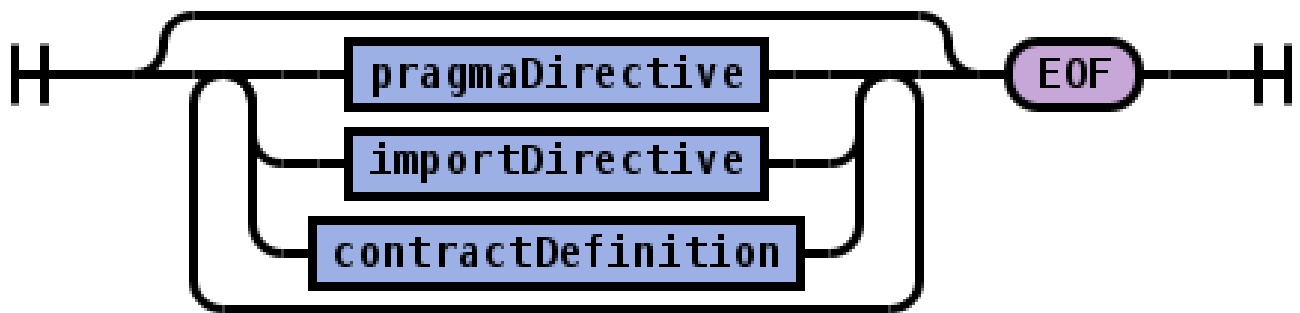


sourceUnit :



pragmaDirective :



pragmaName :



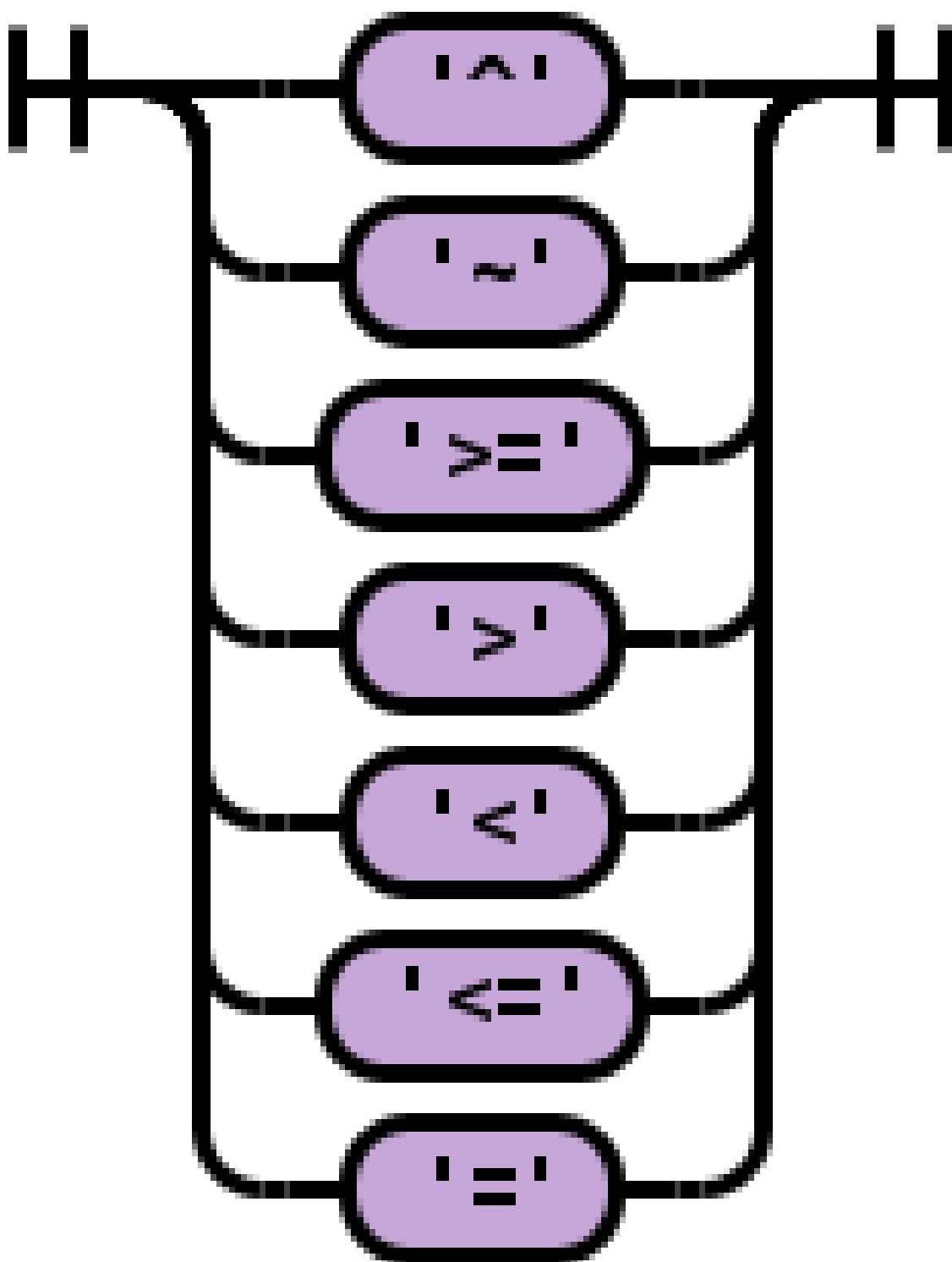
pragmaValue :



version :

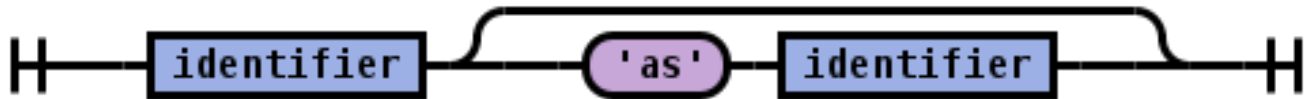


versionOperator :

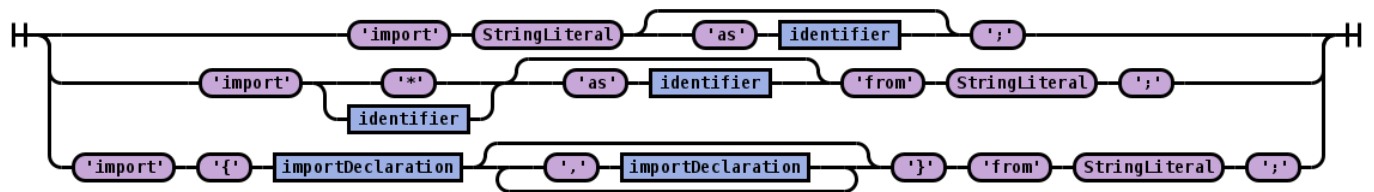


versionConstraint :

importDeclaration :



importDirective :



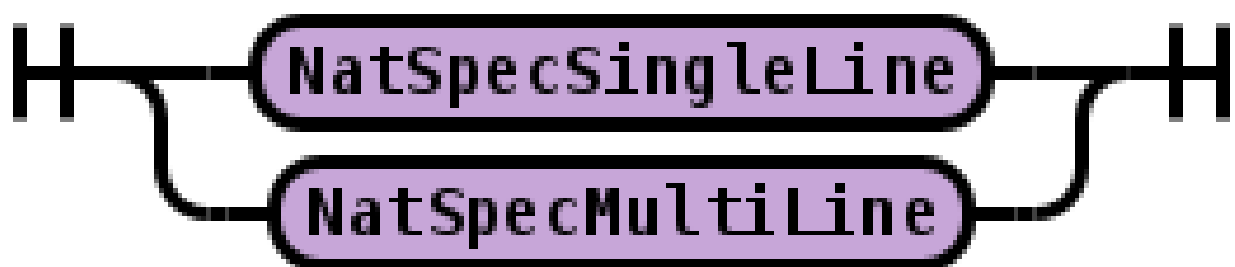
NatSpecSingleLine :



NatSpecMultiLine :

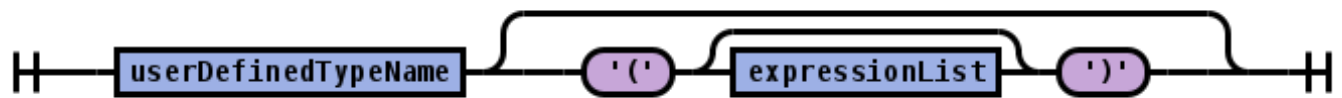
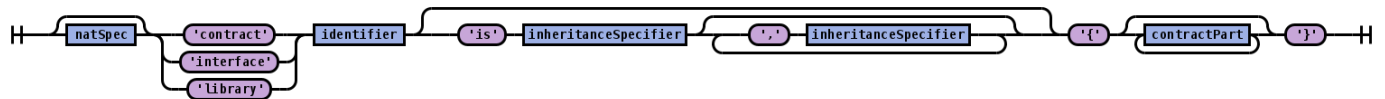


natSpec :

