Math 249 - HW 1

Joshua Turcotti

February 7, 2021

1 Boxes in a Circle

(a) Consider n boxes arranged in a circle, with n boundary lines between them within the circle. Specifically, consider the boundary line $B^{(0)}$ at the very top, and the boundary line $B^{(1)}$ immediately clockwise of it. Let $C_n^{(0)}$ be the tilings of the n boxes with monominoes and dominoes that place a boundary line at $B^{(0)}$ and $C_n^{(1)}$ be the tilings that place a boundary line at $B^{(1)}$. Since no domino or monomino can contain 2 consecutive boundary lines in its interior, $C_n = C_n^{(0)} \cup C_n^{(1)}$, and thus $|C_n| = \left|C_n^{(0)}\right| + \left|C_n^{(1)}\right| - \left|C_n^{(0)} \cap C_n^{(1)}\right|$. We note that the former two terms are simply T_n, the number of ways to tile n consecutive boxes, and the latter term is T_{n-1}, the number of ways to tile $n-1$ consecutive boxes. Thus:

$$\begin{aligned} L_n &= T_n + T_n - T_{n-1} \\ &= F_{n+1} + F_{n+1} - F_n \\ &= F_{n+1} + F_{n-1} \end{aligned}$$

(b) We recall the identity $F_{n+m} = F_{m+1}F_n + F_mF_{n-1}$, and compute (with initially arbitrary k that will later be chosen to be $m-1$ so the desired identity is derived):

$$\begin{aligned} L_{m+n} &= F_{m+n+1} + F_{m+n-1} \\ &= F_{k+1}F_{m+n+1-k} + F_kF_{m+n-k} + F_{k+1}F_{m+n-1-k} + F_kF_{m+n-2-k} \\ &= F_{k+1}L_{m+n-k} + F_kL_{m+n-1-k} \\ &= F_mL_{n+1} + F_{m-1}L_n \end{aligned}$$

(c)

$$\begin{aligned} F_{2n} &= F_nF_{n+1} + F_nF_{n-1} \\ &= F_n(F_{n+1} + F_{n-1}) \\ &= F_nL_n \end{aligned}$$

2 Permutations of a Multiset

3 Methods for computing the permutations of a multiset with n total elements in m classes, with n_i elements of class $1 \le i \le m$.

1. We may first choose the positions of the elements of class 1, which may be done in $\binom{n}{n_1}$ ways, then the positions of class 2, which may be done in $\binom{n-n_1}{n_2}$ ways, similarly class 3 in $\binom{n-n_1-n_2}{n_3}$ ways, until we choose class m in $\binom{n_m}{n_m}$ ways. Writing it out:

$$\frac{n!}{n_1!(n-n_1)!}\frac{(n-n_1)!}{n_2!(n-n_1-n_2)!}\cdots\frac{n_m!}{n_m!0!} = \frac{n!}{n_1!n_2!\dots n_m!}$$

2. We may arrange all n elements as if they are all unique, then note that we have overcountered by $n_i!$ for each class $1 \le i \le m$, so we take our total number of $n!$ permutations and divide by $n_i!$ for each class of redundancies, yielding the same expression as above.

3. We note that the identity holds when each class has but a single element (as all the denominator terms are 1 so it reduces to $n!$), so we perform vector induction by proving that the case of $(n_1, \dots, n_m)$ follows from (WLOG) the case of $(n_1 - 1, n_2, \dots, n_m)$, where $n_1 \neq 1$. We note that in the former case, we may choose the position of a single element of class 1 in n ways, and arrange the rest of the elements in $\frac{(n-1)!}{(n_1-1)!n_2!\dots n_m!}$ ways by the inductive hypothesis. This tells us that there are $\frac{n!}{(n_1-1)!n_2!\dots n_m!}$ ways to permute the $(n_1, \dots, n_m)$ multiset and choose a special element of class 1. To eliminate the choice of the special element of class 1, we group the n_1 different ways to choose that special element into a single case, and obtain the expression $\frac{n!}{n_1!\dots n_m!}$ to count only the permutations of the multiset.

