Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC1253 - Matemáticas Discretas

Ayudantía 10 - Algoritmos y notación asintótica

Héctor Núñez, Paula Grune, Manuel Irarrázaval

Solución

Pregunta 1

Se debe demostrar que $\log(n!) \in \mathcal{O}(n\log(n))$ y $n\log(n) \in \mathcal{O}(\log(n!))$.

Primera parte: $\log(n!) \in \mathcal{O}(n\log(n))$

Basta notar que:

$$n! = 1 \cdot 2 \cdots n \leq n \cdot n \cdots n = n^n$$

Aplicando logaritmo:

$$\log(n!) \leq \log(n^n) = n\log(n)$$

Luego, se concluye que $\log(n!) \in \mathcal{O}(n\log(n))$ con $c = 1$ y $n_0 = 1$.

Segunda parte: $n\log(n) \in \mathcal{O}(\log(n!))$

Usando la fórmula de Stirling y aplicando logaritmo:

$$\exists c, n_0. \forall n \geq n_0, \quad \log\left(c\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\right) \leq \log(n!)$$

Aplicando propiedades del logaritmo:

$$\log(n!) \geq \log\left(\left(\frac{n}{e}\right)^n\right) = n\log(n) - n\log(e)$$

$$\Rightarrow n\log(n) \leq \log(n!) + n\log(e)$$

Usando que $n \log(e) \in \mathcal{O}(\log(n!))$ (se cumple que $n \log(e) \leq \log(n!)$ a partir de cierto n), se tiene que:

$$n \log(n) \leq 2 \log(n!)$$

Si para la notación asintótica tomamos $c = 3$ (se sigue cumpliendo la desigualdad) tenemos que:

$$n \log(n) \leq 3 \log(n!) \Rightarrow n \log(n) \in \mathcal{O}(\log(n!))$$

Con $c = 3$ y n_0 el máximo de los utilizados en la demostración.

Pregunta 2

Sean $f : \mathbb{N} \to \mathbb{R}^+$y $g : \mathbb{N} \to \mathbb{R}^+$dos funciones cualesquiera. Demuestre o entregue un contraejemplo para las siguientes afirmaciones:

1. Si $f(n) \in \Theta(g(n))$ entonces $\min \{f(n), g(n)\} \in \Theta(\max \{f(n), g(n)\})$.
2. Si $f(n) \in O(g(n))$ entonces $f(n)^{g(n)} \in O\left(g(n)^{f(n)}\right)$.

Solución

1. Definiendo $h(n) = \min\{f(n), g(n)\}$ y $H(n) = \max\{f(n), g(n)\}$ Como $f \in \Theta(g)$, existen constantes $c_1, c_2 \in R, n_o \in N$ tal que

   $$c_1 g(n) \leq f(n) \leq c_2 g(n) \forall n \geq n_0 \tag{1}$$

   Despejando de la parte derecha de la desigualdad

   $$\frac{1}{c_2} f(n) \leq g(n) \forall n \geq n_0 \tag{2}$$

   Tenemos dos escenarios desde $n \geq n_0$:
   1. Cuando $f(n) \leq g(n), h(n) = f(n) \wedge H(n) = g(n)$. Usando (1) y la condición de este caso:

   $$c_1 g(n) \leq f(n) \leq 1 \cdot g(n) \text{ para } n \geq n_0 \text{ tq } f(n) \leq 1 \cdot g(n)$$
   $$\min \left\{c_1, \frac{1}{c_2}\right\} g(n) \leq c_1 g(n) \leq f(n) \leq 1 \cdot g(n)$$
   $$\min \left\{c_1, \frac{1}{c_2}\right\} H(n) \leq h(n) \leq 1 \cdot H(n)$$

2. Cuando $g(n) < f(n), h(n) = g(n) \wedge H(n) = f(n)$ Con (2) queda:

$$\frac{1}{c_2} f(n) \leq g(n) \leq 1 \cdot f(n) \text{ para } n \geq n_0 \text{ tq } f(n) > g(n)$$

$$\text{mín}\left\{c_1, \frac{1}{c_2}\right\} f(n) \leq \frac{1}{c_2} f(n) \leq g(n) \leq 1 \cdot f(n)$$

$$\text{mín}\left\{c_1, \frac{1}{c_2}\right\} H(n) \leq \frac{1}{c_2} f(n) \leq h(n) \leq 1 \cdot H(n)$$

