WORK LOG OF JUNE 24

DANIEL R. BARRERO R.

1

According to Pierce, the operational view of the type $\forall X.T$ is that is values are functions which map a type S to a value of type $[X \mapsto S]T$. Namely,

$$v : \forall X.T \quad \vdash \quad v\ S : [X \mapsto S]T.$$

Coherently, the coercion

```
plusZ' :: x n -> Dict (n ~ (n + Z))
plusZ' _  = unsafeCoerce (Dict :: Dict (n ~ n))
```

can replace the one used by Milewski, which is

```
plusZ :: forall n. Dict (n ~ (n + Z))
plusZ = unsafeCoerce (Dict :: Dict (n ~ n))
```

On the other hand, to coerce associativity of addition he uses

```
plusAssoc :: p a -> q b -> r c -> Dict (((a + b) + c) ~ (a + (b + c)))
plusAssoc _ _ _ = unsafeCoerce (Dict :: Dict (a ~ a))
```

So naturally the question arises whether it can be replaced by

```
plusAssoc' :: forall a b c. Dict (((a + b) + c) ~ (a + (b + c)))
plusAssoc' = unsafeCoerce (Dict :: Dict (a ~ a))
```

Date: June 24, 2025.