

Autômato com pilha

Autômato com pilha

- LLC pode ser associado a um formalismo tipo autômato.
- AP é similar ao AF, mas inclui uma pilha como memória auxiliar, mais o não-determinismo.
- A pilha é independente da fita de entrada e não possui limite de tamanho ('infinita').
- O último símbolo que entra é o primeiro que sai.

Autômato com pilha

- A base da pilha é fixa e define seu início.
- O topo é variável e define a posição do último símbolo gravado



Autômato com pilha

- O não-determinismo aumenta o poder computacional, permitindo reconhecer exatamente a Classe das LLC.

$$\{ww^r \mid w \text{ é palavra sobre } \{a,b\}\}$$

- Embora AP seja mais poderoso que o AF, ainda possui restrições e não reconhece linguagens simples como:

$$\{a^r b^r c^r \mid r \geq 0\}$$

Autômato com pilha

- Para reconhecer $\{a^r b^r \mid r \geq 0\}$:
 - Leia os símbolos da entrada.
 - A cada 'a' lido, escreva-o na pilha.
 - Ao receber um 'b', tire um 'a' da pilha.
 - Se ao ler o último símbolo a pilha ficar vazia, aceite a entrada.
 - Se a pilha ficar vazia enquanto ainda houver entradas **ou** se acabarem as entradas e a pilha não estiver vazia **ou** se a entrada voltar a inserir 'a', rejeite a entrada.

Definição de AP

- São duas definições que diferem no critério de parada, mas que são equivalentes:
 - O valor inicial da pilha é vazio e o autômato para aceitando ao atingir um estado final.
 - A pilha contém, inicialmente, um símbolo especial denominado símbolo inicial da pilha. Não existem estados finais e o autômato para aceitando quando a pilha estiver vazia.

Definição de AP

- É composto basicamente por 4 partes:
 - Fita: como no AF.
 - Pilha: memória auxiliar, usada livremente para leitura e gravação.
 - Unidade de controle: reflete o estado corrente da máquina. Possui uma cabeça de fita e uma cabeça de pilha.
 - Programa ou função de transição: comanda a leitura da fita, leitura e gravação da pilha e define o estado da máquina.

Definição de AP

- A pilha é dividida em células, onde cada uma armazena um símbolo do alfabeto auxiliar (pode ser igual ao alfabeto de entrada).
- A leitura ou gravação é sempre feita no topo.
- O tamanho da pilha não é fixo nem finito, depende do tamanho da palavra armazenada.
- O valor inicial é vazio.

Definição de AP

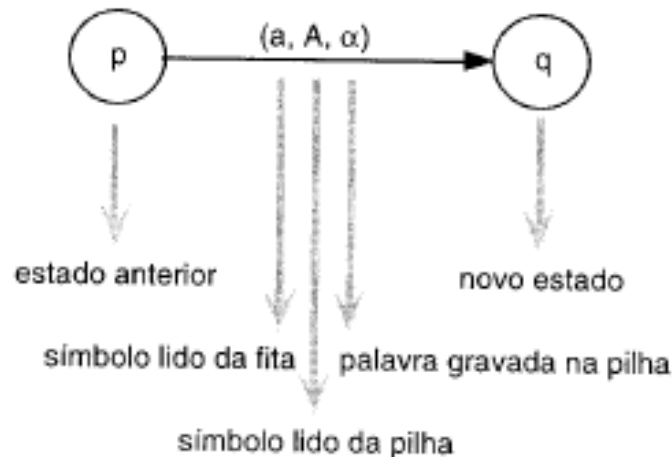
- A unidade de controle possui um número finito e predefinido de estados. Possui também:
 - Cabeça da fita: unidade de leitura que acessa uma célula da fita de cada vez (esquerda → direita).
 - É possível testar se a entrada foi completamente lida.

Definição de AP

- A unidade de controle possui também:
 - Cabeça da pilha: unidade de leitura e gravação que move-se para a esquerda (para cima) quando grava e para a direita (para baixo) quando lê um símbolo.
 - Acessa um símbolo de cada vez no topo.
 - A leitura exclui o símbolo lido.
 - Pode-se testar se a pilha está vazia.
 - Em uma mesma operação de gravação é possível armazenar uma palavra composta por mais de um símbolo.

Definição de AP

- Um APND ou AP é uma 6-upla:
 $M = (\Sigma, Q, \delta, q_0, F, V)$
 V é o alfabeto auxiliar.

