ACH2043
INTRODUÇÃO À TEORIA DA COMPUTAÇÃO

Aula 14
Equivalência APN e GLC

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

Aulas passadas...

Gramáticas Livres de Contexto

- Definição: uma gramática G é uma quádrupla (V, Σ, S, P), onde
  - V é o conjunto de símbolos não-terminais (variáveis)
  - Σ é o conjunto de símbolos terminais
  - S é o símbolo inicial
  - P é o conjunto de produções da forma

    V → (Σ U V)*

$$
\begin{aligned}
A &\rightarrow 0A1 \\
A &\rightarrow B \\
B &\rightarrow \#
\end{aligned}
$$

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

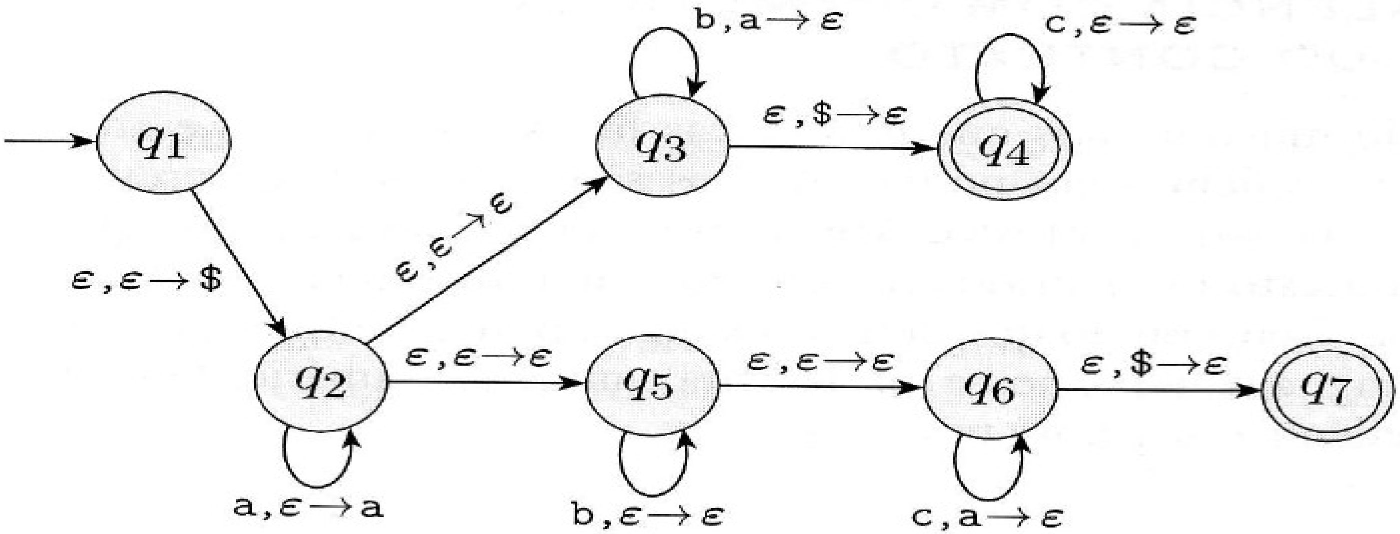
1. Q é o conjunto de estados,
2. Σ é o alfabeto de entrada,
3. Γ é o alfabeto de pilha,
4. $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \longrightarrow \mathcal{P}(Q \times \Gamma_\varepsilon)$ é a função de transição,
5. $q_0 \in Q$ é o estado inicial, e
6. $F \subseteq Q$ é o conjunto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

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!



Aula de hoje...

Equivalência entre APN e GLC

TEOREMA 2.20 ....

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

<=>

Isto é, se uma linguagem é gerada por uma gramática livre de contexto...

AP NÃO DETERMINÍSTICO!!!

Equivalência entre APN e GLC

TEOREMA 2.20 ....

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

PROVAREMOS ESTE! (=>)

LEMA 2.21 ....

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

GLC => APN

LEMA 2.27 ....

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

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)

- Uma gramática aceita uma cadeia w se, começando pela variável inicial, chega-se a uma cadeia apenas de símbolos terminais (w) após uma sequência de derivações diretas (substituições de variáveis).

$\underline{S}$
$S \; ; \; \underline{S}$
$\underline{S} \; ; \; \mathrm{id} \; := E$
$\mathrm{id} := \underline{E} \; ; \; \mathrm{id} := E$
$\mathrm{id} := \mathrm{num} \; ; \; \mathrm{id} := \underline{E}$
$\mathrm{id} := \mathrm{num} \; ; \; \mathrm{id} := E + \underline{E}$
$\mathrm{id} := \mathrm{num} \; ; \; \mathrm{id} := \underline{E} + (S \, , \; E)$
$\mathrm{id} := \mathrm{num} \; ; \; \mathrm{id} := \mathrm{id} + (\underline{S} \, , \; E)$
$\mathrm{id} := \mathrm{num} \; ; \; \mathrm{id} := \mathrm{id} + (\mathrm{id} := \underline{E} \, , \; E)$
$\mathrm{id} := \mathrm{num} \; ; \; \mathrm{id} := \mathrm{id} + (\mathrm{id} := E + E \, , \; \underline{E})$
$\mathrm{id} := \mathrm{num} \; ; \; \mathrm{id} := \mathrm{id} + (\mathrm{id} := \underline{E} + E \, , \; \mathrm{id})$
$\mathrm{id} := \mathrm{num} \; ; \; \mathrm{id} := \mathrm{id} + (\mathrm{id} := \mathrm{num} + \underline{E} \, , \; \mathrm{id})$
$\mathrm{id} := \mathrm{num} \; ; \; \mathrm{id} := \mathrm{id} + (\mathrm{id} := \mathrm{num} + \mathrm{num} \, , \; \mathrm{id})$

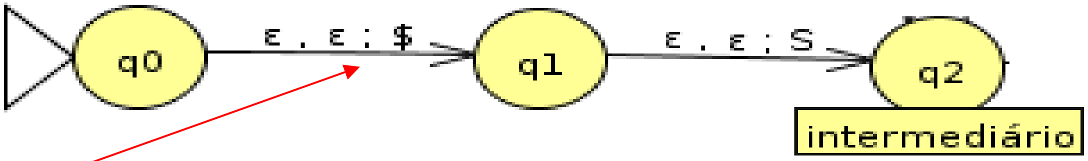
	4 $E \rightarrow \mathrm{id}$	
1 $S \rightarrow S \; ; \; S$	5 $E \rightarrow \mathrm{num}$	8 $L \rightarrow E$
2 $S \rightarrow \mathrm{id} := E$	6 $E \rightarrow E + E$	9 $L \rightarrow L \, , \; E$
3 $S \rightarrow \mathrm{print} \; (L)$	7 $E \rightarrow (S \, , \; E)$	

Conversão GLC em APN (ideia)

- Uma gramática aceita uma cadeia w se, começando pela variável inicial, chega-se a uma cadeia apenas de símbolos terminais (w) após uma sequência de derivações diretas (substituições de variáveis).
- Um autômato aceita uma cadeia w se, começando pelo estado inicial, chega-se ao estado final após uma sequência de mudança de estados (transições)
- **Ideia:** Simular cada substituição (da GLC) por uma transição (do APN) – usa a pilha como apoio para fazer as substituições

Conversão GLC em APN (ideia)

- O APN empilha “$” (estado q0), e depois empilha a variável inicial na pilha (na transição do estado q1 para o estado intermediário)



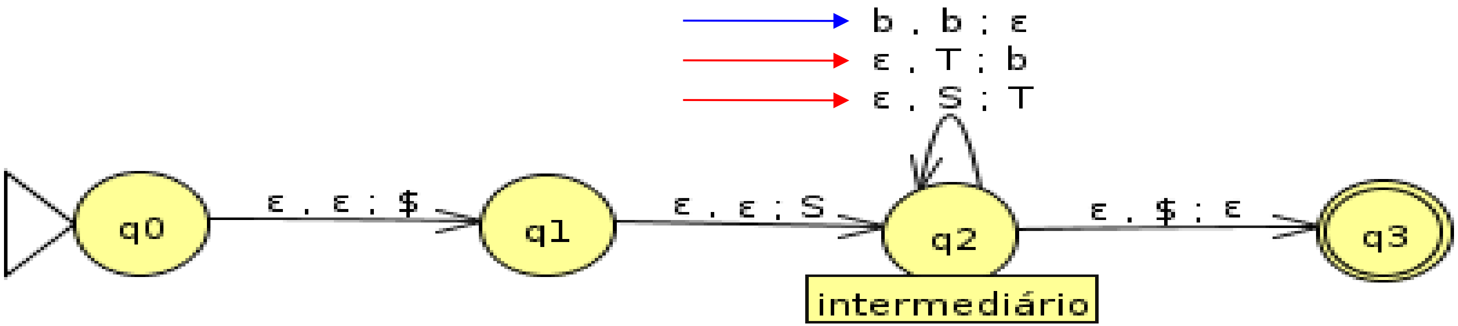
JFlap usa “;” no lugar de “→” para separar os símbolos do desempilha/empilha

Conversão GLC em APN (ideia)

- O APN empilha “$” (estado q0), e depois empilha a variável inicial na pilha (na transição do estado q1 para o estado intermediário)
- O estado intermediário possui transições para ele mesmo, em cada uma fazendo uma substituição (derivação) na cadeia que está na pilha ou desempilhando um símbolo do alfabeto se este bater com o próximo símbolo de entrada
- O APN vai para o estado final quando não há mais substituições a serem feitas (pilha tem “$”) e a cadeia de entrada acabou
- Ex:

  $S \rightarrow T$

  $T \rightarrow b$



Conversão GLC em APN (ideia)

- O APN empilha “$” (estado q0), e depois empilha a variável inicial na pilha (na transição do estado q1 para o estado intermediário)
- O estado intermediário possui transições para ele mesmo, em cada uma fazendo uma substituição (derivação) na cadeia que está na pilha ou desempilhando um símbolo do alfabeto se este bater com o próximo símbolo de entrada

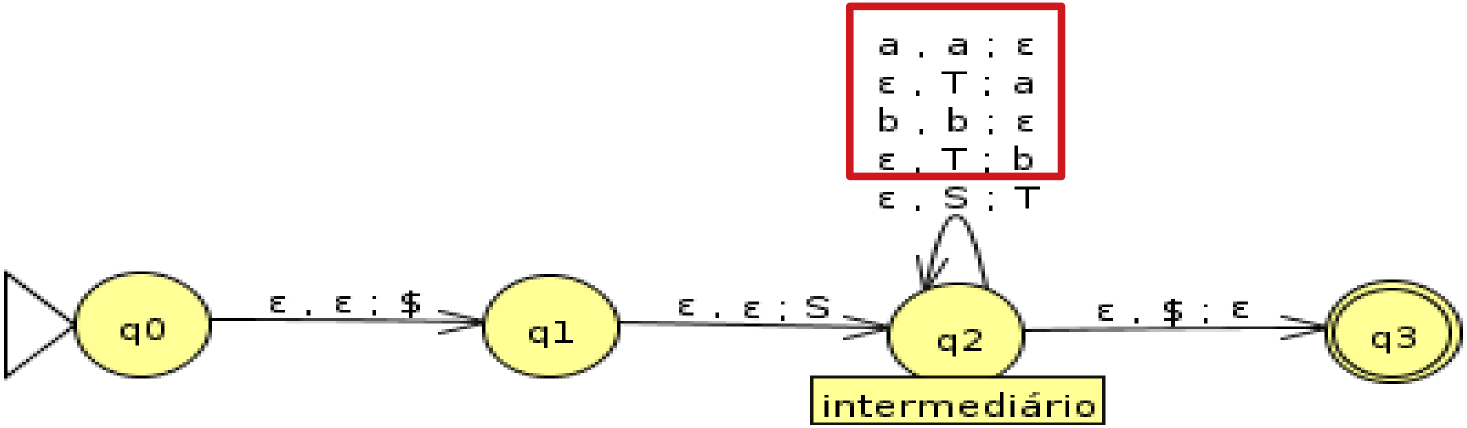
Isso corresponde à derivação mais à esquerda!!!

$$
\begin{array}{l}
\underline{S} \\
\underline{S} \; ; \; S \\
\mathrm{id} \; := \; \underline{E} \; ; \; S \\
\mathrm{id} \; := \; \mathrm{num} \; ; \; \underline{S} \\
\mathrm{id} \; := \; \mathrm{num} \; ; \; \mathrm{id} \; := \; \underline{E} \\
\mathrm{id} \; := \; \mathrm{num} \; ; \; \mathrm{id} \; := \; \underline{E} + E \\
\quad\quad \vdots
\end{array}
$$

Conversão GLC em APN (ideia)

- Problema 1: O que fazer quando há várias opções de substituições?
  - Aproveitar o não determinismo

S → T
T → b | a



Conversão GLC em APN (ideia)

- Problema 2: Como empilhar uma cadeia, e não simplesmente um símbolo?

