Section 5.6 Comparing Rates of Growth

We often need to compare functions f and g to see whether $f(n)$ and $g(n)$ are about the same or whether one grows faster as n increases. The functions might represent running times of algorithms that we need to compare. They might also be approximations with closed forms are easier to evaluate. To compare functions we need to give precise meanings to the phrases "about the same" and "grows faster."

Big Oh. The *growth rate of f is bounded above by the growth rate of g* if there are constants $d > 0$ and m such that

$$|f(n)| \leq d|g(n)| \text{ for } n \geq m$$

If this is the case we write $f(n) = \mathrm{O}(g(n))$ and say, "$f(n)$ is big oh of $g(n)$."

Examples/Quizzes. $f(n) = \mathrm{O}(f(n))$, $200n = \mathrm{O}((1/100)n)$, $n = \mathrm{O}(n^2)$

Big Omega. The *growth rate of f is bounded below by the growth rate of g* if there are constants $c > 0$ and m such that

$$c|g(n)| \leq |f(n)| \text{ for } n \geq m$$

If this is the case we write $f(n) = \Omega(g(n))$ and say, "$f(n)$ is big omega of $g(n)$."

Examples/Quizzes. $f(n) = \Omega(f(n))$, $(1/100)n = \Omega(200n)$, $n^2 = \Omega(n)$

Big Theta. *f has the same growth rate as g* if there are constants $c > 0$, $d > 0$ and m such that

$c|g(n)| \leq |f(n)| \leq d|g(n)|$ for $n \geq m$ (or, if $g(n) \neq 0$ then $c \leq |f(n)/g(n)| \leq d$ for $n \geq m$)

If this is the case we write $f(n) = \Theta(g(n))$ and say, "$f(n)$ is big theta of $g(n)$."

Examples.

1. $\lfloor n \rfloor = \Theta(\lceil n \rceil))$. *Proof*: $(1/2)\lceil n \rceil \leq \lfloor n \rfloor \leq 1 \cdot \lceil n \rceil$ for $n \geq 1$. QED.
2. $n(n + 1)/2 = \Theta(n^2)$. *Proof*: $(1/2) \cdot n^2 \leq n(n + 1)/2 \leq 2 \cdot n^2$ for $n \geq 0$. QED.

Theorem. If $\lim_{n \to \infty} \frac{f(n)}{g(n)} = c$ and $c \neq 0$ and $c \neq \infty$, then $f(n) = \Theta(g(n))$.

Example. $\lim_{n \to \infty} \frac{n(n+1)/2}{n^2} = \lim_{n \to \infty} \frac{n+1}{2n} = \lim_{n \to \infty} \frac{1+(1/n)}{2} = \lim_{n \to \infty} \frac{1}{2} = \frac{1}{2}$. So $n(n + 1)/2 = \Theta(n^2)$.

Example. The converse of the theorem is false. For example, let $f(n) = (1 + n \bmod 2)n^2$ and let $g(n) = n^2$. Then $f(n) = \Theta(g(n))$ by letting $c = 1$, $d = 2$, and $m = 0$. But the following limit does not exist:

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \lim_{n \to \infty} \frac{(1+n \bmod 2)n^2}{n^2} = \lim_{n \to \infty} (1+n \bmod 2).$$

Quiz. Show that $\log(n + 1) = \Theta(\log n)$.

Solution (using the definition). Since $n < n + 1 < n^2$ for $n \geq 2$ it follows that

$\log n < \log (n + 1) < \log n^2 = 2\log n$ for $n \geq 2$. So let $c = 1$, $d = 2$, and $m = 2$.

Therefore, $\log(n + 1) = \Theta(\log n)$. QED.

Solution (using the theorem).

$$\lim_{n\to\infty} \frac{\log(n+1)}{\log n} = \lim_{n\to\infty} \frac{1/(n+1)}{1/n} \qquad \text{(use base } e \text{ and L'Hospital's rule)}$$

$$= \lim_{n\to\infty} \frac{n}{n+1} = \lim_{n\to\infty} \frac{1}{1+(1/n)} = 1.$$

Therefore, $\log(n + 1) = \Theta(\log n)$. QED.

Little oh. *f has lower growth rate than g* if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0.$

If this is the case we write $f(n) = o(g(n))$ and say, "$f(n)$ is little oh of $g(n)$."

Example. $$\lim_{n\to\infty} \frac{\log(\log n)}{\log n} = \lim_{n\to\infty} \frac{(1/\log n)(1/n)}{1/n} \qquad \text{(use base } e \text{ and L'Hospital's rule)}$$

$$= \lim_{n\to\infty} \frac{1}{\log n} = 0.$$

Therefore, $\log(\log n) = o(\log n)$.

Example/Quiz. Show that if $f(n) = \mathrm{o}(g(n))$, then $f(n) = \mathrm{O}(g(n))$.

Proof: Since $f(n) = \mathrm{o}(g(n))$, it follows that $\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$.

So for any $\varepsilon > 0$ there is an m such that $|f(n)/g(n)| < \varepsilon$ for $n \geq m$. In other words,

$$|f(n)| \leq \varepsilon |g(n)| \text{ for } n \geq m.$$

So $f(n) = \mathrm{O}(g(n))$. QED.

Example. Let f and g be defined by

$f(n) =$ *if* n is odd *then* 1 *else* n
$g(n) = n$.

Show that $f(n) = \mathrm{O}(g(n))$, but $f(n) \neq \mathrm{o}(g(n))$ and $f(n) \neq \Theta(g(n))$.

Proof: We have $f(n) \leq g(n)$ for $n \geq 1$. So $f(n) = \mathrm{O}(g(n))$. The limit as n approaches infinity of $f(n)/g(n)$ does not exist. So $f(n) \neq \mathrm{o}(g(n))$. Assume, BWOC, that $f(n) = \Theta(g(n))$. Then there are constants $c > 0$, $d > 0$ and m such that

$$c|g(n)| \leq |f(n)| \leq d|g(n)| \text{ for } n \geq m$$

So $c|n| \leq |1|$ if n is odd and $n \geq m$, which is a contradiction. So $f(n) \neq \Theta(g(n))$. QED.

Example/Quiz. Show that if $f(n) = \mathrm{o}(g(n))$, then $g(n) = \Omega(f(n))$.

Proof: Since $f(n) = \mathrm{o}(g(n))$, it follows that $\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$.

So for any $\varepsilon > 0$ there is an m such that $|f(n)/g(n)| < \varepsilon$ for $n \geq m$. In other words,

$$|f(n)| \leq \varepsilon|g(n)| \text{ for } n \geq m.$$

So $(1/\varepsilon)|f(n)| \leq |g(n)|$ for $n \geq m$. So $g(n) = \Omega(f(n))$. QED.

Example/Quiz. $f(n) = \Omega(g(n))$ iff $g(n) = \mathrm{O}(f(n))$.

Proof: $c|g(n)| \leq |f(n)|$ for $n \geq m$ iff $|g(n)| \leq (1/c)|f(n)|$ for $n \geq m$. QED.

Example. We saw earlier that a divide and conquer algorithm that splits a problem of size $n = b^k$ into b smaller problems of size n/b, and does this recursively, uses

$$a_n = tn + tn\log_b n$$

operations, where tn is the number of operations needed to split up the problem of size n and put a solution back together. We can now observe that $a_n = \Theta(n\log_b n)$,