`impl InverseCDF for ExponentialEV`

Michael Shoemate

April 30, 2025

This proof resides in **"contrib"** because it has not completed the vetting process.

Proves soundness of the implementation of `InverseCDF` for `ExponentialRV`.
The implementation computes the inverse CDF of an exponential random variable.

1 Hoare Triple

Preconditions

Compiler-verified

- Argument `self` of type `ExponentialRV`.
- Argument `r_unif` of type `RBig`.
- Argument `refinements` of type `usize`.
- Generic `R` implements `ODPRound` which denotes the rounding mode, either up or down.

Caller-verified

Argument `r_unif` is in $[0, 1]$.

Pseudocode

```
class InverseCDF(CanonicalRV):
    # type Edge = FBig

    def inverse_cdf(self, r_unif: RBig, refinements: usize, R) -> FBig | None:
        precision = refinements + 1
        r_unif_comp = RBig.ONE - r_unif  #
        f_unif_comp = FBig.from_(r_unif_comp, R.C).with_precision(precision).value() #

        # infinity is not in the range
        if f_unif_comp == FBig.ZERO: #
            return

        f_exp = (-f_unif_comp.ln()).with_rounding(R) #

        f_exp *= self.scale.with_rounding()
        f_exp += self.shift.with_rounding()

        return f_exp.with_rounding()
```

Postcondition

Theorem 1.1. Given a random variable `self` (of type `ExponentialRV`), the algorithm returns `Some(out)` where `out` is the inverse cumulative distribution function of `ExponentialRV` (which includes a rescale and shift) evaluated at `r_unif` with error in direction `R`, or `None`.

The error between `out` and the exactly-computed CDF decreases monotonically as `refinements` increases.

Proof. By the definition of `ExponentialRV`, the density function of the exponential random variable is given by

$$f(x) = \begin{cases} \frac{1}{\lambda} e^{-\frac{x-\mu}{\lambda}} & \text{if } x > \mu \\ 0 & \text{otherwise} \end{cases}$$

where μ denotes `self.shift`, and λ denotes `self.scale`.

The cumulative distribution function (CDF) is given by

$$F(x) = \begin{cases} 0 & \text{if } x \le \mu \\ 1 - e^{-\frac{x-\mu}{\lambda}} & \text{if } x > \mu \end{cases}$$

The inverse CDF is given by

$$F^{-1}(p) = \begin{cases} \mu & \text{if } p = 0 \\ -\lambda \ln(1-p) + \mu & \text{if } 0 < p < 1 \end{cases}$$

Let p denote `r_unif`.

We now compute the inverse CDF of the exponential random variable under the requirements that all computations result in a final output in the direction of `R`. The code directly matches the inverse CDF, but we need to be careful about the rounding directions.

Starting from the end of the algorithm, and working backwards, we see that, after the negation on line **??**, the rounding direction of the output is `R`, consistent with the postcondition. Then, all calculations until the complement on line 6 are conducted with complementary rounding, consistent with the postcondition. The complement on line 6 is then computed with exact rational arithmetic.

Finally, in the case that `r_unif` is exactly one, or close enough for conservative arithmetic to underflow, line 10 returns `None`, which is consistent with the postcondition.

The implementation computes the inverse CDF and in the output, rounding always tends towards `R`, and error decreases monotonically as `refinements` increases, therefore the postcondition is satisfied. □