Issue with CFG in Swam Engine

Issue

Creation of CFG for some of the methods in Wasm module is failing in swam and leading to exception.

Using the CFG and getting it into Instantiator class

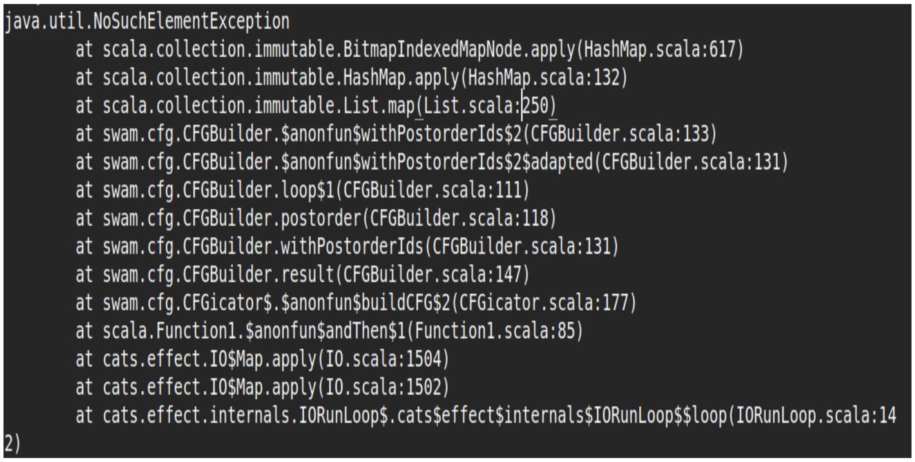
Used the runtime.internal.compiler.CompiledFunction class to store the CFG foreach function at the time of compilation in the form of IO[CFG]. As the idea was to get the CFG in the instantiator class to build the path coverage. While the functions are allocating in the runtime.internal.instance.Instantiator class I am also trying to get the CFG and trying to print the basic blocks information for all the functions except the once injected by Wasi in the Wasm module.But getting the below mentioned exception for few of the programs as mentioned in the Table 1.

Exception in detail and till where I could debug the issue

- Getting an exception java.util.NoSuchElementException while using the swam.cfg.CFG for the getting the basic blocks information for few Wasm modules in

Program Name	Issue exists or not	Exception hit at method	Exception
Largest_number	Yes	__original_main	java.util.NoSuchElementException
URL_Decoding	Yes	decode	java.util.NoSuchElementException
Palindrome	Yes	palindrom	java.util.NoSuchElementException
Hofstader	Yes	doSync	java.util.NoSuchElementException
Fibo	No	N/A	N/A
Deconvolutional	No	N/A	N/A
Thiele	No	N/A	N/A
Middle_Three	No	N/A	N/A
Babbage	No	N/A	N/A

Table 1: Possible area of the problem in the code

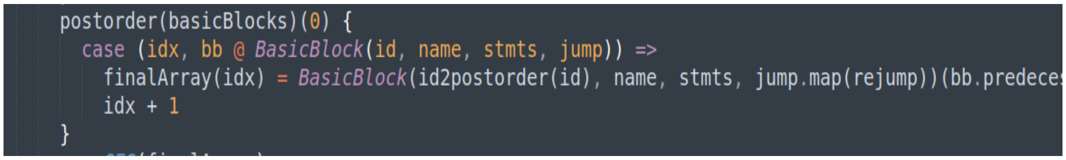


```
java.util.NoSuchElementException
        at scala.collection.immutable.BitmapIndexedMapNode.apply(HashMap.scala:617)
        at scala.collection.immutable.HashMap.apply(HashMap.scala:132)
        at scala.collection.immutable.List.map(List.scala:250)
        at swam.cfg.CFGBuilder.$anonfun$withPostorderIds$2(CFGBuilder.scala:133)
        at swam.cfg.CFGBuilder.$anonfun$withPostorderIds$2$adapted(CFGBuilder.scala:131)
        at swam.cfg.CFGBuilder.loop$1(CFGBuilder.scala:111)
        at swam.cfg.CFGBuilder.postorder(CFGBuilder.scala:118)
        at swam.cfg.CFGBuilder.withPostorderIds(CFGBuilder.scala:131)
        at swam.cfg.CFGBuilder.result(CFGBuilder.scala:147)
        at swam.cfg.CFGicator$.$anonfun$buildCFG$2(CFGicator.scala:177)
        at scala.Function1.$anonfun$andThen$1(Function1.scala:85)
        at cats.effect.IO$Map.apply(IO.scala:1504)
        at cats.effect.IO$Map.apply(IO.scala:1502)
        at cats.effect.internals.IORunLoop$.cats$effect$internals$IORunLoop$$loop(IORunLoop.scala:14
2)
```

