

# Algoritmos e Estruturas de Dados III

Aula 10.3 - Casamento de Padrões - BM

Prof. Felipe Soares  
2022



PUC Minas

# BM - Boyer Moore

- Criado por Robert S. Boyer e J. Strother Moore em 1977
- Comparações de caracteres são feitas da direita para a esquerda
- São feitos dois testes a cada passo:
  - Deslocamento por caractere ruim
  - Deslocamento por sufixo bom

obs: o deslocamento final será o maior dos dois

# BM - Boyer Moore

## Características

- Há fase de pré-processamento – complexidade  $O(m + \sigma)$  em que  $\sigma$  representa o tamanho do alfabeto
- Complexidade é  $O(m + n)$ ,  $m$  = tam. padrão,  $n$  = tam. texto
- Consegue obter uma performance de até –  $O(n / m)$

# BM – Heurísticas de Deslocamento

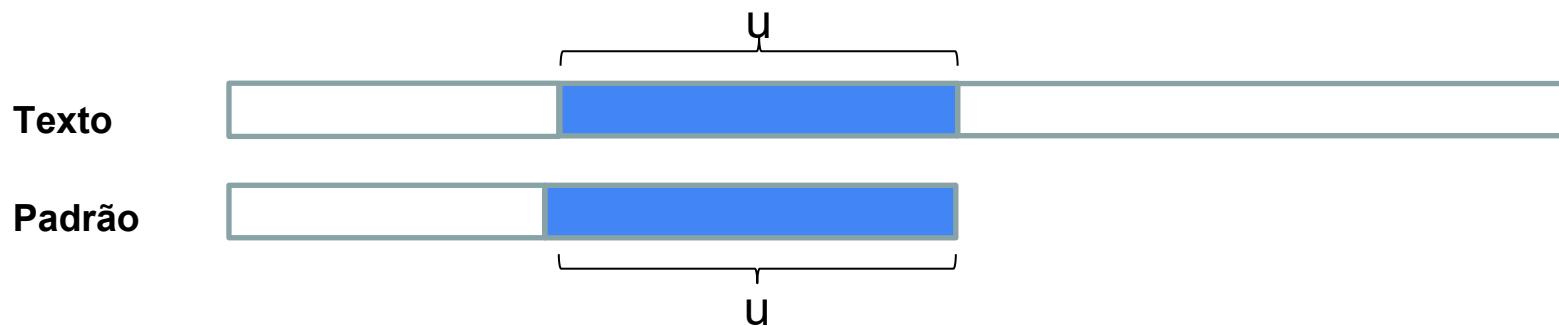
*Bad Character Shift (Deslocamento por Caracter Ruim)*

**Texto**

**Padrão**

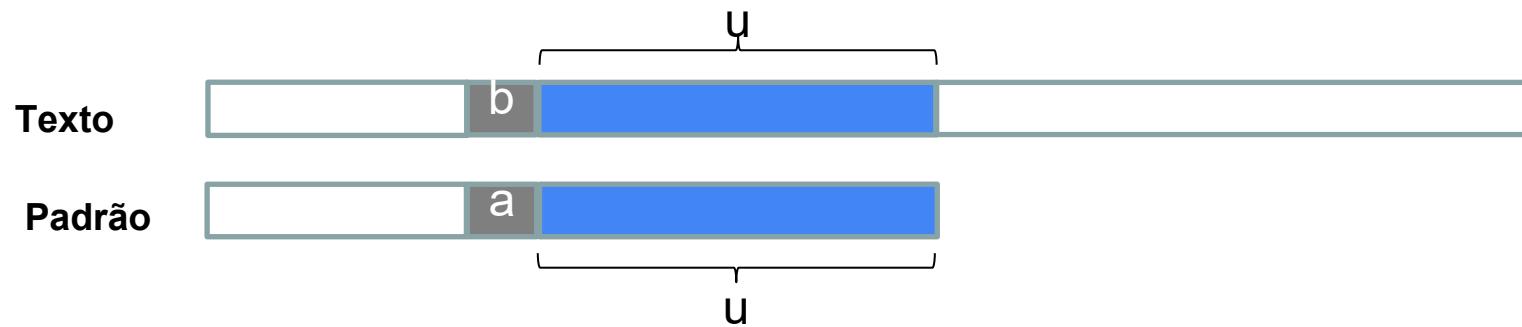
# BM – Heurísticas de Deslocamento

*Bad Character Shift (Deslocamento por Caracter Ruim)*



# BM – Heurísticas de Deslocamento

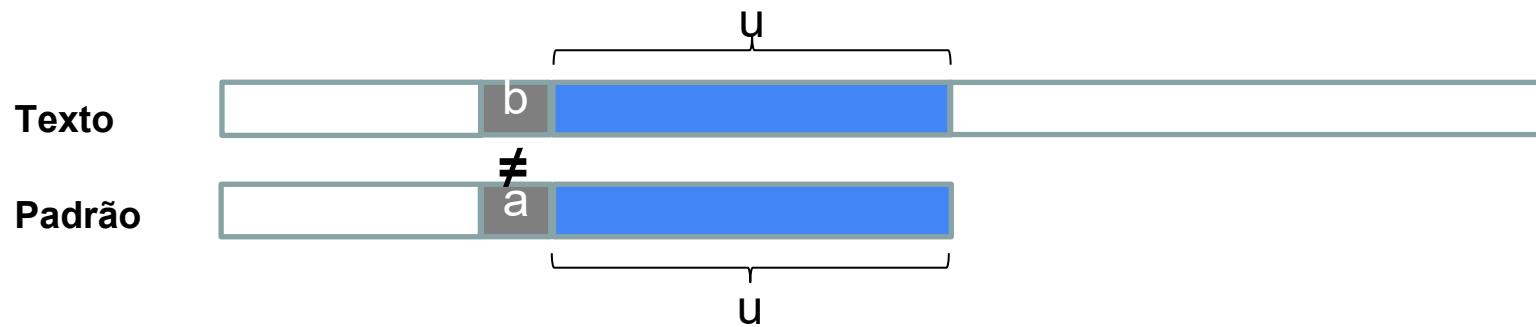
*Bad Character Shift (Deslocamento por Caracter Ruim)*



O caractere T do texto que não foi encontrado no padrão deve aparecer em outra posição do padrão (ou não adianta fazer comparações que o envolvam)

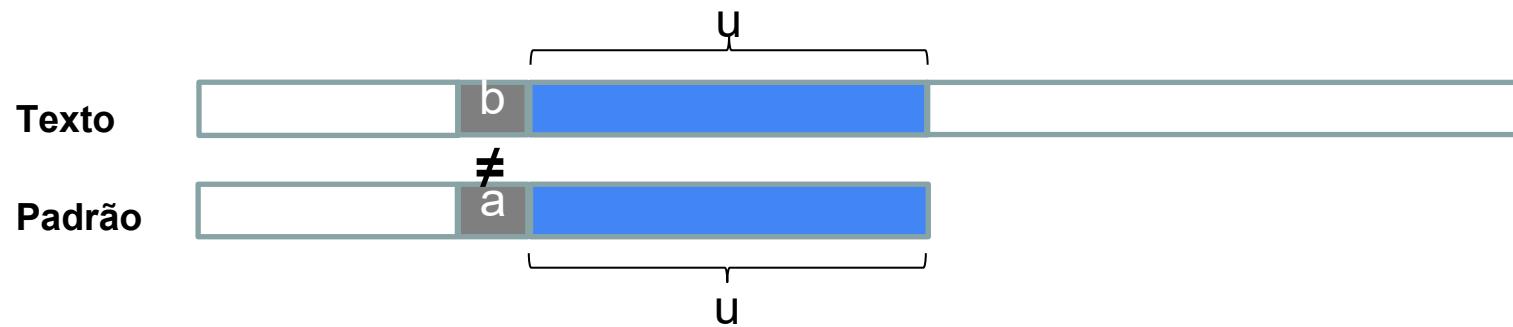
# BM – Heurísticas de Deslocamento

*Bad Character Shift (Deslocamento por Caracter Ruim)*



# BM – Heurísticas de Deslocamento

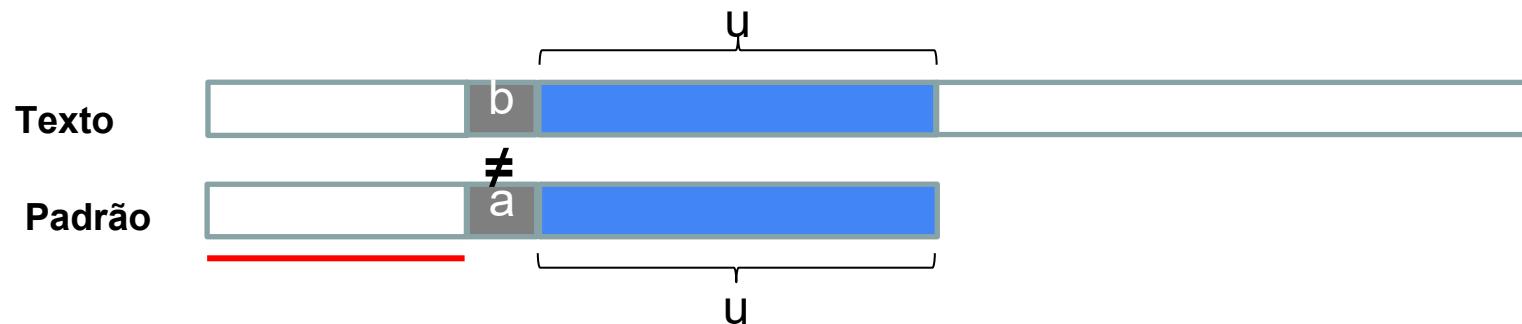
*Bad Character Shift (Deslocamento por Caracter Ruim)*



⇒ Padrão contém b !!!