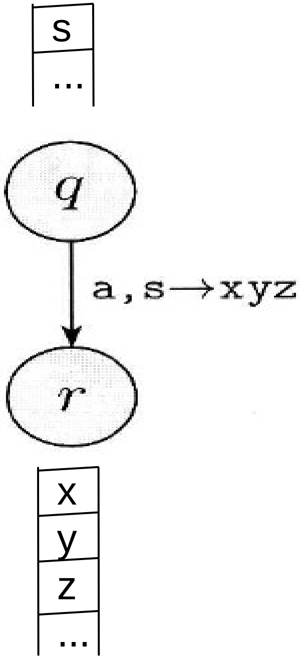
Ex:
$$S \to \mathtt{a}T\mathtt{b} \mid \mathtt{b}$$
$$T \to T\mathtt{a} \mid \varepsilon$$

Conversão GLC em APN (ideia)

- Problema 2: Como empilhar uma cadeia, e não simplesmente um símbolo?

Ex: quero desempilhar s e empilhar xyz (neste exemplo, ao ler o símbolo a)

s → xyz



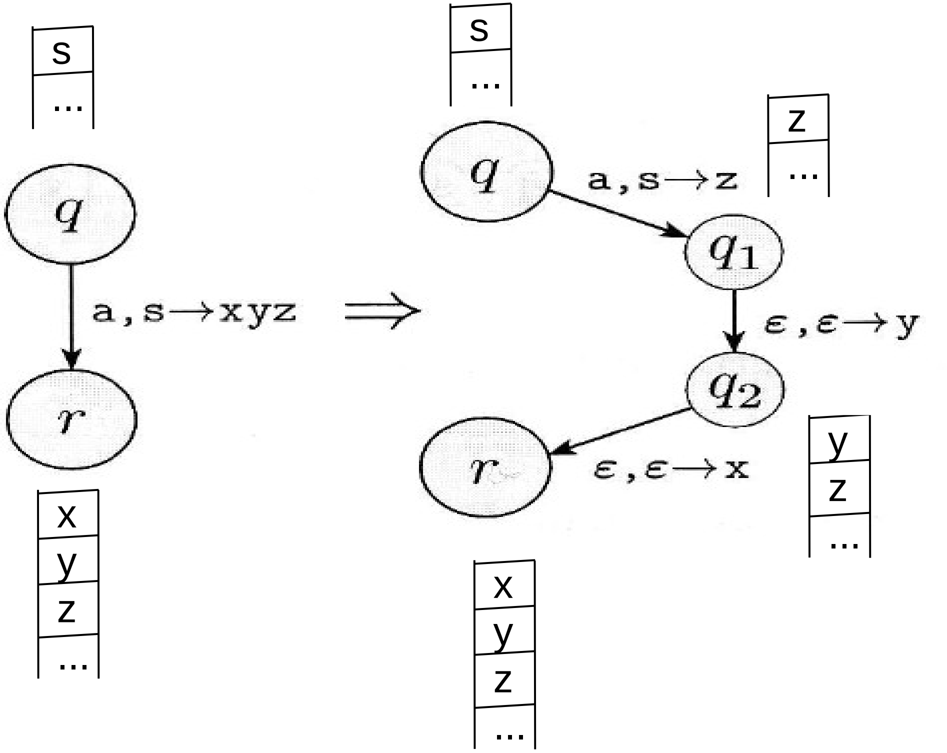
Profa. Ariane Machado Lima

Conversão GLC em APN (ideia)

- Problema 2: Como empilhar uma cadeia, e não simplesmente um símbolo?

Ex: quero desempilhar s e empilhar xyz (neste exemplo, ao ler o símbolo a)

s → xyz

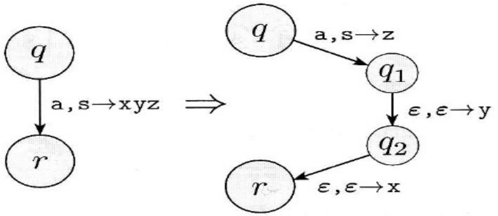


Conversão GLC em APN (ideia)

Problema 2: Como empilhar uma cadeia, e não simplesmente um símbolo?

‘Ex: quero desempilhar s e empilhar xyz (neste exemplo, ao ler o símbolo a)

s → xyz



Sejam q e r estados do AP e suponha que a esteja em Σ_ε e s em Γ_ε. Digamos que queiramos que o AP vá de q para r quando ele lê a e desempilha s. Além do mais, queremos empilhar a cadeia inteira $u = u_1 \cdots u_l$ ao mesmo tempo. Podemos implementar essa ação introduzindo novos estados $q_1, \ldots, q_{l-1}$ e montando a tabela de transição da seguinte maneira

$$\delta(q, a, s) \text{ deve conter } (q_1, u_l),$$
$$\delta(q_1, \varepsilon, \varepsilon) = \{(q_2, u_{l-1})\},$$
$$\delta(q_2, \varepsilon, \varepsilon) = \{(q_3, u_{l-2})\},$$
$$\vdots$$
$$\delta(q_{l-1}, \varepsilon, \varepsilon) = \{(r, u_1)\}.$$

Conversão GLC em APN (ideia)

- Problema 3: Se só podemos ler o topo da pilha, o que fazer quando a primeira variável da forma sentencial não estiver no topo da pilha?

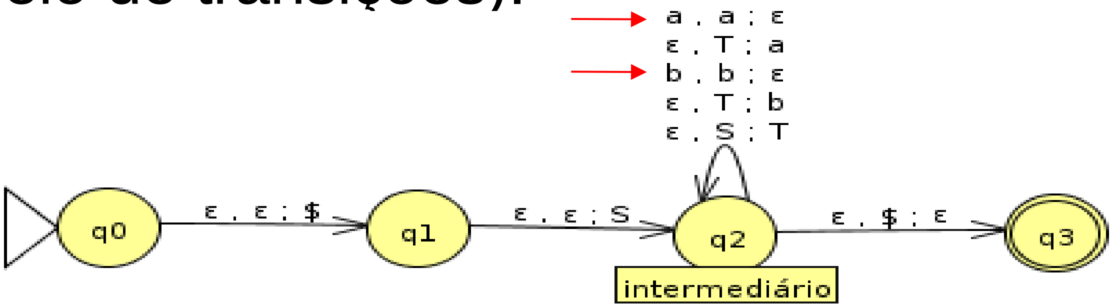
$$
\begin{aligned}
S &\rightarrow \underline{\mathtt{a}T\mathtt{b}} \mid \mathtt{b} \\
T &\rightarrow T\mathtt{a} \mid \varepsilon
\end{aligned}
$$

Conversão GLC em APN (ideia)

- Problema 3: Se só podemos ler o topo da pilha, o que fazer quando a primeira variável da forma sentencial não estiver no topo da pilha?

$$S \rightarrow \underline{aTb} \mid b$$
$$T \rightarrow Ta \mid \varepsilon$$

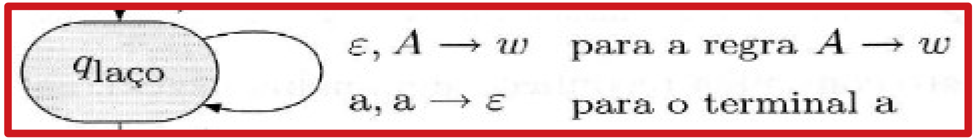
  - Sempre faremos a derivação mais à esquerda
  - Se o começo da forma sentencial contiver terminais, desempilho esses símbolos "casando-os" com a entrada (por meio de transições).



Conversão GLC em APN (ideia)

- Exemplo

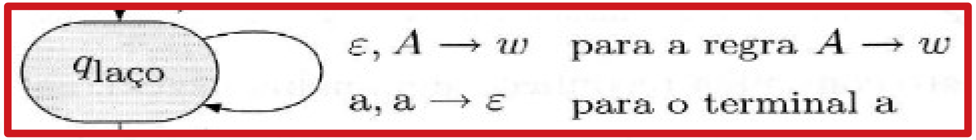
$$
\begin{aligned}
S &\rightarrow \mathtt{a}T\mathtt{b} \mid \mathtt{b} \\
T &\rightarrow T\mathtt{a} \mid \varepsilon
\end{aligned}
$$

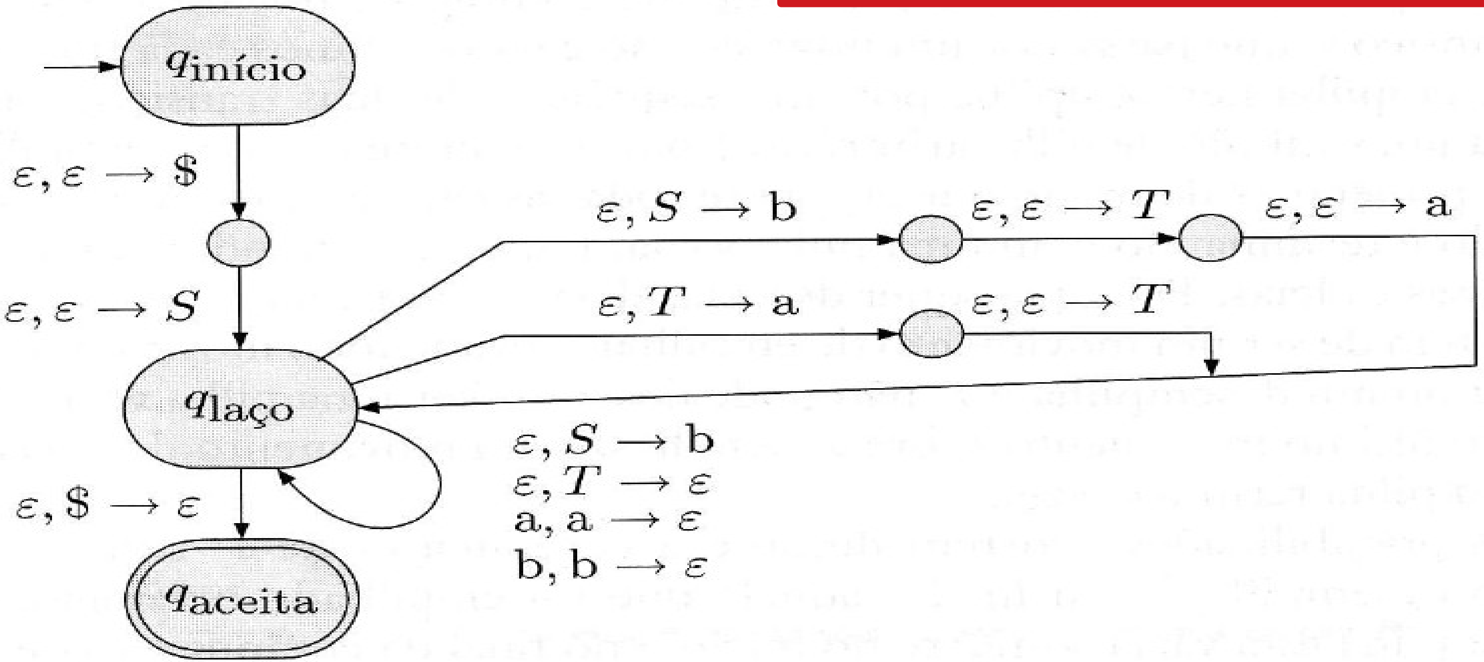


Conversão GLC em APN (ideia)

- Exemplo

$$S \rightarrow aTb \mid b$$
$$T \rightarrow Ta \mid \varepsilon$$

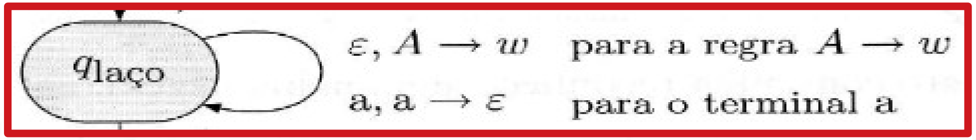


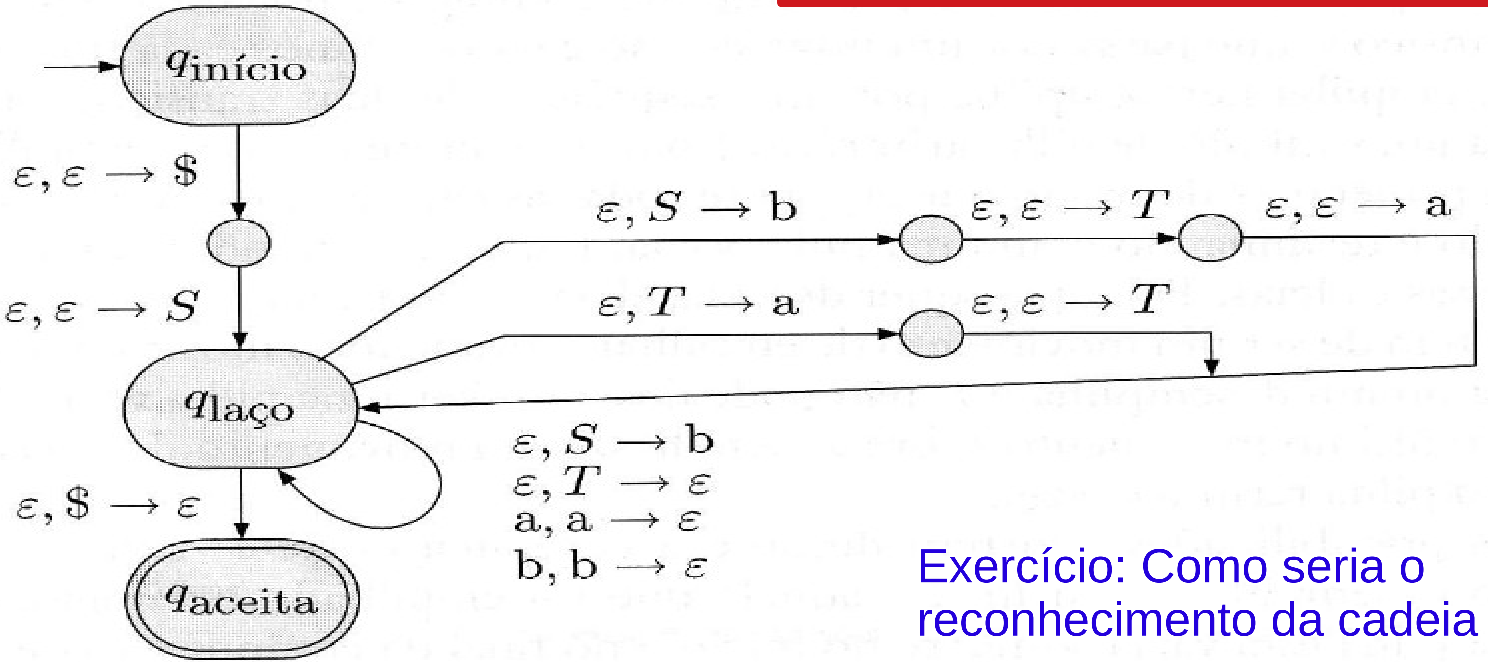


Conversão GLC em APN (ideia)

