



# Tecnológico de Monterrey

Instituto Tecnológico de Estudios Superiores de Monterrey

Escuela de Ingeniería y Ciencias

TC3048. Diseño de Compiladores

Proyecto Final

Avance 1 - Análisis de Léxico y Sintaxis

Gustavo Alejandro Vasquez Acosta

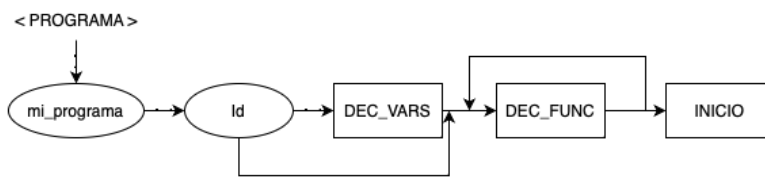
A00823326

Profesora

Elda Quiroga

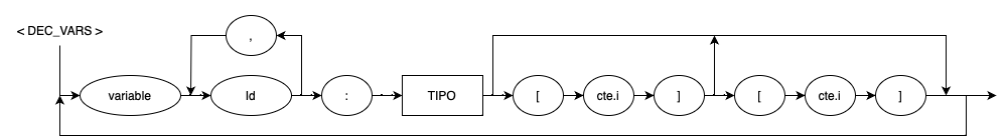
En este avance 1, este documento se enfoca en presentar los diagramas de sintaxis del lenguaje de programación para mi proyecto final, llamado *JuniorScript*. Además de presentar los diagramas, se presentara su gramática normal, y de ser el caso, su factorización y/o eliminación de recursividad izquierda con propósito de mejorar el entendimiento de los diagramas y tener un código mas limpio a la hora de trasladarlo a *Python Lex and Yacc*.

## Programa [ PROG ]



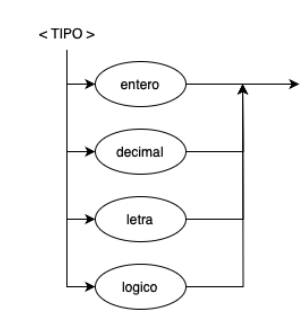
<i>Normal</i>	<i>Factorización</i>	<i>Recursividad Izquierda</i>
PROG : mi_programa id DV PROG' INIT   mi_programa id PROG' INIT	PROG : mi_programa id PROG_B PROG' INIT	PROG : mi_programa id PROG_B PROG' INIT
PROG' : PROG' DF   DF	PROG_B : DV   ε  PROG' : PROG' DF   DF	PROG_B : DV   ε  PROG' : DF PROG"  PROG" : PROG'   ε

DEC\_VARS [ DV ]



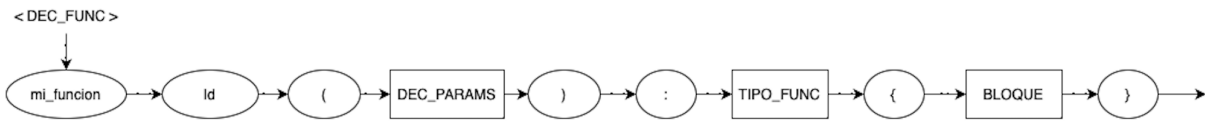
Normal	Factorización	Recursividad Izquierda
DV : DV variable l : TIP [ cte.i ] [ cte.i ]   DV variable l : TIP [ cte.i ]   DV variable l : TIP  l : l id ,   id	DV : DV variable l : TIP DV_B  DV_B : [ cte.i ] [ cte.i ]   [ cte.i ]   ε  l : l id ,   id	DV : variable l : TIP DV_B DV'  DV_B : [ cte.i ] [ cte.i ]   [ cte.i ]   ε  l : id l'  l' : , id l'   ε  DV' : DV   ε

TIPO [ TIP ]



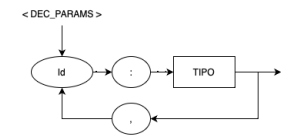
Normal	Factorización	Recursividad Izquierda
TIP : entero   decimal   letra   logico		

DEC\_FUNC [ DF ]



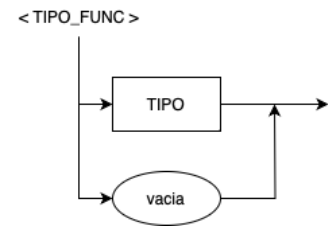
Normal	Factorización	Recursividad Izquierda
DF : mi_funcion id ( DEC_PARAMS ) : TIPO_FUNC { BLOQUE }		

DEC\_PARAMS [ DP ]



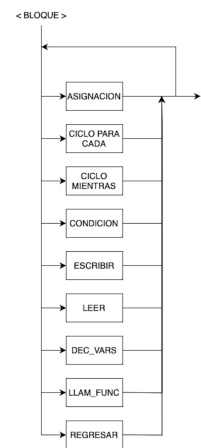
Normal	Factorización	Recursividad Izquierda
DP : id : TIP   id : TIP , DP	DP : id : TIP DP_B  DP_B : , DP   ε	