# BM – Heurísticas de Deslocamento

*Bad Character Shift (Deslocamento por Caracter Ruim)*

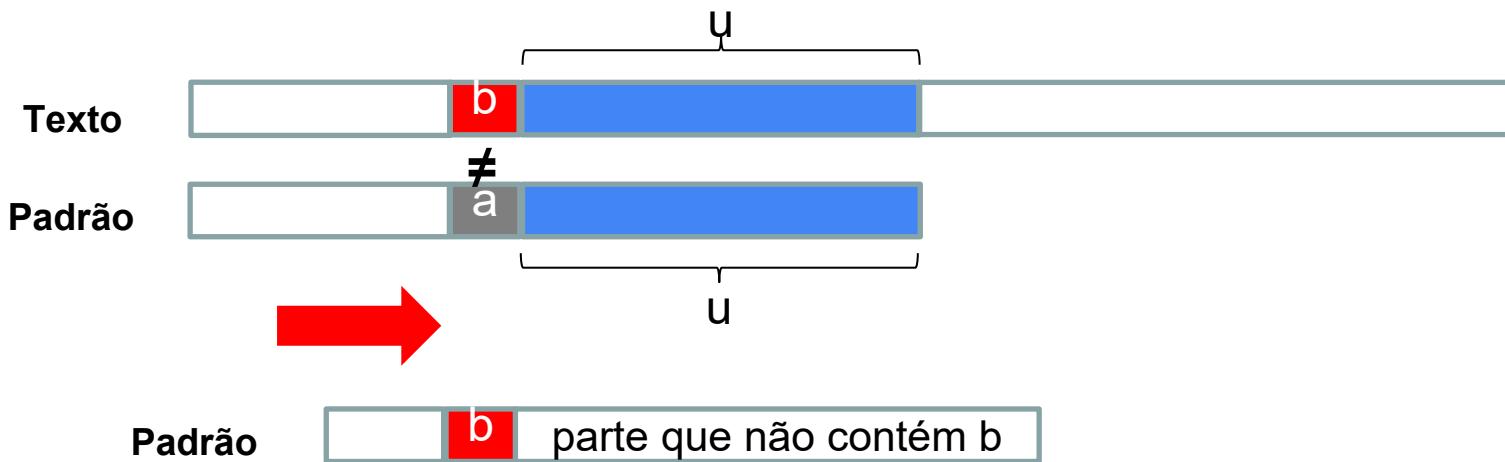


Padrão parte que não contém b

⇒ Padrão contém b !!!

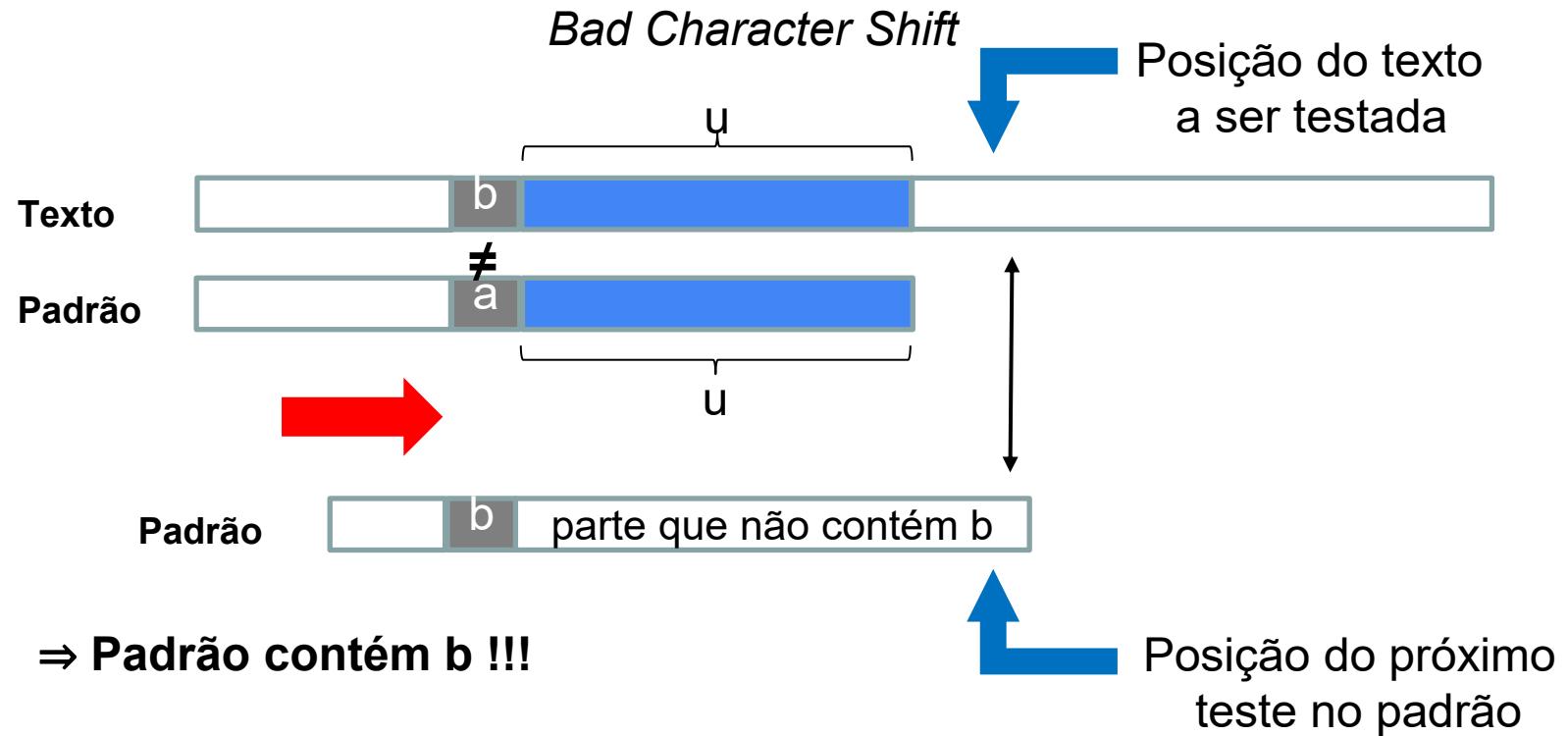
# BM – Heurísticas de Deslocamento

*Bad Character Shift (Deslocamento por Caracter Ruim)*



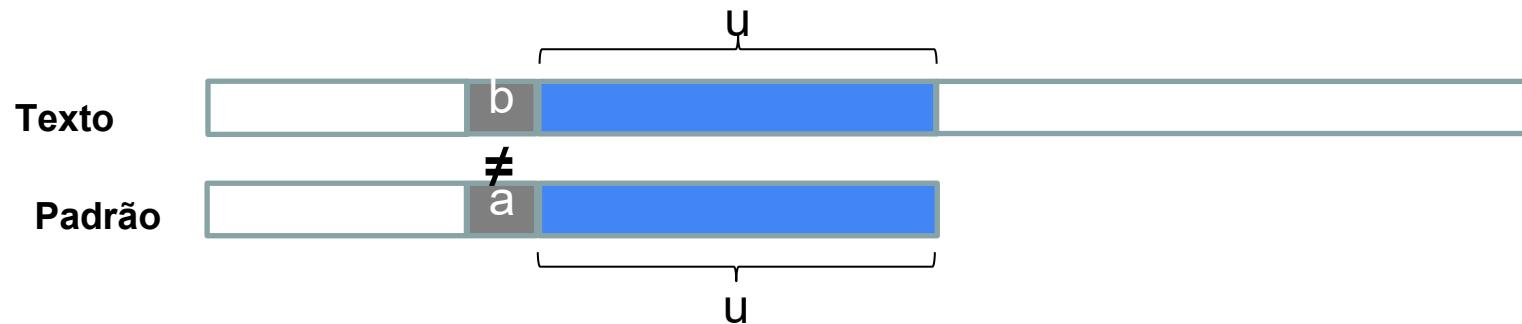
⇒ Padrão contém b !!!

