



14 600 000 test cases

Property Based Testing

Vojta Rylko

Why do we test?

How do we test?

```
sort( [ 3, 1, 2 ] ) == [ 1, 2, 3 ]
```

```
X = sort( list() )  
assert( is_sorted( X ) )
```

Property Based Testing



Examples

1. Compression library

- a. For any **A**: `decompress(compress(A)) == A`
- b. For any **A**: `size(compress(A)) + const ≤ size(A)`

2. Refactoring

- a. For any **Args**: `old_function(Args) == refactored_function(Args)`

3. Lists-related

- a. For any **List**: `sort(sort(List)) == sort(List)`

```
use Test::More tests => 1;
use Test::LectroTest::Compat;

my $prop = Property {
    ##[ x <- Apply {abs(shift)} Float ]##
    sqrt( $x * $x ) == $x;
}, name => "sqrt is inverse of square";

holds( $prop );

$ perl test.pm
1..1
ok 1 - property 'sqrt is inverse of square'
(1000 attempts)
```

```
import com.pholser.junit.quickcheck.Property;
import com.pholser.junit.quickcheck.runner.JUnitQuickcheck;
import org.junit.runner.RunWith;
import static org.junit.Assert.*;

@RunWith(JUnitQuickcheck.class)
public class StringProperties {
    @Property public void concatenationLength(String s1, String s2) {
        assertEquals(s1.length() + s2.length(), (s1 + s2).length());
    }
}
```

```
$ java test.java
```

```
Error: Could not find or load main class test.java
```


Property Based Testing

- Generators
- Distribution - input and output (labelling)
- Shrinking
- Properties

Case Study: XAE

- **Generate** random MAQL (AST) with random Computed Attributes
Run interpreter → optimizer → SQL generator
Check
 - all stages passed OR user fault thrown (“no red bar”)
 - valid internal representation (query tree)
 - properties of optimized QT
- Nightly Jenkins job (40 k of of tries ~ one hour)
- Consider usefull - real bugs are found



Build #533

Summary

Credits

- George Fink and Matt Bishop. 1997. Property-based testing: a new approach to testing for assurance. *SIGSOFT Softw. Eng. Notes* 22, 4 (July 1997), 74-80
- Koen Claessen and John Hughes. 2000. QuickCheck: a lightweight tool for random testing of Haskell programs. In *Proceedings of the fifth ACM SIGPLAN international conference on Functional programming* (ICFP '00). ACM, New York, NY, USA, 268-279.
- <http://fsharpforfunandprofit.com/posts/property-based-testing-2> (CC BY 3.0)
- [Tips for better testing with LectorTest](#)
- [Theft README](#)