fn make_float_to_bigint

Michael Shoemate

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

Proves soundness of the implementation of `make_float_to_bigint` in `mod.rs` at commit f5bb719 (outdated[1]).

1 Hoare Triple

Precondition

Compiler-Verified

- Generic `T` implements trait `SaturatingCast<IBig>`

User-Verified

None

Pseudocode

```
def make_float_to_bigint(
    input_space: tuple[VectorDomain[AtomDomain[T]], LpDistance[P, QI]],
    k: i32
) -> Transformation[
    VectorDomain[AtomDomain[T]],
    VectorDomain[AtomDomain[IBig]],
    LpDistance[P, QI],
    LpDistance[P, RBig],
]:
    input_domain, input_metric = input_space
    if input_domain.element_domain.nullable():
        raise "input_domain may not contain NaN elements"

    size = input_domain.size
    rounding_distance = get_rounding_distance(k, size, T)

    def elementwise_function(x_i): #
        x_i = RBig.try_from(x_i).unwrap_or(RBig.ZERO)  #
        return find_nearest_multiple_of_2k(x_i, k) #

    def stability_map(d_in):
        try:
            d_in = RBig.try_from(d_in)
        except:
            raise f"d_in ({d_in}) must be finite"
        return x_div_2k(d_in + rounding_distance, k) #
```

[1] See new changes with `git diff f5bb719..fa860379 rust/src/measurements/noise/nature/float/mod.rs`

```
27
28     return Transformation.new(
29         input_domain,
30         VectorDomain( #
31             element_domain=AtomDomain.default(IBig),
32             size=size,
33         ),
34         Function.new(lambda x: [elementwise_function(x_i) for x_i in x]),
35         input_metric,
36         LpDistance.default(),
37         StabilityMap.new_fallible(stability_map),
38     )
```

Postcondition

Theorem 1.1.

Theorem 1.2. For every setting of the input parameters (`T`) to `make_float_to_bigint` such that the given preconditions hold, `make_float_to_bigint` raises an exception (at compile time or run time) or returns a valid transformation. A valid transformation has the following properties:

1. (Appropriate output domain). For every element x in `input_domain`, `function`(x) is in `output_domain` or raises a data-independent runtime exception.

2. (Stability guarantee). For every pair of elements x, x' in `input_domain` and for every pair (`d_in`, `d_out`), where `d_in` has the associated type for `input_metric` and `d_out` has the associated type for `output_metric`, if x, x' are `d_in`-close under `input_metric`, `stability_map`(`d_in`) does not raise an exception, and `stability_map`(`d_in`) $\leq$ `d_out`, then `function`(x), `function`(x') are `d_out`-close under `output_metric`.

Proof. In the definition of the function on line 17, `RBig.try_from` is infallible when the input is non-nan making the function infallible. There are no other sources of error in the function, so the function cannot raise data-dependent errors.

The function also always returns a vector of IBigs, of the same length as the input, meaning the output of the function is always a member of the output domain, as defined on line 30.

The stability argument breaks down into three parts:

- The casting from float to rational on line 18 is 1-stable, because the real values of the numbers remain un-changed, meaning the distance between adjacent inputs always remains the same.

- The rounding on line 19 can cause an increase in the sensitivity equal to 2^k.

$$\begin{aligned}
&\max_{x\sim x'} d_{Lp}(f(x), f(x')) && (1)\\
&= \max_{x\sim x'} |r_k(x) - r_k(x')|_p && (2)\\
&\leq \max_{x\sim x'} |(x + 2^{k-1}) - (x' - 2^{k-1})|_p && (3)\\
&\leq \max_{x\sim x'} |x - x'|_p + 2^k && (4)\\
&= \max_{x\sim x'} d_{Lp}(x, x') + 2^k && (5)\\
&= 1 \cdot \texttt{d_in} + 2^k && (6)
\end{aligned}$$

  This increase in the sensitivity is reflected in on line 26, where the rounding distance is added to the sensitivity.

- The discarding of the denominator on line 19 is 2^k-stable, as the denominator is 2^k. This increase in sensitivity is also reflected on line 26, where the sensitivity is increased by a factor of 2^k.

Therefore, it is shown that the stability of the function is governed by the stability map. □