1a Guía Compiladores

Nombre:

Grupo: Fecha:

-Defina **compilador**

-Cuales son las dos partes de la compilación

1.-

2.-

-Describa las 6 fases de un compilador

-Cuales son los 8 modulos de un compilador

A partir de hoc4 se usan dos etapas en hoc. Cuales son y que hacen ?

1.-

2.-

Para que sirve el **Análisis Léxico**

|  |  |  |
| --- | --- | --- |
| a) Para generar el código en lenguaje objeto | b) Nos dice si una cadena pertenece al lenguaje generado por una gramática | ( ) |
| c) Para dividir una cadena en tokens | d) Los compiladores no lo necesitan nunca |

El \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ comprueba que el orden en que el **analizador léxico** le va entregando los tokens es válido.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a) analizador semantico | b) analizador sintáctico | c) optimizador | d) generador de codigo | ( ) |

Es una *gramática que* tiene cuatro componentes:

1. Un conjunto de componentes léxicos*.*

1. Un conjunto de no terminales.
2. Un conjunto de producciones, en el que cada producción consta de un no terminal, llamado *lado izquierdo* de la producción, una flecha y una secuencia de componentes léxicos y no terminales, o ambos, llamado *lado derecho* de la producción.
3. .La denominación de uno de los no terminales como símbolo *inicial.*

|  |  |  |
| --- | --- | --- |
| *a)* Gramática Asociativa por la izquierda | *b) Gramática recursiva* | ( ) |
| c) Gramática libre de contexto | d) Gramática *ambigua* |  |

Cual de las sigs. opciones no es sinónimo de las otras

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a) Componente léxico | b) no terminal | c) token | d) Simbolo gramatical | ( ) |

Es una gramática donde existe una cadena que tiene mas de un árbol de análisis sintáctico.

|  |  |  |
| --- | --- | --- |
| *a)* Gramática recursiva por la izquierda | *b) Gramática recursiva* | ( ) |
| c) Gramática libre de contexto | d) Gramática *ambigua* |  |

Si Una gramática contiene una regla de producción de la forma A *→* A  entonces es una

|  |  |  |
| --- | --- | --- |
| *a)* Gramática recursiva por la izquierda | *b) Gramática ambigua* | ( ) |
| c) Gramática libre de contexto | d) ninguna de las anteriores |  |

Falso o verdadero (F/V)

1-Análisis sintáctico ***descendente*** *es donde la construcción del árbol de análisis sintáctico se*

*inicia en las hojas y avanza hacia la raíz \_\_\_\_\_.*

2-Análisis sintáctico **a*scendente*** *es donde la construcción del árbol de análisis sintáctico se*

*inicia en las hojas y avanza hacia la raíz \_\_\_\_\_\_.*

*3-La variables en HOC son de tipo entero \_\_\_\_\_\_\_\_.*

*4.-yylex llama a yyparse*

*5.-La* *n*otación posfija es una notación matemática libre de paréntesis y en esta *n*otación los operadores aparecen después de los operandos

Considere la siguiente gramática

*S->* ***0*** *S* ***1*** *|* ***01***

a ) Mostrar una derivación de ***00001111***

b ) Dibuje el árbol de análisis sintáctico para la entrada ***00001111***

Considere la siguiente gramática

S → **b**A

A → **b**B

B → **b**C

C → **ε**

a ) Mostrar una derivación de **bbb**

b ) Dibuje el árbol de análisis sintáctico para la entrada **bbb**

Considere la siguiente gramática

*S* → *A*

*A* → *A***+***A* | *B****++***

*B* → ***y***

a ) Mostrar una derivación de **y + + + y + +**

b ) Dibuje el árbol de análisis sintáctico para la entrada **y + + + y + +**

Considere la siguiente gramática

*l→ l* ***,*** *d | d*

*d→* ***0*** *|* ***1*** *|* ***2*** *|* ***3*** *|* ***4*** *|* ***5*** *|* ***6*** *|* ***7*** *|****8*** *|* ***9***

a ) Mostrar una derivación de ***9,8,7,6,5,4,3,2,1,0***

b ) Dibuje el árbol de análisis sintáctico para la entrada ***9,8,7,6,5,4,3,2,1,0***

Dada la gramática

T={**a**, **b**, +, -, \*, /, (,)}, N={E, T, F} S={E}

P={ E->T | E+T | E-T

T->F | T\*F | T/F

F-> **a** | **b** | (E) }

y la cadena **(a+b)/b**

a) Obtenga una derivación de dicha cadena

b) Dibuje el árbol de análisis sintáctico que corresponde a la cadena mencionada

**Análisis sintáctico predictivo descendente recursivo**

Considere la siguiente gramática

S →**a** | **(** S **)**

Escriba el analizador sintáctico predictivo descendente recursivo

**Ambigüedad**

|  |  |  |
| --- | --- | --- |
| Demostrar que la siguiente **gramática** es **ambigua** | S -> A S | **b** | A -> S A | **a** |

