ACH2043
INTRODUÇÃO À TEORIA DA COMPUTAÇÃO

Aula 13

Cap. 2.2 – Autômato com Pilha (parte 3)

Profa. Ariane Machado Lima
ariane.machado@usp.br

Aula passada...

Definição formal
Autômato com Pilha (AP) não determinístico

DEFINIÇÃO 2.13

Um ***autômato com pilha*** é uma 6-upla $(Q, \Sigma, \Gamma, \delta, q_0, F)$, onde Q, Σ, Γ e F são todos conjuntos finitos, e

1. Q é o conjunto de estados,
2. Σ é o alfabeto de entrada,
3. Γ é o alfabeto de pilha,
4. $\delta\colon Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \longrightarrow \mathcal{P}(Q \times \Gamma_\varepsilon)$ é a função de transição,
5. $q_0 \in Q$ é o estado inicial, e
6. $F \subseteq Q$ é o conjunto de estados de aceitação.

Computação com um AP

Um autômato com pilha $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ computa da seguinte maneira. Ele aceita a entrada w se w puder ser escrita como $w = w_1 w_2 \cdots w_m$, onde cada $w_i \in \Sigma_\varepsilon$, e existem uma seqüência de estados $r_0, r_1, \ldots, r_m \in Q$ e cadeias $s_0, s_1, \ldots, s_m \in \Gamma^*$ que satisfazem as três condições a seguir. As cadeias s_i representam a seqüência de conteúdo da pilha que M tem no ramo de aceitação da computação.

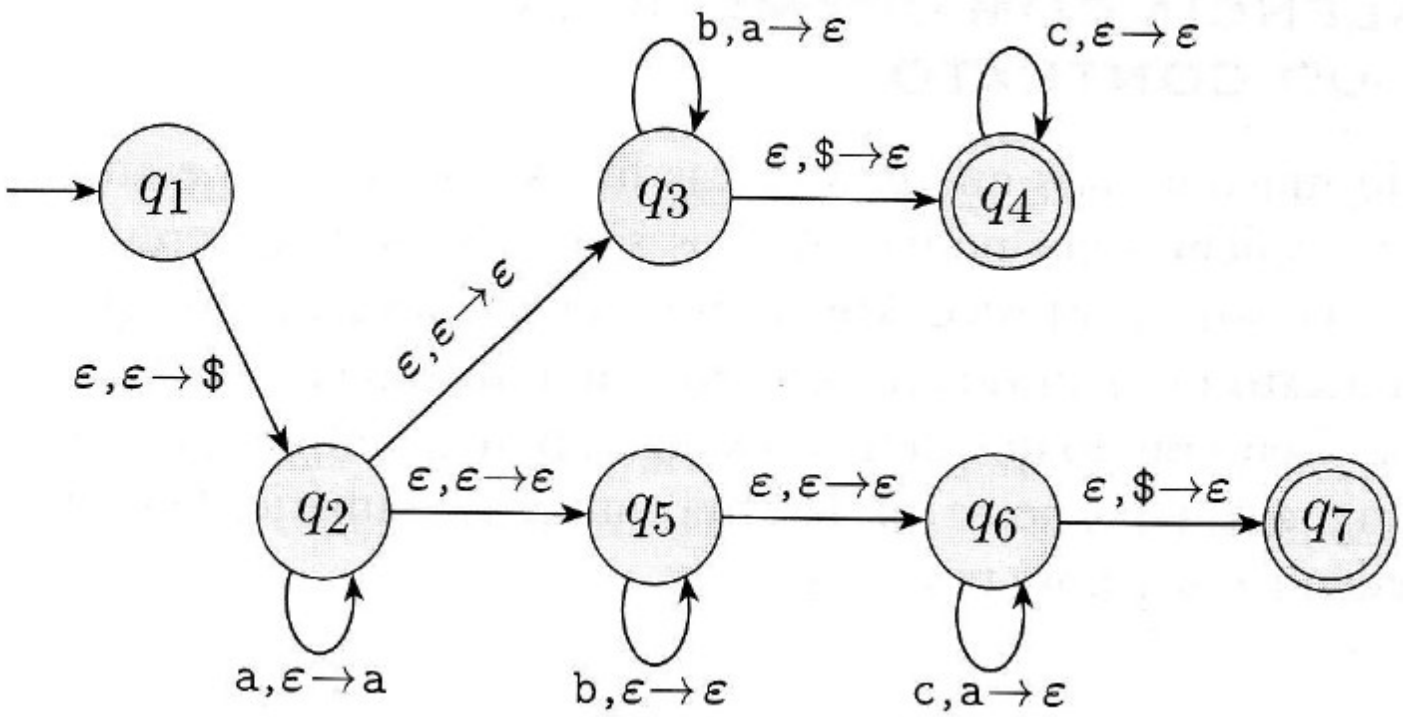
1. $r_0 = q_0$ e $s_0 = \varepsilon$. Essa condição significa que M inicia apropriadamente, no estado inicial e com uma pilha vazia.
2. Para $i = 0, \ldots, m-1$, temos $(r_{i+1}, b) \in \delta(r_i, w_{i+1}, a)$, onde $s_i = at$ e $s_{i+1} = bt$ para algum $a, b \in \Gamma_\varepsilon$ e $t \in \Gamma^*$. Essa condição afirma que M se move apropriadamente, conforme o estado, a pilha e o próximo símbolo de entrada.
3. $r_m \in F$. Essa condição afirma que um estado de aceitação ocorre ao final da entrada.

EXEMPLO 2.16

$$\{a^i b^j c^k \mid i, j, k \geq 0 \text{ e } i = j \text{ ou } i = k\}$$

Empilho quando leio a's, e desempilho quando leio b's ou c's?

Aqui não-determinismo também é essencial!



Equivalência entre APN e GLC

TEOREMA 2.20

Uma linguagem é livre-do-contexto se e somente se algum autômato com pilha a reconhece.

<=>

NÃO DETERMINÍSTICO!!!

