Edit Transform Analyze

```
PROGRAM -> STATEMENT | FUNCLIST | .
FUNCLIST -> FUNCDEF FUNCLIST' .
FUNCLIST' -> FUNCLIST | .
FUNCDEF -> def ident ( PARAMLIST ) {
STATELIST } .
PARAMLIST -> int ident PARAMLIST' |
float ident PARAMLIST'
  | string ident PARAMLIST' | .
PARAMLIST' -> , PARAMLIST | .
STATEMENT -> VARDECL ;
    | ATRIBSTAT ;
    | PRINTSTAT ;
| READSTAT ;
| RETURNSTAT ;
| IFSTAT
| FORSTAT
| { STATELIST }
| break ;
| ; .
VARDECL -> int ident Z | float ident Z
| string ident Z .
Z -> [ int_constant ] Z | .
ATRIBSTAT -> LVALUE = ATRIBSTAT' .
ATRIBSTAT' -> EXPRESSION |
ALLOCEXPRESSION | FUNCCALL .
FUNCCALL -> ident ( PARAMLISTCALL ) .
PARAMLISTCALL -> ident PARAMLISTCALL'
| .
PARAMLISTCALL' -> , PARAMLISTCALL | .
PRINTSTAT -> print EXPRESSION .
READSTAT -> read LVALUE .
RETURNSTAT -> return .
IFSTAT -> if ( EXPRESSION ) STATEMENT
S .
S -> else STATEMENT | ignore .
FORSTAT -> for( ATRIBSTAT ; EXPRESSION
; ATRIBSTAT) STATEMENT .
STATELIST -> STATEMENT STATELIST' .
STATELIST' -> STATEMENT STATELIST' .
STATELIST' -> STATELIST | .
ALLOCEXPRESSION -> new T K .
T -> int | float | string .
K -> [ NUMEXPRESSION ] K' .
K' -> K | .
EXPRESSION -> NUMEXPRESSION G .
G -> P NUMEXPRESSION | .
P -> < | > | <= | >= | == | != .
NUMEXPRESSION -> TERM L .
L -> L' .
O -> TERM UNARYEXPR M .
M -> N UNARYEXPR | .
N -> * | / | % .
UNARYEXPR -> R FACTOR .
R -> + | - | .
FACTOR -> int_constant |
float_constant | string_constant |
null | LVALUE
                          | (
NUMEXPRESSION ) .
LVALUE -> ident K .
L' -> O TERM L' | .
```

Analysis / **LL(1) Parsing Table**

	def	ident	(	)	{	}	int	float	string	,	;	break	[	int_constant	]	=	print	read	return	if	else	ignore	for(| ATRIBSTAT)	new	<	>	<=	>=	==	!=	TERM	*	/
PROGRAM	*PROGRAM* → *FUNCLIST*	*PROGRAM* → *STATEMENT*			*PROGRAM* → *STATEMENT*		*PROGRAM* → *STATEMENT*	*PROGRAM* → *STATEMENT*	*PROGRAM* → *STATEMENT*		*PROGRAM* → *STATEMENT*	*PROGRAM* → *STATEMENT*					*PROGRAM* → *STATEMENT*	*PROGRAM* → *STATEMENT*	*PROGRAM* → *STATEMENT*	*PROGRAM* → *STATEMENT*			*PROGRAM* → *STATEMENT*									
FUNCLIST	*FUNCLIST* → *FUNCDEF FUNCLIST'*																																
FUNCLIST'	*FUNCLIST'* → *FUNCLIST*																																
FUNCDEF	*FUNCDEF* → **def ident (** *PARAMLIST* **) {** *STATELIST* **}**																																
PARAMLIST				*PARAMLIST* → ε			*PARAMLIST* → **int ident** *PARAMLIST'*	*PARAMLIST* → **float ident** *PARAMLIST'*	*PARAMLIST* → **string ident** *PARAMLIST'*																								
PARAMLIST'				*PARAMLIST'* → ε						*PARAMLIST'* → **,** *PARAMLIST*																							
STATEMENT		*STATEMENT* → *ATRIBSTAT* **;**			*STATEMENT* → **{** *STATELIST* **}**		*STATEMENT* → *VARDECL* **;**	*STATEMENT* → *VARDECL* **;**	*STATEMENT* → *VARDECL* **;**		*STATEMENT* → **;**	*STATEMENT* → **break ;**					*STATEMENT* → *PRINTSTAT* **;**	*STATEMENT* → *READSTAT* **;**	*STATEMENT* → *RETURNSTAT* **;**	*STATEMENT* → *IFSTAT*			*STATEMENT* → *FORSTAT*									
VARDECL							*VARDECL* → **int ident** *Z*	*VARDECL* → **float ident** *Z*	*VARDECL* → **string ident** *Z*																								
Z											*Z* → ε		*Z* → **[int_constant]** *Z*																				
ATRIBSTAT		*ATRIBSTAT* → *LVALUE* **=** *ATRIBSTAT'*																															
ATRIBSTAT'		*ATRIBSTAT'* → *FUNCCALL*																								*ATRIBSTAT'* → *ALLOCEXPRESSION*							*ATRIBSTAT'* → *EXPRESSION*			
