

A linguagem EXPR

Agora considere as expressões completamente parentizadas outra vez

$$\begin{aligned} & (4 + 7) \\ & ((18 + 5) * 2) \\ & ((5 + 7) * (15 + 2)) \end{aligned}$$

Só que dessa vez nós não vamos remover nada.

Quer dizer, o autômato vai ler tudo.

E no final, ele precisa dizer se a palavra de entrada é uma expressão válida ou não.

Por exemplo, ele precisa rejeitar coisas como

$$(4 + + 7)$$
$$((18 + 5) 2)$$
$$(5 + 7) * (15 + 2)))$$

e por aí vai ...

Só que isso não é difícil.

Quer dizer, nós já temos um autômato que faz isso para as expressões onde os números foram apagados.

Então, agora é só modificar o autômato para que ele leia os números.

E aqui, nós vamos fazer uma pequena esperteza.

Quer dizer, a ideia é que nós não vamos empilhar os números.

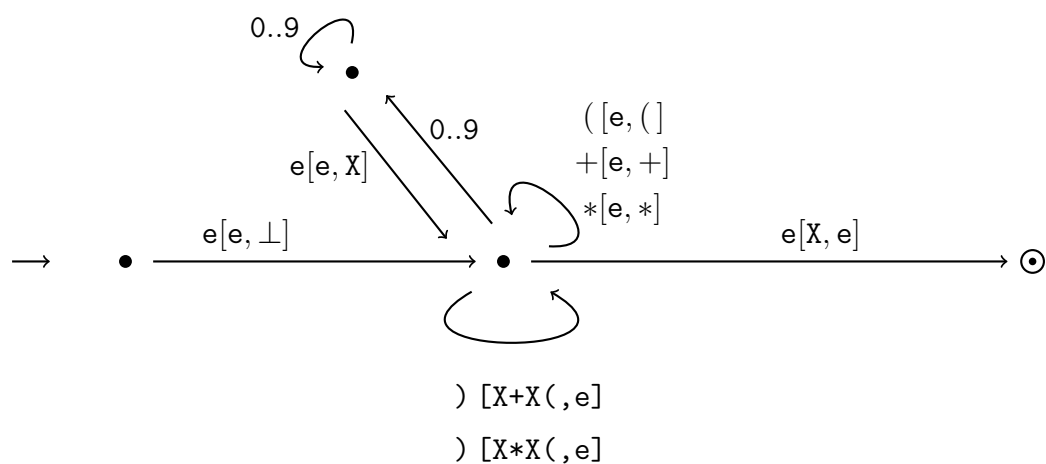
Porque os números podem ter quantidades diferentes de dígitos.

Ao invés disso, a gente vai ter todos os dígitos e deixar um marcador na pilha: a letra X — (*para lembrar mais tarde que havia um número ali*)

E daí mais tarde, quando o autômato ler o **fecha-parênteses**, ele espera encontrar a pilha assim



A coisa fica assim



Exercício: Remova o não-determinismo desse autômato.

Exercício: A linguagem **EXPR2**: expressões aritméticas que não precisam estar completamente parentizadas.