CS 1511 Homework 5

Mathew Varughese, Justin Kramer, Zach Smith

Wednesday, Jan 30

9.

If a C program can output x and another C program can output y, a program can be made which calls both of these programs in order. Effectively, this would output xy. Here, c would be the bits required to create this program that calls the two other programs.

10. Imagine a tree where each node has an infinite number of children. It is still possible to recursively enumerate over all of these.

Picture a tree as follows:

```
        A                       B                        C ...
   D    E    F ...         G    H    I ...          J    K    L ...
M  N  O ...  P  Q  R ...  S  T  U ...
```

Here, A B C make up the first level and the first level is infinitely long. D E F are children of A and A has infinite children. G H I are children of B and B has infinite children. J K L are children of C and C has infinite children. M N O are children of D and D has infinite children. P Q R are children of E, etc. M N O P Q R S T U and so on all also have infinite children.

The idea behind counting them is similar to the proof for prooving that the rational numbers are countable. Although the above tree is hard to visualize, it will be sufficient for the explanation.

The pattern is as follows: Start at A. Then next, go to B. Next, go to A's first child, D. Then go to M. Then go to C. Then go to G. Then go to E. Then go to P. Then go to N. Then go to Ms child. Then go to sibling of C. Then go to the first child of C (J). Then, now you are on the second level, so go back to H and E. Now, go to G's first child. Then that first child. Then go back and do the second child of everything that was touched.

It makes more sense to visualize our explanation with a relation to counting $\mathbb{N} \times \mathbb{N}$.

Drew this out on the back.

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

In the table, each cell represents the all of the children of the parent above. So, say the first cell (first row first column) represents A (the first node in the tree). The first column second row represents **all** of the possible children of A's children on the second level. Every time we "visit" a cell, we go retrace the path and go back through all of the previously visited cells and visit the next node.

So, in the above tree example the order would be: A, B, D, M, *E*, G, *N*, *F*, C, *H*, *O*, *Fs sibling* . . .

The italic text represents the nodes visited when "tracing back".