



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

## FACULTAD DE INGENIERÍA

### DIVISIÓN DE INGENIERÍA ELÉCTRICA

#### Sistemas de comunicaciones

Programa: Analizador sintáctico descendente

Profesor: ING. Laura Sandoval Montaña

Grupo: 03

Alumno:		Pacheco Saavedra Angel Gael	421027688
---------	--	--------------------------------	-----------

Fecha de entrega	09/11/2023
------------------	------------

	Semestre: 2024-1
--	------------------



# Descripción del problema

## Objetivo

Construir, en un mismo programa, los analizadores Léxico y Sintáctico Descendente Recursivo que revisen programas escritos en el lenguaje definido por la gramática.

El analizador sintáctico debe ser capaz de reconocer la estructura jerárquica del código fuente y determinar si cumple con las reglas gramaticales establecidas, por medio de la cadena de átomos. En caso de encontrar un error sintáctico, debe informar de manera clara y precisa sobre la naturaleza del problema, contribuyendo así a la corrección eficiente del código fuente. También debe ser capaz de continuar con el análisis si encuentra un error, guardando el error en una lista y seguir con las funciones como si el átomo fuera correcto.

## Conjuntos de selección

1.  $\langle \text{Programa} \rangle \rightarrow \langle \text{Func} \rangle \langle \text{otraFunc} \rangle$ 
  - Predicción: {f, i, g, \$}
2.  $\langle \text{otraFunc} \rangle \rightarrow \xi$ 
  - Predicción: {f, i, g}
3.  $\langle \text{otraFunc} \rangle \rightarrow \langle \text{Func} \rangle \langle \text{otraFunc} \rangle$ 
  - Predicción: { \$ }
4.  $\langle \text{Func} \rangle \rightarrow \langle \text{TipoF} \rangle a(\langle \text{listArg} \rangle) \{ \langle \text{Cuerpo} \rangle \}$ 
  - Predicción: {f, i, g}
5.  $\langle \text{TipoF} \rangle \rightarrow f$ 
  - Predicción: {f}
6.  $\langle \text{TipoF} \rangle \rightarrow i$ 
  - Predicción: {i}
7.  $\langle \text{TipoF} \rangle \rightarrow g$ 
  - Predicción: {g}
8.  $\langle \text{listArg} \rangle \rightarrow \xi$ 
  - Predicción: { ) }
9.  $\langle \text{listArg} \rangle \rightarrow \langle \text{Tipo} \rangle a \langle \text{otroArg} \rangle$ 
  - Predicción: {f, i}
10.  $\langle \text{otroArg} \rangle \rightarrow \xi$ 
  - Predicción: { ) }
11.  $\langle \text{otroArg} \rangle \rightarrow , \langle \text{Tipo} \rangle a \langle \text{otroArg} \rangle$ 
  - Predicción: { , }
12.  $\langle \text{Cuerpo} \rangle \rightarrow \langle \text{listDecl} \rangle \langle \text{listaS} \rangle$ 
  - Predicción: {f, i, g,  $\xi$ , \$}
13.  $\langle \text{listDecl} \rangle \rightarrow \xi$

- Predicción: {a,u,t,b,q,z,s,p[,]}

14.  $\langle \text{listDecl} \rangle \rightarrow \langle \text{Decl} \rangle \langle \text{listDecl} \rangle$

- Predicción: {f, i, \$}

15.  $\langle \text{Decl} \rangle \rightarrow \langle \text{Tipo} \rangle a \langle \text{valorIni} \rangle \langle \text{listaVar} \rangle ;$

- Predicción: {f, i, }

16.  $\langle \text{Tipo} \rangle \rightarrow f$

- Predicción: {f}

17.  $\langle \text{Tipo} \rangle \rightarrow i$

- Predicción: {i}

18.  $\langle \text{valorIni} \rangle \rightarrow = \langle \text{tipoVal} \rangle$

- Predicción: {= }

19.  $\langle \text{valorIni} \rangle \rightarrow \xi$

- Predicción: {, ;}

20.  $\langle \text{listaVar} \rangle \rightarrow \xi$

- Predicción: {;}

21.  $\langle \text{listaVar} \rangle \rightarrow , a \langle \text{valorIni} \rangle \langle \text{listaVar} \rangle$

- Predicción: {, }

22.  $\langle \text{tipoVal} \rangle \rightarrow n$

- Predicción: {n}

23.  $\langle \text{tipoVal} \rangle \rightarrow v$

- Predicción: {v}

24.  $\langle \text{Asig} \rangle \rightarrow a \langle \text{opArit} \rangle E ;$