Figure 1: Screenshot of the exception

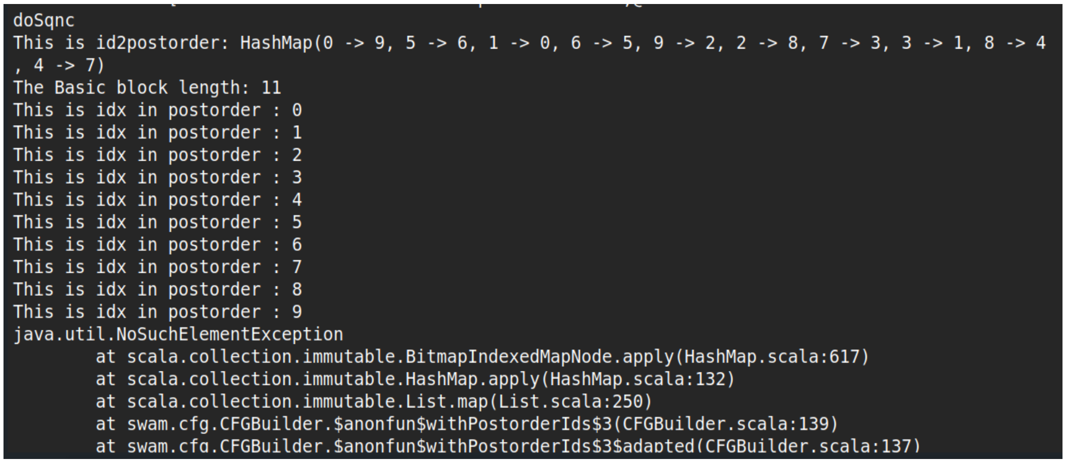
the methods other that methods that are injected by Wasi.

- Exact line at which the exception is occurring is when finalArray which is of type Array[BasicBlocks] [Line 133] is being filled with the BasicBlocks information as shown in Figure 2. The code shown in Figure 2 is part of CFGBuilder class in the method withPostorderIds [Method Link] .
- The Figure 3 shows the details information of the



```
postorder(basicBlocks)(0) {
  case (idx, bb @ BasicBlock(id, name, stmts, jump)) =>
    finalArray(idx) = BasicBlock(id2postorder(id), name, stmts, jump.map(rejump))(bb.predece
    idx + 1
}
```

Figure 2: The line containing the finalArray(idx) is given an exception



```
doSqnc
This is id2postorder: HashMap(0 -> 9, 5 -> 6, 1 -> 0, 6 -> 5, 9 -> 2, 2 -> 8, 7 -> 3, 3 -> 1, 8 -> 4
, 4 -> 7)
The Basic block length: 11
This is idx in postorder : 0
This is idx in postorder : 1
This is idx in postorder : 2
This is idx in postorder : 3
This is idx in postorder : 4
This is idx in postorder : 5
This is idx in postorder : 6
This is idx in postorder : 7
This is idx in postorder : 8
This is idx in postorder : 9
java.util.NoSuchElementException
        at scala.collection.immutable.BitmapIndexedMapNode.apply(HashMap.scala:617)
        at scala.collection.immutable.HashMap.apply(HashMap.scala:132)
        at scala.collection.immutable.List.map(List.scala:250)
        at swam.cfg.CFGBuilder.$anonfun$withPostorderIds$3(CFGBuilder.scala:139)
        at swam.cfg.CFGBuilder.$anonfun$withPostorderIds$3$adapted(CFGBuilder.scala:137)
```

Figure 3: The hash map id2postorder shows map of 10 basic blocks indexed[0-9]. The length of the basic block finalArray is 11. So when trying to access the index 10 in the hashmap it is giving the exception in that line.

place where the issue is found . I was able to locate the place of origin of the issue but I didn't understand how the hashmap for postorderids is being made in the method postorder and the @tailrec loop functions Link in the same class CFGBuilder class.

I was able to print the issue. id2postorder shows map of 10 basic blocks indexed[0-9]. The length of the basic block received in the function withPostorderIds is 11. When accessing the id2postorder with the index 10 which is not present in the hashmap as shown in 3. It is at this point that it throws the exception.

Conclusion

In conclusion of this document, I would like to understand the issue with the creation of basic blocks for certain functions for few of the wasm modules given in the Table 1.

Note

The screenshots taken in the Figure 1,2,3 is using the wasm file 'Hofstadter-Conway_$10 000_sequence.wasm' and the exception is for the function doSqnc which is not a wasi method.