Summary

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

Table 1: Sip4J Analysis Summary

Classes	Methods	States	Unreachable clauses	Unreachable states	Possible concurrent methods	Total. no. of method pairs	No. of concurrent method pairs	Percentage of concurrent methods pairs
Webserver	8	1	0	0	7	36	22	61
Server	2	1	0	0	0	3	0	0
Client	2	1	0	0	1	3	1	33
Total Classes=3	12	3	0	0	8	42	23	55

Contents

1 Webserver

Table 2: Method's Satisfiability(Code Reachabiity Analysis

Method	Satisfiability
Webserver	$\surd$
main	$\surd$
webserver	$\surd$
serveClient	$\surd$
LOG	$\surd$
transfer	$\surd$
transferHeader	$\surd$
transferData	$\surd$

Table 3: State Transition Matrix

	alive
alive	$\uparrow$

Table 4: Methods Concurrency Matrix

	Webserver	main	webserver	serveClient	LOG	transfer	transferHeader	transferData
Webserver	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$
main	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$
webserver	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$
serveClient	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$
LOG	$\nparallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$
transfer	$\nparallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$
transferHeader	$\nparallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$
transferData	$\nparallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$

2 Server

Table 5: Method's Satisfiability(Code Reachabiity Analysis

Method	Satisfiability
Server	√
main	√

Table 6: State Transition Matrix

	alive
alive	↑

Table 7: Methods Concurrency Matrix

	Server	main
Server	∦	∦
main	∦	∦

3 Client

Table 8: Method's Satisfiability(Code Reachabiity Analysis

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
$\surd$	requires clause of the method is satisfiable
$\times$	requires clause of the method is unsatisfiable
$\uparrow$	The row-state can be transitioned to the column-state
$\times$	The row-state cannot be transitioned to the column-state
$\parallel$	The row-method can be possibly executed parallel with the column-method
$\nparallel$	The row-method cannot be executed parallel with the column-method

5 Annotated version of the input 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="unique(this) in alive",
10 ensures="unique(this) in alive")
11   void main(String[] args) {
12
13 }
14 @Perm(requires="unique(this) in alive",
15 ensures="unique(this) in alive")
16   void webserver() {
17
18 }
19 @Perm(requires="share(this) in alive",
20 ensures="share(this) in alive")
21   void serveClient(Socket socketClient) {
22
23 }
24
25   void LOG(String msg, Object... args) {
26
27 }
28
29   void transfer(OutputStream outStream, File file) {
30
31 }
32
33   void transferHeader(Writer writer, File file) {
34
35 }
36
37   void transferData(Writer outWriter, Reader inReader, long count) {
38
39 }
40
41 }ENDOFCLASS
42
43 @ClassStates({@State(name = "alive")})
44
45 class Server {
46 @Perm(ensures="unique(this) in alive")
47 Server() {   }
48
49 @Perm(requires="unique(this) in alive",
50 ensures="unique(this) in alive")
51   void main(String argv[]) {
52
53 }
54
55 }ENDOFCLASS
56
57 @ClassStates({@State(name = "alive")})
58
59 class Client {
60 @Perm(ensures="unique(this) in alive")
61 Client() {   }
62
63
64   void main(String argv[]) {
65
66 }
67
68 }ENDOFCLASS
```