TIPO\_FUNC [ TF ]



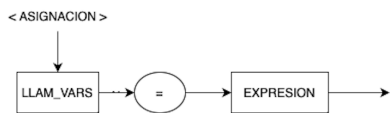
Normal	Factorización	Recursividad Izquierda
TF : TIP   vacia		

BLOQUE [ BLOQ ]



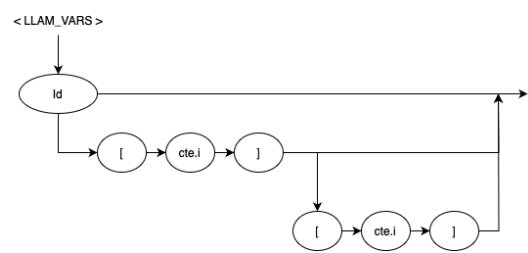
Normal	Factorización	Recursividad Izquierda
BLOQ : BLOQ ASIG   BLOQ CPC   BLOQ CM   BLOQ COND   BLOQ ESC   BLOQ LEE   BLOQ DV   BLOQ LF   BLOQ RET	BLOQ : BLOQ BLOQ_B  BLOQ_B : ASIG   CPC   CM   COND   ESC   LEE   DV   LF   RET	BLOQ: BLOQ_B BLOQ'  BLOQ' : BLOQ_B BLOQ'   ε  BLOQ_B : ASIG   CPC   CM   COND   ESC   LEE   DV   LF   RET

ASIGNACION [ ASIG ]



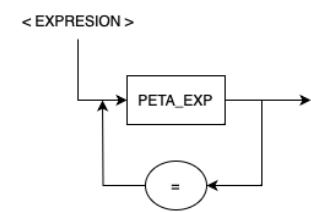
Normal	Factorización	Recursividad Izquierda
ASIG : LLAM_VARS = EXPR		

LLAM\_VARS [ LV ]



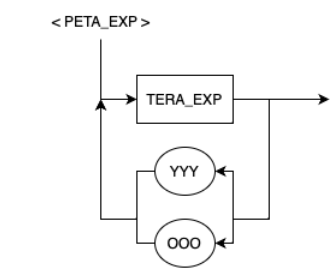
Normal	Factorización	Recursividad Izquierda
LV : id   id [ cte.i ]   id [ cte.i ] [ cte.i ]		

EXPRESION [ EXP ]



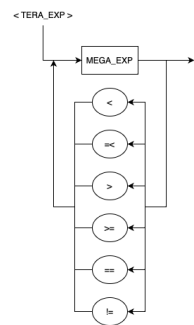
Normal	Factorización	Recursividad Izquierda
EXP : PETA_EXP   PETA_EXP = EXP	EXP : PETA_EXP EXP_B  EXPR_B : = EXP   ε	

PETA\_EXP [ P\_EXP ]



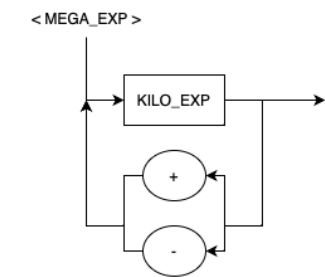
Normal	Factorización	Recursividad Izquierda
P_EXP : TERA_EXP   TERA_EXP YYY P_EXP   TERA_EXP 000 P_EXP	P_EXP: TERA_EXP P_EXP_B  P_EXP_B : YYY P_EXP   000 P_EXP   ε	

TERA\_EXP [ T\_EXP ]



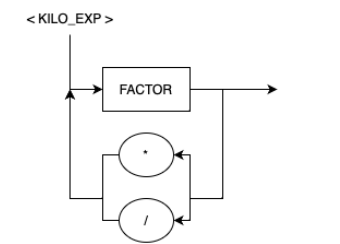
Normal	Factorización	Recursividad Izquierda
T_EXP : MEGA_EXP   MEGA_EXP < T_EXP   MEGA_EXP <= T_EXP   MEGA_EXP > T_EXP   MEGA_EXP >= T_EXP   MEGA_EXP == T_EXP   MEGA_EXP != T_EXP	T_EXP : MEGA_EXP T_EXP_B  T_EXP_B : < T_EXP   <= T_EXP   > T_EXP   >= T_EXP   == T_EXP   != T_EXP   ε	

MEGA\_EXP [ M\_EXP ]



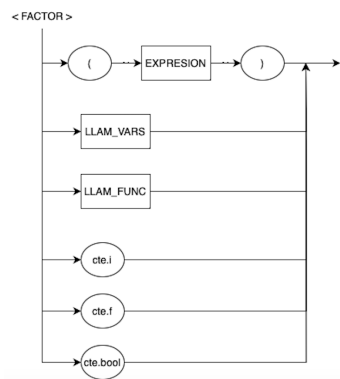
Normal	Factorización	Recursividad Izquierda
M_EXP : KILO_EXP   KILO_EXP + M_EXP   KILO_EXP - M_EXP	M_EXP: KILO_EXP M_EXP_B  M_EXP_B : + M_EXP   - M_EXP   ε	

