

## 1 Countability: True or False

Note 11

- (a) The set of all irrational numbers  $\mathbb{R} \setminus \mathbb{Q}$  (i.e. real numbers that are not rational) is uncountable.
- (b) The set of real solutions for the equation  $x + y = 1$  is countable.

For any two functions  $f : Y \rightarrow Z$  and  $g : X \rightarrow Y$ , let their composition  $f \circ g : X \rightarrow Z$  be given by  $f \circ g = f(g(x))$  for all  $x \in X$ . Determine if the following statements are true or false.

- (c)  $f$  and  $g$  are injective (one-to-one)  $\implies f \circ g$  is injective (one-to-one).
- (d)  $f$  is surjective (onto)  $\implies f \circ g$  is surjective (onto).

### Solution:

- (a) **True.** Proof by contradiction. Suppose the set of irrationals is countable. From Lecture note 10 we know that the set  $\mathbb{Q}$  is countable. Since union of two countable sets is countable, this would imply that the set  $\mathbb{R}$  is countable. But again from Lecture note 10 we know that this is not true. Contradiction!
- (b) **False.** Let  $S \subset \mathbb{R} \times \mathbb{R}$  denote the set of all real solutions for the given equation. For any  $x' \in \mathbb{R}$ , the pair  $(x', y') \in S$  if and only if  $y' = 1 - x'$ . Thus  $S = \{(x, 1 - x) : x \in \mathbb{R}\}$ . Besides, the mapping  $x$  to  $(x, 1 - x)$  is a bijection from  $\mathbb{R}$  to  $S$ . Since  $\mathbb{R}$  is uncountable, we have that  $S$  is uncountable too.
- (c) **True.** Recall that a function  $h : A \rightarrow B$  is injective iff  $a_1 \neq a_2 \implies h(a_1) \neq h(a_2)$  for all  $a_1, a_2 \in A$ . Let  $x_1, x_2 \in X$  be arbitrary such that  $x_1 \neq x_2$ . Since  $g$  is injective, we have  $g(x_1) \neq g(x_2)$ . Now, since  $f$  is injective, we have  $f(g(x_1)) \neq f(g(x_2))$ . Hence  $f \circ g$  is injective.
- (d) **False.** Recall that a function  $h : A \rightarrow B$  is surjective iff  $\forall b \in B, \exists a \in A$  such that  $h(a) = b$ . Let  $g : \{0, 1\} \rightarrow \{0, 1\}$  be given by  $g(0) = g(1) = 0$ . Let  $f : \{0, 1\} \rightarrow \{0, 1\}$  be given by  $f(0) = 0$  and  $f(1) = 1$ . Then  $f \circ g : \{0, 1\} \rightarrow \{0, 1\}$  is given by  $(f \circ g)(0) = (f \circ g)(1) = 0$ . Here  $f$  is surjective but  $f \circ g$  is not surjective.

## 2 Counting Cartesian Products

Note 11

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  $A_1, A_2, \dots, A_n$ , prove that

$$A_1 \times A_2 \times \dots \times A_n$$

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:  $(0, 0), (1, 0), (0, 1), (2, 0), (1, 1), \dots$ . 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  $n \in \mathbb{N}$ ,  $A_1 \times A_2 \times \dots \times A_n$  is countable.  
 Induction Step: Consider  $A_1 \times \dots \times A_n \times A_{n+1}$ . We know from our hypothesis that  $A_1 \times \dots \times A_n$  is countable, call it  $C = A_1 \times \dots \times A_n$ . 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.

## 3 Hello World!

### Note 12

Determine the computability of the following tasks. If it's not computable, write a reduction or self-reference proof. If it is, write the program.

- (a) You want to determine whether a program  $P$  on input  $x$  prints "Hello World!". Is there a computer program that can perform this task? Justify your answer.
- (b) You want to determine whether a program  $P$  prints "Hello World!" before running the  $k$ th line in the program. Is there a computer program that can perform this task? Justify your answer.
- (c) You want to determine whether a program  $P$  prints "Hello World!" in the first  $k$  steps of its execution. Is there a computer program that can perform this task? Justify your answer.

### Solution:

- (a) Uncomputable. We will reduce `TestHalt` to `PrintsHW(P,x)`.

```
TestHalt (P, x) :
  P' (x) :
    run P(x) while suppressing print statements
```

```

    print("Hello World!")
if PrintsHW(P', x):
    return true
else:
    return false

```

If `PrintsHW` exists, `TestHalt` must also exist by this reduction. Since `TestHalt` cannot exist, `PrintsHW` cannot exist.

(b) **Uncomputable.** Reduce `PrintsHW( $P, x$ )` from part (a) to this program `PrintsHWByK( $P, x, k$ )`.

```

PrintsHW(P, x):
    for i in range(len(P)):
        if PrintsHWByK(P, x, i):
            return true
    return false

```

(c) **Computable.** You can simply run the program until  $k$  steps are executed. If  $P$  has printed “Hello World!” by then, return true. Else, return false.

The reason that part (b) is uncomputable while part (c) is computable is that it’s not possible to determine if we ever execute a specific line because this depends on the logic of the program, but the number of computer instructions can be counted.

## 4 Code Reachability

**Note 12** Consider triplets  $(M, x, L)$  where

- $M$  is a Java program
- $x$  is some input
- $L$  is an integer

and the question of: if we execute  $M(x)$ , do we ever hit line  $L$ ?

Prove this problem is undecidable.

**Solution:** Suppose we had a procedure that could decide the above; call it `Reachable( $M, x, L$ )`. Consider the following example of a program deciding whether  $P(x)$  halts:

```

def Halt(P, x):
    def M(t):
        run P(x)    # line 1 of M
        return      # line 2 of M
    return Reachable(M, 0, 2)

```

Program  $M$  reaches line 2 if and only if  $P(x)$  halted. Thus, we have implemented a solution to the halting problem — contradiction.