PLC: Homework 3 [100 points, 120 max]

Deadline

If the first letter of your HawkId is between 'a' and 'k' inclusive, then your solution is due by 9pm Wednesday, February 19th. Otherwise, it is due by 9pm Friday, February 21st. As mentioned in class, students with HawkIds between 'a' and 'k' inclusive will be submitting Wednesdays before Spring Break, and then they will be submitting Fridays after Spring Break.

Remember that late homework is **not** accepted, but we will use second-chance grading as for hw1.

Subversion, Partners

The same instructions as for hw1 and hw2 apply for turning in your solution via subversion, and for extra files you need to submit if you are working with a single partner. Second-chance grading as for hw1 and hw2 applies to this assignment.

For this assignment, you will check in a file to the `hw3` subdirectory of your personal repository (see hw1 instructions for how to check out this repository).

1 Reading

Chapter 4 of the book is not written yet, but should be ready or at least underway by Friday Feb 14th. You can find the book as `book/book.pdf` in the class repository, or directly here:

`https://svn.divms.uiowa.edu/repos/clc/class/111-spring14/book/book.pdf`

2 Problems

There are two sets of problems you can tackle in this assignment. There are some theorems about lists in `hw/hw3/list-todo.agda` in the class repository. These are only worth a total of 50 points. There are also over 100 points worth of problems in `hw/hw3/rle.agda`. You are free to do any combination of these problems you wish, for 120 maximum points (we will not award more than this, though you are free to do more problems if you wish). 100 points is a perfect score. See the files for the number of points that are given for the different problems.

The problems in `rle.agda` concern *run-length encoding*. The basic idea is to compress a list of values by replacing repeated sequences like `a :: a :: a` with just `(3,a)`. Actually, we are working in a very simple case of run-length encoding, where the values are either `tt` or `ff`. In that case, all we have to do is store the starting value, and then a list of numbers corresponding to each maximal subsequence of the same value. For example, we will compress the list

```
ff :: tt :: tt :: tt :: ff :: ff :: []
```

to just

```
(nonempty-run ff 0 (2 :: 1 :: []))
```

This uses the `nonempty-run` constructor for the `run` datatype declared in `rle.agda`. The `empty-run` constructor is for encoding the empty list. The idea for decoding a nonempty run like

```
(nonempty-run b n ns)
```

is that `b` is the boolean value of the next run, `n` is the length of the first run (actually, the length will be `n` plus 1), and then `ns` is the list of the lengths of the following runs in the sequence. Decoding will produce a run with $n + 1$ copies of the boolean value b, and then switch to decode `ns` with boolean value `~ b`. We use $n+1$ to avoid having the `decode` function produce empty subsequences if the run-length is 0.

You are free to use any functions or theorems from the library or that we proved in class or previous homeworks. I will soon be updating the library with solutions from your hw1, so those theorems will be available if you want them.

Very important: as for previous homeworks, for any holes you choose not to fill in, you should remove those problems from the file before you submit. In the end, we are requiring the file you submit to check in Agda, and not to contain any holes. Also, it should not contain any highlighting. Your file may contain other Agda code or comments if you like. Do not change the names of any of the functions or theorems in the file.

3 Grading, Help

As for hw1, we will first confirm that your file can be checked in Agda, and that it does not contain any holes. Then we will determine which problems you solved, in order to compute your grade. For holes 0 and 4 (the definitions of the `decode` and `encode` functions), we may apply a couple additional test cases to determine if your code is correct. For the other holes, we will just check that what you proved indeed has the desired type.

As for previous homeworks, you can post questions in the `hw3` section on Piazza. As before, feel free to post anonymously or privately, though We may subsequently make your question (and our answer) public, if we feel it would help other students. For truly private questions about the homework or class in general, please just email us.

You are also welcome to come to our office hours. See the "Course Staff" section of the Piazza page for times and locations of office hours, which we may change at the start of the week.