

Formal Verification in Scala

Ramon Boss, Anna Doukmak

June 11, 2019

Table of Contents

Formal verification components //TODO: or concept

Process of verification //TODO: change the title

Results

Table of Contents

Formal verification components //TODO: or concept

Process of verification //TODO: change the title

Results

- Pure Scala code
- Imperative features
- Formal specification with pre- and postcondition
- Outcomes of verification: valid, invalid, unknown

Properties to verify

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

Table of Contents

Formal verification components //TODO: or concept

Process of verification //TODO: change the title

Results

Verifying No Inflation Property

- Validation of a transaction with *checkTransaction*
- Integration of Stainless
- Adjusting the property
- Bug in *checkTransaction*

Verifying No Inflation Property

- Validation of a transaction with *checkTransaction*
- Integration of Stainless
- Adjusting the property
- Bug in *checkTransaction*

```
1  val prevOutputTxIds = transaction.inputs.map(_.  
    previousOutput.txId)  
2  val noDuplicateInputs = prevOutputTxIds.distinct.  
    size == prevOutputTxIds.size
```



```

▼ 6 core-test/src/test/scala/org/bitcoins/core/protocol/transaction/TransactionTest.scala
@@ -294,6 +294,12 @@ class TransactionTest extends FlatSpec with MustMatchers {
294 294     }
295 295     }
296 296
297 + it must "check transaction with two out point referencing the same tx with different indexes" in {
298 +   val hex = "0200000002924942b0b7c12ece0dc8100d74a1cd29acd6cfc60698bfc3f07d83890e2c20b6000000006a47304402202831
299 +   val btx = BaseTransaction.fromHex(hex)
300 +   ScriptInterpreter.checkTransaction(btx) must be(true)
301 + }
302 +
297 303 private def findInput(
298 304   tx: Transaction,
299 305   outPoint: TransactionOutPoint): Option[(TransactionInput, Int)] = {

```

```

▼ 4 core/src/main/scala/org/bitcoins/core/script/interpreter/ScriptInterpreter.scala
@@ -779,8 +779,8 @@ sealed abstract class ScriptInterpreter {
779 779     val totalSpentByOutputs: CurrencyUnit =
780 780       outputValues.fold(CurrencyUnits.zero) (_ + _)
781 781     val allOutputsValidMoneyRange = validMoneyRange(totalSpentByOutputs)
782 -   val prevOutputTxIds = transaction.inputs.map(_.previousOutput.txId)
783 -   val noDuplicateInputs = prevOutputTxIds.distinct.size == prevOutputTxIds.size
782 +   val prevOutputs = transaction.inputs.map(_.previousOutput)
783 +   val noDuplicateInputs = prevOutputs.distinct.size == prevOutputs.size
784 784
785 785     val isValidScriptSigForCoinbaseTx = transaction.isCoinbase match {
786 786       case true =>

```

Verifying Addition with Zero

- Datatype Satoshi
- Method for verification

```
+(c: CurrencyUnit): CurrencyUnit
```

- Rewriting 2 classes into Pure Scala

Rewriting the code

Formal specification

```
1  override def +(c: CurrencyUnit): CurrencyUnit = {  
2      require(c.satoshis == Satoshis.zero)  
3      Satoshis(satoshis.underlying + c.satoshis.  
4          underlying)  
5  } ensuring(res => res.satoshis == this.satoshis)
```

Table of Contents

Formal verification components //TODO: or concept

Process of verification //TODO: change the title

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.
```