- Predicción: {a}

25.  $\langle \text{opArit} \rangle \rightarrow =$

- Predicción: {=}

26.  $\langle \text{opArit} \rangle \rightarrow x$

- Predicción: {x}

27.  $\langle \text{opArit} \rangle \rightarrow y$

- Predicción: {y}

28.  $\langle \text{opArit} \rangle \rightarrow k$

- Predicción: {k}

29.  $\langle \text{opArit} \rangle \rightarrow r$

- Predicción: {r}

30.  $\langle \text{opArit} \rangle \rightarrow o$

- Predicción:  $\{o\}$

31.  $E \rightarrow T E'$

- Predicción:  $\{(, a, n, [ \}$

32.  $E' \rightarrow + T E'$

- Predicción:  $\{+\}$

33.  $E' \rightarrow - T E'$

- Predicción:  $\{-\}$

34.  $E' \rightarrow \xi$

- Predicción:  $\{ ) \}$

35.  $T \rightarrow F T'$

- Predicción:  $\{(, a, n, [ \}$

36.  $T' \rightarrow * F T'$

- Predicción:  $\{*\}$

37.  $T' \rightarrow / F T'$

- Predicción:  $\{/ \}$

38.  $T' \rightarrow \% F T'$

- Predicción:  $\{\% \}$

39.  $T' \rightarrow \xi$

- Predicción:  $\{+ - \}$

40.  $F \rightarrow ( E )$

- Predicción:  $\{( \}$

41.  $F \rightarrow a$

- Predicción:  $\{a\}$

42.  $F \rightarrow n$

- Predicción:  $\{n\}$

43.  $F \rightarrow \langle \text{Llama} \rangle$

- Predicción:  $\{[ \}$

44.  $\langle \text{Llama} \rangle \rightarrow [a(\langle \text{listP} \rangle)]$

- Predicción:  $\{[ \}$

45.  $R \rightarrow E \langle \text{opRel} \rangle E$

predicción  $\{ ( a n [ \}$

46.  $\langle \text{opRel} \rangle \rightarrow >$

- Predicción: {>}

47. <opRel> → <

- Predicción: {<}

48. <opRel> → e

- Predicción: {e}

49. <opRel> → d

- Predicción: {d}

50. <opRel> → m

- Predicción: {m}

51. <opRel> → w

- Predicción: {w}

52. <Sent> → <Asig>

- Predicción: {a}

53. <Sent> → Q

- Predicción: {u}

54. <Sent> → <Ret>

- Predicción: {t}

55. <Sent> → <Trocar>

- Predicción: {b}

56. <Sent> → <Enq>

- Predicción: {q}

57. <Sent> → <Faz>

- Predicción: {z}

58. <Sent> → <Se>

- Predicción: {S}

59. <Sent> → <Para>

- Predicción: {p}

60. <Sent> → <Llama>

- Predicción: {}

61. <listaS> → <Sent><listaS>

- Predicción: {a, u, t, b, q, z, s, p, [ }

62. <listaS> → ξ

- Predicción: {# , } }

63.  $\langle \text{expLogica} \rangle \rightarrow !\langle \text{expRel} \rangle$
- Predicción:  $\{!\}$
64.  $\langle \text{expLogica} \rangle \rightarrow R\langle \text{expLog} \rangle$
- Predicción:  $\{(\text{a}, \text{n}, [ \}$
65.  $\langle \text{expLog} \rangle \rightarrow \langle \text{opLog} \rangle R$
- Predicción:  $\{h, j\}$
66.  $\langle \text{expLog} \rangle \rightarrow \xi$
- Predicción:  $\{\}$
67.  $\langle \text{expRel} \rangle \rightarrow \{R\}$
- Predicción:  $\{\{\}$
68.  $\langle \text{expRel} \rangle \rightarrow E$
- Predicción:  $\{(\text{a}, \text{n}, [ \}$
69.  $\langle \text{opLog} \rangle \rightarrow h$
- Predicción:  $\{h\}$
70.  $\langle \text{opLog} \rangle \rightarrow j$
- Predicción:  $\{j\}$
71.  $\langle \text{Enq} \rangle \rightarrow q(\langle \text{expLogica} \rangle) \# \langle \text{listaS} \rangle \#$
- Predicción:  $\{q\}$
72.  $\langle \text{Se} \rangle \rightarrow s(\langle \text{expLogica} \rangle) \# \langle \text{listaS} \rangle \#$
- Predicción:  $\{s\}$
73.  $Q \rightarrow u;$
- Predicción:  $\{u\}$
74.  $\langle \text{Faz} \rangle \rightarrow z \# \langle \text{listas} \rangle \# q(\langle \text{expLogica} \rangle);$
- Predicción:  $\{z\}$
75.  $\langle \text{Ret} \rangle \rightarrow t \langle \text{valRet} \rangle;$
- Predicción:  $\{t\}$
76.  $\langle \text{valRet} \rangle \rightarrow E$
- Predicción:  $\{(\text{a}, \text{n}, [ \}$
77.  $\langle \text{valRet} \rangle \rightarrow \{ \langle \text{expCad} \rangle \}$
- Predicción:  $\{\{\}$
78.  $\langle \text{valRet} \rangle \rightarrow \xi$
- Predicción:  $\{;\}$
79.  $\langle \text{expCad} \rangle \rightarrow a \langle \text{opCad} \rangle$

