CSC 395 - Exam 1

Khoa Ho

October 6, 2017

Problem 1

For this problem, I modified a python script based on `carpet.py` of Titus Klinge. Besides changes to the `print` statements to work on Python 3, the main modification was to change the number that the sum $x + y + z$ mods with, from 3 to 5. The following snippet is an excerpt of the script.

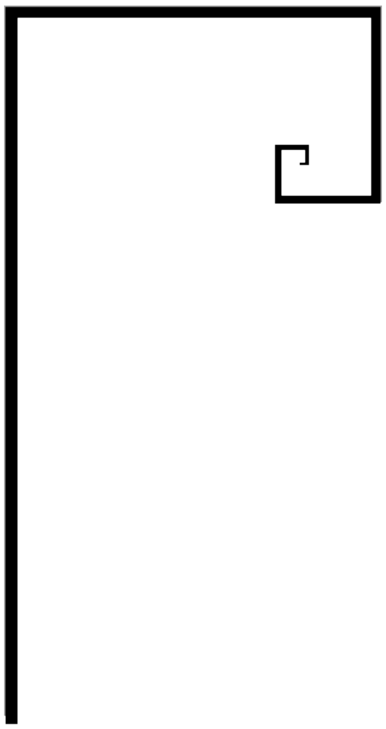
```
def print_mod5_tile(x, y, z):
    name = str(x) + str(y) + str(z)
    left = str(x)
    bot = "(" + str(y) + "," + str(z) + ")"
    w = (x + y + z) % 5
    right = str(w)
    top = "(" + str(x) + "," + str(w) + ")"
    color = "white" if w == 0 else "red"
    label = str(w)
    print_tile(name, label, 1, 1, 1, 1, top, right, bot, left, color)

for x in range(5):
    for y in range(5):
        for z in range(5):
            print_mod5_tile(x, y, z);
```

Problem 2

The main part of this problem was to self-assemble a $N \times 2^N$ rectangle, followed by a $N \times N$ square that allows a $(N-1) \times 2^{N-1}$ rectangle to grow from the top right part of the rectangle-square complex.

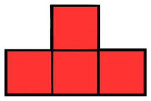
To start, I first used a zig-zag binary counter from the paper, *The Program-Size Complexity of Self-Assembled Squares* (Rothemund, Winfree, 2000). Zigzagging, instead of growing from double bonds on a single edge, allows a controlled growth of the binary counter. However, the increment happens every two rows instead of one, so to get 2^N rows, I starts with 1 follows by $N-1$ zeros and ends with N zeros (with a carry to the next significant bit). Thus, I still get 2^N rows in the end.

Next part of the complex is the $N \times N$ square. To start, the most-significant-zero-bit tile of the rectangle has a double bond, allowing an A tile to attach. Next, I used the same "stair-case" strategy from Assignment 1, where a B tile attaches to the right of an A tile while having a double bond in the North, allowing the next A tile to attach. To make the "turn", i.e. making the base for the next complex, I hard-coded a tile with the East glue strength of 0 to attach on top of the least-significant-zero-bit tile in the top row of the rectangle. This makes the seed row of the next complex smaller by 1. Then, I made different right-edge tiles for the square to signal different seed types (R, S, L). The rest is just adding blank tiles and some additional tiles for special cases ($N = 2$ and $N = 3$). The following figure is the final assembly for $N = 9$.



Problem 3

Suppose we have the following fully-connected shape, composed of 4 tiles.



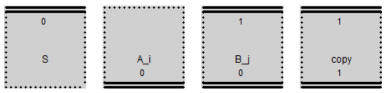
Let A, B, C be the name of each bottom tile from left to right and D be the name of the top tile. For a single-seeded TAS to assemble this shape, all glues need to have strength of 2 regardless of the starting seed. It's clear that there is no totally deterministic TAS that can strictly assemble this shape regardless of the seed we choose.

Suppose we choose A, C, or D as our seed, the only tile that can attach next is B, the center tile. Now, even if we hard-code the glue labels, there is no way to know which of the remaining tiles will attach, in other word, there is no unique assembly sequence. On the other hand, if we start from B, any of the remaining 3 tiles can attach next, so we have the same problem. Thus, we've shown that there is a finite, fully connected shape, $S \subseteq \mathbb{Z}^2$ that cannot be strictly self-assembled by a single-seeded totally deterministic TAS.

Problem 4

To prove that for any rational $p \in \mathbb{Q}$ satisfying $0 \leq p \leq 1$, there is a single seeded TAS $T_p = (T_p, \sigma_p, 2)$ that with probability p will result in a finite terminal assembly and with probability $1p$ result in an infinite terminal assembly, I will show a strategy to come up with a specific TAS for each p.