3 Bijectivity of the Prufer Algorithm

Let T_X be the set of trees on the set $X \subseteq \mathbb{N}$ of nodes. Let P_X be the prufer algorithm, that takes a tree in T_X and outputs a sequence in $S_X := X^{|X|-2}$. Let R_X be the reverse Prufer algorithm, that takes a sequence $s \in S_X$ and returns a tree in T_X with Prufer code s (the existence of this algorithm was shown in lecture). We are thus given that $P_X \circ R_X = Id_{S_X}$, and we will show through induction on $|X|$ that $R_X \circ P_X = Id_{T_X}$, establishing that P_X is a bijection and $|T_X| = |S_X|$. First note that when $|X| = 3$, there are exactly three trees (identified by the unique node with degree 2) and three prufer codes (each of which consists of a single node label). P_X send a tree to the label of its degree-2 node, and R_X sends a node label to a tree with that node as its unique degree-2 node, so the relations $P_X \circ R_X = Id_{S_X}$ and $R_x \circ P_X = Id_{T_X}$ are trivial. Assume the two relations hold for $|X| = n$, and choose any new X with $|X| = n+1$. For the first identity, choose any $t = (X, E) \in T_X$. Where a is the lowest labelled leaf, and $(a, b) \in E$, let $X' = X - \{A\}, E' = E - \{(a, b)\}, t' = (X', E')$. The definition of the prufer algorithm is that $P_X(t) = (b, P_{X'}(t'))$. Now $R_X(P_X(t)) = R_X(b, P_{X'}(t'))$. Since, $|X'| = n$, $R_{X'}(P_{X'}(t')) = t'$. We use the property that the set of leaves in a tree is exactly the set of node labels that do not occur in its Prufer code, combined with the knowledge that the set of leaves in t' is exactly the set of leaves in t minus a and (possibly) plus b, to conclude that a is the least element of $X - \{b\} \cup P_{X'}(t')$. But $R_X(b, P_{X'}(t'))$ is computed exactly by taking the least element a' of $X - \{B\} \cup P_{X'}(t')$, letting $X'' = X - \{a'\}$, and adding the node a' and the edge (a', b) to the tree $R_{X''}(P_{X'}(t'))$. We determined that a is exactly this least element a', so $R_X(P_X(t)) = R_X(b, P_{X'}(t'))$ is exactly the tree $R_{X'}(P_{X'}(t')) = t'$ (by the inductive hypothesis) with the node $a' = a$ and the edge $(a', b) = (a, b)$ added, which is exactly the tree t. Thus $R_X(P_X(t)) = t$, and since $t \in T_X$ was arbitrary and all possible (finite) X are inductively covered, we may conclude that $R_X \circ P_X = Id_{T_X}$ for all X. As claimed at the beginning, we have shown that P_X is a bijection from T_X to S_X.

4 Counting Parking Functions

(a) We will show that the order in which the cars attempt to park does not affect the possibility of all cars parking. Since any permutation can be factoreds as a sequence of consecutive transpositions, it suffices to show that a is a parking function iff a' is a parking function where, for some j, $a'(j) = a(j+1), a'(j+1) = a(j)$, and $a'(i) = a(i)$ for all $i \neq j$. Assume that a is a parking function, and let k_1, k_2 be the values of $a(j)$ and $a(j+1)$, choosen not respectively but such that $k_1 < k_2$. If some spot $i \geq k_1$, $i < k_2$ is available after the first $j-1$ cars have parked, then the same one of cars $j, j+1$ will park in that spot regardless of which tries first, so a' is a parking function. If no such spot i exists, then cars $j, j+1$ will occupy the first two available spaces at least k_2, and which one occupies which will switch but the overall viability will not be affected by which tried to park first. Thus a' is a parking function. We have shown that transposing the parking order of two cars in a parking function yields another parking function, and it cannot be the case that transposing a non-parking function results in a parking function because then we could simply perform the same transposition again and conclude that the original non-parking function is indeed a parking function, so we can conclude that a is a parking function iff a' is a parking function whenever a and a' are transpositions. With this in mind, we constrain our attention to the case in which cars park in ascending order of preference. If it is the case that $b(i) \leq i$ for every i, then it is easy to see that car i will park in spot i and so clearly b is a parking function. If $b(j) > j$ for some j, then all of the cars $j, j+1, \ldots, n$ will attempt to park in the interval $b(j), \ldots, n$, which cannot happen and thus b is not a parking function. This allows us to conclude that any a is a parking function iff its increasing arrangement b is a parking function, which occurs iff $b(i) \leq i$ for all i.

(b) Consider the circular parking case, in which each of n cars chooses one of the $n+1$ spots to park and works clockwise to find the first available. If no car parks in space $n+1$, then no car desired to park in space $n+1$ and no car that desired to park in a space less than $n+1$ was forced to look past the first n. If a car parks in space $n+1$ then it either desired space $n+1$ or desired a space at most n but every space at least its desire and at most n was taken. Thus a function $[n] \to [n+1]$ is a regular, $[n] \to [n]$ parking function iff is a circular parking function that results in no car parking in space $n+1$. By symmetry, the number of functions $[n] \to [n+1]$ that result in no car parking in space $n+1$ is exactly the same as the the number of functions that result in no car parking in space i for any i, and every function $[n] \to [n+1]$ results in exactly one empty space, so exactly $\frac{1}{n+1}$ of all functions $[n] \to [n+1]$ result in no car parking in space $n+1$, and thus exactly $\frac{1}{n+1}$ of all functions $[n] \to [n+1]$ are regular parking functions. This gives us the desired count of $(n+1)^{n-1}$ parking functions $[n] \to [n]$.

