1 Fibonacci Proof

Let F_i be the ith Fibonacci number, defined by $F_{i+2} = F_{i+1} + F_i$ and $F_0 = 0$, $F_1 = 1$. Prove that

$$\sum_{i=0}^{n} F_i^2 = F_n F_{n+1}.$$

Solution:

We proceed by induction on n.

Base case: $\sum_{i=0}^{0} F_i^2 = F_0^2 = 0 = F_0 F_1$.

Inductive hypothesis: Assume $\sum_{i=0}^{n} F_i^2 = F_n F_{n+1}$.

Inductive step: We have

$$\begin{aligned}
\sum_{i=0}^{n+1} F_i^2 &= F_{n+1}^2 + \sum_{i=0}^{n} F_i^2 \\
&= F_{n+1}^2 + F_n F_{n+1} \\
&= F_{n+1}(F_n + F_{n+1}) \\
&= F_{n+1} F_{n+2}
\end{aligned}$$

where the second equality is the inductive hypothesis and the last equality is the definition of the Fibonacci numbers.

2 Make It Stronger

Suppose that the sequence $a_1, a_2, \ldots$ is defined by $a_1 = 1$ and $a_{n+1} = 3a_n^2$ for $n \geq 1$. We want to prove that

$$a_n \leq 3^{(2^n)}$$

for every positive integer n.

(a) Suppose that we want to prove this statement using induction. Can we let our inductive hypothesis be simply $a_n \leq 3^{(2^n)}$? Attempt an induction proof with this hypothesis to show why this does not work.

(b) Try to instead prove the statement $a_n \leq 3^{(2^n - 1)}$ using induction.

(c) Why does the hypothesis in part (b) imply the conclusion from part (a)?

Solution:

(a) Let's try to prove that for every $n \geq 1$, we have $a_n \leq 3^{2^n}$ by induction.

Base Case: For $n = 1$ we have $a_1 = 1 \leq 3^{2^1} = 9$.
Inductive Step: For some $n \geq 1$, we assume $a_n \leq 3^{2^n}$. Now, consider $n+1$. We can write:

$$a_{n+1} = 3a_n^2 \leq 3(3^{2^n})^2 = 3 \times 3^{2 \times 2^n} = 3 \times 3^{2^{n+1}} = 3^{2^{n+1}+1}.$$

However, what we wanted was to get an inequality of the form: $a_{n+1} \leq 3^{2^{n+1}}$. There is an extra $+1$ in the exponent of what we derived.

(b) This time the induction works.

Base Case: For $n = 1$ we have $a_1 = 1 \leq 3^{2-1} = 3$.
Inductive Step: For some $n \geq 1$ we assume $a_n \leq 3^{2^n-1}$. Now, consider $n+1$. We can write:

$$a_{n+1} = 3a_n^2 \leq 3 \times (3^{2^n-1})^2 = 3 \times 3^{2 \times (2^n-1)} = 3 \times 3^{2^{n+1}-2} = 3^{2^{n+1}-1}.$$

This is exactly the induction hypothesis for $n+1$.

(c) For every $n \geq 1$, we have $2^n - 1 \leq 2^n$ and therefore $3^{2^n-1} \leq 3^{2^n}$. This means that our modified hypothesis which we proved in part (b) does indeed imply what we wanted to prove in part (a).

3 Functional Composition

For the following questions, let X, Y, and Z be sets.

(a) Suppose that $f : X \to Y$, $g : Y \to Z$, and $h : Z \to X$ are functions such that the composition $h(g(f(\cdot)))$ is a bijection. For each of the following statements, either prove that the statement is true, or provide a counterexample.

   (i) The function f is injective.
  (ii) The function g is bijective.
 (iii) The function h is surjective.

(b) Suppose that $p : X \to X$ is a function such that $p(p(p(x))) = x$ for all $x \in X$. Prove that p is a bijection.

Solution:

For the following questions, let X, Y, and Z be sets.

(a) (a) This is true. Suppose that $a, b \in X$ are such that $f(a) = f(b)$. Then it follows that $h(g(f(a))) = h(g(f(b)))$, and since $h(g(f(\cdot)))$ is a bijection, it is injective, so $a = b$.

(b) This is not necessarily true. For example, let $X = Z = \mathbb{R}$ and let $Y = \mathbb{R}^2$, and consider the functions $f(x) = (x,x)$, $g(x,y) = x$, and $h(x) = x$. Then $h(g(f(x))) = x$ is a bijection, but g is not injective (and so not a bijection).

