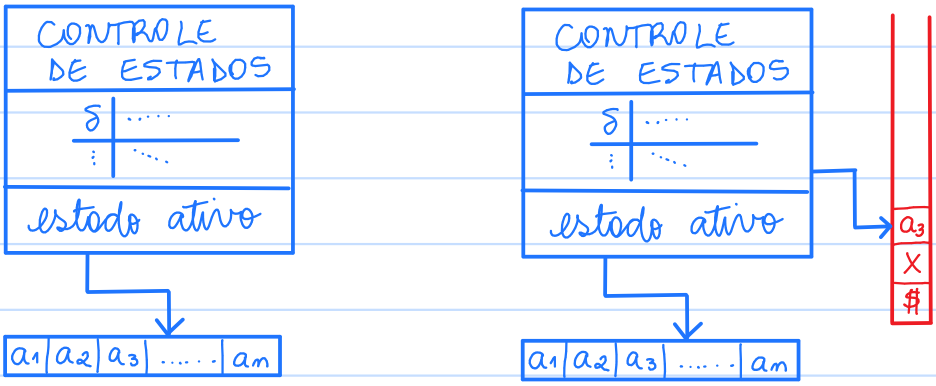
Autômatos com Pilha

→ São basicamente um AFN com uma pilha (em cada cópia).

→ A pilha é uma memória extra ilimitada que funciona no esquema "último a entrar, primeiro a sair".



→ Pilhas nos dão poder de recursão.

Transições em autômatos com pilha

$$p \xrightarrow{a,b \to c} q$$

Cada um dentre a, b, c pode ser ε

"Estando com p ativo, leia a da cadeia de entrada, desempilhe b do topo da pilha e empilhe c"

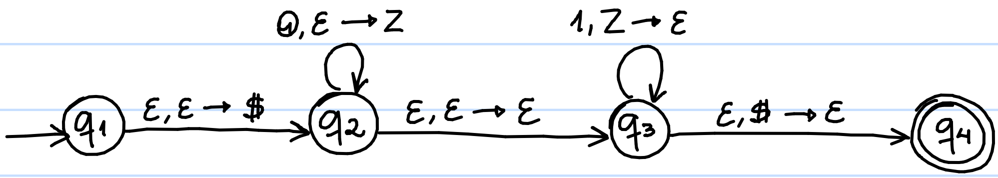
Intuição

→ Como reconhecer $\{0^n 1^n : n \geq 0\}$?

- Use a pilha para lembrar do nº de 0s lidos
- Se ler 0, empilhe-o (ou algum outro símbolo).
  Se ler 1, desempilhe.
- Se a entrada terminar e a pilha estiver vazia, aceite.

→ Questão: como verificar se a pilha está vazia?

Exemplo



Autômatos com pilha

DEFINIÇÃO: Um autômato com pilha (AP) é uma 6-upla $(Q, \Sigma, \Gamma, \delta, q_0, F)$ em que

- Q é um conj. finito de estados
- Σ é o alfabeto de entrada
- Γ é o alfabeto da pilha
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\}) \longrightarrow \mathcal{P}(Q \times (\Gamma \cup \{\varepsilon\}))$ é a função de transição
- $q_0 \in Q$ é o estado inicial
- $F \subseteq Q$ é o conj. de estados finais

Descrição instantânea

DEFINIÇÃO: A descrição instantânea de um AP $P = (Q, \Sigma, \Gamma, \delta, q_0, F)$ é dada pela tripla (q, w, γ), em que

- q é um estado
- $w \in \Sigma^*$ é a entrada restante
- $\gamma \in \Gamma^*$ é o conteúdo da pilha (com o topo à direita)

Sequências de descrições instantâneas

DEFINIÇÃO: Seja um AP $P=(Q,\Sigma,\Gamma,\delta,q_0,F)$ em que $(p,Y)\in\delta(q,x,X)$. Então para qualquer $w\in\Sigma^*$ e $\gamma\in\Gamma^*$,

