Lab 4 - The Growth of Function
Computer Science 1FC3

Author: Dai Tri Man Lê
ledt@mcmaster.ca

Abstract

The purpose of this lab is to help you understand with the *"growth of function"* by reviewing the notions of Big-O, Big-Omega and Big-Theta. Some examples and exercises on these concepts will be discussed in details. Through the use of Maple, we also demonstrate an example where Big-O notation is used to characterize the residual term of a truncated asymptotic series.

1 Big-O (Upper Bound) Notation

Big-O notation is a mathematical notation used to describe the asymptotic behavior of functions. In other words, it is used to estimate the growth of a function in terms of another, usually simpler, function. The functions used in this estimation often include the following:

$$1, \log(n), n, n\log(n), n^2, 2^n, n!$$

(See Figure 3 page 138 in Rosen for the plot of these functions)

It has two main areas of application:

- In computer science, it is useful in the analysis of the complexity of algorithms (the number of operations an algorithm uses as its input grows). (This will not be discussed in this lab but the following lab!)
- In mathematics, it is usually used to characterize the residual term of a truncated infinite series, especially an asymptotic series.

Formally, Big-O notation is defined as following:

Definition 1:

A function $f(x)$ is $O(g(x))$ iff there exist a constant C and a constant k such that, for every $x > k$, $|f(x)| \leq C|g(x)|$. **(End of Definition.)**

Technically, $O(g(x))$ is a a set (also called a *family of functions*):

$$O(g(x)) = \{f(x) \mid \exists C, \exists k, \forall x \geq k, |f(x)| \leq C|g(x)|\}$$

Therefore, from a set theoretical point of view, we have:

$$O(1) \subset O(\log(n)) \subset O(n) \subset O(n\log(n)) \subset O(n^2) \subset O(2^n) \subset O(n!)$$

Remark. Big-O gives us a formal way of expressing *asymptotic upper bounds*, a way of bounding from above the growth of a function. Knowing where a function falls within the big-Oh hierarchy allows us to compare it quickly with other functions, which gives us an idea of which algorithm has the best time performance as you will see in the next lab. **(End of Remark.)**

Example 1:

Use the definition of big-O (Definition 1) to prove that

$$f(x) = 5x^4 - 37x^3 + 13x - 4 = O(x^4)$$

Solution.
We must find integers C and k such that $5x^4 - 37x^3 + 13x - 4 \leq C|x^4|$ for all $x \geq k$. We can proceed as follows:

$$|5x^4 - 37x^3 + 13x - 4| \leq |5x^4 + 37x^3 + 13x + 4| \leq |5x^4 + 37x^4 + 13x^4 + 4x^4| = 59|x^4|$$

where the first inequality is satisfied if $x \geq 0$ and the second inequality is satisfied if $x \geq 1$.
Therefore, $|5x^4 - 37x^3 + 13x - 4| \leq 59|x^4|$ if $x \geq 1$, so we have $C = 59$ and $k = 1$. **(End of Solution.)**

Remark. *The solution we have given is by no means the only possible one.* Here is a second solution. It makes the value C smaller, but requires us to makes the value of k larger:

$$|5x^4 - 37x^3 + 13x - 4| \leq |5x^4 + 37x^3 + 13x + 4| \leq |5x^4 + 4x^4 + x^4 + x^4| = 11|x^4|$$

In the first inequality we changed from subtraction to addition of the second and fourth terms. In the second inequality we replaced the term $37x^3$ by $4x^4$ (which is valid if $x \geq 10$), replaced $13x$ by x^3 (which is valid if $x \geq 4$) and replaced 4 by x^4 (which is valid if $x \geq 2$). Therefore, $|5x^4 - 37x^3 + 13x - 4| \leq 11|x^4|$ if $x \geq 10$. Hence we can use C= 11 and k=10. **(End of Remark.)**

Example 2:

Give a big-O estimate for each of these functions. Use a simple function in your big-O estimate.
(a) $3n + n^3 + 4$
(b) $1 + 2 + 3 + \cdots + n + 3n^2$
(c) $\log_{10}(2^n) + 10^{10}n^2$

Solution.

(a) Since $3n + n^3 + 4 \leq 3n^3 + n^3 + 4n^3 = 8n^3$ for $n > 1$, we have $3n + n^3 + 4 = O(n^3)$.