KILO\_EXP [ K\_EXP ]



Normal	Factorización	Recursividad Izquierda
K_EXP : FACTOR   FACTOR * K_EXP   FACTOR / K_EXP	K_EXP : FACTOR K_EXP_B  K_EXP_B : * K_EXP   / K_EXP   ε	

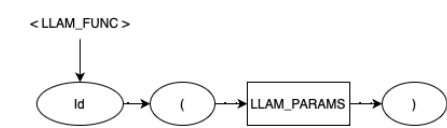
FACTOR [ FACT ]



Normal	Factorización	Recursividad Izquierda
FAC : ( EXP )   LLAM_VARS   LLAM_FUNC   cte.i   cte.f   cte.bool		

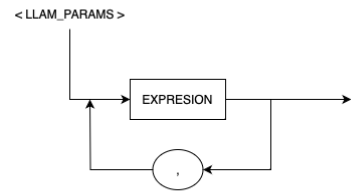


LLAM\_FUNC [ LF ]



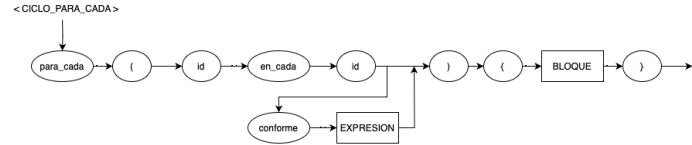
Normal	Factorización	Recursividad Izquierda
LF : id ( LLAM_PARAMS )		

LLAM\_PARAMS [ LP ]



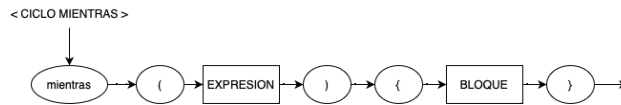
Normal	Factorización	Recursividad Izquierda
LP : EXPR   EXPR , LP	LP : EXPR LP_B  LP_B : , LP   ε	

CICLO PARA CADA [ CPC ]



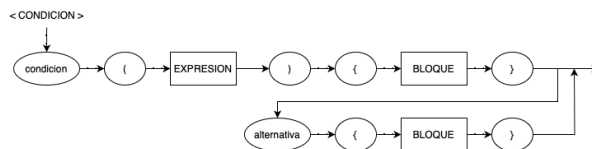
Normal	Factorización	Recursividad Izquierda
CPC : para_cada ( id en _cada id ) { BLOQUE }   para_cada ( id en_cada id conforme EXPR ) { BLOQUE }	CPC : para_cada ( id en_cada id CPC_B ) { BLOQUE }  CPC_B : conforme EXPR   ε	

## CICLO MIENTRAS [ CM ]



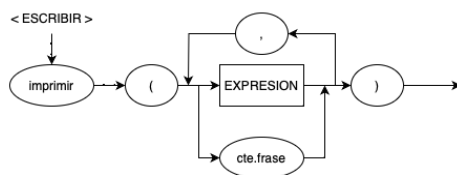
<i>Normal</i>	<i>Factorización</i>	<i>Recursividad Izquierda</i>
CM : mientras ( EXPR ) { BLOQUE }		

## CONDICION [ COND ]



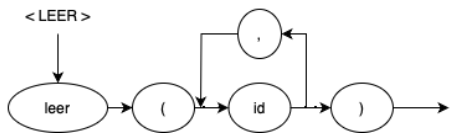
<i>Normal</i>	<i>Factorización</i>	<i>Recursividad Izquierda</i>
COND : condicion ( EXPR ) { BLOQUE }   condicion ( EXPR ) { BLOQUE } alternativa { BLOQUE }	COND : condicion ( EXPR ) { BLOQUE } COND_B  COND_B : alternativa { BLOQUE }   ε	

## ESCRIBIR [ ESC ]



<i>Normal</i>	<i>Factorización</i>	<i>Recursividad Izquierda</i>
ESC : imprimir ( ESC' )  ESC' : EXPR   ESC' EXPR ,   cte.frase   ESC' cte.frase ,		ESC : imprimir ( ESC' )  ESC' : EXPR ESC"   cte.frase ESC"  ESC" : , ESC'   ε

LEER [ LEE ]



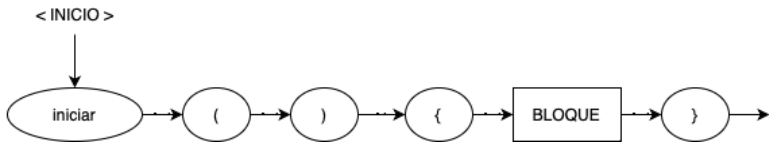
Normal	Factorización	Recursividad Izquierda
LEE : leer ( l )  l : id   l id ,		LEE : leer ( l )  l : id l'  l' : , id l'   ε

REGRESAR [ RET ]



Normal	Factorización	Recursividad Izquierda
RET: regresar ( EXPR )		

Inicio [ INIT ]



Normal	Factorización	Recursividad Izquierda
INIT : iniciar ( ) { BLOQ }		