Como seguramente ha sido observado, la definición de la clase de funciones acotadas superiormente por otra guarda cierta semejanza con la definición de límite. En efecto, la regla que a continuación se enuncia establece esta relación.

Proposición 10 (Regla del Límite) Sean f_1 y f_2 dos funciones de comportamiento.

1. Si $\lim_{n\to\infty} f_1(n)/f_2(n) = 0$ entonces $f_1(n) \in O(f_2(n))$ y $f_2(n) \notin O(f_1(n))$.
2. Si $\lim_{n\to\infty} f_1(n)/f_2(n) = \infty$ entonces $f_2(n) \in O(f_1(n))$ y $f_1(n) \notin O(f_2(n))$.
3. Si $\lim_{n\to\infty} f_1(n)/f_2(n) > 0$ entonces $f_1(n) \in O(f_2(n))$ y $f_2(n) \in O(f_1(n))$.

La demostración de esta proposición es directa a partir de las definiciones. Por ejemplo: $|\frac{f_1(n)}{f_2(n)}| < \delta$ para toda $n > N_\delta$ conduce a $f_1(n) < \delta f_2(n)$, i.e. $f_1(n) \in O(f_2(n))$. Además suponiendo que existieran $c > 0$ y $N \in \mathbb{N}$ tales que $f_2(n) \leq cf_1(n)$ entonces $\frac{f_1(n)}{f_2(n)} \geq \frac{1}{c}$, para toda $n \geq N$, lo cual contradiría que $\lim_{n\to\infty} f_1(n)/f_2(n) = 0$.

Observación 6

1. De acuerdo con el ejercicio 10.2, $\lim_{n\to\infty} \frac{\log n}{\sqrt{n}} = 0$, podemos concluir que $\log n \in O(\sqrt{n})$ y $\sqrt{n} \notin O(\log n)$.
2. Partiendo de que $n \in O(2^{\lfloor \log n \rfloor})$ y $2^{\lfloor \log n \rfloor} \in O(n)$ podría suponerse que $\lim_{n\to\infty} 2^{\lfloor \log n \rfloor}/n$ converge, pero no es así: cada vez que n es una potencia de dos se cumple $2^{\lfloor \log n \rfloor}/n = 1$ pero cuando n crece $2^{\lfloor \log n \rfloor}/n$ oscila. En conclusión, el recíproco de la regla del límite no es cierto en general.

Ejercicio 19

1. Use la proposición 5 para demostrar el segundo inciso de la regla del límite.
2. Demuestre que $n \in O(2^{\lfloor \log n \rfloor})$ y $2^{\lfloor \log n \rfloor} \in O(n)$.
3. ¿Cuál algoritmo sería más rápido, uno con comportamiento $f(n) = (\log\log n)^2$ u otro con comportamiento $g(n) = \log n$? Justifique su respuesta.

3.2. Cota inferior

Así como se estableció que un algoritmo no puede ser más tardado que otro, podemos hacer algo para decidir que un algoritmo no puede ser más veloz que otro, lo cual parece sencillo pues esta idea maneja la negación de lo que antes hemos visto. La clase de funciones que están *acotadas inferiormente* por f es:

$$\Omega(f(n)) = \{g : \mathbb{N} \to \mathbb{R} | (\exists d > 0, N \in \mathbb{N})(\forall n \geq N)(g(n) \geq df(n))\}.$$

La relación entre O y Ω está dada por la proposición que sigue.

Proposición 11 (Regla de la Dualidad) Sean f_1 y f_2 funciones de comportamiento, entonces $f_2(n) \in \Omega(f_1(n))$ sii $f_1(n) \in O(f_2(n))$.

Ejercicio 20

1. Demuestre la regla de la dualidad.
2. Demuestre o refute las siguientes afirmaciones:
   a) $n^3 \in \Omega(n^2)$.
   b) $2^{n+1} \in \Omega(2^n)$.
   c) $n! \in \Omega((n+1)!)$.
3. Tome en cuenta la observación 5 para demostrar que la relación de pertenencia a $\Omega(f(n))$ es una relación de orden parcial.

