COMP 150PP Class Exercise: Abstractions towards inference

September 12, 2016

Towards language design, with semantics

Three questions:

1. How would you design a language for expressing probabilistic computations like the tally-sheet experiment?
2. Use your language to write a program that implements the tally-sheet experiment.
3. Give your language a formal semantics, so that it is clear exactly how the probabilistic computation works. (I recommend an operational semantics, but if you happen to have studied denotational semantics, it is also a possibility.)

Towards implementing inference

General probability distributions

In your language of choice, assume you have an *abstract* type constructor `P`, such that `P a`[1] is the type of probability distributions over values of type `a`. If `D` is the type of a single die and `Col` is `LEFT`, `MIDDLE`, or `RIGHT`, some types from our first class might include

Type	Meaning
`P D`	distribution of a single die
`P (D, D)`	distribution of a pair of dice (in ML, `(d * d) p`)
`P Col`	distribution of a single mark on the tally sheet (`col p`)
`P [Col]`	distribution of a list of marks on the tally sheet (`col list p`)
`P Int`	distribution of number of marks on right column of tally sheet (`int p`)

Questions about creating distributions:

1. What functions can you think of that will introduce new probability distributions?
2. What functions can you think of that will take existing distributions and transform them into new distributions?
3. What functions can you think of that will combine two or more distributions into a single distribution?
4. What functions can you think of that will enable you to combine *dependent* distributions. For example: draw a die from the bag, then throw that die once. What is the resulting distribution over (die, number) pairs? How do you compute it?

Question about once you have a distribution, then what?

5. Given a distribution, what do you think it ought to be reasonable to do with it? (Assume that there are only finitely many values that have nonzero probability.)
6. To solve the tally-sheet inference problem, what distribution would you compute, and what would you do with it?

Probability distributions with finite support

Assuming that only finitely many values have nonzero probability, how would you implement your ideas?

[1] This notation is Haskell notation. An ML programmer would write `a p`, and a C++ programmer would write `P<a>` (or possibly `p<T>`; C++ spelling conventions elude me).