Honors Project 12: The Order of Functions

Objective

In certain applications in science and engineering, the behavior of a function $f = f(x)$ for large values of x ($x > N$ for some N) or small values of x ($|x| < \epsilon$ for some $\epsilon > 0$) in comparison to the behavior of another function $g = g(x)$ is important. In this project we discuss the notation and classification of functions that are used in studying such behavior.

Narrative

The terms "large" and "small" are relative. To illustrate, type the command lines below into Maple in the order in which they are listed; they produce a table of values for $f(x) = x^2$, $g(x) = 2^x$, and $h(x) = x^x$. Observe that while 2^x and x^x are both larger than x^2 for large x, x^x is considerably larger than either x^2 or 2^x.

```
> restart:
> M := matrix(20,4,(Row,Col)->0):
> for N from 1 to 20 do
    M[N,1] := N: M[N,2] := N^2:  M[N,3] := 2^N: M[N,4] := N^N: od:
> eval(M);
```

Here is another example aimed at illustrating why this is important in a specific application: Suppose we have several different algorithms for implementing the same task, and that we must repeat this task many times. (The task might involve alphabetically ordering the names in a telephone book, for example.) To simplify things, we assume that neither space nor resources are a limitation, and hence there is no issue surrounding the scalability of our task. Depending on the algorithm we use, the time T it takes to complete the task N times may vary directly with N, with $N \lg N$, with N^2 or with N^3. "What difference does this make?" Well, suppose it takes 1 second to perform the task when $N = 10^6$ using the algorithm in which T varies directly with N: $T = kN$. Then (since $1 = 10^6 k$), it follows that $k = 10^{-6}$. Since we are assuming our task is scalable, the constant of proportionality for the algorithm in which T varies directly with $N \lg N$ is also $k = 10^{-6}$ — that is, $T = kN \lg N = 10^{-6} N \lg N$ — so using this algorithm it would take

$$T(10^6) = 10^{-6} 10^6 \lg 10^6 = 6 \lg 10 = 19.9 \text{ seconds.}$$

Using the algorithm in which T varies directly with N^2 it would take 11.5 days, and using the algorithm in which T varies directly with N^3 it would take 31,709.8 years! Thus, someone who has to choose between one of several algorithms for performing a task will (or should) be very concerned about the order of the execution time functions of the algorithms.

One way to classasify the complexity of an algorithm involves "big oh" notation: The function f is said to be "big oh" of the function g, and we write $f \in O(g)$, or $f \preceq g$, if

$$\lim_{x \to \infty} \frac{f(x)}{g(x)} \quad \text{is a finite real constant.} \tag{1}$$

Thus, for example,

$$x^2 \in O(x^3) \quad \text{since} \quad \lim_{x \to \infty} \frac{x^2}{x^3} = 0.$$

And

$$x \preceq x \lg x \preceq x^2 \preceq x^3.$$

Note that L'Hopital's Rule is very useful in computing limits such as (1)!

Task

1. Show that if $m \leq n$ then $x^m \preceq x^n$.

2. Show that, for every function $f = f(x)$,

   a) $f \preceq f$,

   b) $f \preceq kf$ for every non-zero positive constant k, and

   c) $kf \preceq f$ for every non-zero positive constant k.

3. Show that if $f \preceq g$ and $g \preceq h$ then $f \preceq h$.

4. Show that $f(x) < g(x)$ for all $x > N$ for some real number N, implies that $f \preceq g$, but that $f \preceq g$ does not imply that $f(x) < g(x)$ for all $x > N$ for some real number N.

5. List the functions below from lowest to highest order. If two (or more) are of the same order, indicate this.

$$
\begin{array}{llll}
x & x^2 & x^3 & 4x^3 + 3x^2 + 2x \\
2^x & e^x & 2^{x-1} & x^x \\
\ln x & \lg x & x^2 + \lg x & (\lg x)^2 \\
\lg \lg x & x \lg x & \sqrt{x} & x^x + 2^x
\end{array}
$$

Comments

There are other ways to classify the behavior of a function $f = f(x)$ relative to another function $g = g(x)$ for large values of x. For example, some work uses $f \in \Theta(g)$ if

$$\lim_{x \to \infty} \frac{f(x)}{g(x)} \quad \text{is a finite } \textit{non-zero} \text{ real constant}$$

while other work uses $f \in \Omega(g)$ if

$$\lim_{x \to \infty} \frac{g(x)}{f(x)} \quad \text{is a finite real constant.}$$

On the other hand, there are also many ways to classify the behavior of a function $f = f(x)$ relative to another function $g = g(x)$ for large values of x ($|x| < \epsilon$ for some $\epsilon > 0$). For example, in writing

$$e^x = 1 + x + O(x^2)$$

we are saying that e^x can be written as $1 + x$ plus a function that can be written as $x^2 h(x)$ where $h(x)$ is continuous in a neighborhood of $x = 0$. More generally, $f \in O(g)$ if

$$\lim_{x \to 0} \frac{f(x)}{g(x)} \quad \text{is a finite real constant.} \tag{2}$$

Note that while the same notation $O(g)$ is used for both (1) and (2), (1) and (2) actually define different classes of functions. (The fact that the notation is the same is an unfortunate conflict in usage.) For example, if $f(x) = x^2$ and $g(x) = x^3$ then $f \in O(g)$ using (1) but $f \notin O(g)$ using (2), and $g \in O(f)$ using (2) but $g \notin O(f)$ using (1).