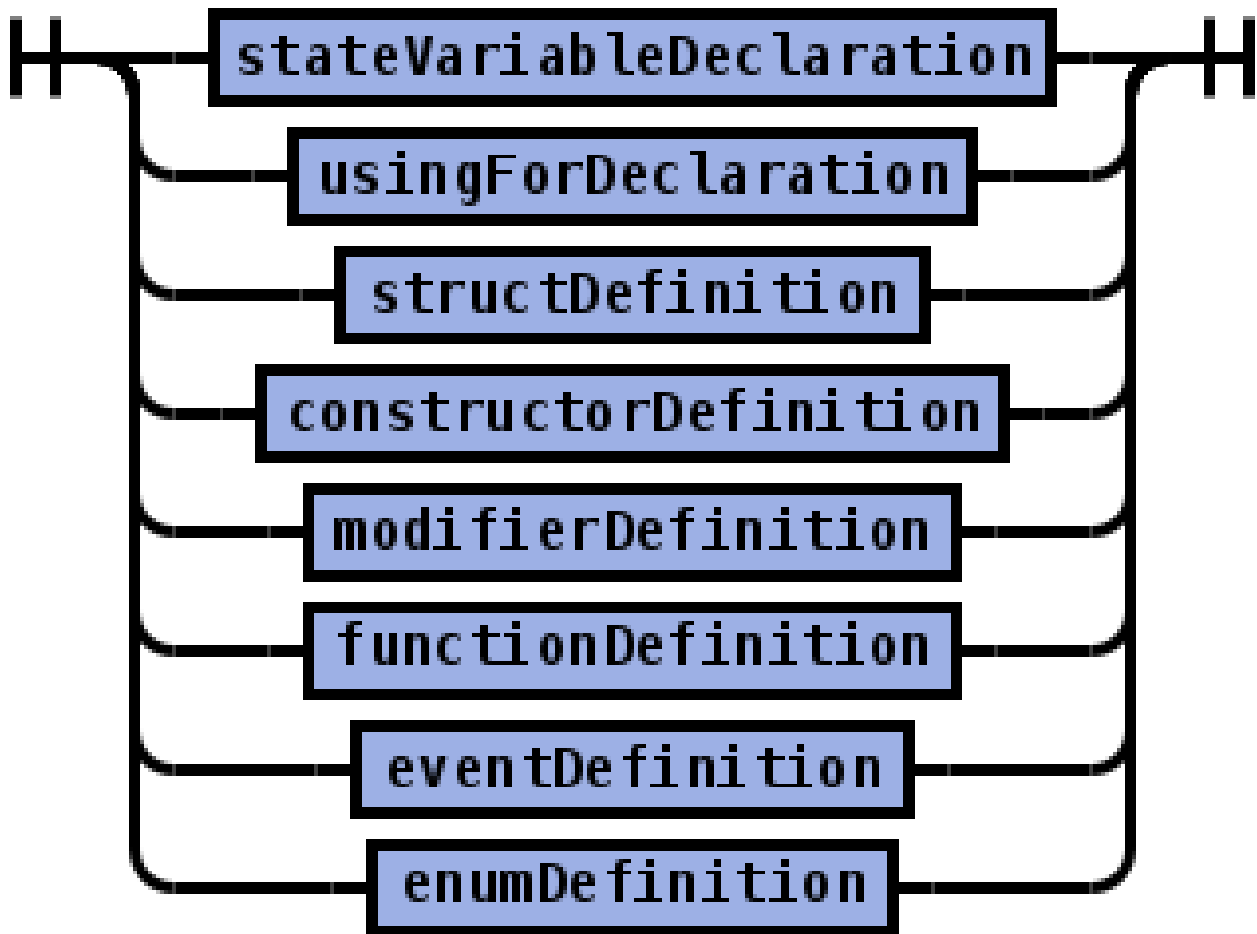


contractDefinition :

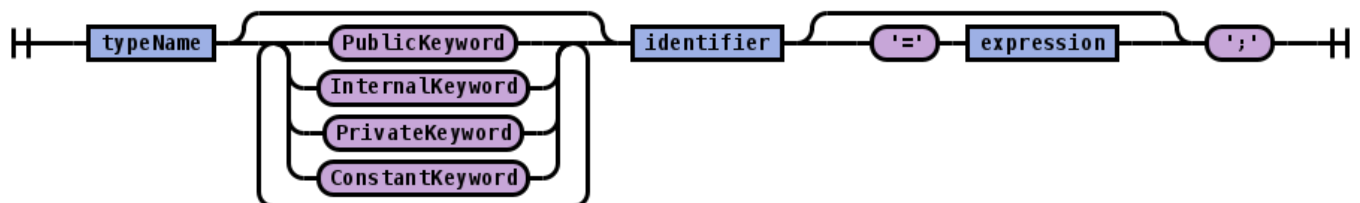
inheritanceSpecifier :



contractPart :

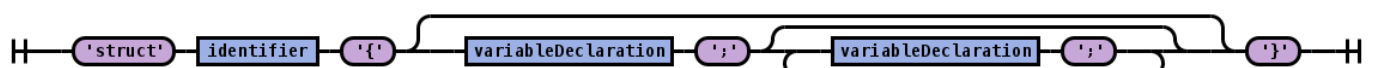


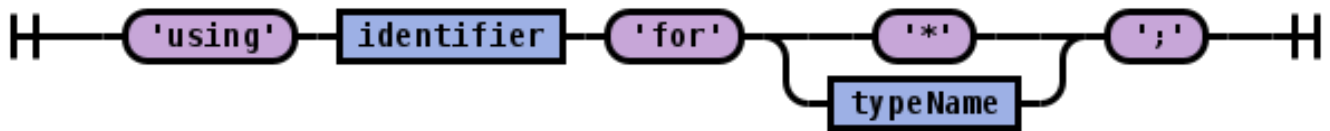
stateVariableDeclaration :



usingForDeclaration :

structDefinition :





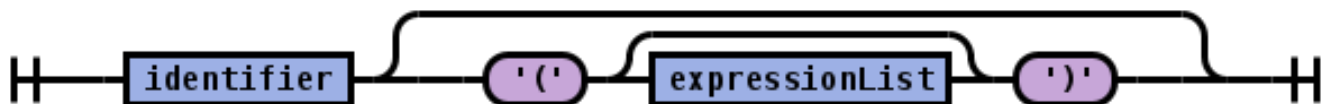
constructorDefinition :



modifierDefinition :



modifierInvocation :



functionDefinition :

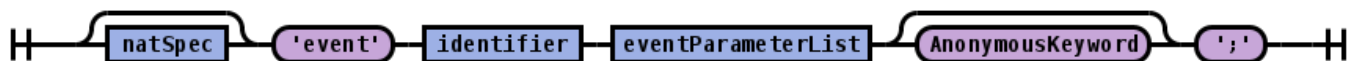


returnParameters :

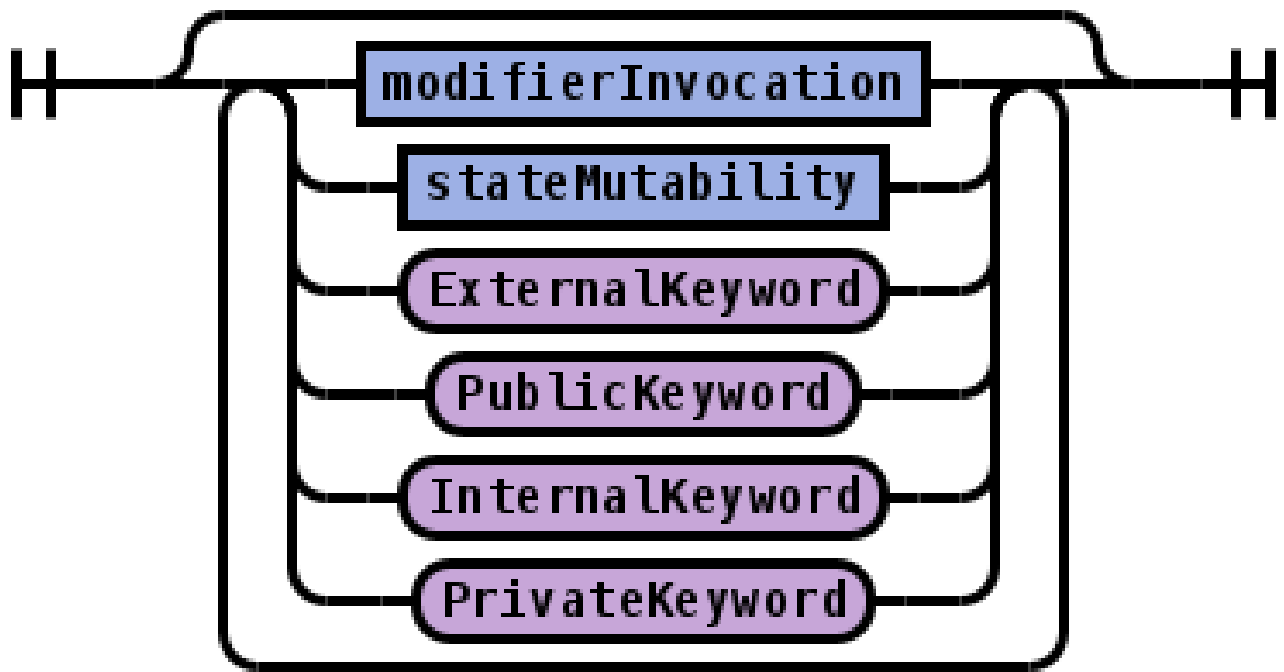


modifierList :

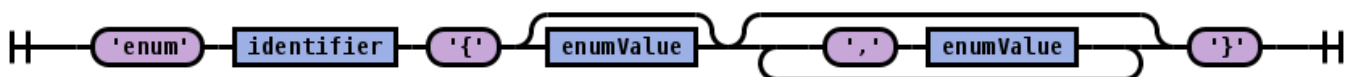
eventDefinition :



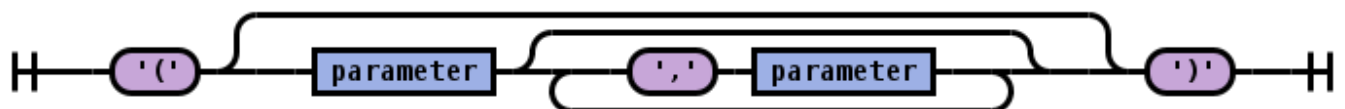
enumValue :



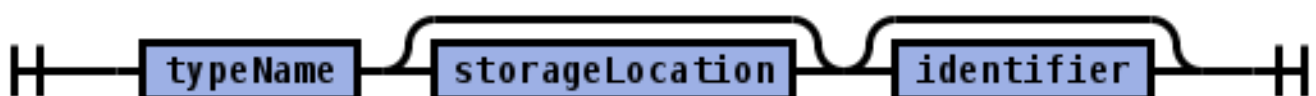
enumDefinition :



parameterList :

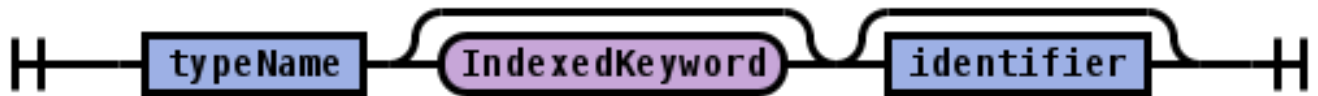
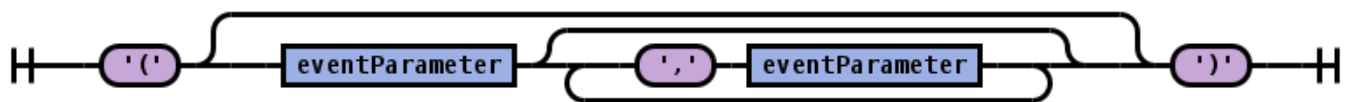


parameter :

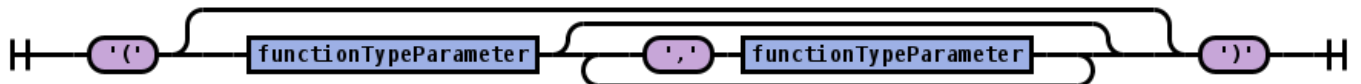


eventParameterList :

eventParameter :



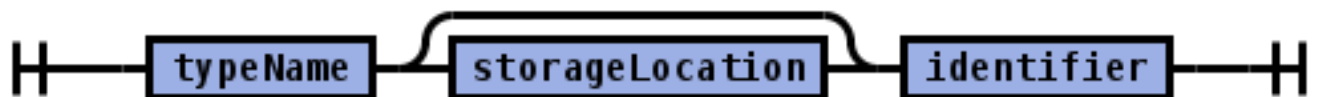
functionTypeParameterList :



