



UNIVERSIDADE  
FEDERAL DE  
SERGIPE



DEPARTAMENTO  
DE COMPUTAÇÃO

# Busca em cadeias (Knuth-Morris-Pratt)

## Projeto e Análise de Algoritmos

Bruno Prado

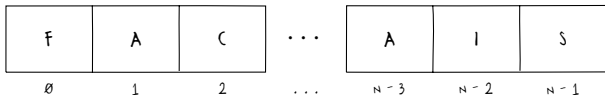
Departamento de Computação / UFS

# Introdução

- ▶ O que é uma cadeia?
  - ▶ É uma sequência de símbolos  $T$  com tamanho  $n$

# Introdução

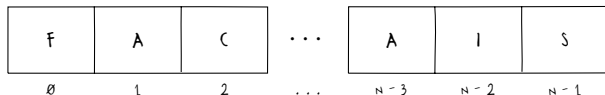
- ▶ O que é uma cadeia?
  - ▶ É uma sequência de símbolos  $T$  com tamanho  $n$
  - ▶ Os símbolos são definidos por um alfabeto finito  $\Sigma$



$$\Sigma = \{a, b, \dots, y, z\}$$

# Introdução

- ▶ O que é uma cadeia?
  - ▶ É uma sequência de símbolos  $T$  com tamanho  $n$
  - ▶ Os símbolos são definidos por um alfabeto finito  $\Sigma$

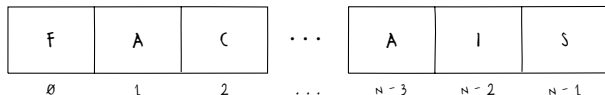


$$\Sigma = \{a, b, \dots, y, z\}$$

- ▶ Aplicações multidisciplinares
  - ▶ Biologia: representação da cadeia de DNA, sendo composta pelos símbolos  $A, C, G, T$

# Introdução

- ▶ O que é uma cadeia?
  - ▶ É uma sequência de símbolos  $T$  com tamanho  $n$
  - ▶ Os símbolos são definidos por um alfabeto finito  $\Sigma$

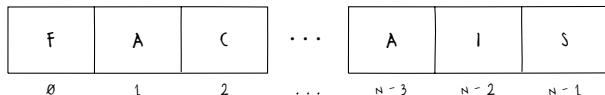


$$\Sigma = \{a, b, \dots, y, z\}$$

- ▶ Aplicações multidisciplinares
  - ▶ Biologia: representação da cadeia de DNA, sendo composta pelos símbolos  $A, C, G, T$
  - ▶ Computação: armazenamento de texto através do tipo *string*, com o padrão de codificação ASCII

# Introdução

- ▶ O que é uma cadeia?
  - ▶ É uma sequência de símbolos  $T$  com tamanho  $n$
  - ▶ Os símbolos são definidos por um alfabeto finito  $\Sigma$



$$\Sigma = \{a, b, \dots, y, z\}$$

- ▶ Aplicações multidisciplinares
  - ▶ Biologia: representação da cadeia de DNA, sendo composta pelos símbolos  $A, C, G, T$
  - ▶ Computação: armazenamento de texto através do tipo *string*, com o padrão de codificação ASCII
  - ▶ ...

# Introdução

- ▶ Notação e terminologia
  - ▶ Todas as cadeias de tamanho finito que podem ser construídas do alfabeto finito  $\Sigma$  é definido por  $\Sigma^*$

# Introdução

- ▶ Notação e terminologia
  - ▶ Todas as cadeias de tamanho finito que podem ser construídas do alfabeto finito  $\Sigma$  é definido por  $\Sigma^*$
  - ▶ Uma cadeia vazia é denotada pelo símbolo  $\varepsilon$

