Automata Theory (CS1.302)

Monsoon 2021, IIIT Hyderabad
Programming Assignment 1 Report

Abhinav S Menon

Question 1

The code related to part n of question 1 is in the Haskell file `part<n>.hs` in the `q1/src` directory.

The `q1/images` folder contains all the images in `.svg` format (as this is how the Haskell library provided renders the images). They are saved as `part<n>.svg` in case of parts 1, 2 and 4, and `part3_<k>.svg` in case of part 3 (where $k = 1, \dots, 5$).

Question 2

Part 1

First, I went backwards and tried to predict what the 0^{th} and 1^{st} iterations of the fractal would have been.
I then noticed that whatever part of the fractal was present up to iteration n is never removed afterwards; parts are only added to it. Therefore, I decided that in the rules, there should be dummy variables representing the *nodes* of the diagram, and not the lines.
Furthermore, the largest branch of the main node has the same shape as the full shape of the previous iteration, and the second-largest has the same shape as the iteration two places previously and so on. Thus the replacement for the nodes is a branch identical to the full structure of the current iteration, and two new nodes must also be created.

Thus, iteration had however many nodes as there were *new branches* in the next iteration. The 0^{th} iteration thus had the axiom `FXFYF` (where `X` represents a branch to the right, and `Y` one to the left).
Then `X` is replaced by two branches and a new node, as described above, which gives us `X -> [+FXFYF]FXFY` and, correspondingly `Y -> [-FXFYF]FXFY`.

Part 2

Here, the fractal is created by *replacing* parts of it by self-similar parts, which makes it different from the fractal in part 1. Thus, here we need a replacement for the variable `F` specifically.

Going backwards, I saw that the 0^th$ iteration must have been simply a straight line. Further, in every iteration, all straight lines are replaced with exactly the same structure. Thus, only one rule was sufficient.

This gave the axiom `F` and the single rule `F -> F-F++F-F`, producing the Koch snowflake.