Chunk : Block

Block : Stats | Stats RetStat

Stats : Stat Stats | e

Stat : **‘;’**

| VarList ‘=’ ExpList

| FunctionCall

| **break**

| **do** Block **end**

**| while** Exp **do** Block **end**

| **repeat** Block **until** Exp

| GeneralIfStat

| ForStat

| **function** FuncName FuncBody

| **local** **function** Name FuncBody

| **local** NameList **‘=’** ExpList

| **local** NameList

GeneralIfStat: IfStat ElseIfStats ElseStat **end**

IfStat : **if** Exp **then** Block

ElseIfStats : ElseIfStat ElseIfStats

| e

Elseifstat : **elseif** Exp **then** Block

Elsestat : **else** Block

| e

Forstat : ForStatWithName | ForStatWithNameList

ForStatWithName : **for** Name **‘=’** Exp **‘,’** Exp **do** Block **end**

| **for** Name **‘=’** Exp **‘,’** Exp **‘,’** Exp **do** Block **end**

ForStatWithNameList: **for** NameList **in** ExpList **do** Block **end**

RetStat : return

| **return ‘;’**

| **return** ExpList

| **return** ExpList **‘;’**

FuncName : Name NameHierarchy **‘:’** Name

| Name NameHierarchy

NameHierarchy: **‘.’** Name NameHierarchy | e

VarList : Var | Var ‘,’ Varlist

Var : Name | PrefixExp **‘[‘** Exp **‘]’** | PrefixExp **‘.’** Name

NameList : Name ‘,’ Namelist | Name

ExpList : Exp ExpList| Exp

Exp : **nil**

| **false**

| **true**

| Number

| String

| **‘…’**

| FunctionDef

| PrefixExp

| TableConstructor

| Exp Binop Exp

| Unop Exp

PrefixExp : Var | FunctionCall | ‘(‘ Exp ‘)’

FunctionCall : PrefixExp Args | PrefixExp **‘:’** Name Args

Args : ‘(‘ ‘)’

| ‘(‘ ExpList ‘)’

| TableConstructor

| String

FunctionDef : **function** FuncBody

Funcbody : **‘(‘ ‘)’** Block **end**

| **‘(‘**  ParList **‘)’** Block **end**

ParList : NameList

| NameList **‘,’ ‘…’**

| ‘…’

TableConstructor : **‘{‘ ‘}’**

| **‘{‘** FieldList **‘}’**

FieldList : Field OtherFields FieldSep

| Field OtherFields

OtherFields : FieldSep Field OtherFields | e

Field : **‘[‘** Exp **‘]’** **‘=’** Exp | Name **‘=’** Exp | Exp

Fieldsep : **‘,’** | **‘;’**

Binop : …

Unop : …

%fallback OPEN '(' .

chunk ::= block .

semi ::= ';' .

semi ::= .

block ::= scope statlist .

block ::= scope statlist laststat semi .

ublock ::= block 'until' exp .

scope ::= .

scope ::= scope statlist binding semi.

statlist ::= .

statlist ::= statlist stat semi .

stat ::= 'do' block 'end' .

stat ::= 'while' exp 'do' block 'end' .

stat ::= repetition 'do' block 'end' .

stat ::= 'repeat' ublock .

stat ::= 'if' conds 'end' .

stat ::= 'function' funcname funcbody .

stat ::= setlist '=' explist1 .

stat ::= functioncall .

repetition ::= 'for' NAME '=' explist23 .

repetition ::= 'for' namelist 'in' explist1 .

conds ::= condlist .

conds ::= condlist 'else' block .

condlist ::= cond .

condlist ::= condlist 'elseif' cond .

cond ::= exp 'then' block .

laststat ::= 'break' .

laststat ::= 'return' .

laststat ::= 'return' explist1 .

binding ::= 'local' namelist .

binding ::= 'local' namelist '=' explist1 .

binding ::= 'local' 'function' NAME funcbody .

funcname ::= dottedname .

funcname ::= dottedname ':' NAME .

dottedname ::= NAME .

dottedname ::= dottedname '.' NAME .

namelist ::= NAME .

namelist ::= namelist ',' NAME .

explist1 ::= exp .

explist1 ::= explist1 ',' exp .

explist23 ::= exp ',' exp .

explist23 ::= exp ',' exp ',' exp .

%left 'or' .

%left 'and' .

%left '<' '<=' '>' '>=' '==' '~=' .

%right '..' .

%left '+' '-' .

%left '\*' '/' '%' .

%right 'not' '#' .

%right '^' .

exp ::= 'nil'|'true'|'false'|NUMBER|STRING|'...' .

exp ::= function .

exp ::= prefixexp .

exp ::= tableconstructor .

exp ::= 'not'|'#'|'-' exp . ['not']

exp ::= exp 'or' exp .

exp ::= exp 'and' exp .

exp ::= exp '<'|'<='|'>'|'>='|'=='|'~=' exp .

exp ::= exp '..' exp .

exp ::= exp '+'|'-' exp .

exp ::= exp '\*'|'/'|'%' exp .

exp ::= exp '^' exp .

setlist ::= var .

setlist ::= setlist ',' var .

var ::= NAME .

var ::= prefixexp '[' exp ']' .

var ::= prefixexp '.' NAME .

prefixexp ::= var .

prefixexp ::= functioncall .

prefixexp ::= OPEN exp ')' .

functioncall ::= prefixexp args .

functioncall ::= prefixexp ':' NAME args .

args ::= '(' ')' .

args ::= '(' explist1 ')' .

args ::= tableconstructor .

args ::= STRING .

function ::= 'function' funcbody .

funcbody ::= params block 'end' .

params ::= '(' parlist ')' .

parlist ::= .

parlist ::= namelist .

parlist ::= '...' .

parlist ::= namelist ',' '...' .

tableconstructor ::= '{' '}' .

tableconstructor ::= '{' fieldlist '}' .

tableconstructor ::= '{' fieldlist ','|';' '}' .

fieldlist ::= field .

fieldlist ::= fieldlist ','|';' field .

field ::= exp .

field ::= NAME '=' exp .

field ::= '[' exp ']' '=' exp .