Equivalência entre APN e GLC

LEMA 2.21

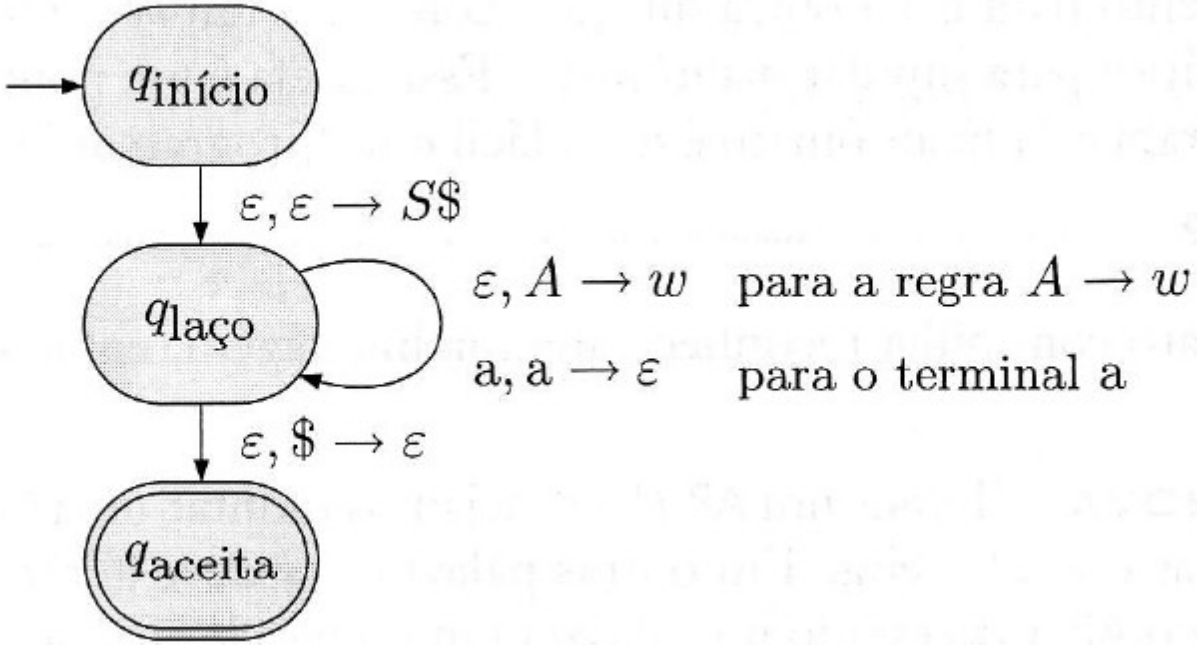
Se uma linguagem é livre-do-contexto, então algum autômato com pilha a reconhece. (=>)

Ideia da prova:

- Uma LLC é gerada por uma GLC
- Mostrar como converter uma GLC em um APN equivalente

Conversão GLC em APN (ideia)

- Caso Geral: (notação simplificada, representando o empilhamento de uma sequência de símbolos todos de uma vez)



Equivalência entre APN e GLC

TEOREMA 2.20

Uma linguagem é livre-do-contexto se e somente se algum autômato com pilha a reconhece.

PROVAMOS ESTE! (=>)

LEMA 2.21

Se uma linguagem é livre-do-contexto, então algum autômato com pilha a reconhece.

LEMA 2.27

Se um autômato com pilha reconhece alguma linguagem, então ela é livre-do-contexto.

Equivalência entre APN e GLC

TEOREMA 2.20

Uma linguagem é livre-do-contexto se e somente se algum autômato com pilha a reconhece.

LEMA 2.21

Se uma linguagem é livre-do-contexto, então algum autômato com pilha a reconhece.

LEMA 2.27

Se um autômato com pilha reconhece alguma linguagem, então ela é livre-do-contexto.

FALTA PROVAR ESTE... (<=)

Aula de hoje...

Conversão APN em GLC (ideia)

- Para facilitar, vamos considerar que o APN possui as seguintes características:

1. Ele tem um único estado de aceitação, q_{aceita}.
2. Ele esvazia sua pilha antes de aceitar.
3. Cada transição ou empilha um símbolo (um movimento de *empilha*) ou desempilha um símbolo (um movimento de *desempilha*), mas não faz ambas as coisas ao mesmo tempo.

Conversão APN em GLC (ideia)

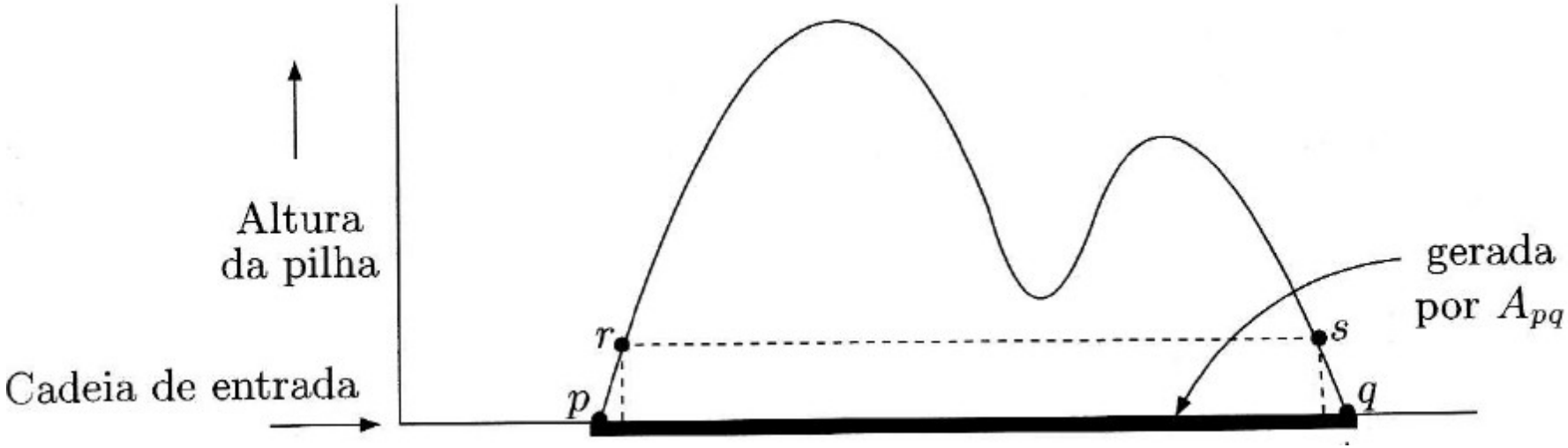
- G deve gerar uma cadeia w se w faz o APN ir do estado inicial ao estado de aceitação.
- Para cada par de estados (p, q), criamos uma variável A_{pq} que gere todas as cadeias x que levam o APN do estado p (com uma pilha vazia) ao estado q (com uma pilha vazia).
- Neste APN:
  - no estado p (com pilha vazia), o primeiro movimento é de EMPILHA.
  - O último movimento é de DESEMPILHA (chegando no estado q, com pilha vazia)

Conversão APN em GLC (ideia)

- No caminho de p a q (reconhecendo x), 2 situações:
  - A pilha só se torna vazia novamente quando chega em q
  - A pilha se torna vazia em algum ponto do caminho, antes de chegar em q

