

Lenguajes y Autómatas II

Tecnológico Nacional de México

Instituto Tecnológico de la Laguna

Ing. En Sistemas Computacionales

Catedrático: Ing. Luis Fernando Gil Vázquez

Alumnos:

Fernando Alfonso Caldera Olivas

José Enrique Ibarra Manriquez

Alexis Enrique Noyola Saenz

<b>PRODUCCION</b>	->	<b>PRODUCE</b>
PROGRAMASQL	->	DECLARACION SENTENCIAS end {1}
ACTREGS	->	update id set IGUALACION where EXPRCOND {2}
COLUMNAS	->	id COLUMNAS {3}
COLUMNAS_P	->	, COLUMNAS {4}
COLUMNAS_P	->	Empty {5}
DECLARACION	->	declare idvar TIPO DECLARACION {6}
DECLARACION	->	Empty {7}
DESPLIEGUE	->	print EXPRARIT {8}
DELREG	->	delete from id where EXPRCOND {9}
EXPRESIONES	->	EXPRARIT EXPRESIONES_P {10}
EXPRESIONES_P	->	, EXPRESIONES {11}
EXPRESIONES_P	->	empty {12}
EXPRARIT	->	OPERANDO EXPRARIT_P {13}
EXPRARIT	->	(EXPRARIT) EXPRARIT_P {14}
EXPRARIT_P	->	opsuma EXPRARIT {15}
EXPRARIT_P	->	opmult EXPRARIT {16}
EXPRARIT_P	->	empty {17}
EXPRCOND	->	EXPRREL EXPRLOG {18}
EXPRREL	->	EXPRARIT1 oprel EXPRARIT2 {19}
EXPRLOG	->	and EXPRREL {20}
EXPRLOG	->	or EXPRREL {21}
EXPRLOG	->	empty {22}
ELIMTAB	->	drop table id {23}
IFELSE	->	if EXPRCOND begin SENTENCIAS end IFELSE' {24}
IFELSE'	->	else begin SENTENCIAS end {25}
IFELSE'	->	empty {26}
IGUALACION	->	id opasig EXPRARIT IGUALACIONP {27}
IGUALACIONP	->	, IGUALACION {28}
IGUALACIONP	->	empty {29}
INSERCIÓN	->	insert into id ( COLUMNAS ) values ( EXPRESIONES ) {30}
LISTAIDS1	->	, id LISTAIDS2 {31}
LISTAIDS	->	empty {32}
NULO	->	null {33}
NULO	->	not null {34}
NULO	->	empty {35}
OPERANDO	->	num {36}
OPERANDO	->	num.num {37}
OPERANDO	->	idvar {38}
OPERANDO	->	literal {39}
OPERANDO	->	id {40}
SENTENCIAS	->	SENTENCIA SENTENCIAS {41}
SENTENCIAS	->	empty {42}
SENTENCIA	->	IFELSE
SENTENCIA	->	SENREP
SENTENCIA	->	DESPLIEGUE
SENTENCIA	->	SENTASIG
SENTENCIA	->	SENTSELECT
SENTENCIA	->	DELREG
SENTENCIA	->	INSERCIÓN
SENTENCIA	->	ACTREGS
SENTENCIA	->	TABLA
SENTENCIA	->	ELIMTAB
SENTENCIA	->	SELECTIVA
SELECTIVA	->	Case SELWHEN SELELSE end
SELWHEN	->	When EXPRCOND then SENTENCIA SELWHEN_P
SELWHEN_P	->	SELWHEN
SELWHEN_P	->	empty
SELELSE	->	Else SENTENCIA
SELELSE	->	empty
SENREP	->	While EXPRCOND begin SENTENCIAS end
SENTASIG	->	Assign idvar opasig EXPRARIT
SENTSELECT	->	Select idvar opasig Id SENTSELECT from id where EXPRCOND

SENTSELECTC	->	, idvar opasig id SENTSELECT
SENTSELECTC	->	empty
TIPO	->	int
TIPO	->	float
TIPO	->	Char(num)
TABLA	->	Create table id (TABCOLUMNAS)
TABCOLUMNAS	->	Id TIPO NULO TABCOLUMNAS_P
TABCOLUMNAS_P	->	, TABCOLUMNAS
TABCOLUMNAS_P	->	empty

