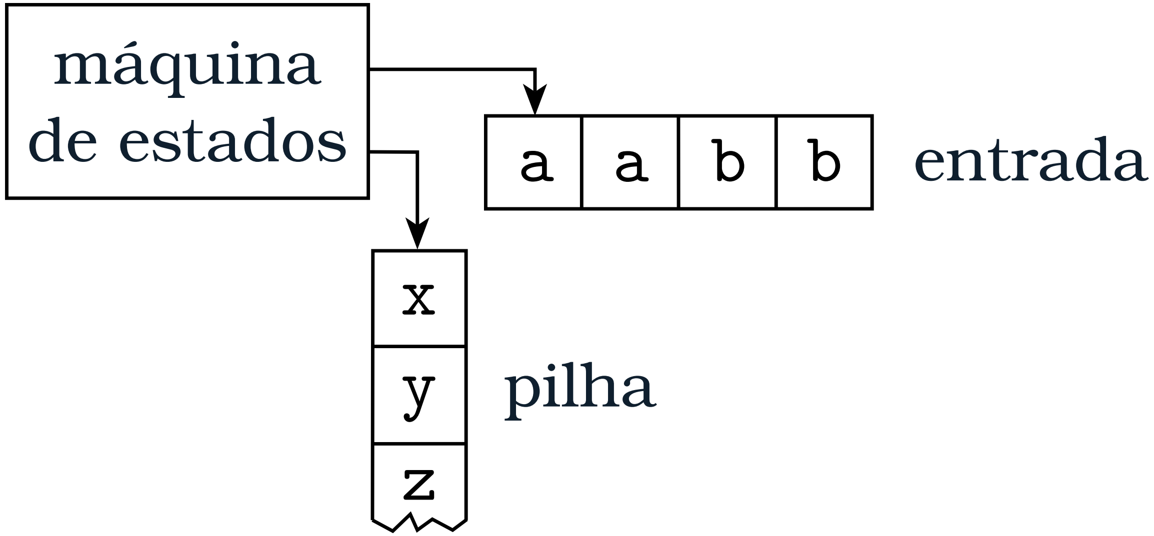
Construa um autômato que reconheça a linguagem $L_1 = \{0^n 1^n \mid n \geq 0\}$

- L_1 não é uma linguagem regular
- Não é possível construir um autômato finito para L_1
- Precisamos de um autômato de pilha

Autômato de Pilha

- Autômato finito **+** Memória LIFO
- Reconhece **linguagens livre de contexto**

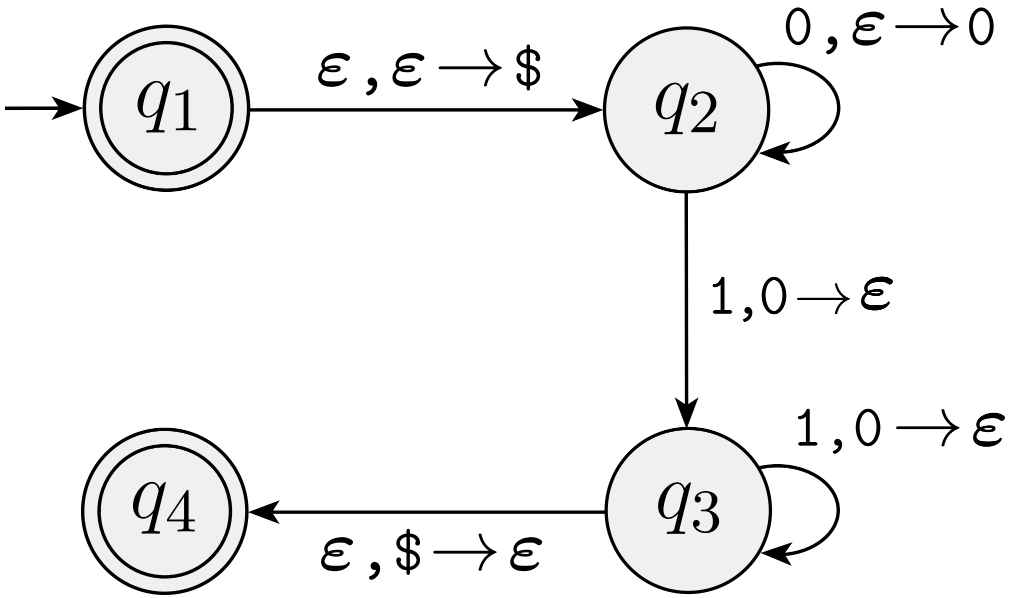


Linguagens Livre de Contexto

- $L_1 = \{0^n 1^n | n \geq 0\}$.
- Verificar parêntese
- Análise sintática de linguagens de programação

Autômato de Pilha de Reconhece $L_1 = \{0^n 1^n \mid n \geq 0\}$

- $M_1 =$



Definição Formal Autômato de Pilha

- 6-tupla: $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$
  - Q — Conjunto finito de estados
  - Σ — Alfabeto de entrada
  - Γ — Alfabeto de pilha
  - δ: $Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to 2^{Q \times \Gamma_\varepsilon}$ — Função de transição
  - $q_0 \in Q$ — Estado inicial
  - $F \subseteq Q$ — Estados finais

Computação com Autômato de Pilha

Configuração do autômato

$$[q, abc, xyz]$$

- $q \in Q$ — Estado atual da máquina
- abc — Entrada não processada
- xyz — Conteúdo da pilha