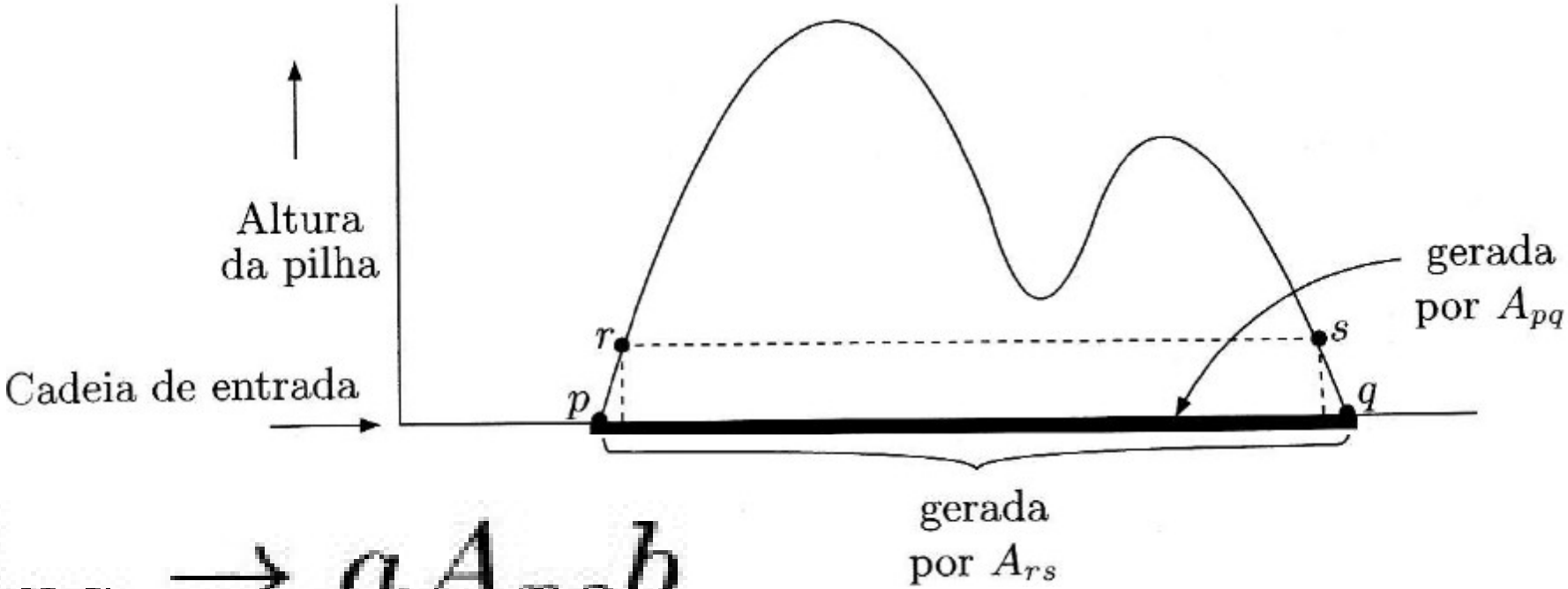
Conversão APN em GLC (ideia)

- No caminho de p a q (reconhecendo x), 2 situações:
  - A pilha só se torna vazia novamente quando chega em q
  - A pilha se torna vazia em algum ponto do caminho, antes de chegar em q



Conversão APN em GLC (ideia)

- No caminho de p a q (reconhecendo x), 2 situações:
  - A pilha só se torna vazia novamente quando chega em q
  - A pilha se torna vazia em algum ponto do caminho, antes de chegar em q



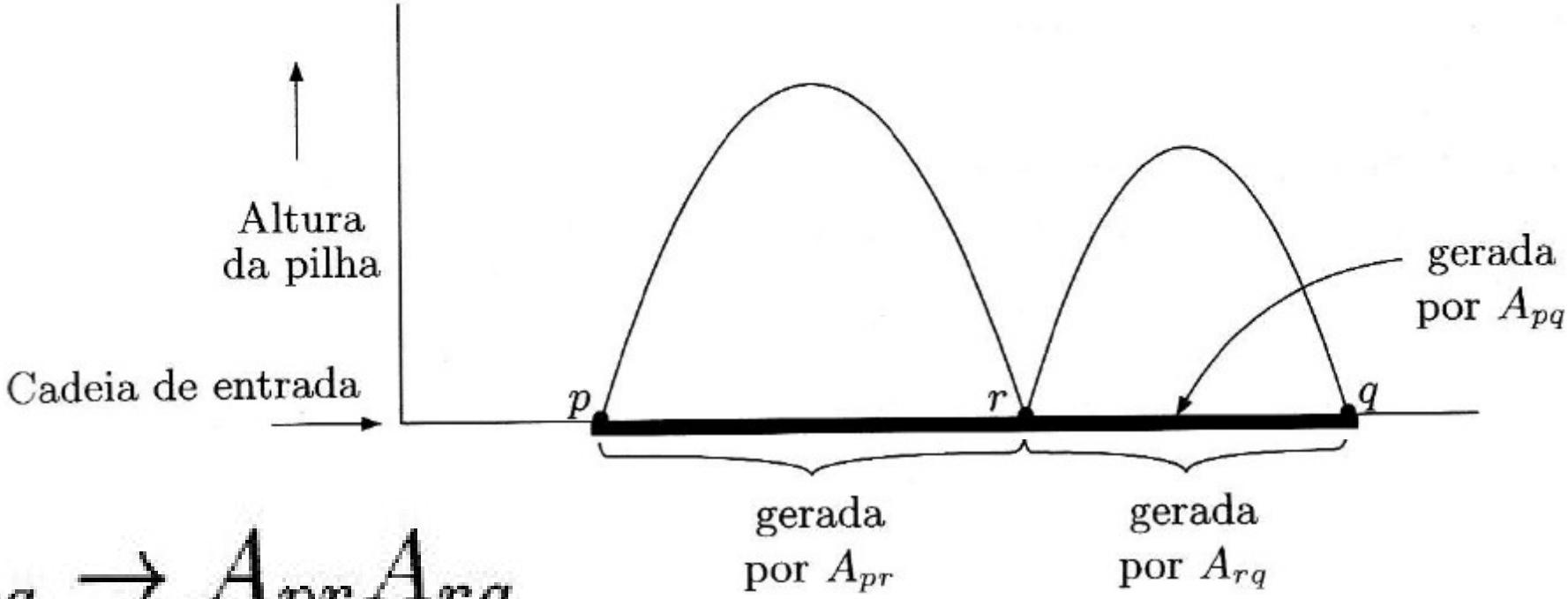
$$A_{pq} \rightarrow aA_{rs}b$$

Conversão APN em GLC (ideia)

