Solution to HW 3, Problem 4.9

Al-Naji, Nader

Problem: If $\alpha < \beta$, show that α^N is exponentially small relative to β^N. For $\beta = 1.2$ and $\alpha = 1.1$, find the absolute and relative errors when $\alpha N + \beta N$ is approximated by βN, for $N = 10$ and $N = 100$. Solution: To show that α^N is exponentially small relative to β^N, we need to show that $\left(\frac{\alpha}{\beta}\right)^N = O\left(\frac{1}{N^M}\right)$ by the definition for what it means for a quantity to be "exponentially small". By the definition of Big-Oh, this implies we need to show that the quotient of these quantities is bounded from above as N tends to infinity. We do this below:

$$
\begin{aligned}
\lim_{N\to\infty} \frac{\left(\frac{\alpha}{\beta}\right)^N}{\frac{1}{N^M}} &= \lim_{N\to\infty} \left(\frac{\alpha}{\beta}\right)^N N^M \\
&= \lim_{N\to\infty} e^{\ln\left(\frac{\alpha}{\beta}\right)^N N^M} && \text{(exp-log trick)} \\
&= \lim_{N\to\infty} e^{N \ln\frac{\alpha}{\beta} + M\ln N} && \text{(simplifying)} \\
&= \lim_{N\to\infty} e^{M\ln N - N} && \left(\text{since } \ln\frac{\alpha}{\beta} < 0\right) \\
&= \lim_{N\to\infty} \frac{M\ln N}{e^N} && \text{(simplifying)} \\
&= 0 && \text{(taking the limit)} \\
\Rightarrow \left(\frac{\alpha}{\beta}\right)^N &= O\left(\frac{1}{N^M}\right) && \text{(def of Big-Oh)}
\end{aligned}
$$

Having shown that $\left(\frac{\alpha}{\beta}\right)^N = O(\frac{1}{N^M})$, we thus conclude that α^N grows exponentially small relative to β^N.

When $\alpha = 1.1$, $\beta = 1.2$, and $N = 10$, we have that the absolute error $= |\beta^N - (\alpha^N + \beta^N)| = |\alpha^N| = 1.1^{10} = 2.5937424601$ with relative error $= \frac{\beta^N}{\alpha^N+\beta^N} - 1 = \frac{1.2^{10}}{1.1^{10}+1.2^{10}} - 1 = -0.2952306294$. When $N = 100$, we have that the absolute error $= 1.1^{10} = 13780.6123398$ with relative error $= \frac{1.2^{100}}{1.1^{100}+1.2^{100}} - 1 = -0.00016636871$.

Solution to HW 3, Problem 4.71

Al-Naji, Nader

Problem: Show that $P(N) = \sum_{0 \leq k \leq N} \frac{(N-k)^k (N-k)!}{N!} = \sqrt{\frac{\pi N}{2}} + O(1)$

Solution:

Lemma 1:
For $k = o(N^{\frac{2}{3}})$, the following relative approximation holds:

$$\frac{(N-k)^k (N-k)!}{N!} = e^{-\frac{k^2}{2N}} \left(1 + O(\frac{k}{N}) + O(\frac{k^3}{N^2})\right)$$

Proof:

$$
\begin{aligned}
\frac{(N-k)^k(N-k)!}{N!} &= \exp(\ln(\frac{(N-k)^k(N-k)!}{N!})) && \text{(exp log trick)}\\
&= \exp\{k\ln(N-k) + \ln((N-k)!) - \ln(N!)\}\\
&= \exp\{k\ln(N-k)\\
&+ (N-k+\frac{1}{2})\ln(N-k) - (N-k) + \ln(\sqrt{2\pi}) + O(\frac{1}{N-k})\\
&- (N+\frac{1}{2})\ln(N) + N - \ln(\sqrt{2\pi}) + O(\frac{1}{N})\} && \text{(Stirling)}\\
&= \exp\{(N+\frac{1}{2})\ln(N(1-\frac{k}{N})) + k - (N+\frac{1}{2})\ln(N)\\
&+ O(\frac{1}{N}) + O(\frac{1}{N-k})\} && \text{(Simplifying)}\\
&= \exp\{(N+\frac{1}{2})\ln(N) + (N+\frac{1}{2})\ln(1-\frac{k}{N}) + k\\
&- (N+\frac{1}{2})\ln(N) + O(\frac{1}{N})\} && \text{(Simplifying)}\\
&= \exp\{(N+\frac{1}{2})\ln(1-\frac{k}{N}) + k + O(\frac{1}{N}) + O(\frac{1}{N-k})\} && \text{(Simplifying)}\\
&= \exp\{(N+\frac{1}{2})(-\frac{k}{N} - \frac{k^2}{2N^2} + O(\frac{k^3}{N^3})) + k + O(\frac{1}{N}) + O(\frac{1}{N-k})\}\\
&\text{(above, we expand: } \ln(1+x) = x - \frac{x^2}{2} + O(\frac{x^3}{3}) \text{ if } x \to 0)\\
&= \exp\{-\frac{k^2}{4N^2} - \frac{k^2}{2N} - \frac{k}{2N} - k + O(\frac{k^3}{N^2}) + O(\frac{k^3}{N^3}) + k\\
&+ O(\frac{1}{N}) + O(\frac{1}{N-k})\} && \text{(Multiplying)}\\
&= \exp\{-\frac{k^2}{2N} + O(\frac{k}{N}) + O(\frac{k^3}{N^2}) + O(\frac{1}{N-k})\}\\
&\text{(We have: } -\frac{k}{2N} + O(\frac{1}{N}) = O(\frac{k}{N})\\
&\text{ and } -\frac{k^2}{4N^2} + O(\frac{k^3}{N^3}) = O(\frac{k^3}{N^2}))\\
O(\frac{1}{N-k}) &= O(\frac{1}{N}\cdot\frac{1}{1-\frac{k}{N}}) && \text{(Pull out N)}\\
&= O(\frac{1}{N}\cdot(1+O(\frac{k}{N})))\\
&\text{(Because } k \le N\text{, we can use the geometric expansion)}\\
&= O(\frac{1}{N}) + O(\frac{k}{N^2})\\
&\text{(Because } O(f(N)g(N)) = O(f(N))O(g(N)))
\end{aligned}
$$

$$
\begin{aligned}
\Rightarrow \frac{(N-k)^k (N-k)!}{N!} &= \exp\{-\frac{k^2}{2N} + O(\frac{k}{N}) + O(\frac{k^3}{N^2}) + O(\frac{1}{N}) + O(\frac{k}{N^2})\} && \text{(Plugging in)} \\
&= \exp\{-\frac{k^2}{2N} + O(\frac{k}{N}) + O(\frac{k^3}{N^2})\} && \text{(Simplifying)} \\
&= \exp\{-\frac{k^2}{2N})\} \cdot \exp\{O(\frac{k}{N}) + O(\frac{k^3}{N^2})\} && \text{(Separating terms)} \\
&= \exp\{-\frac{k^2}{2N})\} \cdot (1 + O(\frac{k}{N}) + O(\frac{k^3}{N^2})) && \text{(Done)}
\end{aligned}
$$

Lemma 2:
For all k, the following absolute approximation holds:

$$\frac{(N-k)^k (N-k)!}{N!} = e^{\frac{-k^2}{2N}} + O(\sqrt{\frac{1}{N}})$$

This proof is identical to the proof of the absolute approximation in theorem 4.4, but we repeat the analysis here for completeness. We first consider the case where $k \leq k_0$ where k_0 is the nearest integer to $N^{\frac{2}{3}}$. The relative approximation from Lemma 1 holds in this case and we have:

$$
\begin{aligned}
\frac{(N-k)^k (N-k)!}{N!} &= e^{-\frac{k^2}{2N}} + e^{-\frac{k^2}{2N}} O(\frac{k}{N}) + e^{-\frac{k^2}{2N}} O(\frac{k^3}{N^2}) \\
&= e^{-\frac{k^2}{2N}} + x e^{-\frac{1}{2}x^2} O(\frac{1}{\sqrt{N}}) + x^3 e^{-\frac{1}{2}x^2} O(\frac{1}{\sqrt{N}}) \qquad (x = \frac{k}{\sqrt{N}}) \\
&= e^{-\frac{k^2}{2N}} + O(\frac{1}{\sqrt{N}})
\end{aligned}
$$