# BM – Heurísticas de Deslocamento



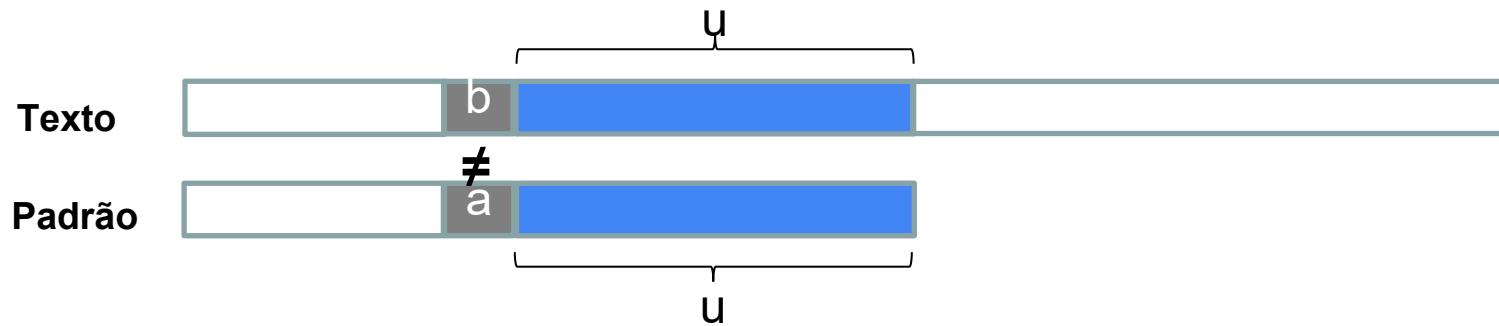
# BM – Heurísticas de Deslocamento

*Bad Character Shift (Deslocamento por Caracter Ruim)*



# BM – Heurísticas de Deslocamento

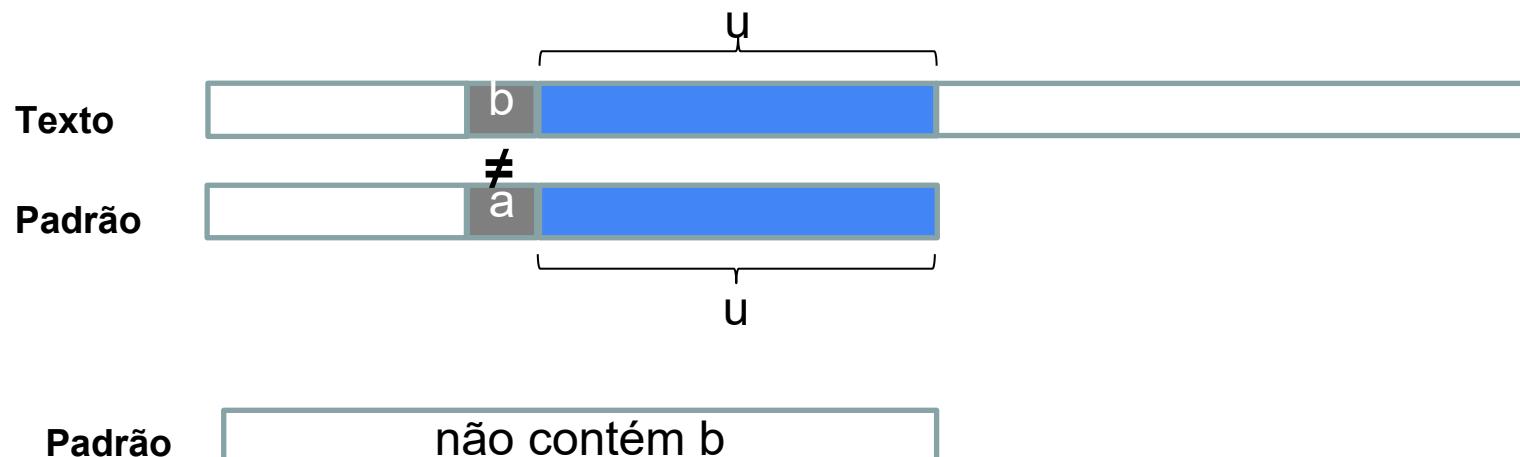
*Bad Character Shift (Deslocamento por Caracter Ruim)*



⇒ Padrão não contém b !!!

# BM – Heurísticas de Deslocamento

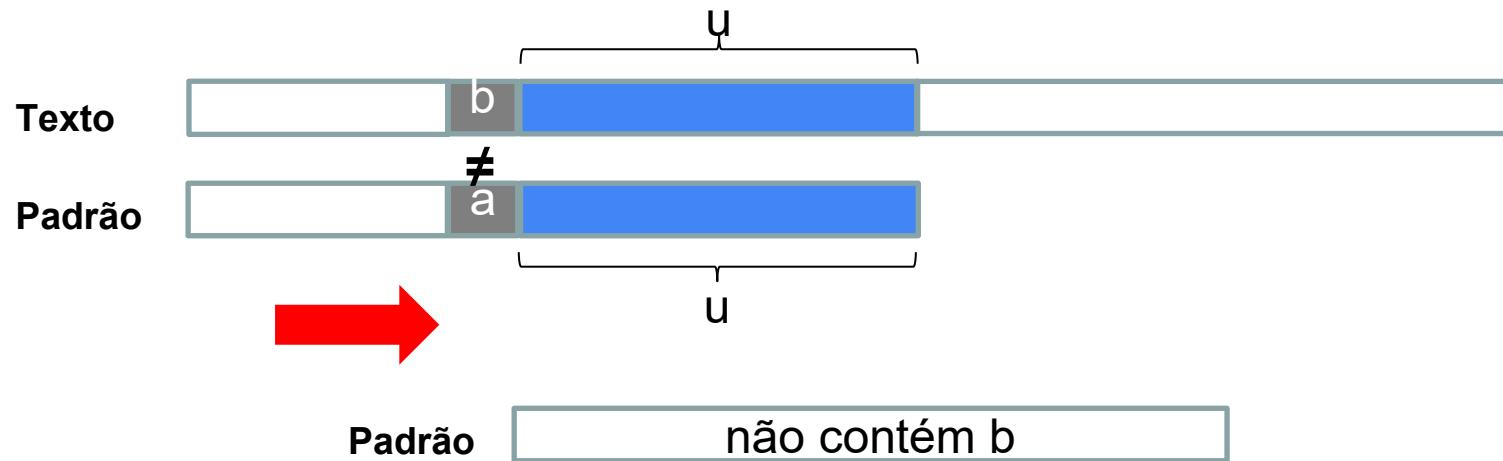
*Bad Character Shift (Deslocamento por Caracter Ruim)*



⇒ Padrão não contém b !!!