No.	Acción
1	PROGRAMASQL.tipo := if DECLARACION.tipo == 'VACIO' AND SENTENCIAS.tipo == 'VACIO' then 'VACIO' else 'ERROR.TIPO'
2	ACTREGS.tipo := if(IGUALACION.tipo == 'VACIO' AND EXPRCOND.tipo == 'VACIO' then 'VACIO' else 'ERROR.TIPO'
3	COLUMNAS.tipo := if(COLUMNAS' == 'VACIO') then 'VACIO' else 'ERROR.TIPO'
4	COLUMNAS_P.tipo := if(COLUMNAS == 'VACIO') then 'VACIO' else 'ERROR_TIPO'
5	COLUMNAS_P.tipo := 'ERROR.TIPO'
6	DECLARACION.tipo := if(TIPO.TIPO == 'VACIO' AND DECLARACION.TIPO == 'VACIO') then 'VACIO' else 'ERROR.TIPO'
7	DECLARACION.tipo := 'ERROR.TIPO'
8	DESPLIEGUE.tipo := if ( EXPRARIT.TIPO == 'VACIO') then 'VACIO'
9	DELREG.tipo := if ( EXPRCOND.TIPO == 'VACIO' ) then 'VACIO'
10	EXPRESIONES.tipo := if ( EXPRARIT == 'VACIO' AND EXPRESIONES' == 'VACIO') then 'VACIO'
11	EXPRESIONES_P.tipo := if ( EXPRESIONES = 'VACIO' ) then 'VACIO'
12	EXPRESIONES_P.tipo := 'ERROR_TIPO'
13	EXPRARIT.tipo := if ( OPERANDO == 'VACIO' AND EXPRARIT' == 'VACIO') then 'VACIO'
14	EXPRARIT.tipo := if ( EXPRARIT == 'VACIO' AND EXPRARIT' == 'VACIO' ) then 'VACIO'
15	EXPRARIT_P.tipo := if ( EXPRARIT == 'VACIO' ) then 'VACIO'
16	EXPRARIT_P.tipo := if ( EXPRARIT == 'VACIO') then 'VACIO'
17	EXPRARIT_P.tipo := 'ERROR_TIPO'
18	EXPRCOND.tipo := if ( EXPRREL == 'VACIO' AND EXPRLOG == 'VACIO' ) then 'VACIO'
19	EXPRREL.tipo := if tiposCompatibles(EXPRARIT1.tipo, EXPRARIT2.tipo) and EXPRARIT1.tipo != ERROR_TIPO and EXPRARIT2.tipo != ERROR_TIPO VACIO Else ERROR_TIPO
20	EXPRLOG.tipo := EXPRREL.tipo
21	EXPRLOG.tipo := EXPRREL.tipo
22	EXPRLOG.tipo := VACIO
23	ELIMTAB.tipo := if buscaTipo(id.entrada) == 'tabla' then VACIO else ERROR_TIPO

24	IFELSE.tipo := if EXPRCOND.tipo == 'boolean' and SENTENCIAS.tipo == VACIO and IFELSE_P.tipo == VACIO then VACIO else ERROR_TIPO
25	IFELSE_P.tipo := SENTENCIAS.tipo
26	IFELSE_P.tipo := VACIO
27	IGUALACION.tipo := if tiposCompatibles(buscaTipo(id.entrada), EXPRARIT.tipo) and IGUALACIONP.tipo == VACIO then VACIO else ERROR_TIPO
28	IGUALACIONP.tipo := IGUALACION.tipo
29	IGUALACIONP.tipo := VACIO
30	INSERCIÓN.tipo := if checarArchivo(id.lexema + '.db') and COLUMNAS.tipo == VACIO and EXPRESIONES.tipo == VACIO then VACIO else ERROR_TIPO
31	LISTAIDS1.tipo := LISTAIDS2.tipo
32	LISTAIDS.tipo := VACIO
33	NULO.tipo := VACIO
34	NULO.tipo := VACIO
35	NULO.tipo := VACIO
36	OPERANDO.tipo := 'integer'
37	OPERANDO.tipo := 'real'
38	OPERANDO.tipo := buscaTipo(idvar.entrada)
39	OPERANDO.tipo := 'char(' + literal.lexema.length() + ')'
40	OPERANDO.tipo := buscaTipo(id.entrada)
41	SENTENCIAS1.tipo := if SENTENCIA.tipo == VACIO and SENTENCIAS2.tipo == VACIO then VACIO else ERROR_TIPO
42	SENTENCIAS.tipo := VACIO

SENTENCIA	->	<pre> IFELSE {1}   SENREP {2}   DESPLIEGUE {3}   SENTASIG {4}   SENSELECT {5}   DELREG {6}   INSERCIÓN {7}   ACTREGS {8}   TABLA {9}   ELIMTAB {10}   SELECTIVA {11}  SENTENCIA.tipo:=  {1}: if IFELSE.tipo == VACIO then     'VACIO' else     'ERROR_TIPO' {2}: if SENREP.tipo == VACIO then     'VACIO' else     'ERROR_TIPO'  {3}: if DESPLIEGUE.tipo == VACIO then     'VACIO' else     'ERROR_TIPO'  {4}: if SENTASIG.tipo == VACIO then     'VACIO' else     'ERROR_TIPO'  {5}: if SENSELECT.tipo == VACIO then     'VACIO' else     'ERROR_TIPO'  {6}: if DELREG.tipo == VACIO then     'VACIO' else     'ERROR_TIPO'  {7}: if INSERCIÓN.tipo == VACIO then     'VACIO' else     'ERROR_TIPO'  {8}: if ACTREGS.tipo == VACIO then     'VACIO' </pre>
-----------	----	---