Combinando ambos casos:

$$\text{mín}\left\{c_1, \frac{1}{c_2}\right\} H(n) \leq h(n) \leq 1 \cdot H(n) \quad \forall n \geq n_0$$

$$c_1' H(n) \leq h(n) \leq c_2' H(n) \quad \forall n \geq n_0$$

Con lo cual, $h(n) \in \Theta(H(n))$

2. La afirmación anterior es falsa, se puede demostrar a través de un contraejemplo: Sea $f(n) = 2$ y $g(n) = n$. De acuerdo a la jerarquía en notación $\mathcal{O}$ vista en clases se cumple que $f(n) \in \mathcal{O}(g(n)$. Se debe demostrar ahora que $f(n)^{g(n)} \notin \mathcal{O}\left(g(n)^{f(n)}\right)$: Nuevamente si consideramos la jerarquía en notación $\mathcal{O}$ vista en clases, si $f(n)^{g(n)} = 2^n$ y $g(n)^{f(n)} = n^2$. Entonces no se cumple que $f(n)^{g(n)} \in \mathcal{O}\left(g(n)^{f(n)}\right)$, lo que es equivalente a $f(n)^{g(n)} \notin \mathcal{O}\left(g(n)^{f(n)}\right)$. Por demostración a través de contraejemplo queda demostrado que la afirmación no se cumple para dos funciones arbitrarias.

Pregunta 3

Una fórmula proposicional φ está en k-DNF si está en DNF y cada conjunción tiene exactamente k literales. Por ejemplo, la fórmula $(p \wedge \neg q \wedge \neg q) \vee (\neg p \wedge q \wedge p)$ está en 3-DNF. Denotamos por C_i a la i-ésima cláusula de φ y por ℓ_{ij} al j-ésimo literal de C_i.

Considere el siguiente algoritmo para determinar si una fórmula en k-DNF es satisfactible.

DNF-SAT($C_1 \vee C_2 \vee \cdots \vee C_m,\ k$) [1] $i \in \{1, \ldots, m\}$ sat $\leftarrow$ TRUE $j \in \{1, \ldots, k-1\}$ $t \in \{j+1, \ldots, k\}$ ℓ_{mj} y ℓ_{mt} son complementarios $sat \leftarrow$ FALSE sat TRUE FALSE

Definimos la función $T(n)$ como el número de ejecuciones de la línea 5 de DNF-SAT cuando se llama para φ con n literales en total (contando repetidos). Determine justificadamente una expresión en notación $\mathcal{O}$ para $T(n)$.

Solución

En primer lugar, obtendremos una expresión para la función T en términos de m y k (dado que $n = mk$). Para esto, es posible realizar un análisis de mejor o peor caso. Mostraremos ambas opciones a continuación. Para ambas se considera un input con mk literales individuales.

(a) El peor caso del algoritmo ocurre cuando la fórmula no es satisfactible. Esto significa que el algoritmo ejecuta todas las iteraciones de los bloques `for` en el código, obteniendo el máximo número de ejecuciones de la línea 5. La siguiente expresión permite calcular de forma explícita dicho número:

$$
\begin{aligned}
T(m,k) &= \sum_{i=1}^{m} \sum_{j=1}^{k-1} \sum_{t=j+1}^{k} 1 \\
&= \sum_{i=1}^{m} \sum_{j=1}^{k-1} (k-j) \\
&= \sum_{i=1}^{m} \left(\sum_{j=1}^{k-1} (k-j) \right) \\
&= \sum_{i=1}^{m} \left(k(k-1) - \sum_{j=1}^{k-1} j \right) \\
&= \sum_{i=1}^{m} \left(k(k-1) - \frac{(k-1)k}{2} \right) \\
&= \sum_{i=1}^{m} \frac{(k-1)k}{2} \\
&= m \cdot \frac{(k-1)k}{2} \\
&= \frac{mk^2 - mk}{2} \in \mathcal{O}(mk^2)
\end{aligned}
$$

Con esto, concluimos que en el peor caso, $T(m,k) \in \mathcal{O}(mk^2)$.

(b) El mejor caso del algoritmo ocurre cuando la primera conjunción revisada por el algoritmo, i.e. C_1, no tiene literales complementarios. En tal caso, el `for` exterior (variable i) solo toma valor $i = 1$, mientras que los `for` interiores completan una iteración completa. Es decir, y recurriendo a cálculos realizados en el peor caso, se obtiene

$$\begin{aligned}
T(m,k) &= \sum_{i=1}^{1}\sum_{j=1}^{k-1}\sum_{t=j+1}^{k} 1 \\
&= \sum_{i=1}^{1}\frac{(k-1)k}{2} \\
&= \frac{(k-1)k}{2} \\
&= \frac{k^2}{2} - \frac{k}{2}
\end{aligned}$$

Con esto, concluimos que en el mejor caso, $T(m,k) \in \mathcal{O}(k^2)$.