

Summary

Sink States:0(0×10^0)

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
Activator	8	1	3	0	7	36	15	42
SampleAction	5	1	0	0	4	15	7	47
JMLAnnotatedJavaClass	6	1	0	0	5	21	5	24
PluralParser	34	1	0	0	34	595	127	21
EJmlSpecification	12	1	0	0	10	78	10	13
EGhost	6	1	0	0	5	21	12	57
Time	2	1	0	0	0	3	0	0
FileReader	2	1	0	0	1	3	1	33
UserSelectedClassesAnalysis	12	1	0	0	10	78	34	44
EVMDDSMCGenerator	11	1	0	0	8	66	35	53
EPackage	6	1	0	0	5	21	12	57
EGeneratedPluralSpecification	3	1	0	0	0	6	0	0
ESMCModel	60	1	0	0	59	1830	339	19
WorkspaceUtilities	9	1	0	0	8	45	30	67
EClass	24	1	0	0	23	300	221	74
EGraphWriter	5	1	0	0	4	15	4	27
EState	10	1	0	0	9	55	35	64
EOutputLatex	22	1	0	0	19	253	19	8
EMethod	22	1	0	0	21	253	126	50
SMCVisitor	6	1	0	0	5	21	14	67
PulseSettings	11	1	0	0	10	66	40	61
EField	11	1	0	0	10	66	27	41
ESpecification	8	1	0	0	7	36	18	50
EParameter	9	1	0	0	8	45	26	58
EInvariant	11	1	0	0	10	66	40	61
EBoolInvariant	3	1	0	0	2	6	3	50
EDim	5	1	0	0	4	15	7	47
specificationStruct	1	1	0	0	0	1	0	0
Clause	1	1	0	0	0	1	0	0
Signature	1	1	0	0	0	1	0	0
MethodFindVisitor	2	1	0	0	0	3	0	0
GAPHandler	7	1	0	0	6	28	18	64
GAPIFileAction	5	1	1	0	4	15	4	27
Main	5	1	0	0	4	15	7	47

TypestateReturn	1	1	0	0	0	1	0	0
AtApPermissionReturn	1	1	0	0	0	1	0	0
AccesspermissionReturn	1	1	0	0	0	1	0	0
PluralLexer	63	1	0	0	62	2016	1952	97
DFA7	2	1	0	0	1	3	1	33
EAPTypeState	5	1	0	0	0	15	0	0
Total Classes=40	418	40	4	0	365	6116	3189	52

Contents

1	Activator	4
2	SampleAction	5
3	JMLAnnotatedJavaClass	6
4	PluralParser	7
5	EJmlSpecification	9
6	EGhost	10
7	Time	11
8	FileReader	12
9	UserSelectedClassesAnalysis	13
10	EVMDDSMCGenerator	15
11	EPackage	16
12	EGeneratedPluralSpecification	17
13	ESMCMModel	18
14	WorkspaceUtilities	21
15	EClass	22
16	EGrarphWriter	24
17	EState	25
18	EOutputLatex	26
19	EMethod	28
20	SMCVisitor	30
21	PulseSettings	31
22	EField	32
23	ESpecification	33
24	EParameter	34
25	EInvariant	35
26	EBoolInvariant	36
27	EDim	37
28	specificationStruct	38
29	Clause	39

30 Signature	40
31 MethodFindVisitor	41
32 GAPHandler	42
33 GAPIFileAction	43
34 Main	44
35 TypestateReturn	45
36 AtApPermissionReturn	46
37 AccesspermissionReturn	47
38 PluralLexer	48
39 DFA7	51
40 EAPTypeState	52
41 Abbreviation	53
42 Annotated Version of Sequential Java Program generated by Sip4j	54

1 Activator

Table 2: Methods Requires Clause Satisfiability

Method	Satisfiability
Activator	✓
start	✓
stop	✓
getDefault	✓
start	✗
stop	✗
getDefault	✗
getImageDescriptor	✓

Table 3: State Transition Matrix

	alive
alive	↑

Table 4: Methods Concurrency Matrix

	Activator	start	stop	getDefault	start	stop	getDefault	getImageDescriptor
Activator	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
start	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
stop	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getDefault	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
start	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
stop	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getDefault	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getImageDescriptor	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘

2 SampleAction

Table 5: Methods Requires Clause Satisfiability

Method	Satisfiability
SampleAction	✓
run	✓
selectionChanged	✓
dispose	✓
init	✓

Table 6: State Transition Matrix

	alive
alive	↑

Table 7: Methods Concurrency Matrix

	SampleAction	run	selectionChanged	dispose	init
SampleAction	⌘	⌘	⌘	⌘	⌘
run	⌘	⌘			⌘
selectionChanged	⌘				
dispose	⌘				
init	⌘	⌘			⌘

3 JMLAnnotatedJavaClass

Table 8: Methods Requires Clause Satisfiability

Method	Satisfiability
JMLAnnotatedJavaClass	✓
translateJMLAnnotationsToPlural	✓
translateClassSpecifications	✓
parseAndStoreJMLAnnotation	✓
translateMethodSpecification	✓
readFileAsString	✓

Table 9: State Transition Matrix

	alive
alive	↑

Table 10: Methods Concurrency Matrix

	JMLAnnotatedJavaClass	translateJMLAnnotationsToPlural	translateClassSpecifications	parseAndStoreJMLAnnotation	translateMethodSpecification	readFileAsString
JMLAnnotatedJavaClass	⌘	⌘	⌘	⌘	⌘	⌘
translateJMLAnnotationsToPlural	⌘	⌘	⌘	⌘	⌘	⌘
translateClassSpecifications	⌘	⌘	⌘	⌘	⌘	⌘
parseAndStoreJMLAnnotation	⌘	⌘	⌘	⌘	⌘	⌘
translateMethodSpecification	⌘	⌘	⌘	⌘	⌘	⌘
readFileAsString	⌘	⌘	⌘	⌘	⌘	⌘

4 PluralParser

Table 11: Methods Requires Clause Satisfiability

Method	Satisfiability
PluralParser	✓
jmlSpecifications	✓
jmlClassSpecifications	✓
jmlGhostDeclaration	✓
jmlMethodSpecification	✓
jmlRequires	✓
jmlReq	✓
jmlLessThanEqualReq	✓
jmlEnsures	✓
jmlEns	✓
getTokenNames	✓
getGrammarFileName	✓
jmlOldEns	✓
specifications	✓
perm	✓
requiresensuresClause	✓
requiresClause	✓
reaccesspermissionTypestates	✓
accesspermission	✓
typestate	✓
attype	✓
atpermission	✓
usevalue	✓
classstates	✓
startClassstates	✓
state	✓
endclassstates	✓
refine	✓
states	✓
dimension	✓
item	✓
invariant	✓
condition	✓
other	✓

Table 12: State Transition Matrix

	alive
alive	↑

Table 13: Methods Concurrency Matrix

	PluralParser	jmlSpecifications	jmlClassSpecifications	jmlGhostDeclaration	jmlMethodSpecification	jmlRequires	jmlReq	jmlLessThanEqualReq	jmlEnsures	jmlEns	getTokenNames	getGrammarFileName	jmlOldEns	specifications	perm	requiresensuresClause	requiresClause	reaccesspermissionTypestates	accesspermission	typestate
PluralParser	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
jmlSpecifications	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
jmlClassSpecifications	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
jmlGhostDeclaration	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
jmlMethodSpecification	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
jmlRequires	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
jmlReq	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
jmlLessThanEqualReq	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
jmlEnsures	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
jmlEns	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
getTokenNames	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
getGrammarFileName	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
jmlOldEns	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
specifications	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
perm	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
requiresensuresClause	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
requiresClause	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
reaccesspermissionTypestates	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
accesspermission	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
typestate	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
attype	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
atapperrmission	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
usevalue	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
classtates	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
startClasstates	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
state	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
endclasstates	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
refine	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
states	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
dimension	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
item	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
invariant	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
condition	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
other	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

5 EJmlSpecification

Table 14: Methods Requires Clause Satisfiability

Method	Satisfiability
EJmlSpecification	✓
setDimensionName	✓
setEnsures	✓
setPerm	✓
addRequires	✓
reset	✓
JmlClassSpec2PluralClassSpec	✓
noRequires	✓
getPerm	✓
determineEnsures	✓
oneRequires	✓
moreRequires	✓

Table 15: State Transition Matrix

	alive
alive	↑

Table 16: Methods Concurrency Matrix

	EJmlSpecification	setDimensionName	setEnsures	setPerm	addRequires	reset	JmlClassSpec2PluralClassSpec	noRequires	getPerm	determineEnsures	oneRequires	moreRequires
EJmlSpecification	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setDimensionName	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setEnsures	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setPerm	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
addRequires	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
reset	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
JmlClassSpec2PluralClassSpec	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
noRequires	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getPerm	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
determineEnsures	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
oneRequires	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
moreRequires	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘

6 EGhost

Table 17: Methods Requires Clause Satisfiability

Method	Satisfiability
EGhost	✓
setDimensionName	✓
setDimensionValues	✓
getDimensionName	✓
getLowValueofInv	✓
getHighValueofInv	✓

Table 18: State Transition Matrix

	alive
alive	↑

Table 19: Methods Concurrency Matrix

	EGhost	setDimensionName	setDimensionValues	getDimensionName	getLowValueofInv	getHighValueofInv
EGhost	⌘	⌘	⌘	⌘	⌘	⌘
setDimensionName	⌘	⌘	⌘			
setDimensionValues	⌘	⌘	⌘			
getDimensionName	⌘					
getLowValueofInv	⌘					
getHighValueofInv	⌘					

7 Time

Table 20: Methods Requires Clause Satisfiability

Method	Satisfiability
Time	✓
toString	✓

Table 21: State Transition Matrix

	alive
alive	↑

Table 22: Methods Concurrency Matrix

	Time	toString
Time	⧻	⧻
toString	⧻	⧻

8 FileReader

Table 23: Methods Requires Clause Satisfiability

Method	Satisfiability
FileReader	✓
readFile	✓

Table 24: State Transition Matrix

	alive
alive	↑

Table 25: Methods Concurrency Matrix

	FileReader	readFile
FileReader	⌈	⌈
readFile	⌈	

9 UserSelectedClassesAnalysis

Table 26: Methods Requires Clause Satisfiability

Method	Satisfiability
UserSelectedClassesAnalysis	✓
getCompilationUnit	✓
analyzeFromCommandLine	✓
analyzeFromPlugin	✓
callModelCheckerThroughCommandLine	✓
getTime	✓
CreatePdfSummary_CommandLine	✓
callModelCheckerThroughPlugin	✓
createPdfSummaryPlugin	✓
makePdfCommandLine	✓
makePdfPlugin	✓
printMethodMetrics	✓

Table 27: State Transition Matrix

	alive
alive	↑

Table 28: Methods Concurrency Matrix

	UserSelectedClassesAnalysis	getCompilationUnit	analyzeFromCommandLine	analyzeFromPlugin	callModelCheckerThroughCommandLine	getTime	CreatePdfSummary_CommandLine	callModelCheckerThroughPlugin	createPdfSummaryPlugin	makePdfCommandLine	makePdfPlugin	printMethodMetrics
UserSelectedClassesAnalysis	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getCompilationUnit	⌘						⌘					
analyzeFromCommandLine	⌘		⌘	⌘	⌘	⌘	⌘	⌘				⌘
analyzeFromPlugin	⌘		⌘	⌘	⌘	⌘	⌘	⌘				⌘
callModelCheckerThroughCommandLine	⌘		⌘	⌘	⌘	⌘	⌘	⌘				⌘
getTime	⌘		⌘	⌘	⌘	⌘	⌘	⌘				⌘
CreatePdfSummary_CommandLine	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
callModelCheckerThroughPlugin	⌘		⌘	⌘	⌘	⌘	⌘	⌘				⌘
createPdfSummaryPlugin	⌘						⌘					
makePdfCommandLine	⌘						⌘					

makePdfPlugin	⌘						⌘					
printMethodMetrics	⌘		⌘	⌘	⌘	⌘	⌘	⌘				⌘

10 EVMDDSMCGenerator

Table 29: Methods Requires Clause Satisfiability

Method	Satisfiability
EVMDDSMCGenerator	✓
reset	✓
addBoolStateInvariant	✓
getPkgObject	✓
modifyConstructorSpecifications	✓
addState	✓
addRequiresParam_AP_TS	✓
addEnsuresParamAPTS	✓
addStateInvariant	✓
addDimensionValue	✓
addEnsuresAPTS	✓

Table 30: State Transition Matrix

	alive
alive	↑

Table 31: Methods Concurrency Matrix

	EVMDDSMCGenerator	reset	addBoolStateInvariant	getPkgObject	modifyConstructorSpecifications	addState	addRequiresParam_AP_TS	addEnsuresParamAPTS	addStateInvariant	addDimensionValue	addEnsuresAPTS
EVMDDSMCGenerator	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
reset	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
addBoolStateInvariant	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getPkgObject	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
modifyConstructorSpecifications	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
addState	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
addRequiresParam_AP_TS	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
addEnsuresParamAPTS	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
addStateInvariant	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
addDimensionValue	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
addEnsuresAPTS	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘

