Exercise sheet 1
Advanced Algorithms

Master Données et Systèmes Connectés

Master Machine Learning and Data Mining

Master Cyber-Physical Social Systems

Amaury Habrard

amaury.habrard@univ-st-etienne.fr

Laboratoire Hubert Curien, UMR CNRS 5516
Université Jean Monnet Saint-Étienne

amaury.habrard@univ-st-etienne.fr

Semester 1

Exercise 1

Since f is $O(g)$, we have for some constants $c > 0$ and $n_0 \geq 0$ that:

$$f(n) \leq c \times g(n).$$

Dividing both sides by c, we get that $g(n) \geq \frac{1}{c} f(n)$ for all $n \geq n_0$.
So we have for all $n \geq n_0$ the existence of a constant $c' = 1/c$ such that

$$g(n) \geq c' \times f(n),$$

this implies that $g = \Omega(f)$.

Exercise 1 (ctd) - item 1

Since f is $O(g)$, there exists constants $c > 0$ and $n_0 \geq 0$ that:
$f(n) \leq c \times g(n)$.

Since g is $O(h)$, there exists constants $c' > 0$ and $n'_0 \geq 0$ that:
$g(n) \leq c' \times h(n)$.

Now when $n \geq \max(n_0, n'_0)$, we have:
$f(n) \leq c \times g(n) \leq c \times c' \times h(n)$

thus f is $O(h)$ using constants $n''_0 = \max(n_0, n'_0)$ and $c'' = c \times c'$.

The proofs for other claims are similar,

Exercise 1 (ctd) - item 2

Since f is $\Omega(g)$, there exists constants $c > 0$ and $n_0 \geq 0$ that:
$f(n) \geq c \times g(n)$.

Since g is $\Omega(h)$, there exists constants $c' > 0$ and $n'_0 \geq 0$ that:
$g(n) \geq c' \times h(n)$.

Now when $n \geq \max(n_0, n'_0)$, we have:
$f(n) \geq c \times g(n) \geq c \times c' \times h(n)$

thus f is $\Omega(h)$ using constants $n''_0 = \max(n_0, n'_0)$ and $c'' = c \times c'$.

The 3rd case with θ (item 3), is deduced by the two preceding cases

Exercise 1 (ctd) - item 4

Since f is $O(h)$, there exists constants $c > 0$ and $n_0 \geq 0$ that:
$f(n) \leq c \times h(n)$.

Since g is $O(h)$, there exists constants $c' > 0$ and $n'_0 \geq 0$ that:
$g(n) \leq c' \times h(n)$.

Now when $n \geq \max(n_0, n'_0)$, we have:
$f(n) + g(n) \leq c \times h(n) + c' \times h(n) = (c + c') \times h(n)$

thus $f + g$ is $O(h)$ using constants $n''_0 = \max(n_0, n'_0)$ and $c'' = c + c'$.

Exercise 1 (ctd) - item 5

Since g is $O(f)$, there exists constants $c > 0$ and $n_0 \geq 0$ that:
$g(n) \leq c \times f(n)$.

Then when $n \geq n_0$, we have:
$f(n) + g(n) \leq f(n) + c \times f(n) = (1 + c) \times f(n)$
thus $f + g$ is $O(f)$ using constants $n_0'' = n_0$ and $c'' = 1 + c$.

Also, we have $n \geq n_0$, we have:
$f(n) + g(n) \geq f(n)$, since $g(n)$ is a positive function,
thus $f + g$ is $\Omega(f)$ using constants $n_0''' = n_0$ and $c''' = 1$.

$f + g$ is both $O(f)$ and $\Omega(f)$, thus $f + g$ is $\theta(f)$.