- Exemplo

$$S \rightarrow aTb \mid b$$
$$T \rightarrow Ta \mid \varepsilon$$

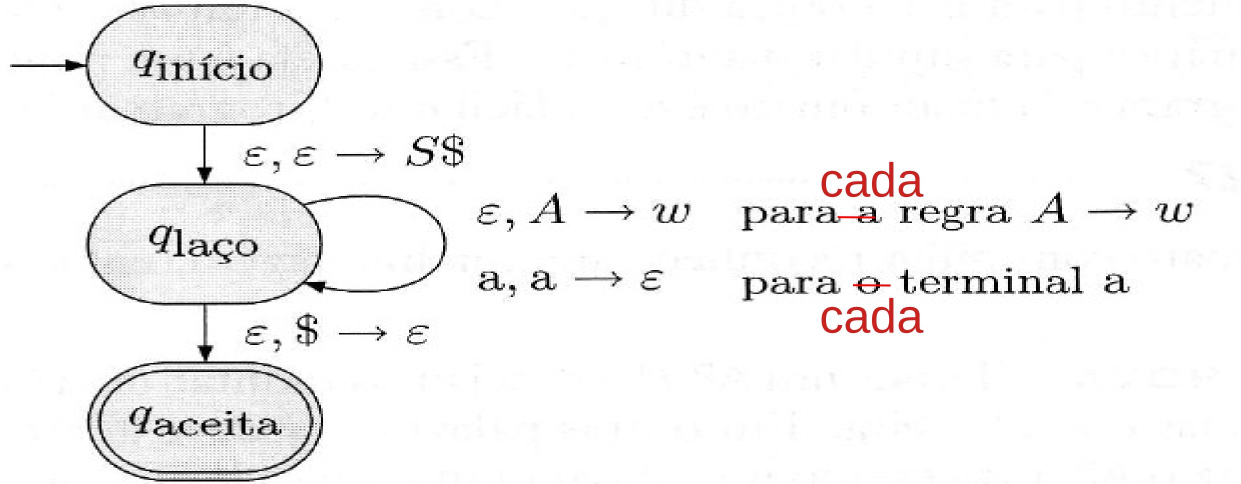




Exercício: Como seria o reconhecimento da cadeia aab?

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.

APN => GLC

FALTA PROVAR ESTE... (<=)

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.

Profa. Ariane Machado Lima

Conversão APN em GLC (ideia)

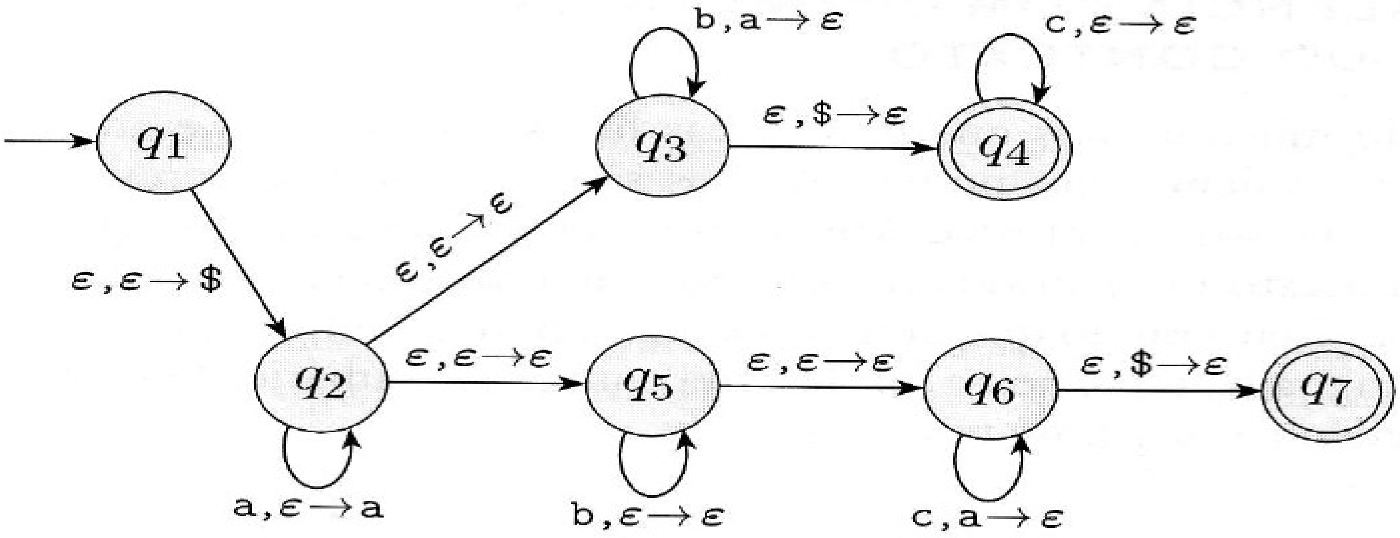
- Para facilitar, vamos considerar que o APN possui as seguintes características:

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.

Dar a P as características 1 e 2 é fácil. Para a característica 3, substituimos cada transição que simultaneamente desempilha e empilha por uma seqüência de duas transições que passa por um novo estado, e substituimos cada transição que nem desempilha nem empilha por uma seqüência de duas transições que empilha e depois desempilha um símbolo de pilha arbitrário.

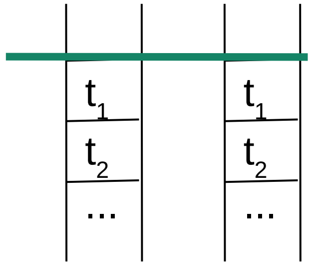
Exercício

- Transforme o seguinte AP para ter as características mencionadas no slide anterior



Conversão APN em GLC (ideia)

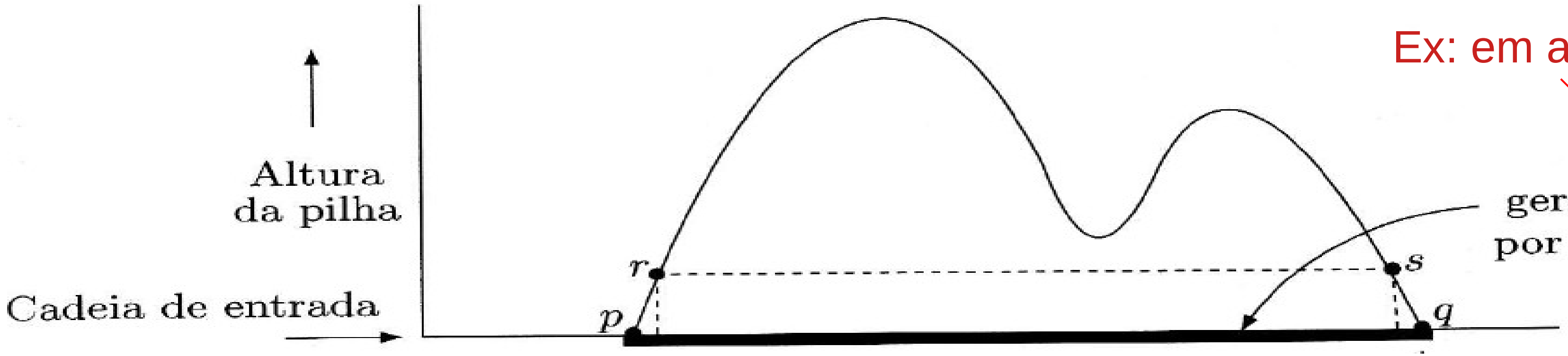
- G deve gerar uma cadeia w se w faz o APN ir do estado inicial ao estado de aceitação.
- Para cada par de estados (p, q), criamos uma variável A_{pq} que gera todas as cadeias x que levam o APN do estado p (com uma pilha "vazia") ao estado q (com uma pilha "vazia").
- Aqui, pilha "vazia" significa considerar só o espaço de pilha a partir do topo existente quando o estado atual é o estado p



- Um caminho do estado p ao estado q 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)

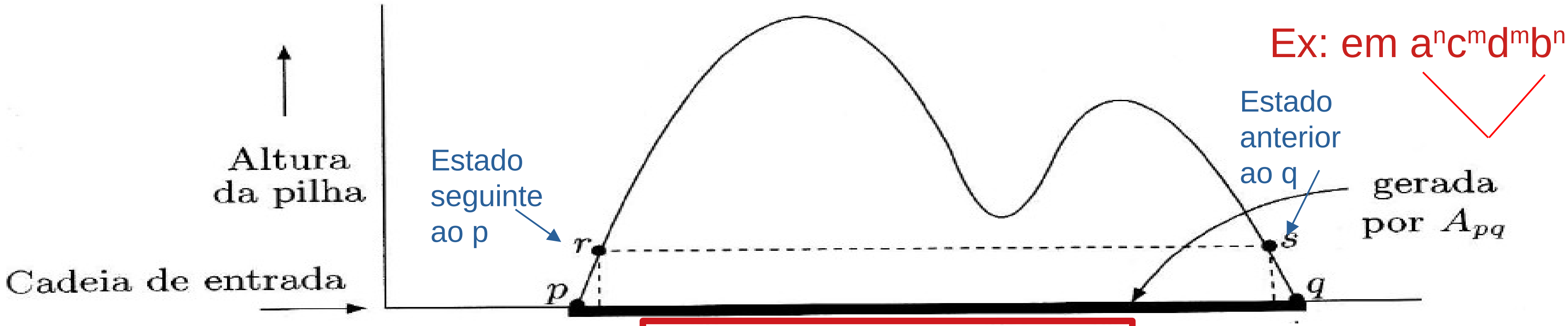
- No caminho de p a q (reconhecendo x), 2 situações:
  - A pilha só se torna vazia novamente quando chega em q
  - A pilha se torna vazia em algum ponto do caminho, antes de chegar em q



Ex: em $a^n c^m d^m b^n$

Conversão APN em GLC (ideia)

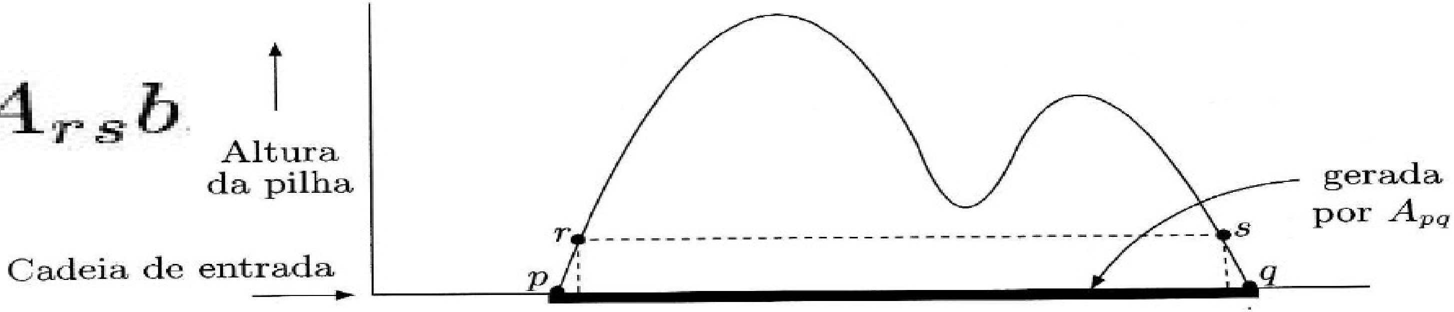
- No caminho de p a q (reconhecendo x), 2 situações:
  - A pilha só se torna vazia novamente quando chega em q
  - A pilha se torna vazia em algum ponto do caminho, antes de chegar em q



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

Conversão APN em GLC (ideia)