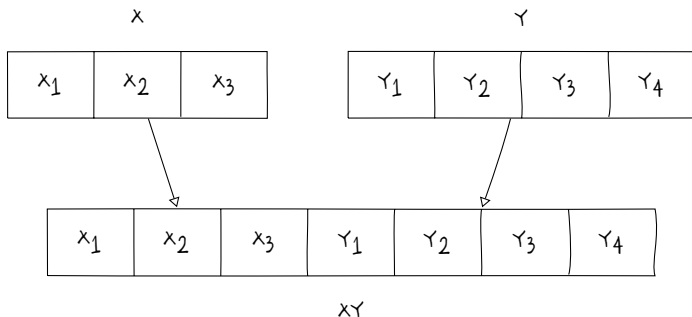


# Introdução

- ▶ Notação e terminologia
  - ▶ Todas as cadeias de tamanho finito que podem ser construídas do alfabeto finito  $\Sigma$  é definido por  $\Sigma^*$
  - ▶ Uma cadeia vazia é denotada pelo símbolo  $\varepsilon$
  - ▶ O tamanho de uma cadeia  $x$  é definida por  $|x|$

# Introdução

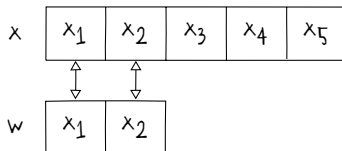
- ▶ Notação e terminologia
  - ▶ Todas as cadeias de tamanho finito que podem ser construídas do alfabeto finito  $\Sigma$  é definido por  $\Sigma^*$
  - ▶ Uma cadeia vazia é denotada pelo símbolo  $\varepsilon$
  - ▶ O tamanho de uma cadeia  $x$  é definida por  $|x|$
  - ▶ A concatenação de duas cadeias  $x$  e  $y$  resulta em uma cadeia  $xy$  com os caracteres de  $x$  seguidos dos caracteres de  $y$ , com tamanho total de  $|x| + |y|$



# Introdução

- ▶ Notação e terminologia

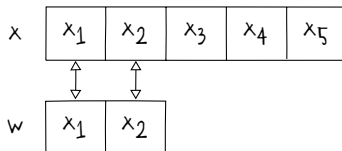
- ▶ Prefixo: a cadeia  $w$  é um prefixo da cadeia  $x$  ( $w \sqsubseteq x$ ) se  $x = wy$ , com  $y \in \Sigma^*$  e  $|w| \leq |x|$



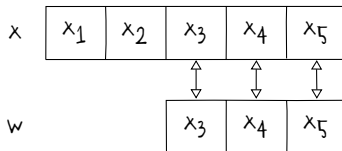
# Introdução

## ► Notação e terminologia

- **Prefixo:** a cadeia  $w$  é um prefixo da cadeia  $x$  ( $w \sqsubseteq x$ ) se  $x = wy$ , com  $y \in \Sigma^*$  e  $|w| \leq |x|$

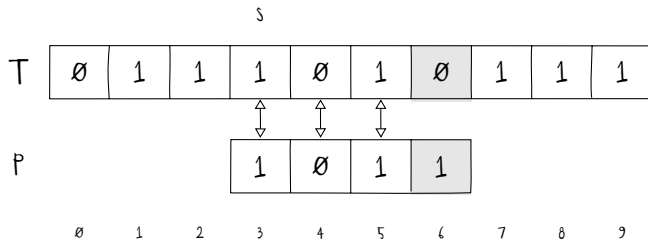


- **Sufixo:** a cadeia  $w$  é sufixo da cadeia  $x$  ( $w \sqsubseteq x$ ) se  $x = yw$ , com  $y \in \Sigma^*$  e  $|w| \leq |x|$



# Introdução

- ▶ Como pode ser definida a busca em cadeias?
  - ▶ É o processo para encontrar todas as ocorrências de um padrão  $P$  em uma cadeia  $T$  que possuem  $m$  e  $n$  símbolos, respectivamente, onde  $m \leq n$



$$\begin{aligned}\Sigma &= \{0, 1\} \\ |P| &= m = 4, \quad |T| = n = 10 \\ 0 &\leq s \leq n - m\end{aligned}$$

# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
  - ▶ É um algoritmo linear para busca em cadeia que utiliza pré-processamento do padrão, armazenando uma tabela para comparação em tempo constante

# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
  - ▶ É um algoritmo linear para busca em cadeia que utiliza pré-processamento do padrão, armazenando uma tabela para comparação em tempo constante
  - ▶ O princípio de funcionamento é baseado em autômatos finitos e na tabela de transição

# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)