Exercise 2.1

This is false in general since it could be that $g(n) = 1$ for all n, $f(n) = 2$ for all n, and then $\log_2 g(n) = 0$, whence we cannot write $\log_2 f(n) \leq c \log_2 g(n)$.
On the other hand, if we simply require $g(n) \geq 2$ for all n beyond some n_1, then the statement holds. Since $f = O(g)$ implies there exists[1] $c > 1, n_0 \geq 0$ s.t. $f(n) \leq c \cdot g(n)$ for all $n \geq n_0$, we have

$$\begin{aligned}
\log_2 f(n) &\leq \log_2(cg(n)) \\
&\leq \log_2(c) + \log_2(g(n)) \text{ by a property of } \log \\
&\leq \log_2(c) \times \log_2(g(n)) + \log_2 g(n) = (\log_2(c) + 1)(\log_2(g(n)))
\end{aligned}$$

once $n \geq max(n_0, n_1)$ since $\log_2 g(n) \geq 1$ beyond this point.
We have then found an index $n_0' = max(n_0, n_1)$ and a constant $c' = \log_2 c + 1$ allowing us to conclude that $\log_2(f) = O(\log_2(g))$.

[1]Note: we must also ensure $\log_2(c) \geq 0$ here, but we can assume $c > 1$ for $O(\cdot)$ relationships without loss of generality since even if a constant between 0 and 1 is considered at first, then a constant greater than one works as well.

Exercise 2.2

This is false: take $f(n) = 2n$ and $g(n) = n$.
Then $2^{f(n)} = 4^n$ while $2^{g(n)} = 2^n$.
If you assume that $2^f = O(2^g)$, then there exists $c > 0$ and $n_0 \geq 0$ s.t. for every $n \geq n_0$: $4^n \leq c2^n$.
This implies that $2^n \leq c$ (or in other words $n \leq \log_2(c)$) which means that it cannot be true for every n greater than n_0, thus the claim is false.

Exercise 2.3

This is true. Since $f(n) \leq cg(n)$ for all $n \geq n_0$, we have $(f(n))^2 \leq c^2(g(n))^2$ for all $n \geq n_0$ (you just have to write the definition properly)

Exercise 3

First, f_1, f_2, f_4 are easy (they belong to classic functions: exponential, polynomial, logarithm): $f_4 = O(f_2)$ and $f_2 = O(f_1)$.

Now for f_3, it starts to be smaller than 10^n but once $n \geq 10$, then clearly $10^n \leq n^n$. This is exactly what we need for the definition of $O(\cdot)$ (take $c = 1$ and $n_0 = 10$), thus $f_1 = O(f_3)$.

Now f_5 is a bit more complex. The solution here is to take logarithms to make things clearer. Here $\log_2(f_5(n)) = \sqrt{\log_2 n} = (\log_2 n)^{1/2}$.
For the other functions we have $\log_2(f_4(n)) = \log_2(\log_2(n))$ while $\log_2(f_2(n)) = \frac{1}{3}\log_2(n)$.
Let $z = \log_2 n$, we can see these as <u>functions of z</u>: $\log_2(f_2(n)) = \frac{1}{3}z$, $\log_2 f_4(n) = \log_2(z)$ and $\log_2(f_5(n)) = z^{1/2}$.

Exercise 3 (ctd)

Now it is easier to see what is going on. First, let's compare f_4 and f_5.

For $z \geq 16$, we have $\log_2(z) \leq \sqrt{z}$ (you can try to do a simple analysis of the function $f(z) = \ln(z)/\ln(2) - \sqrt{z}$, global maximum at $4/\ln(2)^2$)).

But the condition $z \geq 16$ is the same as $n \geq 2^{16} = 65,536$.
Thus once $n \geq 2^{16}$, we have $\log_2 f_4(n) \leq \log_2 f_5(n)$, and so $f_4(n) \leq f_5(n)$ since $\log_2$ is a monotonic increasing function.

Thus we can write $f_4(n) = O(f_5(n))$.

Exercise 3 (ctd)

Similarly we have $z^{1/2} \leq \frac{1}{3}z$ once $z \geq 9$ (you can do a simple analysis of the function $f(z) = z^{1/2} - \frac{1}{3}z$, global maximum at $9/4$)
in other words, once $n \geq 2^9 = 512 = n_0$.

For n above this bound we have $\log_2 f_5(n) \leq log_2 f_2(n)$ and hence $f_5(n) \leq f_2(n)$, and so we can write $f_5(n) = O(f_2(n))$.

Essentially, we have discovered that $2^{\sqrt{\log_2 n}}$ is a function whose growth rate lies somewhere between that of logarithms and polynomials.