11 EPackage

Table 32: Methods Requires Clause Satisfiability

Method	Satisfiability
EPackage	✓
getClasses	✓
getTotalReachableStates	✓
getName	✓
getTotalStates	✓
getSinkStates	✓

Table 33: State Transition Matrix

	alive
alive	↑

Table 34: Methods Concurrency Matrix

	EPackage	getClasses	getTotalReachableStates	getName	getTotalStates	getSinkStates
EPackage	⌘	⌘	⌘	⌘	⌘	⌘
getClasses	⌘					
getTotalReachableStates	⌘					
getName	⌘			⌘	⌘	
getTotalStates	⌘			⌘	⌘	
getSinkStates	⌘					

12 EGeneratedPluralSpecification

Table 35: Methods Requires Clause Satisfiability

Method	Satisfiability
EGeneratedPluralSpecification	✓
createFromCommandLine	✓
createFromPlugin	✓

Table 36: State Transition Matrix

	alive
alive	↑

Table 37: Methods Concurrency Matrix

	EGeneratedPluralSpecification	createFromCommandLine	createFromPlugin
EGeneratedPluralSpecification	⌘	⌘	⌘
createFromCommandLine	⌘	⌘	⌘
createFromPlugin	⌘	⌘	⌘

13 ESMCModel

Table 38: Methods Requires Clause Satisfiability

Method	Satisfiability
ESMCMModel	✓
setK	✓
declarationsAndinitalizations	✓
initialize	✓
comment	✓
modelAlias	✓
isClassExist	✓
Transitions	✓
createInstanceInModel	✓
modelPrimePCandMethod	✓
startMethod	✓
modelPCConstructor	✓
modelAPs	✓
modelPCMethod	✓
startAPTSPARAM	✓
startPrimeTSPARAM	✓
modelPrimeConstructor	✓
modelInheritance	✓
modelPrimeAPStateInvariants	✓
starPrimeAP	✓
modelPrimeAP	✓
getAPIId	✓
modelEndPCMethod	✓
endMethod	✓
modelEndPCConstructor	✓
modelendConstructor	✓
updateBoolStateInvariants	✓
endPrimeAPTS	✓
endPrimeAPTSPARAM	✓
updateState	✓
updateStateInvariants	✓
modelBoolStateInvariants	✓
startAPTS	✓
modelState	✓
modelStateInvariants	✓
methodsReachability	✓
modelAP	✓
getObjectIndex	✓
updateTokens	✓
modelPrimePCCConstructor	✓
modelPrimePC	✓
defineKVariables	✓
defineVariables	✓
initilizeKVariables	✓
initilizeVariables	✓
createAlias	✓
isPrivateAndIndexEqualToZero	✓

getClass	✓
getDimensionIndex	✓
addIndexes	✓
createDimensionsObject	✓
createDimensionAsField	✓
createParentAsField	✓
createParentObject	✓
setInvariantVariableType	✓
addInvariantStateIndex	✓
Spec	✓
concurrentMethods	✓
statesAdjancyMatrix	✓
generateSMCmodelPlugin	✓

Table 39: State Transition Matrix

	alive
alive	↑

	ESMCModel	setK	declarationsAndinitializations	initialize	comment	modelAlias	isClassExist	Transitions	createInstanceInModel	modelPrimePCandMethod	startMethod	modelPCCConstructor	modelAPs	modelPCMethod	startAPTSPARAM	startPrimeTSPARAM	modelPrimeConstructor	modelInheritance	modelPrimeAPStateInvariants	starPrimeAP	modelPrimeAP
ESMCModel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
setK	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
declarationsAndinitializations	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
initialize	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
comment	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
modelAlias	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
isClassExist	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Transitions	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
createInstanceInModel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
modelPrimePCandMethod	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
startMethod	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
modelPCCConstructor	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
modelAPs	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
modelPCMethod	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
startAPTSPARAM	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
startPrimeTSPARAM	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
modelPrimeConstructor	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

modelInheritance	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
modelPrimeAPStateInvariants	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
starPrimeAP	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
modelPrimeAP	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getAPIId	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
modelEndPCMethod	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
endMethod	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
modelEndPCConstructor	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
modelendConstructor	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
updateBoolStateInvariants	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
endPrimeAPTS	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
endPrimeAPTSPARAM	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
updateState	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
updateStateInvariants	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
modelBoolStateInvariants	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
startAPTS	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
modelState	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
modelStateInvariants	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
methodsReachability	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
modelAP	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getObjectIndex	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
updateTokens	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
modelPrimePCConstructor	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
modelPrimePC	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
defineKVVariables	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
defineVariables	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
initilizeKVVariables	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
initilizeVariables	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
createAlias	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
isPrivateAndIndexEqualToZero	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getClass	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getDimensionIndex	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
addIndexes	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
createDimensionsObject	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
createDimensionAsField	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
createParentAsField	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
createParentObject	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setInvariantVariableType	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
addInvariantStateIndex	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
Spec	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
concurrentMethods	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
statesAdjancyMatrix	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
generateSMCmodelPlugin	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘

14 WorkspaceUtilities

Table 41: Methods Requires Clause Satisfiability

Method	Satisfiability
WorkspaceUtilities	✓
getASTNodeFromCompilationUnit	✓
scanForCompilationUnits	✓
collectCompilationUnits	✓
findCompilationUnits	✓
getWorkspaceRelativeName	✓
parseCompilationUnits	✓
scanForMethodDeclarations	✓
scanForMethodDeclarationsFromAST	✓

Table 42: State Transition Matrix

	alive
alive	↑

Table 43: Methods Concurrency Matrix

	WorkspaceUtilities	getASTNodeFromCompilationUnit	scanForCompilationUnits	collectCompilationUnits	findCompilationUnits	getWorkspaceRelativeName	parseCompilationUnits	scanForMethodDeclarations	scanForMethodDeclarationsFromAST
WorkspaceUtilities	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getASTNodeFromCompilationUnit	⌘								
scanForCompilationUnits	⌘		⌘	⌘	⌘				
collectCompilationUnits	⌘		⌘	⌘	⌘				
findCompilationUnits	⌘		⌘	⌘	⌘				
getWorkspaceRelativeName	⌘								
parseCompilationUnits	⌘								
scanForMethodDeclarations	⌘								
scanForMethodDeclarationsFromAST	⌘								

15 EClass

Table 44: Methods Requires Clause Satisfiability

Method	Satisfiability
EClass	✓
getName	✓
getTransitions	✓
getReachableStates	✓
getStates	✓
getMethods	✓
getSuperClassName	✓
findStateIndex	✓
getIndex	✓
getFields	✓
getVariablesOfBooleanInvariants	✓
getDimensions	✓
createObject	✓
getConstructor	✓
setSuperClassName	✓
getTotalStates	✓
getTotalReachableStates	✓
addMethod	✓
addField	✓
addState	✓
getLastObjectIndex	✓
addDimension	✓
hasMoreThanOneDimension	✓
addClassStatesSpecifications	✓

Table 45: State Transition Matrix

	alive
alive	↑

Table 46: Methods Concurrency Matrix

	EClass	getName	getTransitions	getReachableStates	getStates	getMethods	getSuperClassName	findStateIndex	getIndex	getFields	getVariablesOfBooleanInvariants	getDimensions	createObject	getConstructor	setSuperClassName	getTotalStates	getTotalReachableStates	addMethod	addField
EClass	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘

getName	✓	✓												✓	✓	✓	✓		✓	✓
getTransitions	✓													✓					✓	
getReachableStates	✓													✓					✓	
getStates	✓													✓					✓	
getMethods	✓													✓					✓	
getSuperClassName	✓													✓					✓	
findStateIndex	✓													✓					✓	
getIndex	✓													✓					✓	
getFields	✓													✓					✓	
getVariablesOfBooleanInvariants	✓													✓					✓	
getDimensions	✓													✓					✓	
createObject	✓	✓												✓	✓	✓	✓		✓	✓
getConstructor	✓	✓												✓	✓	✓	✓		✓	✓
setSuperClassName	✓	✓												✓	✓	✓	✓		✓	✓
getTotalStates	✓	✓												✓	✓	✓	✓		✓	✓
getTotalReachableStates	✓													✓					✓	
addMethod	✓	✓												✓	✓	✓	✓		✓	✓
addField	✓	✓												✓	✓	✓	✓		✓	✓
addState	✓	✓												✓	✓	✓	✓		✓	✓
getLastObjectIndex	✓													✓					✓	
addDimension	✓	✓												✓	✓	✓	✓		✓	✓
hasMoreThanOneDimension	✓	✓												✓	✓	✓	✓		✓	✓
addClassStatesSpecifications	✓													✓					✓	

16 EGrarphWriter

Table 47: Methods Requires Clause Satisfiability

Method	Satisfiability
EGrarphWriter	✓
addTrnsitions	✓
createGraph	✓
setNumberOfUnReachableMethods	✓
getNumberOfUnReachableMethods	✓

Table 48: State Transition Matrix

	alive
alive	↑

Table 49: Methods Concurrency Matrix

	EGrarphWriter	addTrnsitions	createGraph	setNumberOfUnReachableMethods	getNumberOfUnReachableMethods
EGrarphWriter	⚡	⚡	⚡	⚡	⚡
addTrnsitions	⚡	⚡	⚡	⚡	⚡
createGraph	⚡	⚡	⚡	⚡	⚡
setNumberOfUnReachableMethods	⚡	⚡	⚡	⚡	⚡
getNumberOfUnReachableMethods	⚡	⚡	⚡	⚡	⚡

17 EState

Table 50: Methods Requires Clause Satisfiability

Method	Satisfiability
EState	✓
isReachable	✓
getName	✓
setReachability	✓
getInvariants	✓
getStateIndex	✓
getBoolInvariants	✓
addBoolInvariant	✓
setIndex	✓
isReachableState	✓

Table 51: State Transition Matrix

	alive
alive	↑

Table 52: Methods Concurrency Matrix

	EState	isReachable	getName	setReachability	getInvariants	getStateIndex	getBoolInvariants	addBoolInvariant	setIndex	isReachableState
EState	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
isReachable	⌘									
getName	⌘									
setReachability	⌘			⌘				⌘	⌘	⌘
getInvariants	⌘									
getStateIndex	⌘									
getBoolInvariants	⌘									
addBoolInvariant	⌘			⌘				⌘	⌘	⌘
setIndex	⌘			⌘				⌘	⌘	⌘
isReachableState	⌘			⌘				⌘	⌘	⌘

18 EOutputLatex

Table 53: Methods Requires Clause Satisfiability

Method	Satisfiability
EOutputLatex	✓
create_CommandLine	✓
addUsePackages	✓
writeToLatex	✓
WriteSummary	✓
addSummaryTableHeaders	✓
addSummaryTableRows	✓
writeStateTransitionMatrix	✓
addSTMColumnsHeaders	✓
writeMethodConcurrencyMatrix	✓
addConcurrencyMatrixHeaders	✓
writeAbbervations	✓
setText	✓
parseTransitions	✓
parseSinkStates	✓
create_Plugin	✓
addConcurrencyMatrixRows	✓
getConcurrencyValue	✓
addSummaryTableColumns	✓
addSTMNumberOfColumns	✓
getStateReachabilityValue	✓
getMethod	✓

Table 54: State Transition Matrix

	alive
alive	↑

Table 55: Methods Concurrency Matrix

	EOutputLatex	create_CommandLine	addUsePackages	writeToLatex	WriteSummary	addSummaryTableHeaders	addSummaryTableRows	writeStateTransitionMatrix	addSTMColumnsHeaders	writeMethodConcurrencyMatrix	addConcurrencyMatrixHeaders	writeAbbervations	setText	parseTransitions	parseSinkStates	create_Plugin	addConcurrencyMatrixRows	getConcurrencyValue	addSummaryTableColumns
EOutputLatex	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
create_CommandLine	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
addUsePackages	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘

writeToLatex	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈
WriteSummary	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈
addSummaryTableHeaders	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈
addSummaryTableRows	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈
writeStateTransitionMatrix	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈
addSTMColumnsHeaders	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈
writeMethodConcurrencyMatrix	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈
addConcurrencyMatrixHeaders	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈
writeAbbervations	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈
setText	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈
parseTransitions	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈
parseSinkStates	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈
create_Plugin	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈
addConcurrencyMatrixRows	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈
getConcurrencyValue	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈
addSummaryTableColumns	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈
addSTMNumberOfColumns	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈
getStateReachabilityValue	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈
getMethod	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈	⌈

