

Segundo Programa Análisis léxico-sintáctico

Objetivo

Construir, en un mismo programa, los analizadores Léxico y Sintáctico Descendente que revisen programas escritos en el lenguaje definido por la gramática elaborada en clase y que se presenta al final del documento.

Descripción

- La entrada es un archivo con el programa fuente a analizar que deberá estar escrito en el lenguaje definido por la gramática elaborada en clase, que se presenta al final del documento y deberás completar. Este archivo de entrada se indicará desde la línea de comandos.
- El programa realizará tanto el análisis léxico como el sintáctico. El analizador léxico deberá generar además, la cadena de átomos que será la entrada del analizador sintáctico.
- Deberán adecuar el analizador léxico ya elaborado con lo siguiente:
 - La tabla de clases de componentes léxicos que deberá reconocer es:

Clase	Descripción	Átomo(s)
0	Identificadores (sólo letras minúsculas)	a
1	Palabras reservadas	
2	Operador de asignación =	=
3	Operadores relacionales	
4	Operadores aritméticos	
5	Símbolos especiales , ; [] () @	, ; [] () @
6	Constantes numéricas enteras (base 10)	c
7	Constantes numéricas reales (siempre con .)	n
8	Constante cadena (entre comillas)	s

- El valor en los tokens y los átomos se indican en las siguientes tablas.

Valor	Operador Aritmético	átomo
0	DIVIDE	/
1	MAS	+
2	MENOS	-
3	MULTIPLICA	*

Valor	Op relac.	átomo
0	>	>
1	>=	g
2	<	<
3	<=	p
4	==	q
5	!=	!

Valor	Palabra reservada	átomo
0	ENT	t
1	ESCRIBE	w
2	HAZ	h
3	LEE	l
4	MIENTRAS	m
5	REAL	r
6	SI	i
7	SINO	e

- El átomo del operador de asignación y de los caracteres especiales será el mismo carácter.
- El analizador sintáctico deberá indicar todos los errores sintácticos que encuentre.
- Como resultados, el analizador léxico-sintáctico deberá mostrar el contenido de la tabla de símbolos, los tokens, la tabla de cadenas y la cadena de átomos. Finalmente deberá indicar si está sintácticamente correcto el programa fuente.
- Los errores que vaya encontrando el analizador léxico, los podrá ir mostrando en pantalla o escribirlos en un archivo, así como él o los errores sintácticos.

- El programa deberá estar documentado, con una descripción breve de lo que hace (puede ser el objetivo indicado en este documento), el nombre de quienes elaboraron el programa y fecha de elaboración. Así como lo que hace cada función. Se deberá cuidar mucho la sangría que denota la dependencia de instrucciones.

Entregar:

Un documento con la siguiente estructura:

- Descripción del problema (no del programa).
- Propuesta de solución y fases del desarrollo del sistema: Análisis (planificación, indicando participantes por cada actividad), Diseño e implementación. En el diseño **indicar el conjunto de selección de cada producción**; si resultan conjuntos de selección no disjuntos para producciones de un mismo no-terminal, hacer los ajustes pertinentes para que resulte una gramática LL(1).
- Indicaciones de cómo correr el programa.
- Conclusiones de cada uno de los integrantes del equipo.

Nota: se podrá elaborar individualmente o en equipo de 2

Enviar el documento y sólo el programa fuente definitivo a la plataforma educativa y en su caso, sólo un miembro del equipo. También sería conveniente enviar un archivo fuente de prueba.

Fecha de entrega: 7 de noviembre de 2017.

Gramática del lenguaje definido en clase

1:	$P \rightarrow \langle LF \rangle$	32:	$\langle PC \rangle \rightarrow [\langle BP \rangle]$
2:	$\langle LF \rangle \rightarrow \epsilon$	33:	$A \rightarrow a=E;$
3:	$\langle LF \rangle \rightarrow \langle FUN \rangle \langle LF \rangle$	34:	$W \rightarrow$
4:	$\langle FUN \rangle \rightarrow Va(\langle LA \rangle)[\langle LD \rangle[\langle BP \rangle]]$	35:	$R \rightarrow$
5:	$\langle FUN \rangle \rightarrow a(\langle LA \rangle)[\langle LD \rangle[\langle BP \rangle]]$	36:	$H \rightarrow$
6:	$\langle LA \rangle \rightarrow \epsilon$	37:	$M \rightarrow$
7:	$\langle LA \rangle \rightarrow Va\langle LAP \rangle$	38:	$I \rightarrow$
8:	$\langle LAP \rangle \rightarrow \epsilon$	39:	$\langle REL \rangle \rightarrow E\langle OR \rangle E$
9:	$\langle LAP \rangle \rightarrow ,Va\langle LAP \rangle$	40:	$\langle OR \rangle \rightarrow >$
10:	$\langle LD \rangle \rightarrow \epsilon$	41:	$\langle OR \rangle \rightarrow g$
11:	$\langle LD \rangle \rightarrow D\langle LD \rangle$	42:	$\langle OR \rangle \rightarrow <$
12:	$D \rightarrow VaCL$	43:	$\langle OR \rangle \rightarrow p$
13:	$V \rightarrow t$	44:	$\langle OR \rangle \rightarrow q$
14:	$V \rightarrow r$	45:	$\langle OR \rangle \rightarrow !$
15:	$C \rightarrow \epsilon$	46:	$\langle LP \rangle \rightarrow E\langle LPA \rangle$
16:	$C \rightarrow =N$	47:	$\langle LP \rangle \rightarrow \epsilon$
17:	$N \rightarrow n$	48:	$\langle LPA \rangle \rightarrow \epsilon$
18:	$N \rightarrow c$	49:	$\langle LPA \rangle \rightarrow ,E\langle LPA \rangle$
19:	$L \rightarrow ,aCL$	50:	$E \rightarrow TE'$
20:	$L \rightarrow ;$	51:	$E' \rightarrow +TE'$
21:	$\langle BP \rangle \rightarrow \epsilon$	52:	$E' \rightarrow -TE'$
22:	$\langle BP \rangle \rightarrow \langle PR \rangle \langle BP \rangle$	53:	$E' \rightarrow \epsilon$
23:	$\langle PR \rangle \rightarrow S$	54:	$T \rightarrow FT'$
24:	$\langle PR \rangle \rightarrow \langle PC \rangle$	55:	$T' \rightarrow *FT'$
25:	$S \rightarrow A$	56:	$T' \rightarrow /FT'$
26:	$S \rightarrow W$	57:	$T' \rightarrow \epsilon$
27:	$S \rightarrow R$	58:	$F \rightarrow (E)$
28:	$S \rightarrow H$	59:	$F \rightarrow a$
29:	$S \rightarrow M$	60:	$F \rightarrow n$
30:	$S \rightarrow I$	61:	$F \rightarrow c$
31:	$S \rightarrow @a(\langle LP \rangle);$	62:	$F \rightarrow @a(\langle LP \rangle)$