# fn make\_row\_by\_row

#### Michael Shoemate

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

Proves soundness of make\_row\_by\_row in mod.rs at commit f5bb719 (outdated<sup>1</sup>). This constructor is a special case of make\_row\_by\_row\_fallible. The proof for this constructor appeals to the proof for make\_row\_by\_row\_fallible.

make\_row\_by\_row returns a Transformation that applies a user-specified function to each record in the input dataset.

## **Vetting History**

• Pull Request #562

# 1 Hoare Triple

#### Precondition

- DI (input domain) is a type with trait RowByRowDomain<DO>. This trait provides a way to apply a map function to each record in the input dataset to retrieve a dataset that is a member of the output domain, of type DO. The trait further implies that DatasetDomain is also implemented for DI.
- DO (output domain) is a type with trait DatasetDomain. DatasetDomain is used to define the type of the row domain.
- M (metric) is a type with trait <code>DatasetMetric</code>. <code>DatasetMetric</code> is used to restrict the set of valid metrics to those which measure distances between datasets.
- MetricSpace is implemented for (DI, M). Therefore M is a valid metric on DI.
- MetricSpace is implemented for (DO, M).
- row\_function has no side-effects.
- If the input to row\_function is a member of input\_domain's row domain, then the output is a member of output\_row\_domain.

#### Pseudocode

```
def make_row_by_row_fallible(
   input_domain: DI,
   input_metric: M,
   output_row_domain: DO,
   # a function from input domain's row type to output domain's row type
```

<sup>&</sup>lt;sup>1</sup>See new changes with git diff f5bb719..13842813 rust/src/transformations/manipulation/mod.rs

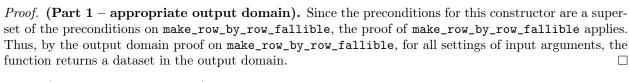
```
row_function: Callable([[DI::RowDomain::Carrier], DO::RowDomain::Carrier])
7 ) -> Transformation:
8 make_row_by_row_fallible(input_domain, input_metric, output_row_domain, row_function)
```

#### Postcondition

For every setting of the input parameters (input\_domain, input\_metric, output\_domain, row\_function, DI, DO, M) to make\_row\_by\_row such that the given preconditions hold, make\_row\_by\_row 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 v in input\_domain, function(v) is in output\_domain or raises a data-independent runtime exception.
- 2. (Stability guarantee). For every pair of elements u, v 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 u, v are d\_in-close under input\_metric and stability\_map(d\_in)  $\leq$  d\_out, then function(u), function(v) are d\_out-close under output\_metric.

### 2 Proofs



*Proof.* (Part 2 – stability map). The proof of make\_row\_by\_row\_fallible similarly applies. Thus, by the stability map proof on make\_row\_by\_row\_fallible, for all settings of input arguments, where u, v are d\_in-close under input\_metric, function(u), function(v) are d\_out-close under output\_metric.