Summary

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

Table 1: Pulse Analysis Summary

Classes	Methods	States	Unsatisfiable Clauses	Unreachable States	Possible concurrent Methods	Total. no. of pairs	No. of concurrent pairs	Percentage of concurrent Methods
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
EGrarphWriter	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

Table 2: Methods Requires Clause Satisfiability

Method	Satisfiability
Activator	$\surd$
start	$\surd$
stop	$\surd$
getDefault	$\surd$
start	$\times$
stop	$\times$
getDefault	$\times$
getImageDescriptor	$\surd$

Table 3: State Transition Matrix

	alive
alive	$\uparrow$

Table 4: Methods Concurrency Matrix

	Activator	start	stop	getDefault	start	stop	getDefault	getImageDescriptor
Activator	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$
start	$\nparallel$	$\nparallel$	$\nparallel$	$\parallel$	$\nparallel$	$\nparallel$	$\parallel$	$\parallel$
stop	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\parallel$
getDefault	$\nparallel$	$\parallel$	$\nparallel$	$\parallel$	$\parallel$	$\nparallel$	$\parallel$	$\parallel$
start	$\nparallel$	$\nparallel$	$\nparallel$	$\parallel$	$\nparallel$	$\nparallel$	$\parallel$	$\parallel$
stop	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\parallel$
getDefault	$\nparallel$	$\parallel$	$\nparallel$	$\parallel$	$\parallel$	$\nparallel$	$\parallel$	$\parallel$
getImageDescriptor	$\nparallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$

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	√
atappermission	√
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	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∥	∥	∦	∦	∦	∦	∦	∦	∥	∦
atappermission	∦	∥	∥	∥	∥	∥	∥	∥	∥	∥	∥	∥	∥	∥	∥	∥	∥	∥	∥	∥
usevalue	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∥	∥	∦	∦	∦	∦	∦	∦	∥	∦
classstates	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∥	∥	∦	∦	∦	∦	∦	∦	∥	∦
startClassstates	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∥	∥	∦	∦	∦	∦	∦	∦	∥	∦
state	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∥	∥	∦	∦	∦	∦	∦	∦	∥	∦
endclassstates	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∥	∥	∦	∦	∦	∦	∦	∦	∥	∦
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