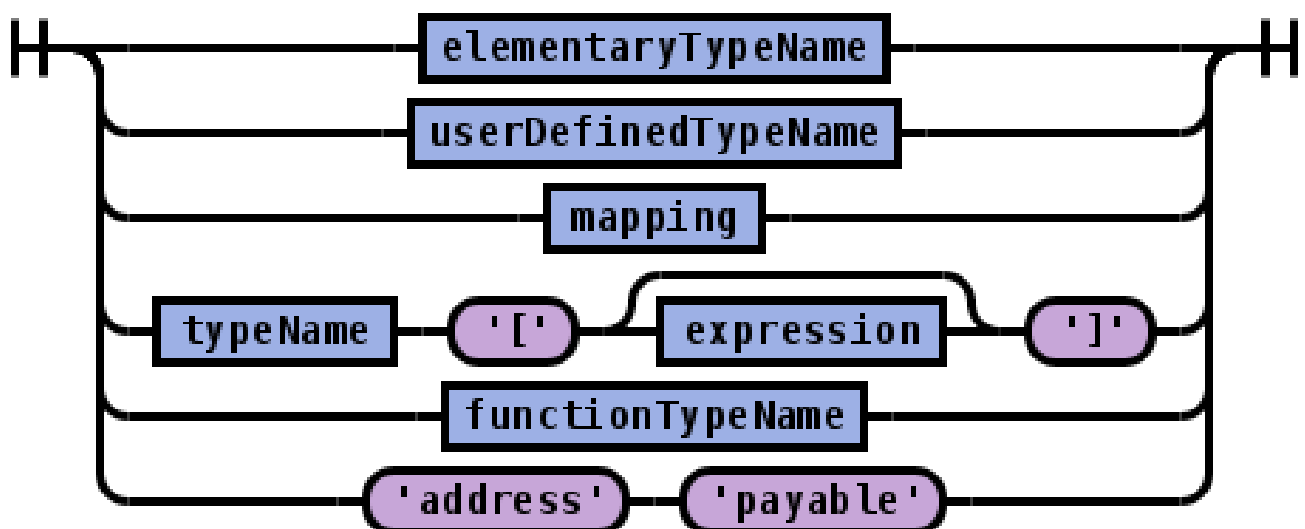
functionTypeParameter :



variableDeclaration :

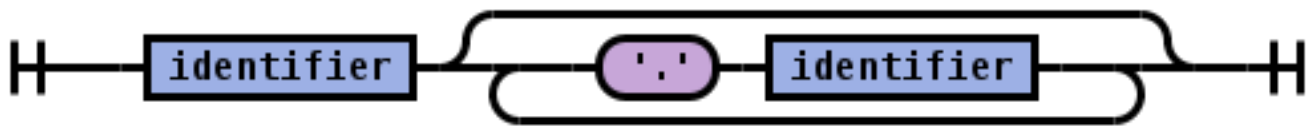


typeName :

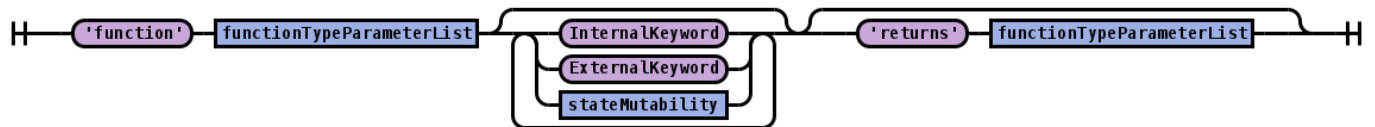


userDefinedTypeName :

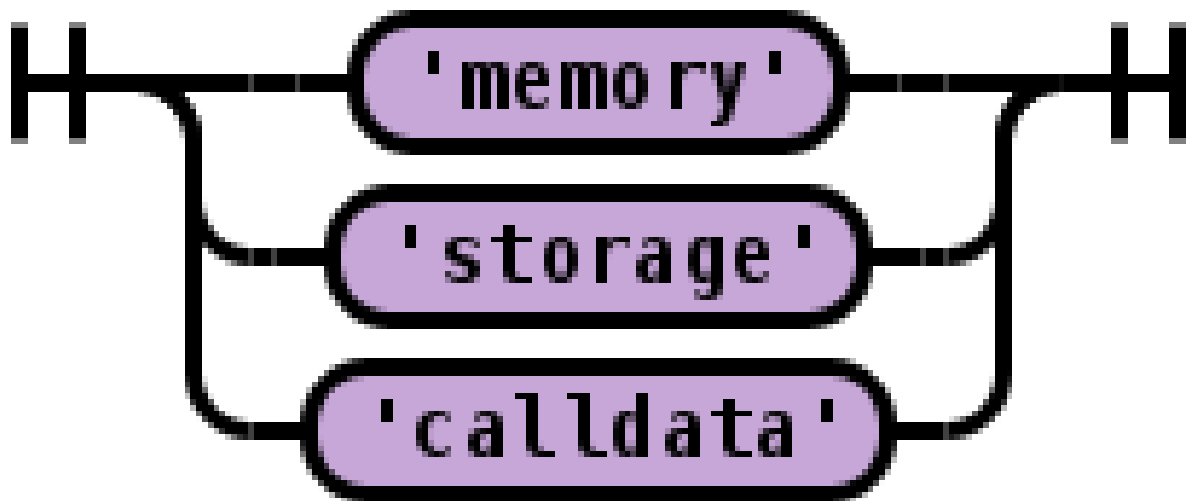
mapping :



functionTypeName :



storageLocation :

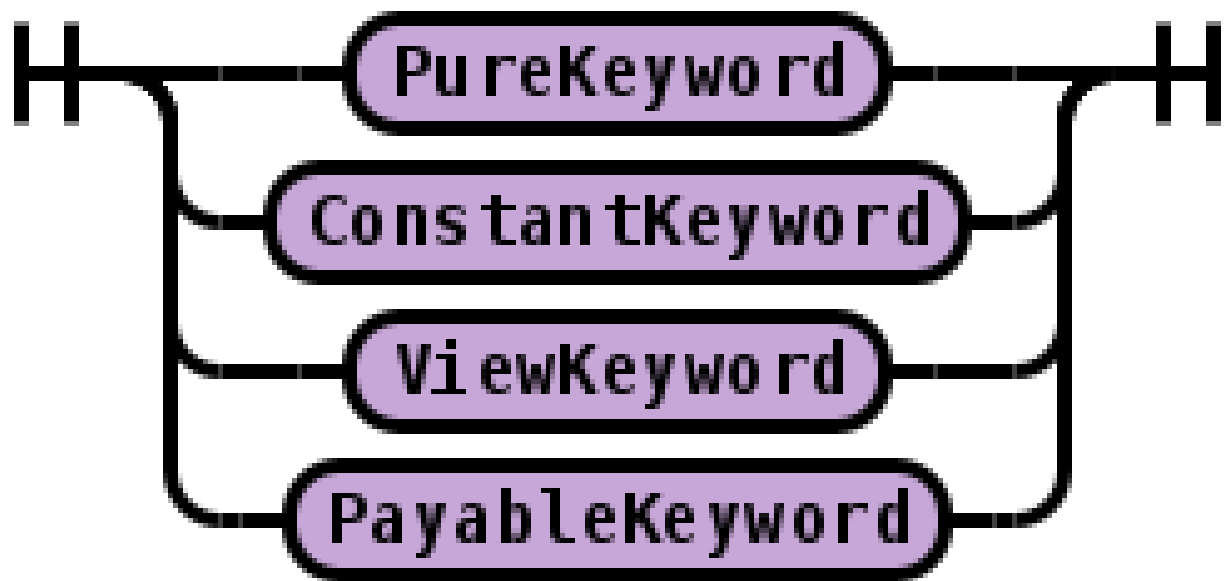


stateMutability :

block :



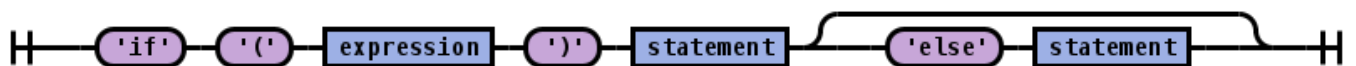
statement :



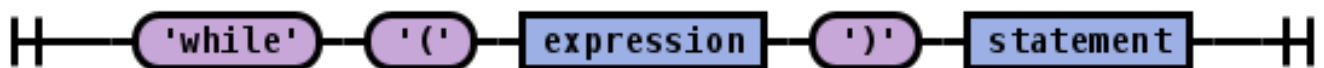
expressionStatement :



ifStatement :



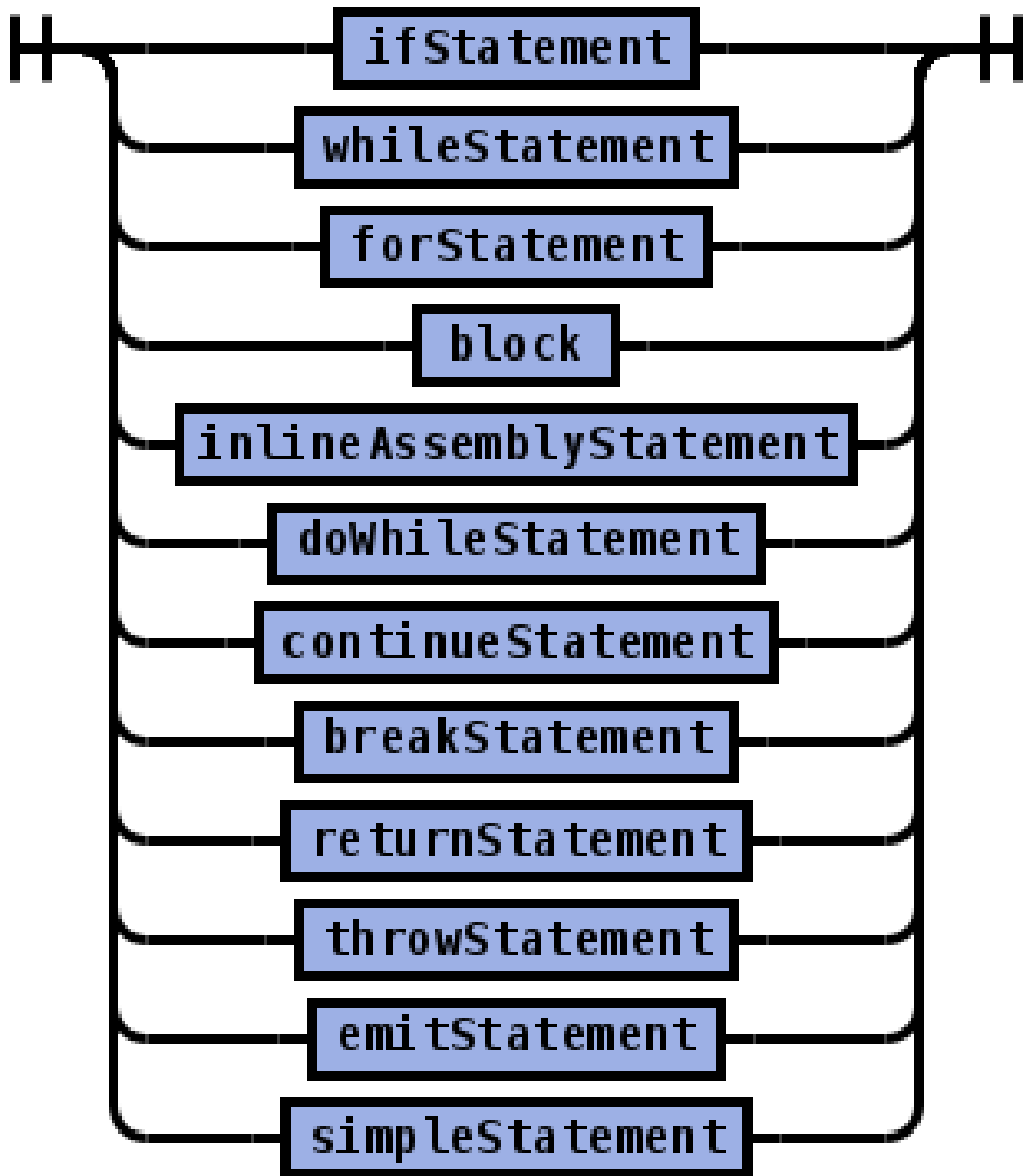
whileStatement :



