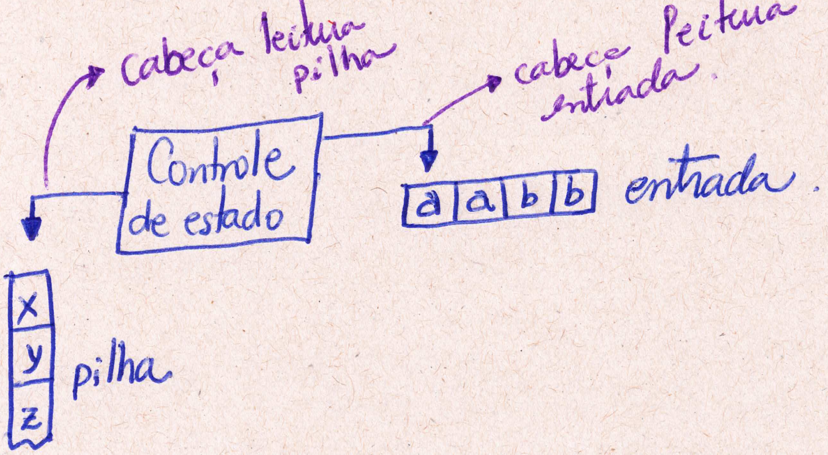
Autômato com Pilha (AP)

Autômatos com Pilha são como AFND, mas têm um componente extra chamado pilha. A pilha fornece memória adicional além da quantidade finita disponível no controle. A pilha permite que o AP reconheça algumas linguagens não-regulares.

Autômatos com Pilha são equivalentes em poder a gramáticas livre do contexto. Essa equivalência nos dá duas opções para provar que um linguagem é livre-do-contexto. Podemos dar uma gramática livre-do-contexto que gera ou um Autômato com pilha que a reconheça.

Certas linguagens são mais facilmente descritas em termos de geradores, enquanto outras são mais facilmente descritas em termos de reconhecedores.



O controle representa os estados e a função de transição, a fita contém a cadeia de entrada. A seta representa a cabeça de entrada, apontando para o próximo símbolo de entrada a ser lido.

Um Autômato com Pilha (AP) pode escrever símbolos sobre a fita e lê-los de volta mais tarde. Escrever um símbolo "empurra para baixo" todos os outros símbolos sobre a pilha. Em qualquer momento, o símbolo no topo da pilha pode ser lido e removido.

Os símbolos remanescentes então voltam a subir.

A escrita na pilha de um símbolo é referenciado como empilhar e remover um símbolo como desempilhar. O acesso à pilha é feito somente no topo.

Definição Formal

O que aumenta da definição de um autômato finito é a pilha. A pilha é um dispositivo contendo símbolos provenientes de algum alfabeto. A máquina pode usar alfabetos diferentes para sua entrada e sua pilha, assim temos um alfabeto de Entrada Σ e um alfabeto de pilha Γ.

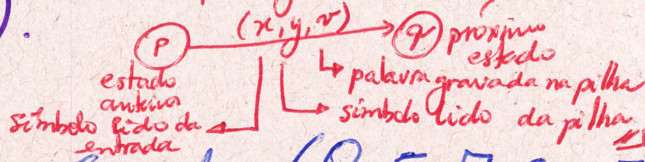
No coração de qualquer definição formal de um autômato está a função de transição, que descreve seu comportamento. Lembremo-nos de que $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$ e $\Gamma_\varepsilon = \Gamma \cup \{\varepsilon\}$. O domínio da função de transição é $Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon$. Portanto, o estado atual, o próximo símbolo lido da entrada e o símbolo no topo da pilha determinam o próximo movimento do AP. Qualquer dos símbolos pode ser ε, levando a máquina a se mover sem ler um símbolo da entrada ou sem ler um símbolo da pilha.

Para o contradomínio da função de transição, precisamos considerar o que permitir o autômato fazer quando ele está em uma situação específica. Ele pode entrar em algum novo estado e possivelmente escrever um símbolo no topo da pilha.

