

CompositionMeasure for ZeroConcentratedDivergence

Michael Shoemate

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

Proves soundness of the implementation of **CompositionMeasure** for **ZeroConcentratedDivergence** in **mod.rs** at **commit f5bb719** (outdated¹).

1 Hoare Triple

Precondition

Compiler-Verified

Types matching pseudocode.

Caller-Verified

None

Pseudocode

```
1 class CompositionMeasure(ZeroConcentratedDivergence):
2     def composability( #
3         self, adaptivity: Adaptivity
4     ) -> Composability:
5         match adaptivity:
6             case Adaptivity.FullyAdaptive:
7                 return Composability.Sequential
8             case _:
9                 return Composability.Concurrent
10
11     def compose(self, d_mids: Vec[Self_Distance]) -> Self_Distance:
12         d_out = 0.0
13         for d_mid in d_mids:
14             d_out = d_out.inf_add(d_mid)
15         return d_out
```

Postcondition

Theorem 1.1. `composability` returns `Ok(out)` if the composition of a vector of privacy parameters `d_mids` is bounded above by `self.compose(d_mids)` under `adaptivity` `adaptivity` and `out-composability`. Otherwise returns an error.

Proof. By the postcondition of **InfAdd** we have that $\sum_i d_mids_i \leq \text{compose}(d_mids)$.

¹See new changes with `git diff f5bb719..6ad9e482 rust/src/combinators/sequential_composition/mod.rs`

Adaptivity	Sequential	Concurrent
Non-Adaptive	Lemma 1.7[BS16]	Corollary 1[Lyu22]
Adaptive	Lemma 1.7[BS16]	Corollary 1[Lyu22]
Fully-Adaptive	Remark 4.4[FZ22]	None

This table is reflected in the implementation of `composability` on line 2.

□

References

- [BS16] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds, 2016.
- [FZ22] Vitaly Feldman and Tijana Zrnic. Individual privacy accounting via a renyi filter, 2022.
- [Lyu22] Xin Lyu. Composition theorems for interactive differential privacy, 2022.