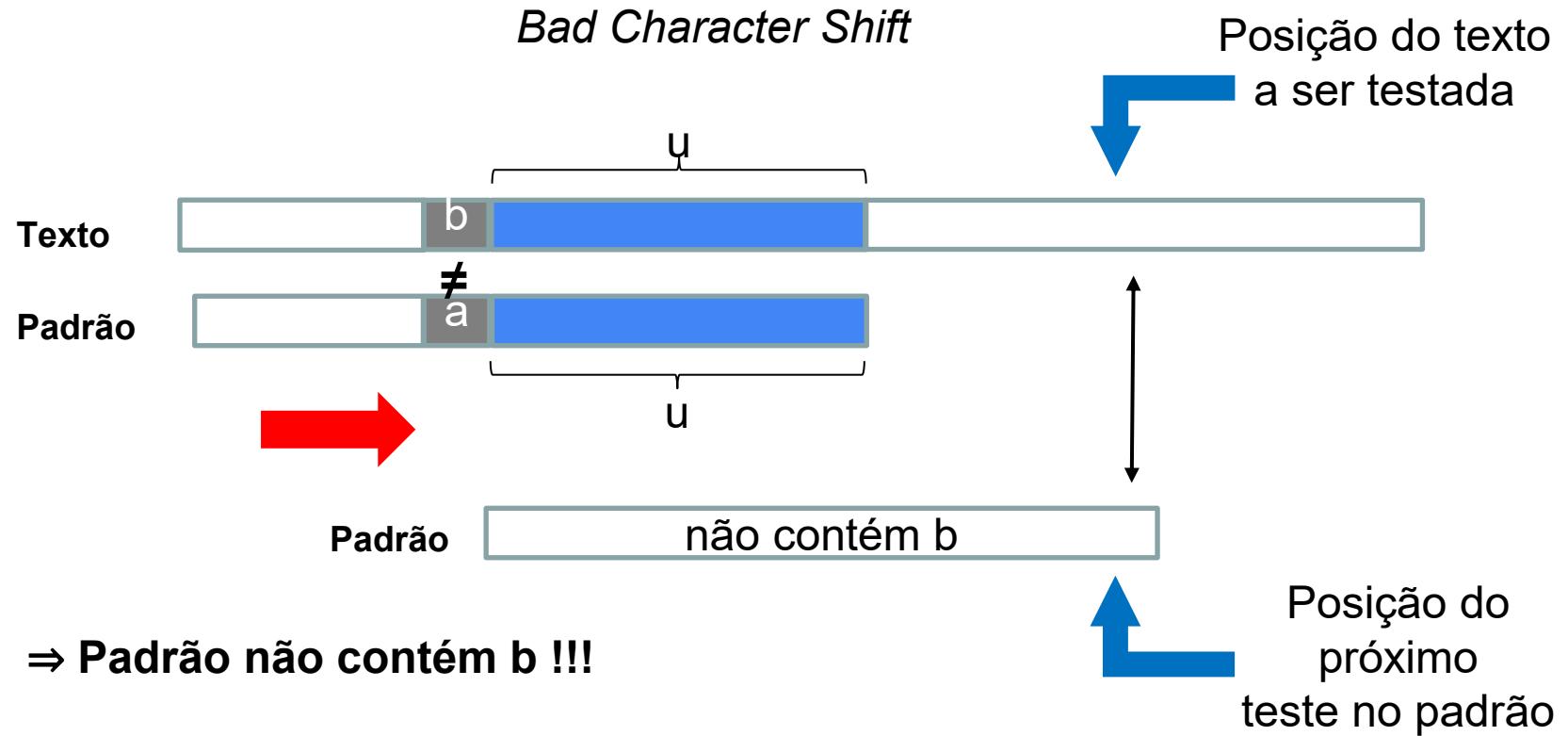
# BM – Heurísticas de Deslocamento

*Bad Character Shift (Deslocamento por Caracter Ruim)*



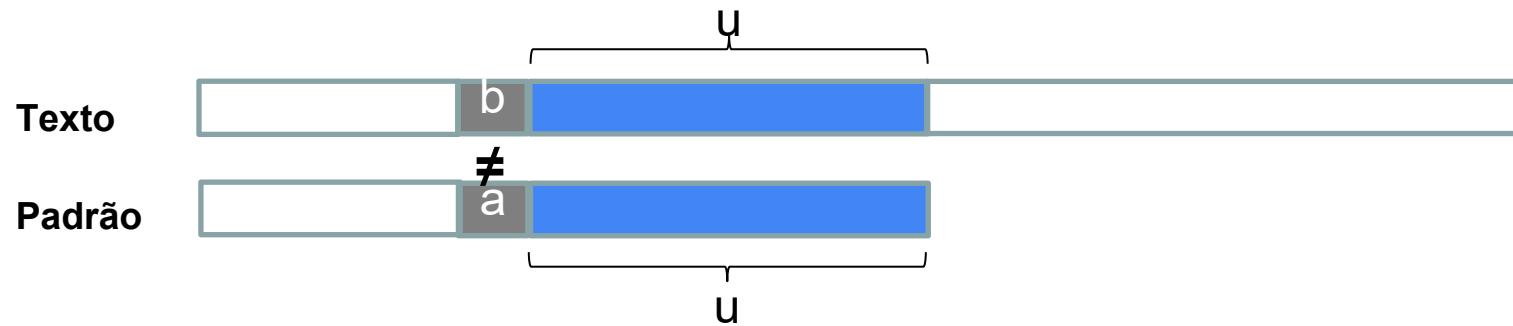
⇒ Padrão não contém b !!!

# BM – Heurísticas de Deslocamento



# BM – Heurísticas de Deslocamento

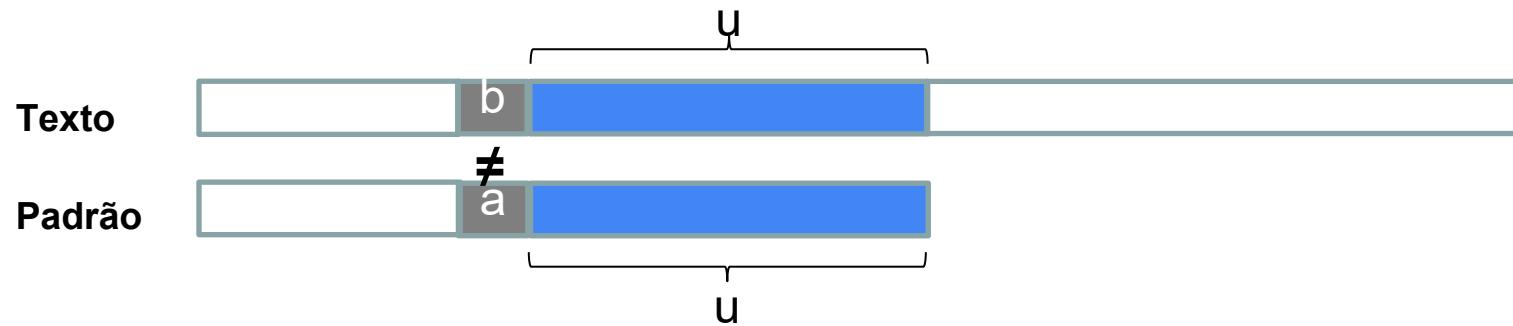
*Good Suffix Shift (Deslocamento por Sufixo Bom)*



Analisa repetição de sufixos no padrão (ao invés de prefixos como no KMP)  
Se um sufixo (ou parte dele) se repetir no padrão, então o deslocamento é feito para testar essa repetição.

# BM – Heurísticas de Deslocamento

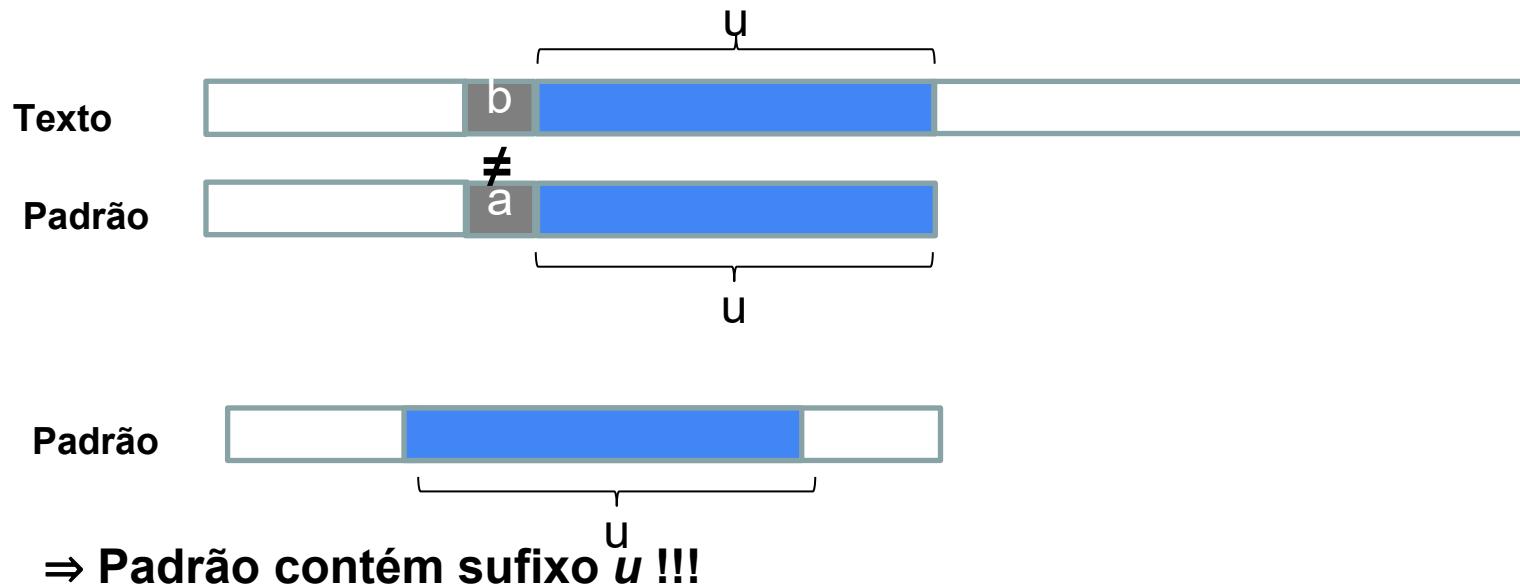
*Good Suffix Shift (Deslocamento por Sufixo Bom)*



⇒ Padrão contém sufixo *u* !!!

