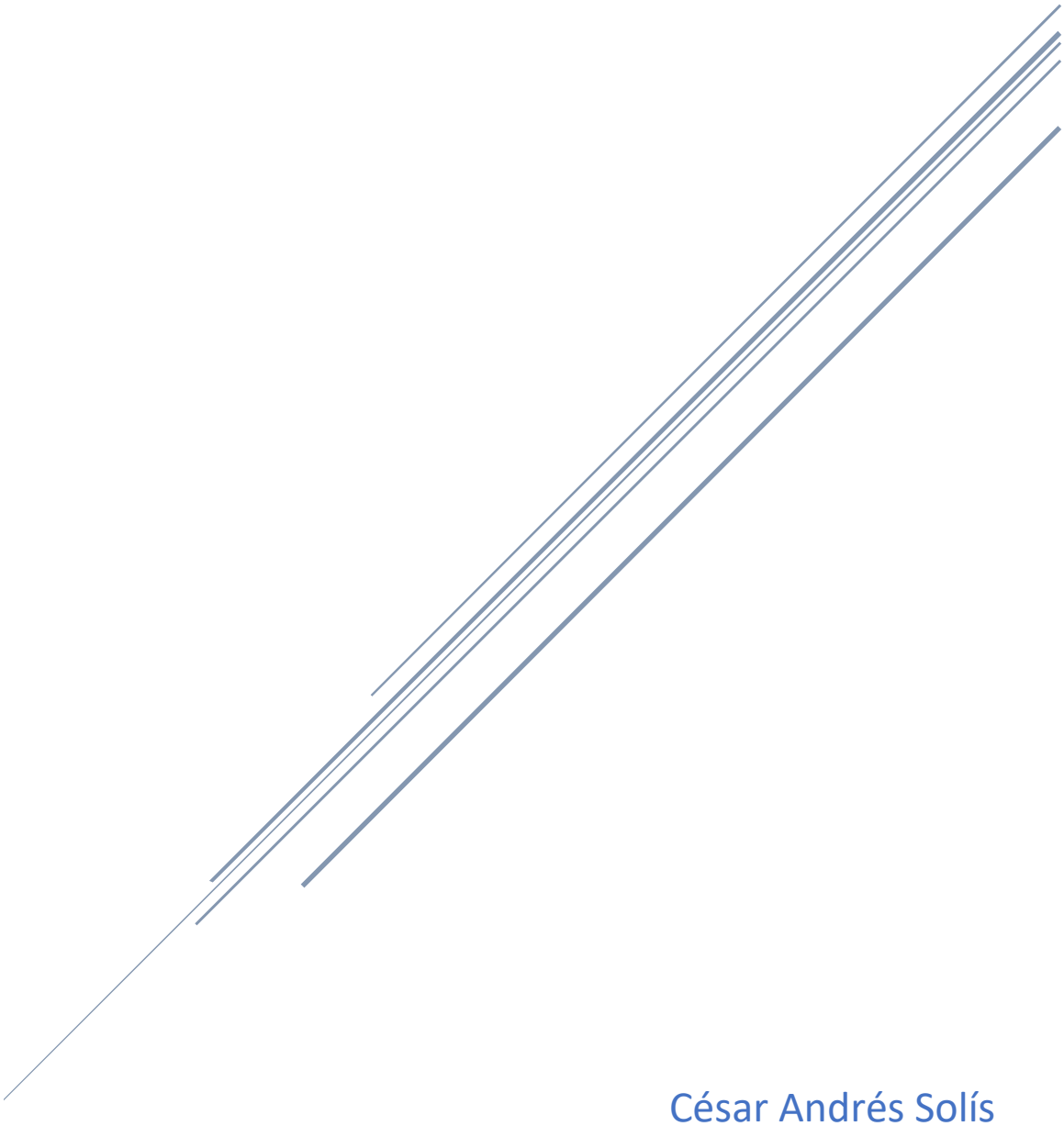
EXAMEN SEMESTRAL

Lenguajes formales, autómatas y compiladores



César Andrés Solís
9-759-1658

Contenido

I. Presentación

Se presenta el diseño de un autómata finito no determinista (AFND), un autómata finito determinista (AFD) y una expresión regular para el lenguaje de cadenas que representan las monedas insertadas en una máquina expendedora de alimentos. El lenguaje aceptado por este autómata es el siguiente:

L = {m1m2m3... | m1, m2, ... son monedas de 0.25, 0.50 o 1.00, y la suma de las monedas es mayor o igual al precio del producto que se desea comprar}

También se presenta el diseño y la implementación de un programa en JavaScript que simula una máquina expendedora de alimentos. El programa se basa en la teoría de lenguajes formales, autómatas y compiladores.