- No caminho de p a q (reconhecendo x), 2 situações:
  - A pilha só se torna vazia novamente quando chega em q
  - A pilha se torna vazia em algum ponto do caminho, antes de chegar em q

Conversão APN em GLC (ideia)

- No caminho de p a q (reconhecendo x), 2 situações:
  - A pilha só se torna vazia novamente quando chega em q
  - A pilha se torna vazia em algum ponto do caminho, antes de chegar em q



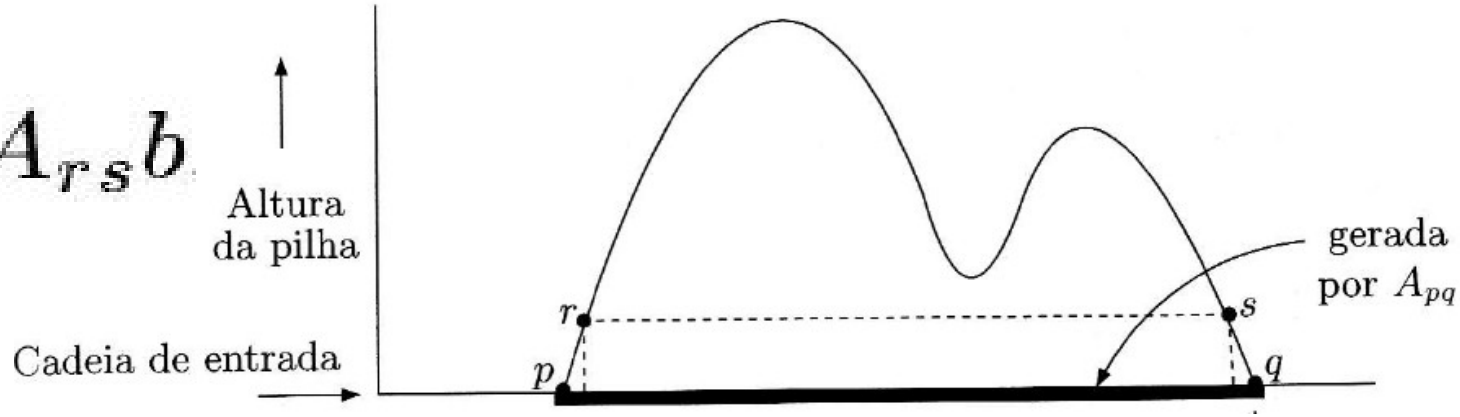
$$A_{pq} \rightarrow A_{pr} A_{rq}$$

Conversão APN em GLC (PROVA)

Passo da Indução: Assuma verdadeiro para computações de comprimento no máximo k, onde $k \geq 0$, e prove verdadeiro para computações de comprimento $k+1$.

Suponha que P tenha uma computação na qual x leva de p para q com pilhas vazias em $k+1$ passos. Ou a pilha está vazia apenas no início e no final dessa computação, ou ela se torna vazia em algum outro ponto também.

$$A_{pq} \rightarrow aA_{rs}b$$

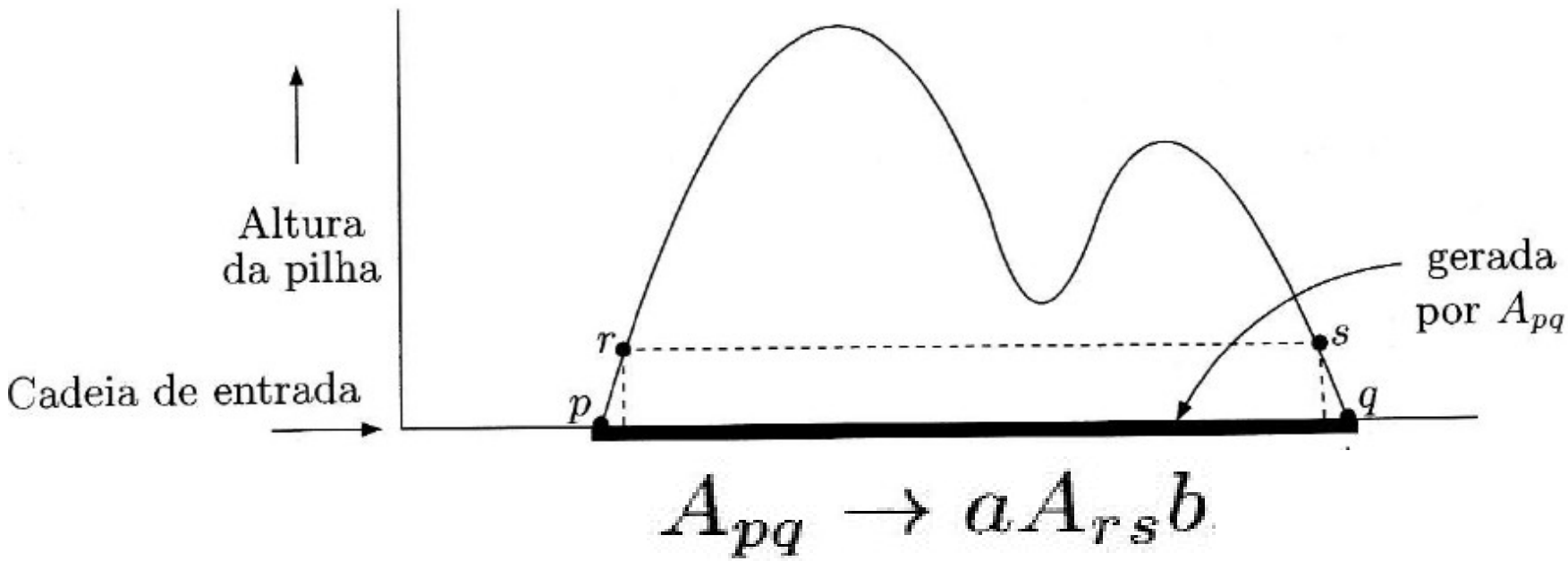


Note que se a cadeia x é uma substring da cadeia w que estamos analisando e p e q não forem os estados inicial e final respectivamente, p não necessariamente está com uma pilha vazia, mas com um conteúdo que NÃO SERÁ MEXIDO até que o estado q seja alcançado e a substring x seja reconhecida ao chegar em q. Ou seja, a SUBSTRING X seria reconhecida de p a q saindo de pilha vazia e chegando a pilha vazia se estes fossem iniciais e finais respectivamente. Ex: é o que acontecerá dos estados r a s nesta figura.

