Staff Solutions to In-Class Problems Week 10, Wed.

STAFF NOTE: Counting with Bijections, Ch. 14-14.2

Problem 1. (a) How many of the billion numbers in the range from 1 to $10^9$ contain the digit 1? (Hint: How many don’t?)

Solution. We can count up how many do not contain the digit 1 and subtract. So (total number) - (number without 1’s) = $10^9 - (9^9 - 1) = 612,579,512$ (the $-1$ is for 0 which is not in our range).

(b) There are 20 books arranged in a row on a shelf. Describe a bijection between ways of choosing 6 of these books so that no two adjacent books are selected and 15-bit strings with exactly 6 ones.

Solution. A selection of six among twenty books on a shelf corresponds in an obvious way to a 20-bit string with exactly six 1’s. For example, the 20-bit string with 1’s in exactly the 3rd, 4th, 5th, 10th, 19th and 20th positions corresponds to selecting 3rd, 4th, 5th, 10th, 19th and 20th books on the shelf.

So the problem reduces to finding a bijection between 20-bit strings with six nonadjacent 1’s and 15-bit strings with six 1’s.

But in a string, $s$, with six nonadjacent 1’s, all but the last 1 must have a 0 to its right. So we can map $s$ to a string with six 1’s and five fewer 0’s by erasing the 0’s immediately to the right of each of the first five 1’s. For example, erasing the underlined 0’s in the 20-bit string 000101010100001010 yields the 15-bit string 000110110000110.

This map is a bijection because given any 15-bit string with six 1’s, there is a unique 20-bit string with nonadjacent 1’s that maps to it, namely, the string obtained by replacing each of the first five 1’s in the 15-bit string by a 10.

Problem 2.

(a) Let $S_{n,k}$ be the possible nonnegative integer solutions to the inequality

$$x_1 + x_2 + \cdots + x_k \leq n.$$  

That is

$$S_{n,k} := \{(x_1, x_2, \ldots, x_k) \in \mathbb{N}_k \mid (1) \text{ is true}\}.$$  

Describe a bijection between $S_{n,k}$ and the set of binary strings with $n$ zeroes and $k$ ones.

Solution. The notation $0^x$ indicates a length $x$ string of 0’s.

$$(x_1, x_2, \ldots, x_k) \leftrightarrow 0^{x_1}10^{x_2}1\ldots0^{x_k}10^{n-s},$$

where $s := \sum_{i=1}^k x_i$. 
(b) Let $\mathcal{L}_{n,k}$ be the length $k$ weakly increasing sequences of nonnegative integers $\leq n$. That is

$$\mathcal{L}_{n,k} := \{(y_1, y_2, \ldots, y_k) \in \mathbb{N}^k \mid y_1 \leq y_2 \leq \cdots \leq y_k \leq n\}.$$ 

Describe a bijection between $\mathcal{L}_{n,k}$ and $S_{n,k}$.

**Solution.** $(y_1, y_2, \ldots, y_k) \longleftrightarrow (y_1, y_2 - y_1, y_3 - y_2, \ldots, y_k - y_{k-1})$.

In the other direction,

$$(x_1, x_2, \ldots, x_k) \longleftrightarrow (x_1, x_1 + x_2, x_1 + x_2 + x_3, \ldots, \sum_{i=1}^{k} x_i).$$

Problem 3.

An $n$-vertex numbered tree is a tree whose vertex set is $\{1, 2, \ldots, n\}$ for some $n > 2$. We define the code of the numbered tree to be a sequence of $n - 2$ integers from 1 to $n$ obtained by the following recursive process:

If there are more than two vertices left, write down the father of the largest leaf, delete this leaf, and continue this process on the resulting smaller tree. If there are only two vertices left, then stop—the code is complete.

For example, the codes of a couple of numbered trees are shown in the Figure 1.

**STAFF NOTE:** Have students begin by deriving the codes shown for the trees in the Figure to be sure they understand the coding process.

(a) Describe a procedure for reconstructing a numbered tree from its code.

\[\text{footnote}{1\text{The necessarily unique node adjacent to a leaf is called its father.}}\]
Solution. The key observation is that, given a code of length \( n - 2 \), the numbers between 1 and \( n \) which do not appear in the code are precisely the leaves of the tree. This follows because the vertices left at the end of the process are both leaves. So the procedure must have changed all the nonleaf vertices into leaves, and this implies that all the nonleaf vertices appear in the code.

Hence, the largest missing number is a leaf attached to the first number of the code. The rest of the tree can now be reconstructed by deleting the first number in the code, henceforth ignoring the largest leaf, and proceeding recursively on the rest of the code. (We’re using the obvious fact that what’s left after deleting a leaf from a tree is another tree.)

More precisely, the reconstruction procedure applies to any finite tree whose vertex set is totally ordered. The procedure takes two parameters: the vertex set, \( V \), and a length \( jV - 2 \) “code” sequence, \( S \), of elements in \( V \). If \( l \) is the largest element in \( V \) which does not appear in \( S \), and \( f \) is the first element of \( S \), then the reconstructed tree is obtained by adding edge \((l, f)\) to the tree reconstructed by calling the procedure recursively with first argument \( V \setminus f \cup l \) and second argument equal to the code obtained by erasing the initial \( f \) from \( S \). The procedure terminates when \( jV \leq 2 \), returning the edge between the two numbers in \( V \).

(b) Conclude there is a bijection between the \( n \)-vertex numbered trees and \( \{1, \ldots, n\}^{n-2} \), and state how many \( n \)-vertex numbered trees there are.

Solution. There are exactly as many \( n \)-vertex numbered trees as the number of possible code words, that is, the number of length \( n - 2 \) sequences of integers between 1 and \( n \). So there are \( n^{n-2} \) numbered trees.

