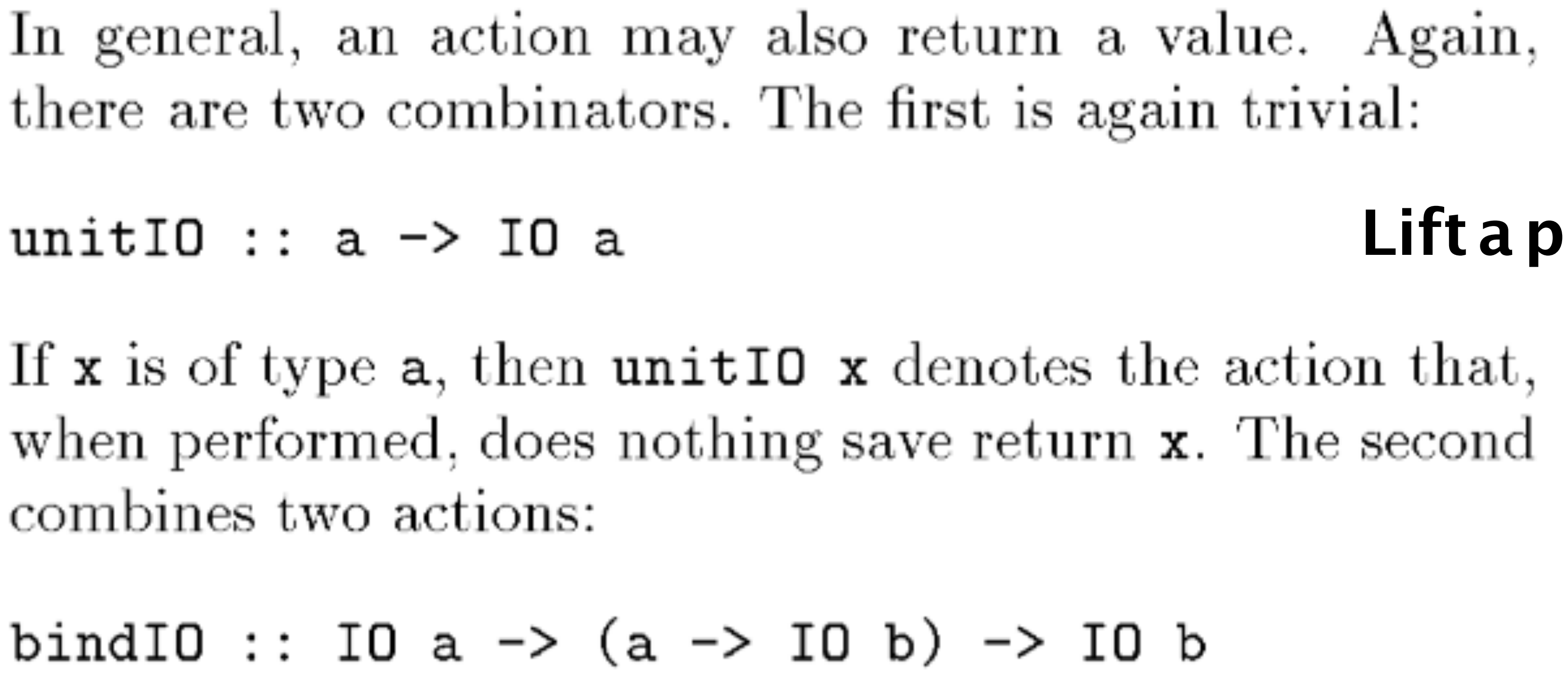
IO monads



In general, an action may also return a value. Again, there are two combinators. The first is again trivial:

```
unitIO :: a -> IO a
```

If **x** is of type **a**, then **unitIO x** denotes the action that, when performed, does nothing save return **x**. The second combines two actions:

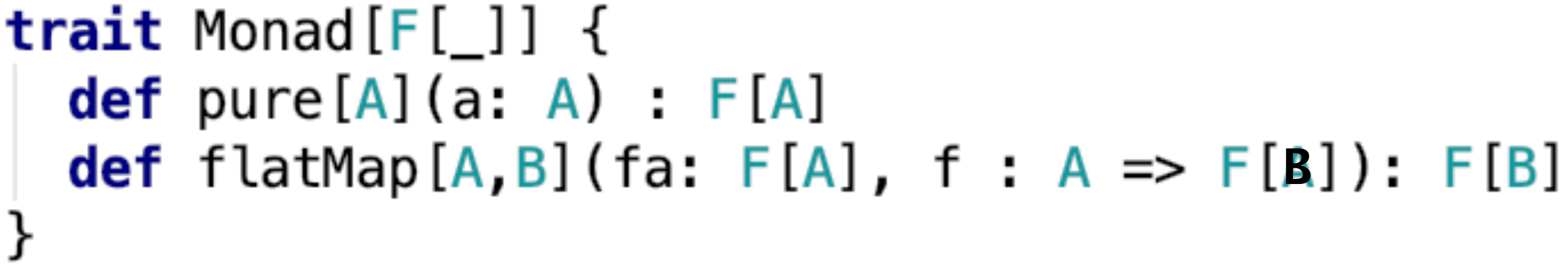
```
bindIO :: IO a -> (a -> IO b) -> IO b
```

Lift a pure A to IO[A]

flatMap or bind

sequences two effects together

Monads in Scala



```
trait Monad[F[_]] {
  def pure[A](a: A) : F[A]
  def flatMap[A,B](fa: F[A], f : A => F[B]): F[B]
}
```

Implemented as a type class

Achieves the same as in Haskell

pure = put an A in a Monad

flatMap = sequence two monads