_2 X^2 + \cdots + a_n X^n \\
&= a_0 + X(a_1 + a_2 X + \cdots + a_n X^n) \\
&= a_0 + X(a_1 + X(a_2 + a_3 X + \cdots + a_n X^n)) \\
&\vdots \\
&= a_0 + X(a_1 + X(a_2 + X(\cdots + X(a_{n-1} + X a_n)\cdots))).
\end{aligned}$$

Observando esto definimos, para cada $k \in \{0, \dots, n\}$, el polinomio

$$b_k = a_k + X(a_{k+2} + X(a_{k+3} + X(\cdots + X(a_{n-1} + Xa_n)\cdots))).$$

Claramente se cumplen:

$$\begin{aligned}
b_0 &= a_n \\
b_1 &= a_{n-1} + Xb_n = a_{n-1} + Xa_n \\
b_2 &= a_{n-2} + Xb_{n-1} = a_{n-2} + X(a_{n-1} + Xb_n) = a_{n-2} + X(a_{n-1} + Xa_n) \\
&\vdots \\
b_{n-2} &= a_2 + Xb_3 = a_2 + X(a_3 + Xb_4) = \cdots = a_2 + X(a_3 + X(\cdots + X(a_{n-1} + Xa_n)\cdots)) \\
b_{n-1} &= a_1 + Xb_2 = a_1 + X(a_2 + Xb_3) = \cdots = a_1 + X(a_2 + X(\cdots + X(a_{n-1} + Xa_n)\cdots)) \\
b_n &= a_0 + Xb_1 = a_0 + X(a_1 + Xb_2) = \cdots = a_0 + X(a_1 + X(\cdots + X(a_{n-1} + Xa_n)\cdots)) = f
\end{aligned}$$

El algoritmo de Horner consiste en calcular la evaluación $f\,(X = x)$ en $n + 1$ pasos calculando en orden $b_0(x), b_1(x) \dots, b_n(x) = f(x)$ donde, aprovechando los cálculos anteriores, reducimos el número de operaciones.

Vamos a calcular el número de sumas y productos en $\mathbb{K}$. En $b_0(x)$ no hacemos ni productos ni sumas. Para cada $k \in \{1, \dots, n\}$, en $b_k(x)$ hacemos el producto $xb_{k-1}(x)$ y la suma $a_k + xb_{k-1}(x)$ donde $b_{k-1}(x)$ había sido calculado ya en el paso anterior. En total, el método de Horner evalúa el polinomio haciendo n sumas y n productos. El coste del algoritmo es $\mathcal{O}(n)$ en el número de operaciones en $\mathbb{K}$.

Utilizando el método de Horner, hacemos $n-1$ productos menos respecto al algoritmo del lema anterior en el que se calculan las potencias a^m con $m \in \{0, \dots, n\}$ aprovechando las anteriores lo que en total son $n - 1$ productos a los que hay que sumar los n productos por los coeficientes del polinomio, es decir, en total $2n - 1$ productos. El número de sumas es el mismo.