

# AST Splitting Rule Specification

```
-----
Input AST  =   Root [Method [Modifier [t*], Return [t], Name [t],
(Params [(Var [Type[t], Name [t]])+] ,)? Body [StmtT*]]
-----
```

```

StmtT      =   SimpleT | BlockT
OvT        =   Root [MethSig [MethBody [(StmtsT_r | BlockT_r)* ]]]
SigT       =   MethSig [Modifier [t*], Return [t], Name [t], (Params
[(Var [Type[t], Name [t]])+]?)
StmtsT     =   StatementsBlock [SimpleT +]
BlockT     =   IfT | ForT | WhileT | DoWhileT | SwitchT | LabelT |
SynchT | TryWithResourceT | TryT
SimpleT    =   VariableDeclaration | Expression | ReturnStatement
StmtsT_r   =   rootOf(StmtsT)
BlockT_r   =   rootOf(BlockT)

```

```

IfT        =   If [Cond[Expression], Body[StmtT +] (,Else[StmtT +])?]
ForT       =   for[forControl,Body[StmtT +]]|for-each[enForControl,
Body[StmtT +]]
forControl =   (Init[VariableDeclaration | Expression])? ,
(Cond[Expression])?, (Update[Expression+])?
enForControl= Modifier[t*] ,Type[t] , Name[t],In [t]
WhileT     =   While[Cond [Expression], Body [StmtT +] ]
DoWhileT   =   DoWhile[Cond [Expression], Body [StmtT +] ]
LabelT     =   Lable[Name(t), StmtT ]
SynchT     =   synchronized[Expression, Body[StmtT +]]
SwitchT    =   Switch[Expression, StmtGroup* switchLabel*]
StmtGroup  =   switchLabel+ ,blockStatement+
switchLabel = case[Body[StmtT +]] | default[Body[StmtT +]]
TryWithResourceT = Try[Resources,Body[SimpleT+],(Catch[StmtT+])* ,
(Finally[StmtT+])?]
Resources  =   [Resource ,( Resource)*]
Resource   =   Modifier[t]* ,Type [t], Name[t], INIT['=' Expression]
Try        =   try[Body[SimpleT+], Catch[StmtT+],(Finally[StmtT+])?]
VariableDeclaration = Modifier[t*] ,Type[t] , Name[t], INIT['='
Expression]
ReturnStatement      = Return expression
t                    =   identifier

```

```

expression =
| this| super | t | break | continue
| expression, bop='.'
( t
| methodCall
| this
| NEW nonWildcardTypeArguments? innerCreator
| super superSuffix
| explicitGenericInvocation
)
| expression '[' expression ']'
| methodCall
| NEW creator
| Type[t],expression

```

```

| expression, postfix=('++' | '--')
| prefix=('+' | '-' | '++' | '--') ,expression
| prefix=('~' | '!'), expression
| expression, bop=('*' | '/' | '%') expression
| expression, bop=('+' | '-') expression
| expression , ('<' '<' | '>' '>' '>' | '>' '>') ,expression
| expression, bop=('<=' | '>=' | '>' | '<'), expression
| expression ,bop=INSTANCEOF , Type[t]
| expression ,bop=('==' | '!=') ,expression
| expression, bop='&', expression
| expression, bop='^', expression
| expression ,bop='|' ,expression
| expression, bop='&&' ,expression
| expression, bop='||', expression
bop=('=' | '+=' | '-=' | '*=' | '/=' | '&=' | '|=' | '^=' | '>>=' |
'>>>=' | '<<=' | '%=')

```

---

```

Subtrees    =    {OvT} U {SigT} U {StmtsT*} U {BlockT*}

```

---

**\*\*note:** The Body and Block are treated the same in implementation.