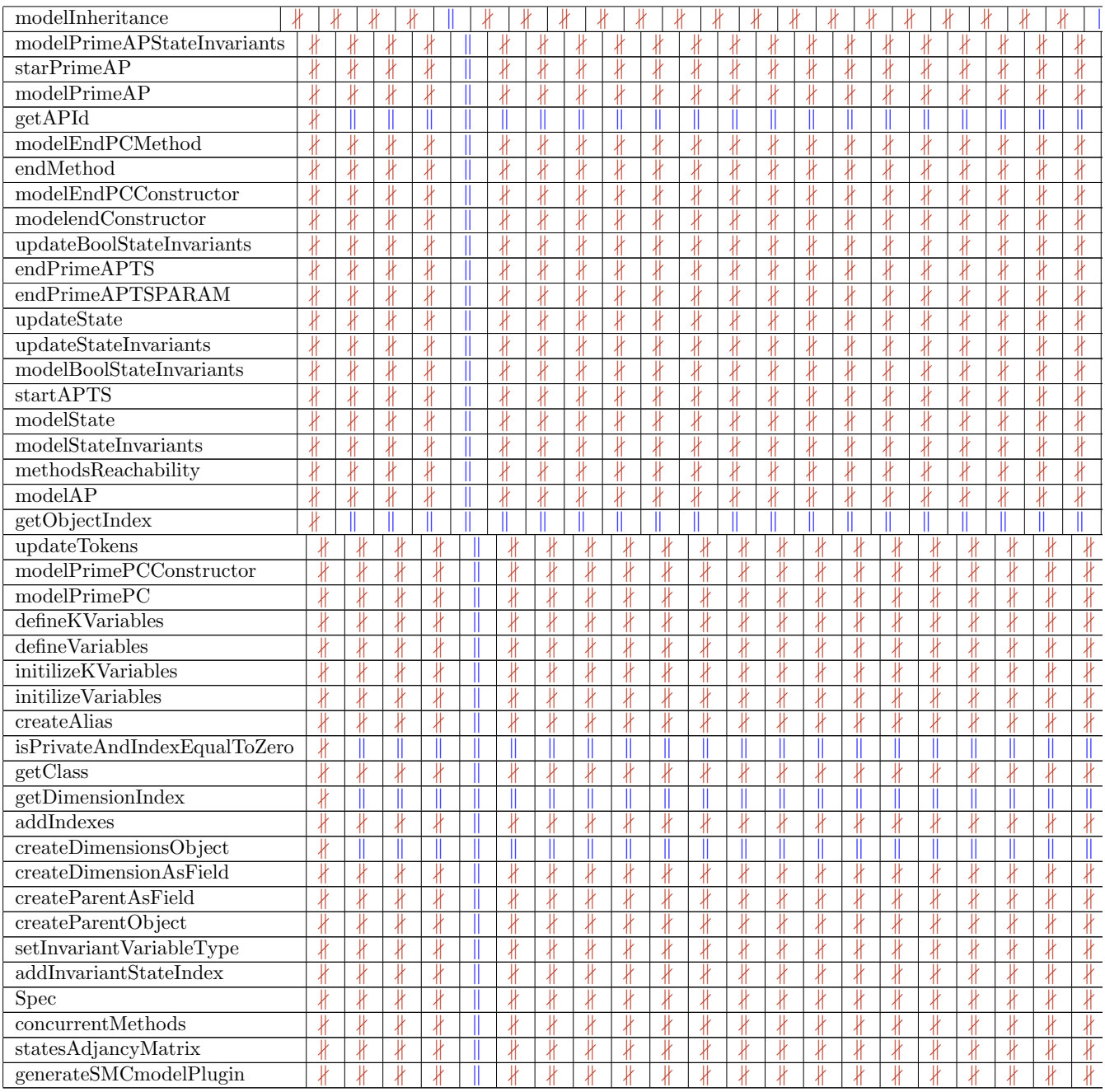
Method	Satisfiability
ESMCModel	√
setK	√
declarationsAndinitilizations	√
initialize	√
comment	√
modelAlias	√
isClassExist	√
Transitions	√
createInstanceInModel	√
modelPrimePCandMethod	√
startMethod	√
modelPCConstructor	√
modelAPs	√
modelPCMethod	√
startAPTSPARAM	√
startPrimeTSPARAM	√
modelPrimeConstructor	√
modelInheritance	√
modelPrimeAPStateInvariants	√
starPrimeAP	√
modelPrimeAP	√
getAPId	√
modelEndPCMethod	√
endMethod	√
modelEndPCConstructor	√
modelendConstructor	√
updateBoolStateInvariants	√
endPrimeAPTS	√
endPrimeAPTSPARAM	√
updateState	√
updateStateInvariants	√
modelBoolStateInvariants	√
startAPTS	√
modelState	√
modelStateInvariants	√
methodsReachability	√
modelAP	√
getObjectIndex	√
updateTokens	√
modelPrimePCConstructor	√
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	declarationsAndinitilizations	initialize	comment	modelAlias	isClassExist	Transitions	createInstanceInModel	modelPrimePCandMethod	startMethod	modelPCConstructor	modelAPs	modelPCMethod	startAPTSPARAM	startPrimeTSPARAM	modelPrimeConstructor	modelInheritance	modelPrimeAPStateInvariants	starPrimeAP	modelPrimeAP	
ESMCModel	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦
setK	∦	∦	∦	∦	‖	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦
declarationsAndinitilizations	∦	∦	∦	∦	‖	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦
initialize	∦	∦	∦	∦	‖	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦
comment	∦	‖	‖	‖	‖	‖	‖	‖	‖	‖	‖	‖	‖	‖	‖	‖	‖	‖	‖	‖	‖	‖
modelAlias	∦	∦	∦	∦	‖	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦
isClassExist	∦	∦	∦	∦	‖	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦
Transitions	∦	∦	∦	∦	‖	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦
createInstanceInModel	∦	∦	∦	∦	‖	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦
modelPrimePCandMethod	∦	∦	∦	∦	‖	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦
startMethod	∦	∦	∦	∦	‖	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦
modelPCConstructor	∦	∦	∦	∦	‖	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦
modelAPs	∦	∦	∦	∦	‖	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦
modelPCMethod	∦	∦	∦	∦	‖	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦
startAPTSPARAM	∦	∦	∦	∦	‖	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦
startPrimeTSPARAM	∦	∦	∦	∦	‖	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦
modelPrimeConstructor	∦	∦	∦	∦	‖	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦	∦


