Casa abierta al tiempo
UNIVERSIDAD AUTÓNOMA METROPOLITANA
Unidad Iztapalapa

Tarea 3. Notación Asintótica

Autores:
Iturbe Pineda Nancy Monserrat
Martínez Buenrostro Jorge Rafael
Romero Moreno Ozomatzin
Vivas Peláez Sofía

Profesor
Miguel Ángel Pizaña López

27 de Septiembre de 2024

1. Proporcione la definición de $f(n) = O(g(n))$. Explique su significado.

Solución. Formalmente, dadas dos funciones $f,\ g : R \to R$ se define $f(n) = O(g(n))$ si y sólo si existen constantes $c,\ N > 0$ tales que

$$|f(n)| \leq c|g(n)|$$

para todo $n \geq N$.

La definición anterior nos dice que la función $|f(n)|$ está acotada superiormente por $c|g(n)|$ para todo $n \geq N$ y con $c,\ N > 0$. En otras palabras, a partir de algún punto, la función $|f(n)|$ no excederá a la función $|g(n)|$, excepto quizás por la constante multiplicativa c.

2. Proporcione la definición de $f(n) = \Omega(g(n))$. Explique su significado.

Solución. Dadas dos funciones $f,\ g : R \to R$ se define $f(n) = \Omega(g(n))$ si y sólo si $g(n) = O(f(n))$. De acuerdo con el ejercicio 1, $f(n) = \Omega(g(n))$ si y sólo si existen constantes $c,\ N > 0$ tales que

$$|g(n)| \leq c|f(n)| \Leftrightarrow c^{-1}|g(n)| \leq |f(n)$$

para todo $n \geq N$.

Entonces la función $|f(n)|$ está acotada inferiormente por $c^{-1}|g(n)|$ para todo $n \geq N$ y con $c^{-1},\ N > 0$. Es decir, a partir de $n = N$, *la función* $|f(n)|$ es al menos tan grande como $c^{-1}|g(n)|$.

3. Proporcione la definición de $f(n) = \Theta(g(n))$. Explique su significado.

Solución. Dada dos funciones $f,\ g :\ R \to R$ se dice que $f(n) = \Theta(g(n))$ si y sólo si $f(n) = O(g(n))$ y $f(n) = \Omega(g(n))$.

De acuerdo con los ejercicios 1 y 2 podemos traducir el enunciado anterior como $f(n) = \Theta(g(n))$ si y sólo si existen constantes $c_1,\ c_2,\ N_1,\ N_2 > 0$ tales que

$$|f(n)| \leq c_2|g(n)| \text{ y } |g(n)| \leq c_1|f(n)| \qquad (1)$$

para todo $n \geq N_2$ y $n \geq N_1$ correspondientemente, es decir, $f(n) = O(g(n))$ y $g(n) = O(f(n)$ .Equivalentemente,

$$c_1^{-1}|g(n)| \leq |f(n)| \leq c_2|g(n)| \qquad (2)$$

donde la desigualdad de la izquierda se satisface para todo $n \geq N_1$ y la de la derecha para todo $n \geq N_2$.Tomemos $c = máx\{c_1,\ c_2\}$ y $N = máx\{N_1, N_2\}$. De la ecuación (2) se deduce que $f(n) = \Theta(g(n))$ si y sólo si existen constantes positivas $c,\ N > 0$ tales que

$$c^{-1}|g(n)| \leq |f(n)| \leq c|g(n)| \qquad (3)$$

para todo $n \geq N$.

Informalmente la ecuación (3) se traduce como $|g(x)|$ acota a $|f(x)|$ salvo constantes cuando $n \geq N$.

4. Proporcione la definición de $f(n) = o(g(n))$. Explique su significado.

Solución. Dadas dos funciones $f,\ g : R \to R$ se dice que $f(n) = o(g(n))$ si y sólo si

$$\lim_{n \to \infty} \left(\frac{f(n)}{g(n)}\right) = 0$$

Podemos decir que el valor de $f(n)$ es insignificante en comparación con $g(n)$ a medida que n se vuelve muy grande.

5. Demuestre el criterio del límite: Si $\lim_{n \to \infty} \left(\frac{f(n)}{g(n)}\right) = a$ con $a \neq 0,\ \pm \infty$, entonces $f(n) = \Theta(g(n))$.

Solución. Primero probaremos $f(n) = O(g(n))$.

Si $\lim_{n \to \infty} \left(\frac{f(n)}{g(n)}\right) = a$ con $a \neq 0, \pm \infty$ por definición de límite para todo $\epsilon > 0$ existe $N > 0$ tal que para todo $n \geq N$ se tiene que

$$\begin{aligned}\left|\frac{f(n)}{g(n)} - a\right| < \epsilon \Leftrightarrow a - \epsilon < \frac{f(n)}{g(n)} &< a + \epsilon,\\ \Leftrightarrow -|a| - \epsilon < \frac{f(n)}{g(n)} &< |a| + \epsilon,\\ \Leftrightarrow \left|\frac{f(n)}{g(n)}\right| &< |a| + \epsilon,\\ \Leftrightarrow |f(n)| &< (|a| + \epsilon)|g(n)|\end{aligned}$$

Sean $\epsilon = 1$, $c_1 = |a| + \epsilon = |a| + 1 > 0$ y la $N > 0$ del inicio de esta prueba, entonces para todo $n \geq N$ se tiene

$$|f(n)| \leq c_1 |g(n)| \tag{4}$$

como se quería.

