

A linguagem PAROP

Agora imagine que as expressões estão completamente parentizadas — (*i.e.*, *cada operação está dentro do seu próprio par de parênteses*)

Por exemplo,

$$\left(\left((1 + 3) * 4 \right) + (2 * 5) \right)$$

E imagine que dessa vez nós removemos apenas os números da expressão — (*e deixamos os parênteses e as operações*)

$$\begin{aligned} (4 + (7 * 5)) &\Rightarrow (+ (*)) \\ ((9 + 6) * (10 + 2)) &\Rightarrow ((+) * (+)) \\ ((((1 + 2) * 3) + 4) * 5) &\Rightarrow ((((+) *) +) *) \end{aligned}$$

Essa é a linguagem PAROP.

E a tarefa agora é construir um autômato de pilha para essa linguagem.

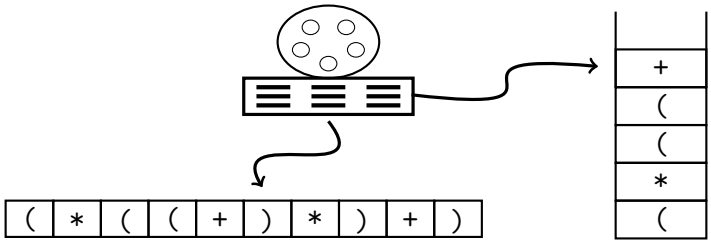
Só que isso não é difícil.

Quer dizer, vamos considerar o seguinte exemplo

$$(* ((+) *) +)$$

Imagine que o autômato vai empilhando tudo o que ele encontra.

E vamos examinar o conteúdo da pilha no momento em que o autômato chega na **fecha-parênteses**



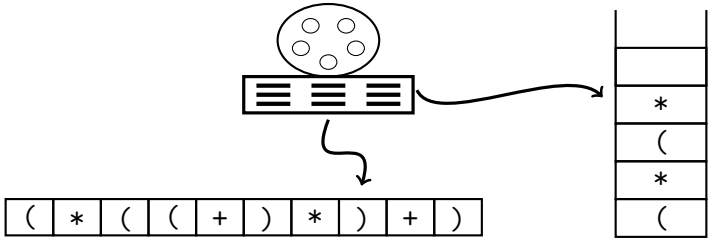
A operação associada a esse **fecha-parênteses** é o +.

E esse + se encontra no topo da pilha.

Então, o autômato pode verificar isso.

E pode remover o + e o **abre-parênteses** da pilha — (*no momento da leitura do fecha-parênteses*)

Daí, quando o autômato encontra o próximo **fecha-parênteses**, a pilha está assim



Mais uma vez, a operação associada a esse **fecha-parênteses** está no topo da pilha.

Então, é só fazer a mesma coisa outra vez.

E continuar fazendo as coisas assim.

O autômato de pilha fica assim

