

WORK LOG OF JUNE 18

DANIEL R. BARRERO R.

1

The comonad definition as a typeclass is

```
class Functor w => Comonad w where
  extract :: w a -> a
  duplicate :: w a -> w (w a)
```

And in Milewski’s tic-tac-toe, *the evaluator* will be a comonad instance, apparently. This evaluator defines a tree-to-vector behavior:

```
newtype W f a = W {runW :: forall n. f n -> Vec n a}
```

The “obvious” way in which this can be made into a functor doesn’t occur to me. I must review *record syntax* (lyahfkg) and universally quantified types (Pierce).