(c) This is true. Suppose that $x \in X$ is arbitrary. Then since $h(g(f(\cdot)))$ is a bijection, we can find some $x' \in X$ such that $h(g(f(x'))) = x$. But, this means that $g(f(x')) \in Z$ is sent to $x \in X$ by h, hence h is surjective.

(b) We need to show that p is both injective and surjective. To see that it is injective, let $a, b \in X$ be such that $p(a) = p(b)$. Then we can see that $a = p(p(p(a))) = p(p(p(b))) = b$, hence p is injective. To see that p is surjective, let $x \in X$ be arbitrary. Then $p(p(p(x))) = x$, so the value $p(p(x))$ is sent to x under p. Thus, p is also surjective and is thus a bijection.

4 Counting Functions

Are the following sets countable or uncountable? Prove your claims.

(a) The set of all functions f from $\mathbb{N}$ to $\mathbb{N}$ such that f is non-decreasing. That is, $f(x) \leq f(y)$ whenever $x \leq y$.

(b) The set of all functions f from $\mathbb{N}$ to $\mathbb{N}$ such that f is non-increasing. That is, $f(x) \geq f(y)$ whenever $x \leq y$.

(c) The set of all bijective functions from $\mathbb{N}$ to $\mathbb{N}$.

Solution:

(a) Uncountable: Let us assume the contrary and proceed with a diagonalization argument. If there are countably many such functions we can enumerate them as

	0	1	2	3	...
f_0	$f_0(0)$	$f_0(1)$	$f_0(2)$	$f_0(3)$	...
f_1	$f_1(0)$	$f_1(1)$	$f_1(2)$	$f_1(3)$	...
f_2	$f_2(0)$	$f_2(1)$	$f_2(2)$	$f_2(3)$	...
f_3	$f_3(0)$	$f_3(1)$	$f_3(2)$	$f_3(3)$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Now go along the diagonal and define f such that $f(x) > f(x-1)$ and $f(x) > f_x(x)$ for each $x \in \mathbb{N}$, which is possible because at step k we only need to find a number $\in \mathbb{N}$ greater than all the $f_j(j)$ for $j \in \{0,\ldots,k\}$. This function differs from each f_i and therefore cannot be on the list, hence the list does not exhaust all non-decreasing functions. As a result, there must be uncountably many such functions.

Alternative Solution 2: We can inject the set of infinitely long binary strings into the set of non-decreasing functions as follows. For any infinitely long binary string b, let $f(n)$ be equal to the number of 1's appearing in the first n-digits of b. It is clear that the function f so defined

is non-decreasing. Also, since the function f is uniquely defined by the infinitely long binary string, the mapping from binary strings to non-decreasing functions is injective. Since the set of infinite binary strings is uncountable, and we produced an injection from that set to the set of non-decreasing functions, that set must be uncountable as well.

(b) Countable: Let D_n be the subset of non-increasing functions for which $f(0) = n$. Any such function must stop decreasing at some point (because $\mathbb{N}$ has a smallest number), so there can only be finitely many (at most n) points $X_f = \{x_1, \ldots, x_k\}$ at which f decreases. Let y_i be the amount by which f decreases at x_i, then f is fully described by $\{(x_1, y_1), \ldots, (x_k, y_k),$ $(-1,0), \ldots, (-1,0)\} \in \mathbb{N}^{2n} = \mathbb{N} \times \mathbb{N} \times \cdots \times \mathbb{N}$ (n times), where we padded the k values associated with f with $n - k$ $(-1,0)$s. In Lecture note 4, we have seen that $\mathbb{N} \times \mathbb{N}$ is countable by the spiral method. Using it repeatedly, we get $\mathbb{N}^{(2^l)}$ is countable for all $l \in \mathbb{N}$. This gives us that $\mathbb{N}^{2n}$ is countable for any finite n (because $\mathbb{N}^n \subset \mathbb{N}^{(2^l)}$ where l is such that $2^l \geq n$). Hence D_n is countable.

Since each set D_n is countable we can enumerate it. Map an element of D_n to (n, j) where j is the label of that element produced by the enumeration of D_n. This produces an injective map from $\cup_{n \in \mathbb{N}} D_n$ to $\mathbb{N} \times \mathbb{N}$ and we know that $\mathbb{N} \times \mathbb{N}$ is countable from Lecture note 4 (via spiral method). Now the set of all non-increasing functions is $\cup_{i \in \mathbb{N}} D_n$, and thus countable.