Conversão APN em GLC (PROVA)

PROVA Digamos que $P = (Q, \Sigma, \Gamma, \delta, q_0, \{q_{\text{aceita}}\})$ e vamos construir G. As variáveis de G são $\{A_{pq} \mid p, q \in Q\}$. A variável inicial é $A_{q_0, q_{\text{aceita}}}$. Agora descrevemos as regras de G.

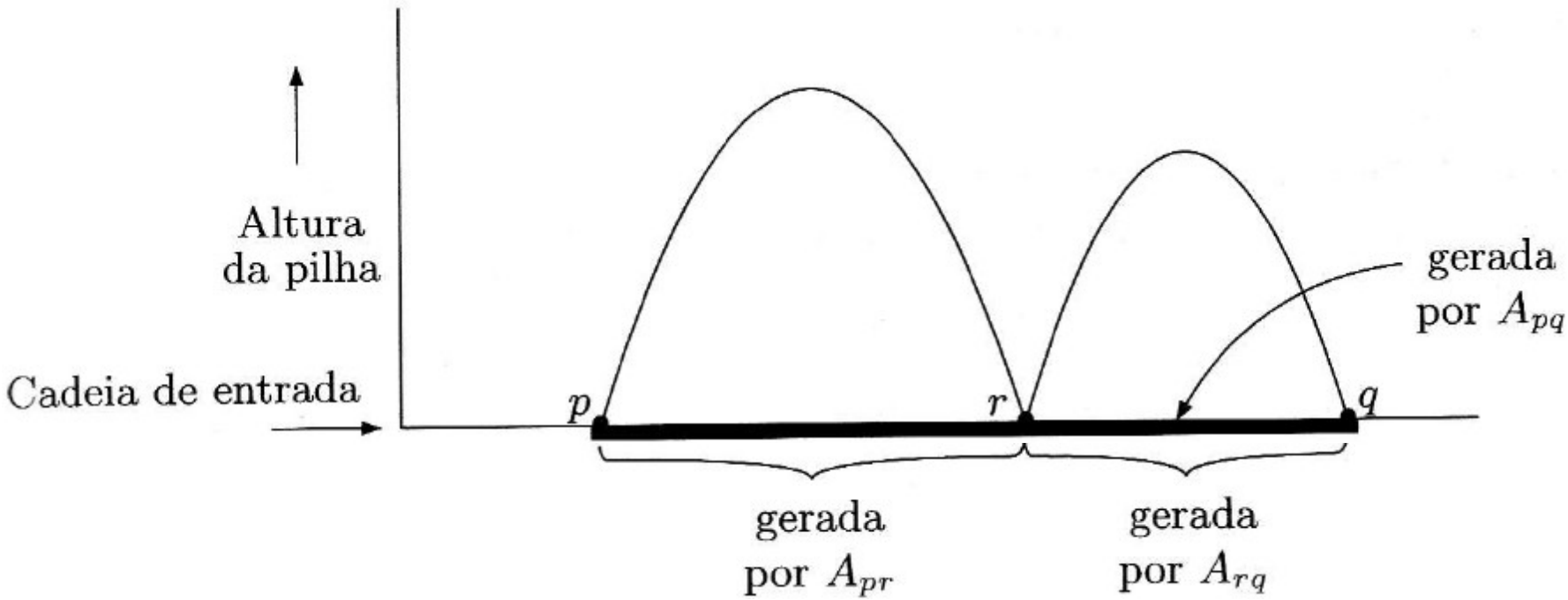
- Para cada $p, q, r, s \in Q$, $t \in \Gamma$ e $a, b \in \Sigma_\varepsilon$, se $\delta(p, a, \varepsilon)$ contém (r, t) e $\delta(s, b, t)$ contém (q, ε), ponha a regra $A_{pq} \to aA_{rs}b$ em G.



Conversão APN em GLC (PROVA)

PROVA Digamos que $P = (Q, \Sigma, \Gamma, \delta, q_0, \{q_{\text{aceita}}\})$ e vamos construir G. As variáveis de G são $\{A_{pq} \mid p, q \in Q\}$. A variável inicial é $A_{q_0, q_{\text{aceita}}}$. Agora descrevemos as regras de G.

- Para cada $p, q, r \in Q$, ponha a regra $A_{pq} \rightarrow A_{pr} A_{rq}$ em G.



Mesmo que algumas não sejam usadas

Conversão APN em GLC (PROVA)

PROVA Digamos que $P = (Q, \Sigma, \Gamma, \delta, q_0, \{q_{\text{aceita}}\})$ e vamos construir G. As variáveis de G são $\{A_{pq} \mid p, q \in Q\}$. A variável inicial é $A_{q_0, q_{\text{aceita}}}$. Agora descrevemos as regras de G.

- Finalmente, para cada $p \in Q$, ponha a regra $A_{pp} \to \varepsilon$ em G.

Para garantir que todas as formas sentenciais sejam uma hora transformadas em cadeias de terminais.

Conversão APN em GLC (PROVA)

PROVA Digamos que $P = (Q, \Sigma, \Gamma, \delta, q_0, \{q_{\text{aceita}}\})$ e vamos construir G. As variáveis de G são $\{A_{pq} \mid p, q \in Q\}$. A variável inicial é $A_{q_0, q_{\text{aceita}}}$. Agora descrevemos as regras de G.

- Para cada $p, q, r, s \in Q$, $t \in \Gamma$ e $a, b \in \Sigma_\varepsilon$, se $\delta(p, a, \varepsilon)$ contém (r, t) e $\delta(s, b, t)$ contém (q, ε), ponha a regra $A_{pq} \to aA_{rs}b$ em G.
- Para cada $p, q, r \in Q$, ponha a regra $A_{pq} \to A_{pr}A_{rq}$ em G.
- Finalmente, para cada $p \in Q$, ponha a regra $A_{pp} \to \varepsilon$ em G.

Conversão APN em GLC (PROVA)

Temos que provar que essa construção funciona, ou seja, que A_{pq} gera x se e somente se x pode levar o APN de p (com pilha vazia) a q (com pilha vazia).

AFIRMATIVA 2.30

Se A_{pq} gera x, então x pode levar P de p com pilha vazia a q com pilha vazia.

(=>)

AFIRMATIVA 2.31

Se x pode levar P de p com pilha vazia para q com pilha vazia, A_{pq} gera x.

(<=)

Conversão APN em GLC (PROVA)

AFIRMATIVA 2.30

Se A_{pq} gera x, então x pode levar P de p com pilha vazia a q com pilha vazia.

(=>)

Provamos essa afirmação por indução sobre o número de passos na derivação de x a partir de A_{pq}.

