Quantum Computation 2017

Sheet 1 Abgabe 21/11/2017

Asymptotic notation

In lecture we introduced 'big O' notation in the study of asymptotic behaviour. Related concepts are 'big Ω' and 'big Θ' notation. You can find discussion of these in Nielsen and Chuang section 3.2.1.

The definitions are as follows.

- $f(n) = O(g(n))$ if there exist finite C and n_0 such that

$$f(n) \leq Cg(n), \quad \forall\, n > n_0.$$

- $f(n) = \Omega(g(n))$ if there exist finite C and n_0 such that

$$f(n) \geq Cg(n), \quad \forall\, n > n_0$$

- $f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

Now solve the following problems.

(a) If $f(n) = O(g(n))$ then $g(n) = \Omega(f(n))$.

(b) If $f(n)$ is a polynomial of degree k, show that $f(n) = O(n^l)$ for any $l \geq k$.

(c) If $a(n) = O(g(n))$ and $b(n) = O(h(n))$, show that $a(n)b(n) = O(g(n)h(n))$.

(d) Show that $e^{\alpha n} = O(e^{\beta n})$ and $e^{\beta n} = \Omega(e^{\alpha n})$ if $\alpha < \beta$.

(e) Show that, for arbitrary finite k, $n^k = O(n^{\log n})$ but $n^{\log n} \neq O(n^k)$

(f) Show that $n^k \log n = O(n^{k+\epsilon})$ for any non-zero ϵ.