- Predicción: {a}
80.  $\langle \text{expCad} \rangle \rightarrow v \langle \text{opCad} \rangle$
- Predicción: {v}
81.  $\langle \text{opCad} \rangle \rightarrow l \langle \text{expCad} \rangle$
- Predicción: {l}
82.  $\langle \text{opCad} \rangle \rightarrow h \langle \text{expCad} \rangle$
- Predicción: {h}
83.  $\langle \text{opCad} \rangle \rightarrow \xi$
- Predicción: { } }
84.  $\langle \text{Para} \rangle \rightarrow pa[n,n] \# \langle \text{listaS} \rangle \#$
- Predicción: {p}
85.  $\langle \text{casos} \rangle \rightarrow \xi$
- Predicción: {#}
86.  $\langle \text{casos} \rangle \rightarrow c(n) \{ \langle \text{listaS} \rangle \} \langle \text{casos} \rangle$
- Predicción: {c}
87.  $\langle \text{casos} \rangle \rightarrow c(n) \{ \langle \text{listaS} \rangle \} \langle \text{casos} \rangle$
- Predicción: { ( }
88.  $\langle \text{listP} \rangle \rightarrow \xi$
- Predicción: { ) }
89.  $\langle \text{listP} \rangle \rightarrow E \langle \text{Param} \rangle$
- Predicción: { (, a, n, [ }
90.  $\langle \text{listP} \rangle \rightarrow \{ \langle \text{expCad} \rangle \} \langle \text{Param} \rangle$
- Predicción: { { }
91.  $\langle \text{Param} \rangle \rightarrow , \langle \text{listP} \rangle$
- Predicción: { , }
92.  $\langle \text{Param} \rangle \rightarrow \xi$
- Predicción: { } }

## Indicaciones de cómo correr el programa.

Sí, tienes razón. En el caso de usar Lex, la secuencia de comandos sería un poco diferente. Si estás utilizando Lex para generar el archivo C a partir de tu archivo `.l`, y luego quieres compilarlo con GCC, la secuencia de comandos podría verse así:

### 1. **\*\*Generar el archivo C con Lex:\*\***

Utiliza Lex para generar el archivo C a partir de tu archivo ``l``:

`Lex analizador.l`

Esto generará un archivo llamado ``analizador.c``.

### 2. **\*\*Compilar el Programa C con GCC:\*\***

Ahora, compila el programa C utilizando GCC:

```
gcc lex.yy.c -o analizador -lfl
```

En este caso, ``-lfl`` se utiliza para enlazar la biblioteca de Flex (o Lex en este caso).

### 3. Ejecutar el Programa:

Después de compilar, ejecuta el programa proporcionando un archivo fuente como entrada:

```
./analizador archivo_fuente.txt
```

Sustituye ``archivo_fuente.txt`` con el nombre de tu propio archivo fuente.

## Conclusión

El cálculo de conjuntos de selección desempeña un papel fundamental en el diseño de compiladores, ofreciendo beneficios cruciales para el análisis sintáctico. Al identificar estos conjuntos para cada producción gramatical, se optimiza la eficiencia del analizador sintáctico predictivo, permitiendo decisiones precisas basadas en conjuntos de símbolos y evitando ambigüedades en la gramática. Este enfoque no solo mejora el rendimiento del compilador, sino que también guía el diseño del analizador, establece condiciones de parada precisas y es compatible con herramientas de generación automática de analizadores.