**SE465 – 1c Inter Procedure Documentation**

**Usage:**

java pipair <input file> <support> <confidence> <Inter procedure>

**Description:**

<Inter procedure> - Boolean value. ‘true’ to enable inter procedure analysis and ‘false’ or leaving it blank to disable inter procedure analysis.

Once inter procedure analysis is enabled, pipair will further analyze bugs that were detected, by expanding the inner functions in the bugged function to find missing function.

*Eg:*

|  |  |
| --- | --- |
| Inter-procedure analysis disabled | Inter-procedure analysis enabled |
| *Scope1{*  *A();*  *Scope2();*  *}*  *Scope2{*  *B();*  *}* | *Scope1{*  *A();*  *Scope2();*  *B();*  *}*  *Scope2{*  *B();*  *}* |
| A()B() Cannot be found in Scope1 | 1. A()B() Cannot be found in Scope1 2. Expand Scope2 3. A()B() is found |
| Bug is reported | A()B() is no longer a bug in Scope1 |

**Algorithm:**

*Pseudo Code****:***

*1. gather bugs that were reported at the end of pipair function, recode the bugged function and missing\_inner\_function*

*2. for each bug in bugs:*

*3. for each inner function in the bugged function*

*4. perform a DFS, the root is the inner function*

*5. if the DFS found the missing\_inner\_function, mark the bugged function as not a bug*

*6. print all bugs that is stilled marked as bug*

**Algorithm explanation:**

Likely-invariants testing test the likelihood of paired function calls. In the previous version of pipair, the internal functions in each function call are paired. In the case of a missing paired function call, no further actions are taken. This could result in a false positive because for example:

|  |  |
| --- | --- |
| *Scope1{*  *A();*  *C();*  *Scope2();*  *}* | A(){  } |
| *Scope2{*  *B();*  *}* | C(){  } |

Assume A()B() is expected to be paired. Although in Scope1, A()B() pair does not exist directly, since Scope1 calls Scope2, and Scope2 contains B(), A() is technically paired with B() because in the execution of Scope1, B() always follow A(). Thus, the example above will result in a false positive.

The algorithm for inter procedure call will perform a DFS in every inner function in Scope1. The DFS will search for the missing function, in this case, the missing function is B(). If any of the inner function returns true for the missing function, then the pair of function is made and the false positive is avoided.

In the example, A() will return false for B(), C() will return false for B(), but Scope2() will return true for B(). Thus, A()B() pair in Scope1 is found and bug is marked as not a bug.

**Experiment with test 3:**

With support = 3, confidence = 65, 205 bugs were found in the test 3 when inter\_procedure disabled. Same setting with inter\_procedure enabled found 156 bugs.