$$map(f \circ g) == map(f) \circ map(g)$$

$$f :: B \Rightarrow C$$
$$g :: A \Rightarrow B \qquad f \circ g :: A \Rightarrow C$$

```scala
trait CovariantFunctor[F[_]] {
  def map[A, B](f: A => B): F[A] => F[B]
}

trait Bar[A] {
  def bar(x: X): A
}

object CoBar extends CovariantFunctor[Bar] {
  def map[A, B](f: A => B): Bar[A] => Bar[B] = (barA: Bar[A]) => new Bar[B] {
    def bar(x: X): B = f(barA.bar(x))
  }
}
```

$$barA :: Bar[A]$$
$$x :: X$$

$$map(f \circ g) :: Bar[A] \Rightarrow Bar[C]$$
$$map(f \circ g)(barA).bar(x) =$$
$$(f \circ g)(barA.bar(x))$$

$$map(f) \circ map(g) :: Bar[A] \Rightarrow Bar[C]$$
$$(map(f) \circ map(g))(barA).bar(x) =$$
$$map(f)(map(g)(barA)).bar(x) =$$
$$f(map(g)(barA).bar(x)) =$$
$$f(g(barA.bar(x))) =$$
$$(f \circ g)(barA.bar(x))$$

Thus Bar preserves morphism composition.