Teoria dos Autômatos e Linguagens Formais

aula 04: O padrão repetitivo

1 Introdução

Na aula de hoje nós vamos ver uma ideia legal.

A ideia é que, no final das contas, as palavras que um autômato aceita sempre possuem um padrão repetitivo — (*e aquelas que ele rejeita também ...*)

Quer dizer, a gente não vai ver essa ideia de maneira completa hoje.

Até porque, não é fácil dizer o que é um *padrão repetitivo* — (*no sentido mais geral da coisa ...*)

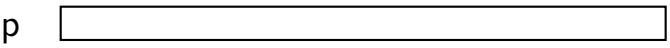
Mas a gente já vai começar a ver como a coisa funciona.

2 Raciocinando sobre propriedades

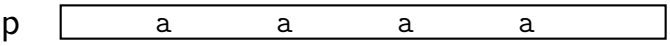
Considere a seguinte propriedade

→ *Todo* `a` *é precedido por ao menos um* `b`

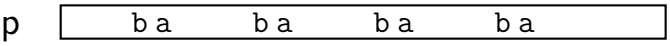
E agora imagine que nós temos uma palavra `p` bem grande que tem essa propriedade



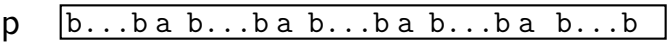
A seguir, imagine que nós estamos vendo todos os `a`'s da palavra `p`



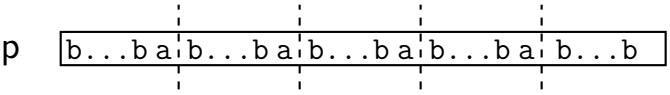
Depois, lembrando da propriedade, nós raciocinamos que cada um desses `a`'s deve ser precedido por ao menos um `b`



Finalmente, todo o resto devem ser `b` — (*porque os* `a`*'s já apareceram todos*)



Nesse ponto, a gente imagina que a palavra é quebrada logo após cada símbolo `a`



E assim a gente vê que essa palavra é uma

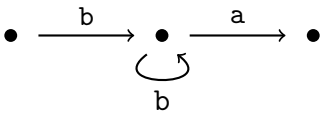
→ concatenação de blocos da forma $\underbrace{\texttt{b}\,..\,\texttt{b}}_{>0}\,\texttt{a}$

seguida opcionalmente por um bloco de `b`'s

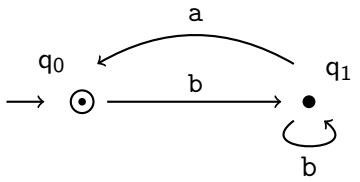
Não é legal?

Sim, mas o mais legal é que agora é a maior moleza construir o autômato que aceita as palavras que tem essa propriedade.

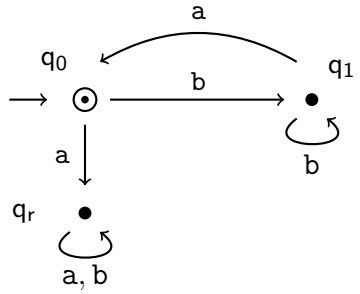
Quer dizer, a gente começa construindo um caminho que lê blocos da forma $\underbrace{\texttt{b}\,..\,\texttt{b}}_{>0}\,\texttt{a}$



Daí, a gente fecha o laço para ler uma concatenação de blocos desse tipo



Finalmente, basta acrescentar o estado de rejeição para completar a construção



A seguir nós vamos ver outros exemplos desse tipo.

Exemplos

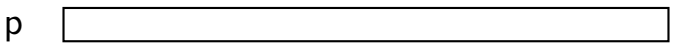
a. Sem b's isolados

Considere a seguinte propriedade

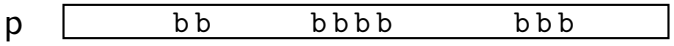
→ *Não existem* b*'s isolados na palavra*

Quer dizer, todo b deve ter um outro b ao seu lado.

Agora imagine uma palavra p bem grande que tem essa propriedade

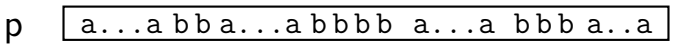


E a seguir imagine que nós estamos vendo todos os símbolos b da palavra p

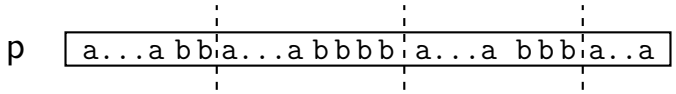


Quer dizer, pela definição da propriedade, os b's sempre aparecem em blocos de tamanho ≥ 2.