5 Other Catalan Countings

We know from lecture that the Catalan numbers C_n count the number of length $2n$ Dyck paths: (N, E)-only lattice paths from $(0,0)$ to (n,n) that do not dip below the line $x = y$. We will show that they also count two similar sets.

(a) Establish a bijection between words $w \in \{0,1\}^{2n}$ with n 0s and n 1s such that any prefix contains at least as many 1s as 0s and length $2n$ Dyck paths by sending 1s to N moves and 0s to E moves. If no prefix of w contains more 0s than 1s and w itself contains an equal number of 1s and 0s, then the corresponding path will never dip below $x = y$ and will terminate at (n,n). Similarly, any Dyck's path corresponds to a words with equal 1s and 0s and no prefix with more 0s than 1s. Thus the Catalan numbers C_n also count such words.

(b) Map a sequence $a_1, \ldots, a_n$ to a path from $(1,1)$ to $(n+1, n+1)$ that contains only vertical and East steps, and contains exactly the n East steps from (i, a_i) to $(i+1, a_i)$ for each i, filled in with n vertical steps to link them. Up to mirroring over the line $x = y$ these are exactly the set of length $2n$ Dyck's paths (shifted by +1, +1), because $a_i \leq i$ gaurantees no North steps are taken above $x = y$ and $a_i \leq a_{i+1}$ gaurantees no South steps are needed. A Dyck's path can be converted to a sequence $a_1, \ldots, a_n$ by shifting it (+1, +1), and letting a_i be the maximum j such that the path contains (i, j), giving us the property that $a_i \leq i$ and that $a_i \leq a_{i+1}$. Thus the sequences $a_1, \ldots, a_n$ with these two properties are counted by the Catalan numbers C_n.

6 Divisor-Free Partitions

Given $n \geq 1$, $m \geq 2$, we will show that the partitions P_d, of n into positive integers repeated fewer than m times each, and P_f, of n into arbitrarily repeated positive integers that are free of the divisor m, are equinumerous. Let g be the map sending $\gamma \in P_d$ to a partition in P_f constructed by replacing each part $p = qm^r$, expressed such that $m \nmid q$, with m^r copies of q. Let f be the map sending $\gamma \in P_f$ to a partition in P_d constructed with c_r copies of qm^r for each q that occurs as a part of γ exactly $c = (c_0 + c_1 m + c_2 m^2 + \ldots)$ times, expressed such that $c_r < m$. It suffices to show that $f \circ g = Id_{P_d}$ and $g \circ f = Id_{P_f}$. For the former identity, choose $\gamma \in P_d$. For each q free of the divisor m, let $c_r^{(q)}$ be the number of times qm^r occurs as a part of P_d, which is clearly less than m. $g(\gamma)$ will contain $c^{(q)} = \sum_r c_r^{(q)} m^r$ copies of q, and since there is a unique way to express $c^{(q)}$ as such a sum of powers of m^i, $f(g(\gamma))$ will contain $c_r^{(q)}$ copies of qm^r for each r. Since this enumeration over m-free q partitions all possible parts of γ, we have shown that $f(g(\gamma)) = \gamma$. For the latter identity, choose $\gamma \in P_f$. For each q that occurs as a part of γ, let $c^{(q)}$ be the number of times it occurs. $f(\gamma)$ will express $c^{(q)} = \sum_r c_r^{(q)} m^r$ and contain $c^{(q)}$ copies of qm^r for each r. $g(f(\gamma))$ will thus contain $c^{(q)}$ copies of q, and thus be identical to γ. We have shown both desired identities, and thus g is a bijection so the two sets of partitions are indeed equinumerous.

