

# Golden round-trip testing

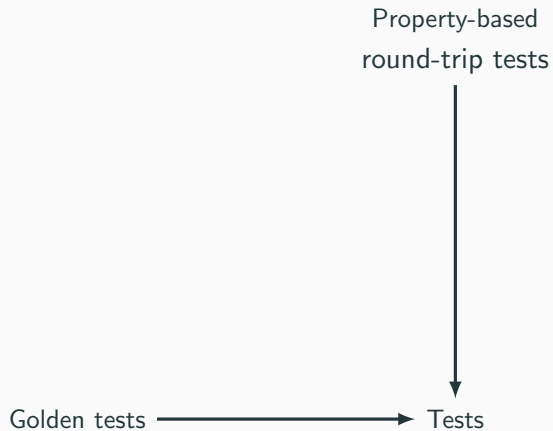
---

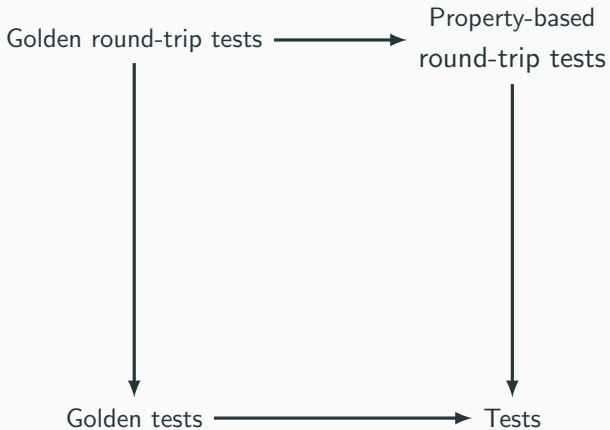
Mark Hopkins

September 2021, **fp-syd**

@antiseifdual

antiseifdual.com





## Golden tests

---

- create files if missing
- optionally update files automatically

- tasty-golden

## Property-based tests

---

# Property-based testing

- Gen a
- properties
- shrinking



Gen a

# Properties

$$\forall x.p(x)$$

$$\forall xy.p(x, y)$$

$$\forall xyz.p(x, y, z)$$

# Properties

$\forall s : \text{String} . \text{reverse}(\text{reverse } s) = s$

$\forall n : \text{NavigationsFromHomepage} . \text{noDeadLinks}(n)$

$\forall s : \text{SequenceOfDBOps} . \text{behaviourMatchesModel}(s)$

$\forall p : \text{ParallelOps} . \text{allLinearisationsGiveTheSameResult}(p)$

# Property-based testing libraries

- QuickCheck
- hedgehog

# Round-tripping

---

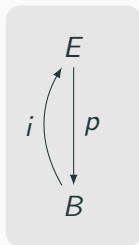
# Bijections

A function has an inverse  $\Leftrightarrow$  it is surjective and injective

$f \circ g = \text{id}, g \circ f = \text{id} \iff f, g$  surjective and injective

# Sections and retractions

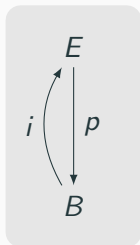
section



retraction

## Sections and retractions

section



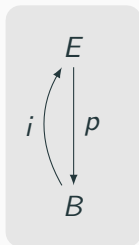
retraction

$$pi = \text{id}_B$$



## Sections and retractions

section

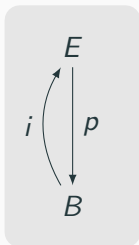
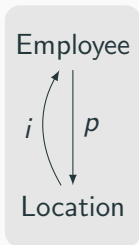


retraction

$$pi = \text{id}_B$$

$$(ip)^2 = ipip = i(pi)p = ip$$

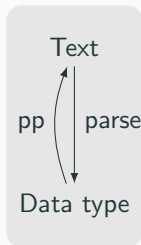
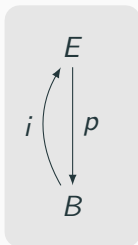
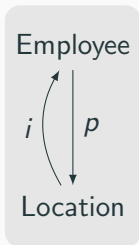
## Sections and retractions



$$pi = \text{id}_B$$

$$(ip)^2 = ipip = i(pi)p = ip$$

## Sections and retractions



$$pi = id_B$$

$$(ip)^2 = ipip = i(pi)p = ip$$

## Parsing is partial!

```
pp :: Data -> Text
parse :: Text -> m Data

parse . pp == pure

g >=> g == g
where g = fmap pp . parse
```

## Property-based round-trip tests

---

## Round-trip support in hedgehog

```
tripping
:: ( Eq (f a), Show (f a), Show b
  , Applicative f
  , MonadTest m
  )
=> a
-> (a -> b)
-> (b -> f a)
-> m ()
```

## Golden round-trip tests

---

Unit tests?



## Combine round tripping with golden tests!

Extension to tasty-golden

## Complications, caveats

- golden and round-tripping start from code not a file
- we have a `Gen` a not a `Gen Text`
- if we add optional data, we'll need to remember to rerun the generator

Demo