Haskell Project

Jan Lucca Thümmel - jathu21@student.sdu.dk

December 1, 2023

Contents

1 Introduction

The goal of the project is to draw fractals using L-Systems and turtle graphics in Haskell. By solving three specific tasks. These task are applying rules to a state, going to target depth and processing a fractal

2 Specification

2.1 Task 1: Applying Rules to a State

Task 1 is solved by implementing the apply function. The apply function takes a state and a list of rules and returns a state. A state is defined by a list of characters. A rule is defined by a character and a state. the apply function takes a state and rules, checking wether a character in the state has a specific rule. If a character has no rule the next character is checked. When a character has a rule the character gets replaced with the state of the rule. As an example, we have the rules: 'X' "XRYF", 'Y' "FXLY" and the state "FXRYF". In this case the apply function would replace every X with "XRYF" and every Y with "FXLY" resulting in the string "FXRYFRFXLYF".

2.2 Task 2: Going to Target Depth

Task 2 is solved by implementing the function expand. The expand function takes a state, a list of rules, a target depth in form of an integer and returns a state. The function will apply the apply function to the state until reaching the target depth and then return the result state. An example could be applying these rules: 'X' "XRYF", 'Y' "FXLY" to the state "FX" with depth two. depth one is achieved by applying apply once, giving the string "FXRYF". Depth two is achieved by applying apply again on the new string. Resulting in the new string "FXRYFRFXLYF". This can be repeated until the target depth is reached.

2.3 Task 3: Processing a Fractal

Task 3 is solved by implementing the function process. The process function takes a fractal and produces a list of commands to draw the fractal. This is achieved by using the expand function to get the state at target depth. Then the turtle graphics commands are mapped to the state. The commands are produced by the charToCom function which converts a character to a turtel command. Testing the function with the main command in GHCI should produce a white on black snowflake

3 Design

4 Implementation

5 Testing

6 Conclusion

7 Appendix

7.1 Code

7.2 Fractals