Fase 0

myR Lang

Diseño de Compiladores

Ernesto Ramón Adame Cendejas
A00825923

Profesor:

M.C. Elda G. Quiroga,

Dr. Héctor Ceballos, PhD

Monterrey, Nuevo León

Agosto 2023

Requerimientos

Tokens

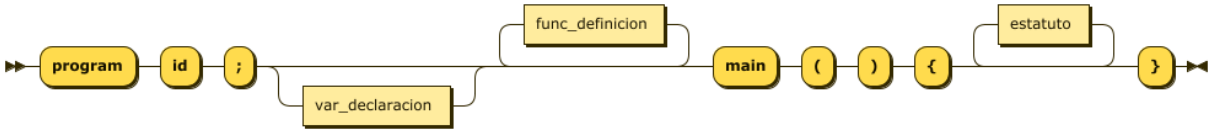
```
Unset
#LITERALS
LPAREN : (
RPAREN : )
LBRACK : [
RBRACK : ]
LCURLY : {
RCURLY : }

GR_THAN : >
```

```
LS_THAN : <

SQUOTE : '
DQUOTE : "

COLON : :
SCOLON : ;
DOT  : .
COMMA : ,

PLUS  : +
MINUS : -
TIMES : *
DIVIDE : /
EQUALS : =

AND : &
OR : |
COMMENT : %

#RESERVED
program
main
vars

int
float
char
string

function
return
void

read
write

if
then
else

while
for
do
to

#TOKENS
ID
[a-zA-Z][a-zA-Z_0-9]*
```

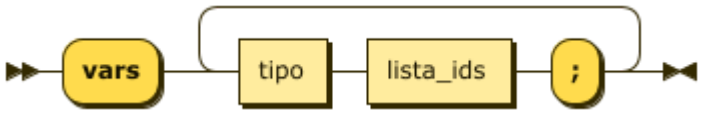
```
VAL_INT
\d+

VAL_FLOAT
\d+\.\d+

VAL_STRING
"[^"]*"

VAL_CHAR
'[a-zA-Z]'

COMP_GR_EQ_THAN # >=
>=

COMP_LS_EQ_THAN # <=
<=

COMP_EQ_TO   # ==
==

COMP_NOT_EQ_TO # !=
!=
```