Base: A derivação tem 1 passo.
Uma derivação com um único passo tem de usar uma regra cujo lado direito não contém variáveis. As únicas regras em G onde nenhuma variável ocorre no lado direito são $A_{pp} \rightarrow \varepsilon$. Claramente, a entrada ε leva P de p com pilha vazia a p com pilha vazia e, portanto, a base está provada.

AFIRMATIVA 2.30

Se A_{pq} gera x, então x pode levar P de p com pilha vazia a q com pilha vazia.

Passo da Indução: Assuma verdadeiro para derivações de comprimento no máximo k, onde $k \geq 1$, e prove verdadeiro para derivações de comprimento $k+1$. Suponha que $A_{pq} \overset{*}{\Rightarrow} x$ com $k+1$ passos. O primeiro passo nessa derivação é ou $A_{pq} \Rightarrow aA_{rs}b$ ou $A_{pq} \Rightarrow A_{pr}A_{rq}$. Lidamos com esses dois casos separadamente.

AFIRMATIVA 2.30

Se A_{pq} gera x, então x pode levar P de p com pilha vazia a q com pilha vazia.

Passo da Indução: Assuma verdadeiro para derivações de comprimento no máximo k, onde $k \geq 1$, e prove verdadeiro para derivações de comprimento $k+1$. Suponha que $A_{pq} \overset{*}{\Rightarrow} x$ com $k+1$ passos. O primeiro passo nessa derivação é ou $A_{pq} \Rightarrow aA_{rs}b$ ou $A_{pq} \Rightarrow A_{pr}A_{rq}$. Lidamos com esses dois casos separadamente.

No primeiro caso, considere a parte y de x que A_{rs} gera, de forma que $x = ayb$. Em razão do fato de que $A_{rs} \overset{*}{\Rightarrow} y$ com k passos, a hipótese da indução nos diz que P pode ir de r com pilha vazia para s com pilha vazia. Em razão do fato de que $A_{pq} \rightarrow aA_{rs}b$ é uma regra de G, $\delta(p, a, \varepsilon)$ contém (r, t) e $\delta(s, b, t)$ contém (q, ε), para algum símbolo de pilha t. Logo, se P começa em p com uma pilha vazia, após ler a ele pode ir para o estado r e empilhar t. Então, ler a cadeia y pode levá-lo a s e deixar t na pilha. E após ler b ele pode ir para o estado q e desempilhar t. Conseqüentemente, x pode levá-lo de p com pilha vazia para q com pilha vazia.

AFIRMATIVA 2.30

Se A_{pq} gera x, então x pode levar P de p com pilha vazia a q com pilha vazia.

Passo da Indução: Assuma verdadeiro para derivações de comprimento no máximo k, onde $k \geq 1$, e prove verdadeiro para derivações de comprimento $k+1$. Suponha que $A_{pq} \overset{*}{\Rightarrow} x$ com $k+1$ passos. O primeiro passo nessa derivação é ou $A_{pq} \Rightarrow aA_{rs}b$ ou $A_{pq} \Rightarrow A_{pr}A_{rq}$. Lidamos com esses dois casos separadamente.

No segundo caso, considere as partes y e z de x que A_{pr} e A_{rq}, respectivamente, geram, de forma que $x = yz$. Como $A_{pr} \overset{*}{\Rightarrow} y$ em no máximo k passos e $A_{rq} \overset{*}{\Rightarrow} z$ em no máximo k passos, a hipótese da indução nos diz que y pode levar P de p para r e z pode levar P de r para q, com pilha vazia no início e no final. Logo, x pode levá-lo de p com pilha vazia para q com pilha vazia. Isso completa o passo da indução.

Conversão APN em GLC (PROVA)

AFIRMATIVA 2.31

Se x pode levar P de p com pilha vazia para q com pilha vazia, A_{pq} gera x.

(<=)

Provamos essa afirmação por indução sobre o número de passos na computação de P que vai de p para q com pilhas vazias sobre a entrada x.

Base: A computação tem 0 passos.
Se uma computação tem 0 passos, ela começa e termina no mesmo estado—digamos, p. Portanto, temos que mostrar que $A_{pp} \overset{*}{\Rightarrow} x$. Em 0 passos, P só tem tempo de ler a cadeia vazia, portanto $x = \varepsilon$. Por construção, G tem a regra $A_{pp} \rightarrow \varepsilon$, portanto a base está provada.

AFIRMATIVA 2.31

Se x pode levar P de p com pilha vazia para q com pilha vazia, A_{pq} gera x.

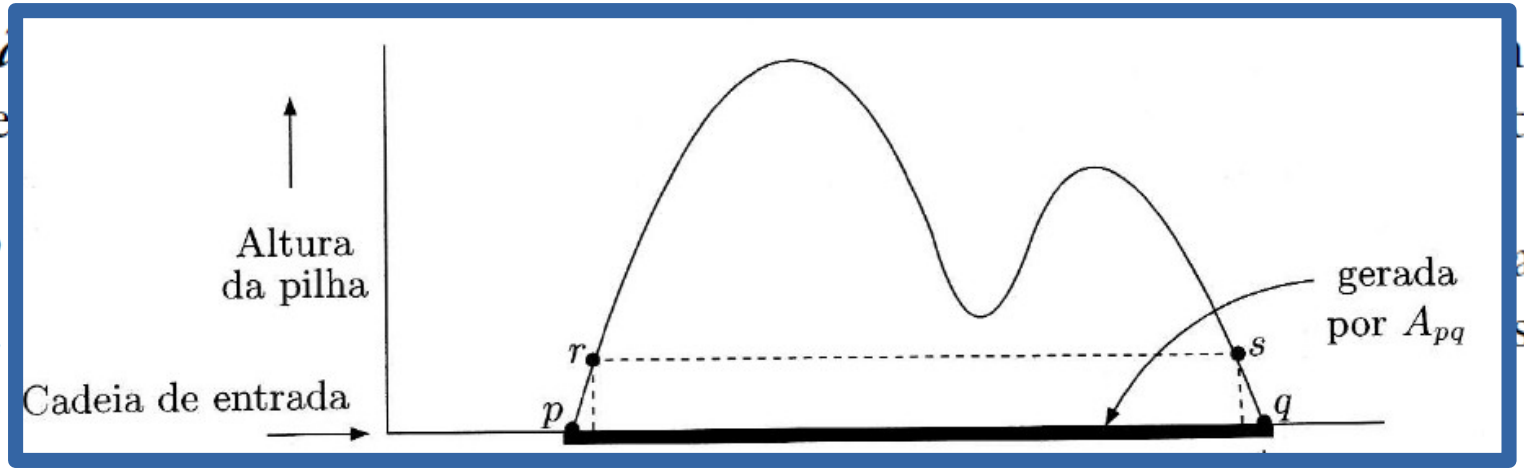
Passo da Indução: Assuma verdadeiro para computações de comprimento no máximo k, onde $k \geq 0$, e prove verdadeiro para computações de comprimento $k + 1$.