- ▶ É um algoritmo linear para busca em cadeia que utiliza pré-processamento do padrão, armazenando uma tabela para comparação em tempo constante
- ▶ O princípio de funcionamento é baseado em autômatos finitos e na tabela de transição
- ▶ Em cada posição da tabela é armazenado o comprimento do maior prefixo de  $P_i$  que é um sufixo de  $P_j$  através da função de prefixo  $k$

$$k(i) = \max \left\{ (j-1) : (j < i) \wedge (P_j \sqsupseteq P_i) \right\}$$



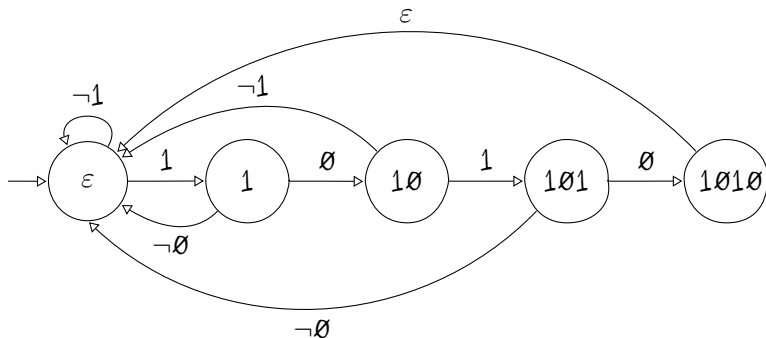
# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
  - ▶ Cálculo da tabela de transição

```
1 // Padrão de tipos por tamanho
2 #include <stdint.h>
3 // Procedimento de cálculo da tabela de transição
4 void calcular_tabela(int32_t* k, char* P) {
5     // i = sufixo, j = prefixo
6     for(int32_t i = 1, j = -1; i < strlen(P); i++) {
7         // Prefixo e sufixo diferentes
8         while(j >= 0 && P[j + 1] != P[i])
9             // Retorno de estado
10            j = k[j];
11        // Combinação de prefixo e sufixo
12        if(P[j + 1] == P[i])
13            // Avanço de estado
14            j++;
15        // Atualização da transição do estado
16        k[i] = j;
17    }
18 }
```

# Busca em cadeias

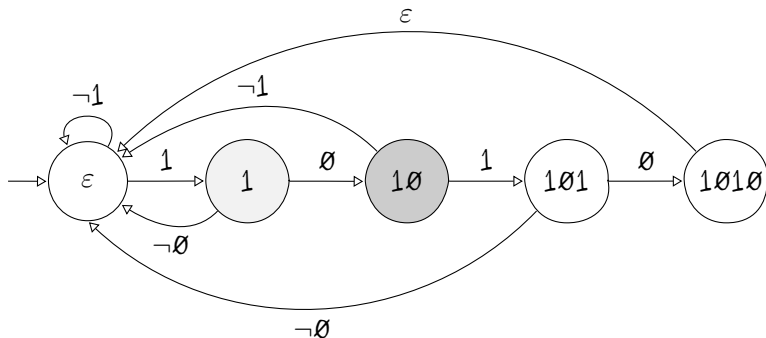
- Knuth-Morris-Pratt (KMP)
- Cálculo da tabela de transição



P	1	0	1	0
K	-1	-1	-1	-1
-1	0	1	2	3
J		1		

# Busca em cadeias

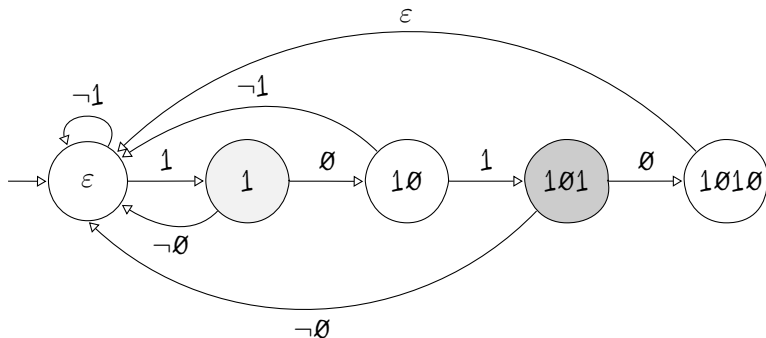
- Knuth-Morris-Pratt (KMP)
- Cálculo da tabela de transição