Where the last line holds because $x \geq 0 \Rightarrow x e^{-\frac{1}{2}x^2} = O(1)$ and $x^3 e^{-\frac{1}{2}x^2} = O(1)$. This shows that the lemma clearly holds for $k \leq k_0$. Then, since $e^{\frac{-k^2}{2N}}$ is exponentially small when $k \geq k_0$, and since the coefficients decrease as k increases, we have $\frac{(N-k)^k (N-k)!}{N!} = O(\frac{1}{\sqrt{N}})$. And, by the same logic, we have $e^{\frac{-k^2}{2N}} + O(\sqrt{\frac{1}{N}}) = O(\sqrt{\frac{1}{N}}) = \frac{(N-k)^k (N-k)!}{N!}$ when $k \geq k_0$ and thus the approximation holds for all k.

The rest of the proof uses Lemmas one and two *exactly* as theorem 4.8 uses theorem 4.4 but we repeat the argument here for completeness.

We define k_0 to be an integer that is $o(N^{\frac{2}{3}})$ and divide the sum into two parts:

$$\sum_{0 \leq k \leq N} \frac{(N-k)^k (N-k)!}{N!} = \sum_{0 \leq k \leq k_0} \frac{(N-k)^k (N-k)!}{N!} + \sum_{k_0 < k \leq N} \frac{(N-k)^k (N-k)!}{N!}$$

For the first (main) term, we use the relative approximation from Lemma 1. For the second term (the tail), the restriction $k > k_0$ and the fact that the terms are decreasing imply that they are all exponentially small, as discussed in the proof of lemma 2. Putting these two observations together, we have shown that:

$$P(N) = \sum_{0 \leq k \leq k_0} e^{-\frac{k^2}{2N}} \left(1 + O(\frac{k}{N}) + O(\frac{k^3}{N^2})\right) + \Delta$$

where we use Δ to represent a term that is exponentially small. Moreover, $e^{-\frac{k^2}{2N}}$ is also exponentially small for $k > k_0$ and we can add the terms for $k > k_0$ back in, so we have:

$$P(N) = \sum_{0 \leq k} e^{-\frac{k^2}{2N}} + O(1)$$

where we have essentially replaced the tail of the original sum with the approximation, which is justified because both are exponentially small. The $O(1)$ absorbs the exponentially small terms. The remaining sum is the sum of values of the function $e^{\frac{x^2}{2}}$ at regularly spaced points with the step $\frac{1}{\sqrt{N}}$. Thus, the Euler-Maclaurin theorem provides the approximation:

$$\sum_{0 \leq k} e^{\frac{k^2}{2N}} = \sqrt{N} \int_0^\infty e^{\frac{x^2}{2}} dx + O(1)$$

The value of this integral is well known to be $\sqrt{\frac{\pi}{2}}$. Substituting the above expression for $P(N)$, we get the desired result:

$$P(N) = \sqrt{\frac{\pi N}{2}} + O(1).$$

Solution to HW 3, Problem 5.1

Al-Naji, Nader

Problem: How many bitstrings of length N have no 000?

Solution:
A bitstring without runs of three zeros is either empty, a single 0, exactly 00, or (1, 01, or 001) followed by a bitstring. Using the symbolic method, we define the following combinatorial construction using the above reasoning:

$$B = \epsilon + Z_0 + Z_0 \times Z_0 + (Z_1 + Z_0 \times Z_1 + Z_0 \times Z_0 \times Z_1) \times B$$

Taking Z_0 and Z_1 to be sets containing single elements of size one, we have that their generating functions are both (number of elements)$z^{\text{size}} = z$. Thus, as we get as our generating function:

$$B(z) = 1 + z + z^2 + (z + z^2 + z^3)B(z)$$
$$B(z) = \frac{1 + z + z^2}{1 - z - z^2 - z^3}$$