Luego, como $a \neq 0$ y por propiedades del límite se tiene que $\lim_{n \to \infty} \left(\frac{g(n)}{f(n)}\right) = \frac{1}{a} \neq 0,\ \pm \infty$ y, por nuestra discusión anterior, si y sólo si

$$|g(n)| \leq c_2 |f(n)| \tag{5}$$

es decir, $g(n) = O(f(n))$. por un razonamiento análogo al que sigue de la ecuación (1) en el ejercicio 3, de las ecuaciones (4) y (5) concluimos que si $\lim_{n \to \infty} \left(\frac{f(n)}{g(n)}\right) = a \neq 0,\ \pm \infty$, entonces $f(n) = \Theta(g(n))$

.

6. Ordene las siguientes funciones de acuerdo con la relación "$\cdot = o(\cdot)$": $log(n)^3$, $log(n)^2$, 2^n, n^n, n^3, $log(n)^n$, $nlog(n)$, n^2, $n^2log(n)^3$, $n^{log(n)}$, 3^n, $log(n)$, $n^2log(n)^4$, $n^4log(n)^2$.

Solución.

$$log(n) \ll log^2(n) \ll \sqrt{n} \ll n \ll nlog(n) \ll n^{log_2(3)} \ll n^2 \ll n^3 \ll \left(\frac{1+\sqrt{5}}{2}\right)^n \ll 2^n \ll 3^n \ll n! \ll n^n$$

7. Suponga que $f(n) = O(h(n))$, $g(n) = O(h(n))$, $h(n) = O(r(n))$, $f(n) = o(s(n))$. ¿Cuáles de las siguientes afirmaciones son ciertas? Explique.

(a) $f(n) = g(n)$.

Tenemos que *f(n) = O(h(n)) && g(n) = O(h(n))*, esto implica que ambos están acotados por arriba por h(n), pero no hay otro dato que nos diga de qué forma crecen, pues puede ser cualquier valor igual o menor a h(n), por lo que no necesariamente crecen de la misma manera, la afirmación es falsa.

(b) $f(n) = O(g(n))$.

Sabemos que *f(n) && g(n)* acotadas por *h(n)*, y basándonos en la explicación anterior, no sabemos si asintóticamente *f(n)* crece igual, menos o más que *g(n)* por lo que no se puede afirmar que se cumpla. Falso.

(c) $f(n) = \Theta(g(n))$.

f(n) && g(n) están acotados por *h(n)* como vimos en los puntos anteriores, pero el mismo caso, no tenemos más información como una cota inferior que nos diga que estén acotados por una misma función *j(n)*, lo cual no nos da información que confirme que sean del mismo órden, por lo tanto, es la afirmación es falsa.

(d) $f(n) = O(r(n))$.

Sabemos que *f(n) = O(h(n)) && h(n) = O(r(n))*, esto significa que *f(n)* está acotada por arriba por *h(n) y h(n)* está acotada por arriba por *r(n)*, por lo tanto *f(n) = O(r(n))*, pues está acotada también por arriba por *r(n)*, es una afirmación verdadera.

(e) $f(n) = O(s(n))$.

Sabemos que *f(n) = o(s(n))*, y que el significado de * =*o(*)* se puede interpretar como que *f(n)*<<*s(n)*, significa que el crecimiento de *f(n)* es despreciable comparado a *s(n)*, por lo cual, la cota superior es real, es una afirmación verdadera.

(f) $f(n) = \Theta(s(n))$.

Tenemos que *f(n) = o(s(n))*, lo que significa que *f(n)* crece de manera despreciable respecto a *s(n)*, por lo que la afirmación es falsa.

8. Suponga que $f(n) = O(n)$ y $g(n) = O(n^2)$ ¿Es posible que $g(n) < f(n)$ para toda n? Explique

Solución. Sea $f(n) = n$ con $n \in N$, entonces $|f(n)| \leq |n|$ para toda $n \in N$, por lo tanto $f(n) = O(n)$. También sea $g(n) = 0.5$ la función constante 0.5, entonces $|g(n)| = 0.5 \leq |n^2|$ para toda $n \in N$, por lo tanto $g(n) = O(n^2)$. Además, $g(n) = 0.5 < n = f(n)$ para toda $n \in N$. Hemos exhibido dos funciones $f(n) = O(n)$ y $g(n) = O(n^2)$ para las cuales es posible que $g(n) < f(n)$ para toda n.

