

Compiladores, 2024-1

Examen parcial 02

Manuel Soto Romero

Javier Enríquez Mendoza

Pedro Ulises Cervantes González

Braulio Aaron Santiago Carrillo

Facultad de Ciencias UNAM

Entrega: 16 de octubre de 2023

1. (25 pts.) Dada la gramática $A \rightarrow (A)A \mid \varepsilon$:

1. Construye los conjuntos Primero y Siguiente para el no terminal A .
2. Muestra que esta gramática es LL(1).

2. (25 pts.) Considera la gramática siguiente:

$$\begin{aligned} \text{lexp} &\rightarrow \text{atom} \mid \text{list} \\ \text{atom} &\rightarrow \text{numero} \mid \text{identificador} \\ \text{list} &\rightarrow (\text{lexp-seq}) \\ \text{lexpr-seq} &\rightarrow \text{lexp-seq lexp} \mid \text{lexp} \end{aligned}$$

1. Elimina la recursión por la izquierda.
2. Construye los conjuntos Primero y Siguiente para los no terminales de la gramática resultante.
3. Muestra que la gramática resultante es LL(1).
4. Construye la tabla de análisis sintáctico LL(1) para la gramática resultante.
5. Muestra las acciones del analizador sintáctico LL(1) correspondiente, dada la cadena de entrada $(a (b (2)) (c))$.

3. (25 pts.)

1. ¿Puede ser ambigua una gramática LL(1)? Justifica tu respuesta.
2. ¿Puede una gramática ambigua ser LL(1)? Justifica tu respuesta.
3. Una gramática ambigua debe ser LL(1)? Justifica tu respuesta.

4. (25 pts.) Un no terminal A es *inútil* si no hay derivación del símbolo inicial para una cadena de tokens en la que aparezca A .

1. Proporciona una formulación matemática de esta propiedad.
2. ¿Es probable que una gramática de lenguaje de programación tenga un símbolo inútil? Explica por qué.
3. Muestra que, si una gramática tiene un símbolo inútil, el cálculo de los conjuntos Primero y Siguiente como se vio en clase puede producir conjuntos que sean demasiado grandes para construir con precisión una tabla de análisis sintáctico LL(1).