P	1	0	1	0
K	-1	-1	-1	-1
-1	0	1	2	3
J		1		

# Busca em cadeias

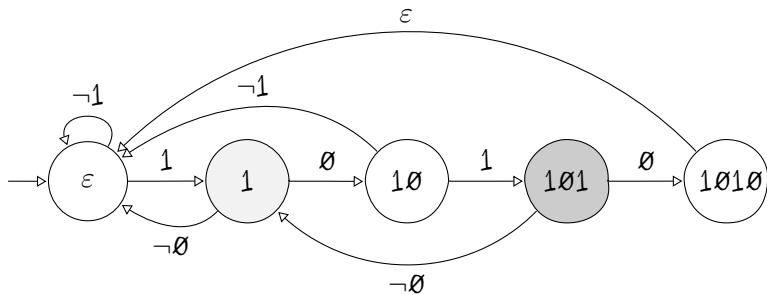
- Knuth-Morris-Pratt (KMP)
- Cálculo da tabela de transição



P	1	0	1	0
K	-1	-1	-1	-1
-1	0	1	2	3
J			1	

# Busca em cadeias

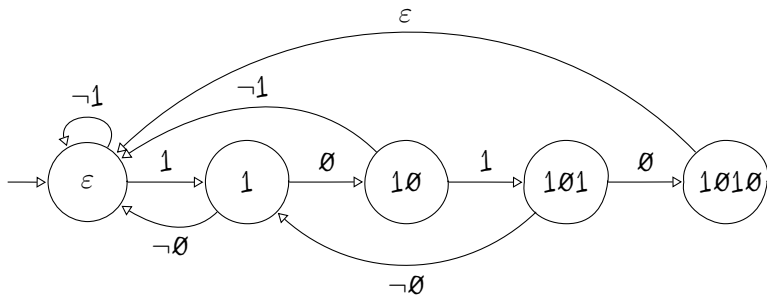
- Knuth-Morris-Pratt (KMP)
- Cálculo da tabela de transição



P	1	0	1	0
K	-1	-1	0	-1
-1	0	1	2	3
	j		i	

# Busca em cadeias

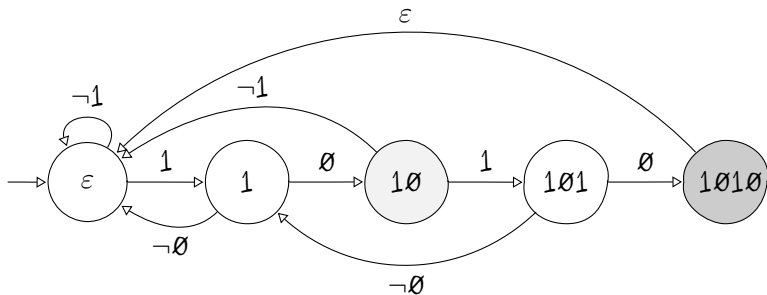
- ▶ Knuth-Morris-Pratt (KMP)
- ▶ Cálculo da tabela de transição



P	1	0	1	0
K	-1	-1	0	-1
-1	0	1	2	3
	j			i

# Busca em cadeias

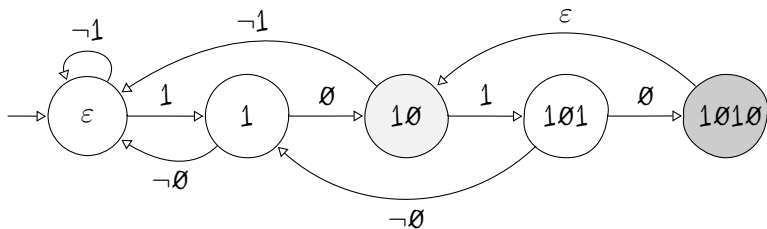
- ▶ Knuth-Morris-Pratt (KMP)
- ▶ Cálculo da tabela de transição



