function ⇒ **\begin** { **eqcode** } { ***id*** }
{ *[idx [* **,** *idx]*]* }
{ *[ext_type [* **,** *ext_type]*]* } { *ext_type* }
instr_list
\end { **eqcode** }

idx ⇒ ***id*** *[upper] [lower]*
| ***num***
| *divide*

upper ⇒ **^** { *(* **[** *linear* **]** | *linear)* }

linear ⇒ ***id*** *[(* **+** | **-** *)* ***num*** *]*
| ***num***

lower ⇒ **_** { *sexpr [* **,** *sexpr]** }

type ⇒ **\type** { *(* **Z** | **R** | **N** | **B** *)* }

ext_type ⇒ *type [* **^** { *sexpr* }
[**_** { *sexpr [* **,** *sexpr]** } *]]*

instr_list ⇒ *[instr* **\endl** *]**

instr ⇒ *definition*
| *declaration*
| *with_loop*
| *return*

definition ⇒ *[idx]+ expr*

boolop ⇒ **\land**
| **\lor**
| **\oplus**

binop ⇒ **+**
| **-**
| **\cdot**
| **\ll**
| **\gg**
| **\mod**

divide ⇒ *(* **\frac** | **\dfrac** *)* { *expr* } { *expr* }

function_call ⇒ **\call** { ***id*** } { *[idx [* **,** *idx]*]* }

sexpr ⇒ *(\lnot | -) (**idx** | function_call) [(binop | boolop) (**idx** | function_call)]**
| (*sexpr*)

filter ⇒ \filter { **id** ^ { [**id**] }
*[, **id** ^ { [**id**] }]**
| *condition* }

genarray ⇒ \genar \limits ^ { *expr* } (*sexpr*)

vector ⇒ \begin { tvector }
[sexpr \endl *]+*
\end { tvector }

matrix ⇒ \begin { tmatrix } { *[* **id** *]+* }
[sexpr [sexpr & *]** \endl *]+*
\end { tmatrix }

expr ⇒ *sexpr*
| *filter*
| *genarray*
| *vector*
| *matrix*

with_loop ⇒ *with_loop_wbr*
| *with_loop_wobr*

with_loop_wbr ⇒ *idx* | *condition* =
\begin { cases }
[expr & *condition]+*
[expr & \otherwise *]+*
\end { cases }

with_loop_wobr ⇒ *idx* | *condition* = *expr*

return ⇒ \return { *expr* }

condition ⇒ \forall **id** *[, **id**]**
| **id** *[, **id**]** : *sexpr [comp sexpr]+*
*[set_op sexpr [comp sexpr]+]**

comp ⇒ <
| >
| \leq
| \geq
| *[* \not *]* =

set_op ⇒ *(\land | \lor)*