fn sample_bernoulli_rational

Michael Shoemate

October 22, 2024

This document proves that the implementations of sample_bernoulli_rational in mod.rs at commit f5bb719 (outdated[1]) satisfies its proof definition.

0.1 Hoare Triple

Preconditions

Compiler-verified

- Variable prob must be of type RBig, a rational of bignums.

User-verified None

Pseudocode

```
1 def sample_bernoulli_rational(prob: RBig) -> bool:
2     numer, denom = prob.into_parts() #
3     sign, numer = numer.into_parts() #
4     if sign == Negative:
5         raise ValueError("prob must not be negative")
6     if numer > denom:
7         raise ValueError("prob must not be greater than one")
8     return numer > sample_uniform_int_below(denom)
```

Postcondition

Definition 0.1. For any setting of the input parameters, returns an error if there is a lack of system entropy or prob is not in $[0, 1]$, returns true with probability prob, otherwise returns false.

Proof. At 2, into_parts returns the signed numerator and unsigned nonzero denominator of prob. At 3, into_parts returns the sign and magnitude of numer.

An error is raised if the sign is negative (if prob is negative) or if the numerator is greater than the denominator (if prob is greater than one).

Since RBig.into_parts never returns a zero denominator (this would be an invalid rational), the precondition of sample_uniform_ubig_below is met, therefore by the postcondition the return value is a uniform sample in $[0, denom)$.

By countable additivity, the probability that the uniform sample is less than the numerator is exactly prob, as there are numer possible disjoint outcomes each with probability 1/denom. It has been shown that the function returns true with probability prob, otherwise returns false. □

[1] See new changes with git diff f5bb719..516d7d7b rust/src/traits/samplers/bernoulli/mod.rs