While we cannot solve this OGF directly, we can use the following transfer theorem to develop an expression for the n^{th} coefficient:

$$[z^n]\frac{f(z)}{g(z)} = -\frac{\beta f(\frac{1}{\beta})}{g'(\frac{1}{\beta})}\beta^n$$

where $\frac{1}{\beta}$ is the biggest root of $g(z)$. Solving numerically, we find that the largest root of $g(z) = 0.543689 \rightarrow \beta = 1.83929$. Also, differentiating yields $g'(z) = -3z^2 - 2z - 1$. Plugging all of this into our transfer theorem yields: $[z^n]B(z) \sim (1.13745)(1.83929)^n$.

Solution to HW 3, Problem 5.3

Al-Naji, Nader

Problem: Let $\mathcal{U}$ be the set of binary trees with the size of a tree defined to be the total number of nodes (internal plus external), so that the generating function for its counting sequence is $\mathcal{U}(z) = z + z^3 + 2z^5 + 5z^7 + 14z^9 + ...$ Derive an explicit expression for $\mathcal{U}(z)$.

Solution:
As before, we can define a binary tree as an external node or an internal node with two binary trees attached to it. We can let $Z_\square$ be a combinatorial class containing a single external node and $Z_\circ$ be a combinatorial class containing a single internal node. Now, because both internal *and* external nodes have size one, the generating function for both of these classes is simply z. Thus, writing down the construction and solving, we get:

$$\begin{aligned}
\mathcal{U} &= Z_\square + Z_\circ \times \mathcal{U} \times \mathcal{U} \\
\mathcal{U}(z) &= z + z\mathcal{U}(z)^2 \\
0 &= z - \mathcal{U}(z) + z\mathcal{U}(z)^2 \\
\Rightarrow \mathcal{U}(z) &= \frac{1 - \sqrt{1 - 4z^2}}{2z}
\end{aligned}$$

which gives us our final answer.

What follows below is not part of the solution; it is just an explicit expression for the counting sequence associated with the generating function.
If the number of internal nodes in a binary tree is k, then from chapter 5 we know that the number of external nodes is $k+1$. This implies that the total number of nodes in a binary tree can be expressed as $2k+1$, where k is the number of internal nodes. Immediately, we see that the total number of nodes in a binary tree must be odd. Further, if the total number of nodes in a binary tree (internal plus external) is n, then we can solve for the number of internal nodes and get: $n = 2k + 1 \to k = \frac{n-1}{2}$, where k is the number of internal nodes. Then, since all binary trees of size n will have $\frac{n-1}{2}$ internal nodes, we can simply express the number of binary trees of total size n as the number of binary trees with $\frac{n-1}{2}$ internal nodes (this is a bijection), which we know is simply the Catalan Numbers formula $\frac{1}{k+1}\binom{2k}{k}$, where k is the number of internal nodes. Thus, the number of binary trees of size n (internal plus external) and, therefore, the coefficients on $\mathcal{U}(z)$ are:

$$[z^n]\mathcal{U}(z) = \begin{cases} 0 & \text{if } n \text{ is even.} \\ \frac{2}{n+1}\binom{n-1}{\frac{n-1}{2}} & \text{if } n \text{ is odd.} \end{cases}$$

giving $\mathcal{U}(z) = \sum_{n\geq 0} \mathbf{1}_{\{\text{n is odd}\}}[n] \cdot \frac{2}{n+1}\binom{n-1}{\frac{n-1}{2}} z^n$.

Solution to HW 3, Problem 5.7

Al-Naji, Nader

Problem: Derive an EGF for the number of permutations whose cycles are all of odd length.