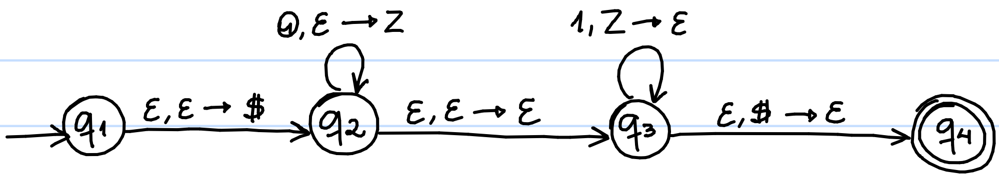
$$(q,xw,\gamma X)\vdash(p,w,\gamma Y)$$

é um *movimento* de P.

$\vdash$: turnstile ; catraca
"acarreta em"

Usamos $\vdash^*$ para representar uma sequência de zero ou mais movimentos em P.

Exemplo



$(q_1, 0001, \varepsilon)\vdash(q_2, 0001, \$)\vdash(q_2, 001, \$Z)\vdash(q_2, 01, \$ZZ)$

$\quad (q_2, 0001, \$)\vdash(q_3, 0001, \$)\vdash(q_4, 0001, \varepsilon)$

$\quad (q_2, 001, \$Z)\vdash(q_3, 001, \$Z)$

$(q_1, 0001, \varepsilon)\vdash^*(q_2, 1, \$ZZZ)\vdash(q_3, 1, \$ZZZ)\vdash(q_3, \varepsilon, \$ZZ)$

Aceite e reconhecimento em AP

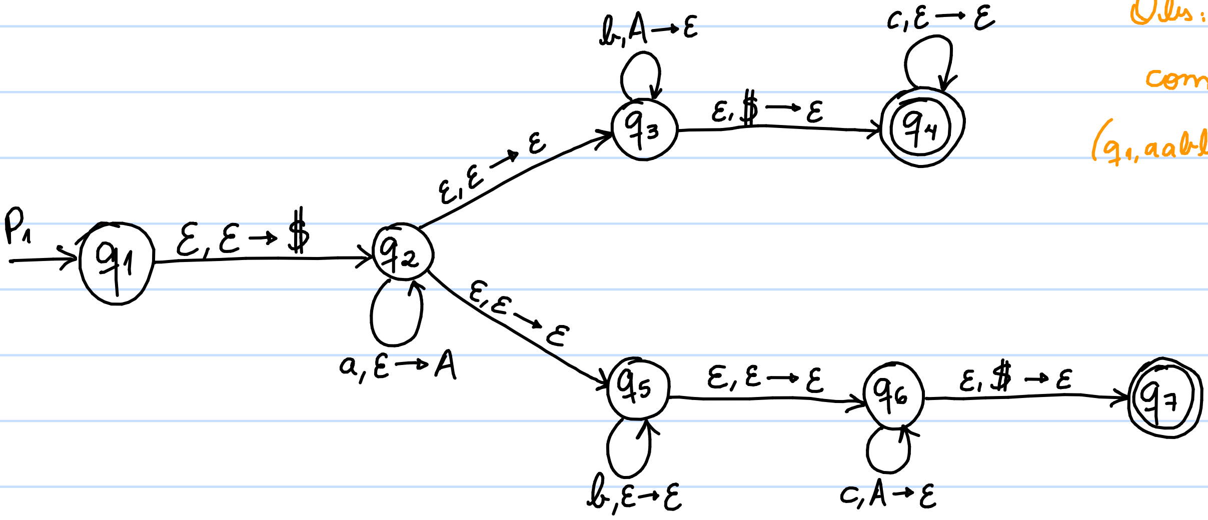
DEFINIÇÃO: Dado um AP $P=(Q,\Sigma,\Gamma,\delta,q_0,F)$ e uma cadeia $w\in\Sigma^*$, dizemos que *P aceita w* se

$$(q_0,w,\varepsilon)\vdash^*(q,\varepsilon,\gamma)$$

sendo $q\in F$ e $\gamma\in\Gamma^*$.

