

Лабораторна робота № 2

з дисципліни

”Функційне програмування мовою Haskell”

18 лютого 2023 р.

1 Робота № 2

Функції вищого порядку

1.1 Мета роботи

Набути досвіду визначення та використання функцій вищого порядку.

1.2 Функції вищого порядку

Функції вищого порядку можуть приймати функції як параметри, або повертати функцію як результат.

Наприклад функція `map`, як це видно з сигнатури `map :: (a -> b) -> [a] -> [b]`, приймає два параметри: функцію з сигнатурою `a -> b` та список з сигнатурою `[a]`.

Інший приклад – функція `curry` з сигнатурою типу `curry :: ((a, b) -> c) -> a -> b -> c`, яку, зважаючи на правоасоціативність `->` можна переписати у формі `curry :: ((a, b) -> c) -> (a -> b -> c)`. Тоді можна говорити, що параметром функції `curry` є функція з сигнатурою `(a, b) -> c`, а результатом — функція з сигнатурою `a -> b -> c`.

Часто кажуть, що будь-яку функцію кількох аргументів можна вважати функцією вищого порядку. Справді, наприклад сигнатуру `fnc :: a -> b -> c` можна переписати у формі `fnc :: a -> (b -> c)` і говорити, що функція `fnc` приймає аргумент типу `a` і повертає функцію типу `b -> c`.

1.3 Завдання

Визначте функції вищого порядку для виконання вказаних завдань. За необхідності завдання можна розширити. Наприклад у завданні 1.1 результат повертати у формі кортежів `('a', 3)` або `(3, 'a')`, або `('a', 3, "a3")` тощо.

Визначте вказані функції в кожному з завдань: а) без застосування, б) із застосуванням вбудованих функцій (визначених у модулі `Prelude`).

Наведіть приклади застосування розроблених функцій для обробки не одного, а кількох наборів вхідних даних. Наприклад у завданні 1.1 — для ["aaabbcaadddd", "2222133333ddd", "00022213888833DDDDFF"]

1. 1.1 Визначити довжину кожної неперервної послідовності тотожних елементів списку, напр.: "aaabbcaadddd" \Rightarrow [(*'a'*,3), (*'b'*,2), (*'c'*,1), (*'a'*,2), (*'d'*,4)].
- 1.2 Видалити повтори елементів списку, напр.: [1,1,1,5,5,3, 1,1,222,222,222,222] \Rightarrow [1,5,3,222].
- 1.3 Визначити частоту кожного елемента списку, напр.: "aaabbcaadddd" \Rightarrow [(*'a'*,5), (*'b'*,2), (*'c'*,1), (*'d'*,4)].
- 1.4 Повторити n -кратно кожен елемент списку, напр. при $n=2$: "asd" \Rightarrow [*'a'*, *'a'*, *'s'*, *'s'*, *'d'*, *'d'*].
- 1.5 Вставити у список через кожні n елементів вказане значення, напр. через $n=2$ значення *'a'*: "1234590" \Rightarrow "12a34a59a0".
- 1.6 Видалити кожен n -й елемент списку, напр. при $n=2$: "1234590" \Rightarrow "1350".
- 1.7 Замінити кожен n -й елемент списку вказаним значенням, напр. при $n=2$ та значенні *'a'*: "1234590" \Rightarrow "1a3a5a0".
- 1.8 Розділити список на дві частини при заданій n довжині першої, напр. при $n=3$: "abcdefghik" \Rightarrow ("abc", "defghik").
- 1.9 Знайти останній елемент списку.
- 1.10 Знайти передостанній елемент списку.
- 1.11 Знайти k -й справа елемент списку.
- 1.12 Переписати список справа наліво.
- 1.13 Чи є список паліндромом?
- 1.14 Понизити рівень багаторівневого списку, напр.: [[1..2],[11..12],[111..112]] \Rightarrow [1,2,11,12,111,112].
- 1.15 Кожну неперервну послідовність тотожних елементів списку замінити одним елементом, напр.: [1,1,1,5,5,3, 1,1,222,222,222,222] \Rightarrow [1,5,3,1,222].
2. 2.1 Циклічний лівий зсув списку на n позицій.
- 2.2 Об'єднання зі змішуванням двох списків довжиною n_1 та n_2 . Вихідний список має довжину $2 * n$, де $n = \min(n_1, n_2)$. Наприклад "abcde" та "123" перетворюються на "a1b2c3".
- 2.3 Визначити, чи є число простим.
- 2.4 Знайти найбільший спільний дільник двох чисел.
- 2.5 Обчислити функцію Ейлера $\varphi(m)$.
- 2.6 Знайти перше просте число у вказаному діапазоні.
- 2.7 Знайти усі прості числа в вказаному діапазоні.

- 2.8 Знайти два прості числа, сума яких дорівнює заданому парному $N \geq 2$ (бінарна проблема/гіпотеза Гольдбаха).
- 2.9 Перевірити гіпотезу Гольдбаха у вказаному діапазоні.
- 2.10 Циклічний правий зсув списку на n позицій.
- 2.11 Перемішування списку: після першого розмістити останній, після другого – передостанній тощо. Наприклад вхідний список "abcde1234" має бути трансформований у "a4b3c2d1e".
- 2.12 Визначити, чи два числа взаємно прості.
- 2.13 Знайти найменше спільне кратне двох чисел.
- 2.14 Знайти прості дільники числа.
- 2.15 Видалити зі списку елементи з i -го по k -й включно, напр. при $i=2$ та $k=4$: "asdfghj" \Rightarrow "aghj".

1.4 Порядок виконання роботи

Робота передбачає виконання двох завдань – 1.N та 2.N, де N – номер студента у списку підгрупи.

Нехай номер у списку підгрупи – 1. Тоді треба виконати завдання 1.1 у двох варіантах – а) та б) і завдання 2.1 також у двох варіантах – а) та б).

1.5 Теми для обговорення

Функції вищого порядку:

- 1. визначення;
- 2. частково визначені функції;
- 3. карринг;
- 4. секції (section);
- 5. оператор застосування (аплікації) $\$$;
- 6. оператор композиції $.$ (крапка);
- 7. λ -функції і функції вищого порядку;
- 8. функції вищого порядку модуля `Prelude`.

1.6 Корисні посилання

- 1. <https://www.haskell.org/>
- 2. Introductory Books for Learning Haskell. <https://www.haskell.org/documentation/>
- 3. <http://learnyouahaskell.com/higher-order-functions>
- 4. Модуль `Prelude` у довідці `GHC` за адресою на комп'ютері:
шлях_до_ghc/doc/html/libraries/base-4.9.1.0/Prelude.html