

Prácticos

1) Un cierto lenguaje de programación utiliza una sentencia for especial, con la siguiente estructura:
for Variable **is range** (Expresión) [**step** Constante] **next** Variable

Este ciclo especial se repetirá desde Variable=0 hasta Variable=Expresión en pasos indicados en la constante que sigue a la palabra reservada **step**. Los corchetes indican que los pasos son opcionales. En caso que no se indique nada, por default asumirá pasos de 1 en 1.

Variable es un identificador que comienza con una letra y puede seguir con letras y dígitos.

Expresión es una expresión aritmética de la forma: Variable + Constante o Variable * Constante.

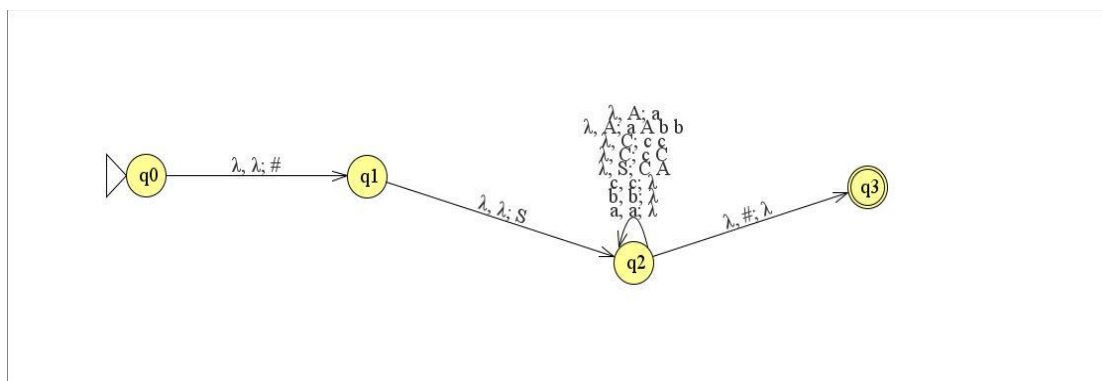
Constante es un número entero positivo.

Son terminales las palabras reservadas **for**, **is**, **range**, **step** y **next**. Son terminales los (), las letras, los dígitos y los símbolos +, -, * y /.

Ejemplo de cadenas: **for** a1 **is range** (c1*3) **step** 2 **next** a1
for i **is range** (n+1) **next** i
for cont **is range** (tope*2) **next** cont

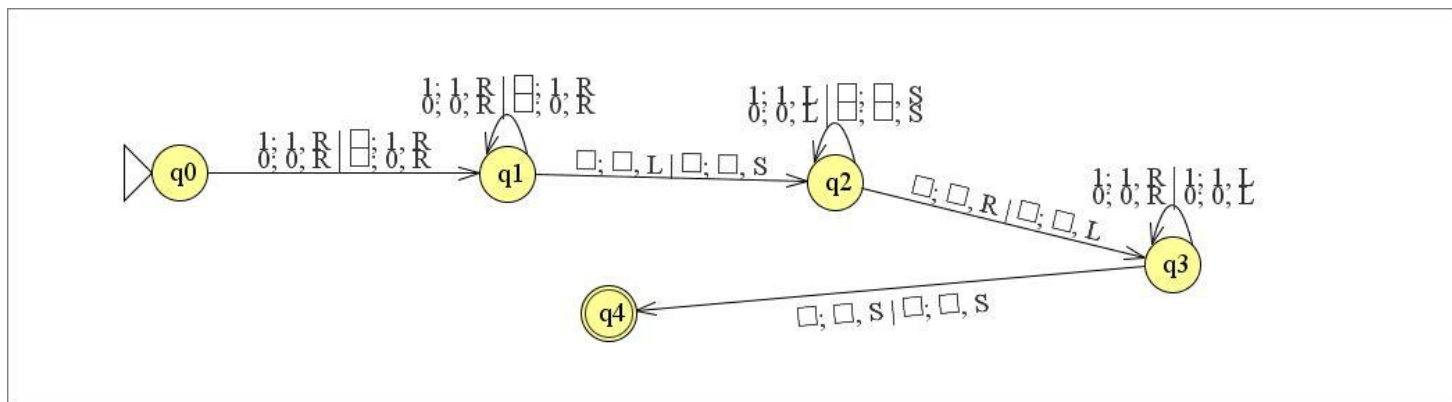
- Escribir las reglas de una gramática independiente de contexto en formato BNF para describir la sintaxis de esta sentencia for especial.
- Hacer el árbol de derivación de alguna de las cadenas ejemplo con la gramática que diseñó.

2) Dado el Parser LL correspondiente a la GIC: $G = \langle \{S, C, A\}, \{a, b, c\}, S, \{S \rightarrow CA; C \rightarrow cC \mid cc, A \rightarrow aAbb \mid a\} \rangle$



Hacer el análisis sintáctico de la cadena **ccccaabb**, mostrando en cada paso cómo queda la pila, y explicar cómo maneja la pila el Parser LL, al hacer el análisis sintáctico de una cadena.

3) Dada la MT= $\langle \{q_0, q_1, q_2, q_3, q_4\}, \{0,1\}, \{0,1, \square\}, \delta, q_0, \square, \{q_4\} \rangle$



Determine si las siguientes cadenas pertenecen o no al lenguaje aceptado por la MT:

- 10100101**
- 11001100**
- 1001001**
- 10110100**

Teóricos

1) Demostrar: El lenguaje: $L = \{ h^n g^j e^{2n} d^{3i} / i, j, n \geq 0 \}$, con alfabeto $\{e, d, g, h\}$ **es un LIC.**

2) Marcar si las siguientes afirmaciones son Verdaderas o Falsas:

- a. La gramática $G = \{ \{A\}, \{+, *, (,), vble\}, A, P \}$ donde $P: A \rightarrow A^*A, A \rightarrow A+A \mid (A) \mid vble$ es ambigua.
- b. ¿Se pueden generar las cadenas de un lenguaje representado por una expresión regular mediante una gramática independiente al contexto?
- c. No puedo diseñar un AP para el lenguaje: $L = \{ x / x = a^n b^m ; n, m \geq 1 ; m = 2 * n \}$
- d. La intersección de un lenguaje libre de contexto y un lenguaje regular da como resultado siempre un lenguaje regular.

3) Marcar si las siguientes afirmaciones son Verdaderas o Falsas:

- a. En un compilador, la tarea de reconocer los componentes del lenguaje de programación la hace el analizador sintáctico.
- b. Las reglas gramaticales de un lenguaje de programación son gramáticas tipo 3 de la Clasificación de Chomsky.
- c. Puedo definir una expresión regular para las constantes hexadecimales de un lenguaje de programación.
- d. En un parser LR, el árbol de parsing se arma desde el axioma hacia la cadena.

4) Marcar si las siguientes afirmaciones son Verdaderas o Falsas: Dado el lenguaje $L = \{ 0^{2p} 1^{2q} 0^p 1^q, p, q \geq 0 \}$ **con** $\Sigma = \{0, 1\}$

- a) El lenguaje puede ser reconocido por un Autómata con Pila.
- b) El lenguaje puede ser reconocido por un Autómata con Pila y por una Máquina de Turing.
- c) El lenguaje puede ser reconocido sólo por una Máquina de Turing.
- d) Si $p \geq 1$ y $q = 3$, el lenguaje puede ser reconocido por un Autómata con Pila.