

Summary

Sink States:0(0×10^0)

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
SeriesTest	6	1	0	0	5	21	9	43
JGFTimer	9	1	0	0	3	45	6	13
JGFInstrumentor	13	1	0	0	12	91	12	13
JGFSeriesBenchSizeB	2	1	0	0	0	3	0	0
JGFSeriesBench	7	1	0	0	1	28	1	4
Total Classes=5	37	5	0	0	21	188	28	15

Contents

1	SeriesTest	3
2	JGFTimer	4
3	JGFInstrumentor	5
4	JGFSeriesBenchSizeB	6
5	JGFSeriesBench	7
6	Abbreviation	8
7	Annotated version of the input program generated by Sip4J	9

1 SeriesTest

Table 2: Method's Satisfiability(Code Reachabiity Analysis

Method	Satisfiability
SeriesTest	✓
buildTestData	✓
Do	✓
TrapezoidIntegrate	✓
thefunction	✓
freeTestData	✓

Table 3: State Transition Matrix

	alive
alive	↑

Table 4: Methods Concurrency Matrix

	SeriesTest	buildTestData	Do	TrapezoidIntegrate	thefunction	freeTestData
SeriesTest	⌘	⌘	⌘	⌘	⌘	⌘
buildTestData	⌘	⌘	⌘	⌘	⌘	⌘
Do	⌘	⌘	⌘	⌘	⌘	⌘
TrapezoidIntegrate	⌘	⌘	⌘	⌘	⌘	⌘
thefunction	⌘	⌘	⌘	⌘	⌘	⌘
freeTestData	⌘	⌘	⌘	⌘	⌘	⌘

2 JGFTimer

Table 5: Method's Satisfiability(Code Reachabiity Analysis

Method	Satisfiability
JGFTimer	✓
reset	✓
start	✓
stop	✓
addops	✓
perf	✓
longprint	✓
print	✓
printperf	✓

Table 6: State Transition Matrix

	alive
alive	↑

Table 7: Methods Concurrency Matrix

	JGFTimer	reset	start	stop	addops	perf	longprint	print	printperf
JGFTimer	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
reset	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
start	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
stop	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
addops	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
perf	⌘	⌘	⌘	⌘	⌘			⌘	
longprint	⌘	⌘	⌘	⌘	⌘			⌘	
print	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
printperf	⌘	⌘	⌘	⌘	⌘			⌘	

3 JGFInstrumentor

Table 8: Method's Satisfiability(Code Reachabiity Analysis

Method	Satisfiability
JGFInstrumentor	✓
addTimer	✓
addOpsToTimer	✓
startTimer	✓
stopTimer	✓
readTimer	✓
resetTimer	✓
printTimer	✓
printperfTimer	✓
storeData	✓
retrieveData	✓
printHeader	✓
main	✓

Table 9: State Transition Matrix

	alive
alive	↑

Table 10: Methods Concurrency Matrix

	JGFInstrumentor	addTimer	addOpsToTimer	startTimer	stopTimer	readTimer	resetTimer	printTimer	printperfTimer	storeData	retrieveData	printHeader	main
JGFInstrumentor	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
addTimer	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
addOpsToTimer	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
startTimer	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
stopTimer	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
readTimer	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
resetTimer	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
printTimer	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
printperfTimer	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
storeData	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
retrieveData	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
printHeader	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
main	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘

4 JGFSeriesBenchSizeB

Table 11: Method's Satisfiability(Code Reachabiity Analysis

Method	Satisfiability
JGFSeriesBenchSizeB	✓
main	✓

Table 12: State Transition Matrix

	alive
alive	↑

Table 13: Methods Concurrency Matrix

	JGFSeriesBenchSizeB	main
JGFSeriesBenchSizeB	⌘	⌘
main	⌘	⌘

5 JGFSeriesBench

Table 14: Method's Satisfiability(Code Reachabiity Analysis

Method	Satisfiability
JGFSeriesBench	✓
JGFrun	✓
JGFsetsize	✓
JGFinitialise	✓
JGFkernel	✓
JGFvalidate	✓
JGFtidyup	✓

Table 15: State Transition Matrix

	alive
alive	↑

Table 16: Methods Concurrency Matrix

	JGFSeriesBench	JGFrun	JGFsetsize	JGFinitialise	JGFkernel	JGFvalidate	JGFtidyup
JGFSeriesBench	⌘	⌘	⌘	⌘	⌘	⌘	⌘
JGFrun	⌘	⌘	⌘	⌘	⌘	⌘	⌘
JGFsetsize	⌘	⌘	⌘	⌘	⌘	⌘	⌘
JGFinitialise	⌘	⌘	⌘	⌘	⌘	⌘	⌘
JGFkernel	⌘	⌘	⌘	⌘	⌘	⌘	⌘
JGFvalidate	⌘	⌘	⌘	⌘	⌘	⌘	⌘
JGFtidyup	⌘	⌘	⌘	⌘	⌘	⌘	⌘