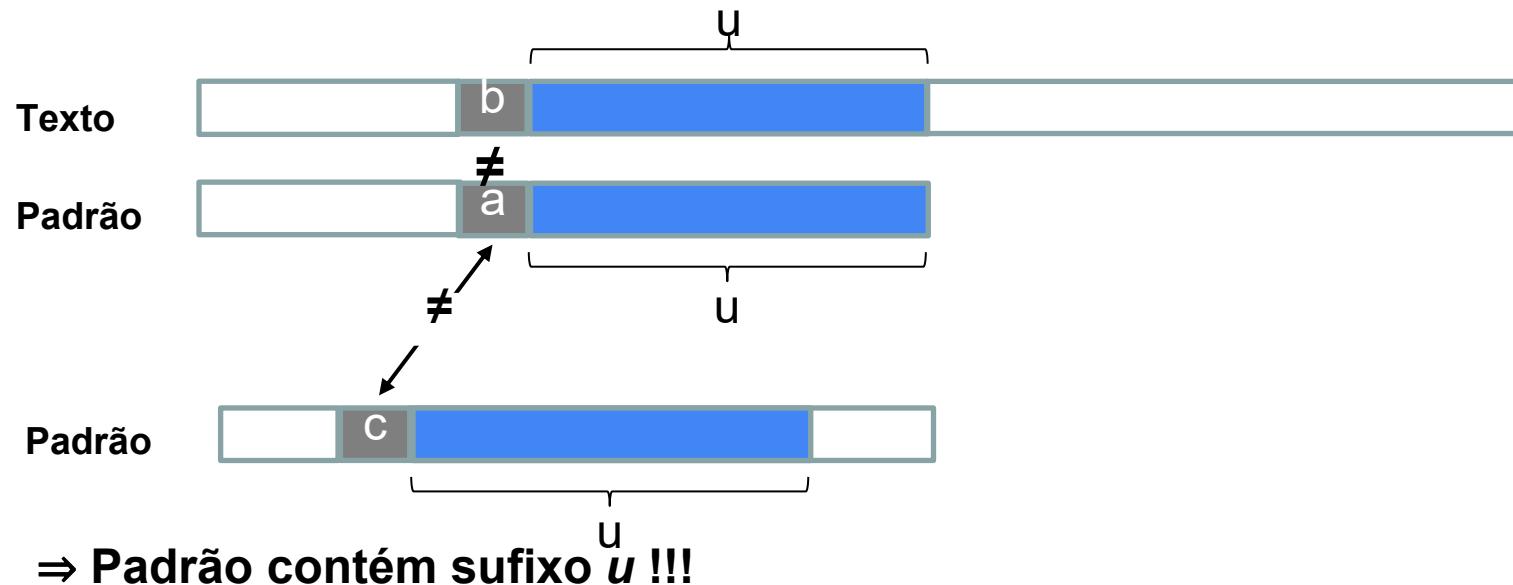
# BM – Heurísticas de Deslocamento

*Good Suffix Shift (Deslocamento por Sufixo Bom)*



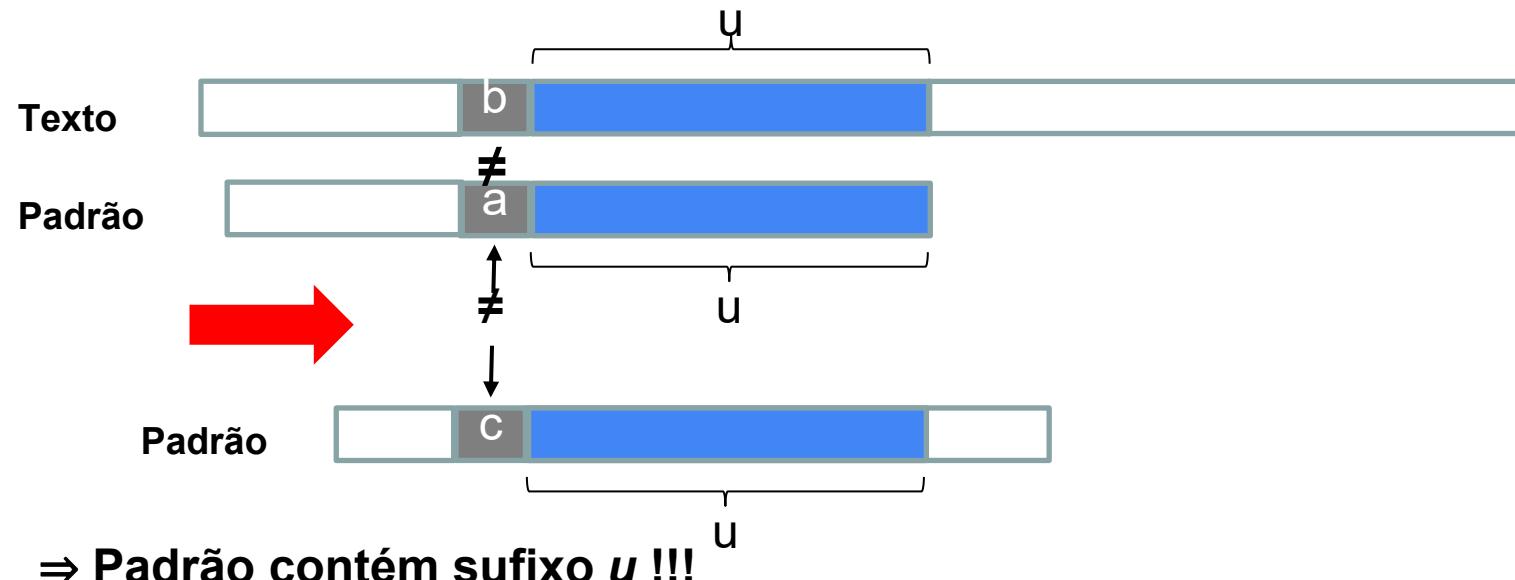
# BM – Heurísticas de Deslocamento

*Good Suffix Shift (Deslocamento por Sufixo Bom)*

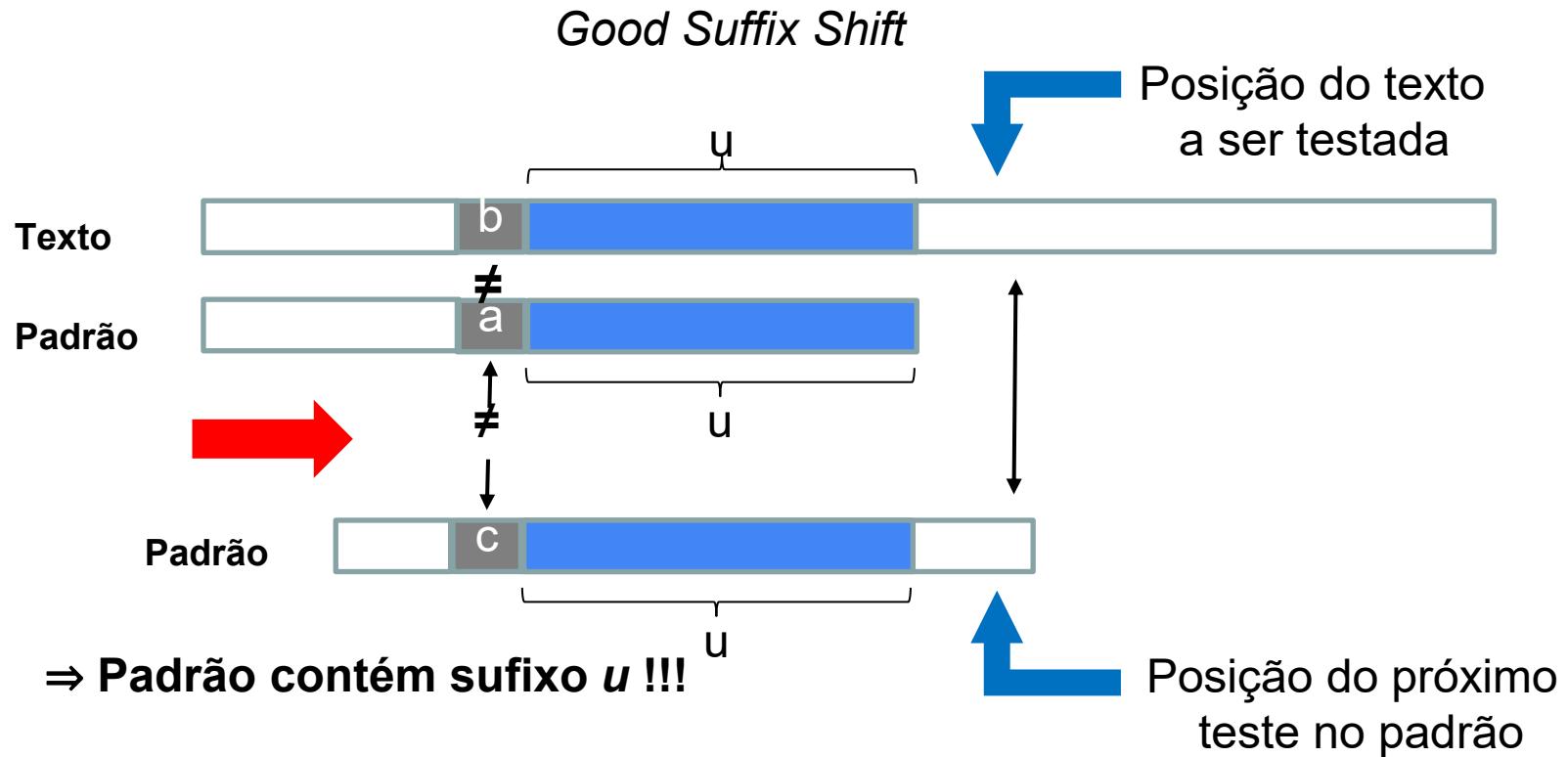


# BM – Heurísticas de Deslocamento

*Good Suffix Shift (Deslocamento por Sufixo Bom)*

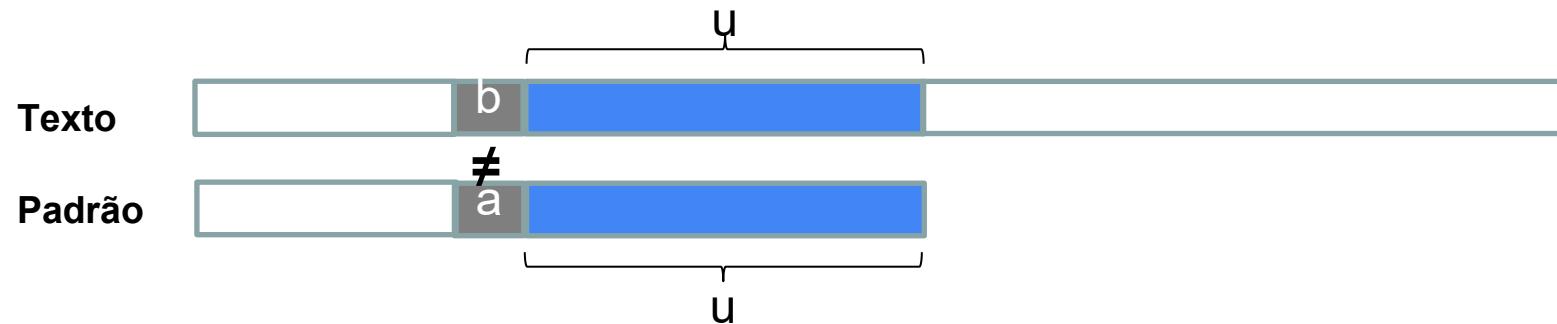