19 EMethod

Table 56: Methods Requires Clause Satisfiability

Method	Satisfiability
EMethod	✓
getRequiresClauseSatisfiability	✓
setReturnType	✓
setIdentifier	✓
getName	✓
getEnsuresAPTS	✓
isConcurrentMethod	✓
setJMLPermission	✓
getJMLPermission	✓
getCaseNumber	✓
setCaseNumber	✓
setName	✓
getReturnType	✓
getIdentifier	✓
addParameter	✓
getRequiresAPTS	✓
getParameters	✓
setIndex	✓
getIndex	✓
setRequiresClauseSatisfiability	✓
setConcurrentMethod	✓
addSpecifications	✓

Table 57: State Transition Matrix

	alive
alive	↑

Table 58: Methods Concurrency Matrix

	EMethod	getRequiresClauseSatisfiability	setReturnType	setIdentifier	getName	getEnsuresAPTS	isConcurrentMethod	setJMLPermission	getJMLPermission	getCaseNumber	setCaseNumber	setName	getReturnType	getIdentifier	addParameter	getRequiresAPTS	getParameters	setIndex	getIndex	setRequiresClauseSatisfiability
EMethod	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getRequiresClauseSatisfiability	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setReturnType	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setIdentifier	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘

getName	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getEnsuresAPTS	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
isConcurrentMethod	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setJMLPermission	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getJMLPermission	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getCaseNumber	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setCaseNumber	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setName	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getReturnType	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getIdentifier	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
addParameter	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getRequiresAPTS	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getParameters	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setIndex	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getIndex	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setRequiresClauseSatisfiability	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setConcurrentMethod	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
addSpecifications	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘

20 SMCVisitor

Table 59: Methods Requires Clause Satisfiability

Method	Satisfiability
SMCVisitor	✓
addUnparsedSpecifications	✓
postVisit	✓
visit	✓
endVisit	✓
callParser	✓

Table 60: State Transition Matrix

	alive
alive	↑

Table 61: Methods Concurrency Matrix

	SMCVisitor	addUnparsedSpecifications	postVisit	visit	endVisit	callParser
SMCVisitor	⌘	⌘	⌘	⌘	⌘	⌘
addUnparsedSpecifications	⌘					
postVisit	⌘					
visit	⌘			⌘		
endVisit	⌘					
callParser	⌘					

21 PulseSettings

Table 62: Methods Requires Clause Satisfiability

Method	Satisfiability
PulseSettings	✓
getInheritance	✓
getFullModel	✓
getInvariants	✓
setInvariants	✓
setAliasPerObject	✓
setFullModel	✓
setDimensions	✓
setInheritance	✓
getAliasPerObject	✓
getDimensions	✓

Table 63: State Transition Matrix

	alive
alive	↑

Table 64: Methods Concurrency Matrix

	PulseSettings	getInheritance	getFullModel	getInvariants	setInvariants	setAliasPerObject	setFullModel	setDimensions	setInheritance	getAliasPerObject	getDimensions
PulseSettings	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getInheritance	⌘										
getFullModel	⌘										
getInvariants	⌘										
setInvariants	⌘				⌘	⌘	⌘	⌘	⌘		
setAliasPerObject	⌘				⌘	⌘	⌘	⌘	⌘		
setFullModel	⌘				⌘	⌘	⌘	⌘	⌘		
setDimensions	⌘				⌘	⌘	⌘	⌘	⌘		
setInheritance	⌘				⌘	⌘	⌘	⌘	⌘		
getAliasPerObject	⌘										
getDimensions	⌘										

22 EField

Table 65: Methods Requires Clause Satisfiability

Method	Satisfiability
EField	✓
setName	✓
setType	✓
setModifier	✓
getObjectIndex	✓
getModifier	✓
setClassIndex	✓
getName	✓
getType	✓
setObjectIndex	✓
getClassIndex	✓

Table 66: State Transition Matrix

	alive
alive	↑

Table 67: Methods Concurrency Matrix

	EField	setName	setType	setModifier	getObjectIndex	getModifier	setClassIndex	getName	getType	setObjectIndex	getClassIndex
EField	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setName	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setType	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setModifier	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getObjectIndex	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getModifier	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setClassIndex	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getName	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getType	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setObjectIndex	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getClassIndex	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘

23 ESpecification

Table 68: Methods Requires Clause Satisfiability

Method	Satisfiability
ESpecification	✓
getParentClass	✓
getTS	✓
getAP	✓
clone	✓
setAPTS	✓
setAP	✓
getFieldName	✓

Table 69: State Transition Matrix

	alive
alive	↑

Table 70: Methods Concurrency Matrix

	ESpecification	getParentClass	getTS	getAP	clone	setAPTS	setAP	getFieldName
ESpecification	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getParentClass	⌘							
getTS	⌘		⌘	⌘		⌘	⌘	
getAP	⌘		⌘	⌘		⌘	⌘	
clone	⌘							
setAPTS	⌘		⌘	⌘		⌘	⌘	
setAP	⌘		⌘	⌘		⌘	⌘	
getFieldName	⌘							

24 EParameter

Table 71: Methods Requires Clause Satisfiability

Method	Satisfiability
EParameter	✓
getRequiresAPTS	✓
setNumber	✓
getNumber	✓
setName	✓
setType	✓
getType	✓
getName	✓
getEnsuresAPTS	✓

Table 72: State Transition Matrix

	alive
alive	↑

Table 73: Methods Concurrency Matrix

	EParameter	getRequiresAPTS	setNumber	getNumber	setName	setType	getType	getName	getEnsuresAPTS
EParameter	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getRequiresAPTS	⌘								
setNumber	⌘		⌘		⌘	⌘		⌘	
getNumber	⌘								
setName	⌘		⌘		⌘	⌘		⌘	
setType	⌘		⌘		⌘	⌘		⌘	
getType	⌘								
getName	⌘		⌘		⌘	⌘		⌘	
getEnsuresAPTS	⌘								

25 EInvariant

Table 74: Methods Requires Clause Satisfiability

Method	Satisfiability
EInvariant	✓
getVariableType	✓
getStateName	✓
setVariableType	✓
setStateIndex	✓
setAP	✓
setVariable	✓
setState	✓
getAP	✓
getVariable	✓
getStateInvariants	✓

Table 75: State Transition Matrix

	alive
alive	↑

Table 76: Methods Concurrency Matrix

	EInvariant	getVariableType	getStateName	setVariableType	setStateIndex	setAP	setVariable	setState	getAP	getVariable	getStateInvariants
EInvariant	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getVariableType	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getStateName	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setVariableType	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setStateIndex	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setAP	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setVariable	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
setState	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getAP	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getVariable	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getStateInvariants	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘

26 EBoolInvariant

Table 77: Methods Requires Clause Satisfiability

Method	Satisfiability
EBoolInvariant	✓
getVariable	✓
getValue	✓

Table 78: State Transition Matrix

	alive
alive	↑

Table 79: Methods Concurrency Matrix

	EBoolInvariant	getVariable	getValue
EBoolInvariant	⌘	⌘	⌘
getVariable	⌘		
getValue	⌘		

27 **EDim**

Table 80: Methods Requires Clause Satisfiability

Method	Satisfiability
EDim	✓
getValues	✓
setName	✓
getName	✓
addValue	✓

Table 81: State Transition Matrix

	alive
alive	↑

Table 82: Methods Concurrency Matrix

	EDim	getValues	setName	getName	addValue
EDim	⌈	⌈	⌈	⌈	⌈
getValues	⌈				
setName	⌈		⌈		⌈
getName	⌈				
addValue	⌈		⌈		⌈

28 `specificationStruct`

Table 83: Methods Requires Clause Satisfiability

Method	Satisfiability
<code>specificationStruct</code>	✓

Table 84: State Transition Matrix

	alive
alive	↑

29 Clause

Table 85: Methods Requires Clause Satisfiability

Method	Satisfiability
Clause	✓

Table 86: State Transition Matrix

	alive
alive	↑

30 **Signature**

Table 87: Methods Requires Clause Satisfiability

Method	Satisfiability
Signature	✓

Table 88: State Transition Matrix

	alive
alive	↑

31 MethodFindVisitor

Table 89: Methods Requires Clause Satisfiability

Method	Satisfiability
MethodFindVisitor	✓
visit	✓

Table 90: State Transition Matrix

	alive
alive	↑

Table 91: Methods Concurrency Matrix

	MethodFindVisitor	visit
MethodFindVisitor	✗	✗
visit	✗	✗

32 GAPHandler

Table 92: Methods Requires Clause Satisfiability

Method	Satisfiability
GAPHandler	✓
addHandlerListener	✓
dispose	✓
execute	✓
extractSettings	✓
isHandled	✓
removeHandlerListener	✓

Table 93: State Transition Matrix

	alive
alive	↑

Table 94: Methods Concurrency Matrix

	GAPHandler	addHandlerListener	dispose	execute	extractSettings	isHandled	removeHandlerListener
GAPHandler	⌘	⌘	⌘	⌘	⌘	⌘	⌘
addHandlerListener	⌘						
dispose	⌘						
execute	⌘			⌘	⌘		
extractSettings	⌘			⌘	⌘		
isHandled	⌘						
removeHandlerListener	⌘						

33 GAPIFileAction

Table 95: Methods Requires Clause Satisfiability

Method	Satisfiability
GAPIFileAction	✓
selectionChanged	✓
setActivePart	✓
run	✓
run	✗

Table 96: State Transition Matrix

	alive
alive	↑

Table 97: Methods Concurrency Matrix

	GAPIFileAction	selectionChanged	setActivePart	run	run
GAPIFileAction	✗	✗	✗	✗	✗
selectionChanged	✗	✗		✗	✗
setActivePart	✗				
run	✗	✗		✗	✗
run	✗	✗		✗	✗

34 Main

Table 98: Methods Requires Clause Satisfiability

Method	Satisfiability
Main	✓
main	✓
testRead	✓
separateJavaFile	✓
anTest	✓

Table 99: State Transition Matrix

	alive
alive	↑

Table 100: Methods Concurrency Matrix

	Main	main	testRead	separateJavaFile	anTest
Main	⌘	⌘	⌘	⌘	⌘
main	⌘	⌘		⌘	
testRead	⌘				
separateJavaFile	⌘	⌘		⌘	
anTest	⌘				

35 **TypestateReturn**

Table 101: Methods Requires Clause Satisfiability

Method	Satisfiability
TypestateReturn	✓

Table 102: State Transition Matrix

	alive
alive	↑

36 **AtApPermissionReturn**

Table 103: Methods Requires Clause Satisfiability

Method	Satisfiability
AtApPermissionReturn	✓

Table 104: State Transition Matrix

	alive
alive	↑

37 AccesspermissionReturn

Table 105: Methods Requires Clause Satisfiability

Method	Satisfiability
AccesspermissionReturn	✓

Table 106: State Transition Matrix

	alive
alive	↑

38 PluralLexer

Table 107: Methods Requires Clause Satisfiability

Method	Satisfiability
PluralLexer	✓
getGrammarFileName	✓
mATFULL	✓
mATPURE	✓
mATIMMUTABLE	✓
mATSHARE	✓
mATUNIQUE	✓
mPUBLICBEHAVIOR	✓
mFULL	✓
mPURE	✓
mIMMUTABLE	✓
mSHARE	✓
mUNIQUE	✓
mNONE	✓
mLSBRACKET	✓
mRSBRACKET	✓
mPERM	✓
mEQUAL	✓
mEQUALOPERATOR	✓
mIN	✓
mTHIS	✓
mRESULT	✓
mPARAM	✓
mREQUIRES	✓
mENSURES	✓
mQUOTE	✓
mAND	✓
mUSE	✓
mUSEFIELDS	✓
mPUNCTUATION	✓
mCASES	✓
mLCBRACKET	✓
mRCBRACKET	✓
mCLASSSTATES	✓
mREFINE	✓
mVALUE	✓
mSTATE	✓
mSTATES	✓
mDIM	✓
mNAME	✓
mINV	✓
mOPERATOR	✓
mSEMICOLON	✓
mLESS	✓
mLESSTHANEQUAL	✓
mGREATER	✓
mGREATERTHANEQUAL	✓

mANDD	✓
mOR	✓
mJMLSTART	✓
mJMLEND	✓
mPLUSMINUSOPERATOR	✓
mASSIGNABLE	✓
mNOTHING	✓
mEVERYTHING	✓
mGHOST	✓
mINT	✓
mINVARIANT	✓
mOLD	✓
mID	✓
mNUMBERS	✓
mWS	✓
mTokens	✓

Table 108: State Transition Matrix

	alive
alive	↑

	PluralLexer	getGrammarFileName	mATFULL	mATPURE	mATIMMUTABLE	mATSHARE	mATUNIQUE	mPUBLICBEHAVIOR	mFULL	mPURE	mIMMUTABLE	mSHARE	mUNIQUE	mNONE	mLSBRACKET	mRSBRACKET	mPERM	mEQUAL	mEQUALOPERATOR	mIN	mTHIS
PluralLexer	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
getGrammarFileName	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
mATFULL	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
mATPURE	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
mATIMMUTABLE	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
mATSHARE	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
mATUNIQUE	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
mPUBLICBEHAVIOR	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
mFULL	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
mPURE	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
mIMMUTABLE	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
mSHARE	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
mUNIQUE	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
mNONE	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
mLSBRACKET	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘

39 **DFA7**

Table 110: Methods Requires Clause Satisfiability

Method	Satisfiability
DFA7	✓
getDescription	✓

Table 111: State Transition Matrix

	alive
alive	↑

Table 112: Methods Concurrency Matrix

	DFA7	getDescription
DFA7	✗	✗
getDescription	✗	✓

40 EAPTypeState

Table 113: Methods Requires Clause Satisfiability

Method	Satisfiability
EAPTypeState	✓
setAP	✓
getAP	✓
setTS	✓
getTS	✓

Table 114: State Transition Matrix

	alive
alive	↑

Table 115: Methods Concurrency Matrix

	EAPTypeState	setAP	getAP	setTS	getTS
EAPTypeState	⌘	⌘	⌘	⌘	⌘
setAP	⌘	⌘	⌘	⌘	⌘
getAP	⌘	⌘	⌘	⌘	⌘
setTS	⌘	⌘	⌘	⌘	⌘
getTS	⌘	⌘	⌘	⌘	⌘

41 Abbreviation

Table 116: 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

42 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 Activator {
6   @Perm(ensures="unique(this) in alive")
7   Activator() { }
8
9   @Perm(requires="full(this) in alive",
10  ensures="full(this) in alive")
11   public void start(BundleContext context) {
12   }
13   @Perm(requires="unique(this) in alive",
14  ensures="unique(this) in alive")
15   public void stop(BundleContext context) {
16   }
17   @Perm(requires="pure(this) in alive",
18  ensures="pure(this) in alive")
19   Activator getDefault() {
20     return null;
21   }
22   @Perm(requires="full(this) in alive",
23  ensures="full(this) in alive")
24   public void start(BundleContext context) {
25   }
26   @Perm(requires="unique(this) in alive",
27  ensures="unique(this) in alive")
28   public void stop(BundleContext context) {
29   }
30   @Perm(requires="pure(this) in alive",
31  ensures="pure(this) in alive")
32   Activator getDefault() {
33     return null;
34   }
35
36   ImageDescriptor getImageDescriptor(String path) {
37     return null;
38   }
39
40 }ENDOFCLASS
41
42 @ClassStates({@State(name = "alive")})
43
44 class SampleAction {
45   @Perm(ensures="unique(this) in alive")
46   SampleAction() { }
47
48   @Perm(requires="full(this) in alive",
49  ensures="full(this) in alive")
50   public void run(IAction action) {
51   }
52
53   public void selectionChanged(IAction action, ISelection selection) {
54   }
55
56   public void dispose() {
57   }
58   @Perm(requires="full(this) in alive",
59  ensures="full(this) in alive")
60   public void init(IWorkbenchWindow window) {
61   }
62
63 }ENDOFCLASS
64
65 @ClassStates({@State(name = "alive")})
66
67 class JMLAnnotatedJavaClass {
68   @Perm(ensures="unique(this) in alive")
69   JMLAnnotatedJavaClass() { }
70
71   @Perm(requires="unique(this) in alive",
72  ensures="unique(this) in alive")
73   public String translateJMLAnnotationsToPlural(String JProgram) {
74     return null;
75   }
76 }
```

```

76 @Perm(requires="full(this) in alive",
77 ensures="full(this) in alive")
78 private String translateClassSpecifications(String JProgram) {
79     return null;
80 }
81 @Perm(requires="full(this) in alive",
82 ensures="full(this) in alive")
83 private void parseAndStoreJMLAnnotation(String JMLAnnotation) {
84 }
85 @Perm(requires="unique(this) in alive",
86 ensures="unique(this) in alive")
87 private String translateMethodSpecification(String JProgram) {
88     return null;
89 }

91 public String readFileAsString(String filePath) {
92     return null;
93 }

95 }ENDOFCLASS

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

99 class PluralParser {
100 @Perm(ensures="unique(this) in alive")
101 PluralParser() { }

103 @Perm(requires="full(this) in alive",
104 ensures="full(this) in alive")
105 void jmlSpecifications() {
106 }
107 @Perm(requires="full(this) in alive",
108 ensures="full(this) in alive")
109 void jmlClassSpecifications() {
110 }
111 @Perm(requires="full(this) in alive",
112 ensures="full(this) in alive")
113 void jmlGhostDeclaration() {
114 }
115 @Perm(requires="full(this) in alive",
116 ensures="full(this) in alive")
117 void jmlMethodSpecification() {
118 }
119 @Perm(requires="full(this) in alive",
120 ensures="full(this) in alive")
121 void jmlRequires() {
122 }
123 @Perm(requires="full(this) in alive",
124 ensures="full(this) in alive")
125 void jmlReq() {
126 }
127 @Perm(requires="full(this) in alive",
128 ensures="full(this) in alive")
129 void jmlLessThanEqualReq() {
130 }
131 @Perm(requires="full(this) in alive",
132 ensures="full(this) in alive")
133 void jmlEnsures() {
134 }
135 @Perm(requires="full(this) in alive",
136 ensures="full(this) in alive")
137 void jmlEns() {
138 }
139 @Perm(ensures="none(this) in alive")
140 public String[] getTokenNames() {
141     return null;
142 }

144 public String getGrammarFileName() {
145     return null;
146 }
147 @Perm(requires="full(this) in alive",
148 ensures="full(this) in alive")
149 void jmlOldEns() {
150 }
151 @Perm(requires="full(this) in alive",
152 ensures="full(this) in alive")
153 void specifications() {
154 }
155 @Perm(requires="full(this) in alive",
156 ensures="full(this) in alive")

```



```

157 void perm() {
158 }
159 @Perm(requires="full(this) in alive",
160 ensures="full(this) in alive")
161 void requiresensuresClause() {
162 }
163 @Perm(requires="full(this) in alive",
164 ensures="full(this) in alive")
165 void requiresClause() {
166 }
167 @Perm(requires="full(this) in alive",
168 ensures="full(this) in alive")
169 void reaccesspermissionTypestates() {
170 }
171 @Perm(requires="pure(this) in alive",
172 ensures="pure(this) in alive")
173 AccesspermissionReturn accesspermission() {
174 return null;
175 }
176 @Perm(requires="full(this) in alive",
177 ensures="full(this) in alive")
178 TypestateReturn typestate() {
179 return null;
180 }
181 @Perm(requires="full(this) in alive",
182 ensures="full(this) in alive")
183 void attype() {
184 }
185 @Perm(requires="pure(this) in alive",
186 ensures="pure(this) in alive")
187 AtApPermissionReturn atappermission() {
188 return null;
189 }
190 @Perm(requires="full(this) in alive",
191 ensures="full(this) in alive")
192 void usevalue() {
193 }
194 @Perm(requires="full(this) in alive",
195 ensures="full(this) in alive")
196 void classstates() {
197 }
198 @Perm(requires="full(this) in alive",
199 ensures="full(this) in alive")
200 void startClassstates() {
201 }
202 @Perm(requires="full(this) in alive",
203 ensures="full(this) in alive")
204 void state() {
205 }
206 @Perm(requires="full(this) in alive",
207 ensures="full(this) in alive")
208 void endclassstates() {
209 }
210 @Perm(requires="full(this) in alive",
211 ensures="full(this) in alive")
212 void refine() {
213 }
214 @Perm(requires="full(this) in alive",
215 ensures="full(this) in alive")
216 void states() {
217 }
218 @Perm(requires="full(this) in alive",
219 ensures="full(this) in alive")
220 void dimension() {
221 }
222 @Perm(requires="full(this) in alive",
223 ensures="full(this) in alive")
224 void item() {
225 }
226 @Perm(requires="full(this) in alive",
227 ensures="full(this) in alive")
228 void invariant() {
229 }
230 @Perm(requires="full(this) in alive",
231 ensures="full(this) in alive")
232 void condition() {
233 }
234 @Perm(requires="full(this) in alive",
235 ensures="full(this) in alive")
236 void other() {
237 }

```

```

239 }ENDOFCLASS
241 @ClassStates({@State(name = "alive")})
243 class EJmlSpecification {
244   @Perm(ensures="unique(this) in alive")
245   EJmlSpecification() { }
247   @Perm(requires="full(this) in alive",
248     ensures="full(this) in alive")
249   void setDimensionName(String str) {
250   }
251   @Perm(requires="full(this) in alive",
252     ensures="full(this) in alive")
253   void setEnsures(String str) {
254   }
255   @Perm(requires="full(this) in alive",
256     ensures="full(this) in alive")
257   void setPerm(String str) {
258   }
259   @Perm(requires="full(this) in alive",
260     ensures="full(this) in alive")
261   void addRequires(String str) {
262   }
263   @Perm(requires="unique(this) in alive",
264     ensures="unique(this) in alive")
265   void reset() {
266   }
267   @Perm(requires="full(this) in alive",
268     ensures="full(this) in alive")
269   String JmlClassSpec2PluralClassSpec() {
270     return null;
271   }
272   @Perm(requires="full(this) in alive",
273     ensures="full(this) in alive")
274   String noRequires() {
275     return null;
276   }
277   @Perm(requires="pure(this) in alive",
278     ensures="pure(this) in alive")
279   String getPerm() {
280     return null;
281   }
282   @Perm(requires="full(this) in alive",
283     ensures="full(this) in alive")
284   String determineEnsures(String req) {
285     return null;
286   }
287   @Perm(requires="full(this) in alive",
288     ensures="full(this) in alive")
289   String oneRequires() {
290     return null;
291   }
292   @Perm(requires="full(this) in alive",
293     ensures="full(this) in alive")
294   String moreRequires() {
295     return null;
296   }
298 }ENDOFCLASS
300 @ClassStates({@State(name = "alive")})
302 class EGhost {
303   @Perm(ensures="unique(this) in alive")
304   EGhost() { }
306   @Perm(requires="full(this) in alive",
307     ensures="full(this) in alive")
308   public void setDimensionName(String str) {
309   }
310   @Perm(requires="full(this) in alive",
311     ensures="full(this) in alive")
312   public void setDimensionValues(int low, int high) {
313   }
314   @Perm(requires="pure(this) in alive",
315     ensures="pure(this) in alive")
316   public String getDimensionName() {
317     return null;
318   }

```

```

319 @Perm(requires="pure(this) in alive",
320 ensures="pure(this) in alive")
321 public int getLowValueofInv() {
322     return 0;
323 }
324 @Perm(requires="pure(this) in alive",
325 ensures="pure(this) in alive")
326 public int getHighValueofInv() {
327     return 0;
328 }
329
330 }ENDOFCLASS
331
332 @ClassStates({@State(name = "alive")})
333
334 class Time {
335 @Perm(ensures="unique(this) in alive")
336 Time() { }
337
338 @Perm(requires="full(this) in alive",
339 ensures="full(this) in alive")
340 public String toString() {
341     return null;
342 }
343
344 }ENDOFCLASS
345
346 @ClassStates({@State(name = "alive")})
347
348 class FileReader {
349 @Perm(ensures="unique(this) in alive")
350 FileReader() { }
351
352 String readFile(String pathname) {
353     return null;
354 }
355
356 }ENDOFCLASS
357
358 @ClassStates({@State(name = "alive")})
359
360 class UserSelectedClassesAnalysis {
361 @Perm(ensures="unique(this) in alive")
362 UserSelectedClassesAnalysis() { }
363
364 private CompilationUnit getCompilationUnit(String prog) {
365     return null;
366 }
367
368 @Perm(requires="unique(this) in alive",
369 ensures="unique(this) in alive")
370 void analyzeFromCommandLine(LinkedList<String> inputFiles, String strType, String strK) {
371 }
372
373 @Perm(requires="unique(this) in alive",
374 ensures="unique(this) in alive")
375 public void analyzeFromPlugin(List<ICompilationUnit> compilationUnitList, int test) {
376 }
377
378 @Perm(requires="unique(this) in alive",
379 ensures="unique(this) in alive")
380 void callModelCheckerThroughCommandLine() {
381 }
382
383 @Perm(requires="full(this) in alive",
384 ensures="full(this) in alive")
385 Time getTime() {
386     return null;
387 }
388
389 @Perm(requires="unique(this) in alive",
390 ensures="unique(this) in alive")
391 void CreatePdfSummary_CommandLine(Time starttime2, Time endtime2) {
392 }
393
394 @Perm(requires="full(this) in alive",
395 ensures="full(this) in alive")
396 void callModelCheckerThroughPlugin() {
397 }
398
399 void createPdfSummaryPlugin(Time starttime2, Time endtime2) {
400 }
401
402 void makePdfCommandLine() {
403 }

```