6 Abbreviation

Table 17: 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

7 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 SeriesTest {
6   @Perm(ensures="unique(this) in alive")
7   SeriesTest() { }
8
9   @Perm(requires="unique(this) in alive",
10  ensures="unique(this) in alive")
11   void buildTestData() {
12
13   }
14   @Perm(requires="share(this) in alive",
15  ensures="share(this) in alive")
16   void Do() {
17
18   }
19
20   private double TrapezoidIntegrate(double x0, double x1, int nsteps, double omegan, int select) {
21     return 0;
22
23   }
24
25   private double thefunction(double x, double omegan, int select) {
26     return 0;
27
28   }
29   @Perm(requires="unique(this) in alive",
30  ensures="unique(this) in alive")
31   void freeTestData() {
32
33   }
34
35 }ENDOFCLASS
36
37 @ClassStates({@State(name = "alive")})
38
39 class JGFTimer {
40   @Perm(ensures="unique(this) in alive")
41   JGFTimer() { }
42
43   @Perm(requires="share(this) in alive",
44  ensures="share(this) in alive")
45   public void reset() {
46
47   }
48   @Perm(requires="share(this) in alive",
49  ensures="share(this) in alive")
50   public void start() {
51
52   }
53   @Perm(requires="share(this) in alive",
54  ensures="share(this) in alive")
55   public void stop() {
56
57   }
58   @Perm(requires="share(this) in alive",
59  ensures="share(this) in alive")
60   public void addops(double count) {
61
62   }
63   @Perm(requires="pure(this) in alive",
64  ensures="pure(this) in alive")
65   public double perf() {
66     return 0;
67
68   }
69   @Perm(requires="pure(this) in alive",
70  ensures="pure(this) in alive")
71   public void longprint() {
72
73   }
74   @Perm(requires="share(this) in alive",
75  ensures="share(this) in alive")
76   public void print() {
77
78   }
```

```

79 @Perm(requires="pure(this) in alive",
80 ensures="pure(this) in alive")
81 public void printperf() {
82
83 }
84
85 }ENDOFCLASS
86
87 @ClassStates({@State(name = "alive")})
88
89 class JGFInstrumentor {
90 @Perm(ensures="unique(this) in alive")
91 JGFInstrumentor() { }
92
93 @Perm(requires="share(this) in alive",
94 ensures="share(this) in alive")
95 void addTimer(String name) {
96
97 }
98 @Perm(requires="share(this) in alive",
99 ensures="share(this) in alive")
100 void addOpsToTimer(String name, double count) {
101
102 }
103 @Perm(requires="share(this) in alive",
104 ensures="share(this) in alive")
105 void startTimer(String name) {
106
107 }
108 @Perm(requires="share(this) in alive",
109 ensures="share(this) in alive")
110 void stopTimer(String name) {
111
112 }
113 @Perm(requires="share(this) in alive",
114 ensures="share(this) in alive")
115 double readTimer(String name) {
116 return 0;
117
118 }
119 @Perm(requires="share(this) in alive",
120 ensures="share(this) in alive")
121 void resetTimer(String name) {
122
123 }
124 @Perm(requires="share(this) in alive",
125 ensures="share(this) in alive")
126 void printTimer(String name) {
127
128 }
129 @Perm(requires="share(this) in alive",
130 ensures="share(this) in alive")
131 void printperfTimer(String name) {
132
133 }
134 @Perm(requires="share(this) in alive",
135 ensures="share(this) in alive")
136 void storeData(String name, Object obj) {
137
138 }
139 @Perm(requires="share(this) in alive",
140 ensures="share(this) in alive")
141 void retrieveData(String name, Object obj) {
142
143 }
144
145 void printHeader(int section, int size) {
146
147 }
148 @Perm(requires="unique(this) in alive",
149 ensures="unique(this) in alive")
150 void main(String argv[]) {
151
152 }
153
154 }ENDOFCLASS
155
156 @ClassStates({@State(name = "alive")})
157
158 class JGFSeriesBenchSizeB {
159 @Perm(ensures="unique(this) in alive")

```

```

160 JGFSeriesBenchSizeB() { }

162 @Perm(requires="unique(this) in alive",
163 ensures="unique(this) in alive")
164 void main(String argv[]) {

166 }

168 }ENDOFCLASS

170 @ClassStates({@State(name = "alive")})

172 class JGFSeriesBench {
173 @Perm(ensures="unique(this) in alive")
174 JGFSeriesBench() { }

176 @Perm(requires="unique(this) in alive",
177 ensures="unique(this) in alive")
178 public void JGFrun(int size) {

180 }

181 @Perm(requires="share(this) in alive",
182 ensures="share(this) in alive")
183 public void JGFsetsize(int size) {

185 }

186 @Perm(requires="unique(this) in alive",
187 ensures="unique(this) in alive")
188 public void JGFinitialise() {

190 }

191 @Perm(requires="share(this) in alive",
192 ensures="share(this) in alive")
193 public void JGFkernel() {

195 }

196 @Perm(requires="pure(this) in alive",
197 ensures="pure(this) in alive")
198 public void JGFvalidate() {

200 }

201 @Perm(requires="unique(this) in alive",
202 ensures="unique(this) in alive")
203 public void JGFtidyup() {

205 }

207 }ENDOFCLASS

```