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

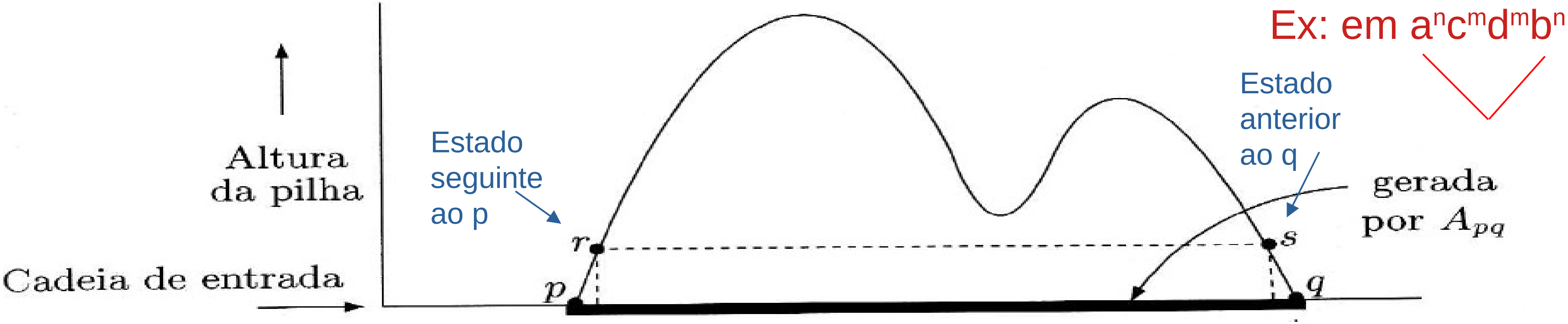


Note que se a cadeia x é uma substring da cadeia w que estamos analisando e p e q não forem os estados inicial e final respectivamente, p não necessariamente está com uma pilha vazia, mas com um conteúdo que NÃO SERÁ MEXIDO até que o estado q seja alcançado e a substring x seja reconhecida ao chegar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

Conversão APN em GLC (PROVA)

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



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

Profa. Ariane Machado Lima

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

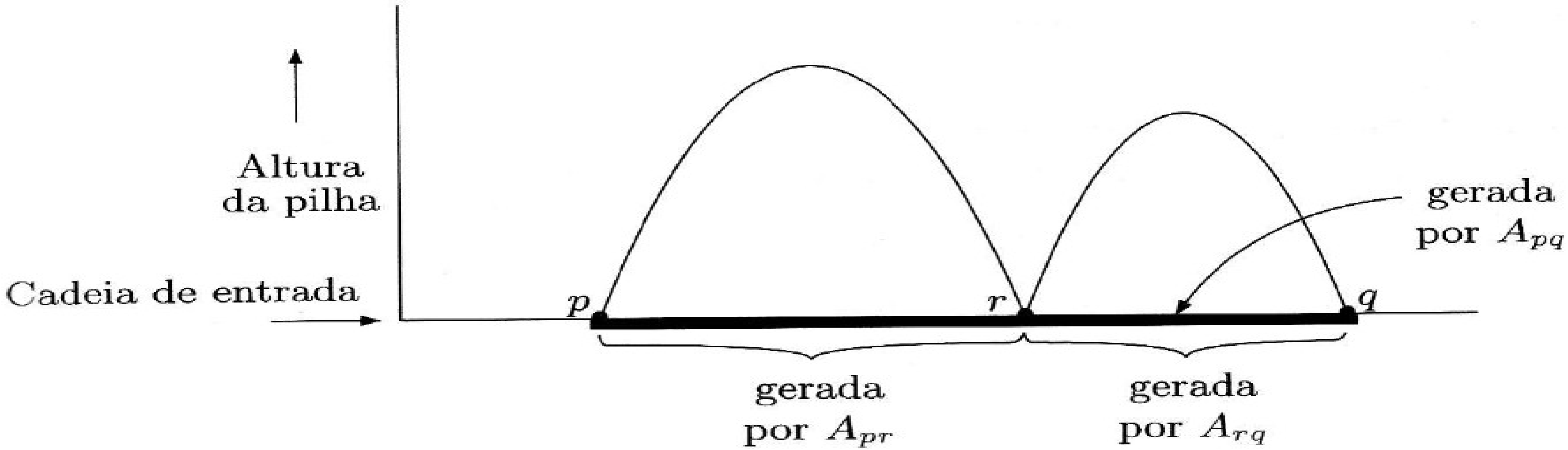
Ex: em $a^n b^n c^m d^m$

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

Conversão APN em GLC (PROVA)

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

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



Mesmo que algumas não sejam usadas

Conversão APN em GLC (PROVA)

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

(=>)

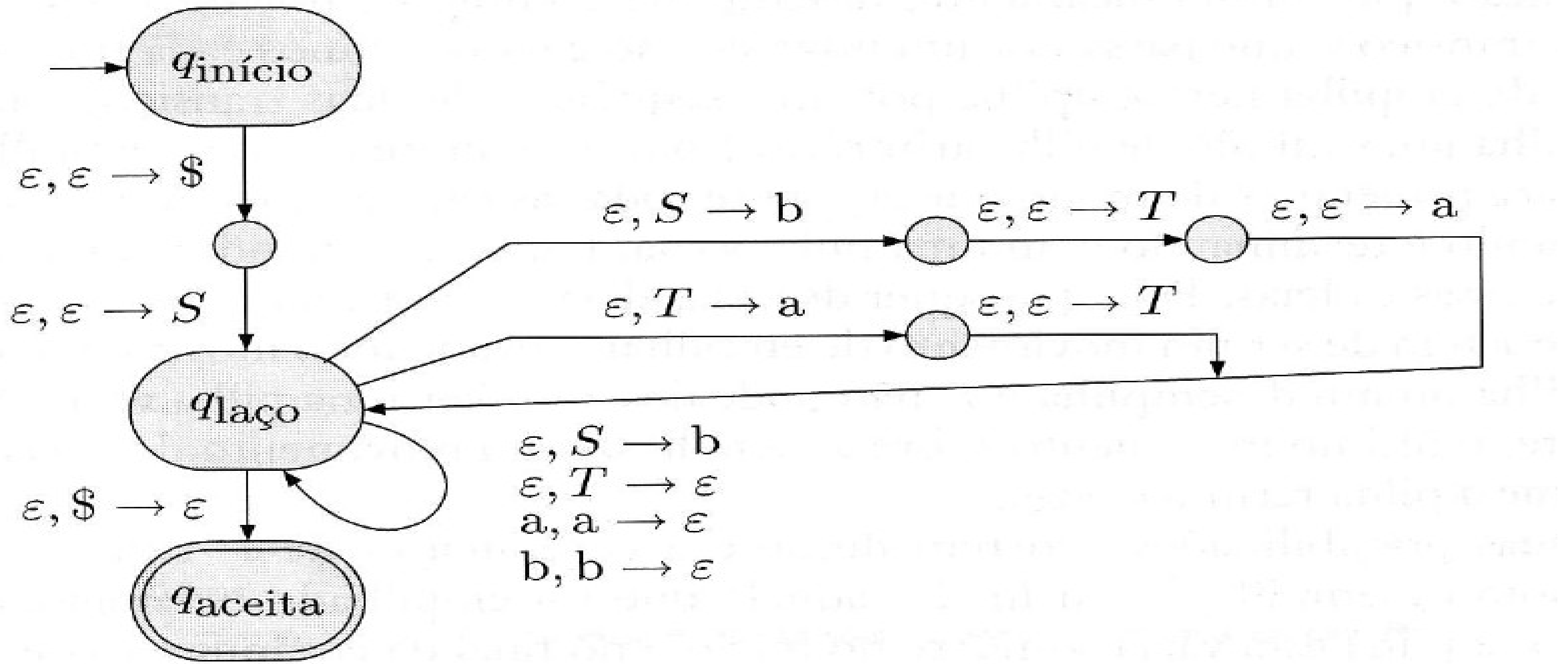
AFIRMATIVA 2.31 ....

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

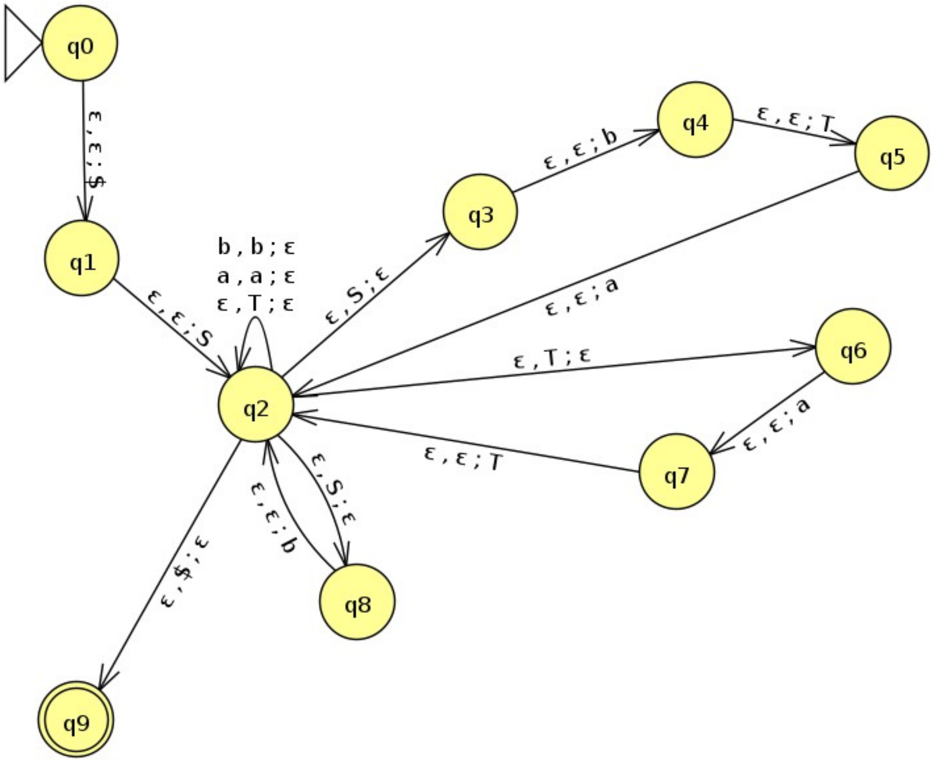
(<=)

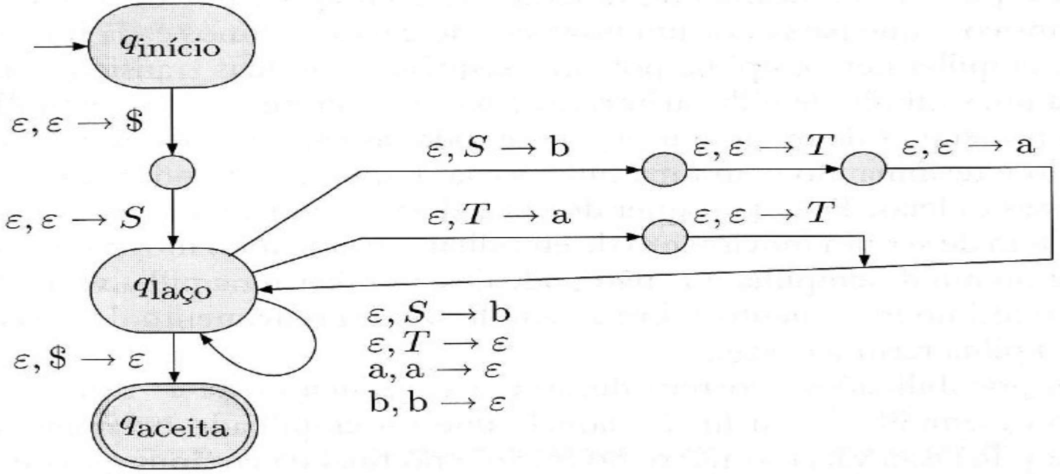
Exemplo

- Transforme o seguinte AP em uma GLC equivalente (numere os estados que não possuem rótulos, não esqueça de preparar o AP de acordo com o slide 33)



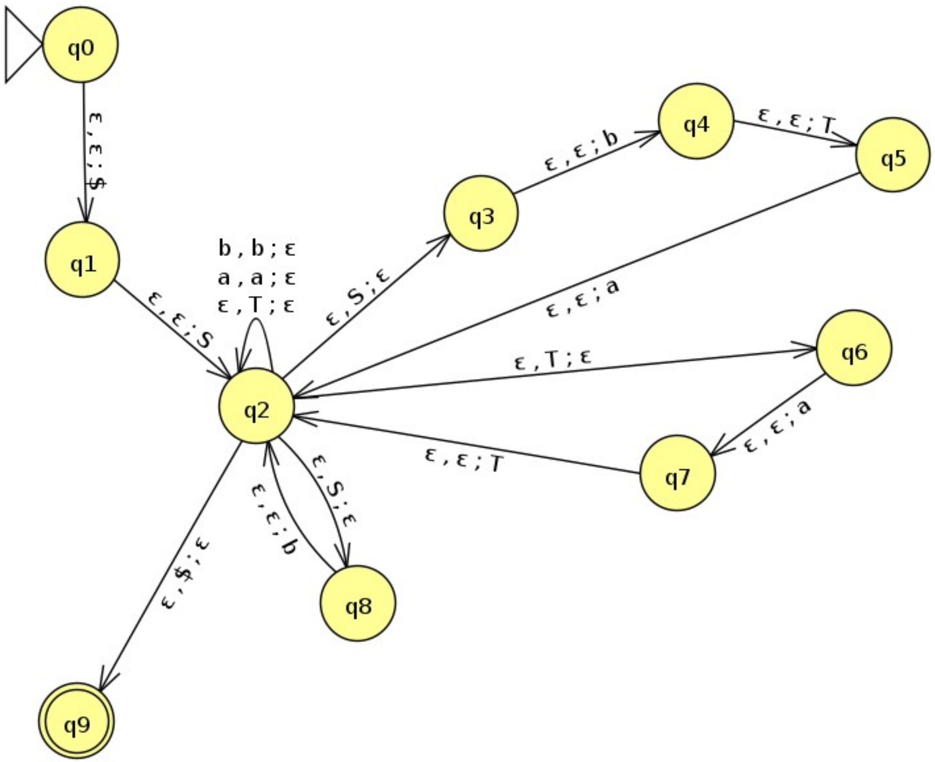
Primeiro passando o APN para o formato assumido (só um estado final, e cada transição só empilha ou desempilha)