Suponha que P tenha uma computação na qual x leva de p para q com pilhas vazias em $k + 1$ passos. Ou a pilha está vazia apenas no início e no final dessa computação, ou ela se torna vazia em algum outro ponto também.

AFIRMATIVA **2.31**

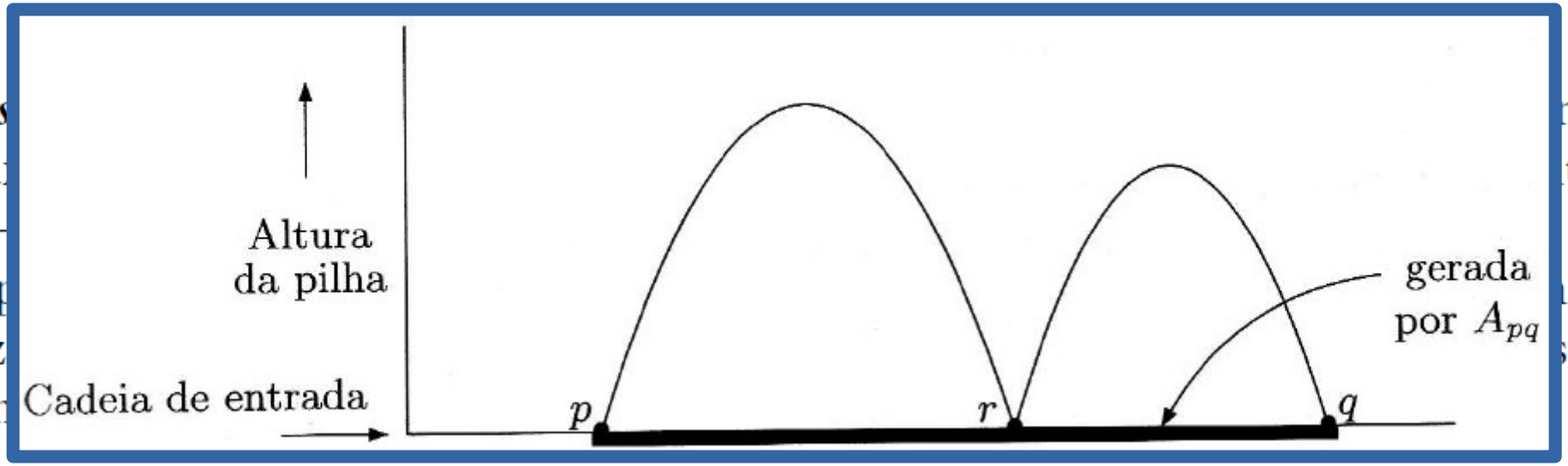
Se x pode levar P de p com pilha vazia para q com pilha vazia, A_{pq} gera x.

Passo da Indução no máximo k, onde to $k+1$.
Suponha que P as vazias em $k+1$ sa computação, ou



Caso 1:

- O primeiro símbolo a ser empilhado (t) deve ser o último a ser desempilhado
- Chamemos de a e b os símbolos lidos da entrada no primeiro e último movimento, respectivamente
  - Chamemos o segundo estado r e o penúltimo estado s
- δ(p,a,ε) contém (r,t) e δ(s,b,t) contém (q,ε), logo $A_{pq} \rightarrow aA_{rs}b$ está em G
- x = ayb. y faz o APN ir de r a s, sem tocar em t, em k-1 passos
- Pela hipótese de indução, A_{rs} =>* y. Logo, A_{pq} =>* x



Caso 2:

- Chamemos de estado r onde a pilha fica vazia no meio da computação de x
- Cada caminho (de p a r, e de r a q) tem no máximo k passos, x = yz
- Pela hipótese de indução: $A_{pr} \Rightarrow^* y$ e $A_{rq} \Rightarrow^* z$.
- Como $A_{pq} \rightarrow A_{pr}A_{rq}$ está em G, $A_{pq} \Rightarrow^* x$

Equivalência entre APN e GLC

TEOREMA 2.20

Uma linguagem é livre-do-contexto se e somente se algum autômato com pilha a reconhece.

LEMA 2.21

Se uma linguagem é livre-do-contexto, então algum autômato com pilha a reconhece.

(=>)

