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

idx $\Rightarrow$ **id** *[upper] [lower]*
| **num**
| *divide*

upper $\Rightarrow$ ^ { ([*linear*] | *linear*) }

linear $\Rightarrow$ **id** *[(+ | -) num]*
| **num**

lower $\Rightarrow$ _ { *sexpr [, sexpr]** }

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

ext_type $\Rightarrow$ *type [* ^ { *sexpr* }
[_ { *sexpr [, sexpr]** } *]]*

instr_list $\Rightarrow$ *[instr* \lend *]**

instr $\Rightarrow$ *assign*
| *declare*
| *with_loop*
| *return*

assign $\Rightarrow$ **id** \gets *expr*

declare $\Rightarrow$ **id** \in *ext_type*

boolop $\Rightarrow$ \land
| \lor
| \oplus

binop $\Rightarrow$ +
| -
| \cdot
| \ll
| \gg
| \mod

divide $\Rightarrow$ (\frac | \dfrac) { *expr* } { *expr* }

function_call $\Rightarrow$ \call { *id* } { *[idx [, idx]*]* }

sexpr	$\Rightarrow$	*(* \lnot *|* - *) (* **idx** *| function_call) [(binop | boolop)* *(* **idx** *| function_call)]** *
	|	*(* *sexpr* *)*
filter	$\Rightarrow$	\filter *{* **id** ^ *{* [**id**] *}* *[,* **id** ^ *{* [**id**] *}]** *| generator }*
genarray	$\Rightarrow$	\genar \limits ^ *{* *sexpr* *}* *(* *sexpr* *)*
vector	$\Rightarrow$	\begin *{* tvector *}* *[sexpr* \lend *]+* \end *{* tvector *}*
matrix	$\Rightarrow$	\begin *{* tmatrix *} {* **id** *}* *[sexpr [* & *sexpr* *]** \lend *]+* \end *{* tmatrix *}*
expr	$\Rightarrow$	*sexpr*
	|	*filter*
	|	*genarray*
	|	*vector*
	|	*matrix*
with_loop	$\Rightarrow$	**id** *|* *generator* \gets *(expr | with_loop_cases)*
with_loop_cases	$\Rightarrow$	\begin *{* cases *}* *[expr* & *generator]+* *[expr* & \otherwise *]+* \end *{* cases *}*
return	$\Rightarrow$	\return *{* *expr* *}*
generator	$\Rightarrow$	\forall **id** *[,* **id** *]**
	|	**id** *[,* **id** *]** : *sexpr [comp sexpr]+* *[set_op sexpr [comp sexpr]+]**
comp	$\Rightarrow$	**<**
	|	**>**
	|	\leq
	|	\geq
	|	*[* \not *]* **=**
set_op	$\Rightarrow$	*(* \land *|* \lor *)*