GHC.Generics

Normal Datatypes

Generic Representation

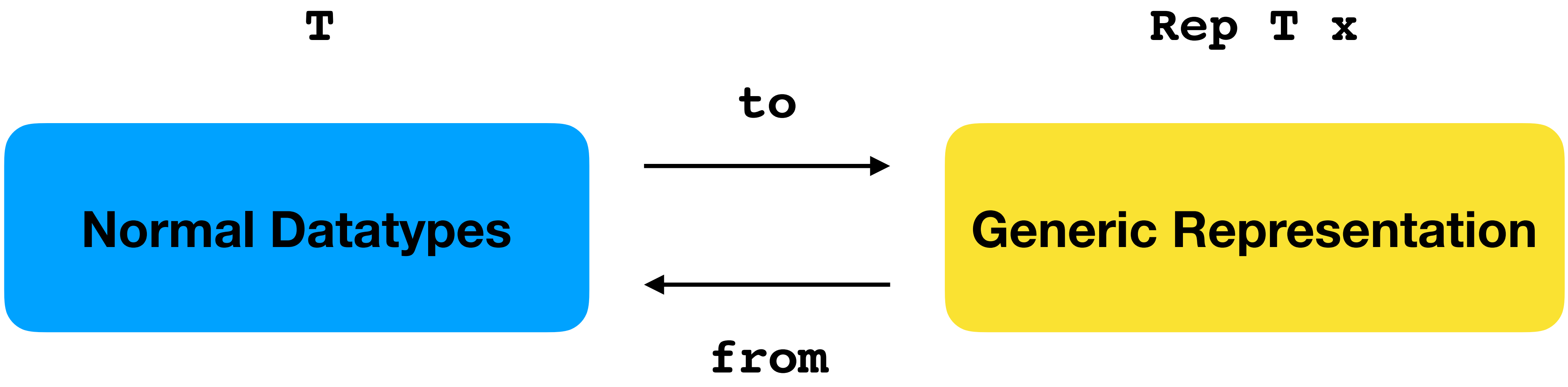
T **Rep T x**

to

from

```
class Generic a where
   type family Rep a :: * → *
   from :: a → Rep a x
   to :: Rep a x → a
```

GHC.Generics



T

Rep T x

to

Normal Datatypes

Generic Representation

from

```
class Generic a where
  type family Rep a :: * → *
  from :: a → Rep a x
  to :: Rep a x → a
```

GHC.Generics

Empty Types	`data V1 p`
Unary Types	`data U1 p = U1`

Product Types	`data (:+:) f g p = L1 (f p)`
	`                  | R1 (g p)`

Sum 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