9. Suponga que $f(n) = \Theta(n)$ y $g(n) = \Theta(n^2)$ ¿Es posible que $g(n) < f(n)$ para toda n? Explique.

Solución. Como $f(n) = \Theta(n)$, existen $N_1, c_1 > 0$ tales que para todo $n \geq N_1$ se cumple

$$f(n) \leq c_1 n. \tag{6}$$

También, $g(n) = \Theta(n^2)$, así, existen $N_2, c_2 > 0$ tales que para todo $n \geq N_2$ se satisface

$$c_2 n^2 \leq g(n). \tag{7}$$

Como N no es acotado, existe $N_3 \in N$ tal que para todo $n \geq N_3$,

$$c_1 \leq c_2 n. \tag{8}$$

(De lo contrario $\frac{c_1}{c_2} \geq n$ para todo $n \geq N_3$ lo cual sería una contradicción con que N no es acotado).
Tomemos $N = máx\{N_1, N_2, N_3\}$, así, para todo $n \geq N$ de las desigualdades (6), (7) y (8) se sigue el siguiente razonamiento:

$$f(n) \leqq c_1 < c_2 n < c_2 n^2 \leqq g(n)\,. \tag{9}$$

Concluimos que si $f(n) = \Theta(n)$ y $g(n) = \Theta(n^2)$ entonces no es posible que $g(n) < f(n)$ para toda n.

10. Suponga que dos algoritmos *A* y *B*, tienen tiempo de ejecución $T_A(n) = \Theta(n)$ y $T_B(n) = \Theta(n^2)$. ¿Cuáles de las siguientes afirmaciones son ciertas? Explique.

(a) *A* es mejor que *B* siempre. No siempre, depende del comportamiento de las funciones de tiempo $T_A(n)$ y $T_B(n)$. Por ejemplo, si $T_A(n)$=n+k y $T_B(n) = n^2$, para k=90, entonces $T_A(n) \leqq T_B(n)$ solo para $n \geqq 10$ como en la figura.

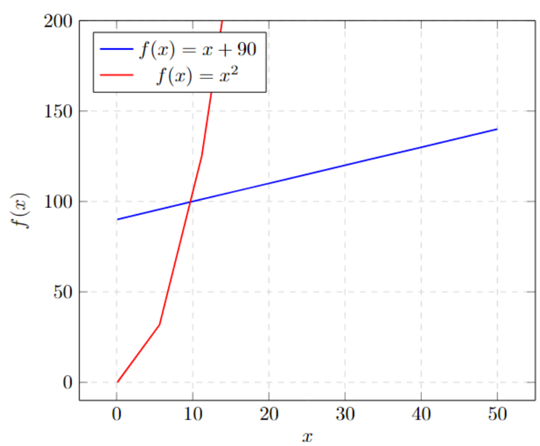

Figure 1: Gráfica de las funciones $f(x) = x + 90$ y $f(x) = x^2$

(b) *A* es mejor que *B* a la larga. *Del razonamiento del ejercicio 9 en la ecuación (9), podemos* *afirmar que existe* $N > 0$ *tal que* $T_A(n) < T_B(n)$ *para toda* $n \geqq N$ *y, por lo tanto, el algoritmo A es mejor que el B a la larga.*

(c) *A* es mejor que *B* en la práctica. *Debido a su menor complejidad lineal frente a la cuadrática de B. Siempre es importante considerar el contexto y los detalles específicos de los algoritmos, ya que en casos excepcionales (por ejemplo, para valores muy pequeños de* n *o debido a optimizaciones particulares), el algoritmo con peor complejidad teórica podría rendir mejor. Sin embargo, de manera general y práctica, el algoritmo A sería la opción preferible.*

11. Suponga que dos algoritmos *A* y *B*, tienen tiempo de ejecución $T_A(n) = \Theta(n)$ y $T_B(n) = \Theta(2^n)$. ¿Cuáles de las siguientes afirmaciones son ciertas? Explique.

(a) *A* es mejor que *B* siempre. *Para cualquier valor de n incluso n<1, B siempre será mayor que A por lo que podemos decir que esto es verdadero.*

(b) *A* es mejor que *B* a la larga. *Esta afirmación es cierta ya que nos dicen que es exactamente del orden de, por lo que A es mejor B en el punto en que se intersectan, ya que en ese punto B crece demasiado mientras que A se mantiene constante. Ya que también sabemos que el exponencial siempre será "malo".*

(c) *A* es mejor que *B* en la práctica. *Esto se debe a que el algoritmo A tiene una complejidad temporal lineal, lo que significa que su tiempo de ejecución aumenta proporcionalmente con el tamaño de la entrada n. haciéndolo eficiente incluso para valores grandes de n. Por otro lado, el algoritmo B tiene una complejidad temporal exponencial, lo que provoca que su tiempo de ejecución crezca muy rápidamente y se vuelva impracticable incluso para valores moderados de n. Por lo tanto, en aplicaciones prácticas, el algoritmo A es significativamente más eficiente y preferible que el algoritmo B.*