P	1	0	1	0
K	-1	-1	0	-1
-1	0	1	2	3
	j			i

# Busca em cadeias

- Knuth-Morris-Pratt (KMP)
- Cálculo da tabela de transição

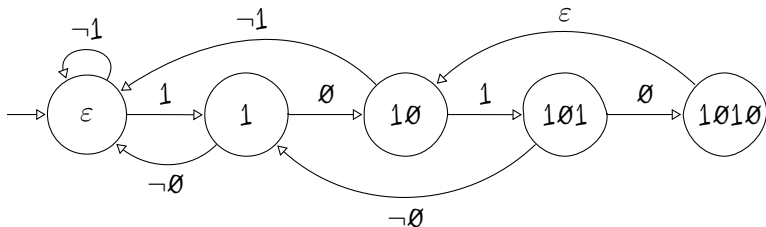


P	1	0	1	0
K	-1	-1	0	1
-1	0	1	2	3
	j		i	



# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
  - ▶ Autômato Finito Determinístico



P	1	0	1	0
K	-1	-1	0	1
-1	0	1	2	3

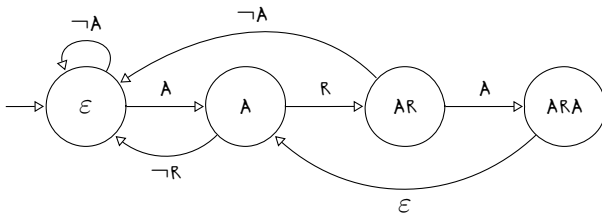
# Busca em cadeias

## ► Knuth-Morris-Pratt (KMP)

```
...
20 // Busca por KMP
21 void KMP(int32_t* k, int32_t* R, char* T, char* P) {
22     // Pré-processamento
23     int32_t n = strlen(T), m = strlen(P);
24     calcular_tabela(k, P);
25     // Iterando na cadeia T
26     for(int32_t i = 0, j = -1; i < n; i++) {
27         // Retorno de estado
28         while(j >= 0 && P[j + 1] != T[i]) j = k[j];
29         // Avanço de estado
30         if(P[j + 1] == T[i]) j++;
31         // Combinação do padrão
32         if(j == m - 1) {
33             inserir(R, i - m + 1);
34             j = k[j];
35         }
36     }
```

# Busca em cadeias

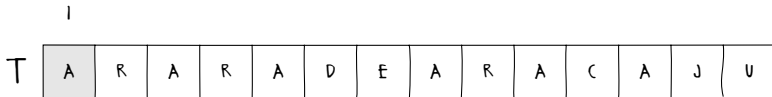
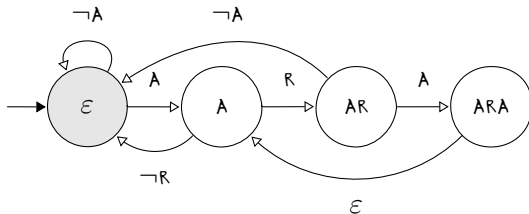
- ▶ Knuth-Morris-Pratt (KMP)
  - ▶ Tabela de transição do padrão  $P = ara$



P	A	R	A
K	-1	-1	0
-1	0	1	2

# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
- ▶ Buscando  $P = ara$  em  $T = araradearacaju$

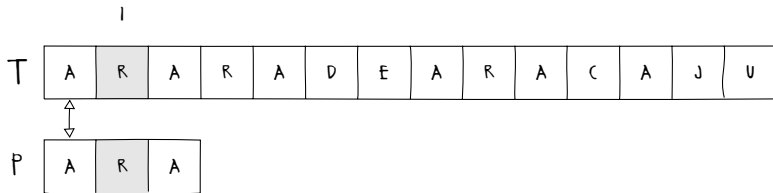
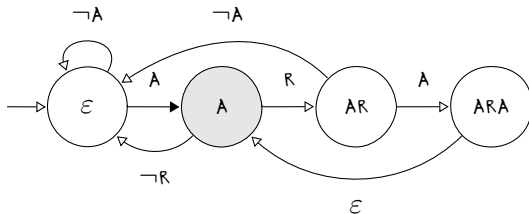


$R$

$\emptyset$     1    2    3    4    5    6    7    8    9    10    11    12    13

# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
- ▶ Buscando  $P = ara$  em  $T = araradearacaju$