A linguagem $X=\{w\in\Sigma^*: P \text{ aceita } w\}$ é a *linguagem reconhecida por P* ou a *linguagem de P*, e escrevemos $L(P)=X$.

Mais exemplos de APs



Obs: chega no q_4 com aabbb

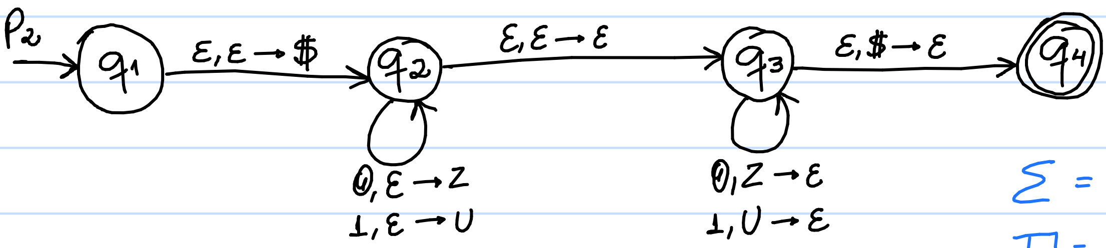
$(q_1, aabbb, \varepsilon) \vdash^* (q_4, b, \varepsilon)$

{ aabcccc chega no q_7

aaccbb tbm

$$L(P_1) = \{ a^i b^j c^k : i, j, k \geq 0 \text{ e } i = j \text{ ou } i = k \}$$

Mais exemplos de APs



$\Sigma = \{0, 1\}$

$\Gamma = \{\$, Z, U\}$

$$L(P_2) = \{ ww^R : w \in \{0,1\}^* \}$$

APs e linguagens livres de contexto

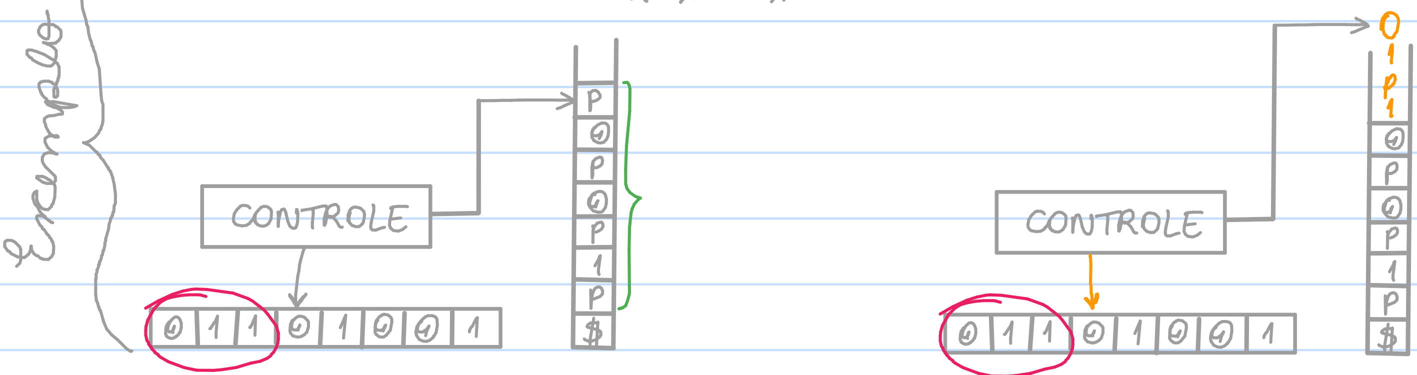
TEOREMA: Uma linguagem é livre de contexto se e somente se algum autômato com pilha a reconhece.

Ideia da prova: Para provar a ida, isto é, que "Se uma linguagem L é livre de contexto, então algum autômato com pilha a reconhece": como L é livre de contexto, tome uma GLC $G = (V, \Sigma, R, S)$ que a gera e construa um AP que simula as derivações à esquerda de G, isto é, derivações em que sempre escolhe-se a variável mais à esquerda para aplicar uma regra.

