

A linguagem DUP

Imagine que a gente pega uma palavra qualquer e concatena outra cópia dela no final

$$a b a a \Rightarrow a b a a a b a a$$

As palavras construídas dessa maneira formam a linguagem DUP.

DUP não é uma linguagem regular.

Porque?

Ora, porque para reconhecer as palavras de DUP é preciso lembrar a 1a metade durante a leitura da 2a metade.

Só que os autômatos não conseguem lembrar nada (muito grande) — *(eles não conseguem nem contar ...)*

Para provar isso, a gente usa o argumento do bombeamento outra vez.

Aqui mais uma vez, não existe um padrão bem definido para as palavras da linguagem.

Então, a gente concentra a atenção nas palavras da forma

$$\underbrace{a a a \dots a a a}_x b \underbrace{a a a \dots a a a}_x b$$

Agora, imagine que o autômato M reconhece a linguagem DOB.

E imagine que M tem *k* estados.

Daí, a gente aplica o nosso truque

$$\underbrace{a a a \dots a a a}_{k+1} b \underbrace{a a a \dots a a a}_{k+1} b \longrightarrow \boxed{M}$$

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?)*

Colocando o loop em foco, a gente vê o seguinte

$$\begin{array}{c} \ell \text{ símbolos} \\ a a \dots \boxed{a a \dots a} a \dots a a \\ \uparrow \qquad \qquad \uparrow \\ q_i \qquad \qquad q_i \end{array}$$

Isso significa que o autômato também aceita as palavras

$$\begin{array}{c} \text{bombeando } \ell \text{ a's} \\ \Downarrow \\ \underbrace{a a a \dots \boxed{a \dots a} \dots a a a}_{{k+1+\ell}} b \underbrace{a a a \dots a a a}_{{k+1}} b \end{array}$$

Só que, a palavra $a^{k+1+\ell} b a^{k+1} b$ não está na linguagem DUP.

Logo, nós temos uma contradição.

Logo, a linguagem DUP não é regular.