modelInheritance
modelPrimeAPStateInvariants
starPrimeAP
modelPrimeAP
getAPId
modelEndPCMethod
endMethod
modelEndPCConstructor
modelendConstructor
updateBoolStateInvariants
endPrimeAPTS
endPrimeAPTSPARAM
updateState
updateStateInvariants
modelBoolStateInvariants
startAPTS
modelState
modelStateInvariants
methodsReachability
modelAP
getObjectIndex
updateTokens
modelPrimePCConstructor
modelPrimePC
defineKVariables
defineVariables
initilizeKVariables
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
specificationStruct	√

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	√
seprateJavaFile	√
anTest	√

Table 99: State Transition Matrix

	alive
alive	↑

Table 100: Methods Concurrency Matrix

	Main	main	testRead	seprateJavaFile	anTest
Main	∦	∦	∦	∦	∦
main	∦	∦	∥	∦	∥
testRead	∦	∥	∥	∥	∥
seprateJavaFile	∦	∦	∥	∦	∥
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

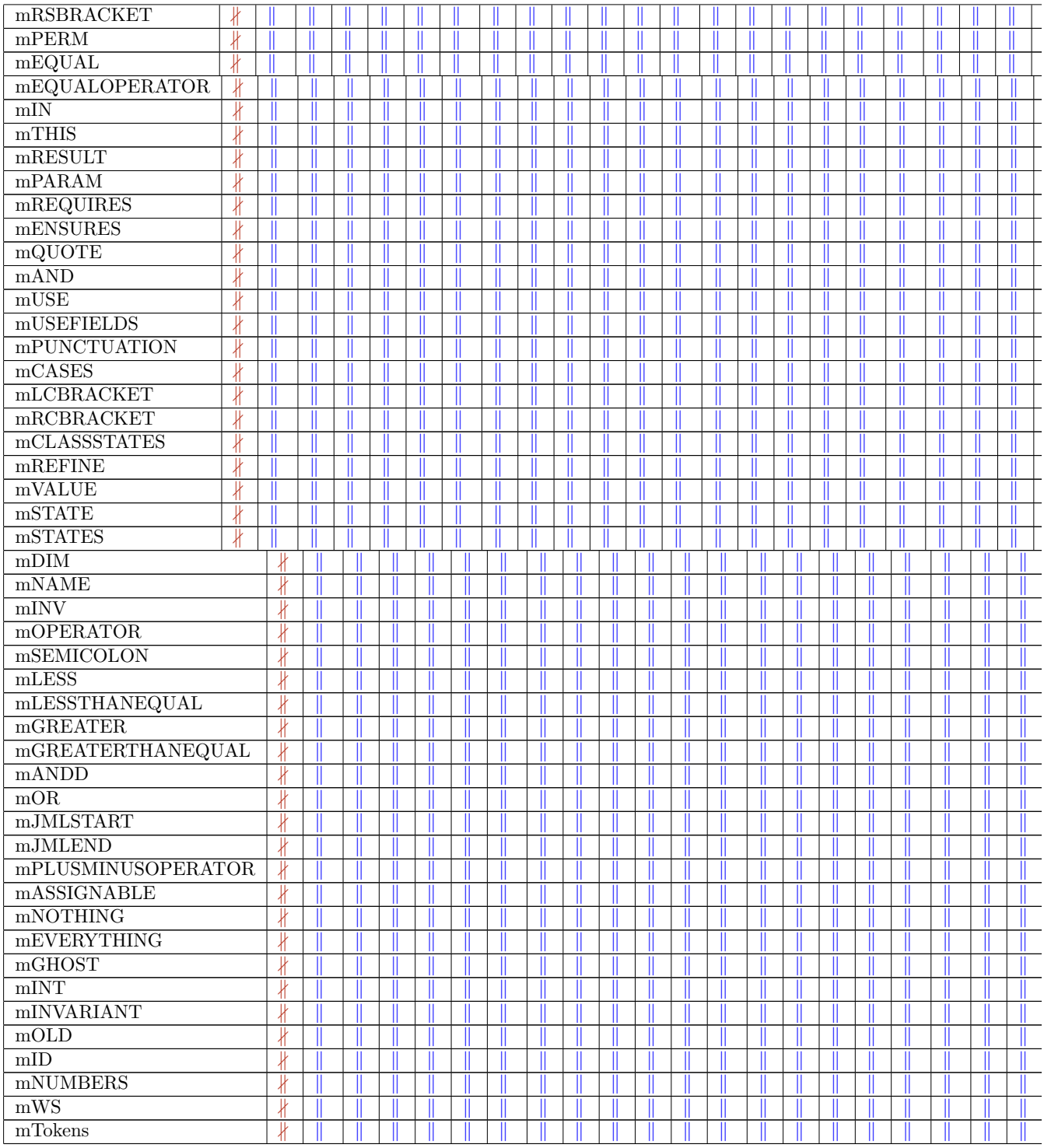
Method	Satisfiability
PluralLexer	√
getGrammarFileName	√
mATFULL	√
mATPURE	√
mATIMMUTABLE	√
mATSHARE	√
mATUNIQUE	√
mPUBLICBEHAVIOR	√
mFULL	√
mPURE	√
mIMMUTABLE	√
mSHARE	√
mUNIQUE	√
mNONE	√
mLSBRACKET	√
mRSBRACKET	√
mPERM	√
mEQUAL	√
mEQUALOPERATOR	√
mIN	√
mTHIS	√
mRESULT	√
mPARAM	√
mREQUIRES	√
mENSURES	√
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	mRESULT
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