A função de transição δ pode indicar essa ação retornando um membro de Q juntamente com um membro de Γ_ε, ou seja, um membro $Q \times \Gamma_\varepsilon$. Por permitirmos não-determinismo nesse modelo, uma situação pode ter vários próximos movimentos legítimos. A função de transição incorpora não-determinismo da maneira usual, retornando um conjunto de membros de $Q \times \Gamma_\varepsilon$, isto é, um membro de $P(Q \times \Gamma_\varepsilon)$.
Nossa função de transição δ toma a forma:

$$\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow P(Q \times \Gamma_\varepsilon).$$

$\delta(p, x, y) = \{(q, v)\}$



Um ~~a~~ 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. $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow P(Q \times \Gamma_\varepsilon)$ 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.

Um AP $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ computa da seguinte maneira. Ele aceita a entrada w se w puder ser escrita como $w = w_1 w_2 \dots w_m$, onde cada $w_i \in \Sigma_\varepsilon$, e existem uma seqüência de estados $r_0, r_1, \dots, r_m \in Q$ e cadeias $s_0, s_1, \dots, s_m \in \Gamma^*$ 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 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 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.

<u>Exemplo</u>: AP que reconhece a linguagem $\{0^n 1^n | n \geq 0\}$.

Um autômato finito é incapaz de reconhecer essa linguagem, pois ele não pode armazenar números muito grandes em sua memória finita (estados). Um AP é capaz de reconhecer pois pode usar uma pilha para armazenar tal informação.

Suponha que M_1 seja $(Q, \Sigma, \Gamma, \delta, q_1, F)$ onde:

$Q = \{q_1, q_2, q_3, q_4\}$,

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

$\Gamma = \{0, \$\}$,

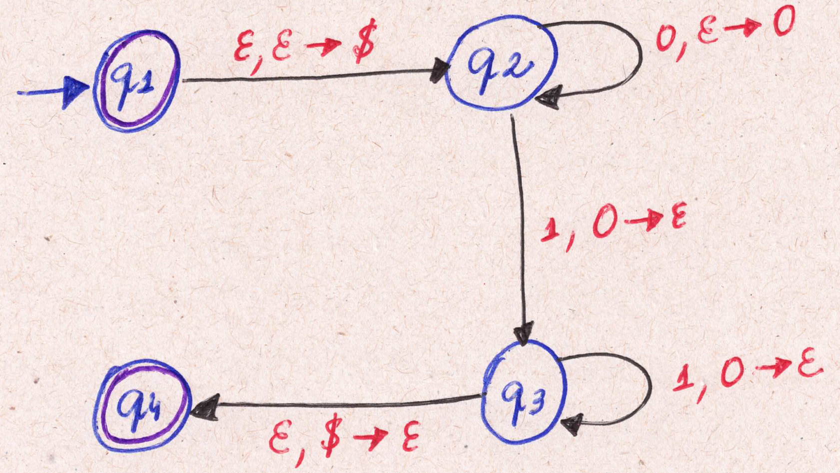
$F = \{q_1, q_4\}$ e

δ dada pela tabela, na qual entradas em branco significam $\emptyset$.

ENTRADA:	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									

Diagrama:

"$\varepsilon,\varepsilon \to \$$ pode ser escrito como $(\varepsilon,\varepsilon,\$)$"



O diagrama mostra como o AP usa a pilha quando vai de estado para estado. Escrevemos "$a, b \to c$" para significar que quando a máquina está lendo a da entrada ela pode substituir o símbolo b no topo da pilha por um c. Qualquer um dos símbolos a, b e c pode ser ε.

Se a é ε, a máquina pode fazer essa transição sem ler qualquer símbolo da entrada. Se b é ε, a máquina pode fazer essa transição sem ler nem desempilhar qualquer símbolo da pilha.
Se c é ε, a máquina não escreve nenhum símbolo na pilha ao fazer essa transição.

A descrição formal de um AP não contém nenhum (6)
mecanismo explícito para permitir ao AP testar por pilha vazia.

Esse AP é capaz de obter o mesmo efeito colocando inicialmente um símbolo especial $ na pilha. Então, se ele em algum momento do futuro vê o $ novamente, ele sabe que a pilha estar efetivamente vazia.

Similarmente, APs não podem testar explicitamente se atingiram o final da cadeia de entrada. Esse AP é capaz o resultado desejado porque o estado de aceitação é atingido somente quando a máquina está no final da entrada.