Colocando a variável inicial

A variável inicial é $A_{q0,q_{aceita}}$

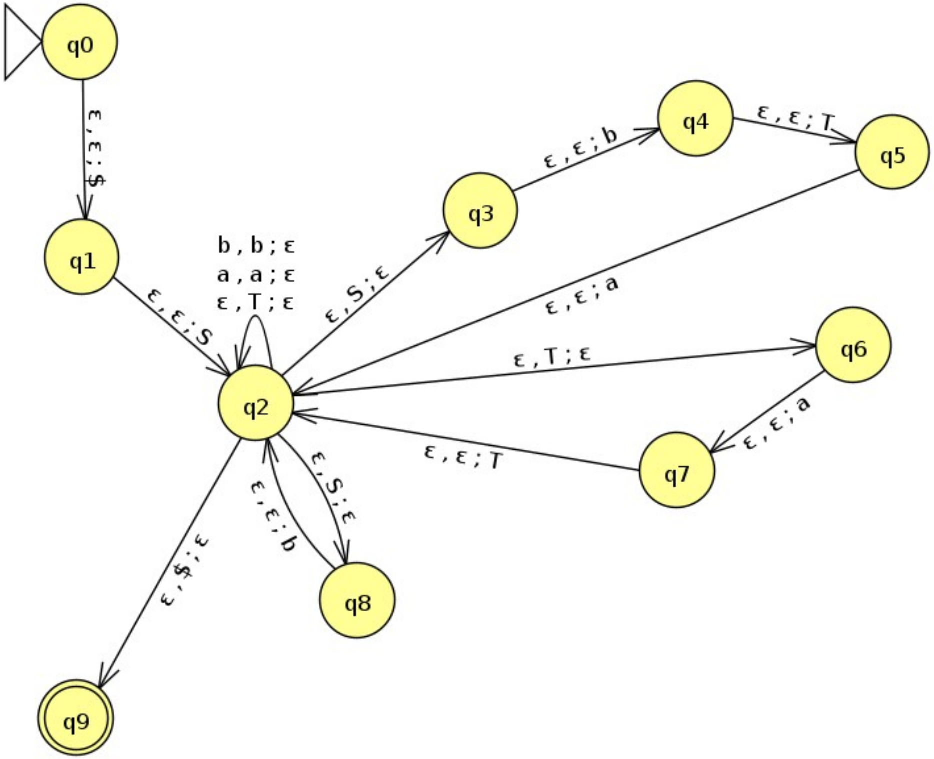


$A_{q0q9} \rightarrow ...$

- 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 $A_{pq} \rightarrow aA_{rs}b$ em G.

O segredo é achar os pares de transições que empilham e desempilham o mesmo símbolo ("t")

$A_{q0q9} \rightarrow \ldots$

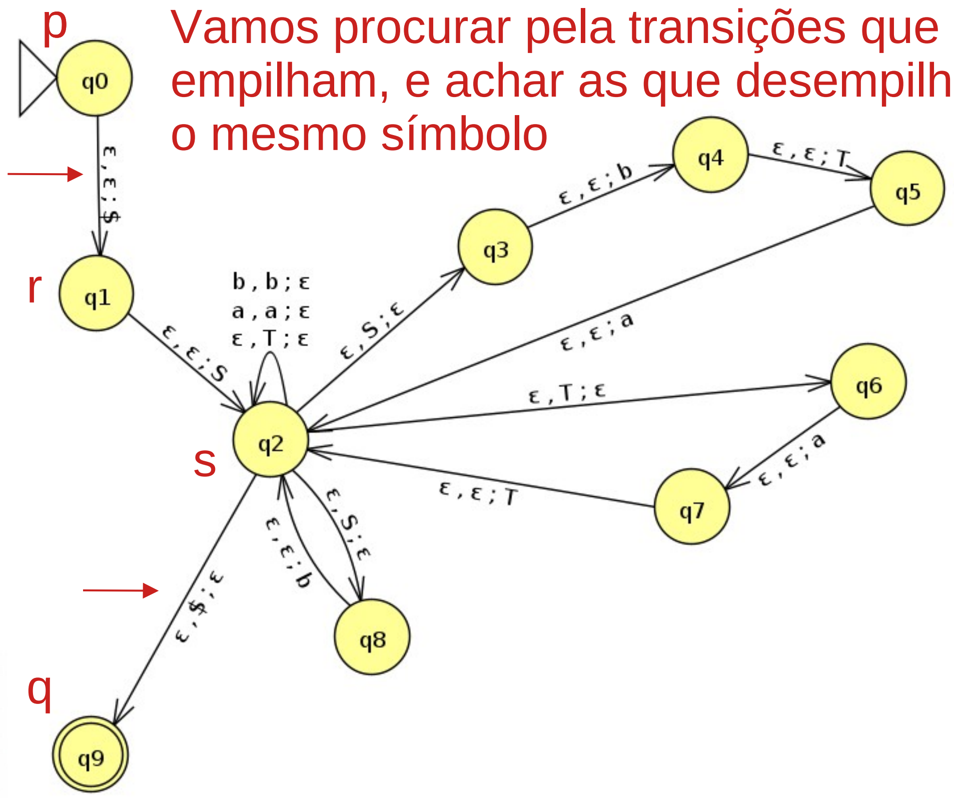


- Para cada $p, q, r, s \in Q$, $t \in \Gamma$, e $a, b \in \Sigma_\varepsilon$, se $\delta(p, a, \varepsilon)$ contém (r, t) e $\delta(s, b, t)$ contém (q, ε), ponha a regra $A_{pq} \rightarrow aA_{rs}b$ em G.

O segredo é achar os pares de transições que empilham e desempilham o mesmo símbolo (“t”)

$A_{q0q9} \rightarrow A_{q1q2}$

Vamos procurar pela transições que empilham, e achar as que desempilham o mesmo símbolo

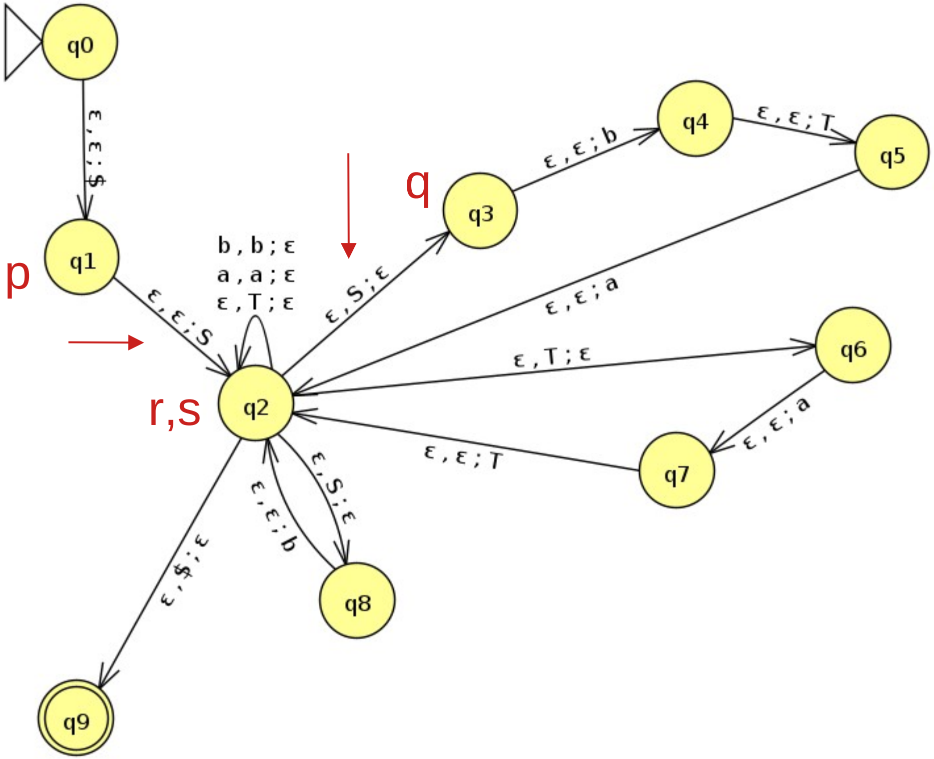


- Para cada $p, q, r, s \in Q$, $t \in \Gamma$, e $a, b \in \Sigma_\varepsilon$, se $\delta(p, a, \varepsilon)$ contém (r, t) e $\delta(s, b, t)$ contém (q, ε), ponha a regra $A_{pq} \longrightarrow aA_{rs}b$ em G.

O segredo é achar os pares de transições que empilham e desempilham o mesmo símbolo (“t”)

$A_{q0q9} \rightarrow A_{q1q2}$

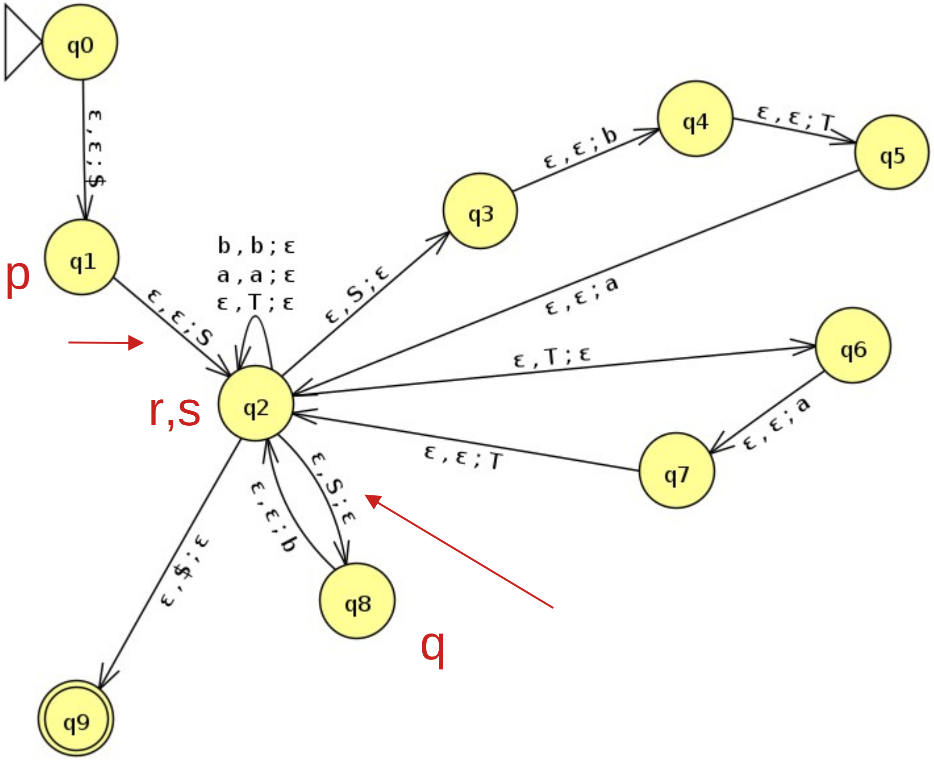
$A_{q1q3} \rightarrow A_{q2q2}$



- 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 $A_{pq} \rightarrow aA_{rs}b$ em G.

O segredo é achar os pares de transições que empilham e desempilham o mesmo símbolo ("t")

$A_{q0q9} \rightarrow A_{q1q2}$
$A_{q1q3} \rightarrow A_{q2q2}$
$A_{q1q8} \rightarrow A_{q2q2}$



- 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 $A_{pq} \longrightarrow aA_{rs}b$ em G.

