```
PROGRAM -> TASK_DEFINITIONS; parbegin TASK_LIST parend
TASK_DEFINITIONS -> TASK_DEFINITION TASK_DEFINITIONSTAG
TASK_DEFINITIONSTAG -> ;TASK_DEFINITION TASK_DEFINITIONSTAG |epsilon
TASK_DEFINITION -> task id begin DECLARATIONS { COMMANDS } end
TASK_LIST -> task_id TASK_LISTTAG
TASK_LISTTAG -> || task_id TASK_LISTTAG|epsilon
DECLARATIONS -> DECLARATION DECLARATIONSTAG
DECLARATIONSTAG -> ; DECLARATIONS|epsilon
DECLARATION -> integer  id   |  real id
COMMANDS -> COMMAND  COMMANDSTAG
COMMANDSTAG -> ; COMMAND COMMANDSTAG | epsilon
COMMAND -> id = EXPRESSION  | do COMMANDS until CONDITION od   |

                        send task_id . signal_id (PARAM_LIST)  |

                      accept signal_id (DECLARATIONS)  |

                           begin DECLARATIONS { COMMANDS } end

PARAM_LIST -> EXPRESSION PARAM_LISTTAG
PARAM_LISTTAG -> , PARAM_LIST | epsilon
EXPRESSION -> int_num |real_num |id EXPRESSIONTAG
EXPRESSIONTAG -> binary_ar_op EXPRESSION |epsilon
CONDITION-> ( id rel_op )
```

	First	Follow	nullable
PROGRAM	task	EOF	NO
TASK_DEFINITIONS	task	;	NO
TASK_DEFINITIONSTAG	;	;	YES
TASK_DEFINITION	task	;	NO
TASK_LIST	task_id	parend	NO
TASK_LISTTAG	||	parend	YES
DECLARATIONS	integer | real	{ |)	NO
DECLARATIONSTAG	;	{ |)	YES
DECLARATION	integer | real	;| { |)	NO
COMMANDS	id|do|send|accept|begin	} |until	NO
COMMANDSTAG	;	} | until	YES
COMMAND	id|do|send|accept|begin	;| } | until	NO
PARAM_LIST	int_num | real_num | id	)	NO
PARAM_LISTTAG	,	)	YES
EXPRESSION	int | real | id	} |until | ,|)	NO
EXPRESSIONTAG	binary_ar_op	} |until | ,|)	YES
CONDITION	(	od	NO