Finalmente, como esses são todos os b's, todo o resto devem ser a's



Nesse ponto, a gente imagina que a palavra é quebrada logo após o último b de cada bloco



E assim a gente vê que essa palavra é uma

→ concatenação de blocos da forma $\underbrace{\texttt{a..a}}_{>0}\ \underbrace{\texttt{b..b}}_{\geq 2}$

seguida (opcionalmente) por um bloco de a's

Mas isso ainda não está completamente certo.

Quer dizer, a nossa palavra pode começar com b's.

E daí ela não corresponderia à descrição acima.

Só que, uma pequena esperteza resolve o problema.

Quer dizer, a gente pode reescrever a coisa assim

→ concatenação de blocos da forma $\underbrace{\texttt{a..a}}_{\geq 0}\ \underbrace{\texttt{b..b}}_{\geq 2}$

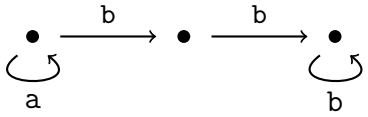
seguida (opcionalmente) por um bloco de a's

e agora a descrição funciona em todos os casos.

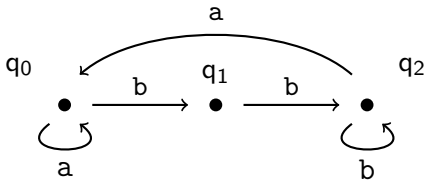
Certo.

O próximo passo é construir o autômato com base nessa descrição.

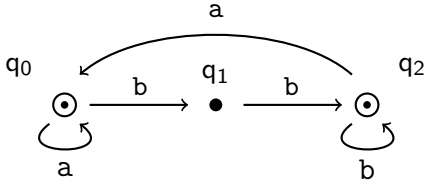
A gente começa construindo um caminho que lê blocos da forma $\underbrace{\texttt{a..a}}_{\geq 0}\ \underbrace{\texttt{b..b}}_{\geq 2}$



Daí a gente fecha o laço para ler uma concatenação de blocos desse tipo

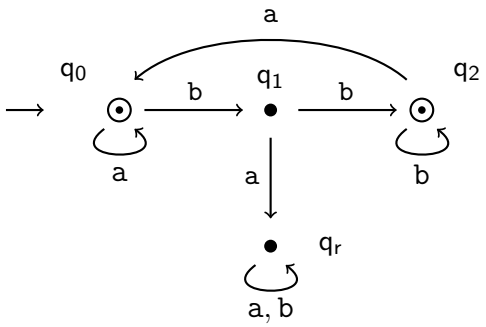


E marca os estados q_0 e q_2 como finais



Pronto.

Agora basta acrescentar o estado de rejeição para concluir a construção



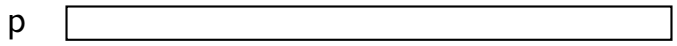
◇

b. Número par de b's

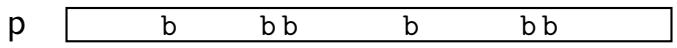
Agora considere a seeguinte propriedade

→ *O número total de* b*'s é par*

Como usual, nós começamos imaginando uma palavra bem grande com essa propriedade



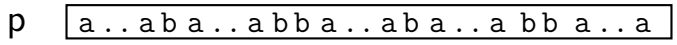
Daí, a gente imagina que está vendo todos os b's da palavra



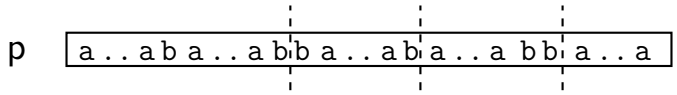
Note que dessa vez não existe qualquer padrão na disposição dos b's dentro da palavra.

A única coisa que nós sabemos é que existe uma quantidade par de b's.

Mas, se esses são todos os b's, então os outros símbolos devem ser todos a's



Nesse ponto, nós imaginamos que a palavra é quebrada imediatamente após cada ocorrência par do símbolo b



E aqui a gente já pode ver que isso é uma

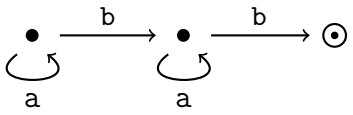
→ concatenação de blocos da forma

$$\underbrace{\texttt{a}\,..\,\texttt{a}\,\texttt{b}}_{\geq 0}\ \underbrace{\texttt{a}\,..\,\texttt{a}\,\texttt{b}}_{\geq 0}$$

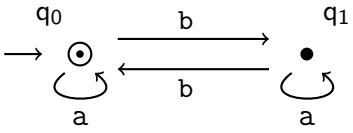
seguida (opcionalmente) por um bloco de a's

Agora que a gente viu isso, é moleza construir o autômato.

