COSC 31 Ungraded Problem Set 2

John Berry

April 9, 2021

Made-up Problems

$f(n) = O(g(n))$

a) $\sqrt{f(n)} \stackrel{?}{=} O(\sqrt{g(n)})$

$$\begin{aligned}
\sqrt{f(n)} &\le c\sqrt{g(n)} && \text{by definition of big-O} \\
\left(\sqrt{f(n)}\right)^2 &\le \left(c\sqrt{g(n)}\right)^2 \\
f(n) &\le c^2 g(n) \\
f(n) &\le c_1 g(n) && c^2 = c_1 \\
f(n) &= O(g(n)) && \text{by definition of big-O}
\end{aligned}$$

b) $\sqrt{f(n)} \stackrel{?}{=} O(\log_2 g(n))$

$$\begin{aligned}
&\sqrt{f(n)} \le c \log_2 g(n) && \text{by definition of big-O} \\
&\sqrt{f(n)} \le \log_2 g(n)^c && \text{properties of logarithms} \\
&2^{\sqrt{f(n)}} \le 2^{\log_2 g(n)^c} \\
&2^{\sqrt{f(n)}} \le g(n)^c
\end{aligned}$$

Since exponential functions grow faster than polynomial ones. $\nexists c : \sqrt{f(n)} \le c \log_2 g(n)$
So, $\sqrt{f(n)} \ne O(\log_2 g(n))$

c) $\sqrt{f(n)} \stackrel{?}{=} O(2^{g(n)})$

$$\begin{aligned}
f(n) &\le cg(n) && \text{by definition of big-O} \\
\log_2(f(n)) &\le cg(n) && \text{by definition of big-O} \\
\log_2 \sqrt{f(n)} &\le \log_2 c2^{g(n)} \\
\frac{1}{2}\log_2 f(n) &\le g(n) + \log_2 c
\end{aligned}$$

UG2-1

Formal definition of big-O notation:
$\exists n_0 > 0\ \exists c > 0 \mid \forall n > n_0\ \ f(n) \leq cg(n)$

Formal definition of big-Ω notation:
$\exists n_0 > 0\ \exists c > 0 \mid \forall n > n_0\ \ f(n) \geq cg(n)$

a)

If $f(n) = O(g(n))$ then $f(n) \leq c_1(g(n))$

If $g(n) = O(h(n))$ then $g(n) \leq c_2(h(n))$

Then $f(n) \leq c_1(g(n)) \leq c_2(h(n))$

So, by the transitive property of inequalities:

$f(n) \leq c_2(h(n))$

This gives $f(n) = O(h(n))$

b)

c) $n^{\sqrt{(n)}} = \Theta(2^n)$

To prove Θ it is necessary to prove O and Ω.

$$
\begin{aligned}
n^{\sqrt{n}} &\leq c2^n && \text{by definition of big-O} \\
\log_2(n^{\sqrt{n}}) &\leq \log_2(c2^n) && \\
\sqrt{n}\log_2 n &\leq n + \log_2 c && \text{properties of logarithms} \\
\sqrt{n}\log_2 n - n &\leq \log_2 c &&
\end{aligned}
$$

The constant $log_2 c$ cannot be greater than $\sqrt{n}\log_2 n - n$ for all $n > n_0$

So $n^{\sqrt{(n)}} \neq \Theta(2^n)$

d) $(\log_2 n)^{\log_2 n} = \Theta(n^{100})$

Thoughts: $(\log_2 n)$ grows much slower than n so n^100 will quickly outpace $(\log_2 n)^{\log_2 n}$. Because of this, you could probably prove O but Ω would be invalid.

How to prove? Perhap assume that $(\log_2 n)^{\log_2 n} = \Omega(n^{100})$ is true

UG2-2

a)

By the definition of big-O:

$\log_b n \leq cn^a$

$n \leq b^{cn^a}$

Given that polynomial functions grow slower than exponential functions then $\log_b n = O(n^a)$

b) $(n + a)^b = \Theta(n^b)$

By the definition of big-O:

$(n + a)^b \leq cn^b$

Using the binomial theorem:

$(n + a)^b = n^b + abn^{b-1} + \cdots + a^{b-1}bn + a^b$

$n^b + abn^{b-1} + \cdots + a^{b-1}bn + a^b \leq cn^b$

$n^b - \{|a_{k-1}|n^{k-1} + \ldots |a_1|n + |a_0|\} \leq cn^b$

$n^b - \{|a_{k-1}| + \dots |a_1| + |a_0|\}n^{k-1} \leq cn^b$

Dropping the low order terms:

$n^b \leq cn^b$ so $(n+a)^b = O(n^b)$

By the definition of big-Ω:

$c(n+a)^b \geq n^b$

Using the binomial theorem:

$c(n+a)^b = c(n^b + abn^{b-1} + \cdots + a^{b-1}bn + a^b)$

$c(n^b + abn^{b-1} + \cdots + a^{b-1}bn + a^b) \geq cn^b$

$cn^b + c\{|a_{k-1}|n^{k-1} + \dots |a_1|n + |a_0|\} \geq cn^b$

$cn^b + c\{|a_{k-1}| + \dots |a_1| + |a_0|\}n^{k-1} \geq cn^b$

$cn^b \geq n^b$ so $(n+a)^b = \Omega(n^b)$

Since $(n+a)^b = \Omega(n^b)$ and $(n+a)^b = O(n^b)$ then $(n+a)^b = \Theta(n^b)$