fn sample_geometric_buffer

Vicki Xu, Hanwen Zhang, Zachary Ratliff

July 10, 2025

This document proves soundness of `sample_geometric_buffer` in `mod.rs` at commit f5bb719 (outdated[1]).

1 Hoare Triple

Preconditions

None

Pseudocode

```
1  def sample_geometric_buffer(
2      buffer_len: usize, constant_time: bool
3  ) -> Optional[uint]:   #
4      if constant_time:
5          buf = bytearray(buffer_len)
6          fill_bytes(buf)   # mutates in-place
7          ret = None
8          for i in range(buffer_len):
9              # find first nonzero event
10             if buf[i] > 0:
11                 # compute index of first nonzero bit buffer
12                 cand = 8 * i + buf[i].leading_zeroes()   #
13                 ret = cand if ret is None else min(ret, cand)
14         return ret
15     else:
16         for i in range(buffer_len):
17             buf = bytearray(1)
18             fill_bytes(buf)   # mutates in-place
19             if buf[0] > 0:
20                 return 8 * i + buf[0].leading_zeroes()
21
22         return None
```

Postcondition

For any setting of the input arguments, `sample_geometric_buffer` either raises an exception if there is insufficient system entropy, or returns `sample` where `sample` is drawn from a discrete distribution.

`sample` is either `geo` where `geo` is a sample from the $Geometric(p = 0.5)$ distribution, and is less than $buffer\_len * 8$, or `None` with probability $2^{-buffer\_len*8}$.

Proof. `sample_geometric_buffer` uses `fill_bytes` as a subroutine to generate a buffer of `buffer_len` bytes. For each bit b in the buffer it follows that $\Pr[b = 1] = \frac{1}{2}$ and $\Pr[b = 0] = \frac{1}{2}$. If there is some bit in the buffer equal to 1, the position of the *first* such bit is a zero-indexed draw from the Geometric distribution

---

[1]See new changes with `git diff f5bb719..c7d74e0 rust/src/traits/samplers/geometric/mod.rs`

$Geom(p)$ with $p = 0.5$, by definition of a Geometric random variable. If the buffer is zero, the function returns `None`. $\square$