MONOID ACTION LAW 2

```
act (m1 <> m2) = act m1 . act m2
```

PROOF

```
transformD (t1 <> t2)

map (act (t1 <> t2))

map (act t1 . act t2)

map (act t1) . map (act t2)

transformD t1 . transformD t2
```

DIAGRAM REVISTED

```
newtype Diagram =

  Diagram (Dual [Prim], Envelope, Trace)

  deriving (Semigroup, Monoid)
```

HELPER FUNCTIONS

```
unD (Diagram (Dual ps, _, _)) = ps

prim p = Diagram (Dual [p], envelopeP p, traceP p)

mkD = hom prim

envelope (Diagram (_, e, _)) = e

trace (Diagram (_, _, t)) = t
```