# BM – Heurísticas de Deslocamento



# BM – Heurísticas de Deslocamento

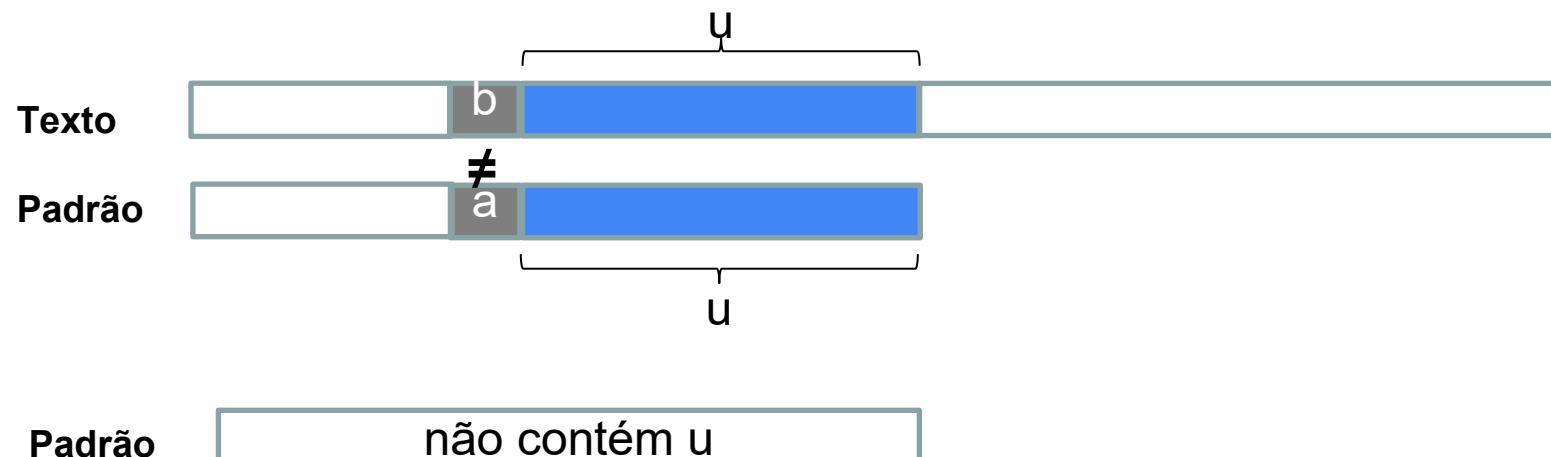
*Good Suffix Shift (Deslocamento por Sufixo Bom)*



⇒ Padrão não contém sufixo *u* !!!

# BM – Heurísticas de Deslocamento

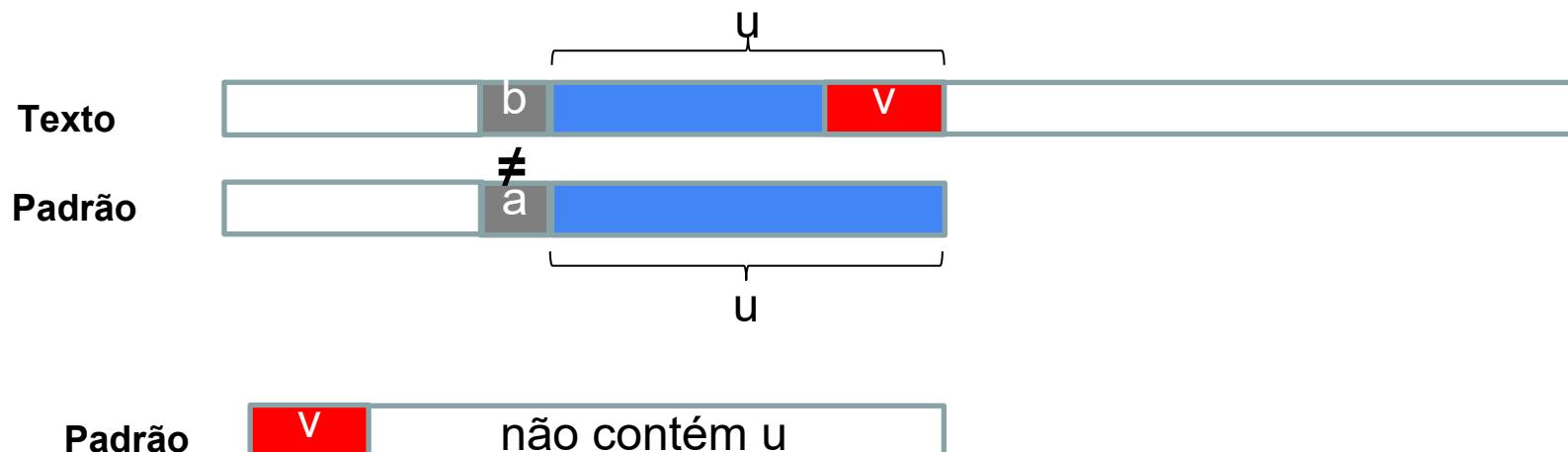
*Good Suffix Shift (Deslocamento por Sufixo Bom)*



⇒ Padrão não contém sufixo *u* !!!

# BM – Heurísticas de Deslocamento

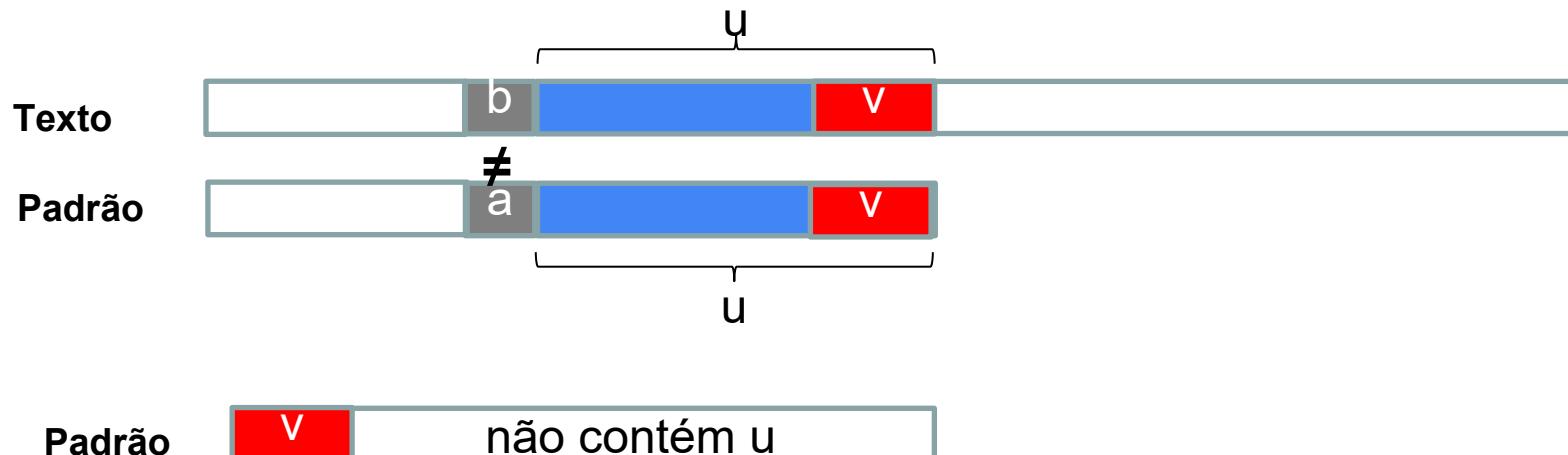
*Good Suffix Shift (Deslocamento por Sufixo Bom)*



⇒ Padrão não contém sufixo *u* !!!