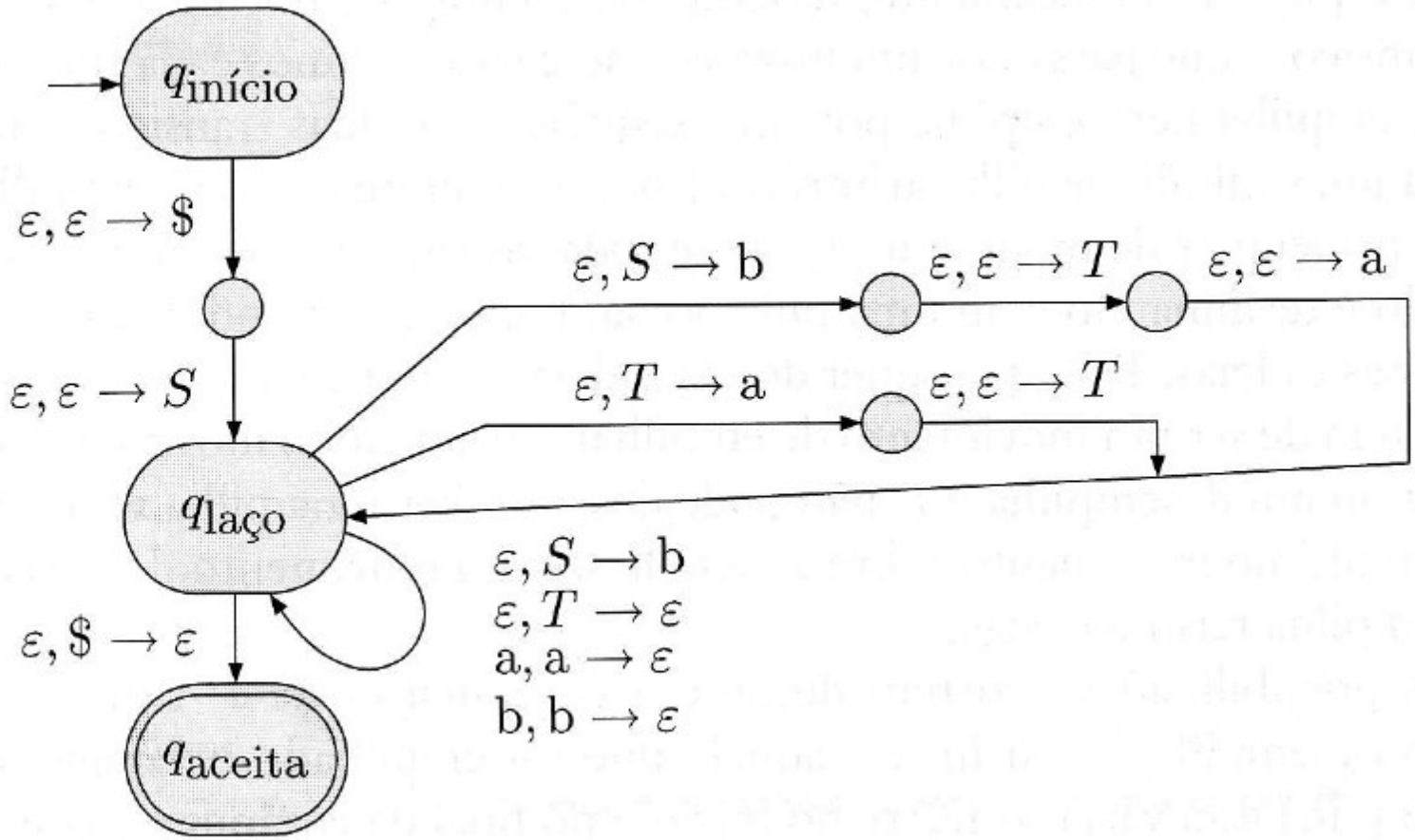
LEMA 2.27

Se um autômato com pilha reconhece alguma linguagem, então ela é livre-do-contexto.

(<=)

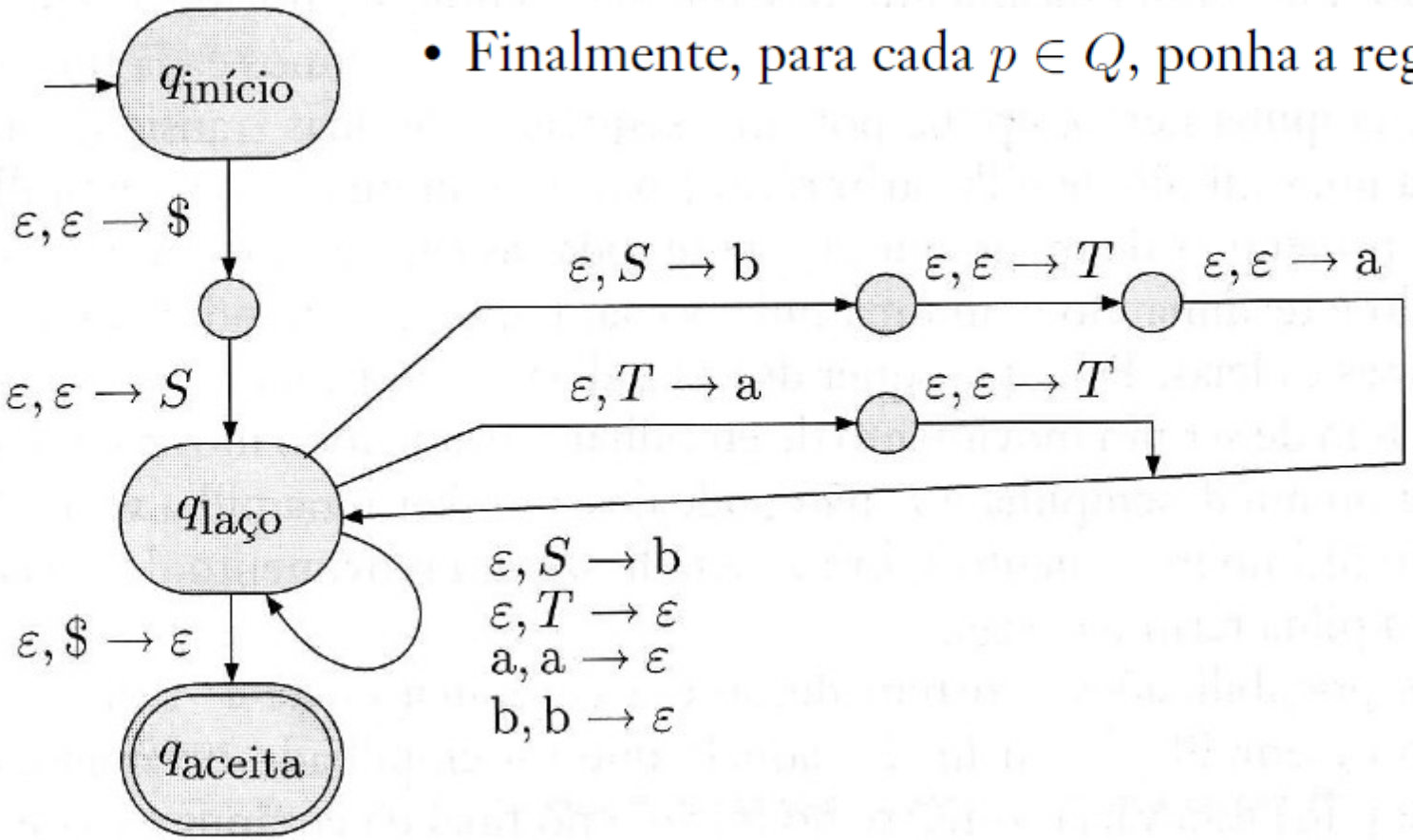
Exercício

- Transforme o seguinte AP em uma GLC equivalente (numere os estados que não possuem rótulos)



Exercício (com uma “colinha”)

- Transforme o seguinte AP em uma GLC equivalente (numere os estados que não possuem rótulos)

A variável inicial é $A_{q_0, q_{\text{aceita}}}$

- Para cada $p, q, r, s \in Q$, $t \in \Gamma$, e $a, b \in \Sigma_\varepsilon$, se $\delta(p, a, \varepsilon)$ contém (r, t) e $\delta(s, b, t)$ contém (q, ε), ponha a regra $A_{pq} \to aA_{rs}b$ em G.
- Para cada $p, q, r \in Q$, ponha a regra $A_{pq} \to A_{pr}A_{rq}$ em G.
- Finalmente, para cada $p \in Q$, ponha a regra $A_{pp} \to \varepsilon$ em G.