Analizador Sintáctico

Ariana Bermúdez,Ximena Bolaños, Dylan Rodríguez

Instituto Tecnológico de Costa Rica

May 26, 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\0
    sed GEEK!" ) ;
do { not = * lie ++ & 0xF00L * ! me ;
```

Código Preprocesado (Sin Pretty Print)

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

Código Preprocesado (Sin Pretty Print)

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

Código Preprocesado (Sin Pretty Print)

```
! 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 ; } } }
get - ! out ;
if ( not -- )
goto hell ;
exit ( ( char ) lotte ) ; }
```

Código con los Errores

```
char * lie ;
double time , me = ! 0XFACE ,
not ; int rested , get , out
/*Pruebas/love.c:4:23 syntax error, found "char"*/
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 =
```

Código con los Errores

```
"The gloves are OFF this time, I detest you, snot\n\0
    sed GEEK!" ) ;
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' ] ) - ! !
( 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
```

Código con los Errores

```
/*Pruebas/love.c:25:3 syntax error, found "do"*/
do { auto * eroticism ,
that ; puts ( * ( out
- 'c'
/*Pruebas/love.c:28:42 syntax error, found "while"*/
/*Pruebas/love.c:28:46 syntax error, found "!"*/
/*Pruebas/love.c:28:61 syntax error, found ""you're at
    it""*/
/*Pruebas/love.c:28:63 syntax error, found ")"*/
- ( 'P' - 'S' ) + die + - 2 ) ) ; } while ( ! "you're
    at it" ) ;
/*Pruebas/love.c:29:4 syntax error, found "for"*/
/*Pruebas/love.c:29:34 syntax error, found "^="*/
for ( * ( ( char * ) & lotte ) ^=
/*Pruebas/love.c:30:36 syntax error, found "-"*/
```

Código con los Errores

```
/*Pruebas/love.c:30:41 syntax error, found ")"*/
/*Pruebas/love.c:30:43 syntax error, found "["*/
( char ) lotte ; ( ( char * ) lie - ly ) [ ( char ) ++
    lotte +
/*Pruebas/love.c:31:13 syntax error, found "]"*/
/*Pruebas/love.c:31:17 syntax error, found ")"*/
! ! 0xBABE ] ; ) { if ( 'l' - lie [ 2 + ( char ) lotte
    ] ) { 'l' - 1l * * * die ; }
else { if ( 'l' * get * out * ( 'l' - 1l * * die [ 2 ]
    ) ) * ( ( char * ) & lotte ) -=
'4' - ( 'l' - 1l ) ; not ; for ( get = !
get ; ! out ; ( char ) * lie & 0xD0 - ! not ) return !
    !
( char ) lotte ; }
( char ) lotte ;
do { not * putchar ( lie [ out
* ! not * ! ! me + ( char ) lotte ] ) ;
```

Código con los Errores

```
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 ; } } }
get - ! out ;
/*Pruebas/love.c:47:3 syntax error, found "if"*/
if ( not -- )
/*Pruebas/love.c:48:5 syntax error, found "goto"*/
goto hell ;
/*Pruebas/love.c:49:28 syntax error, found "}"*/
```

Código con los Errores

```
exit ( ( char ) lotte ) ; }
```