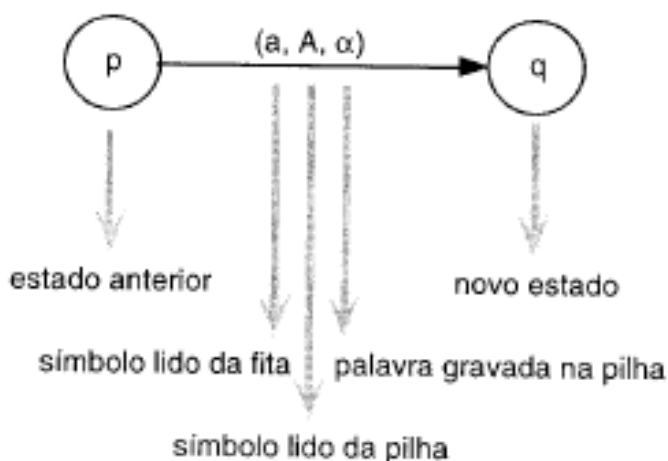


Definição de AP

- Função programa:
 - Pode não ser total; a omissão é representada por “?” e indica o teste de pilha vazia ou toda palavra de entrada lida.
 - O símbolo ϵ indica o movimento vazio da fita ou pilha (o autômato não lê nem move a cabeça).
 - O símbolo ϵ na gravação indica que nenhuma gravação é realizada na pilha (e não move a cabeça).

Definição de AP

- $\delta(p, ?, \varepsilon) = \{(q, \varepsilon)\}$: indica que se a entrada foi completamente lida, não lê da pilha, assume o estado q e não grava a pilha.



Definição de AP

- É possível que um AP nunca atinja uma condição de parada.
 - Fica processando indefinidamente (ciclo ou loop infinito)
 - Empilhar e desempilhar o mesmo símbolo indefinidamente.

Definição de AP

- Um AP pode parar aceitando ou rejeitando a entrada ou ficar em loop:
 - Um dos caminhos alternativos atinge um estado final: o AP para e a palavra é aceita.
 - Todos os caminhos alternativos rejeitam a entrada: o autômato para e a palavra é rejeitada.
 - Pelo menos um caminho alternativo está em loop infinito e os demais rejeitam (ou também estão em loop infinito): o autômato está em loop infinito.

Definição de AP

- Exemplo: $L1 = \{a^r b^r \mid r \geq 0\}$:

$M1 = (\{a, b\}, \{q_0, q_1, q_f\}, \delta, q_0, \{q_f\}, \{B\})$

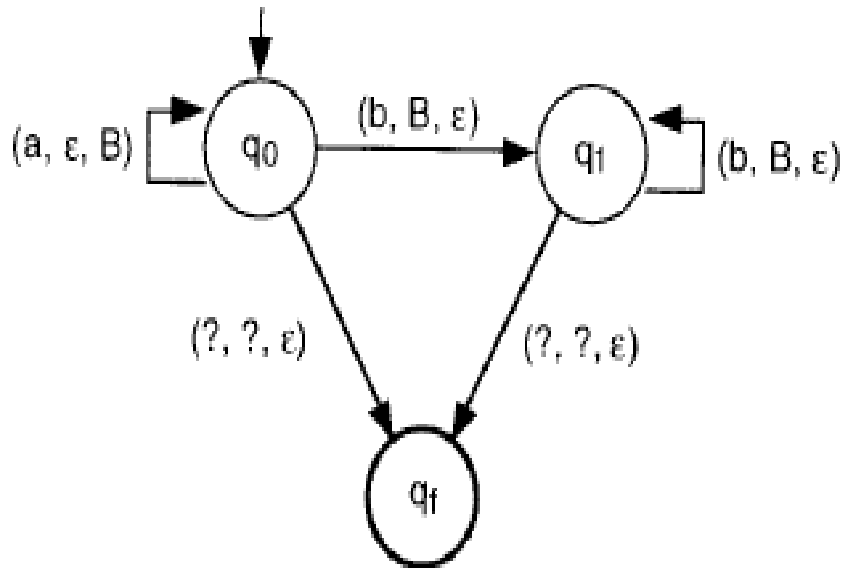
$\delta(q_0, a, \epsilon) = \{(q_0, B)\}$

