Capitulo 2, ejercicio 6

Escribe una expresión regular para describir cada uno de los siguientes

construcciones del lenguaje de programación:

a. Cualquier secuencia de tabulaciones y espacios en blanco (a veces llamados espacios en blanco) segundo.

Respuesta: /\*t

b.Comentarios en el lenguaje de programación c

Respuesta: /\*/

c. Constantes de cadena (sin caracteres de escape)

Respuesta: [A-Z]\*[a-z]\*

d.Números de punto flotante

Respuesta: [-+]?[0-9]\*\.?[0-9]+

Capitulo 3, ejercicio 4

4) La siguiente gramática no es adecuada para un analizador predictivo descendente.

Identifique el problema y vuelva a escribir la gramática para corregir el problema.

Indique que la nueva gramática satisface la condición LL (1)

1 L → R a

2 | Q ba

3 R → a b a

4 | c a b a

5 | R b c

6 Q → segundo segundo c

7 | b c

responder

Primero, la recursividad izquierda R → R b c se convierte en recursión derecha. La sintaxis convertida es la siguiente:

1 L → R a

2 | Q ba

3 R → a b a R '

4 | c a b a R '

5 R '→ b c R'

6 | ε

7 Q → segundo segundo c

8 | b c

A continuación, extraemos factores de las producciones 7, 8 y la sintaxis convertida es la siguiente

1 L → R a

2 | Q ba

3 R → a b a R '

4 | c a b a R '

5 R '→ b c R'

6 | ε

7 Q → b Q '

8 Q '→ b c

9 | c

Siempre que se demuestre que la nueva gramática es sin retroceso, se puede demostrar que la gramática cumple con la condición LL (1).

Para cualquier símbolo no terminal A que coincida con múltiples producciones, A → β1 | β2 | ... βn

PRIMERO + (A → βi) ∩ PRIMERO + (A → βj) = ∅, ∀ 1 ≤ i, j ≤ n, i ≠ j

Cualquier gramática con esta propiedad es sin retroceso

El proceso de prueba es el siguiente

Aquí FIRST + (1) se usa para representar el FIRST + conjunto de producciones etiquetadas como 1.

PRIMERO + (9) = {c}

PRIMERO + (8) = {b}

PRIMERO + (8) ∩ PRIMERO + (9) = ∅

PRIMERO + (5) = {b}

PRIMERO + (6) = {ε, eof}

PRIMERO + (5) ∩ PRIMERO + (6) = ∅

PRIMERO + (4) = {c}

PRIMERO + (3) = {a}

PRIMERO + (3) ∩ PRIMERO + (4) = ∅

PRIMERO + (2) = {b}

PRIMERO + (1) = {a, c}

PRIMERO + (1) ∩ PRIMERO + (2) = ∅