CS 70 Discrete Mathematics and Probability Theory
Fall 2018 Alistair Sinclair and Yun Song **Homework 7**

1 Bijective or not?

Decide whether the following functions are bijections or not. Please prove your claims.

(a) $f(x) = 10^{-5}x$

(i) for domain $\mathbb{R}$ and range $\mathbb{R}$

(ii) for domain $\mathbb{Z} \cup \{\pi\}$ and range $\mathbb{R}$

(b) $f(x) = p \bmod x$, where $p > 2$ is a prime

(i) for domain $\mathbb{N} \setminus \{0\}$ and range $\{0, \ldots, p\}$

(ii) for domain $\{(p+1)/2, \ldots, p\}$ and range $\{0, \ldots, (p-1)/2\}$

(c) $f(x) = \{x\}$, where the domain is $D = \{0, \ldots, n\}$ and the range is $\mathscr{P}(D)$, the powerset of D (that is, the set of all subsets of D).

(d) Consider the number $X = 1234567890$, and obtain X' by shuffling the order of the digits of X. Is $f(i) = (\mathrm{i}+1)^{\text{st}}$ *digit of* X' a bijection from $\{0, \ldots, 9\}$ to itself?

Solution:

(a) It is bijective for (i), but fails to be surjective in (ii):

(i) Firstly, it is injective because if $f(x) = f(y)$, then $10^{-5}x = 10^{-5}y$ and so multiplying by 10^5 on both sides, we get $x = y$, so no two real numbers can be mapped to the same real numbers. Secondly, it is surjective, because for any $y \in \mathbb{R}$, we have $f(10^5 y) = 10^{-5} \cdot 10^5 y = y$, so each y has an $x = 10^5 y$ that maps to it.

(ii) f is injective for the same reason as above, but it is not a surjection, since the only $x \in \mathbb{R}$ that maps to e.g. 10^{-6} is $x = 10^{-1} \notin \mathbb{Z} \setminus \{\pi\}$.

(b) f fails to be injective in (i), but it is bijective in (ii):

(i) $f(x) = p$ for all $x > p$

(ii) We note that for $x \in \{(p+1)/2, (p+1)/2+1, \ldots, p\}$, we have $p = x + (p-x)$, where $(p-x) < x$. That is, the remainder when dividing p by x is $p-x$. Hence $f\left(\frac{p+1}{2}\right) = \frac{p-1}{2}, f\left(\frac{p+3}{2}\right) = \frac{p-3}{2}, \ldots, f(p-1) = 1, f(p) = 0$.

(c) f is injective, but not surjective. There exists a subset $S \subset D$ containing at least two elements of D (in the case of $n = 0$, the two elements are the empty set and 0 itself). However, $f(x)$ always contains exactly one element. Hence there is no $x \in D$ that gets mapped to S.

(d) Yes. Let us show injectivity by noticing that each number between 0 and 9 occurs precisely once in X, and thus precisely once in X' too. As a result, no two digits of X' can be the same. Surjectivity follows from similar reasoning: Since any fixed number $y \in \{0,\dots,9\}$ is a digit of X, it must be a digit of X' too, let's call that digit the i_y^{th} digit. Then $f(i_y) = y$.

2 Counting Tools

Are the following sets countable or uncountable? Please prove your claims.

(a) $A \times B$, where A and B are both countable.

(b) $\bigcup_{i \in A} B_i$, where A, B_i are all countable.

(c) 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 $x \leq y$.

(d) 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 $x \leq y$.

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

Solution:

(a) Countable: If A and B are countable, we have injections $f_A : A \to \mathbb{N}$ and $f_B : B \to \mathbb{N}$. We argue that $f : A \times B \to \mathbb{N} \times \mathbb{N}$ which takes (a,b) to $(f_A(a), f_B(b))$ is an injection too: If $(f_A(a), f_B(b)) = (f_A(a'), f_B(b'))$, then $f_A(a) = f_A(a')$ and $f_B(b) = f_B(b')$, so $a = a', b = b'$.

(b) Countable: Since A and the B_i are countable, we can enumerate the elements of each set. Let us call $b_{i,j}$ the j^{th} element in B_i, then $U = \bigcup_{i \in A} B_i = \bigcup_{i,j \in \mathbb{N}} b_{i,j}$ and so U has at most as many elements as $\bigcup_{i,j \in \mathbb{N}} (i,j) = \mathbb{N} \times \mathbb{N}$.

(c) Uncountable: Let us assume the contrary and proceed with a diagonal argument. If there are countably many such function 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 $f(x) > f_x(x)$ and $f(y) > f(x)$ if $y > x$, 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

$j \in \{0,\dots,k\}$. 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 uncountable many such functions.

Alternative Solution: Look at the subset $\mathscr{S}$ of strictly increasing functions. Any such f is uniquely identified by its image which is an infinite subset of $\mathbb{N}$. But the set of infinite subsets of $\mathbb{N}$ is uncountable, so $\mathscr{S}$ is uncountable and hence the set of all non-decreasing functions must be too.

Alternative Solution 2: We can inject the set of infinitely long binary strings into the set of non-decreasing functions as follows. Let $b(i)$ be the ith digit of the string b. Now, map $b(i)$ to the subset of non-decreasing functions with the following property: for all $i \in \mathbb{N}$, $b(i) = 1 \implies f(i) > f(i-1)$, and $b(i) = 0 \implies f(i) = f(i-1)$. In this way, every infinite binary string gets paired with some set of functions. For example, the string $11111\dots$ is the set of strictly increasing functions. Because we showed 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.