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	$\surd$
setAP	$\surd$
getAP	$\surd$
setTS	$\surd$
getTS	$\surd$

Table 114: State Transition Matrix

	alive
alive	$\uparrow$

Table 115: Methods Concurrency Matrix

	EAPTypeState	setAP	getAP	setTS	getTS
EAPTypeState	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$
setAP	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$
getAP	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$
setTS	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$
getTS	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$	$\nparallel$

41 Abbreviation

Table 116: Used Abbreviation

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

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 }
```

```
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
private String translateClassSpecifications(String JProgram) {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
private void parseAndStoreJMLAnnotation(String JMLAnnotation) {
}
@Perm(requires="unique(this) in alive",
ensures="unique(this) in alive")
private String translateMethodSpecification(String JProgram) {
 return null;
}

public String readFileAsString(String filePath) {
 return null;
}

}ENDOFCLASS

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

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

@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void jmlSpecifications() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void jmlClassSpecifications() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void jmlGhostDeclaration() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void jmlMethodSpecification() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void jmlRequires() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void jmlReq() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void jmlLessThanEqualReq() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void jmlEnsures() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void jmlEns() {
}
@Perm(ensures="none(this) in alive")
public String[] getTokenNames() {
 return null;
}

public String getGrammarFileName() {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void jmlOldEns() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void specifications() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
```

```
 void perm() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void requiresensuresClause() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void requiresClause() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void reaccesspermissionTypestates() {
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
 AccesspermissionReturn accesspermission() {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 TypestateReturn typestate() {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void attype() {
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
 AtApPermissionReturn atappermission() {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void usevalue() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void classstates() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void startClassstates() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void state() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void endclassstates() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void refine() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void states() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void dimension() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void item() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void invariant() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void condition() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void other() {
}
```

```
}ENDOFCLASS

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

class EJmlSpecification {
@Perm(ensures="unique(this) in alive")
EJmlSpecification() {   }