Solution:
Here we want $P^*_{\text{odd}}(z)$, the EGF associated with the combinatorial class P_{odd}, the class of sets of cycles of N items where all cycles are odd. Using the symbolic method, we have the construction:

$$P^*_{\text{odd}} = SET(CYC_{\text{odd}}(\mathcal{Z}))$$

which immediately translates to the EGF:

$$P^*_{\text{odd}}(z) = \exp\left(\sum_{k\geq 0} \frac{z^{2k+1}}{2k+1}\right)$$

$$P^*_{\text{odd}}(z) = \exp\left(\frac{1}{2}\ln\left(\frac{1+z}{1-z}\right)\right)$$

$$P^*_{\text{odd}}(z) = \left(\frac{1+z}{1-z}\right)^{\frac{1}{2}}$$

which is the desired EGF.

Solution to HW 3, Problem 5.15 and 5.16

Al-Naji, Nader

Problem: Find the average number of internal nodes in a binary tree of size N with both children internal. Find the average number of internal nodes in a binary tree of size N with one child internal and one child external.

Solution:
In order to do this problem, we first define our notion of cost. In general, cost can be any property of a combinatorial structure (such as the number of "ones" in a string of length n or, as we'll do here, the number of nodes in a binary tree that have no children). Once we agree on an intuitive "cost" that we want to measure, we can define a bivariate generating function to be $T(z,u) = \sum_{n\geq 0}\sum_{k\geq 0} a_{nk}u^k z^n$ where $a_{nk} =$"the number of structures of size n that have cost k". From this definition, if we plug in $u = 1$, it should be clear that we get $T(z,1) = \sum_{n\geq 0}\sum_{k\geq 0} a_{nk}z^n = \sum_{n\geq 0} a_n z^n$ where $a_n \equiv \sum_{k\geq 0} a_{nk} =$"the total number of structures of size n". Slightly more subtle is the fact that taking the derivative of T with respect to u yields: $\frac{\partial T(z,u)}{\partial u} \equiv T_u(z,u) = \sum_{n\geq 0}\sum_{k\geq 0} k a_{nk}u^{k-1}z^n$, where $a_{nk} =$"the number of structures of size n that have cost k" and, therefore, $ka_{nk} =$"the total cost contributed by all structures of size n and cost k". Then, following the same reasoning as before, to get a value for the total cost associated with objects of size n, all we have to do is set $u = 1$ in the derivative of T: $T_u(z,1) = \sum_{n\geq 0}\sum_{k\geq 0} k a_{nk} z^n = \sum_{n\geq 0} c_n z^n$, where $c_n \equiv \sum_{k\geq 0} k a_{nk} =$"the total cost contributed by objects of size n". Thus, if we can define a bivariate generating function that accurately captures our notion of cost, then the n^{th} term of the derivative of this generating function evaluated at $u = 1$ will yield the total cost associated with all objects of size n. Dividing this quantity by the total number of structures of size n then yields the answer we're looking for: "the average cost associated with a structure of size n".

We now define our generating function. First, define: $Z_{\circ TT} =$"the set containing a node with two trees (recursively defined) as children", $Z_{\circ\square T} =$ "the set containing a single node with one tree (recursively defined) as its child and one external node as its child", and $Z_{\circ\square\square} =$"the set containing a single node with two external nodes as children." The "size" of all the elements in each of these combinatorial classes will clearly be one, since they all contain a just single node. Further, we will let the generating functions of each of these classes be: $Z_{\circ TT}(z,u_1) = zu_1$, $Z_{\circ\square T}(z,u_2) = zu_2$, and $Z_{\circ\square\square}(z,u_3) = zu_3$, since they each have only one element of exactly size one (which gives us the common z term), and since we want to associate a different cost to each (hence the different indexing for the u terms). Now that we have these basic primitives, we can define our bivariate generating function to be:

$$
\begin{aligned}
T &= Z_{\circ TT} \times T \times T + 2Z_{\circ \Box T} \times T + Z_{\circ \Box\Box} \\
\Rightarrow T(z,u_1,u_2,u_3) &= zu_1 T(z,u_1,u_2,u_3)^2 + 2zu_2 T(z,u_1,u_2,u_3) + zu_3 \\
\Rightarrow 0 &= zu_3 + (2zu_2 - 1)T(z,u_1,u_2,u_3) + zu_1 T(z,u_1,u_2,u_3)^2 \\
\Rightarrow T(z,u_1,u_2,u_3) &= \frac{\sqrt{(1-2u_2z)^2 - 4u_1u_3z^2} + 2u_2z - 1}{2u_1z}
\end{aligned}
$$