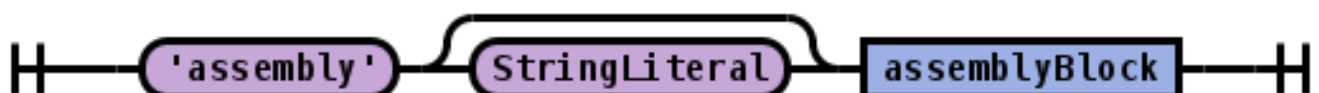
simpleStatement :



forStatement :



inlineAssemblyStatement :



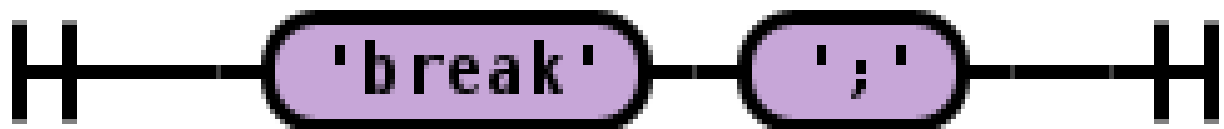
doWhileStatement :



continueStatement :



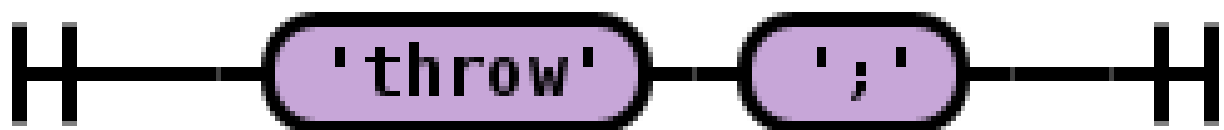
breakStatement :



returnStatement :



throwStatement :

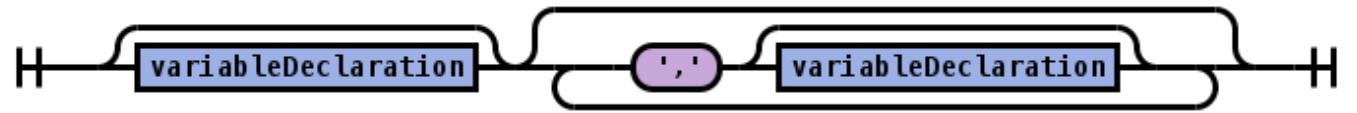
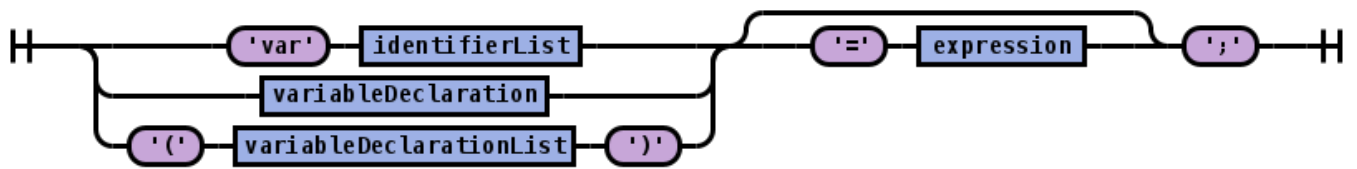


emitStatement :

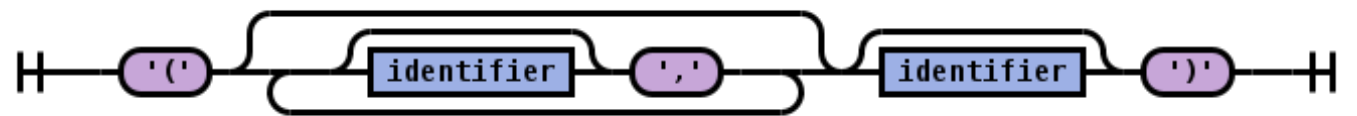


variableDeclarationStatement :

variableDeclarationList :

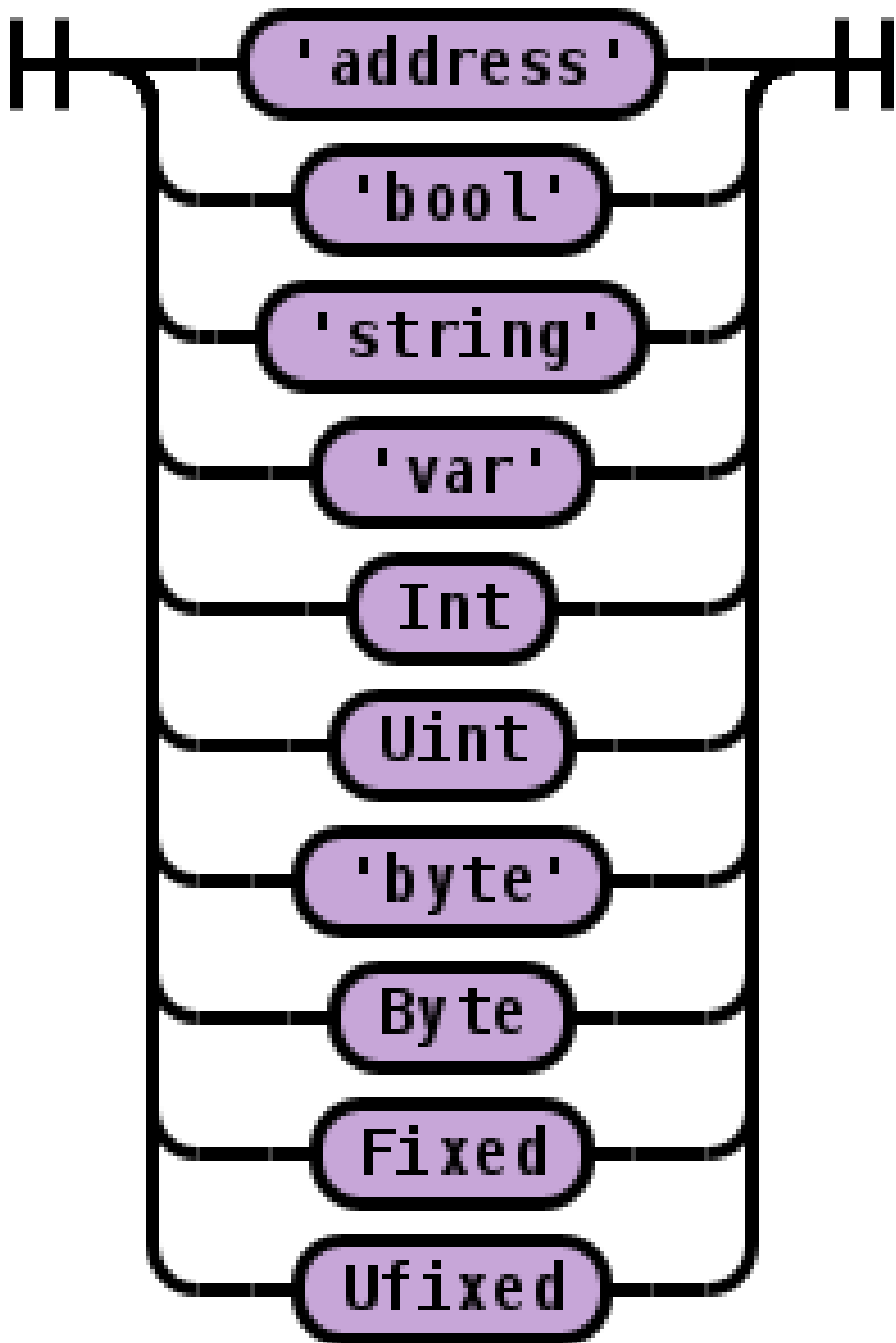


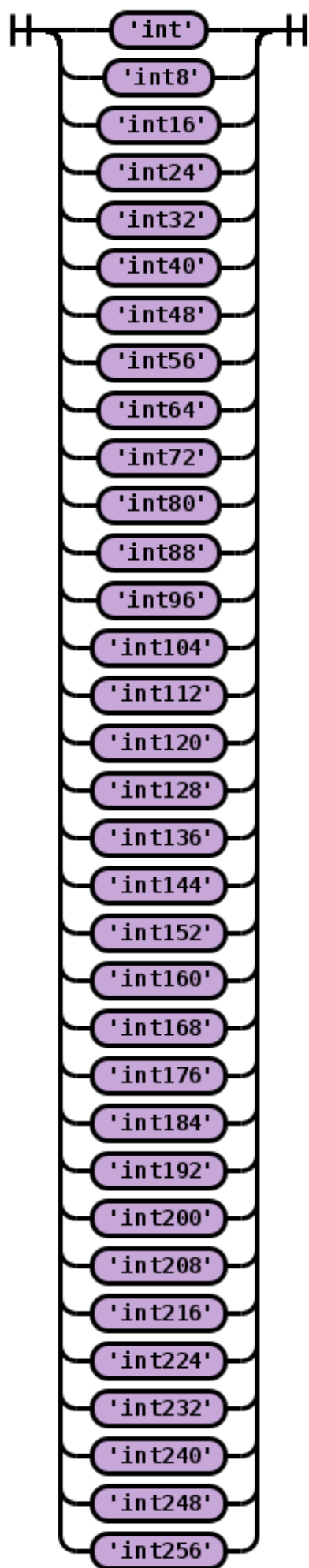
identifierList :



elementaryTypeName :