# BM – Heurísticas de Deslocamento

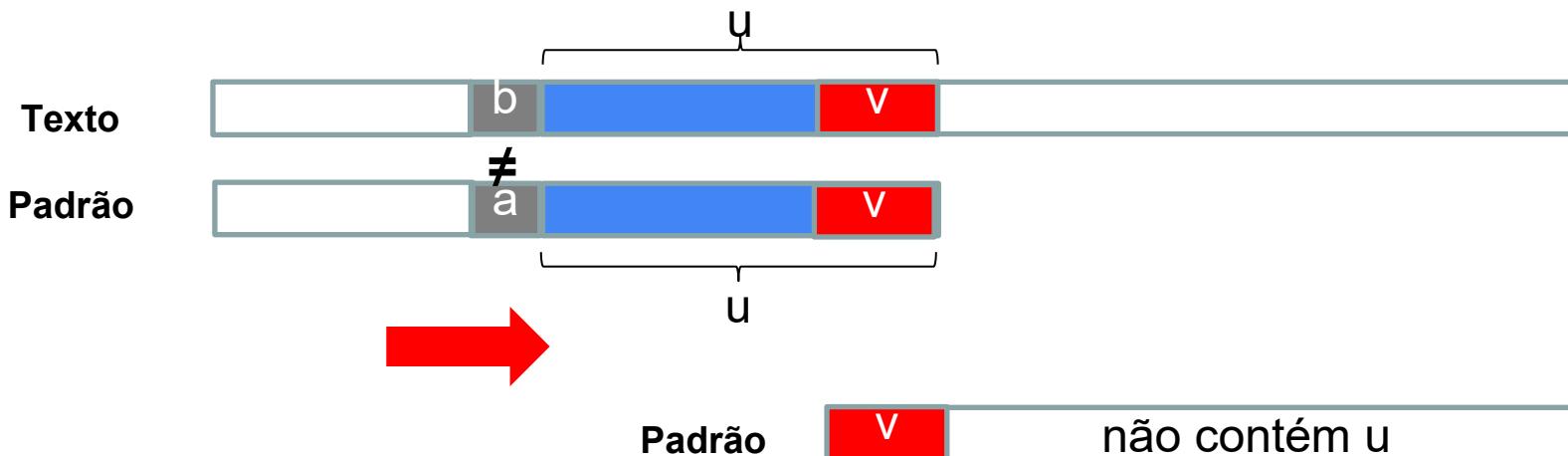
*Good Suffix Shift (Deslocamento por Sufixo Bom)*



⇒ Padrão não contém sufixo *u* !!!

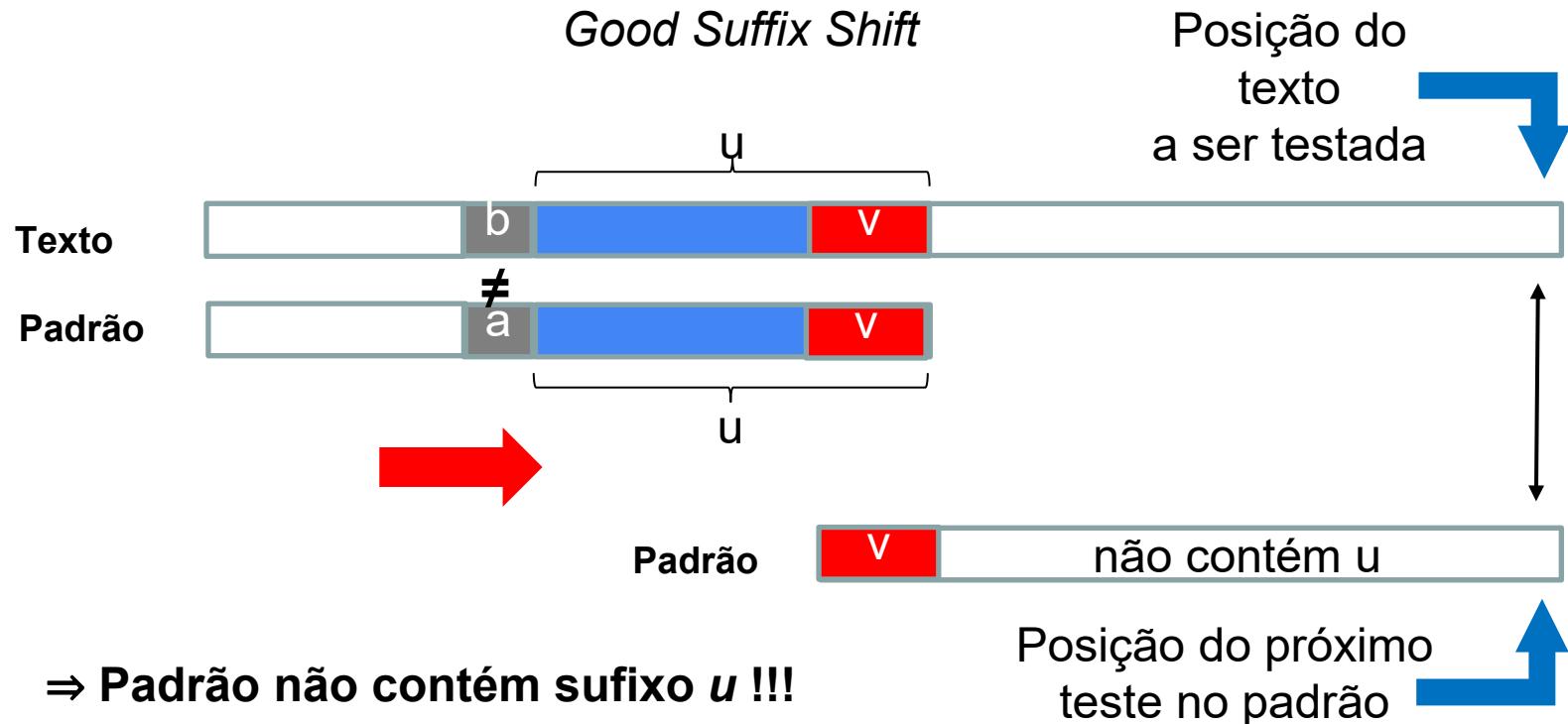
# BM – Heurísticas de Deslocamento

*Good Suffix Shift (Deslocamento por Sufixo Bom)*



⇒ Padrão não contém sufixo *u* !!!

# BM – Heurísticas de Deslocamento



# BM - Exemplo

## Deslocamento por Caracter Ruim

**Caso 1:** O caracter ruim (N) aparece em outra posição do padrão

Texto

0	1	2	3	4	5	6	7
A		A	R	A	N	H	A

Padrão

A	R	A	N	H	A
---	---	---	---	---	---

# BM - Exemplo

## Deslocamento por Caracter Ruim

**Caso 1:** O caracter ruim (N) aparece em outra posição do padrão

Sequência

0	1	2	3	4	5	6	7
A		A	R	A	N	H	A

Padrão

A	R	A	N	H	A
A	R	A	N	H	A

# BM - Exemplo

## Deslocamento por Caracter Ruim

**Caso 2:** O caracter ruim (N) não aparece em outra posição do padrão

Texto

0	1	2	3	4	5	6	7	8	9	10	11
A		Á	G	U	A		P	I	N	G	A

Padrão

A	R	A	N	H	A
---	---	---	---	---	---

# BM - Exemplo

## Deslocamento por Caracter Ruim

**Caso 2:** O caracter ruim (N) não aparece em outra posição do padrão

Sequência

0	1	2	3	4	5	6	7	8	9	10	11
A		Á	G	U	A		P	I	N	G	A

Padrão

A	R	A	N	H	A
---	---	---	---	---	---

# BM - Exemplo

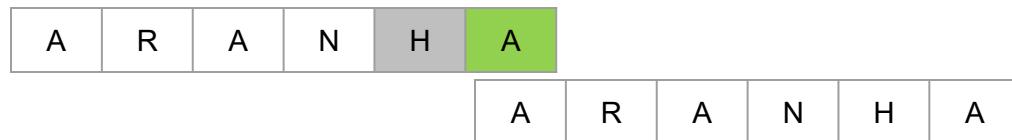
## Deslocamento por Caracter Ruim

**Caso 2:** O caracter ruim (N) não aparece em outra posição do padrão

Sequência

0	1	2	3	4	5	6	7	8	9	10	11
A		Á	G	U	A		P	I	N	G	A

Padrão



# BM - Exemplo

Deslocamento por Caractere Ruim

Cálculo

0	1	2	3	4	5
A	R	A	N	H	A

Observar a ocorrência mais à direita de cada caractere,  
exceto o último e inserir o caractere e sua posição em  
uma tabela hash.

Caracteres que não estiverem na tabela hash valerão -1.

