Order

Definitions

1. $Big - O$

   For a given complexity function $f(n)$, $O(f(n))$ is the set of complexity functions $g(n)$ for which there exists some positive real constant c and some nonnegative integer n_0 such that for all $n \geq n_0$,

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

2. Ω (Omega)

   For a given complexity function $f(n)$, $\Omega(f(n))$ is the set of complexity functions $g(n)$ for which there exists some positive real constant c and some nonnegative integer n_0 such that for all $n \geq n_0$,

   $$g(n) \geq c \times f(n)$$

3. Θ (Theta)

   For a given complexity function $f(n)$,

   $$\Theta(f(n)) = O(f(n)) \cap \Omega(f(n))$$

   This means that $\Theta(f(n))$ is the set of complexity functions $g(n)$ for which there exists some positive real constants c_1 and c_2 and some nonnegative integer n_0 such that for all $n \geq n_0$,

   $$c_1 \times f(n) \leq g(n) \leq c_2 \times f(n)$$

4. $Little - o$

   For a given complexity function $f(n)$, $o(f(n))$ is the set of complexity functions $g(n)$ satisfying the following: For every positive real constant c, there exists a nonnegative integer n_0 such that for all $n \geq n_0$,

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

5. $Little - \omega$

   For a given complexity function $f(n)$, $\omega(f(n))$ is the set of complexity functions $g(n)$ satisfying the following: For every positive real constant c, there exists a nonnegative integer n_0 such that for all $n \geq n_0$,

   $$g(n) \geq c \times f(n)$$

Properties of Order

1. Transitivity
   - If $f(n) \in O(g(n))$ and $g(n) \in O(h(n))$ then $f(n) \in O(h(n))$.
   - If $f(n) \in \Omega(g(n))$ and $g(n) \in \Omega(h(n))$ then $f(n) \in \Omega(h(n))$.
   - If $f(n) \in \Theta(g(n))$ and $g(n) \in \Theta(h(n))$ then $f(n) \in \Theta(h(n))$.
   - If $f(n) \in o(g(n))$ and $g(n) \in o(h(n))$ then $f(n) \in o(h(n))$.
   - If $f(n) \in \omega(g(n))$ and $g(n) \in \omega(h(n))$ then $f(n) \in \omega(h(n))$.

2. Reflexitivity
   - $f(n) \in O(f(n))$.
   - $f(n) \in \Omega(f(n))$.
   - $f(n) \in \Theta(f(n))$.

3. Symmetry
   - $f(n) \in \Theta(g(n))$ iff $g(n) \in \Theta(f(n))$.
   - $f(n) \in O(g(n))$ iff $g(n) \in \Omega(f(n))$.
   - $f(n) \in o(g(n))$ iff $g(n) \in \omega(f(n))$.

4. If $b > 1$ and $a > 1$, then

   $$\log_a n \in \Theta(\log_b n)$$

   This implies that all logarithmic complexity functions are in the same complexity category. We will represent this category by $\Theta(\lg n)$.

5. If $b > a > 0$, then

   $$a^n \in o(b^n)$$

   This implies that all exponential complexity functions are not in the same complexity category.

6. For all $a > 0$

   $$a^n \in o(n!)$$

   This implies that $n!$ is *worse* than any exponential complexity function.

7. Consider the following ordering of complexity categories:

   $$\Theta(1) \quad \Theta(\log^* n) \quad \Theta(\lg n) \quad \Theta(n) \quad \Theta(n \lg n) \quad \Theta(n^2) \quad \Theta(n^j) \quad \Theta(n^k) \quad \Theta(a^n) \quad \Theta(b^n) \quad \Theta(n!) \quad \Theta(n^n)$$

   where $k > j > 2$ and $b > a > 1$. If a complexity function $g(n)$ is in a category that is to the left of the category containing $f(n)$, then

   $$g(n) \in o(f(n))$$

8. If $c \geq 0$, $d \geq 0$, $f_1(n) \in O(g(n))$, and $f_2(n) \in \Theta(g(n))$, then

$$c \cdot f_1(n) + d \cdot f_2(n) \in \Theta(g(n))$$

9. If $f_1(n) \in O(g(n))$ and $f_2(n) \in O(h(n))$, then

$$f_1(n) + f_2(n) \in O(g(n) + h(n))$$

   Less formally, this means $f_1(n) + f_2(n) \in \max\left[O(g(n)), O(h(n))\right]$.

10. If $f_1(n) \in O(g(n))$ and $f_2(n) \in O(h(n))$, then

$$f_1(n) * f_2(n) \in O(g(n) * h(n))$$

11. If $f(n)$ is a polynomial of degree k, then

$$f(n) \in \Theta(n^k)$$

12. For any constant k

$$\lg^k n \in O(n)$$

13. Limits can also be used to determine order.

$$\lim_{n\to\infty} \frac{g(n)}{f(n)} = \begin{cases} c & \text{implies } g(n) \in \Theta(f(n)) \text{ if } c > 0 \\ 0 & \text{implies } g(n) \in o(f(n)) \\ \infty & \text{implies } f(n) \in o(g(n)) \end{cases}$$