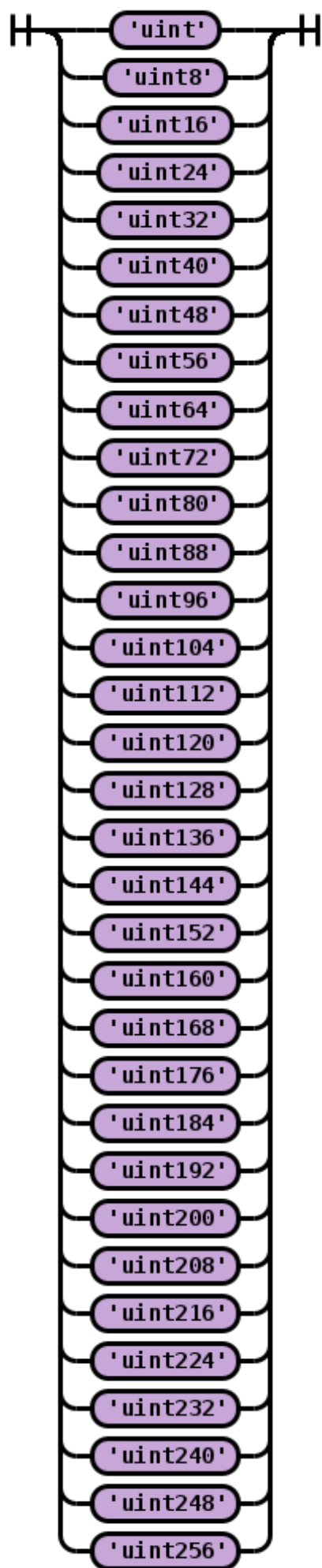
Int :

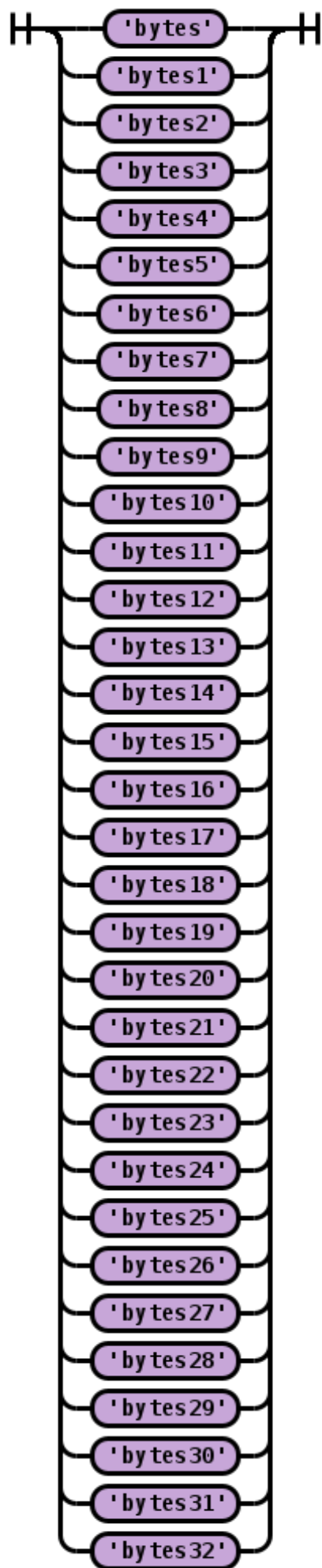




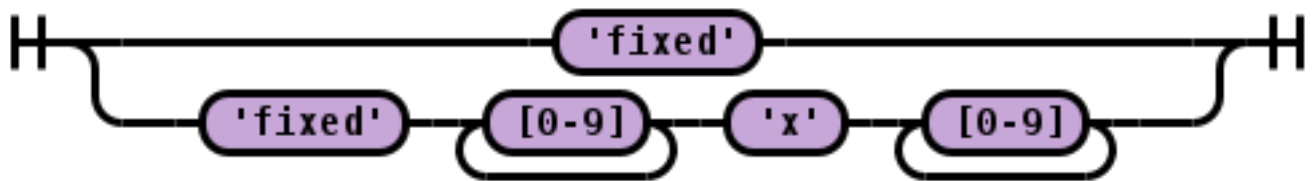
UInt :

Byte :

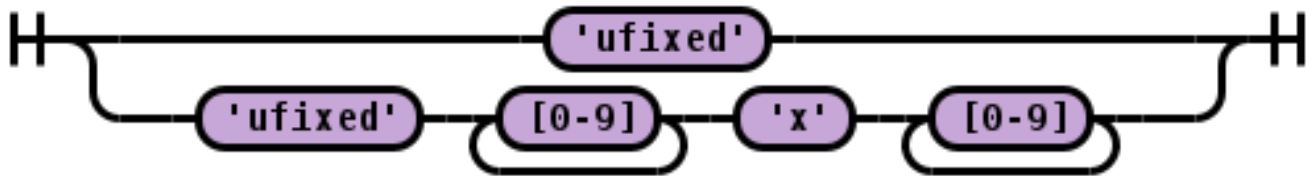




Fixed :

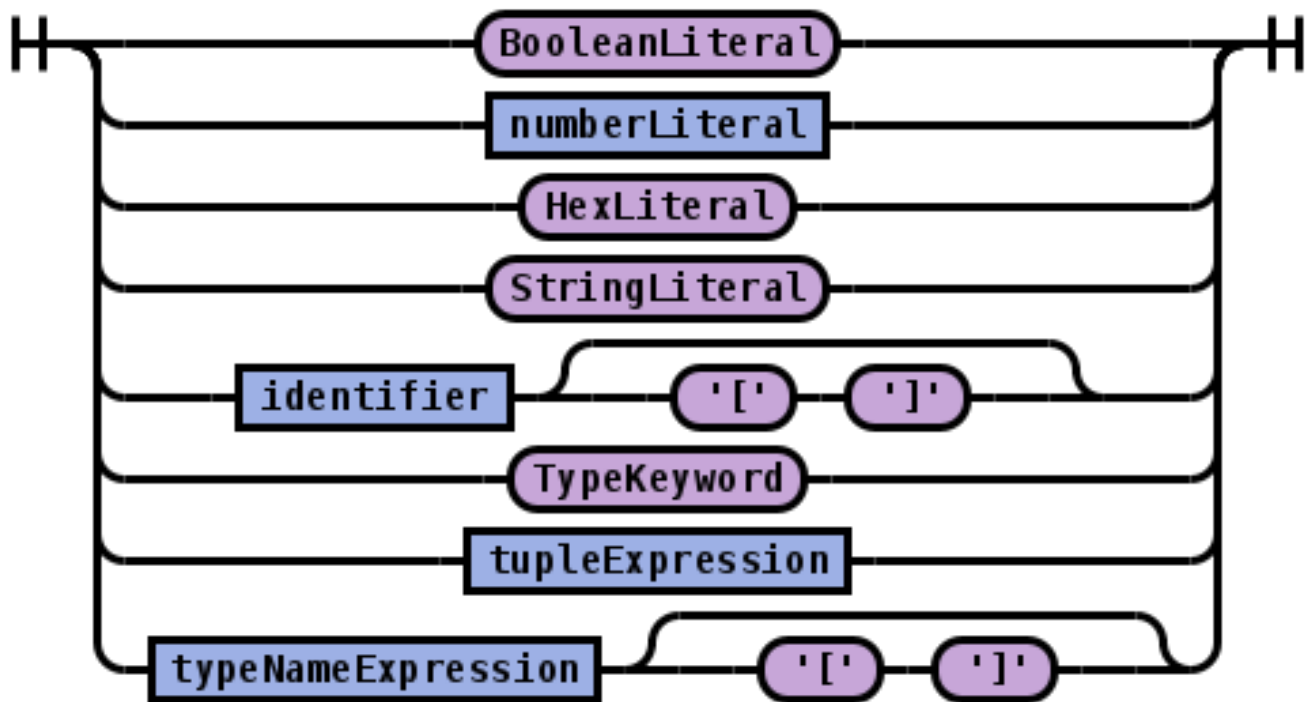


Ufixed :

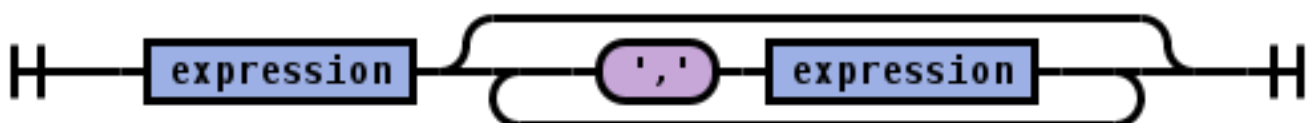


expression :

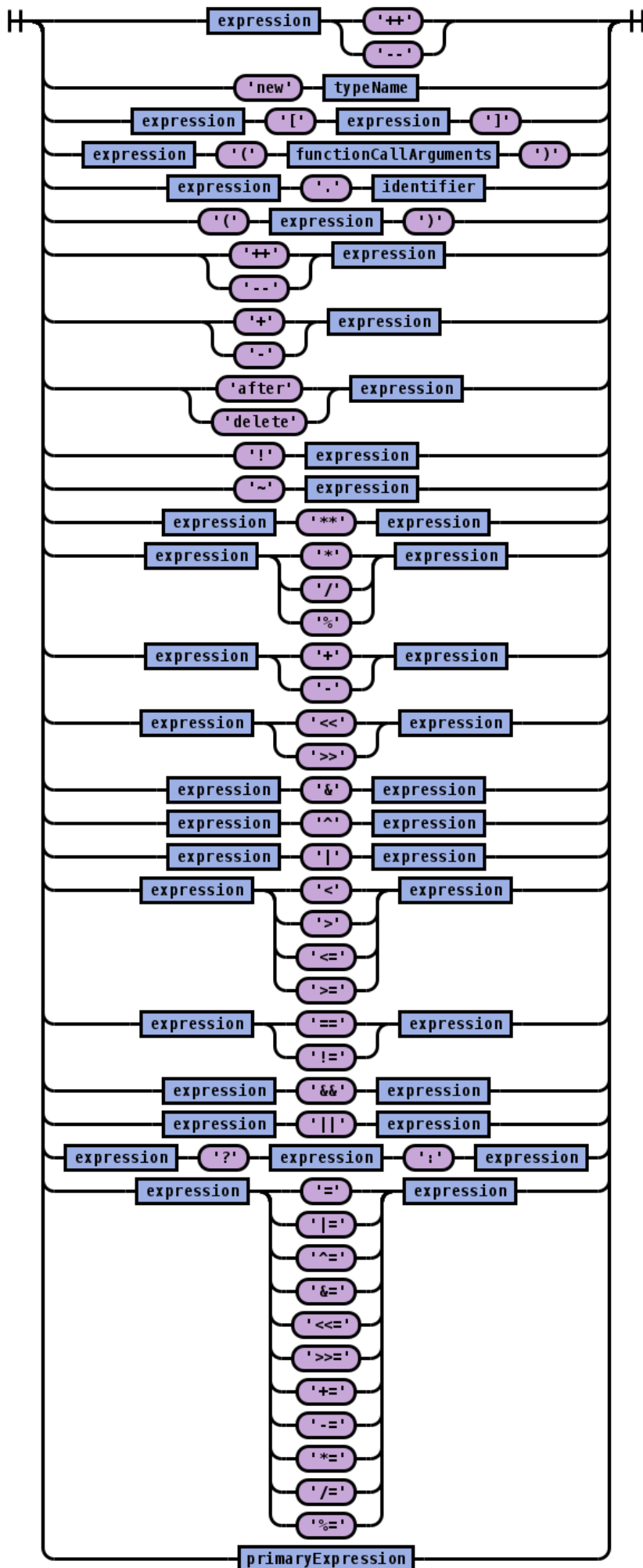
primaryExpression :