Now, as discussed earlier, taking derivatives with respect to each of our u variables will yield the "cost" associated with each of the combinatorial objects. And, further, because we've associated each u variable with a combinatorial object of a different type (namely with nodes having varying types of children), taking the derivative with respect to u_1 and evaluating at $u_1 = u_2 = u_3 = 1$ will give us the generating function for the number of nodes in a tree of size n with two internal nodes as children (since the recursive definition implies trees always have an internal node as their root), taking the derivative with respect to u_2 and evaluating at $u_1 = u_2 = u_3 = 1$ will give us the generating function for the number of nodes in a tree of size n with one internal node and one external node as a children, and (finally...) taking the derivative with respect to u_3 and evaluating at $u_1 = u_2 = u_3 = 1$ will give us the generating function for the number of nodes in a tree of size n with two external nodes as children. Thus, using combinatorial identites from chapter 3 to simplify, we have:

$$
\begin{aligned}
T^{\circ\Box\Box}(z) &= T_{u_3}(z,1,1,1) = \frac{z}{\sqrt{1-4z}} = \sum_{n\geq 0} \binom{2n-2}{n-1} z^n \\
T^{\circ\Box T}(z) &= T_{u_2}(z,1,1,1) = \frac{1}{\sqrt{1-4z}} - \frac{2z}{\sqrt{1-4z}} - 1 = \sum_{n\geq 1} \left(\binom{2n}{n} - 2\binom{2n-2}{n-1} \right) z^n \\
T^{\circ TT}(z) &= T_{u_1}(z,1,1,1) = \frac{2z\left(z + \sqrt{1-4z} - 2\right) - \sqrt{1-4z} + 1}{2z\sqrt{1-4z}} \\
&= \sum_{n\geq 1} \left(\binom{2n-2}{n-1} - \frac{1}{n+1}\binom{2n}{n} \right) z^n
\end{aligned}
$$

Summarizing our results, we have:

Total number of internal nodes across all binary trees of size n with both children internal: $\binom{2n-2}{n-1} - \frac{1}{n+1}\binom{2n}{n}$.
Total number of internal nodes across all binary trees of size n with one child internal and one child external: $\binom{2n}{n} - 2\binom{2n-2}{n-1}$.
Total number of internal nodes across all binary trees of size n with both children external: $\binom{2n-2}{n-1}$.
Total number of internal nodes across all binary trees of size n = sum of three quantities listed above = $\binom{2n-2}{n-1} - \frac{1}{n+1}\binom{2n}{n} + \binom{2n}{n} - 2\binom{2n-2}{n-1} + \binom{2n-2}{n-1} = \frac{n}{n+1}\binom{2n}{n} = n\cdot$(Catalan Numbers) = $n \cdot$ (total number of binary trees of size n). This means everything is consistent.

Finally, to get the average number of each of these quantities, we simply divide by the number of binary trees in each case (assuming each tree has equal probability of occurring), which we know to be the Catalan Numbers. We do this below, giving our final answer:

Average number of internal nodes in a binary tree of size n with both children internal: $\frac{\binom{2n-2}{n-1} - \frac{\binom{2n}{n}}{n+1}}{\frac{\binom{2n}{n}}{n+1}} = \frac{(n-2)(n-1)}{4n-2}$.
Average number of internal nodes in a binary tree of size n with one child internal and one child external: $\frac{\binom{2n}{n} - 2\binom{2n-2}{n-1}}{\frac{\binom{2n}{n}}{n+1}} = \frac{n^2-1}{2n-1}$.
Average number of internal nodes in a binary tree of size n with both children external: $\frac{\binom{2n-2}{n-1}}{\frac{\binom{2n}{n}}{n+1}} = \frac{n(n+1)}{4n-2}$.
And, finally, as a sanity check, we have: $\frac{n^2-1}{2n-1} + \frac{(n-2)(n-1)}{4n-2} + \frac{n(n+1)}{4n-2} = n$.