16: Little-o Little-Omega

Friday, 27 August 2021 22:01

Definition 1.7 [Little "oh"] The function $f(n) = o(g(n))$ (read as "f is little oh of g of n") iff

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

Example 1.14 The function $3n + 2 = o(n^2)$ since $\lim_{n\to\infty} \frac{3n+2}{n^2} = 0$. $2 = o(n \log n)$. $3n + 2 = o(n \log\log n)$. $6 * 2^n + n^2 = o(3^n)$. $6 * 2^n +$ $o(2^n \log n)$. $3n + 2 \neq o(n)$. $6 * 2^n + n^2 \neq o(2^n)$.

$$\lim_{n\to\infty} \frac{3n+2}{n^2} = \lim_{n\to\infty}\left(\frac{3}{n} + \right.$$

$$= 0$$

Definition 1.8 [Little omega] The function $f(n) = \omega(g(n))$ (read as "f of n is little omega of g of n") iff

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

Example 1.15 Let us reexamine the time complexity analyses of the previous section. For the algorithm Sum (Algorithm 1.6) we determined that $t_{\mathsf{Sum}}(n) = 2n + 3$. So, $t_{\mathsf{Sum}}(n) = \Theta(n)$. For Algorithm 1.7, $t_{\mathsf{RSum}}(n) = 2n + 2 = \Theta(n)$.