7 Counting Derangements

Consider the derangements of $[n]$, and let $D(n)$ be their number. Combinatorially, we observe that the element 1 can either be part of a cycle of length 2, or of length greater than 2. There are $(n-1)D(n-2)$ ways to derange $[n]$ placing 1 in a cycle of length 2, as we must choose the other element of 1's cycle and an independent derangement of the remaining $n-2$ elements. There are $(n-1)D(n-1)$ ways to derange $[n]$ placing 1 in a cycle of length greater than 2, as we can choose any derangement of the remaining elements, then choose 1 element to precede the element 1 after its insertion. This is a bijection as distinct derangements of the $n-1$ elements or distinct choice of which of the $n-1$ elements to precede 1 will yield a necessarily different permutation. In total, we have counted that $D(n) = (n-1)(D(n-1) + D(n-2))$. We can also procede with the principle of inclusion exclusion. We begin with the $n!$ permutations of $[n]$. We then remove the $(n-1)!$ permutations that fix i for each $i \in [n]$, removing $n(n-1)!$ permutations total. We then replace the $(n-2)!$ permutations of $[n]$ that fix i, j for each of the $\binom{n}{2}$ choices of distinct i, j. We exhaustively proceed to obtain:

$$\begin{aligned}
D(n) &= \sum_i \left((-1)^i (n-i)! \binom{n}{i}\right) \\
&= n! \sum_{i \le n} \frac{(-1)^i}{i!} \\
&= n\left((n-1)! \sum_{i \le n-1} \frac{(-1)^i}{i!}\right) + (-1)^n \\
&= nD(n-1) + (-1)^n \\
&= (n-1)D(n-1) + D(n-1) + (-1)^n \\
&= (n-1)D(n-1) + (n-1)D(n-2) + (-1)^{n-1} + (-1)^n \\
&= (n-1)(D(n-1) + D(n-2))
\end{aligned}$$

8 Fresh Faces

Given k out of the n couples, there are exactly $(2n-k)!2^k$ ways to seat the $2n$ people such that all k of the chosen couples sit next to each other. This is computed by treating each couple as a unit instead of 2 individual people then counting permutations, then counting the order in which each couple sits. Now, using inclusion-exclusion, the total number of ways to seat the $2n$ people such that no couples sit next to each other is:

$$\sum_{k=0}^{n} \binom{n}{k} (2n-k)!(-2)^k$$

9 Permutation Signing

First we will demosntrate the inductive approach. Letting $c(n,k) = 0$ when $k = 0$ or $k > n$, the identity $c(n,k) = c(n-1,k-1) + (n-1)c(n-1,k)$ can be seen to hold in all cases by observing that of the k-cycle permutations of n elements, $c(n-1,k-1)$ of them place element n in a 1-cycle, and all the rest can be obtained by choosing an arbitrary k-cycle permutation of the elements $1, \ldots, n-1$ and then choosing one of those $n-1$ to precede n upon its insertion into one of the k cycles. Assume that the given identity holds for $n-1$, and we will demonstrate the case for n:

$$\begin{aligned}
\sum_{k=1}^{n}(-1)^{n-k}c(n,k) &= \sum_{k=1}^{n}(-1)^{n-k}(c(n-1,k-1)+(n-1)c(n-1,k)) \\
&= \sum_{k=1}^{n-1}(-1)^{n-k-1}c(n-1,k)-(n-1)\sum_{k=1}^{n-1}(-1)^{n-k-1}c(n-1,k) \\
&= 0-(n-1)0 \\
&= 0
\end{aligned}$$

Now, we can define the sign of a permutation of $[n]$ as -1 raised to the number of cycles. We define the involution $\iota : \sigma \mapsto \sigma \circ (1\ 2)$, which composes any permutation with the transposition $(1\ 2)$. $\iota^2(\sigma) = \sigma \circ (1\ 2) \circ (1\ 2) = \sigma$, so it is indeed an involution, and further it sends any σ in which 1 and 2 are in separate cycles to one in which they share a cycle, and vice versa, while leaving all other cycles untouched. Thus it is a sign-reversing involution that additionally has no fixed points, as no permutation is equal to itself composed with $(1\ 2)$. Since the sum in the identity is exactly $\sum_\sigma \operatorname{sgn} \sigma$, we can conclude it equals 0.

10 Adjacency-Free Partitions

First, we note that for any choice of k pairs of the form $i, i+1 \in [n]$, there are $B(n-k)$ ways to partition $[n]$ such that in each of the pairs, both elements are in the same block. It may seem as if some distinction should be made between overlapping and non-overlapping sets of pairs - but in fact it makes no difference - as we can consider each element of $[n]$ to initially be its own "atom", and then every pair that we consider merges two atoms, decreasing the total number by 1 and leaving us with $n-k$ atoms to be partitioned into blocks of atoms, or equivalently blocks of elements that unite each pair. With this identity ($B(n-k)$ ways to partition $[n]$ when k pairs are chosen to be forcibly united) justified, we can use inclusion-exclusion to write down the total number of partitions of $[n]$ such that no pair of the form $i, i+1$ is united:

$$A(n) = \sum_{k=0}^{n-1}(-1)^k\binom{n-1}{k}B(n-k)$$