

A linguagem dos primos

Uma máquina que não sabe contar não pode saber nada de matemática.

Então agora, a gente considera a linguagem

Pr: blocos de a's cujo tamanho é um número primo

Por exemplo,

aa, aaa, aaaaa, aaaaaaa, etc

A seguir, nós vamos argumentar que Pr não é uma linguagem regular.

Imagine que o autômato M reconhece a linguagem Pr.

E imagine que M tem *k* estados.

Daí, a gente considera um primo *p* qualquer, maior do que o número *k*.

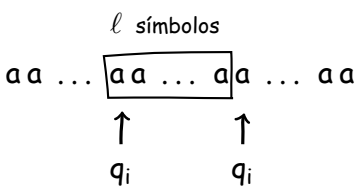
E aplica o nosso truque



Aqui, nós sabemos que

- i. o autômato vai responder SIM — (*porque?*)
- ii. o autômato vai passar duas vezes pelo mesmo estado, durante a leitura do bloco de a's — (*porque p é maior do que k*)

Colocando o loop em foco, a gente vê o seguinte



Só que dessa vez, o bombeamento pode não funcionar.

Porque $p + \ell$ também pode ser um número primo.

E agora?

Bom, agora é a hora para uma esperteza.

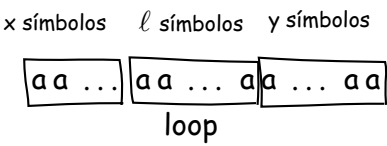
Quer dizer, a esperteza é bombear muitas vezes.

Porque mais cedo ou mais tarde a gente acaba obtendo um número que não é primo.

Quem disse?

Bom, ninguém disse nada — (*isso é só uma intuição ...*)

Então, para entender melhor as coisas a gente quebra a palavra em 3 blocos



E escreve o número *p* assim

$$p \quad = \quad (x + y) + \ell$$

Daí, a gente começa a bombear

$$\begin{aligned} a^{p+\ell} &= (x + y) + \ell + \ell \\ a^{p+2\cdot\ell} &= (x + y) + \ell + \ell + \ell \\ a^{p+3\cdot\ell} &= (x + y) + \ell + \ell + \ell + \ell \end{aligned}$$

E continua bombeando até obter o seguinte

$$a^{p+3\cdot\ell} \quad = \quad (x + y) + \underbrace{\ell + \ell + \dots + \ell}_{x + y}$$

Agora, a gente pode reescrever o número assim

$$(x + y) + \ell \cdot (x + y)$$

e depois assim

$$(x + y) \cdot (1 + \ell)$$

Sim! isso é um número composto.

Logo, a palavra $a^{(x+y)\cdot(1+\ell)}$ não está na linguagem Pr.

Mas, o autômato responde SIM quando recebe essa palavra como entrada — (*porque?*)

Isso é uma contradição.

Logo, a linguagem PR não é regular.