As cadeias nessas derivações são sempre da forma $\alpha X \lambda$, sendo $\alpha \in \Sigma^*$, $X \in V$ e $\lambda \in (\Sigma \cup V)^*$

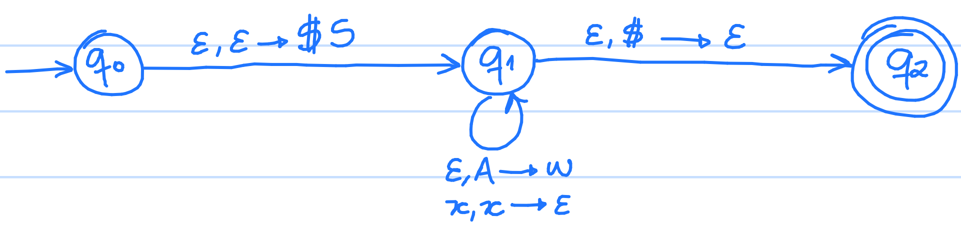
Faremos com que $X\lambda$ fique na pilha, com X no topo e α seja a parte já lida da entrada ($w = \alpha\beta$).

$P \Rightarrow 0P1P \Rightarrow 01P0P1P \Rightarrow \underbrace{011}_{\alpha}\underbrace{P}_{X}\underbrace{0P0P1P}_{\lambda}$ $\qquad P \to 0P1P \mid 1P0P \mid \varepsilon$



Ideia de funcionamento de P:

- Empilha \$ e S (variável inicial)
- Repete:
  - Se o topo da pilha é uma variável A, "escolha" uma das regras $A \to \alpha$, desempilhe e empilhe α.
  - Se o topo da pilha é um terminal x, tente casá-lo com o símbolo da entrada.
  - Se o topo da pilha é \$, vá para o estado final.



Para provar a volta, isto é, que "Se algum AP reconhece uma linguagem L, então ela é livre de contexto": tomamos tal AP $P = (Q, \Sigma, \Gamma, \delta, q_0, F)$ e escrevemos uma gramática que gere as mesmas cadeias aceitos por ele.

Para isso, começamos modificando P para ter um único estado final q_F, para esvaziar a pilha antes de aceitar e para que cada transição não empilhe e desempilhe ao mesmo tempo.

$$q \xrightarrow{x,\, X \to Y} p \quad \equiv \quad q \xrightarrow{x,\, X \to \varepsilon} q_p \xrightarrow{\varepsilon,\, \varepsilon \to Y} p$$

$$q \xrightarrow{x,\, \varepsilon \to \varepsilon} p \quad \equiv \quad q \xrightarrow{x,\, \varepsilon \to N} q_p \xrightarrow{\varepsilon,\, N \to \varepsilon} p$$

Nossa gramática terá uma variável A_{pq} para cada par de estados p e q de P. Ela vai gerar todas as cadeias que levam P de p para q mantendo a pilha vazia.** $(p, \alpha\beta, \varepsilon) \vdash^* (q, \beta, \varepsilon) \Rightarrow A_{pq}$ gera α

Seja α uma cadeia dessas.

Certamente o primeiro movimento de P em α é de empilhar e o último movimento é de desempilhar.

Temos duas possibilidades para o símbolo X empilhado no início:

- ele foi desempilhado no fim $((r, X) \in \delta(p, a, \varepsilon)$ e $(q, \varepsilon) \in \delta(s, b, X))$:

  adicione $A_{pq} \to a A_{rs} b$ à gramática

- ele foi desempilhado antes do fim, momento em que a pilha fica vazia:

  adicione $A_{pq} \to A_{pr} A_{rq}$ à gramática $\forall\, r \in Q$

Adicione ainda $A_{pp} \to \varepsilon$ para cada $p \in Q$.

Faça $A_{q_0 q_F}$ a variável inicial.

"CQD"