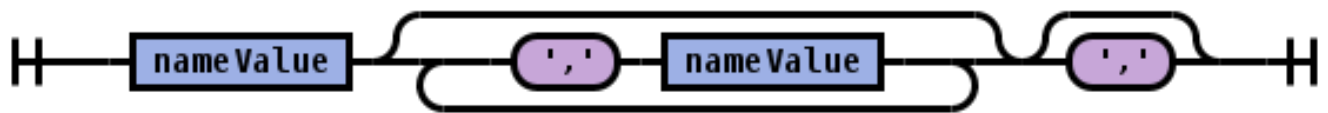


expressionList :



nameValueList :

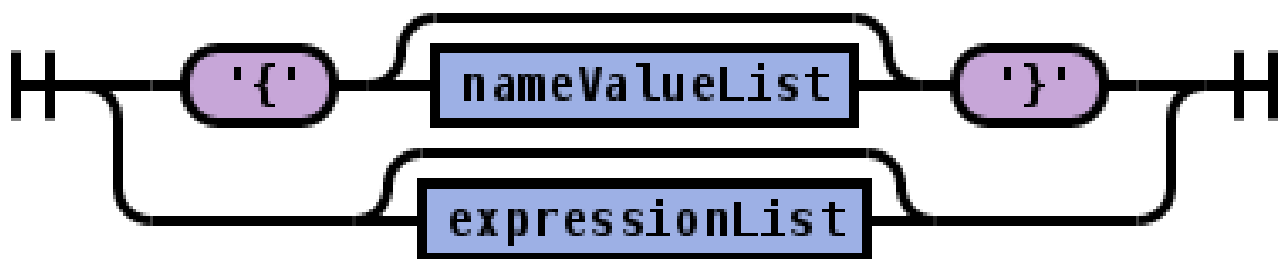




`nameValue :`



`functionCallArguments :`



`functionCall :`



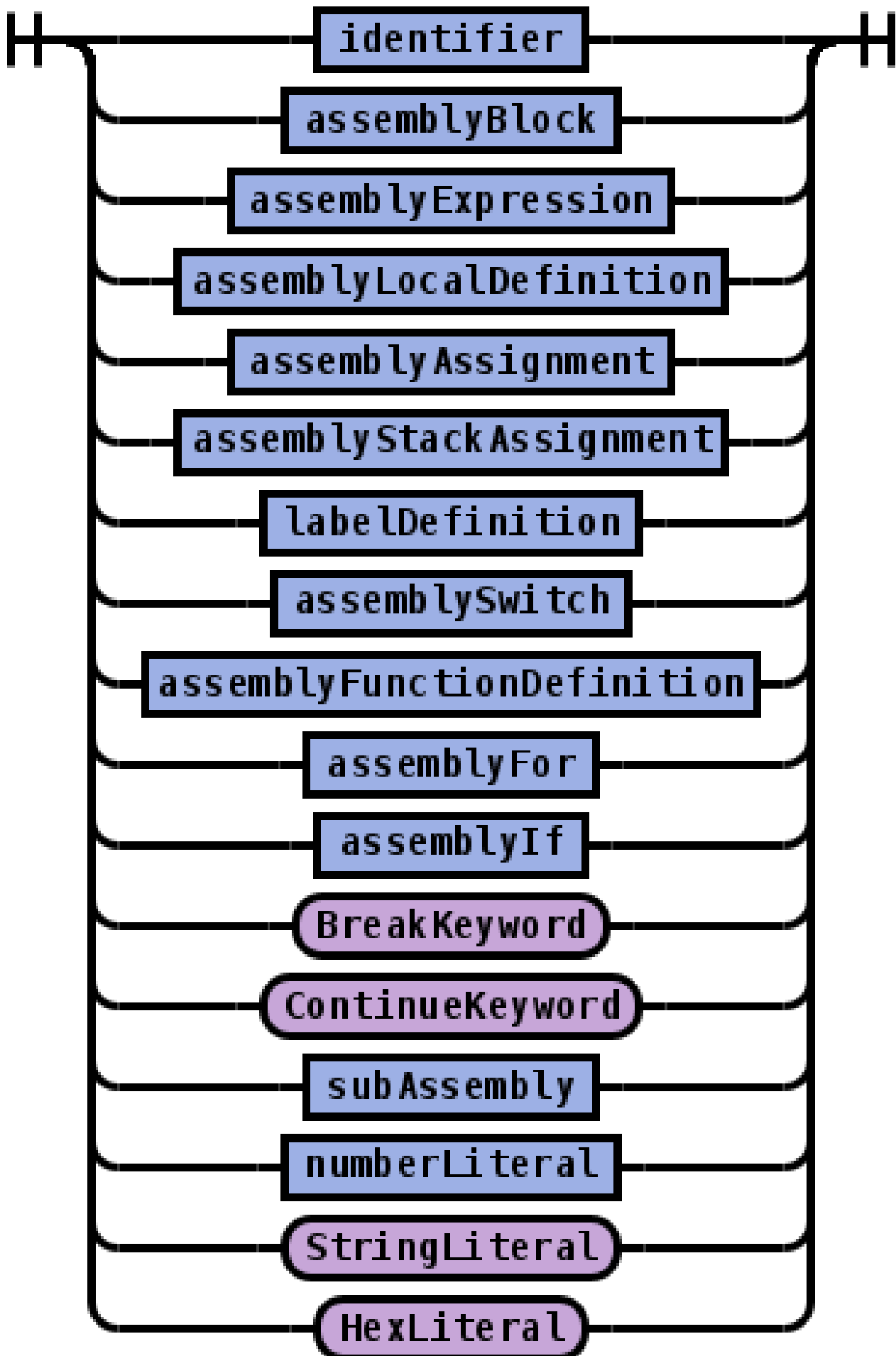
`assemblyBlock :`



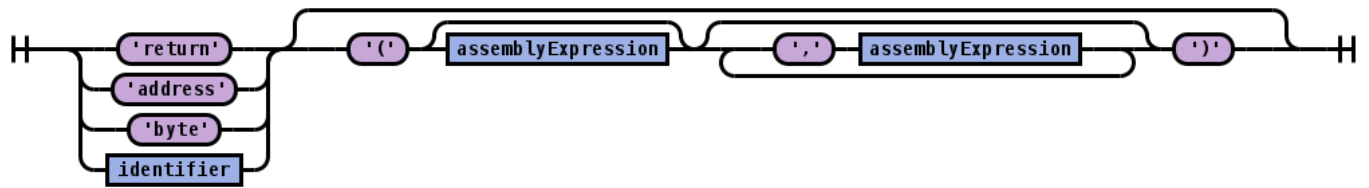
`assemblyItem :`

`assemblyExpression :`





assemblyCall :



assemblyLocalDefinition :



assemblyAssignment :



assemblyIdentifierOrList :



assemblyIdentifierList :



assemblyStackAssignment :



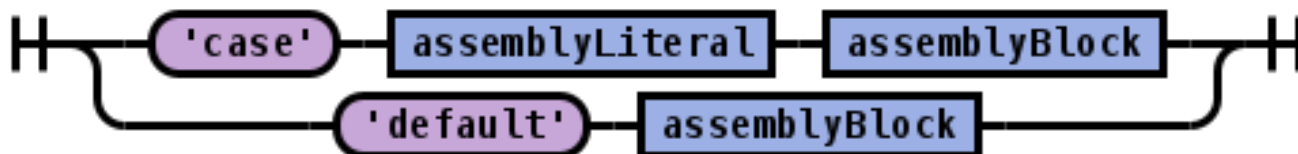
labelDefinition :



assemblySwitch :



assemblyCase :



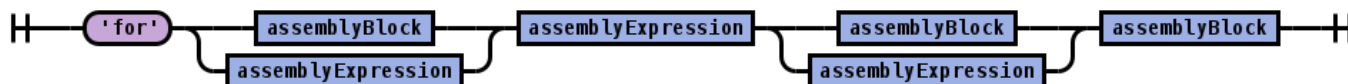
assemblyFunctionDefinition :



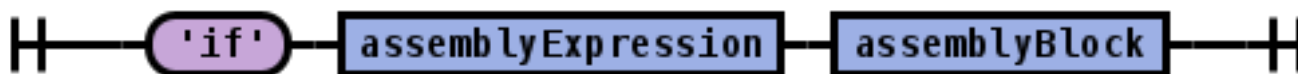
assemblyFunctionReturns :



assemblyFor :



assemblyIf :