II. Descripción del proyecto

La máquina expendedora acepta monedas de B/.0.25, B/.0.50 y B/.1.00. Los productos que dispensa son los siguientes:

- Soda B/.1.00
- Emparedados B/.1.50
- Jugos B/.1.25

El programa permite que el usuario ingrese una cantidad N de monedas. El analizador léxico del programa debe reconocer las monedas válidas y rechazar las monedas inválidas. El analizador sintáctico del programa debe verificar que la cadena de monedas sea válida y que el monto acumulado sea suficiente para adquirir un producto.

III. Autómata finito no determinista (AFND)

Estados

- (Q): $\{q0, q1, q2, q3\}$
- $q0$: $Estado\ inicial$
- $q1, q2, q3$: $estados\ de\ transición$
- Estados finales o de aceptación: No hay estados finales explícitos. La aceptación se basa en llegar a un estado válido después de procesar todas las monedas.

Alfabeto de entrada

- (Σ): $\{0.25, 0.50, 1.00\}$

Función de Transición (δ):

$q0 \xrightarrow{0.25} q1$

$q0 \xrightarrow{0.50} q2$

$q0 \xrightarrow{1.00} q3$

$q1 \xrightarrow{0.25} q1$

$q1 \xrightarrow{0.50} q2$

$q1 \xrightarrow{1.00} q3$

$q2 \xrightarrow{0.25} q1$

$q2 \xrightarrow{0.50} q2$

$q2 \xrightarrow{1.00} q3$

$q3 \xrightarrow{0.25} q1$

$q3 \xrightarrow{0.50} q2$

$q3 \xrightarrow{1.00} q3$

La cadena es aceptada si, después de procesar todas las monedas, el autómata se encuentra en un estado válido. En este caso, cualquier estado $q1, q2, q3$ indica una cadena aceptada, ya que la compra se puede realizar independientemente del estado final específico.

Diagrama de Transiciones

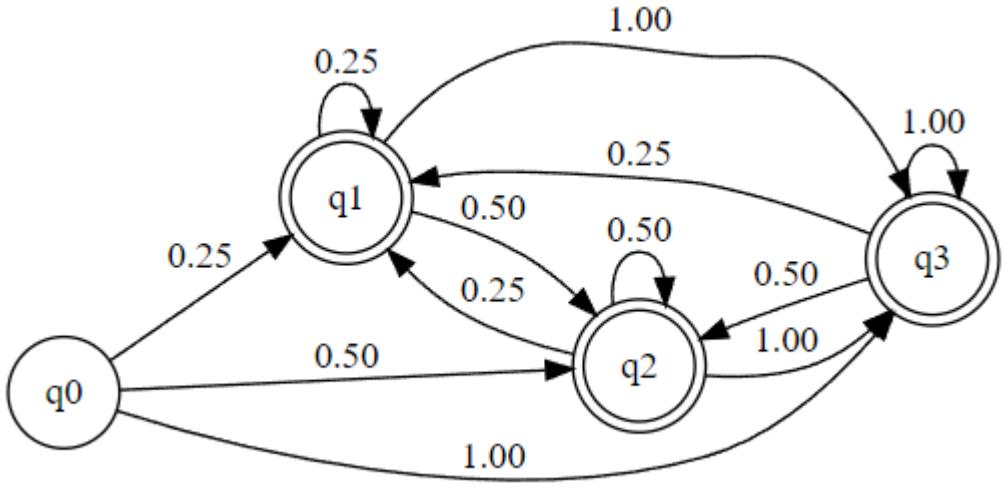


Ilustración 1: AFND

Tabla de transiciones

Estado actual	Moneda	Nuevo estado
q0	0.25	q1
q0	0.50	q2
q0	1.00	q3
q1	0.25	q1
q1	0.50	q2
q1	1.00	q3
q2	0.25	q1
q2	0.50	q2
q2	1.00	q3
q3	0.25	q1
q3	0.50	q2
q3	1.00	q3

Tabla 1: Tabla de transiciones

IV. Autómata finito determinista

A partir del Autómata Finito No Determinista (AFND) anterior, es posible construir un Autómata Finito Determinista (AFD) equivalente. El AFD tendrá un conjunto de estados y transiciones deterministas.