Demostrar que la siguiente gramática es ambigua

A →A **x** B | **x**

B → **x** B | **x**

usando la cadena **xxxxx**

Demostrar que la siguiente gramática es ambigua

S→ **a** S **b** S | **b** S **a** S | **ε**

usando la cadena **abab**

Verificar si las siguientes gramáticas son ambiguas

S → S **+** S | S **-** S | **a**

S → S S **+** | S S **-** | **a**

**Recursividad por la** **izquierda**

Para eliminar la recursividad por la izquierda las producciones

A → A | 

se transforman en

A → 

R → R | **ε**

Ahora considere la siguiente gramática

S→ **(** L **)** | **a**

L→ L **,** S | S

Elimine la recursividad por la izquierda de dicha gramática.

Escriba el analizador sintáctico predictivo descendente recursivo

Escriba la sección de reglas de la especificación de YACC para dicha gramática

**Definiciones dirigidas por la sintaxis**

|  |  |
| --- | --- |
| Producción | Regla Semántica |
| *sec →* **comienza** | *sec.x = 0*  *sec.y = 0* |
| *sec →* *sec1 instr* | *sec.x = sec1.x + instr.dx*  *sec. y = sec1.y + instr.dy* |
| *instr →* **este** | *instr.dx = 1*  *instr.dy = 0* |
| *instr →* **norte** | *instr.dx = 0*  *instr.dy = 1* |
| *instr →* **oeste** | *instr.dx = -1*  *instr.dy = 0* |
| instr → **sur** | *instr.dx = 0*  *instr.dy = -1* |

Dibuje el árbol de análisis sintácticocon anotaciones para la sig cadena

c n e ss oo nnn eee ssss oooo

Escribir la sección de reglas de la especificación de yacc para calcular la posición final del robot.

Escriba una definición dirigida por la sintaxis para evaluar expresiones booleanas.

**Esquemas de traducción**

Escriba un esquema de traducción para convertir una expresión en:

|  |  |
| --- | --- |
| infijo a postfijo | postfijo a infijo |
| infijo a prefijo | prefijo a infijo |

Escriba un esquema de traducción para evaluar expresiones booleanas

Para cada esquema de traducción de arriba escriba la sección de reglas de la especificación de YACC

**Escritura de Gramaticas**

Escribir una gramática que genere todas las cadenas de longitud 4 formadas con los símbolos del alfabeto {a,b,c}

|  |  |  |  |
| --- | --- | --- | --- |
| **Escribir una gramática que sirva para generar las siguientes cadenas** | | | |
| **Especie perro**  **Edad 1**  **Sexo macho**  **Tamaño grande**  **Colores negro , blanco**  **Soy rápido , activo, alegre**  **Aficiones correr, comer** | **Especie gato**  **Edad 2**  **Sexo macho**  **Tamaño mediano**  **Colores negro , blanco , café**  **Soy tranquilo , sociable**  **Aficiones dormir, parrandear, comer** | **Especie perro**  **Edad 2**  **Sexo hembra**  **Tamaño pequeño**  **Colores canela , gris**  **Soy fuerte , alegre, activo.**  **Aficiones aullar** | **Especie gato**  **Edad 2**  **Sexo macho**  **Tamaño grande**  **Colores blanco**  **Soy listo , obediente**  **Aficiones jugar, haraganear** |

|  |  |  |  |
| --- | --- | --- | --- |
| 12.-Escribir una gramática que sirva para generar las siguientes cadenas | | | |
| Etiquetado Nerd  Nivel Junior  Sexo Hombre  Lenguajes Java , C , Logo  Aficiones programar, videogames, comics, hackear, googlear | Etiquetado Geek  Nivel Senior  Sexo Mujer  Lenguajes Pascal , Prolog , SQL  Aficiones chatear, videogames, programar | Etiquetado Nerd  Nivel Junior  Sexo Mujer  Lenguajes PHP , Perl, Java  Aficiones hackear, googlear, gotcha, dormir | Etiquetado Freak  Nivel Senior  Sexo Hombre  Lenguajes Ensamblador, C  Aficiones gotcha, dormir, chatear, comics |

**Expresiones Regulares**

Describa el lenguaje correspondiente a la expresión regular

(0 | 1) ∗ 0

Escriba expresiones regulares para cada uno de lo siguientes.

Las cadenas sobre el alfabeto { a, b, c } , donde la primera a precede a la primera b.

Las cadenas sobre el alfabeto { a, b, c } con un número par de a's.

Los números binarios que son múltiplos de cuatro.

Los números binarios que son mayores que 101.001 .

Las cadenas sobre el alfabeto {a, b , c} que no contienen la sub-cadena contigua baa.