(b) We have $1+2+3+\cdots+n \leq n+n+n+\cdots+n = n \cdot n. = n^2$. Hence, $1+2+3+\cdots+n+3n^2 \leq n^2 + 3n^2 = 4n^2$. Therefore $1 + 2 + 3 + \cdots + n + 3n^2 = O(n^2)$.

(c) $\log_{10}(2^n) + 10^{10}n^2 = n\log_{10} 2 + 10^{10}n^2 \leq n^2 \log_{10} 2 + 10^{10}n^2 \leq (\log_{10} 2 + 10^{10})n^2$ if $n \geq 1$. But $\log_{10} 2 + 10^{10}$ is a constant. Therefore $\log_{10}(2^n) + 10^{10}n^2 = O(n^2)$.

(End of Solution.)

Example 3:

Suppose you wish to prove that $f(x) = 2x^2 + 5x + 9$ is big-O of $g(x) = x^2$ and want to use $C = 3$ in the big-O definition. Find a value k such that $|f(x)| \leq 3|g(x)|$ for all $x > k$.

Solution.
We need a value k such that $|2x^2 + 5x + 9| \leq 3x^2$ for all $x > k$. Begin by grouping the three terms in $2x^2 + 5x + 9$ as $2x^2 + (5x + 9)$. If $5x + 9 \leq x^2$ we have $|2x^2 + 5x + 9| \leq 2x^2 + x^2 = 3x^2$.

By solving the inequality $5x + 9 \leq x^2$, we find that if $x \geq 7$, then $5x + 9 \leq x^2$. Hence $|2x^2 + 5x + 9| \leq 2x^2 + x^2 = 3x^2$ if $x \geq 7$. **(End of Solution.)**

Remark. We can solve the inequality $5x + 9 \leq x^2$ using the Maple command

```
[> solve(5*x+9 <= x^2,x);
```

(End of Remark.)

Example 4:

Use the definition of big-O to prove that $\frac{3x^4-2x}{5x-1} = O(x^3)$.

Solution.
We must find positive integers C and k such that for all $x \geq k$,

$$\left|\frac{3x^4 - 2x}{5x - 1}\right| \leq C|x^3|$$

To make the fraction $\left|\frac{3x^4-2x}{5x-1}\right|$ larger, we can do two things: make the numerator larger and make the denominator smaller:

$$\left|\frac{3x^4 - 2x}{5x - 1}\right| \leq \left|\frac{3x^4}{5x - 1}\right| \leq \left|\frac{3x^4}{5x - x}\right| = \left|\frac{3x^4}{4x}\right| = \frac{3}{4}|x^3|$$

In the first step we made the numerator larger (by not subtracting $2x$) and in the second step we made the denominator smaller by subtracting x, not 1. Note that the first inequality requires $x \geq 0$ and the second inequality requires $x \geq 1$.

Therefore, if $x \geq 1$, $\left|\frac{3x^4-2x}{5x-1}\right| \leq \frac{3}{4}|x^3|$, and hence $\left|\frac{3x^4-2x}{5x-1}\right| = O(x^3)$. **(End of Solution.)**

Remark. Although used extensively to describe the growth of functions, big-O notation provides a *weak upper bound*. For example, $f(x) = x^2 + 7$ may be said to be $O(x^2)$, but it may also be said to be $O(x^3)$ or $O(e^x)$. As a result, we need the notions of big-theta and big-omega as you will see next. **(End of Remark.)**

2 Big-Omega (Lower Bound) And Big-Theta (Tight Bound) Notations

Change "$\leq$" to "$\geq$" and force $C > 0$ in Definition 1, we have the **big-omega notation** (Ω). As a result, Ω gives an *asymptotic lower bound*.