@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void setDimensionName(String str) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void setEnsures(String str) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void setPerm(String str) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void addRequires(String str) {
}
@Perm(requires="unique(this) in alive",
ensures="unique(this) in alive")
 void reset() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 String JmlClassSpec2PluralClassSpec() {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 String noRequires() {
 return null;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
 String getPerm() {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 String determineEnsures(String req) {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 String oneRequires() {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 String moreRequires() {
 return null;
}

}ENDOFCLASS

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

class EGhost {
@Perm(ensures="unique(this) in alive")
EGhost() {   }

@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public void setDimensionName(String str) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public void setDimensionValues(int low, int high) {
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public String getDimensionName() {
 return null;
}
```

```
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public int getLowValueofInv() {
 return 0;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public int getHighValueofInv() {
 return 0;
}

}ENDOFCLASS

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

class Time {
@Perm(ensures="unique(this) in alive")
Time() {   }

@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public String toString() {
 return null;
}

}ENDOFCLASS

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

class FileReader {
@Perm(ensures="unique(this) in alive")
FileReader() {   }


 String readFile(String pathname) {
 return null;
}

}ENDOFCLASS

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

class UserSelectedClassesAnalysis {
@Perm(ensures="unique(this) in alive")
UserSelectedClassesAnalysis() {   }


private CompilationUnit getCompilationUnit(String prog) {
 return null;
}
@Perm(requires="unique(this) in alive",
ensures="unique(this) in alive")
 void analyzeFromCommandLine(LinkedList<String> inputFiles, String strType, String strK) {
}
@Perm(requires="unique(this) in alive",
ensures="unique(this) in alive")
public void analyzeFromPlugin(List<ICompilationUnit> compilationUnitList, int test) {
}
@Perm(requires="unique(this) in alive",
ensures="unique(this) in alive")
 void callModelCheckerThroughCommandLine() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 Time getTime() {
 return null;
}
@Perm(requires="unique(this) in alive",
ensures="unique(this) in alive")
 void CreatePdfSummary_CommandLine(Time starttime2, Time endtime2) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void callModelCheckerThroughPlugin() {
}

 void createPdfSummaryPlugin(Time starttime2, Time endtime2) {
}

 void makePdfCommandLine() {
}
```

```
 void makePdfPlugin() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void printMethodMetrics() {
}

}ENDOFCLASS

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

class EVMDDSMCGenerator {
@Perm(ensures="unique(this) in alive")
EVMDDSMCGenerator() {   }

@Perm(requires="unique(this) in alive",
ensures="unique(this) in alive")
 void reset() {
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
 void addBoolStateInvariant(String variable, String operator, String value) {
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
 EPackage getPkgObject() {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 String modifyConstructorSpecifications(String prog) {
 return null;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
 void addState(String stateName) {
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
 void addRequiresParam_AP_TS(String ap, String ts, String argumentNumber) {
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
 void addEnsuresParamAPTS(String ap, String ts, String argumentNumber) {
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
 void addStateInvariant(String accessPermission, String variable, String state) {
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
 void addDimensionValue(String value) {
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
 void addEnsuresAPTS(String ap, String ts) {
}

}ENDOFCLASS

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

class EPackage {
@Perm(ensures="unique(this) in alive")
EPackage() {   }

@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public LinkedList<EClass> getClasses() {
 return null;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public int getTotalReachableStates() {
 return 0;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public String getName() {
 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 }

493 }ENDOFCLASS

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

497 class EGeneratedPluralSpecification {
498 @Perm(ensures="unique(this) in alive")
499 EGeneratedPluralSpecification() {   }

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 }

510 }ENDOFCLASS

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

514 class ESMCModel {
515 @Perm(ensures="unique(this) in alive")
516 ESMCModel() {   }

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

762 }ENDOFCLASS

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

766 class WorkspaceUtilities {
767 @Perm(ensures="unique(this) in alive")
768 WorkspaceUtilities() {   }


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 }

790  String getWorkspaceRelativeName(IJavaElement element) {
791  return null;
792 }

794  Map<ICompilationUnit,ASTNode> parseCompilationUnits(List<ICompilationUnit> compilationUnits) {
795  return null;
796 }

