fn match_datetime_component

Michael Shoemate

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

Proves soundness of `match_datetime_component` in `mod.rs` at commit f5bb719 (outdated[1]).

`match_datetime_component` returns the data type and number of possible unique values of a compute operation that retrieves a datetime component.

1 Hoare Triple

Precondition

Compiler-verified

- Argument `temporal_function` of type `TemporalFunction`

Human-verified

None

Pseudocode

```
def match_datetime_component(
    temporal_function: TemporalFunction,
) -> tuple[DataType, u32 | None] | None:
    ({
        Millennium: (DataType.UInt32, None),
        Century: (DataType.Int32, None),
        Year: (DataType.Int32, None),
        IsoYear: (DataType.Int32, None),
        Quarter: (DataType.Int8, 4),
        Month: (DataType.Int8, 12),
        Week: (DataType.Int8, 53),
        WeekDay: (DataType.Int8, 7),
        Day: (DataType.Int8, 31),
        OrdinalDay: (DataType.Int16, 366),
        Hour: (DataType.Int8, 24),
        Minute: (DataType.Int8, 60),
        Second: (DataType.Int8, 60),
        Millisecond: (DataType.Int32, 1_000),
        Microsecond: (DataType.Int32, 1_000_000),
        Nanosecond: (DataType.Int32, 1_000_000_000),
    }).get(temporal_function)
```

[1] See new changes with `git diff f5bb719..7e990382 rust/src/transformations/make_stable_expr/namespace_dt/expr_datetime_component/mod.rs`

Postcondition

Theorem 1.1. For every setting of the input parameters `(temporal_function)` to `match_datetime_component` such that the given preconditions hold, `match_datetime_component` 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`.

2 Proof

Proof. For each compute operation, the data type is automatically tested and verified to be correct. Only output cardinalities that are specified must be proven. Of note:

- there may be up to 53 weeks in a year, due to partial weeks overlapping the end of the year.
- There are up to 366 ordinal days due to leap years.
- Polars does not account for leap seconds, so there are at most 60 seconds in a minute.
- Milliseconds, microseconds and nanoseconds are computed modulo seconds.

□