Escribe una expresión regular para describir cada uno de las siguientes construcciones del lenguaje de programación:

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

/\*t

\t

b. Comentarios en el lenguaje de programación c

// Comentar una línea

/\* \*/ Comentar múltiples líneas

. String constants (without escape characters) RESPUESTA: [A-Z]\*[a-z]\*

d. Floating-point numbers RESPUESTA: [-+]?[0-9]\*\.?[0-9]+

La siguiente gramática no es adecuada para un analizador sintáctico de predictivo de arriba hacia abajo. Identifique el problema y corríjalo reescribiendo la gramática. Demuestre que su nueva gramática satisface el ll (1)condition.

L → R a |

Q ba

R → aba

| caba

| R bc

Q → bbc

| bc

Se puede demostrar que la gramática cumple con la condición LL (1). Cómo demostrar que la nueva gramática es sin retroceso puede basarse en la definición de P8 Para cualquier no terminal que coincida con múltiples producciones A, A → β1 | β2 | ... βn PRIMERO + (A → β i) ∩ PRIMERO + (A → β j) = ∅, ∀ 1 ≤ i, j ≤ n, i ≠ j Cualquier gramática con esta propiedad es sin retroceso El proceso de prueba es el siguiente Aquí PRIMERO + (1) se usa para denotar la producción PRIMERO + conjunto etiquetado 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) = ∅