798  List<MethodDeclaration> scanForMethodDeclarations(Map<ICompilationUnit,ASTNode>
       compilationUnitToASTNode) {
```

```
 return null;
}

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

}ENDOFCLASS

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

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

@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public String getName() {
 return null;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public LinkedList<String> getTransitions() {
 return null;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public LinkedList<EState> getReachableStates() {
 return null;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public LinkedList<EState> getStates() {
 return null;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public LinkedList<EMethod> getMethods() {
 return null;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public String getSuperClassName() {
 return null;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public int findStateIndex(String st) {
 return 0;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public int getIndex() {
 return 0;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public LinkedList<EField> getFields() {
 return null;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public ArrayList<String> getVariablesofBooleanInvariants() {
 return null;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public LinkedList<EDim> getDimensions() {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public void createObject() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public EMethod getConstructor() {
 return null;
}
@Perm(requires="full(this) in alive",
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 }

923 }ENDOFCLASS

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

927 class EGrarphWriter {
928 @Perm(ensures="unique(this) in alive")
929 EGrarphWriter() {   }

931 @Perm(requires="full(this) in alive",
932 ensures="full(this) in alive")
933  void addTrnsitions(String str) {
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 }

949 }ENDOFCLASS

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

953 class EState {
954 @Perm(ensures="unique(this) in alive")
955 EState() {   }

957 @Perm(requires="pure(this) in alive",
958 ensures="pure(this) in alive")
959 public int isReachable() {
960  return 0;
```

```
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 }

1000 }ENDOFCLASS

1002 @ClassStates({@State(name = "alive")})

1004 class EOutputLatex {
1005 @Perm(ensures="unique(this) in alive")
1006 EOutputLatex() {   }

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")
```

```
 void writeMethodConcurrencyMatrix(EClass _class) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void addConcurrencyMatrixHeaders(EClass _class) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void writeAbbervations() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void setText(String str) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void parseTransitions(String str) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void parseSinkStates(String str) {
}
@Perm(requires="unique(this) in alive",
ensures="unique(this) in alive")
 void create_Plugin() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void addConcurrencyMatrixRows(EClass _class) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 String getConcurrencyValue(EMethod _method, EMethod __method) {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void addSummaryTableColumns() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void addSTMNumberofColumns(EClass _class) {
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
 String getStateReachabilityValue(EState _state, EState __state) {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 EMethod getMethod(String className, String methodName) {
 return null;
}

}ENDOFCLASS

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

class EMethod {
@Perm(ensures="unique(this) in alive")
EMethod() {   }

@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public boolean getRequiresClauseSatisfiability() {
 return 0;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public void setReturnType(String str) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public void setIdentifier(String str) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public String getName() {
 return null;
}
@Perm(requires="pure(this) in alive",
```

```
ensures="pure(this) in alive")
public LinkedList<ESpecification> getEnsuresAPTS() {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public boolean isConcurrentMethod(String str) {
 return 0;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public void setJMLPermission(String Permission) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public String getJMLPermission() {
 return null;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public int getCaseNumber() {
 return 0;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public void setCaseNumber(int x) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public void setName(String str) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public String getReturnType() {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public String getIdentifier() {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public void addParameter(EParameter parameter) {
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public LinkedList<ESpecification> getRequiresAPTS() {
 return null;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public LinkedList<EParameter> getParameters() {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public void setIndex(int methodIndex) {
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public int getIndex() {
 return 0;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public void setRequiresClauseSatisfiability(Boolean flag) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public void setConcurrentMethod(String toMethod) {
}