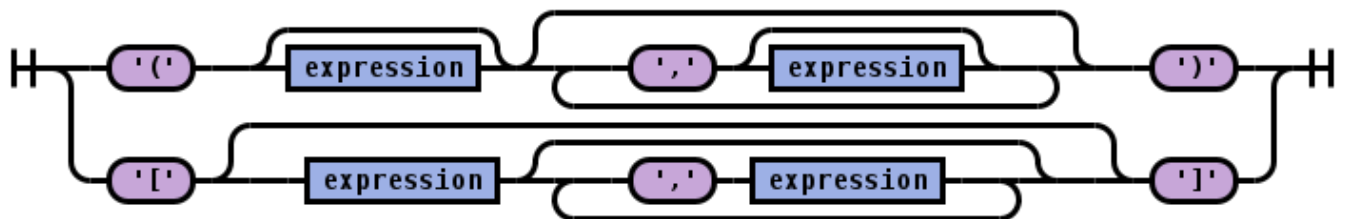
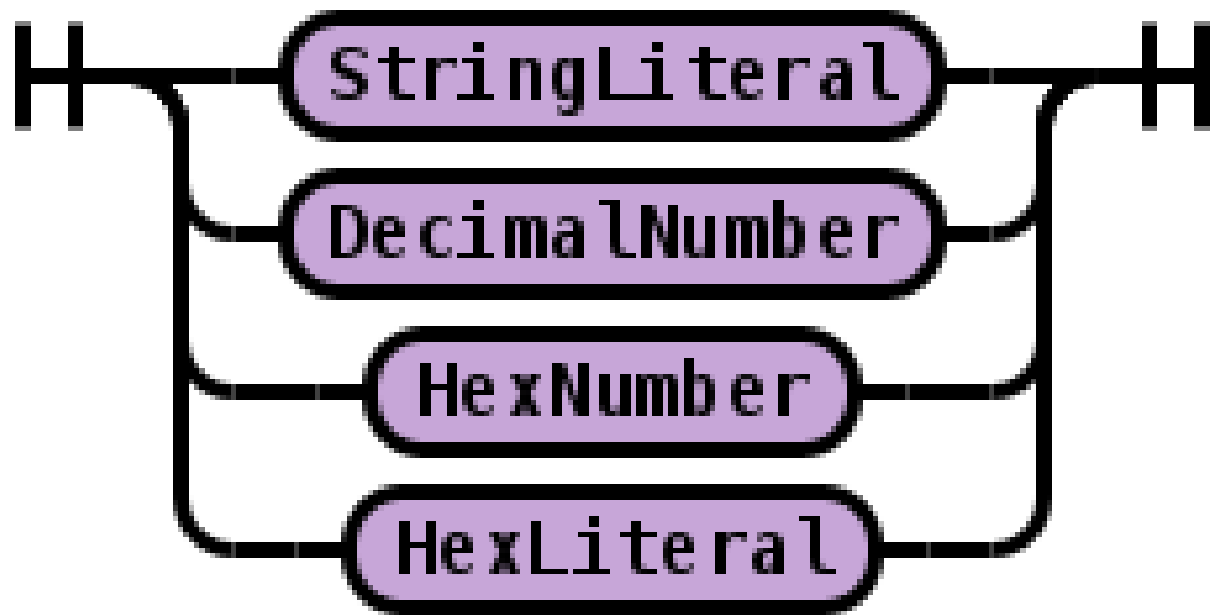


assemblyLiteral :

subAssembly :



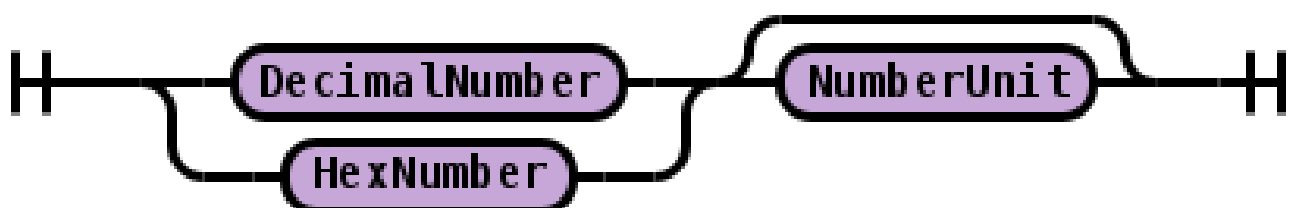
tupleExpression :



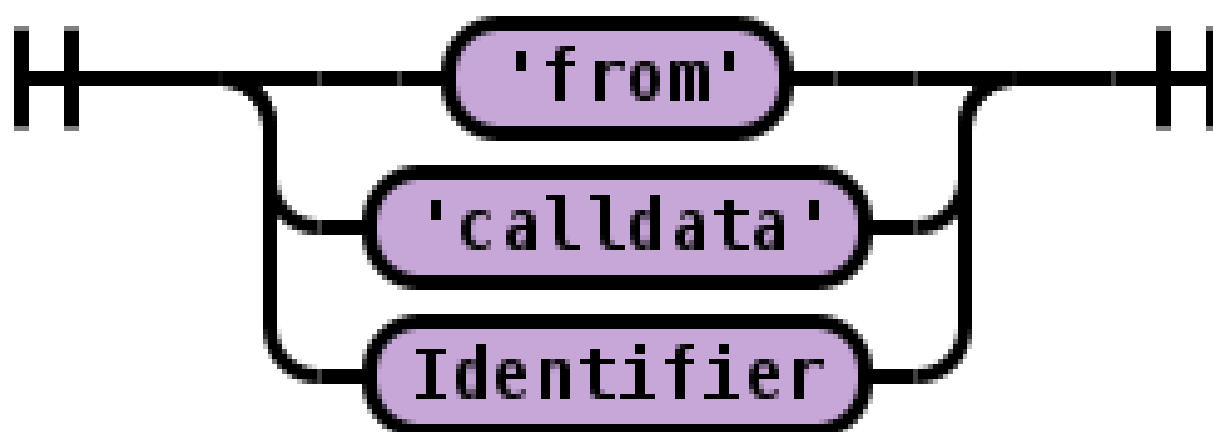
typeNameExpression :



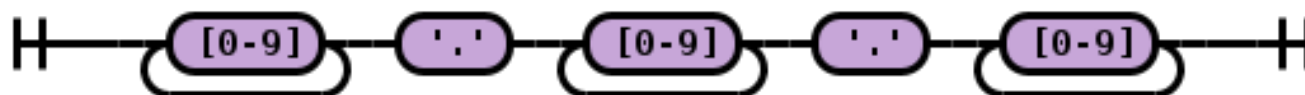
numberLiteral :



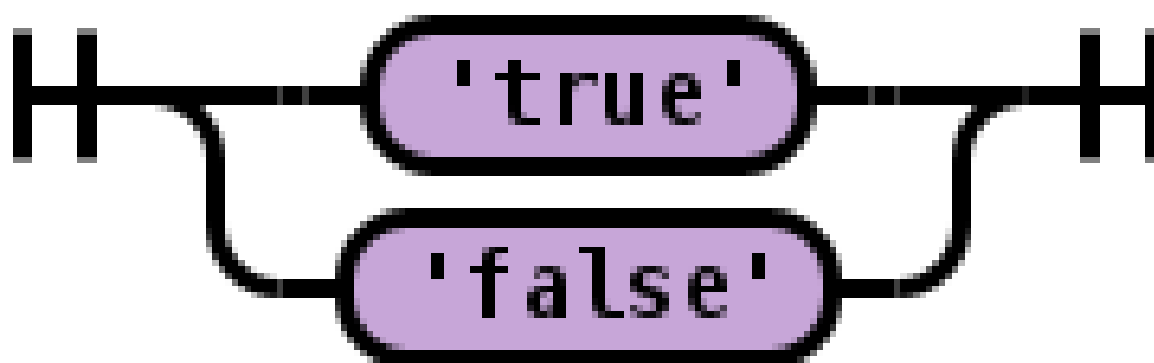
identifier :



VersionLiteral :



BooleanLiteral :

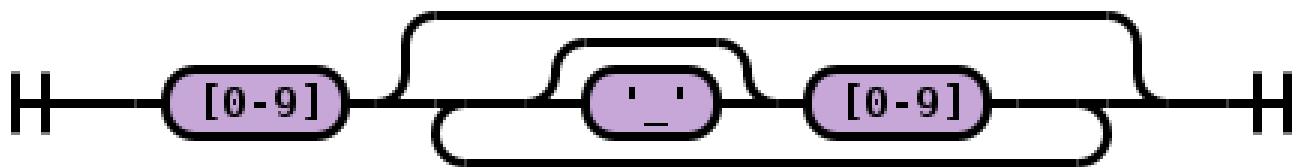


DecimalNumber :



DecimalDigits :

HexNumber :

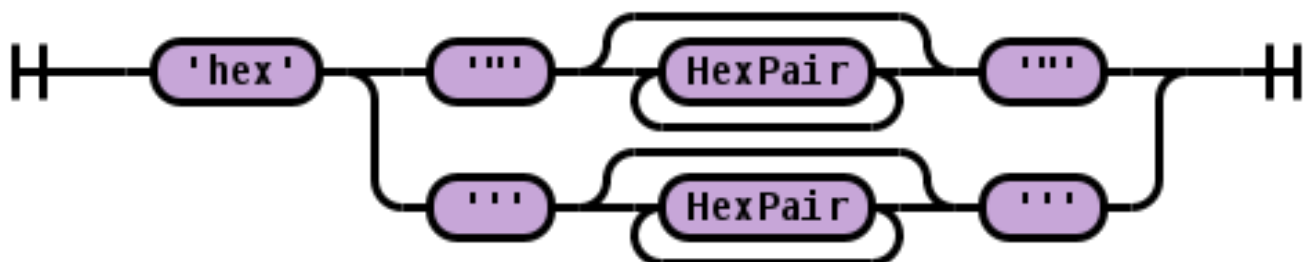


HexDigits :



NumberUnit :

HexLiteral :