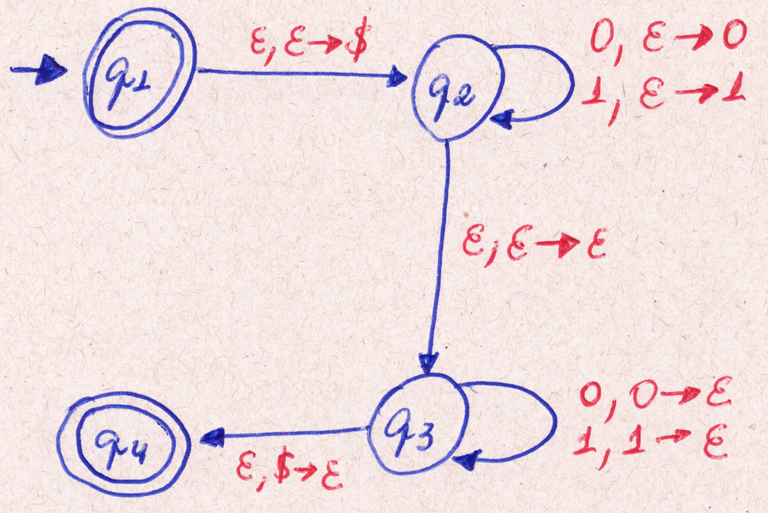
Portanto, de agora em diante, assumimos que APs podem testar pelo final da entrada e sabemos que podemos implementá-los dessa mesma maneira.

<u>Exemplo</u>: M_3 que reconhece a linguagem

$\{ww^R \mid w \in \{0,1\}^*\}$.

Lembremo-nos de que w^R significa w escrita de trás para a frente.

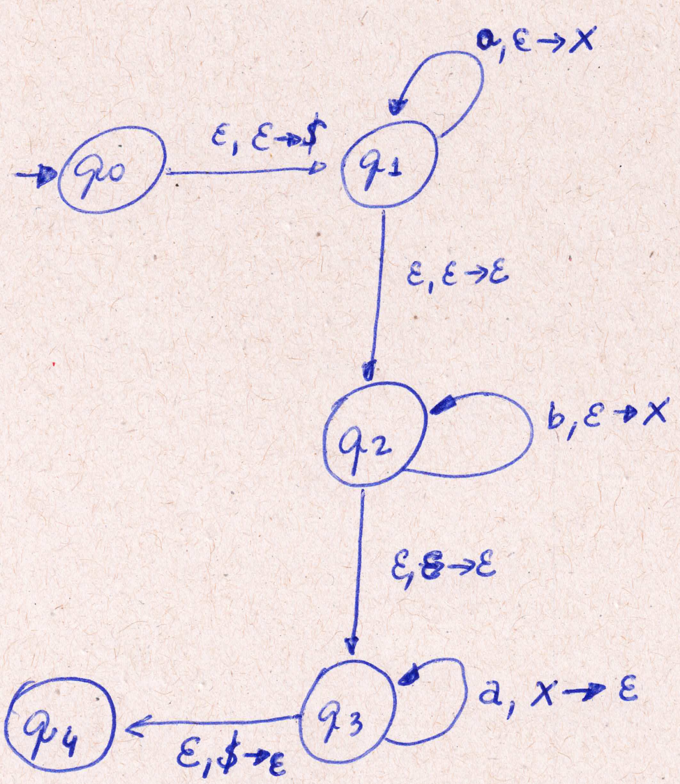
Comece empilhando os símbolos que são lidos. A cada ponto, adivinhe não-deterministicamente se o meio da cadeia foi atingido e, se tiver sido, passe a desempilhar um símbolo para cada símbolo lido, checando para garantir que eles sejam os mesmos. Se eles forem sempre os mesmos e a pilha esvaziar ao mesmo tempo em que a entrada terminar, aceite, caso contrário, rejeite.



Exercício:

Considere a linguagem sobre o alfabeto $\{a, b\}$.

$L_4 = \{ a^n b^m a^{n+m} \mid n \geq 0, m \geq 0 \}$



M_4 empilha um símbolo auxiliar X para cada a ou b em q_1 e q_2. Após q_3 verifica se o número de a no sufixo é igual ao de X na pilha.