

Summary

Sink States: $0(0 \times 10^0)$

Table 1: Pulse Analysis Summary

Classes	Methods	States	Unsatisfiable Clauses	Unreachable States	Possible concurrent Methods	Total. no. of pairs	No. of concurrent pairs	Percentage of concurrent Methods
Webserver	8	1	0	0	7	36	28	78
Server	2	1	0	0	1	3	1	33
Client	2	1	0	0	1	3	1	33
Total Classes=3	12	3	0	0	9	42	30	71

Contents

1	Webserver	3
2	Server	4
3	Client	5
4	Abbreviation	6
5	Annotated Version of Sequential Java Program generated by Sip4j	7

1 Webservice

Table 2: Methods Requires Clause Satisfiability

Method	Satisfiability
Webservice	✓
main	✓
webservice	✓
serveClient	✓
LOG	✓
transfer	✓
transferHeader	✓
transferData	✓

Table 3: State Transition Matrix

	alive
alive	↑

Table 4: Methods Concurrency Matrix

	Webservice	main	webservice	serveClient	LOG	transfer	transferHeader	transferData
Webservice	⧻	⧻	⧻	⧻	⧻	⧻	⧻	⧻
main	⧻							
webservice	⧻							
serveClient	⧻							
LOG	⧻							
transfer	⧻							
transferHeader	⧻							
transferData	⧻							

2 Server

Table 5: Methods Requires Clause Satisfiability

Method	Satisfiability
Server	✓
main	✓

Table 6: State Transition Matrix

	alive
alive	↑

Table 7: Methods Concurrency Matrix

	Server	main
Server	⌈	⌈
main	⌈	⌈

3 Client

Table 8: Methods Requires Clause Satisfiability

Method	Satisfiability
Client	✓
main	✓

Table 9: State Transition Matrix

	alive
alive	↑

Table 10: Methods Concurrency Matrix

	Client	main
Client	⦿	⦿
main	⦿	

4 Abbreviation

Table 11: Used Abbreviation

Symbol	Meaning
✓	requires clause of the method is satisfiable
✗	requires clause of the method is unsatisfiable
↑	The row-state can be transitioned to the column-state
✕	The row-state cannot be transitioned to the column-state
	The row-method can be possibly executed parallel with the column-method
⋈	The row-method cannot be executed parallel with the column-method

5 Annotated Version of Sequential Java Program generated by Sip4j

```
1 package outputs;
2 import edu.cmu.cs.plural.annot.*;
3
4 @ClassStates({@State(name = "alive")})
5 class Webserver {
6   @Perm(ensures="unique(this) in alive")
7   Webserver() { }
8
9   @Perm(requires="none(this) in alive",
10  ensures="unique(this) in alive")
11   void main(String[] args) {
12   }
13   @Perm(requires="none(this) in alive",
14  ensures="unique(this) in alive")
15   void webserver() {
16   }
17   @Perm(requires="full(#0) in alive",
18  ensures="full(#0) in alive")
19   void serveClient(Socket socketClient) {
20   }
21
22   void LOG(String msg, Object... args) {
23   }
24
25   void transfer(OutputStream outStream, File file) {
26   }
27
28   void transferHeader(Writer writer, File file) {
29   }
30
31   void transferData(Writer outWriter, Reader inReader, long count) {
32   }
33 }
34 }ENDOFCLASS
35
36 @ClassStates({@State(name = "alive")})
37
38 class Server {
39   @Perm(ensures="unique(this) in alive")
40   Server() { }
41
42   void main(String argv[]) {
43   }
44 }
45 }ENDOFCLASS
46
47 @ClassStates({@State(name = "alive")})
48
49 class Client {
50   @Perm(ensures="unique(this) in alive")
51   Client() { }
52
53   void main(String argv[]) {
54   }
55 }
56 }ENDOFCLASS
```