Analizador Sintáctico

Ariana Bermúdez,Ximena Bolaños, Dylan Rodríguez

Instituto Tecnológico de Costa Rica

May 29, 2017

Análisis Sintáctico

Se hizo un analizador sintáctico con la ayuda de la herramienta de Bison, para el lenguaje C y que corre en C, este analizador trabaja en conjunto con Flex, para tomar los tokens que este le otorga y revisar con las gramáticas que les sean ingresadas.

Bison

Bison convierte de una gramática libre de contexto a un analizador sintáctico que emplea las tablas de Parsing LALR(1), siendo:

- L: Left algo
- A: ...
- L: ...
- R: rightmost
- (1): donde este uno significa que tiene como lookahead solo un símbolo.

Cabe destacar que Bison es compatible con Yacc. Sirve con C, C++ y Java.

Código

```
char * lie ;
double time , me = ! 0XFACE ,
not ; int rested , get , out ;
main ( ly , die ) char ly , * * die ; {
signed char lotte ,
dear ; ( char ) lotte -- ;
for ( get = ! me ; ; not ) {
1 - out & out ; lie ; {
char lotte , my = dear ,
* * let = ! ! me * ! not + ++ die
( char * ) ( lie =
"The gloves are OFF this time, I detest you, snot\n\0sed GEEK!" ) ;
do { not = * lie ++ & 0xF00L * ! me ;
```

Código

```
( char * ) lie - 1 * ! ( not = atoi ( let
[ get - me ?
( char ) lotte -
( char ) lotte : my - * ( char * ) lie - -
’I’ - * ( char * ) lie - - ’U’ -
’I’ - ( long ) - 4 - ’U’ ] ) - ! !
( time = out = ’a’ ) ) ; } while ( my - dear
&& ’I’ - 1l - get - ’a’ ) ; break ; } }
( char ) * lie ++ ;
( char ) * lie ++ , ( char ) * lie ++ ; hell : 0 , ( char ) * lie ;
get * out * ( short ) ly - 0 - ’R’ - get - ’a’ ^ rested ;
do { auto * eroticism ,
that ; puts ( * ( out
```

Código

```
- 'c'
- ( 'P' - 'S' ) + die + - 2 ) ) ; } while ( ! "you're at it" ) ;
for ( * ( ( char * ) & lotte ) ^=
( char ) lotte ; ( ( char * ) lie - ly ) [ ( char ) ++ lotte +
! ! 0xBABE ] ; ) { if ( 'I' - lie [ 2 + ( char ) lotte ] ) { 'I' - 1l * * * die ; }
else { if ( 'I' * get * out * ( 'I' - 1l * * die [ 2 ] ) ) * ( ( char * ) & lotte ) -=
'4' - ( 'I' - 1l ) ; not ; for ( get = !
get ; ! out ; ( char ) * lie & 0xD0 - ! not ) return ! !
( char ) lotte ; }
( char ) lotte ;
do { not * putchar ( lie [ out
* ! not * ! ! me + ( char ) lotte ] ) ;
not ; for ( ; ! 'a' ; ) ; } while (
```

Código

```
( char * ) lie - ( char * ) lie ) ; {
register this ; switch ( ( char ) lie
[ ( char ) lotte ] - 1 * ! out ) {
char * les , get = 0xFF , my ; case ’ ’ :
* ( ( char * ) & lotte ) += 15 ; ! not + ( char ) * lie * ’s’ ;
this + 1 + not ; default : 0xF + ( char * ) lie ; } } }
get - ! out ;
if ( not -- )
goto hell ;
```

Código con los Errores

```
char * lie ;
double time , me = ! 0XFACE ,
not ; int rested , get , out ;
main ( ly , die ) char ly , * * die ; {
signed char lotte ,
dear ; ( char ) lotte -- ;
for ( get = ! me ; ; not ) {
1 - out & out ; lie ; {
char lotte , my = dear ,
* * let = ! ! me * ! not + ++ die
/*Pruebas/love.c:11:6 syntax error, unexpected CHAR*/
/*Pruebas/love.c:11:10 syntax error, unexpected RIGHT_PARENTHESIS*/
```

Código con los Errores

```
( char * ) ( lie =
"The gloves are OFF this time, I detest you, snot\n\0sed GEEK!" ) ;
/*Pruebas/love.c:13:2 syntax error, unexpected DO*/
do { not = * lie ++ & 0xF00L * ! me ;
( char * ) lie - 1 * ! ( not = atoi ( let
[ get - me ?
( char ) lotte -
( char ) lotte : my - * ( char * ) lie - -
'I' - * ( char * ) lie - - 'U' -
'I' - ( long ) - 4 - 'U' ] ) - ! !
/*Pruebas/love.c:20:32 syntax error, unexpected WHILE*/
/*Pruebas/love.c:20:39 syntax error, unexpected MINUS*/
( time = out = 'a' ) ) ; } while ( my - dear
```

