Prof. Friedrich Eisenbrand
Alfonso Cevallos Discussions from: February 17, 2015

Computer Algebra

Spring 2015

Assignment Sheet 1

Note: These are just notes and not necessarily full solutions for each exercise. Please report any mistakes you may find.

Exercise 1

The following list shows the functions in "ascending order": If f is shown to the left of g, then $f = O(g)$. If they are separated by a semicolon, then $f = o(g)$; and if they are separated by a comma, then $f = \Theta(g)$.

$$13;\ \log(n^{1337}),\ \log n;\ \log^2 n;\ \sqrt{n};\ 2^{3+\log n},\ 3n;\ \log n^n;\ e^{\log n};\ 2^{4\log n};\ n^6 - 5n^2;\ 2^{\log^2 n};\ 2^n;\ 4^n.$$

Proofs are left to the reader. The function $-n^6 + 5n^2$ is strictly negative for $n \geq 2$, hence definitions of the big O notation do not apply.

Exercise 2

Let $f, g : \mathbb{N} \to \mathbb{R}_+$. To show: $f = O(g)$ if and only if $\limsup_{n\to\infty} \frac{f(n)}{g(n)} < \infty$.

Let $f = O(g)$. By definition, there is a $c > 0$ and an $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$ we have $f(n) \leq cg(n)$. This means that $\frac{f(n)}{g(n)} \leq c$ for all $n \geq n_0$, hence $\sup_{n\geq n_0} \frac{f(n)}{g(n)} \leq c$. As the lim sup can only decrease from this value, we get $\limsup_{n\to\infty} \frac{f(n)}{g(n)} \leq c < \infty$.

Now suppose $\limsup_{n\to\infty} \frac{f(n)}{g(n)} = c < \infty$. Clearly $c \geq 0$, since f and g are strictly positive functions. By definition, this means that for all $\varepsilon > 0$ there exists an $n_0 \in \mathbb{N}$ such that for all $n_1 \geq n_0$ one has $|\sup\left\{\frac{f(n)}{g(n)} \mid n \geq n_1\right\} - c| \leq \varepsilon$. In particular, $\sup\left\{\frac{f(n)}{g(n)} \mid n \geq n_0\right\} \leq c + \varepsilon$. Let us choose $\varepsilon = 1$. Then it follows that for all $n \geq n_0$ we have $\frac{f(n)}{g(n)} \leq c + 1$, i.e. $f(n) \leq (c+1)g(n)$ and thus $f = O(g)$.

Exercise 3

(a) Let $f, g : \mathbb{N} \to \mathbb{R}_+$ with $f \sim g$. This implies that $\forall \epsilon > 0$, there exists $N \in \mathbb{N}$ such that $|\frac{f(n)}{g(n)} - 1| \leq \epsilon$ for all $n \geq N$. Choosing $\epsilon = 1$, we deduce that $f(n) \leq 2g(n)$ for each $n \geq N$. Since $\sim$ is clearly symmetric, $g(n) \leq 2f(n)$ for $n \geq N'$, where N' is an appropriate natural number. Hence $f = \theta(g)$. The converse is in general false, take e.g. $f(n) = n$ and $g(n) = 2n$.

(b) Let $f, g: \mathbb{N} \to \mathbb{R}_+$ with $f \sim g$. We need to show that for each $\epsilon > 0$ there exists $N \in \mathbb{N}$ such that $(1-\epsilon)g(n) \le f(n) \le (1+\epsilon)g(n)$ for all $n \ge N$. Pick $\epsilon > 0$. Since $f \sim g$, for each $\delta > 0$ there exists $N' \in \mathbb{N}$ such that $|\frac{f(n)}{g(n)} - 1| \le \delta$, that is, $(1-\delta)g(n) \le f(n) \le (1+\delta)g(n)$. Hence, we can take $\delta = \epsilon$ and obtain $N = N'$. The converse is also true, as can be seen by reversing the argument above.

(c) Let $f, g: \mathbb{N} \to \mathbb{R}_+$ with $f \sim g$. This implies that for each $\epsilon > 0$ there exists $N(\epsilon) \in \mathbb{N}$ such that $|f(n) - g(n)| \le \epsilon g(n)$ for all $n \ge N(\epsilon)$. We need to show that, for each $\delta > 0$, there exists $N' \in \mathbb{N}$ such that $|\frac{F(n)}{G(n)} - 1| \le \delta$ for each $n \ge N'$. Fix $\delta > 0$. Let $N := N(\frac{\delta}{3})$ and choose $N' \ge N$ such that $G(N), F(N) \le \frac{\delta}{3} G(n)$ for all $n \ge N'$. Note that such an N' exists since $G(N), F(N)$ are constants and $G(n) \to \infty$ when $n \to \infty$. Then for all $n \ge N'$, we have:

$$\begin{aligned} |\tfrac{F(n)}{G(n)} - 1| &= |\tfrac{F(N)}{G(n)} + \tfrac{\sum_{i=N+1}^{n}(f(i)-g(i))}{G(n)} - \tfrac{G(N)}{G(n)}| &\le |\tfrac{F(N)}{G(n)}| + |\tfrac{\sum_{i=N+1}^{n}(f(i)-g(i))}{G(n)}| + |\tfrac{G(N)}{G(n)}| \\ &\le \tfrac{\delta}{3} + \tfrac{\delta}{3} + \tfrac{\delta}{3} &\le \delta, \end{aligned}$$

as required.

Some students tried to prove part (c) without using the assumption $G(n) \to \infty$ as $n \to \infty$. Without this condition, the statement is in general false. Analyse the example $g(n) = 2^{-n}$; $f(1) = 100$, $f(n) = 2^{-n}$ for $n \ge 2$.

Exercise 4

$n \log n = \log n^n \ge \log(n!)$ for all $n \in \mathbb{N}$, so $\log(n!) = O(n \log n)$.
Conversely, we have

$$\begin{aligned} \log(n!) &= \log(n \cdot (n-1) \cdot \dots \cdot 1) &= \textstyle\sum_{i=2}^{n} \log i &\ge \textstyle\sum_{i=2}^{n} \int_{i-1}^{i} \log x dx &= \int_1^n \log x dx \\ &= n \log n - n \log e + 1 &= \theta(\log n). \end{aligned}$$

Hence $\log(n!) \in \Omega(\log n)$, which implies that all three relations are true.

Exercise 5

Let us call $s_1(n) = T_1(n) \cdot 10^{-10}$ and $s_2(n) = T_2(n) \cdot 10^{-6}$ the number of seconds to process an input of size n in the first and second scenarios, respectively. One easily sees that s_1 starts out smaller but grows faster than s_2. So where is the point where s_1 and s_2 reach parity?

$$s_1(n) = s_2(n) \implies 5 \cdot 10^{-10} n^2 = 1000 \cdot 10^{-6} n \log n \implies \frac{n}{\log n} = 2 \cdot 10^6$$

There is no elementary closed form solution to such an equation, so we have to do it numerically using e.g. Python. We obtain that the first scenario is faster until a problem size of about $5.1 \cdot 10^7$, or 51 millions, at which point both settings take a little more than 15 days to run.