Estados

- (Q): $\{q0, q1, q2, q3\}$
- $q0$: $Estado\ inicial.$
- $q1, q2, q3$: $Estados\ de\ transición.$

Alfabeto de entrada

- (Σ): {0.25, 0.50, 1.00}

Función de Transición (δ)

$q0 \xrightarrow{0.25} q1$

$q0 \xrightarrow{0.50} q2$

$q0 \xrightarrow{1.00} q3$

$q1 \xrightarrow{0.25} q1$

$q1 \xrightarrow{0.50} q2$

$q1 \xrightarrow{1.00} q3$

$q2 \xrightarrow{0.25} q1$

$q2 \xrightarrow{0.50} q2$

$q2 \overset{1.00}{\rightarrow} q3$

$q3 \overset{0.25}{\rightarrow} q1$

$q3 \overset{0.50}{\rightarrow} q2$

$q3 \overset{1.00}{\rightarrow} q3$

Diagrama de transiciones

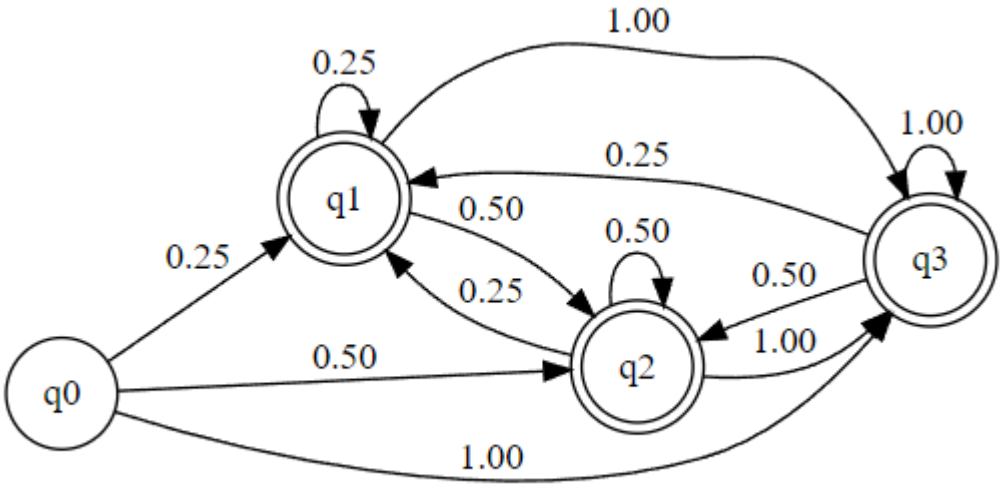


Ilustración 2: AFD

V. Expresión regular

En términos simples, la expresión regular sería una combinación de las monedas permitidas que resultan en una cadena válida.

Expresión regular	JavaScript
$(\mathbf{0.25}\|\mathbf{0.50}\|\mathbf{1.00})*$	$const\ regex\ = /\^(0\.25\|0\.50\|1\.00)*\$/;$

Esta expresión regular significa que cualquier combinación (cero o más veces) de las monedas permitidas $(0.25, 0.50, 1.00)$ es aceptada. Cada moneda puede aparecer en cualquier orden y repetirse varias veces.