Diagramas de sintaxis

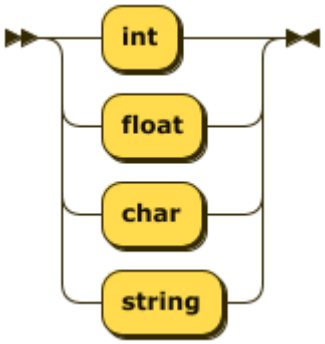
programa



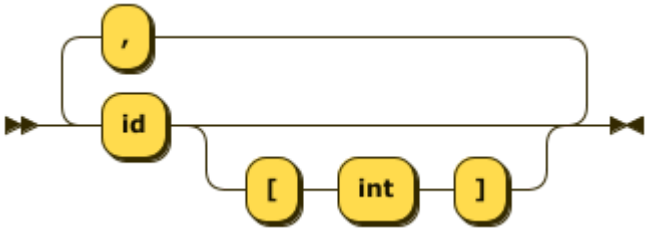
var_declaracion



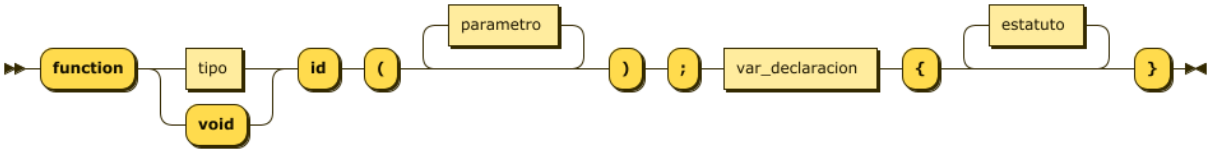
tipo



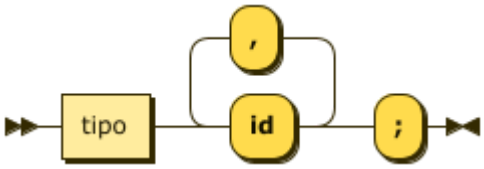
lista_ids



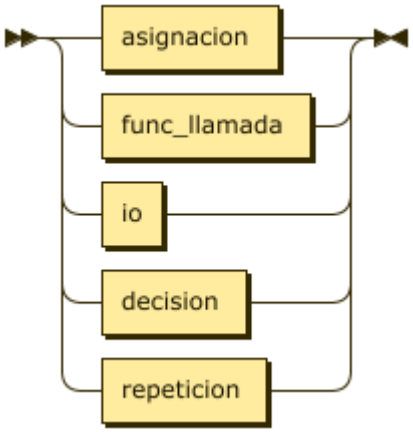
func_definicion



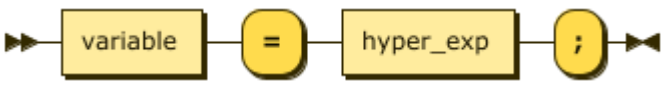
parametro



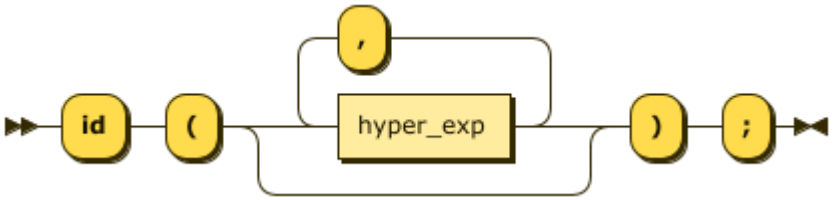
estatuto



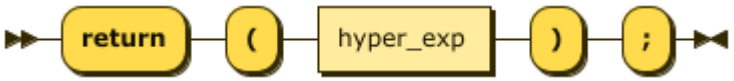
asignacion



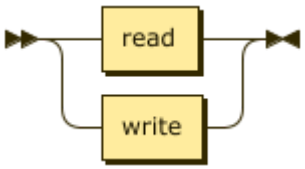
func_llamada



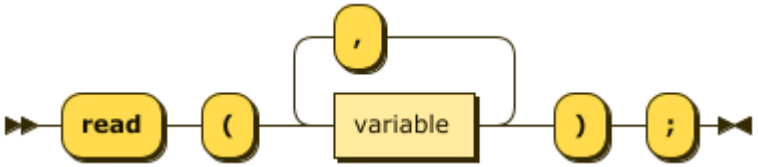
func_retorno



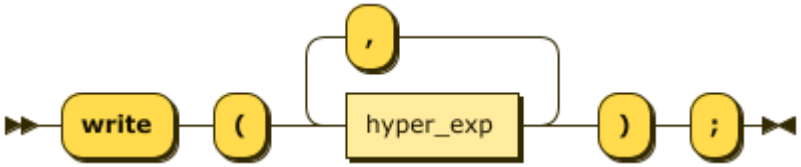
io



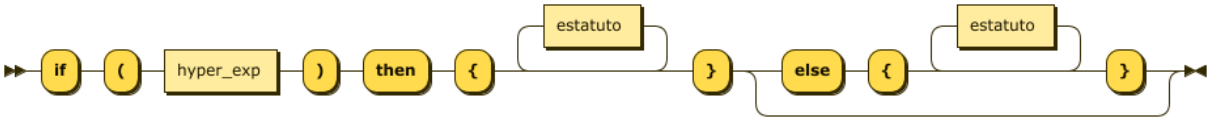
read



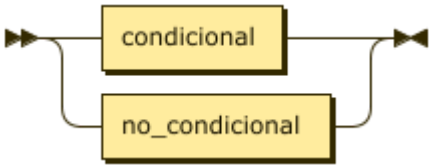
write



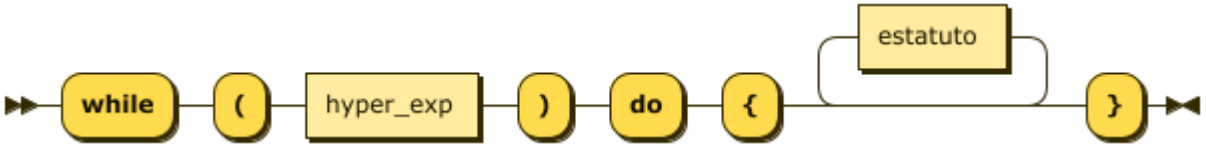
decision



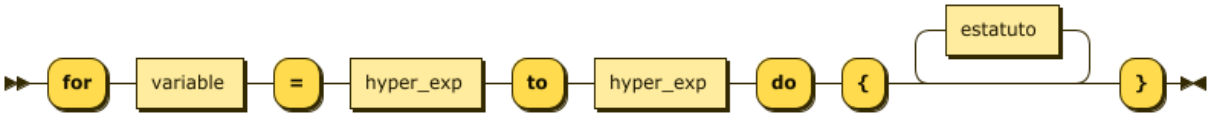
repeticion



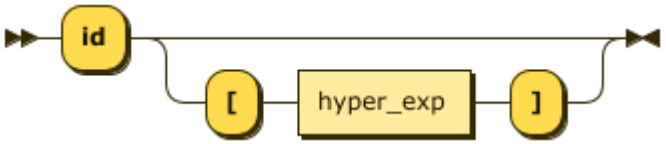
condicional



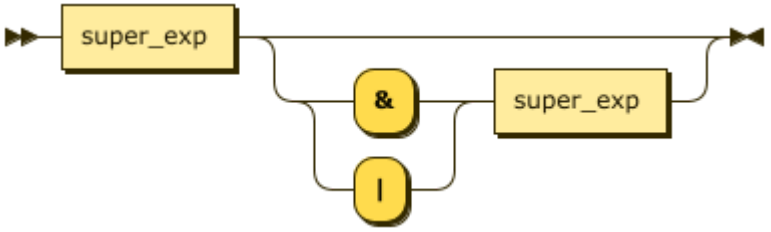
no_condicional



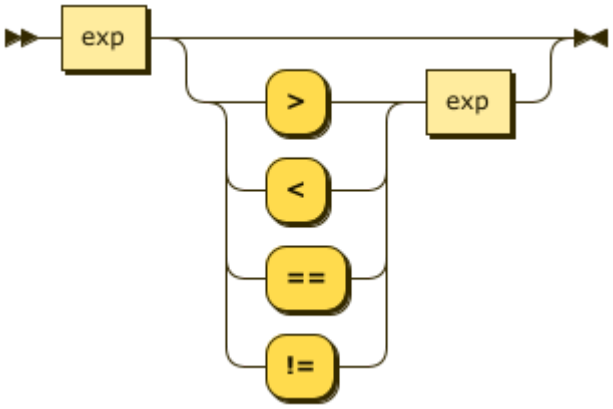
variable



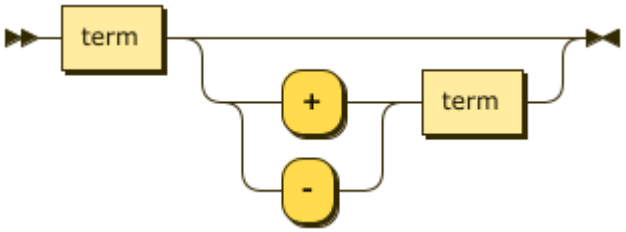
hyper_exp



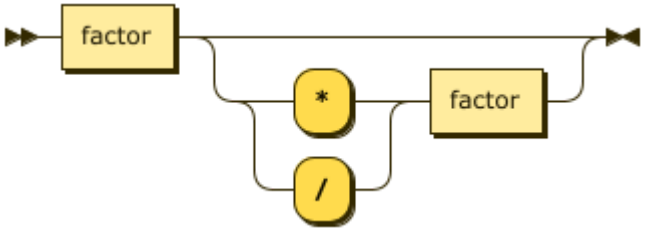
super_exp



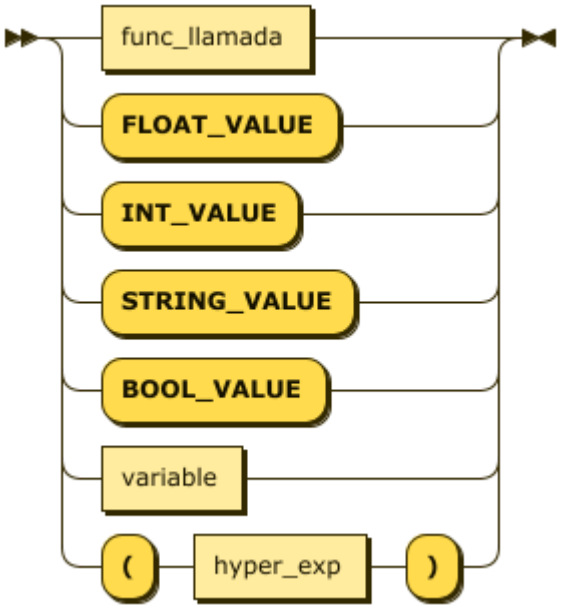
exp



term



factor



Gramática

```
Unset
programa ->
       PROGRAM ID ; var_opcional func_programa_loop MAIN ( ) { loop_estatuto }

var_opcional ->
       var_declaracion
       | VOID

var_declaracion ->
       VARS tipo lista_ids ; loop_var_decl

loop_var_decl ->
       tipo lista_ids ; loop_vd
       | VOID

func_programa_loop ->
       func_definicion func_programa_loop
       | empty

func_definicion ->