		<pre> else     'ERROR_TIPO'  {9}: if TABLA.tipo == VACIO then     'VACIO' else     'ERROR_TIPO'  {10}: if ELIMTAB.tipo == VACIO then     'VACIO' else     'ERROR_TIPO'  {11}: if SELECTIVA.tipo == VACIO then     'VACIO' else     'ERROR_TIPO' </pre>
SELECTIVA	->	<pre> <b>case</b> SELWHEN SELELSE <b>end</b> {12}  SELECTIVA.tipo:= {12}: if SELWHEN.tipo == VACIO and SELELSE.tipo == VACIO then     'VACIO' else     'ERROR_TIPO' </pre>
SELWHEN	->	<pre> <b>when</b> EXPRCOND <b>then</b> SENTENCIA SELWHEN' {13}  SELWHEN.tipo:= {13}: if EXPRCOND.tipo == 'booleano' and SENTENCIA.tipo == VACIO then     'VACIO' else     'ERROR_TIPO' </pre>
SELWHEN'	->	<pre> SELWHEN {14}   <b>empty</b> {15}  SELWHEN'.tipo:=  {14}: if SELWHEN.tipo == VACIO then     'VACIO' else     'ERROR_TIPO'  {15}: SELWHEN'.tipo := </pre>

		<b>'VACIO'</b>
SELELSE	->	<b>else</b> SENTENCIA {16}   <b>empty</b> {17}  SELELSE.tipo := {16}: if SENTENCIA.tipo == VACIO then <b>'VACIO'</b> else <b>'ERROR_TIPO'</b>  {17}: SELELSE.tipo := <b>'VACIO'</b>
SENREP	->	<b>while</b> EXPRCOND <b>begin</b> SENTENCIAS <b>end</b> {18} SENREP.tipo:= {18}: if EXPRCOND.tipo == <b>'booleano'</b> and SENTENCIAS.tipo == VACIO then <b>'VACIO'</b>
SENTASIG	->	<b>assign idvar opasig</b> EXPRARIT {19}  SENTASIG.tipo:= {19}: if EXPRARIT.tipo == VACIO then <b>'VACIO'</b> else <b>'ERROR_TIPO'</b>
SENTSELECT	->	<b>select idvar opasig id</b> SENTSELECTC <b>from id</b> <b>where</b> EXPRCOND {20} SENTSELECT.tipo := {20}: if SENTSELECT.tipo == VACIO and EXPRCOND.tipo == <b>'booleano'</b> then <b>'VACIO'</b> else <b>'ERROR_TIPO'</b>
SENTSELECTC	->	<b>, idvar opasig id</b> SENTSELECTC {21}   <b>empty</b> {22}  SENTSELECTC.tipo:= {21}: if SENTSELECTC.tipo == VACIO then <b>'VACIO'</b> else <b>'ERROR_TIPO'</b>  {22}: SENTSELECTC.tipo

		<code>:= 'VACIO'</code>
TIPO	->	<code>Int {23}   float {24}  </code> <code>char ( num ) {25}</code>  <code>{23}: TIPO.tipo := Int</code>  <code>{24}: TIPO.tipo := float</code>  <code>{25}: TIPO.tipo :=</code> <code>char(num.lexico)</code>
TABLA	->	<code>create table id (</code> <code>TABCOLUMNAS ) {26}</code>  <code>TABLA.tipo :=</code> <code>{26}: if</code> <code>TABLACOLUMNAS.tipo ==</code> <code>VACIO then</code> <code>    'VACIO'</code> <code>    else</code> <code>        'ERROR_TIPO'</code>
TABCOLUMNAS	->	<code>id TIPO NULO</code> <code>TABCOLUMNAS' {27}</code>  <code>TABCOLUMNAS.tipo :=</code> <code>{27}: if TIPO.tipo !=</code> <code>'ERROR_TIPO' and</code> <code>NULO.tipo !=</code> <code>'ERROR_TIPO' and</code> <code>TABCOLUMNAS'.tipo ==</code> <code>VACIO then</code> <code>    'VACIO'</code> <code>    else</code> <code>        'ERROR_TIPO'</code>
TABCOLUMNAS'	->	<code>, TABCOLUMNAS {28} </code> <code>empty {29}</code>  <code>TABCOLUMNAS'.tipo:=</code> <code>{28}: if</code> <code>TABCOLUMNAS.tipo ==</code> <code>VACIO then</code> <code>    'VACIO'</code> <code>    else</code> <code>        'ERROR_TIPO'</code>  <code>{29}: TABCOLUMNAS.tipo</code> <code>:= EMPTY</code>