```

401 void makePdfPlugin() {
402 }
403 @Perm(requires="full(this) in alive",
404 ensures="full(this) in alive")
405 void printMethodMetrics() {
406 }
407
408 }ENDOFCLASS
409
410 @ClassStates({@State(name = "alive")})
411
412 class EVMDDSMCGenerator {
413 @Perm(ensures="unique(this) in alive")
414 EVMDDSMCGenerator() { }
415
416 @Perm(requires="unique(this) in alive",
417 ensures="unique(this) in alive")
418 void reset() {
419 }
420 @Perm(requires="pure(this) in alive",
421 ensures="pure(this) in alive")
422 void addBoolStateInvariant(String variable, String operator, String value) {
423 }
424 @Perm(requires="pure(this) in alive",
425 ensures="pure(this) in alive")
426 EPackage getPkgObject() {
427 return null;
428 }
429 @Perm(requires="full(this) in alive",
430 ensures="full(this) in alive")
431 String modifyConstructorSpecifications(String prog) {
432 return null;
433 }
434 @Perm(requires="pure(this) in alive",
435 ensures="pure(this) in alive")
436 void addState(String stateName) {
437 }
438 @Perm(requires="pure(this) in alive",
439 ensures="pure(this) in alive")
440 void addRequiresParam_APTS(String ap, String ts, String argumentNumber) {
441 }
442 @Perm(requires="pure(this) in alive",
443 ensures="pure(this) in alive")
444 void addEnsuresParam_APTS(String ap, String ts, String argumentNumber) {
445 }
446 @Perm(requires="pure(this) in alive",
447 ensures="pure(this) in alive")
448 void addStateInvariant(String accessPermission, String variable, String state) {
449 }
450 @Perm(requires="pure(this) in alive",
451 ensures="pure(this) in alive")
452 void addDimensionValue(String value) {
453 }
454 @Perm(requires="pure(this) in alive",
455 ensures="pure(this) in alive")
456 void addEnsuresAPTS(String ap, String ts) {
457 }
458
459 }ENDOFCLASS
460
461 @ClassStates({@State(name = "alive")})
462
463 class EPackage {
464 @Perm(ensures="unique(this) in alive")
465 EPackage() { }
466
467 @Perm(requires="pure(this) in alive",
468 ensures="pure(this) in alive")
469 public LinkedList<EClass> getClasses() {
470 return null;
471 }
472 @Perm(requires="pure(this) in alive",
473 ensures="pure(this) in alive")
474 public int getTotalReachableStates() {
475 return 0;
476 }
477 @Perm(requires="full(this) in alive",
478 ensures="full(this) in alive")
479 public String getName() {
480 return null;

```

```

481 }
482 @Perm(requires="full(this) in alive",
483 ensures="full(this) in alive")
484 public int getTotalStates() {
485     return 0;
486 }
487 @Perm(requires="pure(this) in alive",
488 ensures="pure(this) in alive")
489 public String getSinkStates() {
490     return null;
491 }
492
493 }ENDOFCLASS
494
495 @ClassStates({@State(name = "alive")})
496
497 class EGeneratedPluralSpecification {
498     @Perm(ensures="unique(this) in alive")
499     EGeneratedPluralSpecification() { }
500
501     @Perm(requires="unique(this) in alive",
502     ensures="unique(this) in alive")
503     void createFromCommandLine(String prog, String className) {
504     }
505     @Perm(requires="unique(this) in alive",
506     ensures="unique(this) in alive")
507     void createFromPlugin(String prog, String className) {
508     }
509
510 }ENDOFCLASS
511
512 @ClassStates({@State(name = "alive")})
513
514 class ESMCModel {
515     @Perm(ensures="unique(this) in alive")
516     ESMCModel() { }
517
518     @Perm(requires="full(this) in alive",
519     ensures="full(this) in alive")
520     void setK(int k) {
521     }
522     @Perm(requires="full(this) in alive",
523     ensures="full(this) in alive")
524     void declarationsAndinitializations() {
525     }
526     @Perm(requires="full(this) in alive",
527     ensures="full(this) in alive")
528     void initialize(LinkedList<EClass> _listClasses) {
529     }
530     @Perm(requires="pure(this) in alive",
531     ensures="pure(this) in alive")
532     String comment(String str) {
533         return null;
534     }
535     @Perm(requires="full(this) in alive",
536     ensures="full(this) in alive")
537     void modelAlias(String className, Integer objectIndex, Integer refIndex) {
538     }
539     @Perm(requires="full(this) in alive",
540     ensures="full(this) in alive")
541     boolean isClassExist(String className) {
542         return 0;
543     }
544     @Perm(requires="full(this) in alive",
545     ensures="full(this) in alive")
546     void Transitions() {
547     }
548     @Perm(requires="full(this) in alive",
549     ensures="full(this) in alive")
550     void createInstanceInModel(EClass _class, String name, int objectIndex, int J) {
551     }
552     @Perm(requires="full(this) in alive",
553     ensures="full(this) in alive")
554     void modelPrimePCandMethod(EClass _class, EMethod _method, Integer objectIndex, Integer refIndex) {
555     }
556     @Perm(requires="full(this) in alive",
557     ensures="full(this) in alive")
558     void startMethod(EClass _class, EMethod _method, Integer objectIndex, Integer refIndex) {
559     }
560     @Perm(requires="full(this) in alive",
561     ensures="full(this) in alive")

```

```

562 void modelPCCConstructor(EClass _class, Integer objectIndex, Integer refIndex, EClass _currentClass) {
563 }
564 @Perm(requires="full(this) in alive",
565 ensures="full(this) in alive")
566 void modelAPs(EClass _class, Integer objectIndex, Integer refIndex) {
567 }
568 @Perm(requires="full(this) in alive",
569 ensures="full(this) in alive")
570 void modelPCMethod(EClass _class, Integer objectIndex, Integer refIndex) {
571 }
572 @Perm(requires="full(this) in alive",
573 ensures="full(this) in alive")
574 void startAPTSPARAM(EMethod _method, Integer J) {
575 }
576 @Perm(requires="full(this) in alive",
577 ensures="full(this) in alive")
578 void startPrimeTSPARAM(EMethod method, Integer refIndex) {
579 }
580 @Perm(requires="full(this) in alive",
581 ensures="full(this) in alive")
582 void modelPrimeConstructor(EClass _class, Integer objectIndex, Integer refIndex) {
583 }
584 @Perm(requires="full(this) in alive",
585 ensures="full(this) in alive")
586 void modelInheritance(EClass _class, Integer objectIndex, Integer refIndex) {
587 }
588 @Perm(requires="full(this) in alive",
589 ensures="full(this) in alive")
590 void modelPrimeAPStateInvariants(EClass _class, Integer refIndex, String stateName) {
591 }
592 @Perm(requires="full(this) in alive",
593 ensures="full(this) in alive")
594 void starPrimeAP(String ap, EClass _class, Integer objectIndex, Integer refIndex, String stateName) {
595 }
596 @Perm(requires="full(this) in alive",
597 ensures="full(this) in alive")
598 void modelPrimeAP(String ap, String className, Integer objectIndex, Integer refIndex) {
599 }
600 @Perm(requires="pure(this) in alive",
601 ensures="pure(this) in alive")
602 int getAPId(String ap) {
603     return 0;
604 }
605 @Perm(requires="full(this) in alive",
606 ensures="full(this) in alive")
607 void modelEndPCMethod(EClass _class, EMethod _method, Integer objectIndex, Integer refIndex) {
608 }
609 @Perm(requires="full(this) in alive",
610 ensures="full(this) in alive")
611 void endMethod(EClass _class, EMethod _method, Integer objectIndex, Integer refIndex) {
612 }
613 @Perm(requires="full(this) in alive",
614 ensures="full(this) in alive")
615 void modelEndPCCConstructor(EClass _class, EMethod _method, Integer objectIndex, Integer refIndex,
    EClass _currentClass) {
616 }
617 @Perm(requires="full(this) in alive",
618 ensures="full(this) in alive")
619 void modelendConstructor(EClass _class, EMethod _method, Integer refIndex) {
620 }
621 @Perm(requires="full(this) in alive",
622 ensures="full(this) in alive")
623 void updateBoolStateInvariants(EClass _class, String methodName, String stateName, Integer objectIndex)
    {
624 }
625 @Perm(requires="full(this) in alive",
626 ensures="full(this) in alive")
627 void endPrimeAPTS(EClass _class, String methodName, String ap, String stateName, Integer objectIndex,
    Integer refIndex) {
628 }
629 @Perm(requires="full(this) in alive",
630 ensures="full(this) in alive")
631 void endPrimeAPTSPARAM(EMethod method, Integer refIndex) {
632 }
633 @Perm(requires="full(this) in alive",
634 ensures="full(this) in alive")
635 void updateState(String methodName, String state, EClass _class, Integer objectIndex) {
636 }
637 @Perm(requires="full(this) in alive",
638 ensures="full(this) in alive")
639 void updateStateInvariants(EClass _class, String methodName, String stateName, Integer refIndex) {

```

```

640 }
641 @Perm(requires="full(this) in alive",
642 ensures="full(this) in alive")
643 void modelBoolStateInvariants(EClass _class, Integer objectIndex, String stateName) {
644 }
645 @Perm(requires="full(this) in alive",
646 ensures="full(this) in alive")
647 void startAPTS(EClass _class, EMethod _method, String ap, String stateName, Integer objectIndex,
648 Integer refIndex) {
649 }
650 @Perm(requires="full(this) in alive",
651 ensures="full(this) in alive")
652 void modelState(EClass _class, Integer objectIndex, EMethod _method, String stateName) {
653 }
654 @Perm(requires="full(this) in alive",
655 ensures="full(this) in alive")
656 void modelStateInvariants(EClass _class, int refIndex, EMethod _method, String stateName) {
657 }
658 @Perm(requires="full(this) in alive",
659 ensures="full(this) in alive")
660 void methodsReachability(EClass _class, Integer objectIndex, Integer refIndex) {
661 }
662 @Perm(requires="full(this) in alive",
663 ensures="full(this) in alive")
664 void modelAP(EClass _class, Integer objectIndex, Integer refIndex, String ap) {
665 }
666 @Perm(requires="pure(this) in alive",
667 ensures="pure(this) in alive")
668 int getObjectIndex(EClass _class, String variable) {
669 return 0;
670 }
671 @Perm(requires="full(this) in alive",
672 ensures="full(this) in alive")
673 void updateTokens(String ap, String className, Integer objectIndex, Integer refIndex) {
674 }
675 @Perm(requires="full(this) in alive",
676 ensures="full(this) in alive")
677 void modelPrimePCConstructor(EClass _class, EMethod _method, Integer objectIndex, Integer refIndex,
678 EClass _currentClass) {
679 }
680 @Perm(requires="full(this) in alive",
681 ensures="full(this) in alive")
682 void modelPrimePC(EClass _class, Integer objectIndex, Integer refIndex) {
683 }
684 @Perm(requires="full(this) in alive",
685 ensures="full(this) in alive")
686 void defineKVariables(String className, int objectIndex, EClass _class, int K) {
687 }
688 @Perm(requires="full(this) in alive",
689 ensures="full(this) in alive")
690 void defineVariables(String className, int objectIndex, EClass _class, int modifier) {
691 }
692 @Perm(requires="full(this) in alive",
693 ensures="full(this) in alive")
694 void initilizeKVariables(String className, int objectIndex, int K) {
695 }
696 @Perm(requires="full(this) in alive",
697 ensures="full(this) in alive")
698 void initilizeVariables(String className, int objectIndex, EClass _class, int modifier) {
699 }
700 @Perm(requires="full(this) in alive",
701 ensures="full(this) in alive")
702 void createAlias() {
703 }
704 @Perm(requires="pure(this) in alive",
705 ensures="pure(this) in alive")
706 boolean isPrivateAndIndexEqualToZero(int refIndex, EField _field) {
707 return 0;
708 }
709 @Perm(requires="full(this) in alive",
710 ensures="full(this) in alive")
711 EClass getClass(String className) {
712 return null;
713 }
714 @Perm(requires="pure(this) in alive",
715 ensures="pure(this) in alive")
716 int getDimensionIndex(EClass _class, String ts) {
717 return 0;
718 }
719 @Perm(requires="full(this) in alive",
720 ensures="full(this) in alive")

```