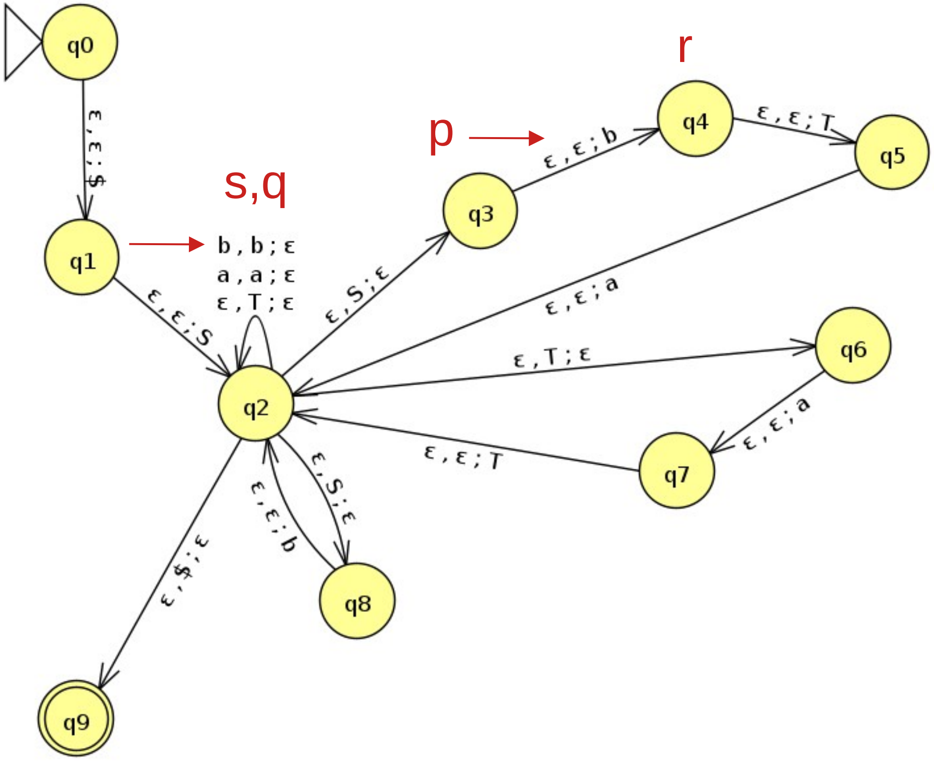
O segredo é achar os pares de transições que empilham e desempilham o mesmo símbolo (“t”)

$A_{q0q9} \rightarrow A_{q1q2}$

$A_{q1q3} \rightarrow A_{q2q2}$

$A_{q1q8} \rightarrow A_{q2q2}$

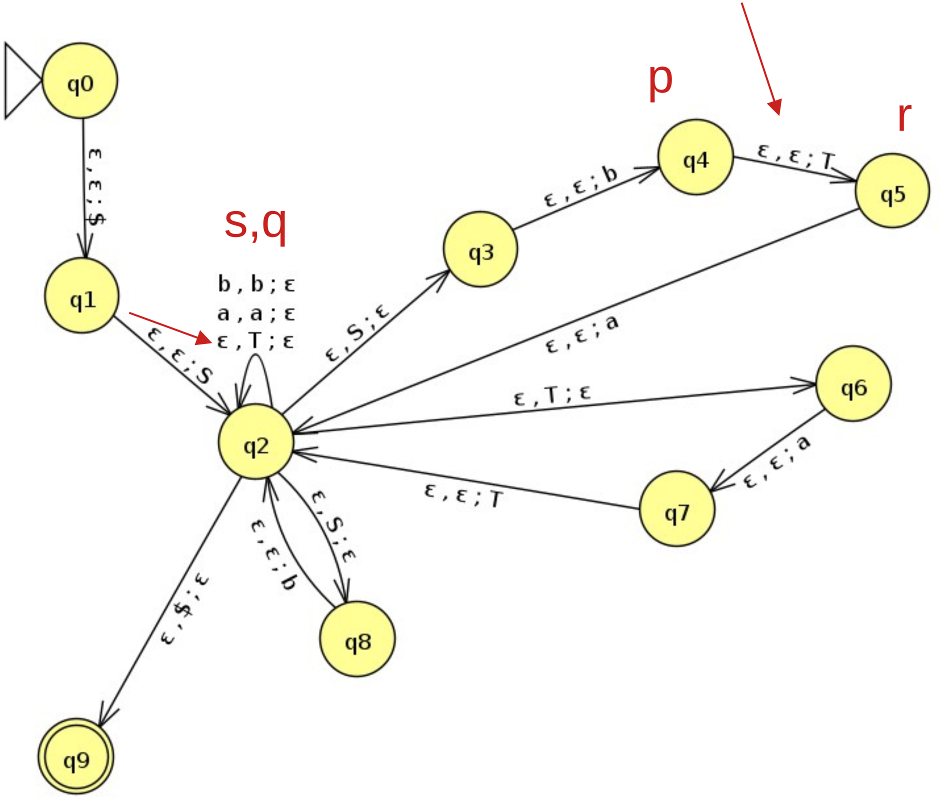
$A_{q3q2} \rightarrow A_{q4q2}b$



- 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.

O segredo é achar os pares de transições que empilham e desempilham o mesmo símbolo ("t")

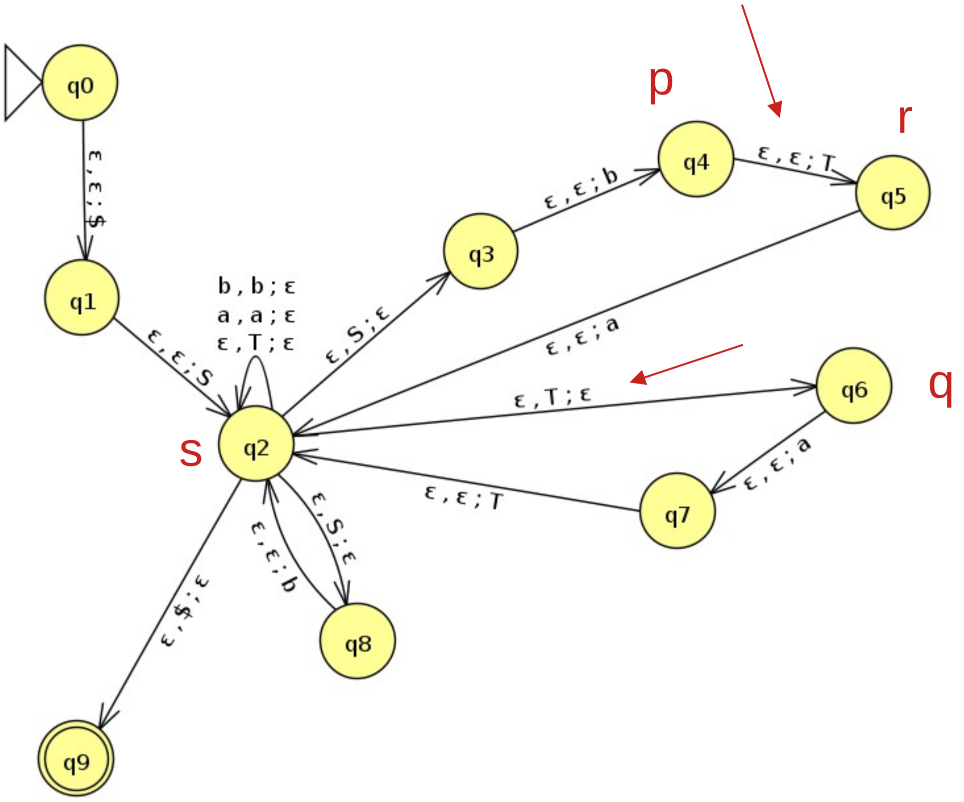
$A_{q0q9} \to A_{q1q2}$
$A_{q1q3} \to A_{q2q2}$
$A_{q1q8} \to A_{q2q2}$
$A_{q3q2} \to A_{q4q2}b$
$A_{q4q2} \to A_{q5q2}$



- 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 $A_{pq} \rightarrow aA_{rs}b$ em G.

O segredo é achar os pares de transições que empilham e desempilham o mesmo símbolo ("t")

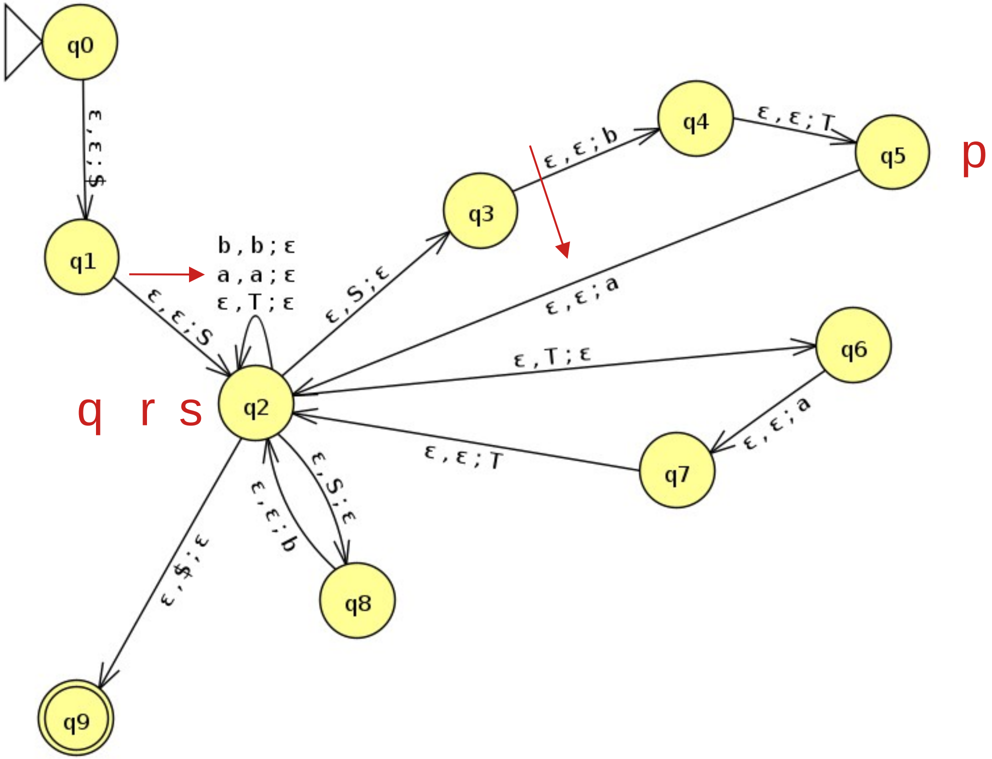
$A_{q0q9} \rightarrow A_{q1q2}$
$A_{q1q3} \rightarrow A_{q2q2}$
$A_{q1q8} \rightarrow A_{q2q2}$
$A_{q3q2} \rightarrow A_{q4q2}b$
$A_{q4q2} \rightarrow A_{q5q2}$
$A_{q4q6} \rightarrow A_{q5q2}$



- 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 $A_{pq} \longrightarrow aA_{rs}b$ em G.

O segredo é achar os pares de transições que empilham e desempilham o mesmo símbolo ("t")

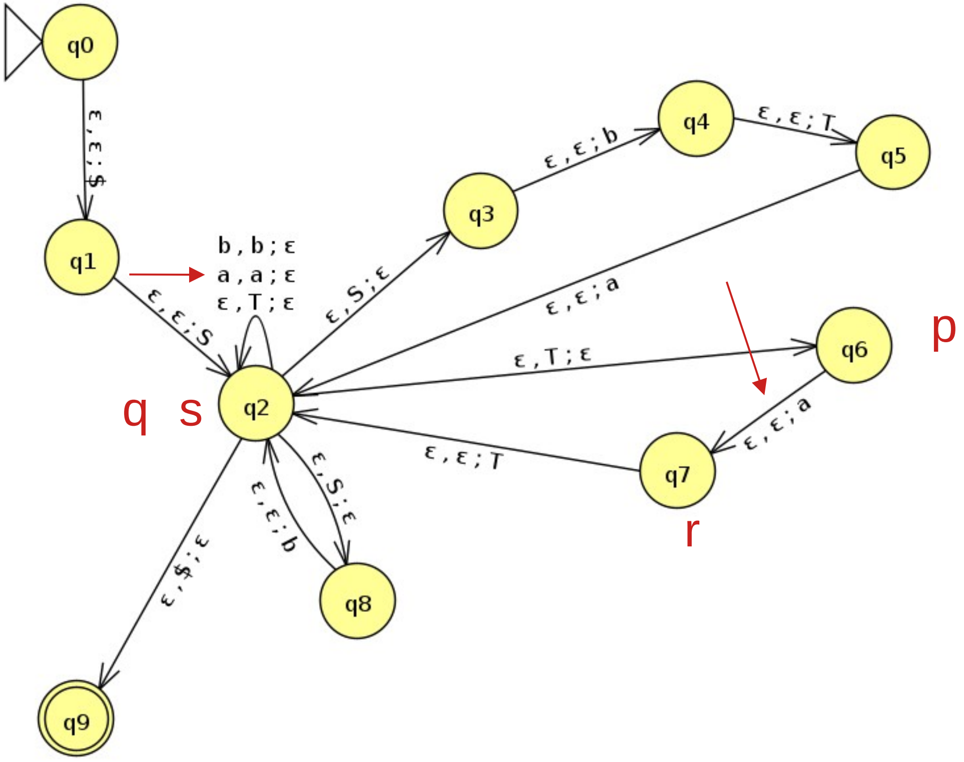
$A_{q0q9} \rightarrow A_{q1q2}$
$A_{q1q3} \rightarrow A_{q2q2}$
$A_{q1q8} \rightarrow A_{q2q2}$
$A_{q3q2} \rightarrow A_{q4q2}b$
$A_{q4q2} \rightarrow A_{q5q2}$
$A_{q4q6} \rightarrow A_{q5q2}$
$A_{q5q2} \rightarrow A_{q2q2}a$



- 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 $A_{pq} \rightarrow aA_{rs}b$ em G.

