Compilador: Programa que traduce un programa en lenguaje fuente a lenguaje objeto equivalente

Analizador léxico recibe una cadena de entrada y divide en tokens

Token: Componente léxico = conjunto de caracteres con significado colectivo

:= asignación en pascal

Analizador sintáctico: Se asegura que estén en el orden correcto. Crea un árbol de análisis sintáctico. (los recibe uno por uno y va armando el árbol) Validar el programa como bien escrito.

Semántica: significado de las cosas

Analizador semántico: validar que sean el mismo tipo de operandos

Generador de código intermedio: debe ser fácil de generar

Código de 3 de acciones: Dos operadores uno el de asignación y debe tener como máximo 3 operandos y se utilizan variables temporales.

Optimizar: hacer un uso eficiente de los recursos = mejorar la forma de hacer algo

Optimador de código: usar menos líneas de código, ser equivalente al código anterior y al original

Gramática: Es una cuádrupla de componentes léxicos (no terminales, elementos terminales, símbolo inicial y un conjunto de producciones)

Una producción es una regla de sustitución entre lado izquierdo y lado derecho

Una terminal ya no puede ser sustituido ya que están del lado derecho en cambio un no terminal como aparece del lado izquierdo puede ser sustituido.

Derivacion

1-2+3-4

Lista 🡪 lista - digito 🡪 lista + digito - digito 🡪 lista – digito + digito – digito 🡪 digito – digito + digito – digito 🡪 1-2+3-4

Demuestra que la cadena pertenece al lenguaje generado por la gramática

Árbol de análisis sintáctico indica gráficamente como se deriva una cadena de lenguaje

9 – 5 + 2

Cadena +cadena 🡪 cadena – cadena + cadena 🡪9 – 5 + 2

Gramática Ambigua: Se dice que es ambigua si existe una cadena para la cual se pueden dibujar 2 árboles de análisis sintáctico.

Asociatividad sirve para ver que operador es dueño de que operandos (va a trabajar)

Asociatividad de operadores: asociatividad izquierda, asociatividad derecha, no asociatividad

Notación postfija:

Árbol de análisis sintáctico con anotaciones = Árbol decorado

Partes de Yacc: Declaraciones, reglas y código de soporte🡪La sección de regla se parece a un código de traducción🡪Se encuentra entre %%

$$ guarda el valor que se le asigna a la traducción🡪$1 El primer símbolo gramatical del lado derecho de la producción🡪$2el 2do🡪$3 el 3ero

Acción gramatical🡪Acción semántica=fragmento de código y están intercalados en el lado derecho de la producción

Yylex = (yy propiedad de yyac) Es una función, es el analizador léxico🡪Divide la cadena de entrada en tokens🡪divide en componentes léxicos🡪lee de izquierda a derecha y corta cada que encuentra el final de un token y el inicio de otro y se lo envia al analizador sintactico

#define YYSTYPE🡪(S=stack) indica a yyac de que tipo son los elementos en la pila de yacc reemplaza por un doble (o el tipo que se diga)

Yyparse🡪Analizador sintáctico