```

719 void addIndexes() {
720 }
721 @Perm(requires="pure(this) in alive",
722 ensures="pure(this) in alive")
723 void createDimensionsObject(EClass _class) {
724 }
725 @Perm(requires="full(this) in alive",
726 ensures="full(this) in alive")
727 void createDimensionAsField(EClass _class, EDim _dim, int count) {
728 }
729 @Perm(requires="full(this) in alive",
730 ensures="full(this) in alive")
731 void createParentAsField(EClass _class, EClass _currentClass) {
732 }
733 @Perm(requires="full(this) in alive",
734 ensures="full(this) in alive")
735 void createParentObject(EClass _class) {
736 }
737 @Perm(requires="full(this) in alive",
738 ensures="full(this) in alive")
739 void setInvariantVariableType(EClass _class, EInvariant inv) {
740 }
741 @Perm(requires="full(this) in alive",
742 ensures="full(this) in alive")
743 void addInvariantStateIndex(EClass _class) {
744 }
745 @Perm(requires="full(this) in alive",
746 ensures="full(this) in alive")
747 void Spec() {
748 }
749 @Perm(requires="full(this) in alive",
750 ensures="full(this) in alive")
751 void concurrentMethods(EClass _class, Integer objectIndex, Integer refIndex) {
752 }
753 @Perm(requires="full(this) in alive",
754 ensures="full(this) in alive")
755 void statesAdjacencyMatrix(EClass _class, Integer objectIndex) {
756 }
757 @Perm(requires="full(this) in alive",
758 ensures="full(this) in alive")
759 void generateSMCmodelPlugin(EPackage _pkg, int testType) {
760 }
761 }ENDOFCLASS
762
763 @ClassStates({@State(name = "alive")})
764
765 class WorkspaceUtilities {
766 @Perm(ensures="unique(this) in alive")
767 WorkspaceUtilities() { }
768
769
770
771 ASTNode getASTNodeFromCompilationUnit(ICompilationUnit compUnit) {
772 return null;
773 }
774 @Perm(requires="full(this) in alive",
775 ensures="full(this) in alive")
776 List<ICompilationUnit> scanForCompilationUnits() {
777 return null;
778 }
779 @Perm(requires="full(this) in alive",
780 ensures="full(this) in alive")
781 List<ICompilationUnit> collectCompilationUnits(IJavaElement javaElement) {
782 return null;
783 }
784 @Perm(requires="full(this) in alive",
785 ensures="full(this) in alive")
786 List<ICompilationUnit> findCompilationUnits(List<String> files) {
787 return null;
788 }
789
790 String getWorkspaceRelativeName(IJavaElement element) {
791 return null;
792 }
793
794 Map<ICompilationUnit,ASTNode> parseCompilationUnits(List<ICompilationUnit> compilationUnits) {
795 return null;
796 }
797
798 List<MethodDeclaration> scanForMethodDeclarations(Map<ICompilationUnit,ASTNode>
compilationUnitToASTNode) {

```



```

799     return null;
800 }

802 List<MethodDeclaration> scanForMethodDeclarationsFromAST(ASTNode node) {
803     return null;
804 }

806 }ENDOFCLASS

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

810 class EClass {
811     @Perm(ensures="unique(this) in alive")
812     EClass() { }

814     @Perm(requires="full(this) in alive",
815     ensures="full(this) in alive")
816     public String getName() {
817         return null;
818     }
819     @Perm(requires="pure(this) in alive",
820     ensures="pure(this) in alive")
821     public LinkedList<String> getTransitions() {
822         return null;
823     }
824     @Perm(requires="pure(this) in alive",
825     ensures="pure(this) in alive")
826     public LinkedList<EState> getReachableStates() {
827         return null;
828     }
829     @Perm(requires="pure(this) in alive",
830     ensures="pure(this) in alive")
831     public LinkedList<EState> getStates() {
832         return null;
833     }
834     @Perm(requires="pure(this) in alive",
835     ensures="pure(this) in alive")
836     public LinkedList<EMethod> getMethods() {
837         return null;
838     }
839     @Perm(requires="pure(this) in alive",
840     ensures="pure(this) in alive")
841     public String getSuperClassName() {
842         return null;
843     }
844     @Perm(requires="pure(this) in alive",
845     ensures="pure(this) in alive")
846     public int findStateIndex(String st) {
847         return 0;
848     }
849     @Perm(requires="pure(this) in alive",
850     ensures="pure(this) in alive")
851     public int getIndex() {
852         return 0;
853     }
854     @Perm(requires="pure(this) in alive",
855     ensures="pure(this) in alive")
856     public LinkedList<EField> getFields() {
857         return null;
858     }
859     @Perm(requires="pure(this) in alive",
860     ensures="pure(this) in alive")
861     public ArrayList<String> getVariablesOfBooleanInvariants() {
862         return null;
863     }
864     @Perm(requires="pure(this) in alive",
865     ensures="pure(this) in alive")
866     public LinkedList<EDim> getDimensions() {
867         return null;
868     }
869     @Perm(requires="full(this) in alive",
870     ensures="full(this) in alive")
871     public void createObject() {
872     }
873     @Perm(requires="full(this) in alive",
874     ensures="full(this) in alive")
875     public EMethod getConstructor() {
876         return null;
877     }
878     @Perm(requires="full(this) in alive",
879     ensures="full(this) in alive")

```

```

880 public void setSuperClassName(String str) {
881 }
882 @Perm(requires="full(this) in alive",
883 ensures="full(this) in alive")
884 public int getTotalStates() {
885     return 0;
886 }
887 @Perm(requires="pure(this) in alive",
888 ensures="pure(this) in alive")
889 public int getTotalReachableStates() {
890     return 0;
891 }
892 @Perm(requires="full(this) in alive",
893 ensures="full(this) in alive")
894 public void addMethod(EMethod method) {
895 }
896 @Perm(requires="full(this) in alive",
897 ensures="full(this) in alive")
898 public void addField(EField field) {
899 }
900 @Perm(requires="full(this) in alive",
901 ensures="full(this) in alive")
902 public void addState(EState state) {
903 }
904 @Perm(requires="pure(this) in alive",
905 ensures="pure(this) in alive")
906 public int getLastObjectIndex() {
907     return 0;
908 }
909 @Perm(requires="full(this) in alive",
910 ensures="full(this) in alive")
911 public void addDimension(EDim dim) {
912 }
913 @Perm(requires="full(this) in alive",
914 ensures="full(this) in alive")
915 public boolean hasMoreThanOneDimension() {
916     return 0;
917 }
918 @Perm(requires="pure(this) in alive",
919 ensures="pure(this) in alive")
920 public void addClassStatesSpecifications(String annotation) {
921 }
922 }ENDOFCLASS
923
924 @ClassStates({@State(name = "alive")})
925
926 class EGraphWriter {
927 @Perm(ensures="unique(this) in alive")
928 EGraphWriter() { }
929
930 @Perm(requires="full(this) in alive",
931 ensures="full(this) in alive")
932 void addTrnsitions(String str) {
933 }
934
935 @Perm(requires="full(this) in alive",
936 ensures="full(this) in alive")
937 void createGraph() {
938 }
939 @Perm(requires="full(this) in alive",
940 ensures="full(this) in alive")
941 void setNumberOfUnReachableMethods() {
942 }
943 @Perm(requires="pure(this) in alive",
944 ensures="pure(this) in alive")
945 int getNumberOfUnReachableMethods() {
946     return 0;
947 }
948 }ENDOFCLASS
949
950 @ClassStates({@State(name = "alive")})
951
952 class EState {
953 @Perm(ensures="unique(this) in alive")
954 EState() { }
955
956 @Perm(requires="pure(this) in alive",
957 ensures="pure(this) in alive")
958 public int isReachable() {
959     return 0;
960 }

```

```

961 }
962 @Perm(requires="pure(this) in alive",
963 ensures="pure(this) in alive")
964 public String getName() {
965     return null;
966 }
967 @Perm(requires="full(this) in alive",
968 ensures="full(this) in alive")
969 public void setReachability(int value) {
970 }
971 @Perm(requires="pure(this) in alive",
972 ensures="pure(this) in alive")
973 public LinkedList<EInvariant> getInvariants() {
974     return null;
975 }
976 @Perm(requires="pure(this) in alive",
977 ensures="pure(this) in alive")
978 public int getStateIndex() {
979     return 0;
980 }
981 @Perm(requires="pure(this) in alive",
982 ensures="pure(this) in alive")
983 public LinkedList<EBoolInvariant> getBoolInvariants() {
984     return null;
985 }
986 @Perm(requires="full(this) in alive",
987 ensures="full(this) in alive")
988 public void addBoolInvariant(EBoolInvariant inv) {
989 }
990 @Perm(requires="full(this) in alive",
991 ensures="full(this) in alive")
992 public void setIndex(int stateIndex) {
993 }
994 @Perm(requires="full(this) in alive",
995 ensures="full(this) in alive")
996 public Boolean isReachableState(String str) {
997     return null;
998 }
999
1000 }ENDOFCLASS
1001
1002 @ClassStates({@State(name = "alive")})
1003
1004 class EOutputLatex {
1005     @Perm(ensures="unique(this) in alive")
1006     EOutputLatex() { }
1007
1008     @Perm(requires="unique(this) in alive",
1009     ensures="unique(this) in alive")
1010     void create_CommandLine() {
1011     }
1012     @Perm(requires="full(this) in alive",
1013     ensures="full(this) in alive")
1014     void addUsePackages() {
1015     }
1016     @Perm(requires="full(this) in alive",
1017     ensures="full(this) in alive")
1018     void writeToLatex() {
1019     }
1020     @Perm(requires="full(this) in alive",
1021     ensures="full(this) in alive")
1022     void WriteSummary() {
1023     }
1024     @Perm(requires="full(this) in alive",
1025     ensures="full(this) in alive")
1026     void addSummaryTableHeaders() {
1027     }
1028     @Perm(requires="full(this) in alive",
1029     ensures="full(this) in alive")
1030     void addSummaryTableRows() {
1031     }
1032     @Perm(requires="full(this) in alive",
1033     ensures="full(this) in alive")
1034     void writeStateTransitionMatrix(EClass _class) {
1035     }
1036     @Perm(requires="full(this) in alive",
1037     ensures="full(this) in alive")
1038     void addSTMColumnsHeaders(EClass _class) {
1039     }
1040     @Perm(requires="full(this) in alive",
1041     ensures="full(this) in alive")

```

```

1042 void writeMethodConcurrencyMatrix(EClass _class) {
1043 }
1044 @Perm(requires="full(this) in alive",
1045 ensures="full(this) in alive")
1046 void addConcurrencyMatrixHeaders(EClass _class) {
1047 }
1048 @Perm(requires="full(this) in alive",
1049 ensures="full(this) in alive")
1050 void writeAbbervations() {
1051 }
1052 @Perm(requires="full(this) in alive",
1053 ensures="full(this) in alive")
1054 void setText(String str) {
1055 }
1056 @Perm(requires="full(this) in alive",
1057 ensures="full(this) in alive")
1058 void parseTransitions(String str) {
1059 }
1060 @Perm(requires="full(this) in alive",
1061 ensures="full(this) in alive")
1062 void parseSinkStates(String str) {
1063 }
1064 @Perm(requires="unique(this) in alive",
1065 ensures="unique(this) in alive")
1066 void create_Plugin() {
1067 }
1068 @Perm(requires="full(this) in alive",
1069 ensures="full(this) in alive")
1070 void addConcurrencyMatrixRows(EClass _class) {
1071 }
1072 @Perm(requires="full(this) in alive",
1073 ensures="full(this) in alive")
1074 String getConcurrencyValue(EMethod _method, EMethod __method) {
1075 return null;
1076 }
1077 @Perm(requires="full(this) in alive",
1078 ensures="full(this) in alive")
1079 void addSummaryTableColumns() {
1080 }
1081 @Perm(requires="full(this) in alive",
1082 ensures="full(this) in alive")
1083 void addSTMNumberOfColumns(EClass _class) {
1084 }
1085 @Perm(requires="pure(this) in alive",
1086 ensures="pure(this) in alive")
1087 String getStateReachabilityValue(EState _state, EState __state) {
1088 return null;
1089 }
1090 @Perm(requires="full(this) in alive",
1091 ensures="full(this) in alive")
1092 EMethod getMethod(String className, String methodName) {
1093 return null;
1094 }
1095
1096 }ENDOFCLASS
1097
1098 @ClassStates({@State(name = "alive")})
1099
1100 class EMethod {
1101 @Perm(ensures="unique(this) in alive")
1102 EMethod() { }
1103
1104 @Perm(requires="pure(this) in alive",
1105 ensures="pure(this) in alive")
1106 public boolean getRequiresClauseSatisfiability() {
1107 return 0;
1108 }
1109 @Perm(requires="full(this) in alive",
1110 ensures="full(this) in alive")
1111 public void setReturnType(String str) {
1112 }
1113 @Perm(requires="full(this) in alive",
1114 ensures="full(this) in alive")
1115 public void setIdentifier(String str) {
1116 }
1117 @Perm(requires="full(this) in alive",
1118 ensures="full(this) in alive")
1119 public String getName() {
1120 return null;
1121 }
1122 @Perm(requires="pure(this) in alive",

```

