

<i>function</i>	\Rightarrow	$\begin{aligned} & \backslash\textbf{begin} \{ \textbf{eqcode} \} \{ \textbf{id} \} \\ & \{ [id [, id]^*] \} \\ & \{ [ext_type [, ext_type]^*] \} \{ ext_type \} \\ & instr_list \\ & \backslash\textbf{end} \{ \textbf{eqcode} \} \end{aligned}$
<i>indexes</i>	\Rightarrow	$([upper] [lower] \mid lower\ upper)$
<i>idx</i>	\Rightarrow	$\textbf{id} \ indexes$
<i>numx</i>	\Rightarrow	$(\textbf{num} \mid divide) \ indexes$
<i>idx_numx</i>	\Rightarrow	$(idx \mid numx)$
<i>upper</i>	\Rightarrow	$\begin{aligned} & \wedge \{ ([(linear \mid expr)] \mid expr) \} \\ & \mid \\ & \wedge (\textbf{id} \mid \textbf{num}) \end{aligned}$
<i>linear</i>	\Rightarrow	$(\backslash\textbf{iter} [(+ \mid -) \textbf{num}] \mid \textbf{num})$
<i>lower</i>	\Rightarrow	$\begin{aligned} & - \{ expr [, expr]^* \} \\ & \mid \\ & - (\textbf{id} \mid \textbf{num}) \end{aligned}$
<i>type</i>	\Rightarrow	$\backslash\textbf{type} \{ (\textbf{Z} \mid \textbf{R} \mid \textbf{N} \mid \textbf{B}) \}$
<i>arraytype</i>	\Rightarrow	$\begin{aligned} & \backslash\textbf{arraytype} \{ (sexpr \mid \backslash\textbf{ldots}) [, (sexpr \mid \backslash\textbf{ldots})] \} \\ & \{ (\textbf{Z} \mid \textbf{R} \mid \textbf{N} \mid \textbf{B}) \} \end{aligned}$
<i>functiontype</i>	\Rightarrow	$([ext_type]^* \backslash\textbf{to} [ext_type]^+)$
<i>ext_type</i>	\Rightarrow	$(type \mid arraytype \mid functiontype)$
<i>instr_list</i>	\Rightarrow	$[instr \ \backslash\textbf{lend}]^*$
<i>instr</i>	\Rightarrow	$\begin{aligned} & assign \\ & \mid \\ & declare \\ & \mid \\ & index_loop \\ & \mid \\ & comment \\ & \mid \\ & if_cond \\ & \mid \\ & return \end{aligned}$
<i>if_cond</i>	\Rightarrow	$\begin{aligned} & \backslash\textbf{qif} \{ cond_block \} \\ & instr_list \\ & [\backslash\textbf{qelseif} \{ cond_block \} \\ & instr_list]^* [\backslash\textbf{qelse} \\ & instr_list]^* \backslash\textbf{qendif} \end{aligned}$
<i>cond_block</i>	\Rightarrow	$\begin{aligned} & expr [comp \ expr]^+ \\ & [set_op \ expr [comp \ expr]^+]^* \end{aligned}$

<i>assign</i>	\Rightarrow	<i>idx</i> [, <i>idx</i>]* \b{gets} <i>expr</i> [, <i>expr</i>]*
<i>declare</i>	\Rightarrow	<i>idx</i> [, <i>idx</i>]* \b{in} <i>ext_type</i> [, <i>ext_type</i>]*
<i>boolop</i>	\Rightarrow	\b{land}
		\b{lor}
		\b{oplus}
<i>binop</i>	\Rightarrow	+
		-
		\b{cdot}
		\b{ll}
		\b{gg}
		\b{mod}
<i>divide</i>	\Rightarrow	(\b{frac} \b{dfrac}) { <i>expr</i> } { <i>expr</i> }
<i>function_call</i>	\Rightarrow	\b{call} { <i>id</i> } { [<i>expr</i> [, <i>expr</i>]*] }
<i>sexpr</i>	\Rightarrow	(\b{lnot} -) <i>sexpr_op</i> [(<i>binop</i> <i>boolop</i>) <i>sexpr_op</i>]*
		(<i>expr</i>)
		{ <i>expr</i> }
<i>sexpr_op</i>	\Rightarrow	(<i>idx_num</i> <i>x</i> <i>function_call</i> <i>matrix</i>)
<i>filter</i>	\Rightarrow	\b{filter} { <i>id</i> [, <i>id</i>]* }
		<i>cond_block</i> }
<i>genarray</i>	\Rightarrow	\b{genar} \b{limits} ^ { <i>expr</i> } (<i>expr</i>)
<i>matrix</i>	\Rightarrow	\b{begin} { <i>tmatrix</i> }
		[<i>expr</i> [& <i>expr</i>]* \b{lend}]+
		\b{end} { <i>tmatrix</i> }
<i>expr</i>	\Rightarrow	(<i>sexpr</i> <i>filter</i> <i>genarray</i>) <i>indexes</i>
<i>index_loop</i>	\Rightarrow	<i>idx</i> <i>generator</i> \b{gets} (<i>expr</i> <i>index_loop_cases</i>)
<i>index_loop_cases</i>	\Rightarrow	\b{begin} { <i>cases</i> }
		[<i>expr</i> & <i>generator</i>]+
		[<i>expr</i> & \b{otherwise}]+
		\b{end} { <i>cases</i> }

<i>print</i>	\Rightarrow	<code>\print { expr }</code>
<i>return</i>	\Rightarrow	<code>\return { expr }</code>
<i>generator</i>	\Rightarrow	<code>\forall id [, id]*</code>
		<code>id [, id]* : cond_block</code>
<i>comp</i>	\Rightarrow	<code><</code>
		<code>></code>
		<code>\leq</code>
		<code>\geq</code>
		<code>[\not] =</code>
<i>set_op</i>	\Rightarrow	<code>(\cup \cap)</code>