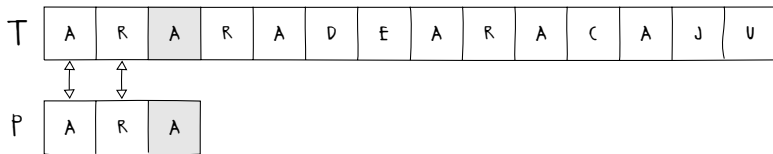
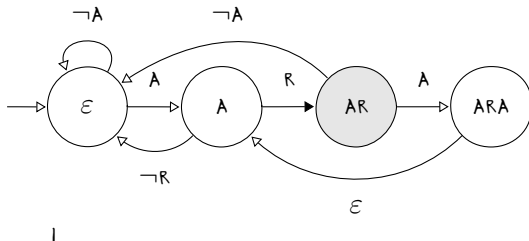


$R$

0 1 2 3 4 5 6 7 8 9 10 11 12 13

# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
  - ▶ Buscando  $P = ara$  em  $T = araradearacaju$

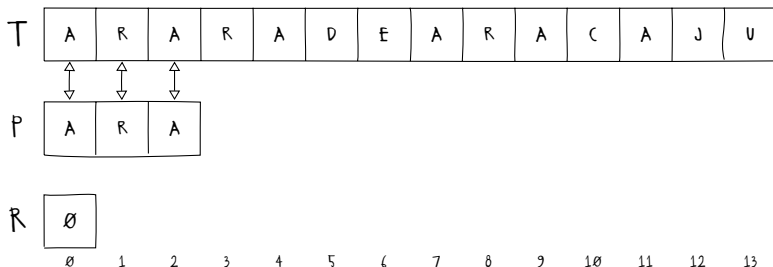
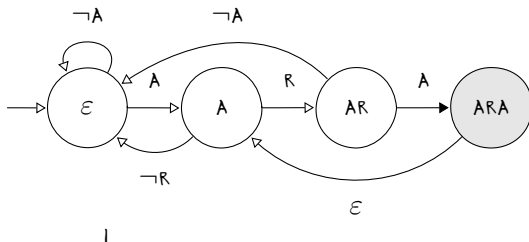


$R$

0 1 2 3 4 5 6 7 8 9 10 11 12 13

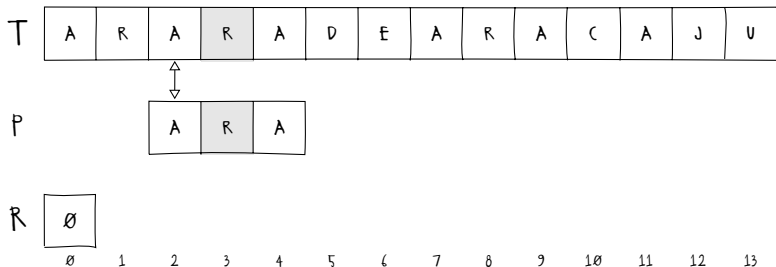
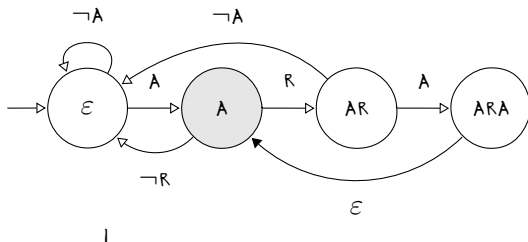
# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
- ▶ Buscando  $P = ara$  em  $T = araradearacaju$



# Busca em cadeias

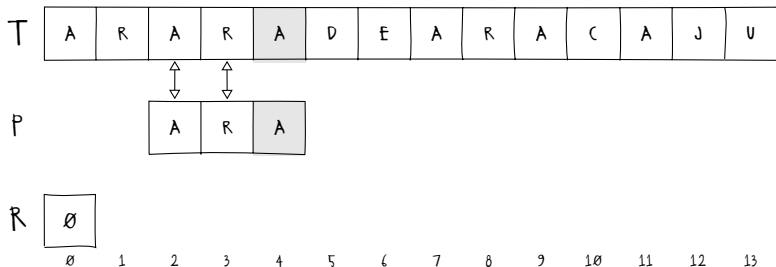
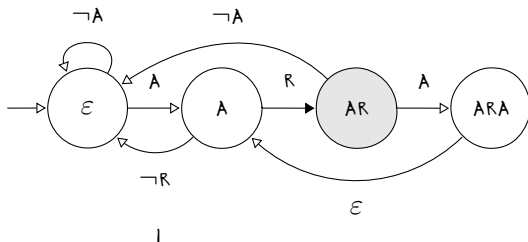
- ▶ Knuth-Morris-Pratt (KMP)
- ▶ Buscando  $P = ara$  em  $T = araradearacaju$