Sequência de configurações

$$[p, ab, xy] \vdash [q, b, zy]$$

- $((p, a, x) \to (q, z)) \in \delta$
- $b \in \Sigma^*$ e $y \in \Gamma^*$

M aceita w se

$$[q_0, w, \varepsilon] \vdash_M^* [q_f, \varepsilon, \varepsilon]$$

- $q_f \in F$

$M_1 = (Q, \Sigma, \Gamma, \delta, q_1, F)$

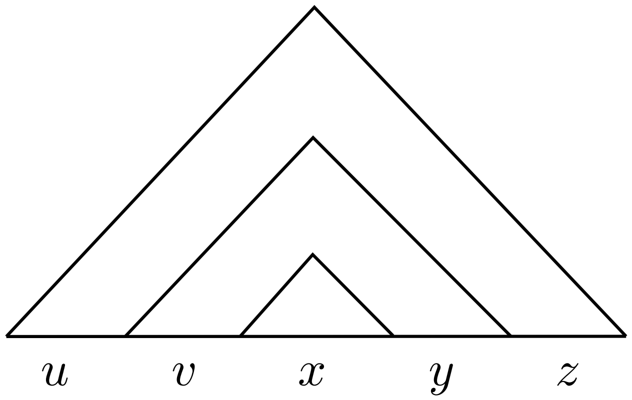
- $Q = \{q_1, q_2, q_3, q_4\}$ — Conjunto finito de estados
- $\Sigma = \{0, 1\}$ — Alfabeto de entrada
- $\Gamma = \{\$, 0\}$ — Alfabeto de pilha
- $F = \{q_1, q_4\}$ — Estados finais
- δ — Função de transição

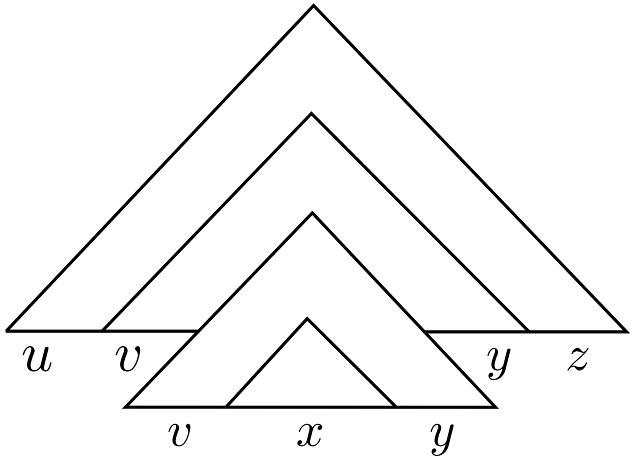
Input:	0			1			ε		
Pilha:	0	\$	ε	0	\$	ε	0	\$	ε
q_1									$\{(q_2, \$)\}$
q_2			$\{(q_2, 0)\}$	$\{(q_3, \varepsilon)\}$					
q_3				$\{(q_3, \varepsilon)\}$				$\{(q_4, \varepsilon)\}$	
q_4									

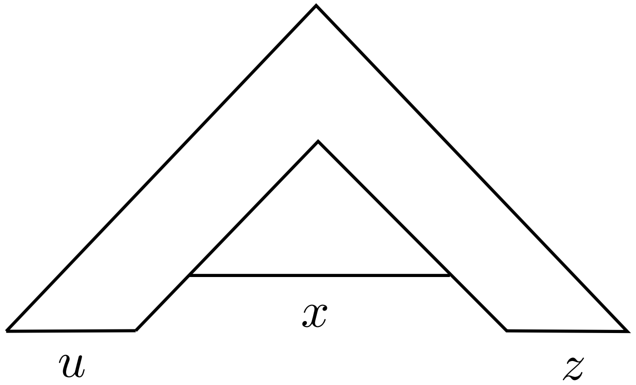
Exemplo de Linguagens Livre de Contexto (LLC)

- $L_2 = \{w\#w^{\mathcal{R}} \mid w \in \{0,1\}^*\}$
- $L_3 = \{ww^{\mathcal{R}} \mid w \in \{0,1\}^*\}$
- $L_4 = \{a^i b^j c^k \mid i,j,k \geq 0 \;\wedge\; (i = j \;\vee\; j = k)\}$

A Forma de uma LLC







Equivalência Entre Autômato de Pilha e Autômato de Pilha Determinístico

Autômatos de pilha determinísticos reconhecem um subconjunto das linguagens reconhecidas pelos autômatos de pilha.

LLC Inerentemente Ambígua

- Exemplo: $L_4 = \{a^i b^j c^k \mid i, j, k \geq 0 \ \wedge \ (i = j \ \vee \ j = k)\}$
- LLC $\supset$ LLC não ambígua $\supset$ LLC Determinística
- Autômato de Pilha $\supset$ Autômato de Pilha Determinístico

LLC: Fechamento Sob as Operações

- ✓ União — $L_a \cup L_b = \{w \mid w \in L_a \ \lor \ w \in L_b\}$
- ✓ Concatenação — $L_a \cdot L_b = \{wz \mid w \in L_a \ \land \ z \in L_b\}$
- ✓ Estrela de Kleene — L^*
- ✓ Intersecção com linguagem regular — $L \cap R = \{w \mid w \in L \ \land \ w \in R\}$
- ❌ Complemento — $\neg L = \{w \mid w \notin L\}$
- ❌ Intersecção — $L_a \cap L_b = \{w \mid w \in L_a \ \land \ w \in L_b\}$
  - $L_5 = \{a^n b^n c^k \mid n, k \geq 0\} \in$ LLC
  - $L_6 = \{a^k b^n c^n \mid n, k \geq 0\} \in$ LLC
  - $L_7 = L_5 \cap L_6 = \{a^n b^n c^n \mid n \geq 0\} \notin$ LLC