```
function        ⇒   \begin  {  eqcode  }  {  id  }
                      {  [id [  ,  id ]* ]  }
                      {  [ext_type [  ,  ext_type ]* ]  }  {  ext_type  }
                      instr_list
                      \end  {  eqcode  }
indexes         ⇒   (  [upper ] [lower ]  |  lower upper  )
idx             ⇒   id  indexes
numx            ⇒   (  num  | divide ) indexes
idx_numx        ⇒   ( idx | numx )
upper           ⇒   ˆ  {  (  [  expr  ]  | expr )  }
                |   ˆ  (  id  |  num  )
linear          ⇒   id  [(  +  |  -  )  num  ]
                |   num
lower           ⇒   _  {  expr [  ,  expr ]*  }
                |   _  (  id  |  num  )
type            ⇒   \type  {  (  Z  |  R  |  N  |  B  )  }
ext_type        ⇒   type [ ˆ  (  {  sexpr  }  |  num  |  id  )
                      [ _  (  {  sexpr [  ,  sexpr ]*  }  ] ]  |  id  |  num  )
instr_list      ⇒   [instr  \lend  ]*
instr           ⇒   assign
                |   declare
                |   with_loop
                |   comment
                |   if_cond
                |   return
if_cond         ⇒   \qif  {  cond_block  }
                      instr_list
                      [  \qelseif  {  cond_block  }
                      instr_list  ]* [  \qelse
                      instr_list  ]*  \qendif
cond_block      ⇒   expr [comp expr ]+
                      [set_op expr [comp expr ]+ ]*
assign          ⇒   idx [  generator  ]  \gets  expr
declare         ⇒   idx  \in  ext_type
```

```
boolop             ⇒   \land
                   |   \lor
                   |   \oplus
binop              ⇒   +
                   |   -
                   |   \cdot
                   |   \ll
                   |   \gg
                   |   \mod
divide             ⇒  ( \frac | \dfrac ) { expr } { expr }
function_call      ⇒   \call { id } { [expr [ , expr ]* ] }
sexpr              ⇒  ( \lnot | - ) sexpr_op [( binop | boolop ) sexpr_op ]*
                   |   ( expr )
                   |   { expr }
sexpr_op           ⇒  ( idx_numx | function_call | vector | matrix )
filter             ⇒   \filter { id ^ { [ id ] }
                        [ , id ^ { [ id ] } ]*
                         | generator }
genarray           ⇒   \genar \limits ^ { expr } ( expr )
vector             ⇒   \begin { tvector }
                        [expr \lend ]+
                         \end { tvector }
matrix             ⇒   \begin { tmatrix } { id }
                        [expr [ & expr ]* \lend ]+
                         \end { tmatrix }
expr               ⇒  ( sexpr | filter | genarray ) indexes
with_loop          ⇒  idx | generator \gets ( expr | with_loop_cases )
with_loop_cases    ⇒   \begin { cases }
                        [expr & generator ]+
                        [expr & \otherwise ]+
                         \end { cases }
```

return	⇒	**\return** { *expr* }
generator	⇒	**\forall** ***id*** *[* **,** ***id*** *]**
	|	***id*** *[* **,** ***id*** *]** **:** *cond_block*
comp	⇒	**<**
	|	**>**
	|	**\leq**
	|	**\geq**
	|	*[* **\not** *]* **=**
set_op	⇒	*(* **\cup** | **\cap** *)*