$\delta(q_0, b, B) = \{(q_1, \epsilon)\}$

$\delta(q_0, ?, ?) = \{(q_f, \epsilon)\}$

$\delta(q_1, b, B) = \{(q_1, \epsilon)\}$

$\delta(q_1, ?, ?) = \{(q_f, \epsilon)\}$



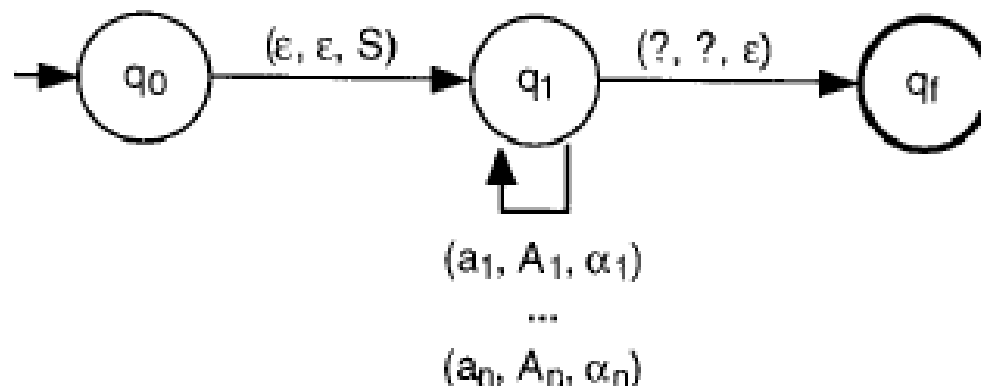
Equivalência com GLC

- Ambos descrevem a classe das linguagens livre de contexto.
 - Construir um reconhecedor para LLC a partir de sua gramática é simples e imediata.
 - Qualquer LLC pode ser reconhecida por um AP com somente um estado lógico (a memorização de informações através de estados não aumenta o poder de AP).

Equivalência com GLC

- Teorema: Uma linguagem é livre de contexto sse algum AP reconhece-a.
- Lema: Se uma linguagem é livre de contexto, então há um AP que reconhece-a.
- Prova: Suponha que a palavra vazia não pertença à L .
 - Construimos um AP a partir da Gramática na FNG, $A \rightarrow a\alpha$ (onde α é uma palavra de variáveis), como derivação mais à esquerda:
 - Lê o símbolo 'a' da fita \rightarrow Lê o símbolo 'A' da pilha \rightarrow empilha a palavra α .

Equivalência com GLC



- Seja $M = (T', \{q_0, q_1, q_f\}, \delta, q_0, \{q_f\}, V')$, onde
$$\delta(q_0, \varepsilon, \varepsilon) = \{(q_1, S)\}$$
$$\delta(q_1, a, A) = \{(q_1, \alpha) \mid A \rightarrow a\alpha \text{ in } P'\}$$
$$\delta(q_1, ?, ?) = \{(q_f, \varepsilon)\}$$

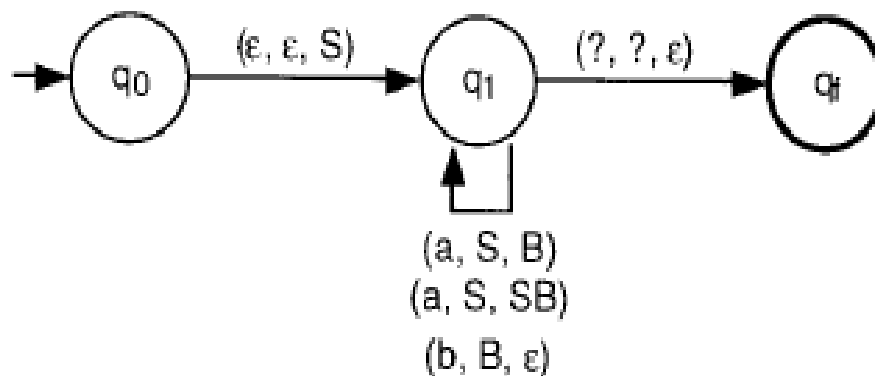
Equivalência com GLC

- Exemplo: $L = \{a^n b^n \mid n \geq 1\}$

$G = (\{S, B\}, \{a, b\}, P, S)$, onde

$P = \{S \rightarrow aB \mid aSB, B \rightarrow b\}$

é reconhecida pelo AP $M = (\{a, b\}, \{q_0, q_1, q_f\}, \delta, q_0, \{q_f\}, \{S, B\})$.



Equivalência com GLC

- Corolário: AP x Número de estados
 - Existe M , um AP com controle de aceitação por estados finais, com somente 3 estados.
 - Existe M , um Ap com controles de aceitação por pilha vazia, com somente um estado.

Equivalência com GLC

- Corolário: Existência de um AP que sempre para.
 - Se L é LLC, então existe M , AP tam que:
 - $ACEITA(M) = L$
 - $REJEITA(M) = \Sigma^* - L$
 - $LOOP(M) = \text{vazio}$.
 - Assim, para qualquer LLC existe um AP que sempre para para qualquer entrada.

Equivalência com GLC

- $AP \rightarrow GLC$
 - Teorema: Se L é aceita por um AP, então L é LLC.
 - A combinação do Teorema acima e os Corolários de AP X Número de estados e do AP que sempre para, comprova que o uso de estados com memória não aumenta o poder de reconhecimento do AP.