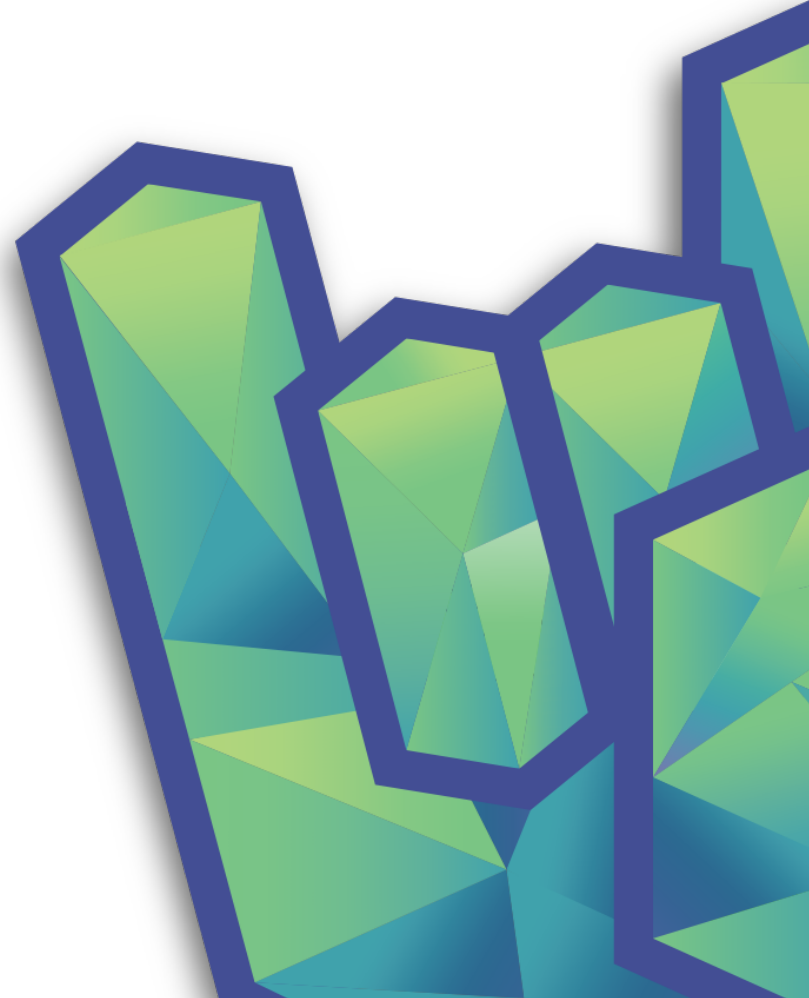


FlatMap



Weaker Monads

= Apply + flatMap method

```
def flatMap[A, B](ma: M[A])(f: A => M[B]): M[B]
```

Monad doesn't have its own new methods

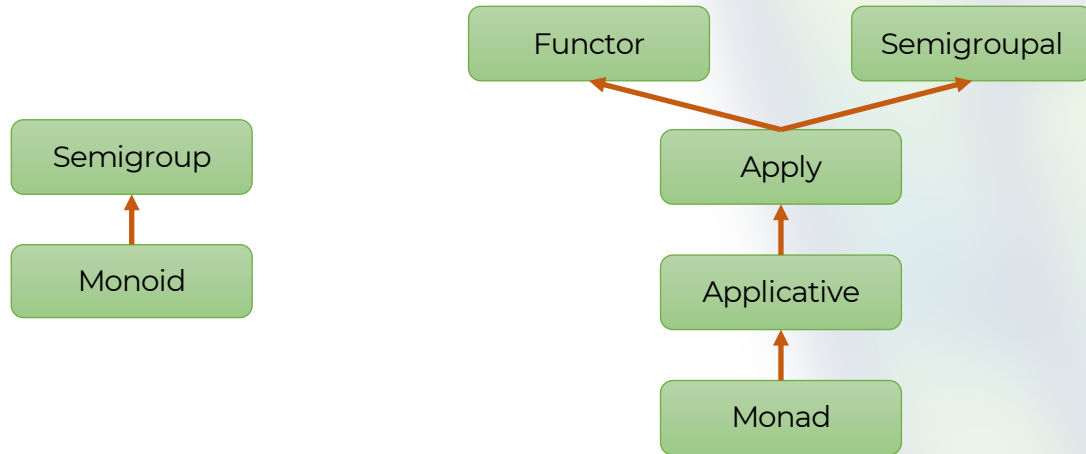
```
trait Monad[F[_]] extends FlatMap[F] with Applicative[F]
```

FlatMap extends Apply, which extends Functor

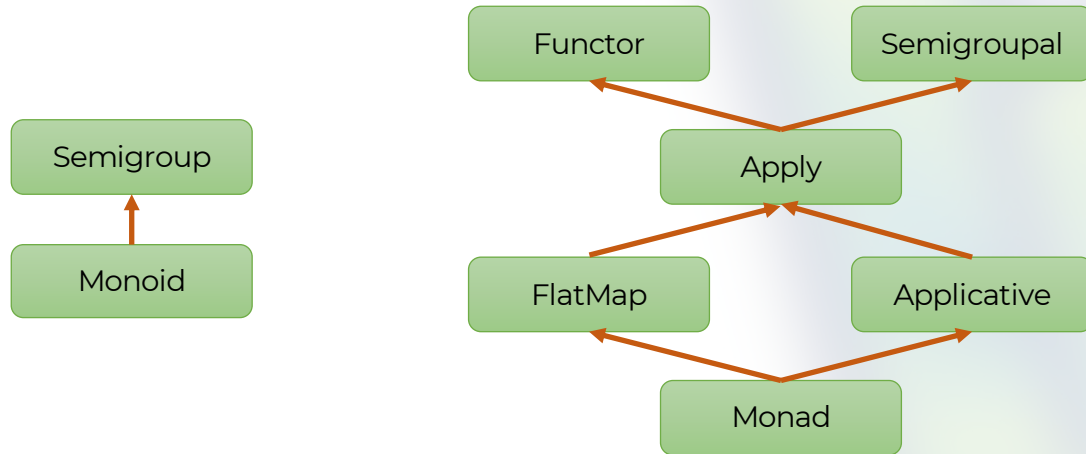
```
import cats.syntax.flatMap._ // flatMap extension method
import cats.syntax.functor._ // map extension method

def getPairs[M[_] : FlatMap, A, B](numbers: M[A], chars: M[B]): M[(A, B)] = for {
  n <- numbers
  c <- chars
} yield (n, c)
```

Cats TC Hierarchy



Cats TC Hierarchy



Cats rock

