Paper Review: Big Omicron and Big Omega and Big Theta

Submitted by: Rajib Dey

Summary

Algorithms are a big part of Computer Science. To find a more efficient algorithm we need to use different kinds of analysis on them. Asymptotic analysis is one of the most useful ones. The research paper titled "Big Omicron and Big Omega and Big Theta" by renowned author and scientist Donald E. Knuth shows us how to properly use the big omicron (O) (also called the big O), big omega (Ω), and big theta (Θ) notations while doing algorithm analysis. This research paper is written as a letter to the editor of SIGACT News to address the fact that some researchers are actually misusing the O-Notation as the lower bounds of an algorithm's time complexity while the Ω-notation can do it perfectly. Even though the author knew that some researchers are using the Ω notation to denote lower bounds [1]. He also wanted to address the origin of the definition of big Omega(Ω). Which turned out to be an arduous work for the author where he had to spent many hours in the library sifting through books.

The author mentions that according to Landau [2], the first appearance of O was spotted in Bachman's 1894 book [3]. After which it is being used widely. The author then finds that Hardy and Littlewood's classic 1914 memoir [4] might be the probable origin of the Ω-notation. Here, Ω-notation is defined as a function whose absolute value will exceed C(f(n)), where $n \to \infty$ and C is a constant with small positive value. This notation has no viable use in Computer science as computers work with non-infinite numbers. To fix that, the author then defined the three notations so that it can be used in computer science:

1. $O(f(n))$ denotes the set of all functions $g(n)$ such that there exist a positive constant C and n_0 with $\mid g(n) \mid \leq Cf(n), \forall n \geq n_0$.

2. $\Omega(f(n))$ denotes the set of all functions $g(n)$ such that there exist a positive constant C and n_0 with $g(n) \geq Cf(n), \forall n \geq n_0$.

3. $\Theta(f(n))$ denotes the set of all functions $g(n)$ such that there exist a positive constants C, $C^{'}$ and n_0 with $Cf(n) \leq g(n) \leq C^{'}f(n), \forall n \geq n_0$.

We can infer from the above definitions that, O(f(n)) denotes the worst time complexity. While Ω(f(n)) denotes the best time complexity and $\Theta(f(n))$ denotes the exact time complexity. These notations are valid only when $n \to \infty$; $n \to$0 and for all n $\geq$n$_0$. The notations here does not denote mathematical equality. According to the author they represent "one-way equalities".

Three Strengths

1. This paper corrects the research community's mistake of misusing the big-O notation.

2. It is easier to use than other notations. As O is widely used in English language and have more mnemonic significance than other notations.

3. This paper shows us why we should be using the asymptotic notations while doing algorithm analysis using examples. By far it has stood the test of time.

Three Weaknesses

1. In some rare cases, if the bounds can not be expressed by the asymptotic notations then relational notations would have to be used.

2. If the value of constant C is very large then asymptotic notations can not be used.

3. These notations are used for asymptotic analysis and use large values on n. But small values of n it might not make sense.

Future Research opportunities

Using the above mentioned asymptotic notations we can analyze our algorithm in many branches of computer science. Which should lead to more and more efficient algorithms.

References

[1] K. Prarchar, *Primazahlverteilung.* Berlin: Springer, 1957.

[2] E. Landau, *Handbuch der Lehre yon der Verteilung der Primzahlen*, vol. 2. 1909.

[3] P. Bachmann, *Die Analytische Zahlentheorie. Zahlentheorie*, vol. 2. 1894.

[4] G. H. Hardy and J. E. Littlewood, *Some problems of Diophantine approximation*, vol. 37. Acta Mathematica, 1914.