A	2
H	4
N	3
R	1

Outros: -1

# BM - Exemplo

## Deslocamento por Sufixo Bom

**Caso 1:** Sufixo (BA) se repete com caractere anterior diferente

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
B	A	C	B	D	A	C	B	A	D	B	A	D	B	A	C	C	B	A	B	A	D	A	B



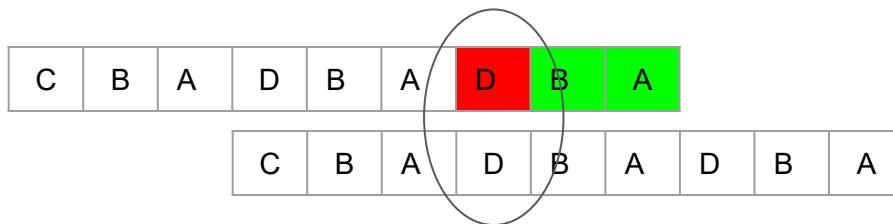
Mesmo caractere precede o sufixo BA!

# BM - Exemplo

## Deslocamento por Sufixo Bom

**Caso 1:** Sufixo (BA) se repete com caractere anterior diferente

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
B	A	C	B	D	A	C	<b>B</b>	<b>A</b>	D	B	A	D	B	A	C	C	B	A	B	A	D	A	B



Mesmo caractere precede o sufixo BA!

# BM - Exemplo

## Deslocamento por Sufixo Bom

**Caso 1:** Sufixo (BA) se repete com caractere anterior diferente

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
B	A	C	B	D	A	C	<b>B</b>	<b>A</b>	D	B	A	D	B	A	C	C	B	A	B	A	D	A	B



Busco até encontrar caractere anterior  
diferente, se não encontrar, cai no Caso 2

# BM - Exemplo

## Deslocamento por Sufixo Bom

**Caso 2:** Sufixo (BA) não se repete com caractere anterior diferente

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
B	A	C	B	D	A	C	<b>B</b>	<b>A</b>	D	B	A	D	B	A	C	C	B	A	B	A	D	A	B

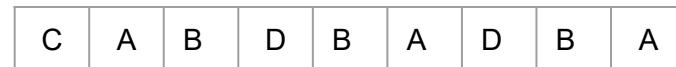
C	A	B	D	B	A	<b>D</b>	<b>B</b>	<b>A</b>
---	---	---	---	---	---	----------	----------	----------

# BM - Exemplo

## Deslocamento por Sufixo Bom

**Caso 2:** Sufixo (BA) não se repete com caractere anterior diferente

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
B	A	C	B	D	A	C	B	A	D	B	A	D	B	A	C	C	B	A	B	A	D	A	B



Pulo tudo!

# BM - Exemplo

## Deslocamento por Sufixo Bom

**Caso 3:** Parte do sufixo se repete no início do padrão

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
B	A	C	B	D	A	D	B	A	D	B	A	D	B	A	C	C	B	A	B	A	D	A	B

B	A	B	D	B	A	D	B	A
---	---	---	---	---	---	---	---	---

# BM - Exemplo

## Deslocamento por Sufixo Bom

**Caso 3:** Parte do sufixo se repete no início do padrão

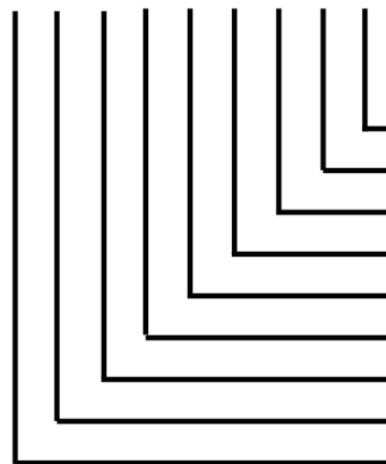
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
B	A	C	B	D	A	D	B	A	D	B	A	D	B	A	C	C	B	A	B	A	D	A	B



# BM - Exemplo

0	1	2	3	4	5	6	7	8
B	A	A	D	B	D	C	B	A

7	7	7	7	7	7	7	6	1
---	---	---	---	---	---	---	---	---



← Vetor de deslocamentos

(o índice representa o ponto onde ocorreu a diferença)

→ Nenhum sufixo foi encontrado

→ Sufixo A válido. Sufixo se repete 6 posições atrás e a letra anterior é diferente de b

→ Sufixo BA válido. Sufixo se repete 7 posições atrás.

→ Sufixo CBA válido. Sufixo não se repete, mas fragmento BA existe 7 posições atrás.

→ Sufixo DCBA válido. Sufixo não se repete, mas fragmento BA existe 7 posições atrás.

→ Sufixo BDCBA válido. Sufixo não se repete, mas fragmento BA existe 7 posições atrás.

→ Sufixo DBDCBA válido. Sufixo não se repete, mas fragmento BA existe 7 posições atrás.

→ Sufixo ADBDCBA válido. Sufixo não se repete, mas fragmento BA existe 7 posições atrás.

→ Sufixo AADBDCBA válido. Sufixo não se repete, mas fragmento BA existe 7 posições atrás.

## Regras:

1. sufixo bom aparece no padrão com prefixo diferente?
2. sufixo bom é o prefixo do padrão?
  - 2.1. Enquanto tiver sufixo bom, vou reduzindo o sufixo bom e repetindo a pergunta 2.
3. Se não encontrado 1 e nem 2, usar o tamanho do padrão

# BM - Exemplo

**Padrão**

0	1	2	3	4	5	6	7
G	C	A	G	A	G	A	G

**Tabela hash para cálculo dos deslocamentos por caráter ruim**

A	C	G	...
6	1	5	-1

**Vetor para cálculo dos deslocamentos por sufixo bom (DSB)**

0	1	2	3	4	5	6	7
G	C	A	G	A	G	A	G
7	7	7	2	7	4	7	1

# BM - Exemplo

**DCR**

A	C	G	...
6	1	5	-1

**DSB**

0	1	2	3	4	5	6	7
G	C	A	G	A	G	A	G

0	1	2	3	4	5	6	7
G	C	A	G	A	G	A	G
7	7	7	2	7	4	7	1

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
G	C	A	T	C	G	C	A	G	A	G	A	G	T	A

# BM - Exemplo

Caractere ruim **A**. Deslocamento: **7** - DCR['A'] = 7 - 6 = 1

Sufixo bom. Deslocamento: DSB[**7**] = 1

Melhor deslocamento: 1

0	1	2	3	4	5	6	<b>7</b>
G	C	A	G	A	G	A	G



0	1	2	3	4	5	6	<b>7</b>	8	9	10	11	12	13	14
G	C	A	T	C	G	C	<b>A</b>	G	A	G	A	G	T	A

**DCR**

A	C	G	...
6	1	5	-1

**DSB**

0	1	2	3	4	5	6	7
G	C	A	G	A	G	A	G
7	7	7	2	7	4	7	1

# BM - Exemplo

**DCR**

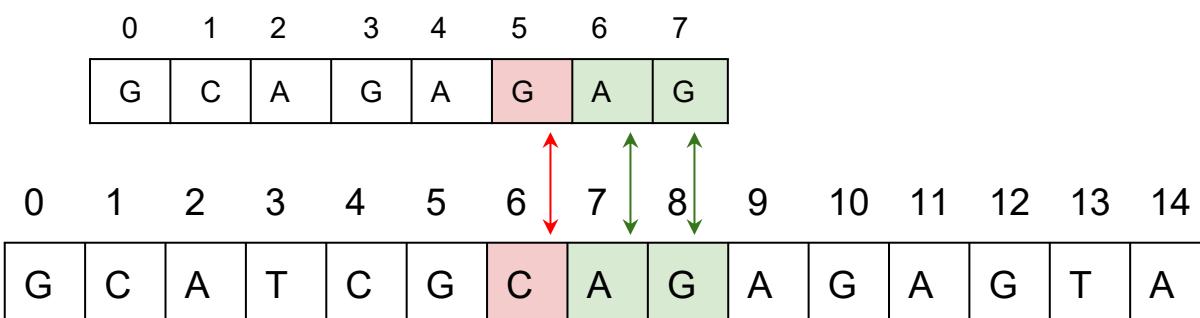
A	C	G	...
6	1	5	-1

**DSB**

0	1	2	3	4	5	6	7
G	C	A	G	A	G	A	G

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
G	C	A	T	C	G	C	A	G	A	G	A	G	T	A



# BM - Exemplo

Caractere ruim **C**. Deslocamento: **5** - DCR['C'] = 5 - 1 = 4

Sufixo bom. Deslocamento: DSB[**5**] = 4

Melhor deslocamento: 4

**DCR**

A	C	G	...
6	1	5	-1

**DSB**

0	1	2	3	4	5	6	7
G	C	A	G	A	G	A	G

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
G	C	A	T	C	G	C	A	G	A	G	A	G	T	A

0	1	2	3	4	5	6	7
G	C	A	G	A	G	A	G

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
G	C	A	T	C	G	C	A	G	A	G	A	G	T	A

# BM - Exemplo

Padrão encontrado na posição 5 do texto.

**DCR**

A	C	G	...
6	1	5	-1

**DSB**

0	1	2	3	4	5	6	7
G	C	A	G	A	G	A	G
7	7	7	2	7	4	7	1

0	1	2	3	4	<b>5</b>	6	7	8	9	10	11	12	13	14
G	C	A	T	C	G	C	A	G	A	G	A	G	T	A

# BM - Exemplo

Online

<http://jovilab.sinaapp.com/visualization/algorithms/strings/boyer-moore-horspool>