

# 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.
- 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 ) -> Odometer[DI, MI, MO, Measurement[DI, MI, MO, TO], TO]:  
6  
7     # check if fully adaptive composition is supported  
8     output_measure.composability(Adaptivity.FullyAdaptive)  
9  
10    def function(  
11        arg: DI_Carrier,  
12    ) -> OdometerQueryable[Measurement[DI, MI, MO, TO], TO, MO_Distance]:  
13        return new_fully_adaptive_composition_queryable(  
14            input_domain, input_metric, output_measure, arg  
15        )  
16  
17    return Odometer.new(  
18        function,  
19        input_domain, input_metric, output_measure,  
20        TO, MO_Distance,  
21        Adaptivity.FullyAdaptive,  
22        Adaptivity.FullyAdaptive
```

```

18         input_domain, input_metric, output_measure, Function.new_fallible(function)
19     )

```

## Postcondition

**Theorem 1.1.** For every setting of the input parameters (`input_domain`, `input_metric`, `output_measure`, `DI`, `MI`, `MO`, `TO`) 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`, `invoke(x)` and `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 `function(x)` does not raise an error, `function(x)` returns a valid odometer queryable.

*Proof of data-independent errors.* Errors are data-independent by the the postcondition of `new_sequential_odometer_queryable`. □

*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. □