Para cada una de las siguientes expresiones regulares , use la construcción de Thompson para derivar un autómata finito no determinista (NFA) que reconoce el mismo lenguaje.

1.- **aaa**

2.-  **b\*a | bb**

3.- **(ab)∗ab**

4.- **a∗bc∗d**

5.- **(a|bc∗)a∗**

6.- **(a|b)∗**

7.- **a∗|b∗**

8.- **(a∗|b∗)∗**

9.- **((aa)∗(ab)∗(ba)∗(bb)∗)∗**

10.-  **( ( b | a \* c ) x) \* | x \* a**

Para cada una de la NFA de en el ejercicio anterior , utilizar la construcción de conjuntos para derivar un autómata finito determinista equivalente ( DFA )

Para cada uno de los DFA en el ejercicio anterior, utilice el método de partición para derivar un mínimo DFA equivalente.

**YACC**

1.-Los %% se usan para indicar

|  |  |  |
| --- | --- | --- |
| a)inicio de la sección de declaraciones | b)inicio de la sección de reglas | ( ) |
| c)precedencia de los operadores | d)fin del código de soporte |

2.-%token sirve para indicar

|  |  |  |
| --- | --- | --- |
| a)inicio de la sección de declaraciones | d)los no terminales de la gramática | ( ) |
| c)precedencia de los operadores | d)los terminales de la gramática |

.-$$ sirve para indicar

.-$n sirve para indicar

3.-Como le indica el analizador léxico (yylex) al analizador sintáctico (yyparse) que ya no hay mas tokens en la entrada

|  |  |  |
| --- | --- | --- |
| a) retornando cero | b) retornando -1 | ( ) |
| c) almacenando -1 en yylval | d) almacenando 0 en yylval |

4.-Una acción gramatical debe ir entre

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a) comillas | b) paréntesis | c) corchetes | d) llaves | ( ) |

5.-Considere la producción

S : S ' a' S 'b'

$4 a cual de los miembros del lado derecho de la producción se refiere?

|  |  |  |
| --- | --- | --- |
| a)la 'a' | b)la 1er S | ( ) |
| c)la segunda S | d)la 'b' |

Si el codigo de yylex es el siguiente

int yylex() { return getchar(); }

De cuantos caracteres son los tokens?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a) 0 | b) 1 | c) 2 | d) la cantidad de caracteres de los tokens varia | ( ) |

Considere la siguiente gramática (los terminales se indican en negritas)

L-> L**,** D | D

D-> **0** | **1**

Escriba la sección de reglas de la especificación de yacc para dicha gramática

Escriba la especificación de yacc para la gramática

S → U | V

U → T**a**U | T**a**T

V → T**b**V | T**b**T

T → **a**T**b**T | **b**T**a**T | **ε**

Escriba las acciones gramaticales para que imprima el numero de b's en la cadena de entrada

%{

/\*escriba el tipo de los elementos en la pila de yacc \*/

#define YYSTYPE

%}

%%

S : ’(’ B ’)’ {}

;

B : ’(’ B ’)’ { }

| D { $$=$1; }

;

D : { }

| ’b’ D { }

;

%%

Considere la siguiente gramática (los terminales se indican en negritas)

lista->lista **,** figura | figura

figura-> triangulo | cuadrilatero

triangulo-> **lado lado lado**

cuadrilatero-> **lado lado lado lado**

Escriba la sección de reglas de la especificación de yacc para dicha gramática y las acciones semánticas respectivas para que se imprima si un triangulo es equilátero y si un cuadrilátero es un cuadrado

**Análisis Sintáctico Predictivo no Recursivo**

-Para las siguientes GLC construya la tabla Análisis Sintáctico Predictivo no Recursivo

-Use dicho análisis para analizar las cadenas propuestas:

-Muestre el contenido de la pila, la entrada y la acción a realizar

**Problema 1.-**Considere la gramática para generar paréntesis anidados

|  |  |
| --- | --- |
| 1) A → **(** A **)** | 2) A → **a** |

Cadenas propuestas:

**( a )**

**( ( a ) )**

**( ( ( a ) ) )**

**( ( ( ( a ) ) ) )**

**Problema 2.-**Considere la siguiente gramática :

|  |  |  |  |
| --- | --- | --- | --- |
| 1) S → **a** | 2) S → **(** S R | 3) R → **,** S R | 4) R → **)** |

Cadenas propuestas:

**( a )**

**(a , a)**

**(a , a, a)**

**(a , a, a, a )**

**Problema 3.-**Considere la siguiente gramática :

|  |  |  |  |
| --- | --- | --- | --- |
| 1) S → A**a**A**b** | 2) S → B**b**B**a** | 3) A → **ε** | 4) B → **ε** |

Cadenas propuestas:

**ab** y **ba**

**Problema 4.-**Considere la siguiente gramática :

S → A

A → **ε**

A → **bb**A

Cadena propuesta:

**bbbb**