Código con los Errores

```
/*Pruebas/love.c:21:2 syntax error, unexpected AND_OP*/
/*Pruebas/love.c:21:6 syntax error, unexpected CONSTANT*/
/*Pruebas/love.c:21:8 syntax error, unexpected MINUS*/
/*Pruebas/love.c:21:11 syntax error, unexpected CONSTANT*/
/*Pruebas/love.c:21:13 syntax error, unexpected MINUS*/
/*Pruebas/love.c:21:25 syntax error, unexpected RIGHT_PARENTHESIS*/
/*Pruebas/love.c:21:33 syntax error, unexpected BREAK*/
/*Pruebas/love.c:21:37 syntax error, unexpected RIGHT_BRACKET*/
/*Pruebas/love.c:21:39 syntax error, unexpected RIGHT_BRACKET*/
&& 'I' - 11 - get - 'a' ) ; break ; } }
( char ) * lie ++ ;
( char ) * lie ++ , ( char ) * lie ++ ; hell : 0 , ( char ) * lie ;
/*Pruebas/love.c:24:26 syntax error, unexpected MINUS*/
```

Código con los Errores

```
/*Pruebas/love.c:24:28 syntax error, unexpected INTEGER*/
/*Pruebas/love.c:24:30 syntax error, unexpected MINUS*/
/*Pruebas/love.c:24:34 syntax error, unexpected CONSTANT*/
/*Pruebas/love.c:24:36 syntax error, unexpected MINUS*/
/*Pruebas/love.c:24:48 syntax error, unexpected UP_ARROW*/
get * out * ( short ) ly - 0 - ’R’ - get - ’a’ ^ rested ;
/*Pruebas/love.c:25:2 syntax error, unexpected DO*/
do { auto * eroticism ,
that ; puts ( * ( out
- ’c’
/*Pruebas/love.c:28:41 syntax error, unexpected WHILE*/
/*Pruebas/love.c:28:45 syntax error, unexpected EXCLAMATION*/
/*Pruebas/love.c:28:60 syntax error, unexpected LITERAL*/
```

Código con los Errores

```
/*Pruebas/love.c:28:62 syntax error, unexpected RIGHT_PARENTHESIS*/
- ( ’P’ - ’S’ ) + die + - 2 ) ) ; } while ( ! "you’re at it" ) ;
/*Pruebas/love.c:29:3 syntax error, unexpected FOR*/
/*Pruebas/love.c:29:33 syntax error, unexpected XOR_ASSIGN*/
for ( * ( ( char * ) & lotte ) ^=
/*Pruebas/love.c:30:35 syntax error, unexpected MINUS*/
/*Pruebas/love.c:30:40 syntax error, unexpected RIGHT_PARENTHESIS*/
/*Pruebas/love.c:30:42 syntax error, unexpected LEFT_SBRACKET*/
( char ) lotte ; ( ( char * ) lie - ly ) [ ( char ) ++ lotte +
/*Pruebas/love.c:31:12 syntax error, unexpected RIGHT_SBRACKET*/
/*Pruebas/love.c:31:16 syntax error, unexpected RIGHT_PARENTHESIS*/
! ! 0xBABE ] ; ) { if ( ’I’ - lie [ 2 + ( char ) lotte ] ) { ’I’ - 11 * * * die ; }
else { if ( ’I’ * get * out * ( ’I’ - 11 * * die [ 2 ] ) ) * ( ( char * ) & lotte ) -=
```

Código con los Errores

```
’4’ - ( ’I’ - 11 ) ; not ; for ( get = !
get ; ! out ; ( char ) * lie & 0xD0 - ! not ) return ! !
( char ) lotte ; }
( char ) lotte ;
do { not * putchar ( lie [ out
* ! not * ! ! me + ( char ) lotte ] ) ;
not ; for ( ; ! ’a’ ; ) ; } while (
( char * ) lie - ( char * ) lie ) ; {
register this ; switch ( ( char ) lie
[ ( char ) lotte ] - 1 * ! out ) {
char * les , get = 0xFF , my ; case ’ ’ :
* ( ( char * ) & lotte ) += 15 ; ! not + ( char ) * lie * ’s’ ;
this + 1 + not ; default : 0xF + ( char * ) lie ; } } }
```

Código con los Errores

```
get - ! out ;
/*Pruebas/love.c:47:2 syntax error, unexpected IF*/
if ( not -- )
/*Pruebas/love.c:48:4 syntax error, unexpected GOTO*/
goto hell ;
/*Pruebas/love.c:49:27 syntax error, unexpected RIGHT_BRACKET*/
```