(c) Uncountable: We can inject the set of infinitely long binary strings into the set of bijective functions from $\mathbb{N}$ to $\mathbb{N}$. For any binary string $b = \{b_0, b_1, b_2, \ldots\}$, consider the function $f : \mathbb{N} \to \mathbb{N}$ given by

$$f(2n) = 2n \text{ and } f(2n+1) = 2n+1 \qquad \text{if } b_n = 0,$$
$$f(2n) = 2n+1 \text{ and } f(2n+1) = 2n \qquad \text{if } b_n = 1.$$

Note that this is a bijective function. Also, since f is uniquely defined by the binary string b, the mapping from infintely long binary strings to bijective functions is injective. Since the set of infinitely long binary strings is uncountable, and we produced an injection from that set to the set of bijective functions on $\mathbb{N}$, that set must be uncountable as well.

Alternative Solution: We will show that the set of bijections of $\mathbb{N}$ is at least as large as the powerset $\mathscr{P}(\mathbb{N})$ of $\mathbb{N}$, which we know to be uncountable. To do so, we need a little lemma:

Lemma (Shufflability of subsets): If A is a subset of $\mathbb{N}$ and $|A| > 1$, then we can find a bijection $h : A \to A$, so that for all $x \in A$, $h(x) \neq x$. That is, h maps every element to an element *other than itself*.

Proof: If $|A| = 1 < n < \infty$, then we can write $A = \{a_1, \ldots, a_n\}$ and define $h(a_i) = a_{i+1 \bmod n}$ and are done. If $|A| = \infty$, then we write $A = \{a_1, a_2, \ldots\}$ and define h to swap any two consecutive elements, i.e. $h(a_1) = a_2, h(a_2) = a_1, h(a_3) = a_4, h(a_4) = a_3$, etc.

Now we are in shape to associate with each subset S of $\mathbb{N}$ (ignoring subsets that are of the form $\mathbb{N} \setminus \{x\}$, which we will take care of later), a bijection g_S of $\mathbb{N}$: Namely, let us define g_S so that for all $x \in S$, $g_S(x) = x$, and on $\mathbb{N} \setminus S$, we let g_S be any function h_S from the lemma above. All we need to prove is that g_S and $g_{S'}$ are distinct for distinct S and S'. But if $S \neq S'$, then without loss of generality there exists some $s \in S \setminus S'$. For this s, we have $g_S(s) = s \neq g_{S'}(s)$ and so

g_S and $g_{S'}$ must be different. Now, we have constructed an injection that maps the power set $\mathscr{P}(\mathbb{N})$ to a subset of bijective functions on $\mathbb{N}$, except for the special subsets of the form $\mathbb{N} \setminus \{x\}$ for some number x. The reason we excluded these sets is because then we would have to apply the shuffability lemma to the singleton $\{x\}$, which is not possible. Does this break our proof? No! The number of sets that we have ignored is countable, so the *remaining* subset of the power set that we have mapped into bijective functions *is still uncountable*, and thus the set of bijective functions from $\mathbb{N} \to \mathbb{N}$ is uncountable.

5 Counting Cartesian Products

For two sets A and B, define the cartesian product as $A \times B = \{(a,b) : a \in A, b \in B\}$.

(a) Given two countable sets A and B, prove that $A \times B$ is countable.

(b) Given a finite number of countable sets $A_1, A_2, \ldots, A_n$, prove that

$$A_1 \times A_2 \times \cdots \times A_n$$

is countable.

Solution:

(a) As shown in lecture, $\mathbb{N} \times \mathbb{N}$ is countable by creating a zigzag map that enumerates through the pairs: $(0,0), (1,0), (0,1), (2,0), (1,1), \ldots$. Since A and B are both countable, there exists a bijection between each set and a subset of $\mathbb{N}$. Thus we know that $A \times B$ is countable because there is a bijection between a subset of $\mathbb{N} \times \mathbb{N}$ and $A \times B : f(i,j) = (A_i, B_j)$. We can enumerate the pairs (a,b) similarly.

(b) Proceed by induction.
Base Case: $n = 2$. We showed in part (a) that $A_1 \times A_2$ is countable since both A_1 and A_2 are countable.
Induction Hypothesis: Assume that for some $n \in \mathbb{N}$, $A_1 \times A_2 \times \cdots \times A_n$ is countable.
Induction Step: Consider $A_1 \times \cdots \times A_n \times A_{n+1}$. We know from our hypothesis that $A_1 \times \cdots \times A_n$ is countable, call it $C = A_1 \times \cdots \times A_n$. We proved in part (a) that since C is countable and A_{n+1} are countable, $C \times A_{n+1}$ is countable, which proves our claim.