VI. Gramática correspondiente

$S -> Compras$

$Compras -> \varepsilon \mid Compra\ Compras$

$Compra -> Monedas\ Producto$

$Monedas -> \varepsilon \mid Moneda\ Monedas$

$Moneda -> 0.25 \mid 0.50 \mid 1.00$

$Producto -> Soda \mid Emparedado \mid Jugo$

$Soda -> 1.00$

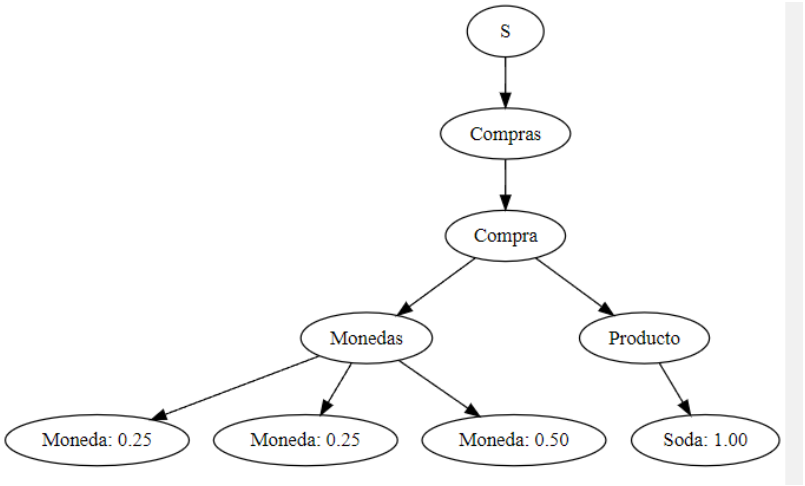
$Emparedado -> 1.50 \mid 0.50\ 1.00$

$Jugo -> 1.25 \mid 0.25\ 0.50\ 0.50$

Esta gramática permite representar la compra de productos en una máquina expendedora, donde el usuario puede tener cualquier combinación y orden de monedas antes de cada compra, siempre que el saldo sea suficiente para cubrir el costo del producto seleccionado.

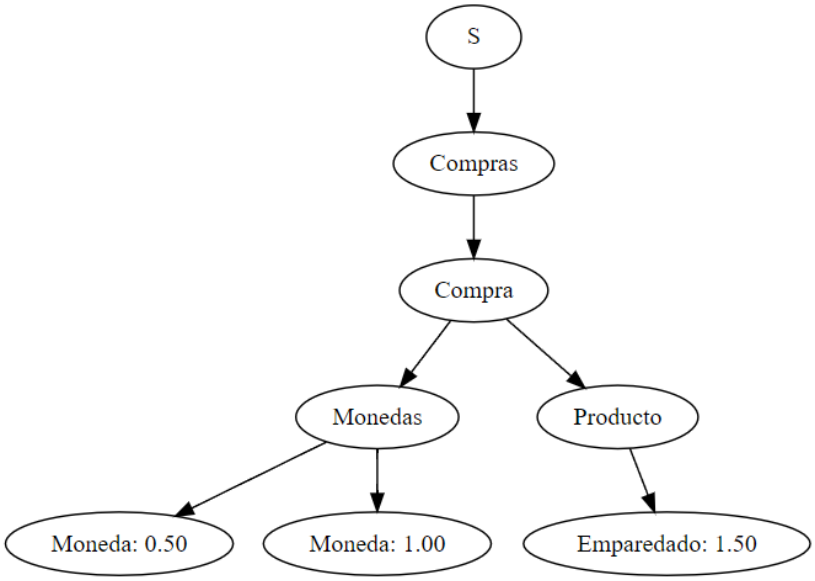
a. Árbol 1 (Soda)

- **Cadena:** 0.25 0.25 0.50
- **Árbol sintáctico**



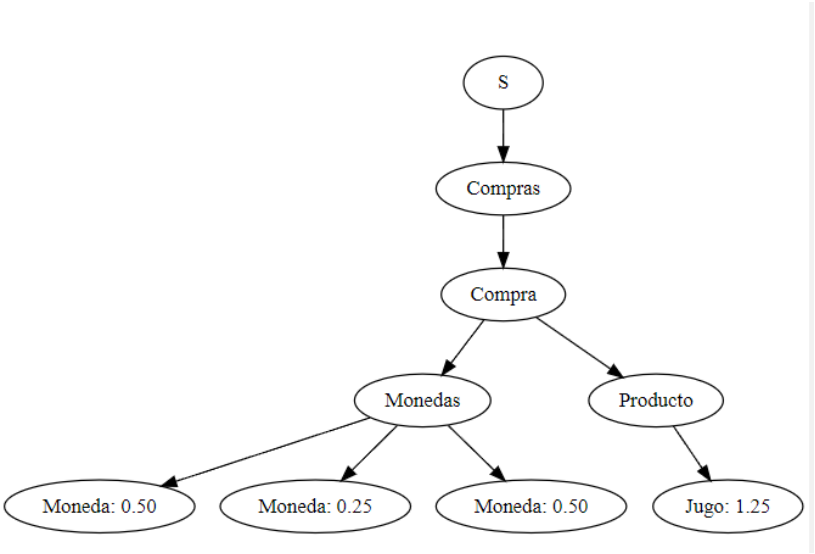
b. Árbol 2 (Emparedado)

- **Cadena:** 0.50 1.00
- **Árbol sintáctico**



c. Árbol 3 (Jugo)

- **Cadena:** 0.50 0.25 0.50
- **Árbol sintáctico**



VII. Conclusión