

# Formal Verification in Scala

Ramon Boss, Anna Doukmak

2019-06-13

# Table of Contents

Formal Verification

Problems

Results

# Table of Contents

Formal Verification

Problems

Results

- Pure Scala
- Pre- and Postcondition
- Outcome: valid, invalid, unknown

## Example: max

```
1 def max(x: Int, y: Int): Int = {  
2   require(0 <= x && 0 <= y)  
3   val d = x - y  
4   if (d > 0) x  
5   else y  
6 } ensuring (res =>  
7   x <= res && y <= res && (res == x || res == y))
```

# Properties

- Bitcoin-S
- No Inflation Property
- Addition with Zero Property

# Table of Contents

Formal Verification

Problems

Results

# Rewrite Code: Turning Object into Case Object

This:

```
1 object Satoshi extends BaseNumbers[Satoshi] {  
2   val zero = Satoshi(Int64.zero)  
3   val one = Satoshi(Int64.one)  
4 }
```

Becomes this:

```
1 case object Satoshi extends BaseNumbers[Satoshi] {  
2   val zero = Satoshi(Int64.zero)  
3   val one = Satoshi(Int64.one)  
4 }
```



# Rewrite Code: BigInt &-Function to Bound Check

This:

```
1 sealed abstract class Number {  
2   def andMask: BigInt  
3   def checkResult(result: BigInt): BigInt = {  
4     require((result & andMask) == result)  
5     result  
6   }  
7 }
```

Becomes this:

```
1 sealed abstract class Number {  
2   def checkResult(result: BigInt): BigInt = {  
3     require(  
4       result <= BigInt("9223372036854775807") &&  
5       result >= BigInt("-9223372036854775808")  
6     )  
7     result  
8   }  
9 }
```

# Rewrite Code: Getting Rid of Abstract Type Member

This:

```
1 sealed abstract class CurrencyUnit {  
2   type A  
3  
4   protected def underlying: A  
5 }  
6  
7 sealed abstract class Satoshi extends CurrencyUnit
```

Becomes this:

```
1 ???
```

# Table of Contents

Formal Verification

Problems

Results

# Output of Stainless

```
[Warning] The Z3 native interface is not available. Falling back onto smt-z3.
[Info] - Checking cache: 'cast correctness' VC for underlying @59:30...
[Info] - Checking cache: 'cast correctness' VC for inv @59:30...
[Info] - Checking cache: 'cast correctness' VC for inv @59:30...
[Info] Cache hit: 'cast correctness' VC for inv @59:30...
[Info] Cache hit: 'cast correctness' VC for inv @59:30...
[Info] Cache hit: 'cast correctness' VC for underlying @59:30...
[Info] - Checking cache: 'cast correctness' VC for underlying @43:33...
[Info] Cache hit: 'cast correctness' VC for underlying @43:33...
[Info] - Checking cache: 'precond. (call checkResult(thiss, underlying(thiss) + u ...)' VC for + @22:11...
[Info] - Checking cache: 'precond. (call +(@unchecked ( ...))' VC for + @4:3...
[Info] Cache hit: 'precond. (call checkResult(thiss, underlying(thiss) + u ...)' VC for + @22:11...
[Info] Cache hit: 'precond. (call +(@unchecked ( ...))' VC for + @4:3...
[Info]
[Info] stainless summary
[Info]
[Info] +      precondition. (call +(@unchecked ( ...))      valid from cache      BasicArithmetic.scala:4:3      0.022
[Info] +      precondition. (call checkResult(thiss, underlying(thiss) + u ...) valid from cache      NumberType.scala:22:11      0.021
[Info] inv      cast correctness      valid from cache      NumberType.scala:59:30      0.249
[Info] inv      cast correctness      valid from cache      NumberType.scala:59:30      0.247
[Info] underlying cast correctness    valid from cache      CurrencyUnits.scala:43:33    0.010
[Info] underlying cast correctness    valid from cache      NumberType.scala:59:30      0.849
[Info]
[Info] -----
[Info] total: 6      valid: 6      (6 from cache) invalid: 0      unknown: 0      time: 1.398
[Info]
[Info] Shutting down executor service.
```

# Lessons Learned

- Write Verifiable Code
- We Verified Not Original Bitcoin-S Code