FUNCCALL		*FUNCCALL* → **ident (** *PARAMLISTCALL* **)**																															
PARAMLISTCALL		*PARAMLISTCALL* → **ident** *PARAMLISTCALL'*		*PARAMLISTCALL* → ε																													
PARAMLISTCALL'				*PARAMLISTCALL'* → ε						*PARAMLISTCALL'* → **,** *PARAMLISTCALL*																							
PRINTSTAT																	*PRINTSTAT* → **print** *EXPRESSION*															
READSTAT																		*READSTAT* → **read** *LVALUE*															
RETURNSTAT																			*RETURNSTAT* → **return**													
IFSTAT																				*IFSTAT* → **if (** *EXPRESSION* **)** *STATEMENT S*													
S																					*S* → **else** *STATEMENT*	*S* → **ignore**										
FORSTAT																							*FORSTAT* → **for(** *ATRIBSTAT* **;** *EXPRESSION* **;** **ATRIBSTAT)** *STATEMENT*											
STATELIST		*STATELIST* → *STATEMENT STATELIST'*			*STATELIST* → *STATEMENT STATELIST'*		*STATELIST* → *STATEMENT STATELIST'*	*STATELIST* → *STATEMENT STATELIST'*	*STATELIST* → *STATEMENT STATELIST'*		*STATELIST* → *STATEMENT STATELIST'*	*STATELIST* → *STATEMENT STATELIST'*					*STATELIST* → *STATEMENT STATELIST'*	*STATELIST* → *STATEMENT STATELIST'*	*STATELIST* → *STATEMENT STATELIST'*	*STATELIST* → *STATEMENT STATELIST'*			*STATELIST* → *STATEMENT STATELIST'*									
STATELIST'		*STATELIST'* → *STATELIST*			*STATELIST'* → *STATELIST*	*STATELIST'* → ε	*STATELIST'* → *STATELIST*	*STATELIST'* → *STATELIST*	*STATELIST'* → *STATELIST*		*STATELIST'* → *STATELIST*	*STATELIST'* → *STATELIST*					*STATELIST'* → *STATELIST*	*STATELIST'* → *STATELIST*	*STATELIST'* → *STATELIST*	*STATELIST'* → *STATELIST*			*STATELIST'* → *STATELIST*									
ALLOCEXPRESSION																								*ALLOCEXPRESSION* → **new** *T K*									
T							*T* → **int**	*T* → **float**	*T* → **string**																								
K													*K* → **[** *NUMEXPRESSION* **]** *K'*																				
K'											*K'* → ε		*K'* → *K*			*K'* → ε															*K'* → ε	*K'* → ε	*K'* → ε
EXPRESSION																															*EXPRESSION* → *NUMEXPRESSION G*		
G				*G* → ε							*G* → ε															*G* → *P NUMEXPRESSION*	*G* → *P NUMEXPRESSION*	*G* → *P NUMEXPRESSION*	*G* → *P NUMEXPRESSION*	*G* → *P NUMEXPRESSION*	*G* → *P NUMEXPRESSION*			
P																										*P* → **<**	*P* → **>**	*P* → **<=**	*P* → **>=**	*P* → **==**	*P* → **!=**			
NUMEXPRESSION																															*NUMEXPRESSION* → **TERM** *L*		
L				*L* → *L'*							*L* → *L'*				*L* → *L'*											*L* → *L'*	*L* → *L'*	*L* → *L'*	*L* → *L'*	*L* → *L'*	*L* → *L'*	*L* → *L'*		
O																															*O* → **TERM** *UNARYEXPR M*		
M																															*M* → ε	*M* → *N UNARYEXPR*	*M* → *N UNARYEXPR*
N																																*N* → *	*N* → /
UNARYEXPR		*UNARYEXPR* → *R FACTOR*	*UNARYEXPR* → *R FACTOR*											*UNARYEXPR* → *R FACTOR*																			
R		*R* → ε	*R* → ε											*R* → ε																			
FACTOR		*FACTOR* → *LVALUE*	*FACTOR* → **(** *NUMEXPRESSION* **)**											*FACTOR* → **int_constant**																			
LVALUE		*LVALUE* → **ident** *K*																															
L'				*L'* → ε							*L'* → ε				*L'* → ε											*L'* → ε	*L'* → ε	*L'* → ε	*L'* → ε	*L'* → ε	*L'* → ε	*L'* → *O* **TERM** *L'*		