O segredo é achar os pares de transições que empilham e desempilham o mesmo símbolo (“t”)

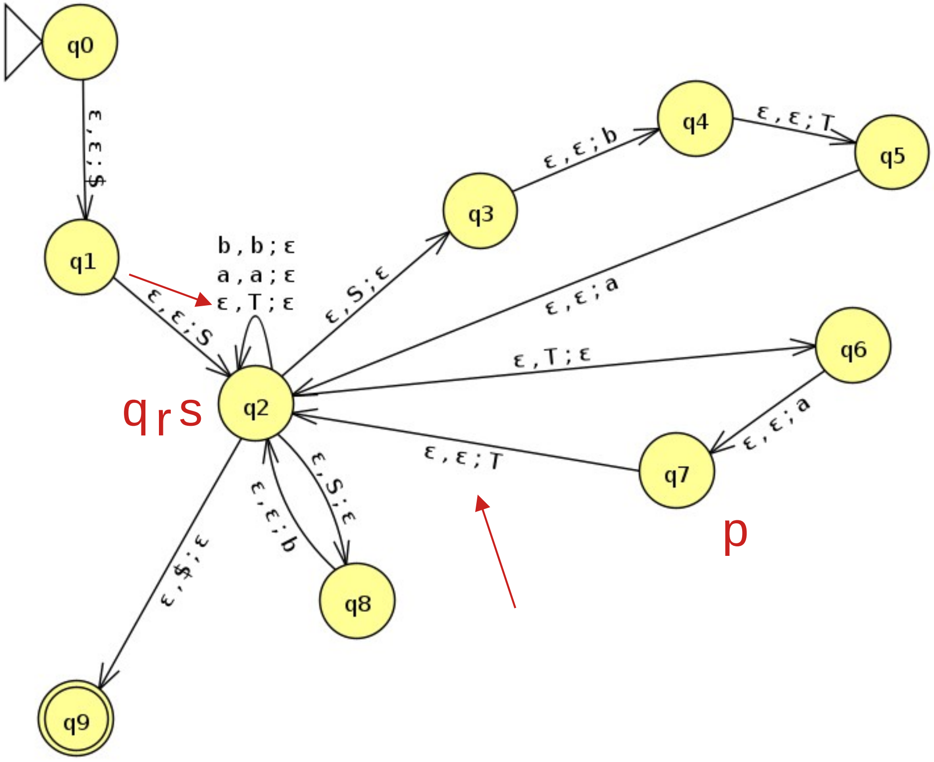
$A_{q0q9} \rightarrow A_{q1q2}$
$A_{q1q3} \rightarrow A_{q2q2}$
$A_{q1q8} \rightarrow A_{q2q2}$
$A_{q3q2} \rightarrow A_{q4q2}b$
$A_{q4q2} \rightarrow A_{q5q2}$
$A_{q4q6} \rightarrow A_{q5q2}$
$A_{q5q2} \rightarrow A_{q2q2}a$
$A_{q5q2} \rightarrow A_{q7q2}a$



- 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 $A_{pq} \longrightarrow aA_{rs}b$ em G.

O segredo é achar os pares de transições que empilham e desempilham o mesmo símbolo (“t”)

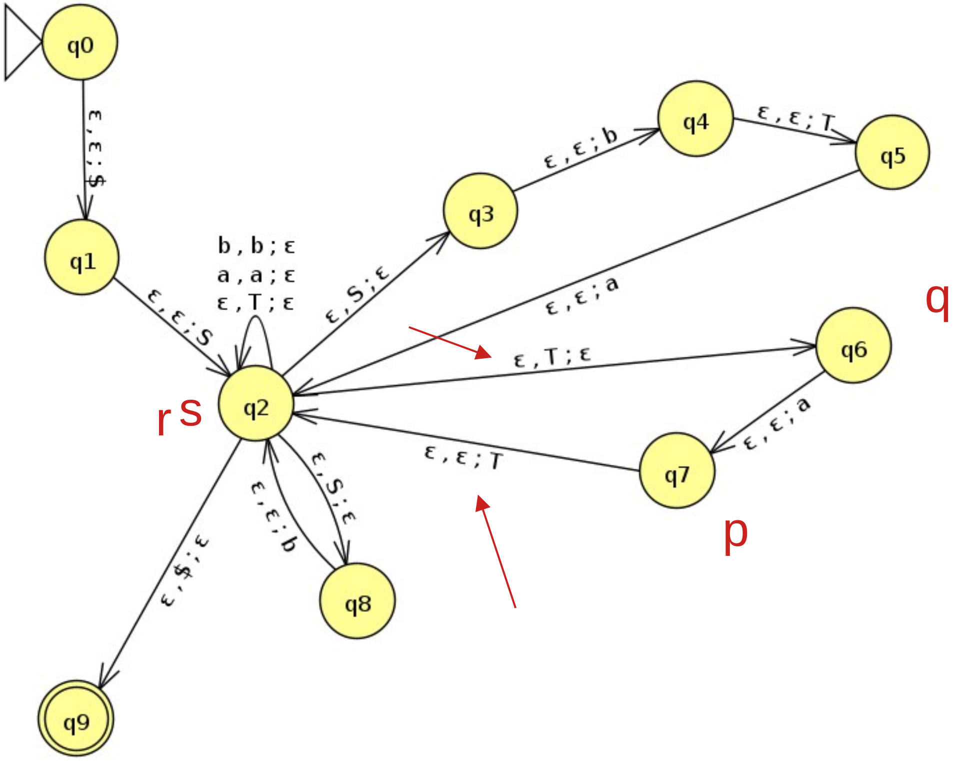
$A_{q0q9} \rightarrow A_{q1q2}$
$A_{q1q3} \rightarrow A_{q2q2}$
$A_{q1q8} \rightarrow A_{q2q2}$
$A_{q3q2} \rightarrow A_{q4q2}b$
$A_{q4q2} \rightarrow A_{q5q2}$
$A_{q4q6} \rightarrow A_{q5q2}$
$A_{q5q2} \rightarrow A_{q2q2}a$
$A_{q5q2} \rightarrow A_{q7q2}a$
$A_{q7q2} \rightarrow A_{q2q2}$



- 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 $A_{pq} \rightarrow aA_{rs}b$ em G.

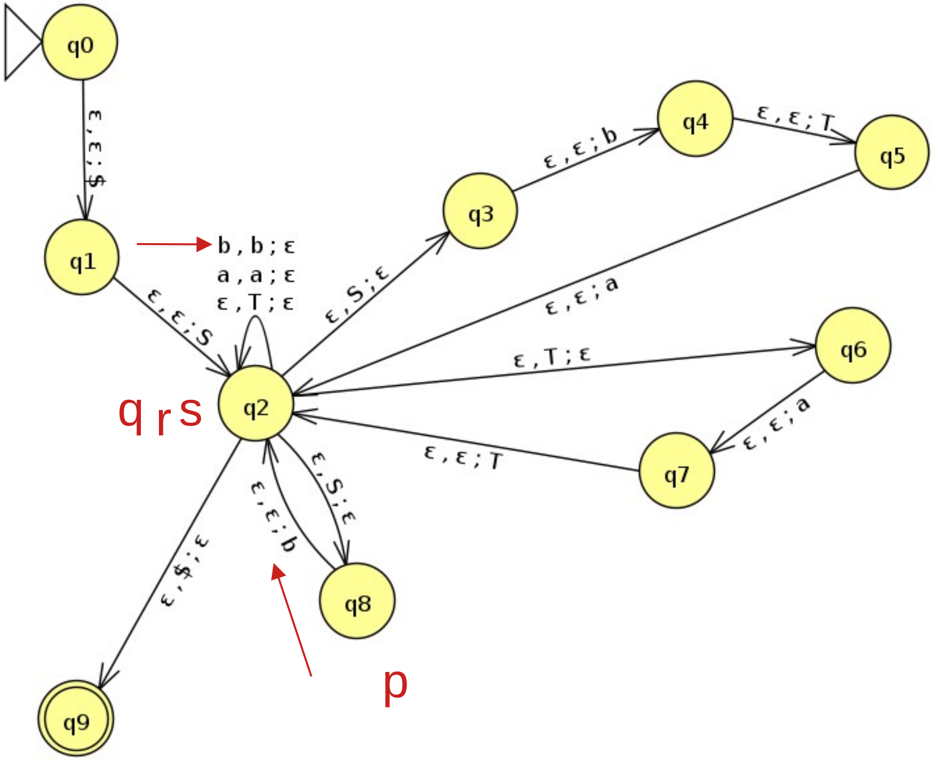
O segredo é achar os pares de transições que empilham e desempilham o mesmo símbolo (“t”)

$A_{q0q9} \rightarrow A_{q1q2}$
$A_{q1q3} \rightarrow A_{q2q2}$
$A_{q1q8} \rightarrow A_{q2q2}$
$A_{q3q2} \rightarrow A_{q4q2}b$
$A_{q4q2} \rightarrow A_{q5q2}$
$A_{q4q6} \rightarrow A_{q5q2}$
$A_{q5q2} \rightarrow A_{q2q2}a$
$A_{q5q2} \rightarrow A_{q7q2}a$
$A_{q7q2} \rightarrow A_{q2q2}$
$A_{q7q6} \rightarrow A_{q2q2}$



- 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.

O segredo é achar os pares de transições que empilham e desempilham o mesmo símbolo (“t”)