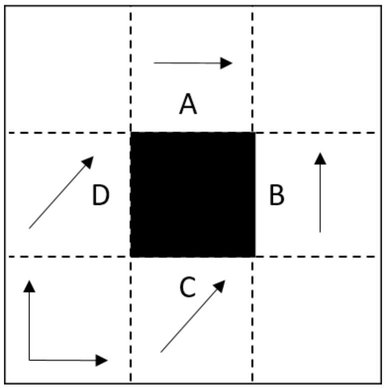
For any $p \in \mathbb{Q}$, we can always express it as a fraction of two integers $\frac{m}{n}$. Consider a TAS where the a portion of its tile set is below and the seed is tile S.



In the complete tile set for a specific $p = \frac{m}{n}$, there are m tile A_is and $n - m$ tile B_js, where i goes from 0 to m and j from 0 to $n - m$. At the start of the assembly, only tile A_i or B_j can attach to the seed tile, so there are exactly n tiles that can attach. Thus, since the probability of choosing a specific tile has a uniform distribution, the probability that an A_i tile attaches is $P(A_i) = \frac{m}{n} = p$, and similarly, $P(B_j) = \frac{n-m}{n} = 1 - p$. Moreover, if any tile A_i attaches, the assembly will stop, and if any tile B_j attaches, the assembly will continue indefinitely due to the infinite attachment of *copy* tiles. In short, we've proven the result.

Problem 5

To see why the Sierpinski carpet cannot be strictly self-assembled by any single-seeded, temperature 2, TAS that is less than 3-directional, we first consider the following figure.

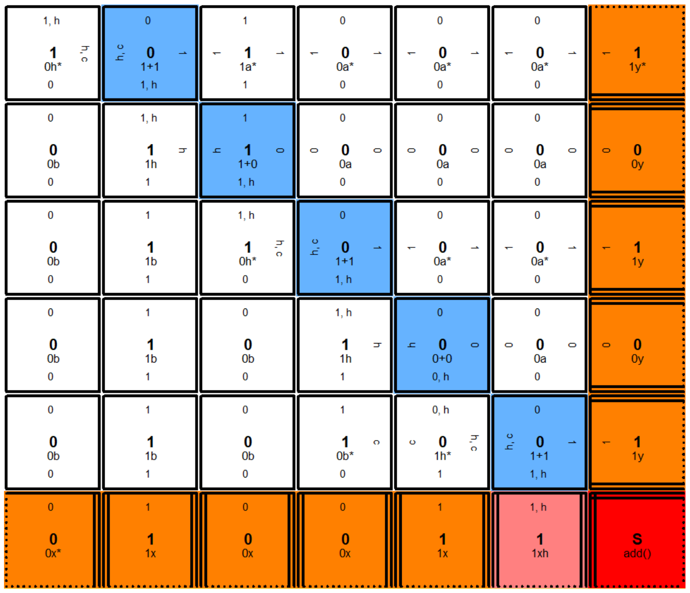


suppose this is a magnified squared portion of the carpet, centering on a hole. Let's subdivide this portion into 9 smaller squares (each square can have an arbitrary number of tiles and holes not shown here). To prove by contradiction, let's assume this TAS is only 2-directional. It's obvious that we can grow the bottom left square, square C, D, and the two squares on the right and on top of them respectively. This is because every tile in these squares receives signal from the bottom and from the left, enough information to to compute the mod function.

However, if we consider the bottom edge of square A, it's clear that the tiles there only receive information from the left, because this TAS is 2-directional and the information flow from the bottom is blocked by the hole. This means every tile on that edge attaches using double bonds, or glue of strength 2. Furthermore, the tiles have to be unique because if any tile from the ongoing line can attach back to the line, the line will grow indefinitely. So for any square A of length L (measured by the number of tiles for each edge), we need L unique tiles. This is problematic because A can have arbitrary size, up to infinity, implying we need infinite number of tiles in the tile set. Thus, we have a contradiction.

Problem 6

For this problem, I created a binary adder so that given a seed column representing a binary number y and a seed row for x, the result of adding these two numbers will be calculated row by row and eventually be displayed as the top row of a rectangle. There are some specifications regarding the use of my TAS. First, the two numbers need to have the same number of digits, and if one number is shorter than the other, zeros have to be added in front (or on top) of the most significant bit of the shorter one until they have the same length. Following is the final assembly of adding 10101 to 10011.



My general strategy is to add bits at the same significant level from each binary number. Specifically, bits from y will be passed horizontally from the seed column to the left, and bits from x will be passed vertically up from the seed row. There is an *add* tile (blue) that moves diagonally to add bits. The carry, if any, will be passed to the left of the *add* tiles.