

# fn sample\_geometric\_buffer

Vicki Xu, Hanwen Zhang, Zachary Ratliff

August 19, 2024

This document proves soundness of `sample_geometric_buffer` in `mod.rs` at commit `f5bb719` (outdated<sup>1</sup>).

## 1 Hoare Triple

### Preconditions

None

### Pseudocode

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

---

<sup>1</sup>See new changes with `git diff f5bb719..d0515a7 rust/src/traits/samplers/geometric/mod.rs`

buffer equal to 1, the position of the *first* such bit is a zero-indexed draw from the Geometric distribution  $Geom(p)$  with  $p = 0.5$ , by definition of a Geometric random variable. If the buffer is zero, the function returns **None**. □