ScalaCheck

Scalathon 2011 Rickard Nilsson

What is ScalaCheck?

- An automated, property-based testing tool
- A port of Haskell QuickCheck

How do I use it?

```
object MyProperties extends Properties("MyProperties") {

  property("String.startsWith") =
    forAll { (a: String, b: String) =>
      (a+b).startsWith(a)
    }

}

scala> MyProperties.check
+ MyProperties.String.startsWith: OK, passed 100 tests
```

Basic concepts

- Properties

  `org.scalacheck.Prop`

- Generators

  `org.scalacheck.Gen`

- Test runner = property evaluator

  `org.scalacheck.Test`

Generators

- No dependencies to other parts of ScalaCheck, can be used on its own
- Basically a function:

```
Gen.Params => Option[T]
```

Generator combinators

- The trait `Gen` is a *monad*
- The module `Gen` contains building blocks for creating new generators

Generator combinators, cont.

```
import org.scalacheck.{Gen, Arbitrary}
import org.scalacheck.Gen.{oneOf, choose}
import org.scalacheck.Arbitrary.{arbitrary}

val genVowel: Gen[Char] = oneOf('a','e','i','o','u','y')

val genRange: Gen[(Int,Int)] = for {
  start <- arbitrary[Int]
  end   <- choose(start, Int.MaxValue)
} yield (start, end)
```

```
scala> genVowel.sample
res0: Option[Char] = Some(o)

scala> genRange.sample
res1: Option[(Int, Int)] = Some((-1640017041,1989177566))
```

Properties

- Also a function:
  `Prop.Params => Prop.Result`
- Module `Prop` contains methods for creating properties
- `Prop.forAll` is the most common one

```
val p = Prop.forAll(arbitrary[Int], arbitrary[Int]) {
  (m: Int, n: Int) => m+n == n+m
}

val q = Prop.forAll { (m: Int, n: Int) => m+n == n+m }
```

Property evaluation

- `Test.check`

```
scala> check(Test.Params(minSuccessfulTests = 500), myProp)
res0: org.scalacheck.Test.Result =
  Result(Passed,500,0,Map(),60)

scala> myProp.check
+ OK, passed 100 tests.
```

Source code

- Available at GitHub

  https://github.com/rickynils/scalacheck

- No overwhelming amount of code

```
Gen.scala
Prop.scala
Test.scala
Arbitrary.scala
Shrink.scala
Commands.scala
```

Building and testing

- SBT is bundled with the project

```
$ git clone https://github.com/rickynils/scalacheck
$ cd scalacheck
$ ./sbt update
$ ./sbt compile
$ ./sbt test
```

- Properties in `src/test/scala`

- SBT's test action bootstraps the source in place