       FUNCTION func_tipo_retorno ID ( func_parametro_loop ) ; var_opcional {
loop_estatuto }

func_tipo_retorno ->
       tipo
       | VOID

func_parametro ->
```

```
        parametro func_parametro
        | VOID

parametro ->
        tipo ID loop_parametro ;

loop_parametro ->
        , ID loop_parametro
        | VOID

tipo -> INT
        | FLOAT
        | CHAR
        | STRING

lista_ids ->
        id loop_lista_ids
        | id [ INT ] loop_lista_ids

loop_lista_ids ->
        , id loop_lista_ids
        |, id [ INT ] loop_lista_ids
        | VOID

estatuto ->
        asignacion
        | func_llamada
        | io
        | decision
        | repeticion

asignacion ->
        variable = hyper_exp ;

func_llamada ->
        ID ( ) ;
        | ID ( hyper_exp_loop ) ;

hyper_exp_loop ->
        hyper_exp hyper_exp_loop_1

hyper_exp_loop_1 ->
        , hyper_exp hyper_exp_loop_1
        | VOID

func_retorno ->
        RETURN ( hyper_exp ) ;

io ->
```

```
        read
        | write

read ->
        READ ( variable_loop ) ;

variable_loop ->
        variable variable_loop_1

variable_loop_1 ->
        , variable variable_loop_1
        | VOID

write ->
        write ( hyper_exp_loop ) ;

decision ->
        IF ( hyper_exp ) THEN { loop_estatuto } decision_1

decision1 ->
        ELSE { loop_estatuto }
        | VOID

variable->
        ID
        ID [ hyper_exp ]

loop_estatuto ->
        estatuto loop_estatuto_1
        | VOID

loop_estatuto_1 ->
        estatuto loop_estatuto_1
        | VOID

repeticion ->
        condicional
        | no_condicional

condicional ->