```

1123 ensures="pure(this) in alive")
1124 public LinkedList<ESpecification> getEnsuresAPTS() {
1125     return null;
1126 }
1127 @Perm(requires="full(this) in alive",
1128 ensures="full(this) in alive")
1129 public boolean isConcurrentMethod(String str) {
1130     return 0;
1131 }
1132 @Perm(requires="full(this) in alive",
1133 ensures="full(this) in alive")
1134 public void setJMLPermission(String Permission) {
1135 }
1136 @Perm(requires="full(this) in alive",
1137 ensures="full(this) in alive")
1138 public String getJMLPermission() {
1139     return null;
1140 }
1141 @Perm(requires="pure(this) in alive",
1142 ensures="pure(this) in alive")
1143 public int getCaseNumber() {
1144     return 0;
1145 }
1146 @Perm(requires="full(this) in alive",
1147 ensures="full(this) in alive")
1148 public void setCaseNumber(int x) {
1149 }
1150 @Perm(requires="full(this) in alive",
1151 ensures="full(this) in alive")
1152 public void setName(String str) {
1153 }
1154 @Perm(requires="full(this) in alive",
1155 ensures="full(this) in alive")
1156 public String getReturnType() {
1157     return null;
1158 }
1159 @Perm(requires="full(this) in alive",
1160 ensures="full(this) in alive")
1161 public String getIdentifier() {
1162     return null;
1163 }
1164 @Perm(requires="full(this) in alive",
1165 ensures="full(this) in alive")
1166 public void addParameter(EParameter parameter) {
1167 }
1168 @Perm(requires="pure(this) in alive",
1169 ensures="pure(this) in alive")
1170 public LinkedList<ESpecification> getRequiresAPTS() {
1171     return null;
1172 }
1173 @Perm(requires="pure(this) in alive",
1174 ensures="pure(this) in alive")
1175 public LinkedList<EParameter> getParameters() {
1176     return null;
1177 }
1178 @Perm(requires="full(this) in alive",
1179 ensures="full(this) in alive")
1180 public void setIndex(int methodIndex) {
1181 }
1182 @Perm(requires="pure(this) in alive",
1183 ensures="pure(this) in alive")
1184 public int getIndex() {
1185     return 0;
1186 }
1187 @Perm(requires="full(this) in alive",
1188 ensures="full(this) in alive")
1189 public void setRequiresClauseSatisfiability(Boolean flag) {
1190 }
1191 @Perm(requires="full(this) in alive",
1192 ensures="full(this) in alive")
1193 public void setConcurrentMethod(String toMethod) {
1194 }
1196 public void addSpecifications(String annotation) {
1197 }
1199 }ENDOFCLASS
1201 @ClassStates({@State(name = "alive")})
1203 class SMCVisitor {

```

```

1204 @Perm(ensures="unique(this) in alive")
1205 SMCVisitor() { }

1207 @Perm(requires="pure(this) in alive",
1208 ensures="pure(this) in alive")
1209 private void addUnparsedSpecifications(String annotation) {
1210 }

1212 public void postVisit(ASTNode node) {
1213 }
1214 @Perm(requires="full(this) in alive",
1215 ensures="full(this) in alive")
1216 public boolean visit(PackageDeclaration node) {
1217     return 0;
1218 }

1220 public void endVisit(PackageDeclaration node) {
1221 }

1223 private void callParser(String annotation) {
1224 }

1226 }ENDOFCLASS

1228 @ClassStates({@State(name = "alive")})

1230 class PulseSettings {
1231     @Perm(ensures="unique(this) in alive")
1232     PulseSettings() { }

1234     @Perm(requires="pure(this) in alive",
1235 ensures="pure(this) in alive")
1236     int getInheritance() {
1237         return 0;
1238     }
1239     @Perm(requires="pure(this) in alive",
1240 ensures="pure(this) in alive")
1241     int getFullModel() {
1242         return 0;
1243     }
1244     @Perm(requires="pure(this) in alive",
1245 ensures="pure(this) in alive")
1246     int getInvariants() {
1247         return 0;
1248     }
1249     @Perm(requires="full(this) in alive",
1250 ensures="full(this) in alive")
1251     void setInvariants(int x) {
1252     }
1253     @Perm(requires="full(this) in alive",
1254 ensures="full(this) in alive")
1255     void setAliasPerObject(int x) {
1256     }
1257     @Perm(requires="full(this) in alive",
1258 ensures="full(this) in alive")
1259     void setFullModel(int x) {
1260     }
1261     @Perm(requires="full(this) in alive",
1262 ensures="full(this) in alive")
1263     void setDimensions(int x) {
1264     }
1265     @Perm(requires="full(this) in alive",
1266 ensures="full(this) in alive")
1267     void setInheritance(int x) {
1268     }
1269     @Perm(requires="pure(this) in alive",
1270 ensures="pure(this) in alive")
1271     int getAliasPerObject() {
1272         return 0;
1273     }
1274     @Perm(requires="pure(this) in alive",
1275 ensures="pure(this) in alive")
1276     int getDimensions() {
1277         return 0;
1278     }
1279 }

1280 }ENDOFCLASS

1282 @ClassStates({@State(name = "alive")})

1284 class EField {

```

```

1285 @Perm(ensures="unique(this) in alive")
1286 EField() { }

1288 @Perm(requires="full(this) in alive",
1289 ensures="full(this) in alive")
1290 public void setName(String str) {
1291 }
1292 @Perm(requires="full(this) in alive",
1293 ensures="full(this) in alive")
1294 public void setType(String str) {
1295 }
1296 @Perm(requires="full(this) in alive",
1297 ensures="full(this) in alive")
1298 public void setModifier(int mod) {
1299 }
1300 @Perm(requires="pure(this) in alive",
1301 ensures="pure(this) in alive")
1302 public int getObjectIndex() {
1303     return 0;
1304 }
1305 @Perm(requires="pure(this) in alive",
1306 ensures="pure(this) in alive")
1307 public int getModifier() {
1308     return 0;
1309 }
1310 @Perm(requires="full(this) in alive",
1311 ensures="full(this) in alive")
1312 public void setClassIndex(int classIndex) {
1313 }
1314 @Perm(requires="full(this) in alive",
1315 ensures="full(this) in alive")
1316 public String getName() {
1317     return null;
1318 }
1319 @Perm(requires="full(this) in alive",
1320 ensures="full(this) in alive")
1321 public String getType() {
1322     return null;
1323 }
1324 @Perm(requires="full(this) in alive",
1325 ensures="full(this) in alive")
1326 public void setObjectIndex(int objectIndex) {
1327 }
1328 @Perm(requires="pure(this) in alive",
1329 ensures="pure(this) in alive")
1330 public int getClassIndex() {
1331     return 0;
1332 }

1334 }ENDOFCLASS

1336 @ClassStates({@State(name = "alive")})

1338 class ESpecification {
1339 @Perm(ensures="unique(this) in alive")
1340 ESpecification() { }

1342 @Perm(requires="pure(this) in alive",
1343 ensures="pure(this) in alive")
1344 public EClass getParentClass() {
1345     return null;
1346 }
1347 @Perm(requires="full(this) in alive",
1348 ensures="full(this) in alive")
1349 public String getTS() {
1350     return null;
1351 }
1352 @Perm(requires="full(this) in alive",
1353 ensures="full(this) in alive")
1354 public String getAP() {
1355     return null;
1356 }

1358 public Object clone() {
1359     return null;
1360 }
1361 @Perm(requires="full(this) in alive",
1362 ensures="full(this) in alive")
1363 public void setAPTS(String ap, String ts) {
1364 }
1365 @Perm(requires="full(this) in alive",

```

```

1366 ensures="full(this) in alive")
1367 public void setAP(String ap) {
1368 }
1369 @Perm(requires="pure(this) in alive",
1370 ensures="pure(this) in alive")
1371 public String getFieldname() {
1372     return null;
1373 }
1375 }ENDOFCLASS
1377 @ClassStates({@State(name = "alive")})
1379 class EParameter {
1380 @Perm(ensures="unique(this) in alive")
1381 EParameter() {    }
1383 @Perm(requires="pure(this) in alive",
1384 ensures="pure(this) in alive")
1385 public LinkedList<ESpecification> getRequiresAPTS() {
1386     return null;
1387 }
1388 @Perm(requires="full(this) in alive",
1389 ensures="full(this) in alive")
1390 public void setNumber(int n) {
1391 }
1392 @Perm(requires="pure(this) in alive",
1393 ensures="pure(this) in alive")
1394 public int getNumber() {
1395     return 0;
1396 }
1397 @Perm(requires="full(this) in alive",
1398 ensures="full(this) in alive")
1399 public void setName(String str) {
1400 }
1401 @Perm(requires="full(this) in alive",
1402 ensures="full(this) in alive")
1403 public void setType(String str) {
1404 }
1405 @Perm(requires="pure(this) in alive",
1406 ensures="pure(this) in alive")
1407 public String getType() {
1408     return null;
1409 }
1410 @Perm(requires="full(this) in alive",
1411 ensures="full(this) in alive")
1412 public String getName() {
1413     return null;
1414 }
1415 @Perm(requires="pure(this) in alive",
1416 ensures="pure(this) in alive")
1417 public LinkedList<ESpecification> getEnsuresAPTS() {
1418     return null;
1419 }
1421 }ENDOFCLASS
1423 @ClassStates({@State(name = "alive")})
1425 class EInvariant {
1426 @Perm(ensures="unique(this) in alive")
1427 EInvariant() {    }
1429 @Perm(requires="pure(this) in alive",
1430 ensures="pure(this) in alive")
1431 public String getVariableType() {
1432     return null;
1433 }
1434 @Perm(requires="pure(this) in alive",
1435 ensures="pure(this) in alive")
1436 public String getStateName() {
1437     return null;
1438 }
1439 @Perm(requires="full(this) in alive",
1440 ensures="full(this) in alive")
1441 public void setVariableType(String type) {
1442 }
1444 public void setStateIndex(int stateIndex) {
1445 }
1446 @Perm(requires="full(this) in alive",

```



```

1447 ensures="full(this) in alive")
1448 public void setAP(String str) {
1449 }
1450 @Perm(requires="full(this) in alive",
1451 ensures="full(this) in alive")
1452 public void setVariable(String str) {
1453 }
1454 @Perm(requires="full(this) in alive",
1455 ensures="full(this) in alive")
1456 public void setState(String str) {
1457 }
1458 @Perm(requires="pure(this) in alive",
1459 ensures="pure(this) in alive")
1460 public String getAP() {
1461     return null;
1462 }
1463 @Perm(requires="pure(this) in alive",
1464 ensures="pure(this) in alive")
1465 public String getVariable() {
1466     return null;
1467 }
1468 @Perm(requires="full(this) in alive",
1469 ensures="full(this) in alive")
1470 public LinkedList<Invariant> getStateInvariants(EPackage _pkg) {
1471     return null;
1472 }
1473 }ENDOFCLASS
1474
1475 @ClassStates({@State(name = "alive")})
1476
1477 class EBoolInvariant {
1478 @Perm(ensures="unique(this) in alive")
1479 EBoolInvariant() { }
1480
1481 @Perm(requires="pure(this) in alive",
1482 ensures="pure(this) in alive")
1483 public String getVariable() {
1484     return null;
1485 }
1486 }
1487 @Perm(requires="pure(this) in alive",
1488 ensures="pure(this) in alive")
1489 public String getValue() {
1490     return null;
1491 }
1492 }ENDOFCLASS
1493
1494 @ClassStates({@State(name = "alive")})
1495
1496 class EDim {
1497 @Perm(ensures="unique(this) in alive")
1498 EDim() { }
1499
1500 @Perm(requires="pure(this) in alive",
1501 ensures="pure(this) in alive")
1502 public ArrayList<String> getValues() {
1503     return null;
1504 }
1505 }
1506 @Perm(requires="full(this) in alive",
1507 ensures="full(this) in alive")
1508 public void setName(String str) {
1509 }
1510 @Perm(requires="pure(this) in alive",
1511 ensures="pure(this) in alive")
1512 public String getName(String str) {
1513     return null;
1514 }
1515 @Perm(requires="full(this) in alive",
1516 ensures="full(this) in alive")
1517 public void addValue(String str) {
1518 }
1519 }ENDOFCLASS
1520
1521 @ClassStates({@State(name = "alive")})
1522
1523 class specificationStruct {
1524 @Perm(ensures="unique(this) in alive")
1525 specificationStruct() { }

```