The reason is that the map from trees to codes is a bijection. To see this, note that the tree reconstruction procedure finds the only possible tree with that code. So there can’t be two trees with the same code, that is, the map from a tree to its code is an injection. But since the reconstruction procedure finds a tree for every possible codeword, the map from trees to codes is also a surjection.

Supplemental problem

Problem 4.

Let \( X \) and \( Y \) be finite sets.

(a) How many binary relations from \( X \) to \( Y \) are there?

Solution. The set of all pairs \( X \times Y \) has \( |X| \cdot |Y| \) elements. Any subset of \( X \times Y \) can be the graph of a relation, hence there are \( 2^{|X| \cdot |Y|} \) relations.

(b) Define a bijection between the set \([X \to Y]\) of all total functions from \( X \) to \( Y \) and the set \( Y^{|X|} \). (Recall \( Y^n \) is the cartesian product of \( Y \) with itself \( n \) times.) Based on that, what is \( |[X \to Y]|? \)

Solution. We can encode a given function from \( X \) to \( Y \) by first giving an ordering to elements in \( X \), say, calling them \( x_1, x_2, \ldots, x_{|X|} \).

Now given an element \( f \in [X \to Y] \) we can associate it with and element \( g \in Y^{|X|} \) by following the rule \( g[i] = f(x_i) \), where \( g[i] \) is the \( i \)th entry of the vector.

This is a total, bijective function, since it is defined for every \( f \in [X \to Y] \). It is also surjective and injective, as we show next.

To prove it is surjective, suppose \((y_1, y_2, y_3, \ldots, y_{|X|}) \in Y^{|X|} \). Now, the function \( h \in X \) with \( h(x_i) := y_i \) will map to it under our definition. To prove it is injective, suppose \( g, h \in X \) map to the same vector
The function which maps a permutation \( p \) to the set of permutations of \( X \) is a total function from \( p \) to \( X \). Hence \( g = h \).

Based on this bijection we can easily count the number of total functions \([X \rightarrow Y]\) by counting the elements of \( Y^{\mid X \mid} \). Since we know how to count cartesian products, we know the answer is \( \mid Y \mid^{\mid X \mid} \). For this reason, many books use the notation \( Y^X \) in place of \([X \rightarrow Y] \).

(e) Using the previous part how many functions, not necessarily total, are there from \( X \) to \( Y \)? How does the fraction of functions vs. total functions grow as the size of \( X \) grows? Is it \( O(1) \), \( O(\mid X \mid) \), \( O(2^{\mid X \mid}) \), \( \ldots \)?

**Solution.** We can model this by adding a dummy element to \( Y \), which indicates whether a given \( x \in X \) has an actual image or not. After using the previous part, we get there are \((\mid Y \mid + 1)^{\mid X \mid}\) functions, not necessarily total. By taking the ratio of this answer and the previous questions, we see the ratio is

\[
\left( \frac{\mid Y \mid + 1}{\mid Y \mid} \right)^{\mid X \mid}
\]

so it is not \( O(1) \) nor \( O(\mid X \mid) \) but exponential in \( \mid X \mid \). Also, since \( \mid Y \mid + 1 \leq 2\mid Y \mid \), then the ratio above is indeed \( O(2^{\mid X \mid}) \).

(d) Show a bijection between the powerset, \( \text{pow}(X) \), and the set \([X \rightarrow \{0, 1\}]\) of 0-1-valued total functions on \( X \).

**Solution.** Consider bijection \( b : \text{pow}(X) \rightarrow [X \rightarrow \{0, 1\}] \) defined as follows. For \( S \in \text{pow}(X) \), then let \( b_S(x_i) := 1 \) iff \( x_i \in S \). We make \( b_S(x_i) := 0 \) otherwise. It can be shown this correspondence is a bijection. Firstly, to show it is injective, we can consider two different elements in \( \text{pow}(X) \), call them \( S_1 \) and \( S_2 \). According to the definition, these two are distinct sets with all their elements in \( X \). Therefore we can assume without losing generality there is an \( x_0 \in S_1 \) but \( x_0 \not\in S_2 \). So according to our mapping \( b_{S_1}(x_0) = 1 \) but \( b_{S_2}(x_0) = 0 \), so the two functions are not equal. Now we need to show it is surjective, and we know this is the case because given any such binary function, we can construct a subset of \( X \) that maps to it. Namely, \( \{x \in X \mid f(x) = 1\} \).

This and the previous part show why \( \text{pow}(x) \) is sometimes denoted as \( 2^X \).

(e) Let \( X \) be a set of size \( n \) and \( B_X \) be the set of all bijections from \( X \) to \( X \). Describe a bijection from \( B_X \) to the set of permutations of \( X \).\(^2\) This implies that there are how many bijections from \( X \) to \( X \)?

**Solution.** Suppose \( X = \{x_1, x_2, \ldots, x_n\} \). For any permutation \( p := (p_1, p_2, \ldots, p_n) \) of \( X \), define the function \( f_p \) by the rule

\[
f_p(x_i) := p_i.
\]

The function \( f_p \) is a bijection since every \( x \in X \) appears exactly once in \( p \).

The function which maps a permutation \( p \) to the bijection \( f_p \) is a total function from permutations of \( X \) to \( B_X \), and in fact obviously is a bijection since every \( b \in B_X \) equals \( f_{p_b} \) for some permutation \( p_b \), namely,

\[
p_b := (b(x_1), b(x_2), \ldots, b(x_n)).
\]

\(^2\)A sequence in which all the elements of a set \( X \) appear exactly once is called a permutation of \( X \).