Definition 2:

A function $f(x)$ is $\Omega(g(x))$ iff there exist a constant $C > 0$ and a constant k such that, for every $x > k$, $|f(x)| \geq C|g(x)|$. **(End of Definition.)**

Neither O or Ω are completely satisfying; we would like a tight bound on how quickly our function grows. To say it does not grow any faster than something does not tell us how slowly it grows, and vice-versa. So we need something to give us a tigher bound; something that bounds a function from both above and below. We can combine O and Ω to give us **big-theta notation** (Θ). Although theoretically the same idea, the definition of Θ notation will be stated somewhat differently from one from Rosen (Definition 3 page 141):

Definition 3:

A function $f(x)$ is $\Theta(g(x))$ iff there exist constants $C_1 > 0$ and C_2 and a constant k such that, $\forall x > k, C_1|g(x)| \leq |f(x)| \leq C_2|g(x)|$. **(End of Definition.)**

As we can see in Definition 3, big-theta notation is a stricter version of big-oh and big-omega notations, because:

$$f(x) \text{ is } \Theta(g(x)) \text{ if and only if } f(x) \text{ is both } O(g(x)) \text{ and } \Omega(g(x)).$$

which is equivalent to saying that

$$\Theta(g(x)) = O(g(x)) \cap \Omega(g(x))$$

Example 5:

Use the definition of big-theta to prove that $7x^2 + 1 = \Theta(x^2)$.

Solution.
Clearly, we have $7x^2 + 1 = \Omega(x^2)$, we now need to show $7x^2 + 1 = O(x^2)$.
We have $7x^2 \le 7x^2 + 1 \le 7x^2 + x^2 \le 8x^2$ (where we need $x \ge 1$ to obtain the second inequality). Therefore, if $x \ge 1$, $7x^2 \le 7x^2 + 1 \le 8x^2$. This says that $7x^2 + 1 = \Theta(x^2)$. **(End of Solution.)**

3 Big-O notation and asymptotic series (Optional)

In mathematics an asymptotic expansion, asymptotic series (also called Poincaré expansion) is a formal series of functions which has the property that truncating the series after a finite number of terms provides an approximation to a given function as the argument of the function tends towards a particular, often infinite, point.

In Maple, the function $asympt$ computes the asymptotic series of a function with respect to some variable, where the truncation order can be specified by users as an optional argument. For more information on $asympt$, use the following Maple command.

```
[>  ?asympt
```

We will now consider some examples.

The following command

```
[>  S:=asympt(x/(1-x-x^2),x);
```

will produce

$$S := -x^{-1} + x^{-2} - 2\,x^{-3} + 3\,x^{-4} - 5\,x^{-5} + O\left(x^{-6}\right)$$

The Maple output means that the evaluation of a function $f(x) = x/(1 - x - x^2)$ and the evaluation of $-x^{-1} + x^{-2} - 2\,x^{-3} + 3\,x^{-4} - 5\,x^{-5}$ at some point a differs in some error approximately $\le C|a|^{-6}$. But why? (Hint: Consider $|f(x) - S(x)|$.)

Similarly, the following command (with truncation order 1 specified as one of the argument)

```
[>P:=  asympt((3*x^4-2*x)/(5*x-1),x,1);
```

will produce

$$P := 3/5\,x^3 + \frac{3}{25}\,x^2 + \frac{3}{125}\,x - \frac{247}{625} + O\left(x^{-1}\right)$$

4 Exercises

1. Give a big-O estimate for each of these functions. Use a simple function in your big-O estimate. Give some justification.

   (a) $\frac{n}{10000000000000000} + 999999999999999999$
   (b) $\log(n^2 + 1)$
   (c) $\sqrt{n^2 + 1}$
   (d) $(n^2 + 1)(n \log(n) + 3)$
   (e) 10^{10}
   (f) $(n + 3)/(n + 1)$
   (g) $(n^3 + 1)/(n + 1)$
   (h) $2^{\log(n)\cdot 3} + n^3 + 4$
   (i) $\frac{n!}{999999999} + 9999999999 \cdot 2^n$
   (j) $\log_{10}(2^n) + 10^{10}n^2$

2. What are the big-Omega and big-Theta of the functions in Exercise 1 (a) - (e)?

3. Use the definition of big-O (Definition 1) to prove that

   (a) $x^3 - 37x^{5/2} + 13x - 4 = O(x^3)$
   (b) $x \log(x) + x^2 = O(x^2)$
   (c) $5n \log(n) + n^{3/2} = O(n^2)$ (Harder)
   (d) $n \log(n^8) + n^{3/2} = O(n^{3/2})$ (VERY HARD! - Homework)