(d) 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 $X_f = \{x_1,\dots,x_k\}$ at which f decreases. Let y_i be the amount by which f decreases at x_i, then f is fully described by $\{(x_1,y_1),\dots,(x_k,y_k),$ $(-1,0),\dots,(-1,0)\} \in \mathbb{N}^n = \mathbb{N}\times\mathbb{N}\times\cdots\times\mathbb{N}$ (n times), where we padded the k values associated with f with $n-k$ $(-1,0)$s. But $\mathbb{N}^n$ is countable by repeated application of part (a), and hence D_n is countable. Now the set of all non-increasing functions is $\bigcup_{i\in\mathbb{N}} D_n$, which is the countable union of countable sets, and thus countable by part (b).

(e) Uncountable: 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 $A = \{a_1,\dots,a_n\}$ and define $h(a_i) = a_{i+1 \bmod n}$ and are done. If $|A| = \infty$, then we write $A = \{a_1,a_2,\dots\}$ 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.

Linking to the cardinality of power set of N

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

N \ {x} is countable, which means that power_set(N) \ {N \ {x}} is still uncountable.

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.

3 Impossible Programs

Show whether the following programs can exist or not.

(a) A program P that takes in any program F, input x and output y and returns true if $F(x)$ outputs y and false otherwise.

(b) A program P that takes in two programs F and G and returns true if F and G halt on the same inputs, and false otherwise.

Solution:

(a) P cannot exist, for otherwise we could solve the halting problem:

```
def Halt(F, x):
  def Q(x):
    F(x)
    return 0
  return P(Q, x, 0)
```

`Halt` defines a subroutine `Q` that first simulates F and then returns 0, that is `Q`(x) returns 0 if $F(x)$ halts, and nothing otherwise. Knowing the output of `P`(F,x,0) thus tells us whether F(x) halts or not.

(b) We solve the halting problem once more:

```
def Halt(F, x):
  def Q(y):
    loop
  def R(y):
    If y = x:
      F(x)
    Else:
      loop
  return not P(Q, R)
```

`Q` is a subroutine that loops forever on all inputs. `R` is a subroutine that loops forever on every input except x, and runs $F(x)$ on input x when handed x as an argument. Knowing if `Q` and `R` halt on the same inputs is thus tantamount to knowing whether F halts on x (since that is the only case in which they could possibly differ). Thus, if $P(Q,R)$ returns "True", then we know they behave the same on all inputs and F must not halt on x, so we return `not P(Q,R)`.

4 Undecided?

Let us think of a computer as a machine which can be in any of n states $\{s_1, \dots, s_n\}$. The state of a 10 bit computer might for instance be specified by a bit string of length 10, making for a total of 2^{10} states that this computer could be in at any given point in time. An algorithm $\mathscr{A}$ then is a list of k instructions $(i_0, i_2, \dots, i_{k-1})$, where each i_l is a function of a state c that returns another state u and a number j. Executing $\mathscr{A}(x)$ means computing

$$(c_1, j_1) = i_0(x), \qquad (c_2, j_2) = i_{j_1}(c_1), \qquad (c_3, j_3) = i_{j_2}(c_2), \qquad \dots$$

until $j_\ell \geq k$ for some ℓ, at which point the algorithm halts and returns $c_{\ell-1}$.

(a) How many iterations can an algorithm of k instructions perform on an n-state machine (at most) without repeating any computation?

(b) Show that if the algorithm is still running after $2n^2k^2$ iterations, it will loop forever.

(c) Give an algorithm that decides whether an algorithm $\mathscr{A}$ halts on input x or not. Does your contruction contradict the undecidability of the halting problem?

Solution:

(a) Each of the k instruction can be called on at most n different states, therefore there are at most $n \cdot k$ distinct computations that can be performed during any execution. After $n \cdot k + 1$ iterations we must have repeated one of these computations.

(b) Since $2n^2k^2 > n \cdot k + 1$, $\mathscr{A}$ must repeat a computation $i_s(c_t)$ for some $(s,t) \in \{1, \dots, n\} \times \{0, \dots, k-1\}$. But we know that when $i_s(c_t)$ is performed the second time, its consective computations will be precisely the same that followed the first evaluation of $i_s(c_t)$. In particular, we will see $i_s(c_t)$ a third time, and hence a fourth, fifth time etc.

(c) From our solution to part (b) it follows that we only need to check whether after $2n^2k^2$ iterations, $\mathscr{A}(x)$ is still running or not. If it is, $\mathscr{A}(x)$ does not halt, otherwise it does. This does not contradict the undecidability of the halting problem, since it only states the inability to decide whether an *arbitrary* algorithm halts. Here we only proved the decidability for algorithms that can be run on an n-state machine, of which there are only finitely many!

5 Clothing Argument

(a) There are four categories of clothings (shoes, trousers, shirts, hats) and we have ten distinct items in each category. How many distinct outfits are there if we wear one item of each category?

(b) How many outfits are there if we wanted to wear exactly two categories?

(c) How many ways do we have of hanging four of our ten hats in a row on the wall? (Order matters.)

(d) We can pack four hats for travels. How many different possibilities for packing four hats are there? Can you express this number in terms of your answer to part (c)?

(e) If we have exactly 3 red hats, 3 green hats and 4 turquoise hats, and hats of the same colour are indistinguishable, how many distinct sets of three hats can we pack?

Solution:

(a) 10^4

(b) $\binom{4}{2} \cdot 10^2$

(c) $\binom{10}{4} \cdot 4! = \dfrac{10!}{6!}$

(d) $\binom{10}{4}$ or written as a function of the previous part, $c/4!$.

(e) Treating hats as balls and colours as bins, we want to distribute three balls over three bins. Hence the answer is $\binom{3+2}{2} = \binom{5}{2}$.