Combinatorial Analyis	**Fall 2010**
Lecture 10: September 08	
Lecturer: Neil Calkin	*Scribe: Reid Sanders and Grady Thomas*

Disclaimer: *These notes are intended for students in the class listed above: they are not guaranteed to be complete or even necessarily correct. They may only be redistributed with permission, which you may expect will be liberally granted. Ask first, please.*

10.1 Computing values of functions efficiently

Aside: Today we need to work over $\mathbb{Q}$, not over $\mathbb{R}$ - we need division.
How to compute values of $f(x)^n$ assuming they exist? In fact for each of the following functions how do we compute them:

1. $f(x)^n$: f arbitrary, $n \in \mathbb{N}$
2. $f(x)^a : f(0) = 1$
3. $e^{f(x)}, f(0) = 0$
4. $log(1 + f(x)) : f(0) = 0$

10.2 Computing $g(x) = f(x)^n$

Last time we saw that we can compute $f(x)^n$ via iterated squaring:
Compute $f(x)^2, f(x)^{2^2}, f(x)^{2^3}, \ldots$ and multiply appropriate ones to get $f(x)^n$. How long will this take to compute coefficients up to $x^N, N = 2^k$ say?
Assume g(a) is defined over a ring. Multiplication takes time m. Then computing $g(x)^2$ up to x^N requires computing

$$\sum_{l=0}^{j} g_l g_{j-l} \qquad 0 \le j \le N$$

$$n+1 \text{ multiplications} \qquad 1+2+3+...+(N+1) = \binom{N+2}{2} \text{ total multiplications}$$

$$n \text{ additions} \qquad 1+2+3+...+N = \binom{N+1}{2} \text{ total additions}$$

We need to compute $f(x)^{2^j}$ for all j with $2^j \le n$.
So $j \le log_2 n$.
So we need to do $\lfloor log_2 n \rfloor$ squarings. So:

$$2\lfloor log_2 n \rfloor \binom{N+2}{2} \text{ multiplications in } \mathbb{R}$$

$$2\lfloor log_2 n \rfloor \binom{N+1}{2} \text{ additions in } \mathbb{R}$$

For now we will skip the question of the possibility of doing these operations

10.3 Computing $g(x) = e^{f(x)}$

Aside: From here on we are assuming working in $\mathbb{Q}$. How to compute coefficient of $g(x) = e^{f(x)}, f(0) = 0$ for terms up to x^N? Differentiate and multiply by x:

$$x\frac{d}{dx}g(x) = xg'(x) = \sum_{n\geq 0} nx^n g_n = \sum_{n\geq 1} ng_n x^n$$

$$\begin{aligned}
& x\frac{d}{dx}e^{f(x)} \\
&= xf'(x)e^{f(x)} \\
&= xf'(x)g(x) \\
&= \left(\sum_{s\geq 0} sf_n x^n\right)\left(\sum_{r=0} g_r x^r\right)
\end{aligned}$$

So extracting the coefficient:

$$\begin{aligned}
& [x^n]\, g'(x) \\
&= ng_n \\
&= \sum_{s=0} sf_s g_{n-s} \\
&= \sum_{s=1} sf_s g_{n-s} \qquad\qquad g_0, g_1, \ldots, g_{n-1}
\end{aligned}$$

Perhaps we can improve things a little:
Let $g(x) = \sum_{s=0}^{\infty} h_s \frac{x^s}{s!}$ instead. So, $g'(x) = \sum_{s=0}^{\infty} h_{s+1}\frac{x^s}{s!}$
Then, $\frac{h_n}{(n-1)!} = [x^{n-1}]g'(x)$ and, as before, $g'(x) = f'(x)g(x)$.

exercise NTBHI: Find the recurrence for h_n in terms of $h_0, h_1, \ldots, h_{n-1}$ that this gives.

Computing g_n once $g_0, \ldots, g_{n-1}$ are computed takes $2n$ multiplications, $n-1$ additions, and 1 division. So computing g_N takes 2 $\binom{N+1}{2}$ multiplications, $\binom{N}{2}$ additions, and N divisions. So this is quadratic in N.
$g(x) = log(1 + f(x))$:

$$x\frac{d}{dx}log(1 + f(x)) = \frac{xf'(x)}{1 + f(x)} \text{ so, } xg'(x)(1 + f(x)) = xf'(x) \tag{10.1}$$

exercise: Determine the recurrence you get in the form $ng_n = \sum \cdots$. Using this find terms up to x^{10} in $log(cos(x))$. (Use Sage/Maple to compare with Taylor series of $log(cos(x))$)

10.4 Back to $g(x) = f(x)^a$

Differentiating:

$$
\begin{aligned}
&xg'(x) = xaf(x)^{a-1}f'(x) \\
&\text{so } xg'(x)f(x) = xaf'(x)g(x) \\
&\text{so } [x^n]xg'(x)f(x) = [x^n]xaf'(x)g(x) \\
&\sum_{k=0}^{n} kg_k f_{n-k} = \sum_{k=0}^{n} ag_k(n-k)f_{n-k} \\
&ng_n = -\sum_{k=0}^{n-1} kg_k f_{n-k} + \sum_{k=0}^{n-1} a(n-k)g_k f_{n-k} = \sum_{k=0}^{n-1}(a_n - (a+1)k)g_k f_{n-k}
\end{aligned}
$$

exercise: By hand, compute terms up to x^8 of $(1 + 2x + 4x^2 + 7x^3)^8$ and $(1 + 2x + 4x^2 + 7x^3)^7$ using

1. Iterated squaring
2. The method above.