$A_{q0q9} \to A_{q1q2}$

$A_{q1q3} \to A_{q2q2}$

$A_{q1q8} \to A_{q2q2}$

$A_{q3q2} \to A_{q4q2}b$

$A_{q4q2} \to A_{q5q2}$

$A_{q4q6} \to A_{q5q2}$

$A_{q5q2} \to A_{q2q2}a$

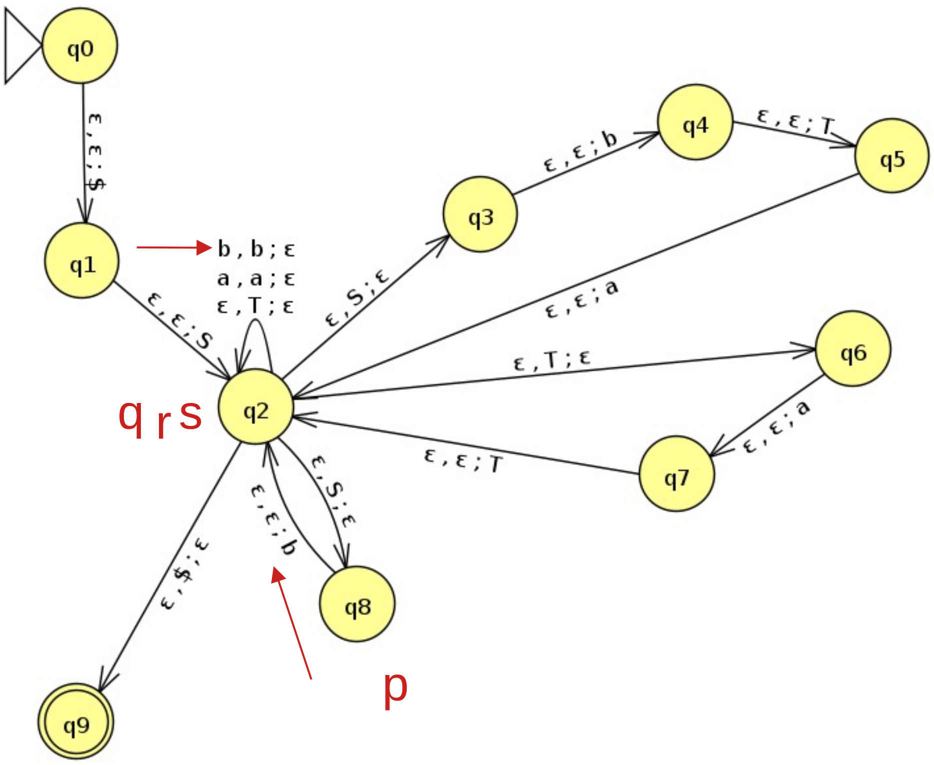
$A_{q5q2} \to A_{q7q2}a$

$A_{q7q2} \to A_{q2q2}$

$A_{q7q6} \to A_{q2q2}$

$A_{q8q2} \to A_{q2q2}b$

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



$A_{q0q9} \rightarrow A_{q1q2}$

$A_{q1q3} \rightarrow A_{q2q2}$

$A_{q1q8} \rightarrow A_{q2q2}$

$A_{q3q2} \rightarrow A_{q4q2}b$

$A_{q4q2} \rightarrow A_{q5q2}$

$A_{q4q6} \rightarrow A_{q5q2}$

$A_{q5q2} \rightarrow A_{q2q2}a$

$A_{q5q2} \rightarrow A_{q7q2}a$

$A_{q7q2} \rightarrow A_{q2q2}$

$A_{q7q6} \rightarrow A_{q2q2}$

$A_{q8q2} \rightarrow A_{q2q2}b$

$A_{q0q0} \rightarrow A_{q0q0}A_{q0q0}$

$A_{q0q0} \rightarrow A_{q0q1}A_{q1q0}$

…

$A_{q0q0} \rightarrow A_{q0q9}A_{q9q0}$

$A_{q0q1} \rightarrow A_{q0q0}A_{q0q1}$

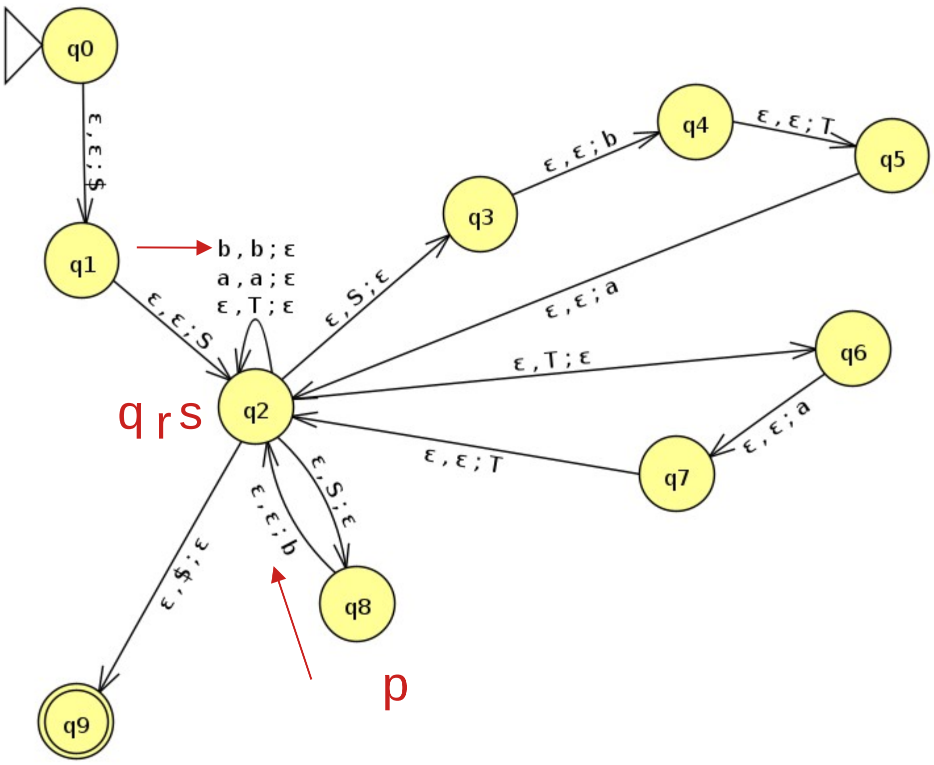
…

$A_{q0q1} \rightarrow A_{q0q9}A_{q9q0}$

…

(10^3 regras)

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



$A_{q0q9} \rightarrow A_{q1q2}$
$A_{q1q3} \rightarrow A_{q2q2}$
$A_{q1q8} \rightarrow A_{q2q2}$
$A_{q3q2} \rightarrow A_{q4q2}b$
$A_{q4q2} \rightarrow A_{q5q2}$
$A_{q4q6} \rightarrow A_{q5q2}$
$A_{q5q2} \rightarrow A_{q2q2}a$
$A_{q5q2} \rightarrow A_{q7q2}a$
$A_{q7q2} \rightarrow A_{q2q2}$
$A_{q7q6} \rightarrow A_{q2q2}$
$A_{q8q2} \rightarrow A_{q2q2}b$

$A_{q0q0} \rightarrow A_{q0q0}A_{q0q0}$
$A_{q0q0} \rightarrow A_{q0q1}A_{q1q0}$
…
$A_{q0q0} \rightarrow A_{q0q9}A_{q9q0}$
$A_{q0q1} \rightarrow A_{q0q0}A_{q0q1}$
…
$A_{q0q1} \rightarrow A_{q0q9}A_{q9q0}$
…
(10^3 regras)

$A_{q0q0} \rightarrow \varepsilon$
…
$A_{q9q9} \rightarrow \varepsilon$
(9 regras)

$$S \to \mathtt{a}T\mathtt{b} \mid \mathtt{b}$$
$$T \to T\mathtt{a} \mid \varepsilon$$

$A_{q0q9} \Rightarrow A_{q1q2} \Rightarrow A_{q1q8}A_{q8q2} \Rightarrow A_{q2q2}A_{q2q2}b \Rightarrow b$

$A_{q0q9} \to A_{q1q2}$
$A_{q1q3} \to A_{q2q2}$
$A_{q1q8} \to A_{q2q2}$
$A_{q3q2} \to A_{q4q2}b$
$A_{q4q2} \to A_{q5q2}$
$A_{q4q6} \to A_{q5q2}$
$A_{q5q2} \to A_{q2q2}a$
$A_{q5q2} \to A_{q7q2}a$
$A_{q7q2} \to A_{q2q2}$
$A_{q7q6} \to A_{q2q2}$
$A_{q8q2} \to A_{q2q2}b$

$A_{q0q0} \to A_{q0q0}A_{q0q0}$
$A_{q0q0} \to A_{q0q1}A_{q1q0}$
…
$A_{q0q0} \to A_{q0q9}A_{q9q0}$
$A_{q0q1} \to A_{q0q0}A_{q0q1}$
…
$A_{q0q1} \to A_{q0q9}A_{q9q0}$
…
(10^3 regras)

$A_{q0q0} \to \varepsilon$
…
$A_{q9q9} \to \varepsilon$
(9 regras)

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.

(<=)

Slides EXTRAS

Prova da corretude do algoritmo de conversão APN → GLC

Profa. Ariane Machado Lima

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 $A_{pp} \to \varepsilon$. 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.

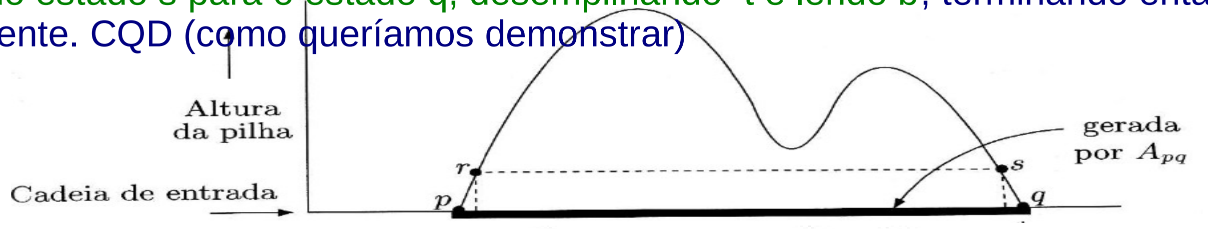
Passo da Indução: Assuma verdadeiro para derivações de comprimento no máximo k, onde $k \geq 1$, e prove verdadeiro para derivações de comprimento $k + 1$. Suponha que $A_{pq} \overset{*}{\Rightarrow} x$ com $k + 1$ passos. O primeiro passo nessa derivação é ou $A_{pq} \Rightarrow aA_{rs}b$ ou $A_{pq}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

Seja y uma cadeia de terminais tal que Ars =>* y → x = ayb

Ars =>* y em k passos, logo (pela hipótese de indução) y leva P do estado r (com pilha vazia) ao estado s (com pilha vazia) (*)

Pelas regras de construção desta GLC, Apq → a Ars b é uma produção de G porque no APN P sendo convertido δ(p,a,ε) contém (r,t) e δ(s,b,t) contém (q,ε), para t є Γ

Logo, P começa no estado p (com uma pilha vazia, porque esta começando a análise agora), lê a, empilha t, vai para o estado r e, devido a (*), quando chega no estado s o símbolo t está no topo da pilha (e a substring y foi lida). Então P vai do estado s para o estado q, desempilhando t e lendo b, terminando então a leitura de x e tornando a pilha vazia novamente. CQD (como queríamos demonstrar)



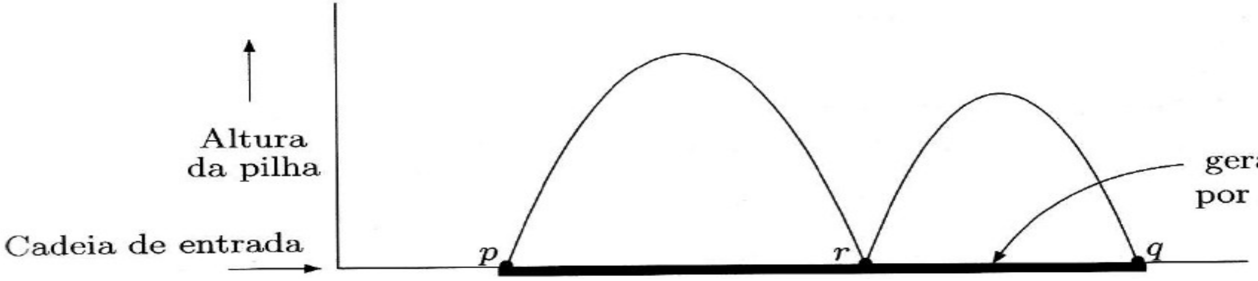
$A_{pq} \rightarrow aA_{rs}b$

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 $A_{pp} \stackrel{*}{\Rightarrow} x$. 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.

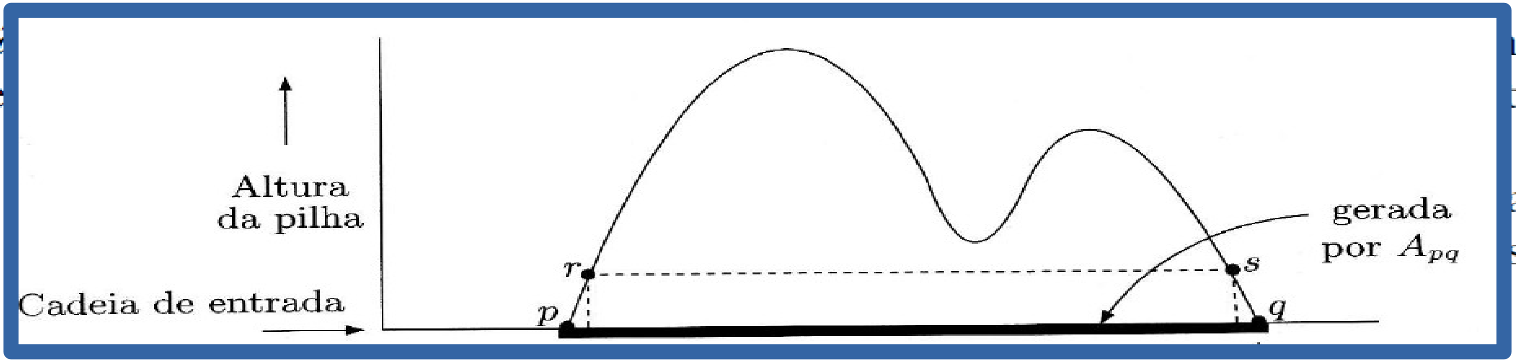
Passo da Indução: Assuma verdadeiro para computações de comprimento no máximo k, onde $k \geq 0$, e prove verdadeiro para computações de comprimento $k + 1$.

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 ....

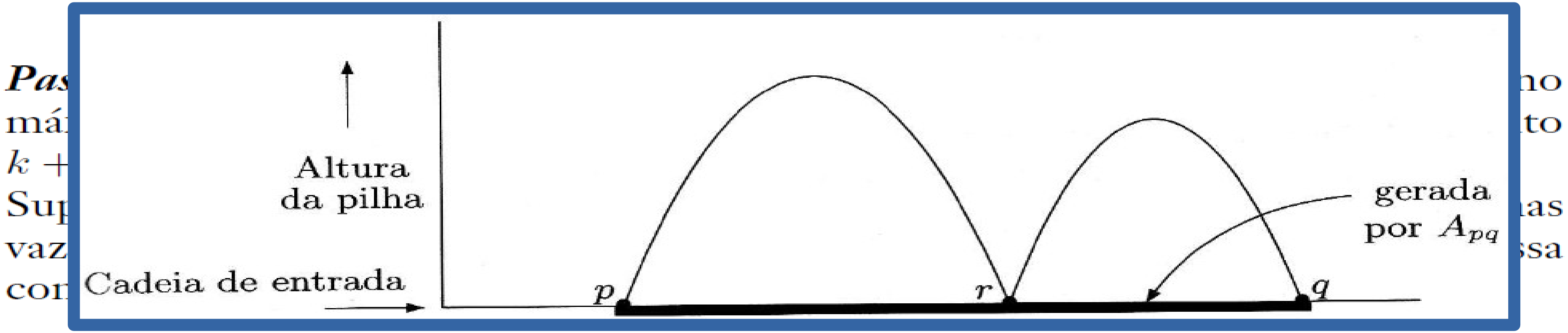
Se x pode levar P de p com pilha vazia para q com pilha vazia, A_{pq} gera x.

Passo da Induç ... o
máximo k, onde ... o
$k + 1$.
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} =>* y e A_{rq} =>* z.
- Como $A_{pq} \rightarrow A_{pr}A_{rq}$ está em G, A_{pq} =>* x

Exercícios

Transforme as gramáticas dos exercícios 2.1, 2.3, 2.13 e 2.14 em um APN equivalente utilizando o método de conversão de GLC em APN do teorema da equivalência entre GLC's e APN's.

ACH2043
INTRODUÇÃO À TEORIA DA COMPUTAÇÃO

Aula 15
Equivalência APN e GLC

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