24.118: Paradox and Infinity

Ryan Lacey <rlacey@mit.edu>

`Collaborator(s): Evan Thomas`

March 6, 2014

1. (a) This randomized procedure is the same as the one listed in the lecture notes, from which we draw the fact that selecting a number in the range of [0,k] is k. This is because we have a uniform distribution of probabilities for any infinite series under the given method. We desire the probability of a value being in the range [$m/2^k$, $n/2^k$]. Another way to state this is the probability of getting a value in the range [0, $n/2^k$], excluding the values in the range [0, $m/2^k$]. The probability of a value in [0, $n/2^k$] is $n/2^k$ from what was stated previously. Excluding the values in [0, $m/2^k$] means we are simply removing $m/2^k$ from that probability, which we can just subtract off due to the uniform distribution. Therefore the probability of a value in [$m/2^k$, $n/2^k$] is $n/2^k - m/2^k = \frac{n-m}{2^k}$.

   (b) To have a $\frac{1}{3}$ probability of being above one half we need to simulate a fair three-sided coin. To achieve this we flip two fair coins once. If the coins land `TT` then we flip them again. Else if the coins land `HH` then we assign the beginning of our sequence value `0.1`. Else the coins landed `TH` or `HT` and we assign the beginning of our sequence `0.0`. After this initialization the remaining bits of the sequence are set in the standard manner of 1 for heads and 0 for tails.

   All of the coin combinations have equal probability. Throwing out the possibility for `TT`, then each of the other three combinations has probability $\frac{1}{3}$. Setting `0.1` to head the sequence for `HH` means that with probability $\frac{1}{3}$ we ensure that the output will have value greater than ore equal to one half. Intuitively the remaining options make it so the output is less than one half with probability $\frac{1}{3} + \frac{1}{3} = \frac{2}{3}$.

2. (a) Define a binary string constructed from the coin tosses as described in problem `(1a)`, namely that a head corresponds to a `0` and a tail corresponds to a `1`. The string is prepended with `0.` so that the number constructed is a real positive number up to value one. The side-length output of the cube machine is the result of the infinite coin flips converted into a value [0,1].

   (b) Flip a fair coin once. If its heads, then assign 0.111 to be the beginning of the sequence. Set the bits from the fourth position onward in the standard manner of 1 for heads and 0 for tails. If the first flip was tails, then assign 0.0 to be the beginning of the sequence. Set the bits from the second position onward in the standard manner. The value of 0.111 is $\frac{1}{2} + \frac{1}{4} + \frac{1}{8} = \frac{7}{8}$. Thus with probability $\frac{1}{2}$ we are setting our output to be above or below $\frac{7}{8}$ from the first flip. All subsequent flips merely determine where in the range from $[0, \frac{7}{8}]$ or $[\frac{7}{8}, 1]$ the side length will be.

3. (a) (i) *Reflexive* - It is obvious that f_1 is in the same orbit as f_1. There is a finite number of differences between them, namely zero differences, as they are one and the same.

   (ii) *Symmetric* - If f_1 is in the same orbit as f_2 then they must differ at a finite number of locations. Let us say that the indicies where f_1 differs from f_2 is represented in a list L. We now want to consider the locations in which f_2 differs from f_1. The list that represents these differences is also L and must be of finite length due to how we defined L initially.

   (i) *Transitive* - Let us extend the terminology of `(ii)` to define three lists L_{12}, L_{23}, and L_{13}. The subscripts of each list denote which f's it holds the indexed differences of. Assume that each f is distinct, else we wold have one of the previous two cases and the respective L would be an empty-set. Each L is of finite size due to the nature of its construction. Therefore the union of the lists results in a list that must also be of finite size. At most the size of $L_a \bigcup L_b$ is $|L_a| + |L_b|$. Each f has a finite number of differences from each other f, so must all be in the same orbit.

   (b) Since f_0 consists of an infinite string of 0s, then everything else in its orbit differs by have a finite amount of bits flipped, ie. a finite amounts of 1s. Each function f_i outputs the natural number i, represented in binary notation. For example, if $i = 9$ then the output of f_9 has binary representation 1001 and would differ from f_0 in two locations, indexed (1,4). As each natural number has a unique representation in binary, each f_i is a unique mapping of the natural numbers.

   (c) We take a similar approach to `(3b)`. Arbitrarily pick a function f of O to be f_0 (Axiom of Choice). All other functions f_i output a natural number i represented in binary. At each location in which the output has a 1 we flip the bit of f_0. The result has a finite number of differences of f_0, just the bits that were flipped, and therefore must be a member of O. In this manner we can get a mapping to all natural numbers.