Quer dizer, a gente começa com um caminho que lê blocos da forma $\underbrace{\texttt{a}\,..\,\texttt{a}}_{\geq 0}\ \texttt{b}\ \underbrace{\texttt{a}\,..\,\texttt{a}}_{\geq 0}\ \texttt{b}$



Daí a gente fecha o laço para ler uma concatenação de blocos desse tipo



E acabou.

Quer dizer, dessa vez nós não precisamos de um estado de rejeição.

E nós também não precisamos fazer mais nada para ler o possível último bloco de a's.

◇

c. Paridade de blocos

Agora considere a seguinte propriedade

→ *Todo bloco par de* a*'s é seguido por um bloco par de* b*'s*

Bom, dessa vez o padrão repetitivo já está claro.

Quer dizer, nós vamos ter blocos da forma

$$\underbrace{\texttt{a}\,..\,\texttt{a}}_{\text{par}}\ \underbrace{\texttt{b}\,..\,\texttt{b}}_{\text{par}}$$

E quando o bloco de a's tem tamanho ímpar, a seguir pode vir qualquer coisa

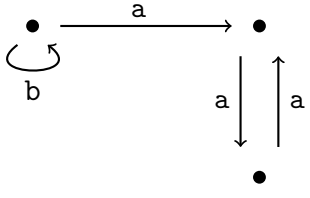
$$\underbrace{\texttt{a}\,..\,\texttt{a}}_{\text{ímpar}}\ \underbrace{\texttt{b}\,..\,\texttt{b}}_{\text{q.q.c.}}$$

Portanto, a descrição da propriedade fica assim

→ concatenação de blocos da forma $\underbrace{\texttt{a}\,..\,\texttt{a}}_{\text{par}}\ \underbrace{\texttt{b}\,..\,\texttt{b}}_{\text{par}}$ ou $\underbrace{\texttt{a}\,..\,\texttt{a}}_{\text{ímpar}}\ \underbrace{\texttt{b}\,..\,\texttt{b}}_{\text{q.q.c.}}$

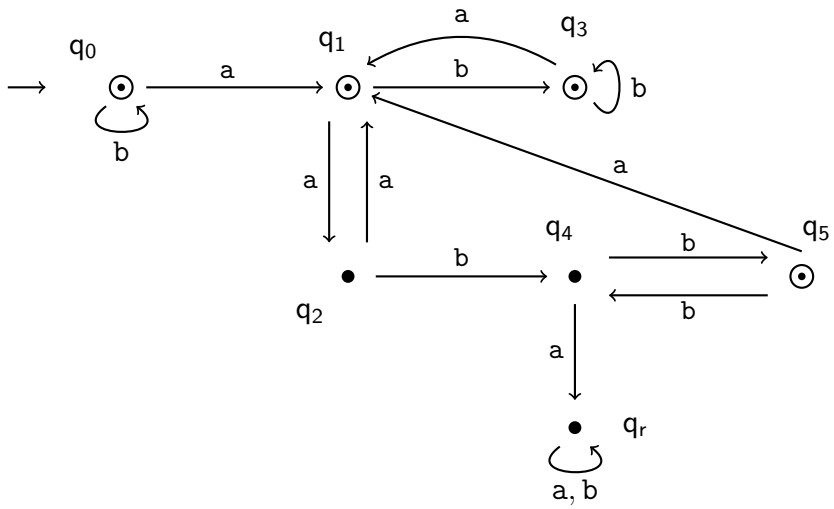
precedida (oppcionalmente) por um bloco de b's

Para construir o autômato, nós começamos com a parte que testa a paridade do bloco de a's



Daí, no caso par a gente lê um bloco par de b's.

E no caso ímpar a gente lê qualquer coisa



◇

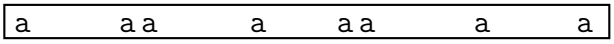
d. O primeiro e o último

Agora considere a seguinte propriedade

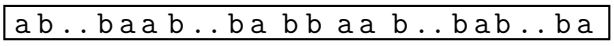
→ *O primeiro e o último símbolos da palavra são iguais*

Aqui nós vamos ter dois casos

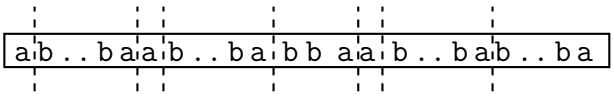
Para analisar o primeiro caso, nós visualizamos todos os a's



Daí se esses são todos os a's, o resto devem ser b's



Nesse ponto, nós imaginamos que a palavra é quebrada logo após cada símbolo a



O outro caso, claro, é análogo.

