HW 07 (Optional)

1 Unprogrammable Programs

Note 12

Prove whether the programs described below can exist or not.

(a) A program $P(F,x,y)$ that returns true if the program F outputs y when given x as input (i.e. $F(x) = y$) and false otherwise.

(b) A program P that takes two programs F and G as arguments, and returns true if for all inputs x, F halts on x iff G halts on x (and returns false if this equivalence is not always true).

Hint: Use P to solve the halting problem, and consider defining two subroutines to pass in to P, where one of the subroutines always loops.

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 $P(Q,x,0)$ 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 boils down 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)`.

2 Realign

Note 10 AC Transit is reworking their bus numbering system and they need your help. Bus numbers can fit into one of three categories:

- A letter route, containing either one or two letters (eg. F or NL). If there are two letters, they must be distinct (FF would not be valid).
- A numbered route with a three digit number (eg. 052 or 800).
- A number or letter route with one of eight suffixes appended: A, B, L, M, R, S, T and X.

With this numbering system, what is the maximum number of routes AC transit can run?

Make sure not to overcount!

Solution: There are 26 letters in the alphabet. Therefore there are $26 * 25 + 26 = 676$ possible letter route names.

There are 10 possible digits. Therefore there are $10^3 = 1,000$ possible numbered route names.

There are eight suffixes and the option of no suffix, therefore there are $(676 + 10^3) * 9 = 15,084$ route names with suffixes.

However, this number over counts single letter routes with a suffix (eg. FX). As these would already be included in the 676. We can subtract $26 * 8$ to account for this and then add back 8 due to the distinct constraint (LL is valid if L is a suffix).

$15,084 - 26 * 8 + 8 = 14,884$

3 Unions and Intersections

Note 11 Given:

- X is a countable, non-empty set. For all $i \in X$, A_i is an uncountable set.
- Y is an uncountable set. For all $i \in Y$, B_i is a countable set.

For each of the following, decide if the expression is "Always countable", "Always uncountable", "Sometimes countable, Sometimes uncountable".

For the "Always" cases, prove your claim. For the "Sometimes" case, provide two examples – one where the expression is countable, and one where the expression is uncountable.

(a) $X \cap Y$

(b) $X \cup Y$

(c) $\bigcup_{i \in X} A_i$

(d) $\bigcap_{i \in X} A_i$

(e) $\bigcup_{i \in Y} B_i$

(f) $\bigcap_{i \in Y} B_i$

Solution:

(a) Always countable. $X \cap Y$ is a subset of X, which is countable.

(b) Always uncountable. $X \cup Y$ is a superset of Y, which is uncountable.

(c) Always uncountable. Let x be any element of X. A_x is uncountable. Thus, $\bigcup_{i \in X} A_i$, a superset of A_x, is uncountable.

(d) Sometimes countable, sometimes uncountable.

Countable: When the A_i are disjoint, the intersection is empty, and thus countable. For example, let $X = \mathbb{N}$, let $A_i = \{i\} \times \mathbb{R} = \{(i,x) \mid x \in \mathbb{R}\}$. Then, $\bigcap_{i \in X} A_i = \varnothing$.

Uncountable: When the A_i are identical, the intersection is uncountable. Let $X = \mathbb{N}$, let $A_i = \mathbb{R}$ for all i. $\bigcap_{i \in X} A_i = \mathbb{R}$ is uncountable.

(e) Sometimes countable, sometimes uncountable.

Countable: Make all the B_i identical. For example, let $Y = \mathbb{R}$, and $B_i = \mathbb{N}$. Then, $\bigcup_{i \in Y} B_i = \mathbb{N}$ is countable.

Uncountable: Let $Y = \mathbb{R}$. Let $B_i = \{i\}$. Then, $\bigcup_{i \in Y} B_i = \mathbb{R}$ is uncountable.

(f) Always countable. Let y be any element of Y. B_y is countable. Thus, $\bigcap_{i \in Y} B_i$, a subset of B_y, is also countable.

4 Hilbert's Hotel

Note 11

You don't have any summer plans, so you decide to spend a few months working for a magical hotel with a countably infinite number of rooms. The rooms are numbered according to the natural numbers, and all the rooms are currently occupied. Assume that guests don't mind being moved from their current room to a new one, so long as they can get to the new room in a finite amount of time (i.e. guests can't be moved into a room infinitely far from their current one).

(a) A new guest arrives at the hotel. All the current rooms are full, but your manager has told you never to turn away a guest. How could you accommodate the new guest by shuffling other guests around? What if you instead had k guests arrive, for some fixed, positive $k \in \mathbb{Z}$?

(b) Unfortunately, just after you've figured out how to accommodate your first wave of guests, a countably infinite number of guests arrives in town on an infinitely long train. The guests on the train are sitting in seats numbered according to the natural numbers. How could you accommodate all the new guests?

(c) Thanks to a (literally) endless stream of positive TripAdvisor reviews, word of the infinite hotel gets around quickly. Soon enough you find out that a countably infinite number of trains have arrived in town. Each is of infinite length, and carries a countably infinite number of passengers. How would you accommodate all the new passengers?

Solution:

(a) Shift all guests into the room number that is k greater than their current room number. So for a guest in room i move him/her to room $i+k$. Then place the k new guests in the k first rooms in the hotel which will now be unoccupied.

(b) Place all existing guests in room $2i$ where i is their current room number. Place all the new guests in room $2j+1$ where j is their seat number on the train.

(c) **Solution 1**: We first set up a bijection between the newly arriving guests and the set $\mathbb{N}\times\mathbb{N}$. Notice that each guest has an "address": his/her train number i and his/her seat number j. Let this guest be mapped to (i,j). It is clear that this is a bijection.

We know from Lecture Note 10 that the set $\mathbb{N}\times\mathbb{N}$ is countable (via the spiral method) and hence there is a bijection from $\mathbb{N}^2$ to $\mathbb{N}$. Thus the newly arriving guests can be enumerated and considered as if arriving in a single infinite length train with their corresponding seat numbers given by the enumeration. This reduces to the same exact problem as the previous part! Therefore, we can accommodate these guests.

Solution 2: Place all existing guests in room 2^i where i is their current room number. Assign the $(k+2)$th prime, p_{k+2}, to the kth train (e.g. the 0th train will be assigned the 2nd prime, 3). We then place each new guest in room p_{k+2}^{j+1}, where j is the seat number of the new guest on that train.

This works because any power of a prime p will not have any prime factors other than p.

Yes, there will be plenty of empty rooms, but that's okay because every guest will still have somewhere to stay.