```

1529 }ENDOFCLASS
1531 @ClassStates({@State(name = "alive")})
1533 class Clause {
1534   @Perm(ensures="unique(this) in alive")
1535   Clause() { }
1537 }ENDOFCLASS
1540 @ClassStates({@State(name = "alive")})
1542 class Signature {
1543   @Perm(ensures="unique(this) in alive")
1544   Signature() { }
1546 }ENDOFCLASS
1549 @ClassStates({@State(name = "alive")})
1551 class MethodFindVisitor {
1552   @Perm(ensures="unique(this) in alive")
1553   MethodFindVisitor() { }
1555   @Perm(requires="full(this) in alive",
1556         ensures="full(this) in alive")
1557   public boolean visit(MethodDeclaration methodDeclaration) {
1558     return 0;
1559   }
1561 }ENDOFCLASS
1563 @ClassStates({@State(name = "alive")})
1565 class GAPHandler {
1566   @Perm(ensures="unique(this) in alive")
1567   GAPHandler() { }
1569   public void addHandlerListener(IHandlerListener handlerListener) {
1570   }
1572   public void dispose() {
1573   }
1575   @Perm(requires="full(this) in alive",
1576         ensures="full(this) in alive")
1577   public Object execute(ExecutionEvent event) {
1578     return null;
1579   }
1581   @Perm(requires="full(this) in alive",
1582         ensures="full(this) in alive")
1583   private void extractSettings(ExecutionEvent event) {
1584   }
1586   public boolean isHandled() {
1587     return 0;
1588   }
1590   public void removeHandlerListener(IHandlerListener handlerListener) {
1591   }
1593 }ENDOFCLASS
1596 @ClassStates({@State(name = "alive")})
1598 class GAPIFileAction {
1599   @Perm(ensures="unique(this) in alive")
1600   GAPIFileAction() { }
1602   @Perm(requires="full(this) in alive",
1603         ensures="full(this) in alive")
1604   public void selectionChanged(IAction action, ISelection selection) {
1605   }
1607   public void setActivePart(IAction action, IWorkbenchPart targetPart) {
1608   }
1610   @Perm(requires="full(this) in alive",
1611         ensures="full(this) in alive")

```

```

1609 public void run(IAction action) {
1610 }
1612 }ENDOFCLASS
1614 @ClassStates({@State(name = "alive")})
1616 class {
1617   @Perm(ensures="unique(this) in alive")
1618   () { }
1620   @Perm(requires="unique(this) in alive",
1621     ensures="unique(this) in alive")
1622   protected IStatus run(IProgressMonitor monitor) {
1623     return null;
1624   }
1626 }ENDOFCLASS
1628 @ClassStates({@State(name = "alive")})
1630 class Main {
1631   @Perm(ensures="unique(this) in alive")
1632   Main() { }
1634   @Perm(requires="unique(this) in alive",
1635     ensures="unique(this) in alive")
1636   void main(String[] args) {
1637   }
1639   String testRead(String file) {
1640     return null;
1641   }
1642   @Perm(requires="full(this) in alive",
1643     ensures="full(this) in alive")
1644   void seprateJavaFile(String str) {
1645   }
1647   void anTest() {
1648   }
1650 }ENDOFCLASS
1652 @ClassStates({@State(name = "alive")})
1654 class TypestateReturn {
1655   @Perm(ensures="unique(this) in alive")
1656   TypestateReturn() { }
1659 }ENDOFCLASS
1661 @ClassStates({@State(name = "alive")})
1663 class AtApPermissionReturn {
1664   @Perm(ensures="unique(this) in alive")
1665   AtApPermissionReturn() { }
1668 }ENDOFCLASS
1670 @ClassStates({@State(name = "alive")})
1672 class AccesspermissionReturn {
1673   @Perm(ensures="unique(this) in alive")
1674   AccesspermissionReturn() { }
1677 }ENDOFCLASS
1679 @ClassStates({@State(name = "alive")})
1681 class PluralLexer {
1682   @Perm(ensures="unique(this) in alive")
1683   PluralLexer() { }
1686   public String getGrammarFileName() {
1687     return null;
1688   }
1689   @Perm(requires="pure(this) in alive",

```

```

1690 ensures="pure(this) in alive")
1691 void mATFULL() {
1692 }
1693 @Perm(requires="pure(this) in alive",
1694 ensures="pure(this) in alive")
1695 void mATPURE() {
1696 }
1697 @Perm(requires="pure(this) in alive",
1698 ensures="pure(this) in alive")
1699 void mATIMMUTABLE() {
1700 }
1701 @Perm(requires="pure(this) in alive",
1702 ensures="pure(this) in alive")
1703 void mATSHARE() {
1704 }
1705 @Perm(requires="pure(this) in alive",
1706 ensures="pure(this) in alive")
1707 void mATUNIQUE() {
1708 }
1709 @Perm(requires="pure(this) in alive",
1710 ensures="pure(this) in alive")
1711 void mPUBLICBEHAVIOR() {
1712 }
1713 @Perm(requires="pure(this) in alive",
1714 ensures="pure(this) in alive")
1715 void mFULL() {
1716 }
1717 @Perm(requires="pure(this) in alive",
1718 ensures="pure(this) in alive")
1719 void mPURE() {
1720 }
1721 @Perm(requires="pure(this) in alive",
1722 ensures="pure(this) in alive")
1723 void mIMMUTABLE() {
1724 }
1725 @Perm(requires="pure(this) in alive",
1726 ensures="pure(this) in alive")
1727 void mSHARE() {
1728 }
1729 @Perm(requires="pure(this) in alive",
1730 ensures="pure(this) in alive")
1731 void mUNIQUE() {
1732 }
1733 @Perm(requires="pure(this) in alive",
1734 ensures="pure(this) in alive")
1735 void mNONE() {
1736 }
1737 @Perm(requires="pure(this) in alive",
1738 ensures="pure(this) in alive")
1739 void mLSBRACKET() {
1740 }
1741 @Perm(requires="pure(this) in alive",
1742 ensures="pure(this) in alive")
1743 void mRSBRACKET() {
1744 }
1745 @Perm(requires="pure(this) in alive",
1746 ensures="pure(this) in alive")
1747 void mPERM() {
1748 }
1749 @Perm(requires="pure(this) in alive",
1750 ensures="pure(this) in alive")
1751 void mEQUAL() {
1752 }
1753 @Perm(requires="pure(this) in alive",
1754 ensures="pure(this) in alive")
1755 void mEQUALOPERATOR() {
1756 }
1757 @Perm(requires="pure(this) in alive",
1758 ensures="pure(this) in alive")
1759 void mIN() {
1760 }
1761 @Perm(requires="pure(this) in alive",
1762 ensures="pure(this) in alive")
1763 void mTHIS() {
1764 }
1765 @Perm(requires="pure(this) in alive",
1766 ensures="pure(this) in alive")
1767 void mRESULT() {
1768 }
1769 @Perm(requires="pure(this) in alive",
1770 ensures="pure(this) in alive")

```

```

1771 void mPARAM() {
1772 }
1773 @Perm(requires="pure(this) in alive",
1774 ensures="pure(this) in alive")
1775 void mREQUIRES() {
1776 }
1777 @Perm(requires="pure(this) in alive",
1778 ensures="pure(this) in alive")
1779 void mENSURES() {
1780 }
1781 @Perm(requires="pure(this) in alive",
1782 ensures="pure(this) in alive")
1783 void mQUOTE() {
1784 }
1785 @Perm(requires="pure(this) in alive",
1786 ensures="pure(this) in alive")
1787 void mAND() {
1788 }
1789 @Perm(requires="pure(this) in alive",
1790 ensures="pure(this) in alive")
1791 void mUSE() {
1792 }
1793 @Perm(requires="pure(this) in alive",
1794 ensures="pure(this) in alive")
1795 void mUSEFIELDS() {
1796 }
1797 @Perm(requires="pure(this) in alive",
1798 ensures="pure(this) in alive")
1799 void mPUNCTUATION() {
1800 }
1801 @Perm(requires="pure(this) in alive",
1802 ensures="pure(this) in alive")
1803 void mCASES() {
1804 }
1805 @Perm(requires="pure(this) in alive",
1806 ensures="pure(this) in alive")
1807 void mLCBRACKET() {
1808 }
1809 @Perm(requires="pure(this) in alive",
1810 ensures="pure(this) in alive")
1811 void mRCBRACKET() {
1812 }
1813 @Perm(requires="pure(this) in alive",
1814 ensures="pure(this) in alive")
1815 void mCLASSSTATES() {
1816 }
1817 @Perm(requires="pure(this) in alive",
1818 ensures="pure(this) in alive")
1819 void mREFINE() {
1820 }
1821 @Perm(requires="pure(this) in alive",
1822 ensures="pure(this) in alive")
1823 void mVALUE() {
1824 }
1825 @Perm(requires="pure(this) in alive",
1826 ensures="pure(this) in alive")
1827 void mSTATE() {
1828 }
1829 @Perm(requires="pure(this) in alive",
1830 ensures="pure(this) in alive")
1831 void mSTATES() {
1832 }
1833 @Perm(requires="pure(this) in alive",
1834 ensures="pure(this) in alive")
1835 void mDIM() {
1836 }
1837 @Perm(requires="pure(this) in alive",
1838 ensures="pure(this) in alive")
1839 void mName() {
1840 }
1841 @Perm(requires="pure(this) in alive",
1842 ensures="pure(this) in alive")
1843 void mINV() {
1844 }
1845 @Perm(requires="pure(this) in alive",
1846 ensures="pure(this) in alive")
1847 void mOPERATOR() {
1848 }
1849 @Perm(requires="pure(this) in alive",
1850 ensures="pure(this) in alive")
1851 void mSEMICOLON() {

```

```

1852 }
1853 @Perm(requires="pure(this) in alive",
1854 ensures="pure(this) in alive")
1855 void mLESS() {
1856 }
1857 @Perm(requires="pure(this) in alive",
1858 ensures="pure(this) in alive")
1859 void mLESSTHANEQUAL() {
1860 }
1861 @Perm(requires="pure(this) in alive",
1862 ensures="pure(this) in alive")
1863 void mGREATER() {
1864 }
1865 @Perm(requires="pure(this) in alive",
1866 ensures="pure(this) in alive")
1867 void mGREATERTHANEQUAL() {
1868 }
1869 @Perm(requires="pure(this) in alive",
1870 ensures="pure(this) in alive")
1871 void mANDD() {
1872 }
1873 @Perm(requires="pure(this) in alive",
1874 ensures="pure(this) in alive")
1875 void mOR() {
1876 }
1877 @Perm(requires="pure(this) in alive",
1878 ensures="pure(this) in alive")
1879 void mJMLSTART() {
1880 }
1881 @Perm(requires="pure(this) in alive",
1882 ensures="pure(this) in alive")
1883 void mJMLEND() {
1884 }
1885 @Perm(requires="pure(this) in alive",
1886 ensures="pure(this) in alive")
1887 void mPLUSMINUSOPERATOR() {
1888 }
1889 @Perm(requires="pure(this) in alive",
1890 ensures="pure(this) in alive")
1891 void mASSIGNABLE() {
1892 }
1893 @Perm(requires="pure(this) in alive",
1894 ensures="pure(this) in alive")
1895 void mNOTHING() {
1896 }
1897 @Perm(requires="pure(this) in alive",
1898 ensures="pure(this) in alive")
1899 void mEVERYTHING() {
1900 }
1901 @Perm(requires="pure(this) in alive",
1902 ensures="pure(this) in alive")
1903 void mGHOST() {
1904 }
1905 @Perm(requires="pure(this) in alive",
1906 ensures="pure(this) in alive")
1907 void mINT() {
1908 }
1909 @Perm(requires="pure(this) in alive",
1910 ensures="pure(this) in alive")
1911 void mINVARIANT() {
1912 }
1913 @Perm(requires="pure(this) in alive",
1914 ensures="pure(this) in alive")
1915 void mOLD() {
1916 }
1917 @Perm(requires="pure(this) in alive",
1918 ensures="pure(this) in alive")
1919 void mID() {
1920 }
1921 @Perm(requires="pure(this) in alive",
1922 ensures="pure(this) in alive")
1923 void mNUMBERS() {
1924 }
1925 @Perm(requires="pure(this) in alive",
1926 ensures="pure(this) in alive")
1927 void mWS() {
1928 }
1929 @Perm(requires="full(this) in alive",
1930 ensures="full(this) in alive")
1931 public void mTokens() {
1932 }

```

```

1934 }ENDOFCLASS
1936 @ClassStates({@State(name = "alive")})
1938 class DFA7 {
1939     @Perm(ensures="unique(this) in alive")
1940     DFA7() { }
1942
1943     public String getDescription() {
1944         return null;
1945     }
1947 }ENDOFCLASS
1949 @ClassStates({@State(name = "alive")})
1951 class EAPTypeState {
1952     @Perm(ensures="unique(this) in alive")
1953     EAPTypeState() { }
1955     @Perm(requires="full(this) in alive",
1956           ensures="full(this) in alive")
1957     public void setAP(String str) {
1958     }
1959     @Perm(requires="full(this) in alive",
1960           ensures="full(this) in alive")
1961     public String getAP(String str) {
1962         return null;
1963     }
1964     @Perm(requires="full(this) in alive",
1965           ensures="full(this) in alive")
1966     public void setTS(String str) {
1967     }
1968     @Perm(requires="full(this) in alive",
1969           ensures="full(this) in alive")
1970     public String getTS() {
1971         return null;
1972     }
1974 }ENDOFCLASS

```