La interpretación de Ω contrasta con la de O: Cuando f_2 mide el peor comportamiento y $f_2 \in O(f_1(n))$, entonces en cualquier ejemplar f_2 será a lo más tan lento como f_1. En tanto, afirmar $f_2 \in \Omega(f_1(n))$ significa que f_2 no puede mejorar a f_1 (en el caso peor), pero es posible que haya ejemplares en los que f_2 mejore a f_1 (casos mejores).

3.3. Clase de equivalencia

Recordemos que el principio de invarianza establece que cualesquier implantaciones de un mismo algoritmo con comportamientos $f_1(n)$ y $f_2(n)$ satisfacen que $f_1(n) \leq cf_2(n)$ y $df_2(n) \leq f_1(n)$ para ciertas constantes $c, d > 0$ y para toda $n \geq N$. Desde entonces hemos considerado que los comportamientos "equivalentes" satisfacen las desigualdades mencionadas. Este hecho aclara el uso de $g(n) \in O(f(n))$ y $g(n) \in \Omega(f(n))$, al afirmar que $g(n)$ no puede ser más lenta que $f(n)$ y $g(n)$ no puede ser más rápida que $f(n)$, la equivalencia de comportamientos es entonces cuando $g(n)$ no puede ser más lenta ni más rápida que $f(n)$. Tenemos así que la clase de funciones *equivalentes* a f es:

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

O bien $\Theta(f(n)) = \{g(n) | (\exists c, d > 0, N \in \mathbb{N})(\forall n \geq N)(df(n) \leq g(n) \leq cf(n))\}$

Observación 7

1. Análogas a las reglas del umbral y el máximo para O, son válidas para Θ:

   Regla del Umbral para Θ: Sea f estrictamente positiva. $g(n) \in \Theta(f(n))$ si y sólo si existen $c, d > 0$ tal que $df(n) \leq g(n) \leq cf(n)$, para toda $n \in \mathbb{N}$.

   Regla del Máximo para Θ: $\Theta(f_1(n) + f_2(n)) = \Theta(\text{máx}\{f_1(n), f_2(n)\})$.

2. Con ligeros cambios respecto de O tenemos la regla del límite para Θ:

   Regla del Límite para Θ: Sea $\lim_{n\to\infty} f_1(n)/f_2(n) = a$,
   a) si $a > 0$ entonces $f_1(n) \in \Theta(f_2(n))$.
   b) si $a = 0$ entonces $f_1(n) \in O(f_2(n))$ y $f_1(n) \notin \Theta(f_2(n))$.
   c) Si $a = \infty$ entonces $f_1(n) \in \Omega(f_2(n))$ y $f_1(n) \notin \Theta(f_2(n))$.

Como esperaríamos Θ divide el conjunto de funciones en clases las cuales corresponden a los puntos de la "escala de funciones". Específicamente tenemos la proposición que sigue.

Proposición 12 La relación entre funciones de comportamiento $f(n) \sim g(n)$ sii $f(n) \in \Theta(g(n))$ es una relación de equivalencia.

Tal y como hemos planteado intuitivamente en una sección anterior podemos concebir una escala que permita dar respuesta a una pregunta fundamental del análisis de algoritmos ¿cuál algoritmo es mejor? Se ha dicho que la respuesta es relativa a la situación en la que será aplicado. Dentro del enfoque asitótico la respuesta está dada por comparar las funciones. En virtud de la observación 5 y la proposición 12 distinguimos a las funciones constantes $\Theta(1)$, funciones logarítmicas $\Theta(\log n)$, funciones lineales $\Theta(n)$, funciones cuadráticas $\Theta(n^2)$, etc. como los puntos de la escala. Por otro lado, la relación $f(n) \lessdot g(n)$ sii $f(n) \in O(g(n))$ es una relación de orden y podríamos decir que un algoritmo con función de comportamiento $f(n)$ sería mejor que otro con función $g(n)$ si $\hat{f}(n) \lessdot \hat{g}(n))$ donde $f(n) \in \Theta(\hat{f}(n))$ y $g(n) \in \Theta(\hat{g}(n))$. Finalmente, hemos de precisar que la escala no es tal pues la relación $\lessdot$ es parcial. Esto es, no es cierto que cualesquier par de funciones sea comparable. Tenemos, más bien, una jerarquía de clases de funciones, y la escala es solamente una aproximación útil a esta estructura.

Ejercicio 21