E assim a gente vê que o padrão repetitivo é

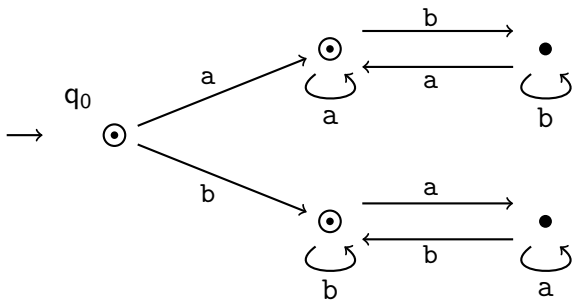
→ um a

seguido (opcionalmente) por uma concatenação de blocos da forma $\underbrace{\text{b .. b a}}_{\geq 0}$

ou um b

seguido (opcionalmente) por uma concatenação de blocos da forma $\underbrace{\text{a .. a b}}_{\geq 0}$

Uma vez que a gente tem o padrão repetitivo, é bem fácil construir o autômato



◇

e. Sem 3 consecutivos iguais

Agora considere a seguinte propriedade

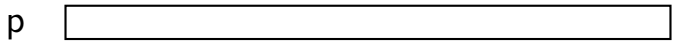
→ *Não existem 3 símbolos consecutivos iguais*

A novidade desse exemplo é que a propriedade não está diretamente relacionada aos símbolos `a` ou `b`.

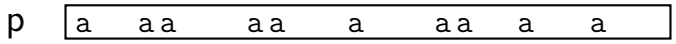
Quer dizer, não importa qual é o símbolo, ele não pode aparecer mais do que duas vezes consecutivas.

Apesar disso, nós podemos visualizar primeiro os `a`'s, e depois os `b`'s.

Imagine uma palavra bem grande que tem essa propriedade

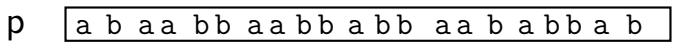


Agora imagine que você está vendo todos os `a`'s da palavra `p`



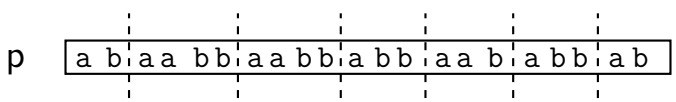
Quer dizer, de acordo com a definição da propriedade, os `a`'s aparecem em bloquinhos de tamanho 1 ou 2.

Daí, se esses são todos os `a`'s, então os outros símbolos devem ser todos `b`'s



E mais uma vez, de acordo com a definição da propriedade, os `b`'s aparecem em bloquinhos de tamanho 1 ou 2.

Nesse ponto, a gente imagina que a palavra é quebrada logo após cada bloquinho de `b`'s



Aqui a gente vê que o bloco que se repete tem a forma

$$\underbrace{a..a}_{1 \text{ ou } 2}\ \underbrace{b..b}_{1 \text{ ou } 2}$$

Mas, antes de tudo pode aparecer um bloquinho de b's.

E depois de tudo pode aparecer um bloquinho de a's.

Portanto, nós descrevemos a propriedade assim

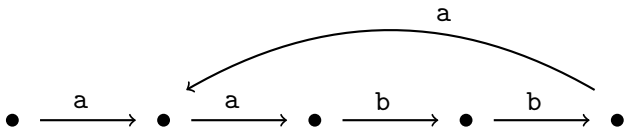
→ concatenação de blocos da forma $\underbrace{a\,..\,a}_{1\text{ ou }2}\ \underbrace{b\,..\,b}_{1\text{ ou }2}$

precedida (oppcionalmente) por 1 ou 2 a's

e sucedida (opcionalmente) por 1 ou 2 b's

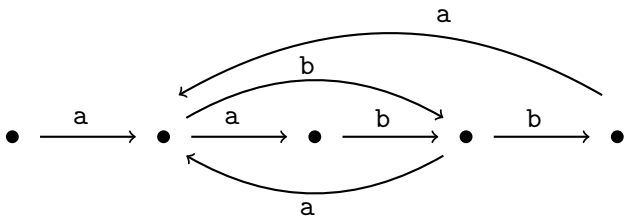
Para construir o autômato, nós começamos com um caminho que lê o bloco a a b b

Daí, a gente fecha o laço para ler uma concatenação de blocos desse tipo



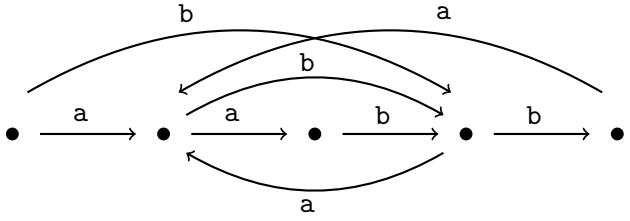
Mas, logo após o primeiro a já pode vir um b.

