P. Thiemann, G. Radanne Wintersemester 2019/20

Functional Programming

http://proglang.informatik.uni-freiburg.de/teaching/functional-programming/2019/

Exercise Sheet 2 – Recursion, Datatypes

2019-05-22

Exercise 1 (Fib, or "Hello World! of Functional Programming")
Define the function which, for any natural number n, computes the n -the Fibunacci number:

$$\text{fib}(n) = \begin{cases} 0 & \text{for } n = 0 \\ 1 & \text{for } n = 1 \\ \text{fib}(n-1) + \text{fib}(n-2) & \text{otherwise} \end{cases}$$

Try to compute fib(30). How long does it take ?

Define an optimized version that is at least capable of calculating the fib(20000) in a reasonable time. Test the optimized version using the naive specification. (You must limit the test size, see lecture).

Exercise 2 (Undup)
Define and test a function `undup`, which removes duplicated elements in a list. (There is already a function `Data.List.nub` in the standard libraries which implements this functionality . . . which you should not use)

Note: If you get type errors when testing properties with QuickCheck, first try to add a type signature to the properties.

Exercise 3 (Smallest factor)
A natural number $k \geq 2$ is *factor* of a number n if there exists m such that $n = k \cdot m$. Define a function that computes the smallest factor of a given `Int`.

Try to solve the tasks in two different ways and test them against each other.

Exercise 4 (Media Library)
Many music playback programs, such as iTunes, Windows Media Player, or Amarok, have an integrated media library that organizes the available using metadata such as popularity or number of plays. The goal of this exercise is to design the data model for a mini-media library and implement simple queries.

Our library saves the title, the artist and the duration (in seconds) for each song. Furthermore, songs can be combined into albums. Additionally, each user of the library can rate songs (either "good" or "bad").

1. Define appropriate data types and type aliases for the media library.
2. Define the following operations on the Media Library:
   - `addAlbum`: adds a song to an album. One possible type would be:
     `addAlbum :: Track -> Album -> MediaBib -> MediaBib`
   - `rateTrack`: rate a song for a particular user. One possible type would be:
     `rateTrack :: User -> Track -> Rating -> MediaBib -> MediaBib`
3. Define the following queries on the library:
   - Return all albums and their durations.
   - For a particular user, return all albums where more than 50% of the songs are well rated.

To make testing easier, you can find a file `Tracks.hs` containing lists of songs in the format `(Maybe Album, Title, Artist, Duration)` on the lecture home page. You can then translate these into your data model.

Exercise 5 (Tic-Tac-Toe)
Tic-Tac-Toe (`http://en.wikipedia.org/wiki/Tic-tac-toe`) is a popular game in computer science, mainly because of its simplicity.

1. Design a data model for Tic-Tac-Toe.
2. Define functions that determine whether a game is in progress, won, or invalid. If a game is won, then by whom?