Structure-directed

```
data S = S { a :: Int
           , b :: String
           }
```

```
data T = T { a :: Int
           , b :: String
           , c :: Bool
           }
```

```
> view (super @S) (T 5 “abc” True)
S 5 “abc”
```

```
> :t (super @S)
Lens T S
```

```
> set (super @S) (S 6 “Matt”) (T 5 “abc” True)
T 6 “Matt” True
```

18

Structure-directed

```
data S = S { a :: Int
           , b :: String
           }
```

```
data T = T { a :: Int
           , b :: String
           , c :: Bool
           }
```

```
> :t (super @S)
Lens T S
> view (super @S) (T 5 “abc” True)
S 5 “abc”
> set (super @S) (S 6 “Matt”) (T 5 “abc” True)
T 6 “Matt” True
```

```
data FormResponse = FormResponse { name :: String
                                 , age :: Int }


data Person = Person { name :: String
                     , age :: Int
                     , uniqueID :: UID
                     , dateModified :: Date
                     , userSettings :: Settings }
```

```
updatePerson :: Person -> IO Person
updatePerson p = do
  formResponse <-
    getFormResponse (view (typed @Settings
                             <o> super @FormSettings))
                          p)
  date <- getCurrentDate
  return $ p & set (super @FormResponse)
                   formResponse
             & set (field @“dateModified”)
                   date
```