1 Intro

I need to prove that haskell types and terms that I expose wouldn't break the system. It means two things:

1. Terms have the same interface: any combination of `APPLY` that can be used to original(ignoring types) term must be usable with generated; and primitives(numbers, strings, ... and their ops) are the same.
2. Types preserve the same set of invariants

2 Preserving term interface

This is dealing purely with MAlonzo. I need to show that I pass(and skip) the right arguments when writing wrappers. As for primitives - they are already the same.

3 Preserving type invariants

For example: head on empty vector. From terms perspective it's legal, from types - it's illegal.

This is dealing with logic behind Haskell type system and Agda type system. For every exported Agda thingy I need to construct a logical statement from Agda perspective and from Haskell perspective and prove their equivalence.