fn make_fully_adaptive_composition

Michael Shoemate

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

Proves soundness of fn make_fully_adaptive_composition.

1   Hoare Triple

Precondition

Compiler-verified

- Argument input_domain of type DI.

- Argument input_metric of type MI.

- Argument output_measure of type MO.

- Argument d_in of type MI_Distance.

- Generic DI implements Domain.

- Generic MI implements Metric.

- Generic MO implements CompositionMeasure.

- (DI, MI) implements MetricSpace.

Caller-verified

None

Pseudocode

```
1  def make_fully_adaptive_composition(
2      input_domain: DI,
3      input_metric: MI,
4      output_measure: MO,
5      d_in: MI_Distance,
6  ) -> Odometer[DI, MI, MO, Measurement[DI, TO, MI, MO], TO]:
7
8      # check if fully adaptive composition is supported
9      output_measure.theorem(Adaptivity.FullyAdaptive)
10
11     def function(
12         arg: DI_Carrier,
13     ) -> OdometerQueryable[Measurement[DI, TO, MI, MO], TO, MO_Distance]:
14         return new_fully_adaptive_composition_queryable(
15             input_domain, input_metric, output_measure, d_in, arg
```

```
16          )
17
18      return Odometer.new(
19          input_domain, input_metric, output_measure, Function.new_fallible(function)
20      )
```

Postcondition

Theorem 1.1. For every setting of the input parameters (`input_domain`, `input_metric`, `output_measure`, `d_in`, `DI`, `TO`, `MI`, `MO`) to `make_fully_adaptive_composition` such that the given preconditions hold, `make_fully_adaptive_composition` raises an error (at compile time or run time) or returns a valid odometer. A valid odometer has the following properties:

1. (Data-independent runtime errors). For every pair of members x and x' in `input_domain`, $\mathtt{invoke}(x)$ and $\mathtt{invoke}(x')$ either both return the same error or neither return an error.

2. (Valid odometer queryable). For every member x in `input_domain`, where $\mathtt{function}(x)$ does not raise an error, $\mathtt{function}(x)$ returns a valid odometer queryable.

Proof of data-independent errors. Errors are data-independent by the the postcondition of `new_sequential_odometer_query`
$\square$

Proof of valid odometer queryable. Under the assumption that the input data is a member of the input domain, the precondition of `new_fully_adaptive_composition_queryable` is met, so by its postcondition the return is a valid odometer queryable.
$\square$