



TRANSFORMATIONS

BENIGN GENERAL

```
class Monoid m  $\Rightarrow$  Action m a where
```

```
  act :: m  $\rightarrow$  a  $\rightarrow$  a
```

```
instance Action Transformation Prim where
```

```
  act = transformP
```

Note: This requires additional language extensions because type classes usually accept a single argument

BEING MORE GENERAL

```
class Monoid m  $\Rightarrow$  Action m a where
```

```
  act :: m  $\rightarrow$  a  $\rightarrow$  a
```

```
instance Action Transformation Prim where
```

```
  act = transformP
```

Note: This requires additional language extensions because type classes usually accept a single argument

BEING MORE GENERAL