E logo após o primeiro b já pode vir um a



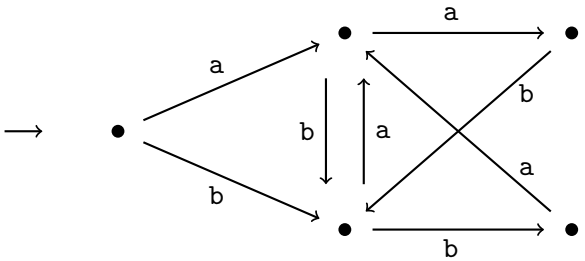
Esse diagrama já leva em conta que a coisa pode terminar com a's.

Mas ainda é preciso considerar o caso em que a palavra começa com b's.

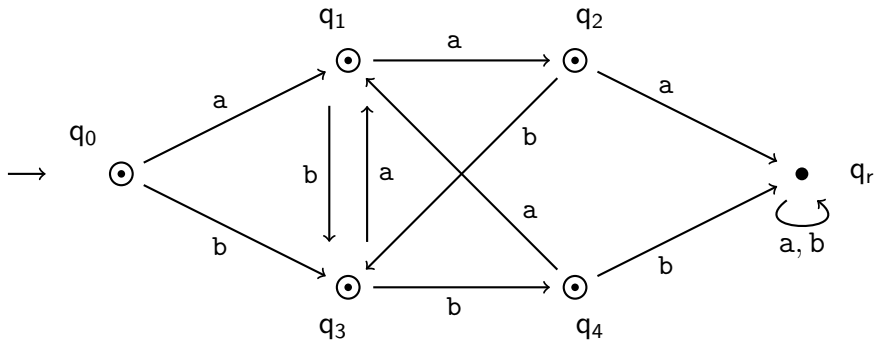


O diagrama parece complicado.

Mas a gente pode reorganizar ele assim



Finalmente, a gente conclui a construção adicionando o estado de rejeição e marcando os estados finais



◇

f. Propriedade dupla

Agora considere a seguinte propriedade

→ *Todo* `b` *é precedido por ao menos 3* `a`*'s*

e

a quantidade total de `a`*'s é par*

Bom, ignorando por um momento a segunda parte da propriedade, a gente obtém a seguinte descrição

→ concatenação de blocos da forma $\underbrace{\texttt{a}\,..\,\texttt{a}}_{\geq 3}\ \texttt{b}$

seguida (opcionalmente) por um bloco de a's

Certo.

Daí a gente observa que se aparece um bloco ímpar de `a`'s, mais adiante deve aparecer outro bloco ímpar de `a`'s, para que a quantidade total seja par.

Além disso, entre os dois blocos ímpares de `a`'s pode aparecer qualquer quantidade de blocos pares de `a`'s.

Daí que, o padrão repetitivo fica assim

→ concatenação de blocos da forma

$$\underbrace{\texttt{a}\,..\,\texttt{a}}_{\geq 3,\ \text{par}}\ \texttt{b} \qquad \text{ou} \qquad \underbrace{\texttt{a}\,..\,\texttt{a}}_{\geq 3,\ \text{ímpar}}\ \texttt{b}\ \Big(\ \text{concatenação de}\ \underbrace{\texttt{a}\,..\,\texttt{a}}_{\geq 3,\ \text{par}}\ \texttt{b}\ \Big)\ \underbrace{\texttt{a}\,..\,\texttt{a}}_{\geq 3,\ \text{ímpar}}\ \texttt{b}$$

seguida (opcionalmente) por um bloco par de a's

$$\text{ou} \quad \underbrace{\texttt{a}\,..\,\texttt{a}}_{\geq 3,\ \text{ímpar}}\ \texttt{b}\ \Big(\ \text{concatenação de}\ \underbrace{\texttt{a}\,..\,\texttt{a}}_{\geq 3,\ \text{par}}\ \texttt{b}\ \Big)\ \underbrace{\texttt{a}\,..\,\texttt{a}}_{\text{ímpar}}$$

(A construção do autômato desse exemplo ficará como exercício)

◇

g. **Enrolada**

Agora considere a seguinte propriedade

→ *Todo* `b` *é precedido ou sucedido por ao menos 3* `a`*'s*

(. . .) ◇