COMP 150PP Homework: Programming Probability

Due Monday, October 3, 2016, at 11:59PM

Here is our revised plan for completing solutions of the dice problems. See `http://www.cs.tufts.edu/comp/150PP/handouts/0914problems2c.pdf`.

- Design or adapt a polymorphic abstraction for probabilistic modeling. In Haskell this would be a type class; in ML it would be a signature.
- Build one or more implementations (exact calculation using exact rational arithmetic, approximate calculation using floating-point numbers, and/or sampling). In Haskell, each implementation would be an instance declaration; in ML, it would be a functor.
- Solve your chosen subset of problems using the abstraction, so you can try multiple implementations. In Haskell, each solution would be a function that has a type-class constraint. In ML, each solution should also be a function, but it makes the most sense to put that function in the body of an ML functor.
- Send me code that I can compile and run to print your chosen solutions. Ideally, the command-line arguments to your code would determine what solutions are run—in Haskell, try `getArgs` from `System.Environment`; in ML, try `CommandLine.arguments`.
- Include a README file that says what problems you've solved *and how long it takes to compute solutions*. If there are problems that hit combinatorics, let me know.

Finally, here is an additional problem for you to think about. We'll get deeper into this problem next month:

M. Somebody gives Norman a hat containing five slips of paper, numbered 1 to 5 respectively. Norman draws a slip from the hat. The number on the slip is called n. Norman then repeats the following procedure ten times:

   - Take n dice from the bag, throw them, report the total t, then put the dice back in the bag.

   The totals reported are 21, 15, 34, 12, 18, 38, 46, 13, 24, and 27. The question is, *what is the number on the slip Norman drew?* (That is, what is n?) We will call this problem the *slip problem*.