public void addSpecifications(String annotation) {
}

}ENDOFCLASS

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

class SMCVisitor {
```

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

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

public void postVisit(ASTNode node) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public boolean visit(PackageDeclaration node) {
 return 0;
}

public void endVisit(PackageDeclaration node) {
}

private void callParser(String annotation) {
}

}ENDOFCLASS

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

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

@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
 int getInheritance() {
 return 0;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
 int getFullModel() {
 return 0;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
 int getInvariants() {
 return 0;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void setInvariants(int x) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void setAliasPerObject(int x) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void setFullModel(int x) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void setDimensions(int x) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
 void setInheritance(int x) {
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
 int getAliasPerObject() {
 return 0;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
 int getDimensions() {
 return 0;
}

}ENDOFCLASS

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

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",
```

```
ensures="full(this) in alive")
public void setAP(String ap) {
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public String getFieldName() {
 return null;
}

}ENDOFCLASS

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

class EParameter {
@Perm(ensures="unique(this) in alive")
EParameter() {   }

@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public LinkedList<ESpecification> getRequiresAPTS() {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public void setNumber(int n) {
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public int getNumber() {
 return 0;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public void setName(String str) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public void setType(String str) {
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public String getType() {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public String getName() {
 return null;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public LinkedList<ESpecification> getEnsuresAPTS() {
 return null;
}

}ENDOFCLASS

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

class EInvariant {
@Perm(ensures="unique(this) in alive")
EInvariant() {   }

@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public String getVariableType() {
 return null;
}
@Perm(requires="pure(this) in alive",
ensures="pure(this) in alive")
public String getStateName() {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public void setVariableType(String type) {
}

public void setStateIndex(int stateIndex) {
}
@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<EInvariant> getStateInvariants(EPackage _pkg) {
1471  return null;
1472 }

1474 }ENDOFCLASS

1476 @ClassStates({@State(name = "alive")})

1478 class EBoolInvariant {
1479 @Perm(ensures="unique(this) in alive")
1480 EBoolInvariant() {   }

1482 @Perm(requires="pure(this) in alive",
1483 ensures="pure(this) in alive")
1484 public String getVariable() {
1485  return null;
1486 }
1487 @Perm(requires="pure(this) in alive",
1488 ensures="pure(this) in alive")
1489 public String getValue() {
1490  return null;
1491 }

1493 }ENDOFCLASS

1495 @ClassStates({@State(name = "alive")})

1497 class EDim {
1498 @Perm(ensures="unique(this) in alive")
1499 EDim() {   }

1501 @Perm(requires="pure(this) in alive",
1502 ensures="pure(this) in alive")
1503 public ArrayList<String> getValues() {
1504  return null;
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 }

1520 }ENDOFCLASS

1522 @ClassStates({@State(name = "alive")})

1524 class specificationStruct {
1525 @Perm(ensures="unique(this) in alive")
1526 specificationStruct() {   }
```

```
}ENDOFCLASS

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

class Clause {
@Perm(ensures="unique(this) in alive")
Clause() {   }


}ENDOFCLASS

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

class Signature {
@Perm(ensures="unique(this) in alive")
Signature() {   }


}ENDOFCLASS

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

class MethodFindVisitor {
@Perm(ensures="unique(this) in alive")
MethodFindVisitor() {   }

@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public boolean visit(MethodDeclaration methodDeclaration) {
 return 0;
}

}ENDOFCLASS

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

class GAPHandler {
@Perm(ensures="unique(this) in alive")
GAPHandler() {   }


public void addHandlerListener(IHandlerListener handlerListener) {
}

public void dispose() {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public Object execute(ExecutionEvent event) {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
private void extractSettings(ExecutionEvent event) {
}

public boolean isHandled() {
 return 0;
}

public void removeHandlerListener(IHandlerListener handlerListener) {
}

}ENDOFCLASS

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

class GAPIFileAction {
@Perm(ensures="unique(this) in alive")
GAPIFileAction() {   }

@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public void selectionChanged(IAction action, ISelection selection) {
}

public void setActivePart(IAction action, IWorkbenchPart targetPart) {
}
@Perm(requires="full(this) in alive",
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  }
```

```
}ENDOFCLASS

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

class DFA7 {
@Perm(ensures="unique(this) in alive")
DFA7() {   }


public String getDescription() {
 return null;
}

}ENDOFCLASS

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

class EAPTypeState {
@Perm(ensures="unique(this) in alive")
EAPTypeState() {   }

@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public void setAP(String str) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public String getAP(String str) {
 return null;
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public void setTS(String str) {
}
@Perm(requires="full(this) in alive",
ensures="full(this) in alive")
public String getTS() {
 return null;
}

}ENDOFCLASS
```