        WHILE ( hyper_exp ) do { loop_estatuto }

no_condicional ->
        FOR variable = hyper_exp to hyper_exp do { loop_estatuto }

hyper_exp ->
        super_exp hyper_exp_1
```

```
hyper_exp_1 ->
       '&' super_exp
       | '|' super_exp
       | VOID

super_exp ->
       exp super_exp_1

super_exp_1 ->
       > exp
       | < exp
       | == exp
       | != exp
       | VOID

exp ->
       term exp_1

exp_1 ->
       + term
       | - term
       | void

term ->
       factor term1

term1 ->
       * factor
       | / factor
       | VOID

factor ->
       func_llamada
       | FLOAT_VALUE
       | INT_VALUE
       | STRING_VALUE
       | BOOL_VALUE
       | variable
       | ( hyper_exp )
```

Consideraciones Semánticas

Jerarquía de operadores

Nombre	Símbolo	Prioridad
Paréntesis	( )	0
Multiplicación	*	1
División	/	1
Suma	+	2

Nombre	Símbolo	Prioridad
Resta	-	2
Relacionales	< > == !=	3
Lógicos	and or	4
Asignación	=	5

Tipos de Datos

A	B	*	/	+	-	Relational Op	Logical
int	int	int	float	int	int	bool	bool
int	float	float	float	float	float	bool	bool
int	bool	err	err	err	err	err	bool
int	string	err	err	err	err	err	err
float	int	float	float	float	float	bool	bool
float	float	float	float	float	float	bool	bool
float	bool	err	err	err	err	err	bool
float	string	err	err	err	err	err	err
bool	int	err	err	err	err	err	bool
bool	float	err	err	err	err	err	bool
bool	bool	err	err	err	err	err	bool

bool	string	err	err	err	err	err	err
string	int	err	err	err	err	err	err
string	float	err	err	err	err	err	err
string	bool	err	err	err	err	err	err
string	string	err	err	err	err	err	err