fn sample_geometric_exp_slow

Michael Shoemate

December 22, 2025

Proves soundness of `fn sample_geometric_exp_slow` in `mod.rs` at commit 1f9230c (outdated[1]). This proof is adapted from subsection 5.2 of [CKS20].

1 Hoare Triple

Precondition

Compiler-verified

Argument `x` is of type `RBig`, a bignum rational

User-verified

$\texttt{x} > 0$

Pseudocode

```python
def sample_geometric_exp_slow(x) -> int:
    k = 0
    while True:
        if sample_bernoulli_exp(x): #
            k += 1
        else:
            return k
```

Postcondition

Theorem 1.1. For any setting of the input parameter `x` such that the given preconditions hold, `sample_geometric_exp_slow` either returns `Err(e)` due to a lack of system entropy, or `Ok(out)`, where `out` is distributed as $\text{Geometric}(1 - \exp(-x))$.

Definition 1.2. If $K \sim \text{Geometric}(p)$, then for $k \in \{0, 1, \ldots\}$

$$\Pr[K = k] = (1 - p)^k \cdot p. \tag{1}$$

Definition 1.3. If $B \sim \text{Bernoulli}(p)$, then for $b \in \{\top, \bot\}$

$$\Pr[B = b] = \begin{cases} p & b = \top \\ 1 - p & b = \bot \end{cases} \tag{2}$$

[1]See new changes with `git diff 1f9230c..fe1abca rust/src/traits/samplers/cks20/mod.rs`

2 Proof

Assume the preconditions are met.

Lemma 2.1. `sample_geometric_exp_slow` only returns `Err(e)` when there is a lack of system entropy.

Proof. The preconditions on `x` satisfy the preconditions on `sample_bernoulli_exp`, so by its definition, it only returns an error if there is a lack of system entropy. The only source of errors is from this function, therefore `sample_geometric_exp_slow` only returns `Err(e)` when there is a lack of system entropy. □

Theorem 2.2. [CKS20] If the outcome of `sample_geometric_exp_slow` is `Ok(out)`, then `out` is distributed as $\text{Geometric}(1 - \exp(-x))$.

Proof. The distribution of the i^{th} boolean returned on line 4 is $B_i \sim \text{Bernoulli}(\exp(-x))$, because the preconditions on `x` satisfy the preconditions for `sample_bernoulli_exp`.

$$\begin{aligned}
\Pr[\texttt{out} = k] &= \Pr[B_1 = B_2 = ... = B_k = \top \wedge B_{k+1} = \bot] \\
&= \Pr[B_{k+1} = \bot] \prod_{i=1}^{k} \Pr[B_i = \top] && \text{All } B_i \text{ are independent.} \\
&= (1 - \exp(-x)) \exp(-x)^k
\end{aligned}$$

By Definition 1.2, setting $p = 1 - \exp(-x)$, then $\texttt{out} \sim \text{Geometric}(1 - \exp(-x))$. □

Proof of Theorem 1.1. Holds by 2.1 and 2.2. □

References

[CKS20] Clément L. Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. *CoRR*, abs/2004.00010, 2020.