Fuzzy Testing
A golden bullet for writing tests?

Julian Bieber

May 16, 2018

Definition

- General input satisfying precondition
- Property that must hold true on output

Should every component be tested this way?

No, most small unit tests do not lend themselves well to Property Based Tests.
In many cases the of writing Property Based Tests is hard to justify.
But they excel in a few cases.

Bijective Functions

- Serialization - Serialization must be reversible
- DB read/write - Every object written to the db must be retrievable
- Generally applicable $f'(f(x)) = x$

Example: `bijective.SerializationSpec`

Invariants

- Algorithms - sorting, compression
- Repricing strategies
- Integration tests
- Generally applicable if there are simple Pre- and Postconditions

Example: `invariant.CompressionSpec`

Regression

- There is an old component; no tests : (
- Works mostly correct; one known bug
- Goal: make sure new implementation does not break existing behavior
- $\forall x : notKnownBug(x) \implies old(x) = new(x)$
- $\forall x : previousBug(x) \implies expectation(new(x)) = true$

Example: `regression.FibonacciSpec`

Notes On ScalaCheck - Preconditions

- Bound to Generator with `Gen[A].suchThat(predicate)` or `Gen[A].retryUntil(predicate)`
- Generator built to include precondition `BasicGenerators.genNonEmptySeq[A](Gen[A]): Gen[Seq[A]]`
- Inside `forAll` block with `whenever(predicate(generated)) {...}`

Notes on ScalaCheck - Configuration

- set through `implicit val PropertyCheckConfiguration`
- `minSuccessful`: number of successful evaluations for pass
- `maxDiscardedFactor`: `minSuccessful * maxDiscardFactor` = number of generated values that can be discarded
- `minSize`: minimun size for iterable types (String, Seq, Set ...)
- `sizeRange`: `minSize * sizeRange` = max size for iterable types
- `workers`: number of threads evaluating the `forAll` block

Notes on ScalaCheck - Shrink

- Goal: Find smallest input to produce failure
- Bug: When a value is shrunk preconditions are ignored
- Bug: Input shrunk without failure
- Symptom: The test is executed for unexpected values $\Longrightarrow$ test reporting failure where nothing is wrong
- Workaround: Disable shrinking when there are preconditions
- Workaround: Preconditions through specific generator + `whenever(p(x)) {...}` block

```
implicit val noShrink: Shrink[A] = Shrink.shrinkAny
```