GHC.Generics

Empty Types	`data V1 p`
Unary Types	`data U1 p = U1`
Sum Types	`data (:+:) f g p = L1 (f p)` `  | R1 (g p)`
Product Types	`data (:*:) f g p = (f p) :*: (g p)`
Constant Types	`newtype K1 i c p = K1 { unK1 :: c }`

Structure becomes evident by inspecting the type

GHC.Generics

From GHC.Generics documentation

```
data Tree a = Leaf a | Node (Tree a) (Tree a)
  deriving Generic
```