COMP2209: Programming III
Coursework: Haskell Programming Challenges

Daniel Braghis (ID: 32161204)

January 12, 2023

1. Report

This report outlines the tools and techniques used to complete the coursework assignment.

For the choice of development environment, I continued using IntelliJ IDEA 2022.1.4 connected to the remote machine provided by the university. This approach ensured the code is kept in the more reliable university machines (besides GitHub repository) instead of my personal computer with no backups.

IntelliJ integrates well remote files, letting development and testing in the same environment (one I am very comfortable with). Throughout the development I used two main tabs – one for the code and another for the console used for testing. This setup proved very efficient, but I lost some Haskell quality of life improvements because I couldn't use IntelliJ plugins with this setup.

As a general software development process, I opted for Test-driven development. Each challenge would start with writing the test cases before development starting with intermediate functions if those were evident, basic examples and then adapting the examples given in the specification. Besides these, I would occasionally add edge cases identified while developing and an empty input test where relevant.

After finishing each challenge, I removed the intermediate function tests to not overcrowd the already large test file output with irrelevant information. Having only top-level tests in the suite means that if a bug causes one test to fail, any future developer will have to write new tests to identify the actual function that causes the bug. This is not an issue in my opinion because each challenge has only a few main functions with complex behaviours and the issue cause can be narrowed down further by looking at the pattern matching used extensively.

Finally, testing is done in a rudimentary way, without using a testing library, because my concerns around setting up a testing library in the remote machine environment, opting to dedicate more time to actual development. Function output is compared with the expected output and a True/False output is printed. While this approach is reasonable for these smaller challenges, I would use a Haskell testing library for larger codebases.

2. Bibliography

1. Kaan Dedeoglu (2014), Check if two lists have the same elements [StackOverflow] 1 Jun. Available at: https://stackoverflow.com/questions/23978834/check-if-two-lists-have-the-same-elements (Accessed: 27 December 2022)

2. Willem Van Onsem (2021), How do I find the intersection of two lists in Haskell? [StackOverflow] 4 Oct. Available at: https://stackoverflow.com/questions/69442683/how-do-i-find-the-intersection-of-two-lists-in-haskell (Accessed: 27 December 2022)

3. Cartesian product of two or more lists. Available at: https://rosettacode.org/wiki/Cartesian_product_of_two_or_more_lists#Haskell (Accessed: 10 January 2022)

4. Rathke J, 2022, Lecture 38: Implementing Beta Reduction, lecture notes, COMP2209 - Programming III, University of Southampton

5. Rathke J, 2022, Lecture 45: Monadic Parsing, lecture notes, COMP2209 - Programming III, University of Southampton