6 Bipartite Graphs

An undirected graph is bipartite if its vertices can be partitioned into two disjoint sets L, R such that each edge connects a vertex in L to a vertex in R (so there does not exist an edge that connects two vertices in L or two vertices in R).

(a) Suppose that a graph G is bipartite, with L and R being a bipartite partition of the vertices. Prove that $\sum_{v \in L} \deg(v) = \sum_{v \in R} \deg(v)$.

(b) Suppose that a graph G is bipartite, with L and R being a bipartite partition of the vertices. Let s and t denote the average degree of vertices in L and R respectively. Prove that $s/t = |R|/|L|$.

Solution:

(a) Since G is bipartite, each edge connects one vertex in L with a vertex in R. Since each edge contributes equally to $\sum_{v \in L} \deg(v)$ and $\sum_{v \in R} \deg(v)$, we see that these two values must be equal.

(b) By part (a), we know that $\sum_{v \in L} \deg(v) = \sum_{v \in R} \deg(v)$. Thus $|L| \cdot s = |R| \cdot t$. A little algebra gives us the desired result.
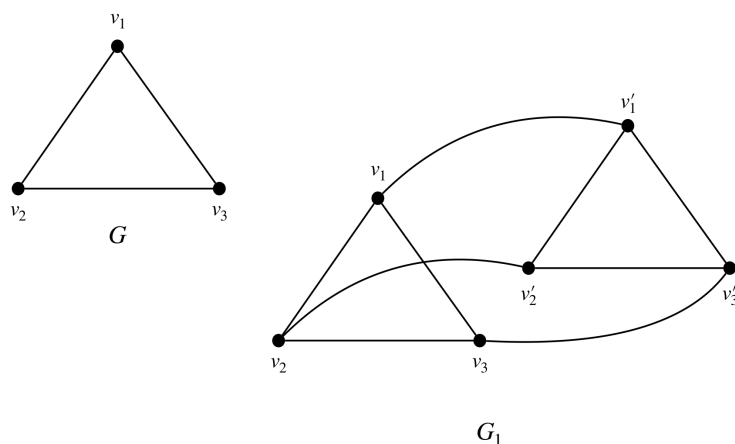
7 Doubled Graphs

The *double* of a graph G consists of two copies of G with edges joining the corresponding vertices. More precisely, if $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices and E the set of edges, then the double of the graph G is given by $G_1 = (V_1, E_1)$, where

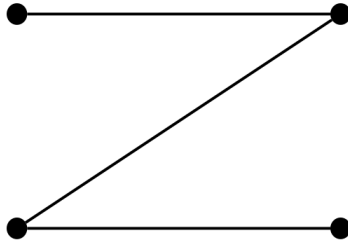$$V_1 = \{v_1, v_2, \ldots, v_n, v'_1, v'_2, \ldots, v'_n\},$$

and

$$E_1 = E \cup \{(v'_i, v'_j) | (v_i, v_j) \in E\} \cup \{(v_i, v'_i), \forall i\}.$$

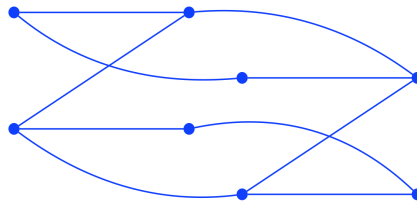Here is an example,

(a) Draw the double of the following graph:

(b) Now suppose that G_1 is an arbitrary bipartite graph (see Problem 6), and that for each integer $n > 0$, we define G_{n+1} as the double of G_n. Show that $\forall n \geq 1$, G_n is bipartite.

Hint: Use induction on n.

Solution:

(a) The double of the graph is as shown below:

(b) We proceed with induction on n. By definition, G_1 is bipartite, hence this establishes our base case.

Now suppose that for some integer $k \geq 1$ the graph G_k is bipartite, so that we can split $G_k = L \cup R$ into two disjoint vertex sets such that every edge goes from L to R. The process by which we construct G_{k+1} involves taking two copies of G_k and connecting corresponding vertices - let G_k^1 and G_k^2 be those two copies, considered as subgraphs of G_{k+1}. By the inductive hypothesis, we can split each of them into two disjoint vertex sets $G_k^1 = L_1 \cup R_1$ and $G_k^2 = L_2 \cup R_2$ such that all edges in G_k^i connect a vertex in L_i with another vertex in R_i.