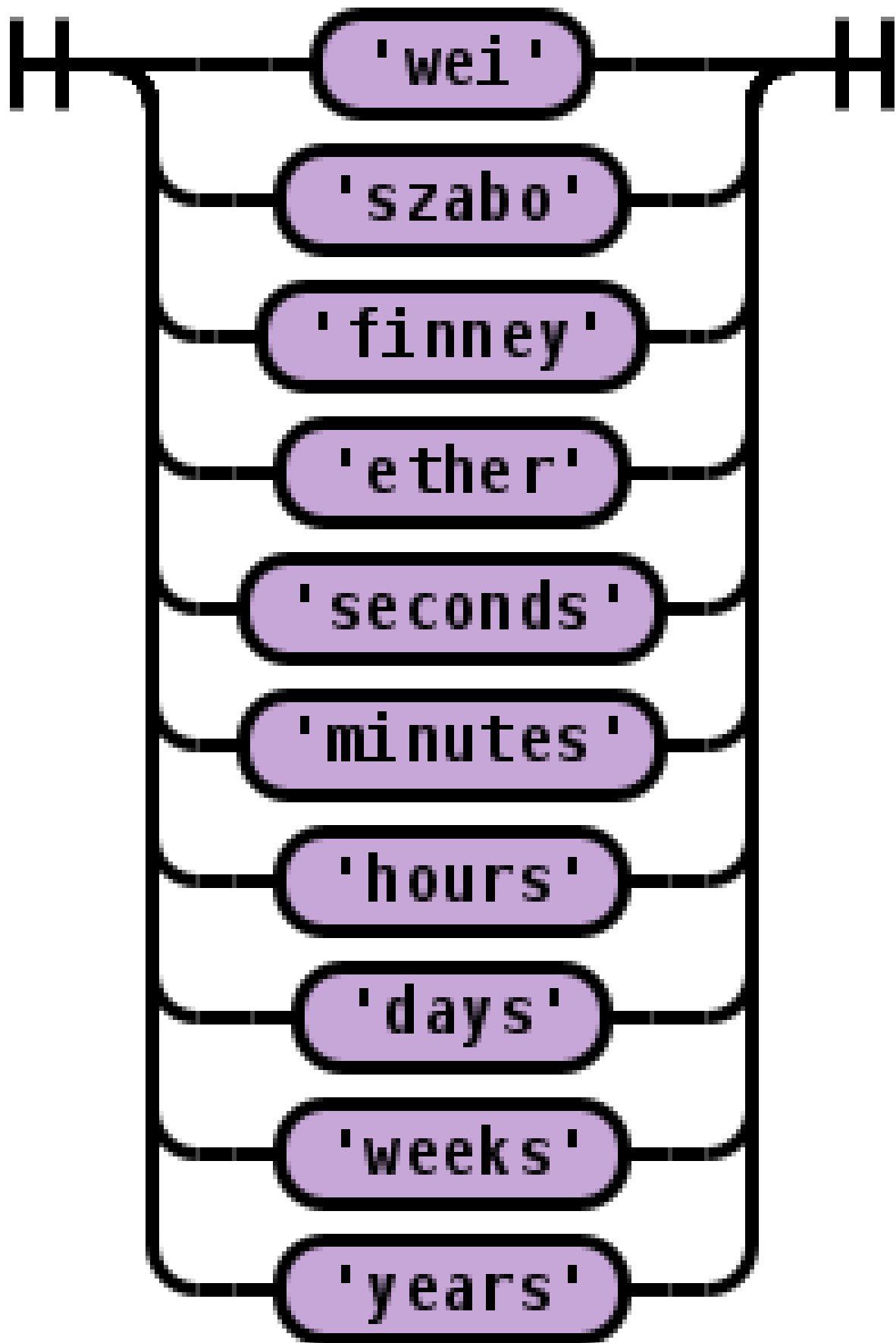


HexPair :



HexCharacter :





ReservedKeyword :

AnonymousKeyword :

A diagram showing the structure of an anonymous keyword. It consists of a horizontal line with two vertical bars at each end, resembling a double-headed arrow. In the center of this line is a purple rounded rectangle with a black border, containing the text 'anonymous' in a monospaced font.

BreakKeyword :

A diagram showing the structure of a break keyword. It consists of a horizontal line with two vertical bars at each end, resembling a double-headed arrow. In the center of this line is a purple rounded rectangle with a black border, containing the text 'break' in a monospaced font.

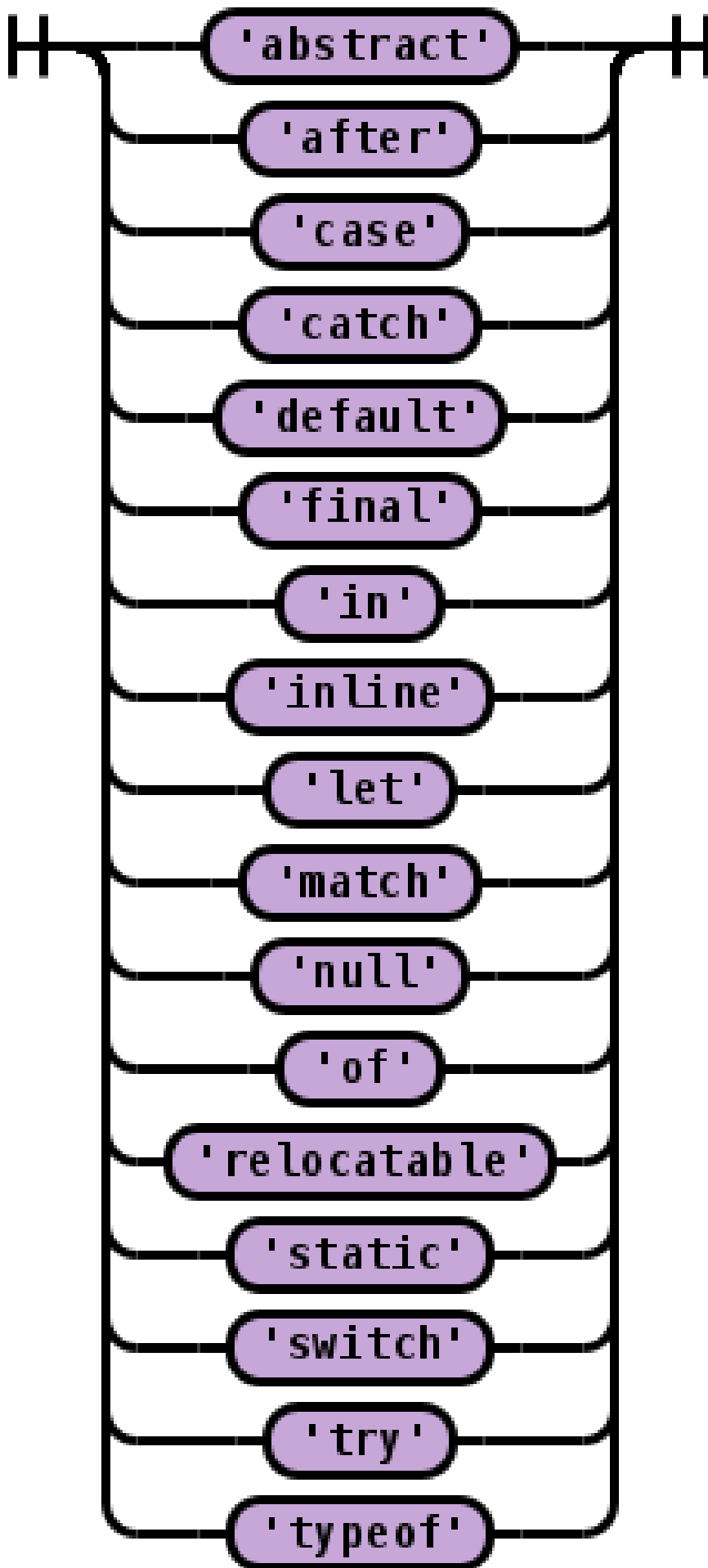
ConstantKeyword :

A diagram showing the structure of a constant keyword. It consists of a horizontal line with two vertical bars at each end, resembling a double-headed arrow. In the center of this line is a purple rounded rectangle with a black border, containing the text 'constant' in a monospaced font.

ContinueKeyword :

A diagram showing the structure of a continue keyword. It consists of a horizontal line with two vertical bars at each end, resembling a double-headed arrow. In the center of this line is a purple rounded rectangle with a black border, containing the text 'continue' in a monospaced font.

ExternalKeyword :





IndexedKeyword :



InternalKeyword :



PayableKeyword :



PrivateKeyword :

PublicKeyword :



PureKeyword :



TypeKeyword :



ViewKeyword :

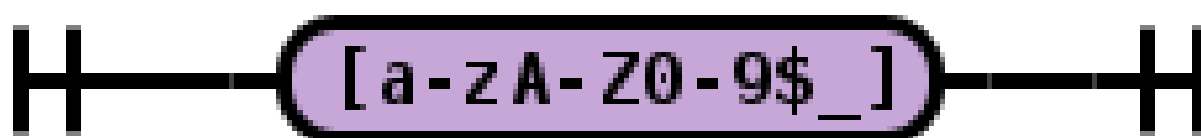
Identifier :



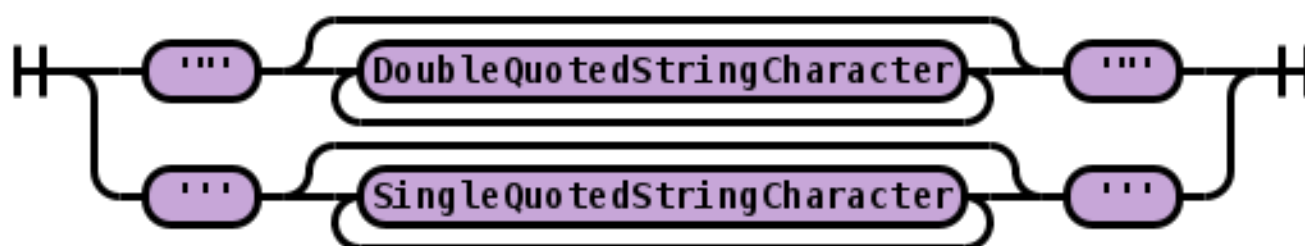
IdentifierStart :



IdentifierPart :

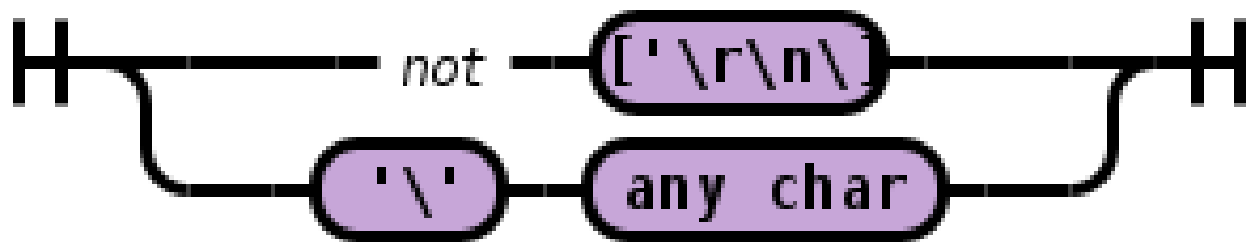
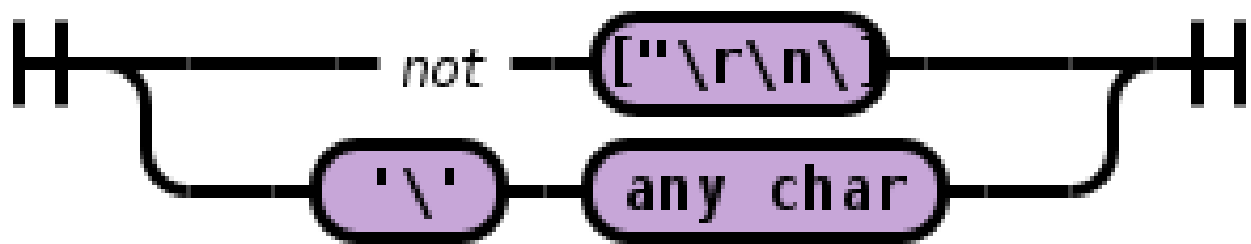


StringLiteral :



DoubleQuotedStringCharacter :

SingleQuotedStringCharacter :



WS :



COMMENT :



LINE_COMMENT :

