

# Haskell code

Lorenzo Fundaró García

November 13, 2010

## 1 Parsec Tool - lhsParser.hs

A “Literate Haskell” file (.lhs) contains Haskell code along with any arbitrary text. This exercise consist of transforming a .lhs file into an HTML. To do so, the following conventions are used:

- Any line starting with:
  - \* will be treated as a main header.
  - # will be treated as a secondary header.
- A paragraph ends when a return character is found.
- If a line starts exactly with > followed by a blank space, we can assume that the rest of the text is the code. The first two characters shall not be preserved.

Write a program based on the Parsec tool that, given a list of .lhs files, will parse and convert each of them to HTML files.

### 1.1 How to run

```
$ ./lhsParser file.lhs [file2.lhs]*
```

in the folder you will find two lhs files to test the code, those are tarea-01.lhs, tarea-06.lhs.

## 2 The Mexican dinner. A software transactional memory (STM) exercise.

A flauta is a typical food from Mexico. It has the following ingredients:

- A tortilla
- Meat
- Guacamole

In a rounded table there are three mariachis seated, each of them has an infinite supply of his ingredient, one has tortillas, the second one has meat and the third one has guacamole. Since they are a little drunk there's a fourth mariachi making them eat so they can get better. This mariachi chooses arbitrarily (non deterministically) two of the three seated mariachis, then he asks them their ingredients and puts them on the table. Then, he notifies the third mariachi to take the ingredients and combine them with his own so he can make himself a flauta and eat afterwards. Without attending the third one to finish his

flauta, the fourth mariachi chooses again two of the seated mariachis arbitrarily, takes their ingredients and notifies the third one to start eating. If the third one was eating, then when he's finished he makes himself the pending flauta. The mariachis eat at different speeds each time. The fourth mariachi never eats. This process continues indefinitely. Propose a solution for this "Mexican dinner" using STM in a way in which none of the three seated mariachis ends up without eating. Your program shall show the mariachis activity indicating for each made flauta which are the ingredients contributors, who makes it and when is being eaten.

## 2.1 How to run

```
$ ./mexicanDinner
```

## 3 Parallelism

The collapse of an integer corresponds to the sum of its digits, a procedure that can be followed recursively until it collapses to a single digit. For example, the collapse of 134957 is 29, which in turns collapses to 11 and finally to 2. We'll define the collapse of a list of integers as the collapse of the sum of the collapse of each list components. Write a program that calculates the collapse of a list of integers. You are recommended to take advantage of Quickcheck to generate random instances.

## 3.1 How to run

```
$ ./collapse 122 1000000
```

where the first argument is the seed to generate de random instances and the second argument is the size of the list to be collapsed.

## 4 Using the State Monad

You are provided with a text file of unknown size in which you have to select a random line, with uniform probability. That is, if the file has  $n$  lines, a line must be chosen with  $1/n$  probability. Nevertheless, since the file may be too long, it's desirable to go through it once. There is a very simple algorithm to solve this problem: the  $k^{th}$  line is selected with  $1/k$  probability. Write a Haskell program that takes the name of a file as an argument, it opens it, and once it's processed it shows the selected line. The process of selecting a line must be done with monadic style using the aforementioned algorithm. Do not implement yourself

the State Monad, just design the necessary data types to represent the mutable state and take advantage of *Control.Monad.State*.

#### **4.1 How to run**

```
$ ./pickAline file
```