Lets now zoom out to G_{k+1} as a whole. Every edge in G_{k+1} either goes from L_1 to R_1, from L_2 to R_2, or from G_k^1 to G_k^2. Those edges between the two copies of G_k connect corresponding vertices, hence they either go from L_1 to L_2 or from R_1 to R_2. It then follows that the two sets $L_1 \cup R_2$ and $L_2 \cup R_1$ form a partition of the vertices of G_{k+1} and that every edge in G_{k+1} must travel between the two sets. Thus, we can conclude that G_{k+1} is bipartite, which completes the induction.

8 Binary Trees

You may have seen the recursive definition of binary trees from previous classes. Here, we define binary trees in graph theoretic terms as follows (**Note:** here we will modify the definition of leaves slightly for consistency).

- A binary tree of height > 0 is a tree where exactly one vertex, called the **root**, has degree 2, and all other vertices have degrees 1 or 3. Each vertex of degree 1 is called a **leaf**. The **height** h is defined as the maximum length of the path between the root and any leaf.

- A binary tree of height 0 is the graph with a single vertex. The vertex is both a leaf and a root.

(a) Let T be a binary tree of height > 0, and let $h(T)$ denote it's height. Let r be the root in T and u and v be it's neighbors. Show that removing r from T will result in two binary trees, L, R with roots u and v respectively. Also, show that $h(T) = \max(h(L), h(R)) + 1$.

(b) Using the graph theoretic definition of binary trees, prove that the number of vertices in a binary tree of height h is at most $2^{h+1} - 1$.

(c) Prove that all binary trees with n leaves have $2n - 1$ vertices.

Solution:

(a) Since r has degree 2, removing it will break T into two connected components, call them L and R. By symmetry, we just need to prove that L is a binary tree. Without loss of generality, suppose $u \in L$. Before removing r, u had degree 1 or 3. If u had degree 1, then after removing r, u is a single vertex, and so is a binary tree of height 0, and also is a root. If u had degree 3, then after removing r, u has degree 2, and all other vertices in L have degree 1 or 3. Thus, L is a binary tree with root u.

To prove that $h(T) = \max(h(L), h(R)) + 1$, we note that because T is a tree, any path from r to a leaf must go through either u or v but not both. Thus the maximum distance from r to any leaf is one plus either the maximum distance from u to any leaf in L (as the path cannot go back through r) or the maximum distance from v to any leaf in R. Formally, if we define $\mathscr{L}(L)$ and $\mathscr{L}(R)$ to be the set of leaves in L and R respectively, and $d(r, l)$ as the length of the path from r to some leaf l, then we have

$$h(T) = 1 + \max(\max_{l \in \mathscr{L}(L)} (d(u, l)), \max_{l \in \mathscr{L}(R)} (d(u, l)))$$
$$= 1 + \max(h(L), h(R))$$

(b) Induction: **Base Case** a binary tree of height 0 is a singleton and so has $2^1 - 1 = 1$ vertex. **Inductive Hypothesis:** assume for all $k < h$, a binary tree of height k has at most $2^{k+1} - 1$ vertices. **Inductive Step:** By part a, we can remove the root from a binary tree and obtain two

binary trees: L, and R of height k and l respectively. Since $h(T) = \max(h(L), h(R)) + 1$, we know that $k, l \leq h - 1$ so we can apply the inductive hypothesis to L and R. Thus, we have that the number of vertices in T is at most $1 + 2^{k+1} - 1 + 2^{l+1} - 1 \leq 2^{h-1+1} + 2^{h-1+1} - 1 = 2^{h+1} - 1$.

(c) Induction: **Base Case** if a binary tree has one leaf, it is a singleton and so has $1 = 2 * 1 - 1$ vertices. **Inductive Hypothesis:** assume for all $k < n$, a binary tree with k leaves has $2k - 1$ vertices. **Inductive Step:** For a binary tree, T, with $n > 1$ leaves, remove the root, r and break T into binary trees L and R. Suppose L has a leaves and R has b leaves. Note that all the leaves of T are in L or R, as $n > 1$ implies the root is not a leaf, which means $a + b = n$. By the inductive hypothesis, L has $2a - 1$ vertices, and R has $2b - 1$ vertices, and so the number of vertices in T is $2a - 1 + 2b - 1 + 1 = 2(a + b) - 1 = 2n - 1$.