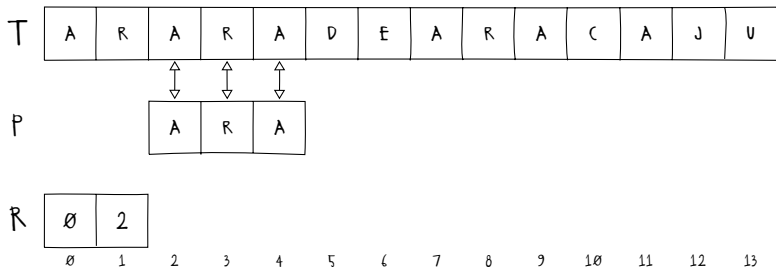
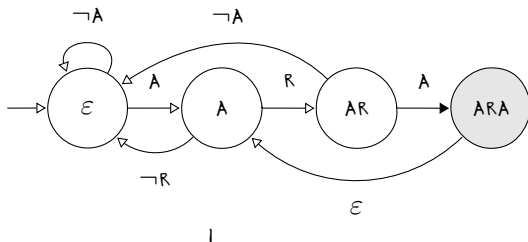
# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
- ▶ Buscando  $P = ara$  em  $T = araradearacaju$



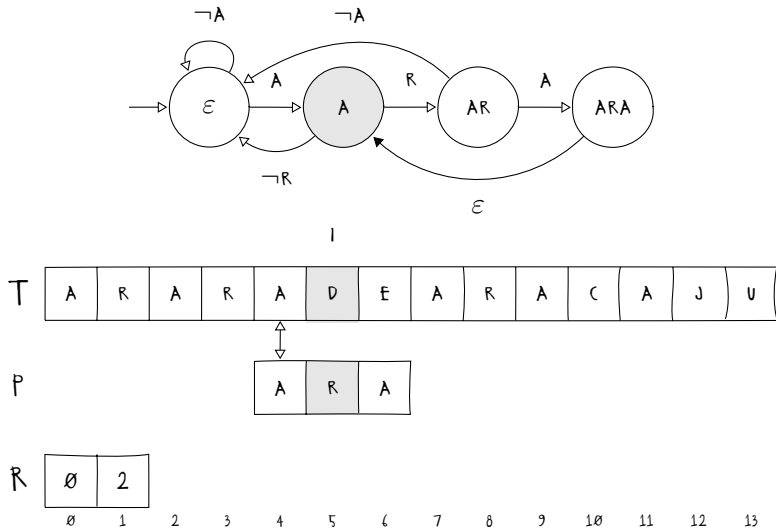
# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
- ▶ Buscando  $P = ara$  em  $T = araradearacaju$



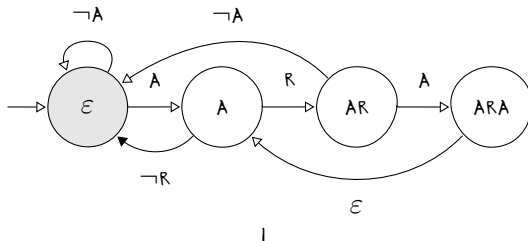
# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
- ▶ Buscando  $P = ara$  em  $T = araradearacaju$



# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
- ▶ Buscando  $P = ara$  em  $T = araradearacaju$



T    A   R   A   R   A   D   **E**   A   R   A   C   A   J   U

P

**A**   R   A

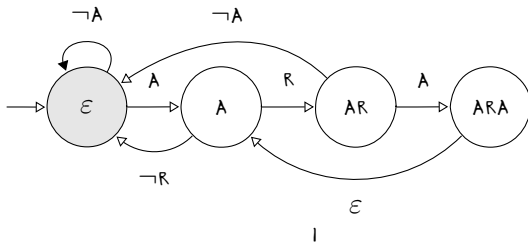
R

0	2
---	---

0   1   2   3   4   5   6   7   8   9   10   11   12   13

# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
  - ▶ Buscando  $P = ara$  em  $T = araradearacaju$



$T$     A   R   A   R   A   D   E   **A**   R   A   C   A   J   U

$P$

**A**   R   A

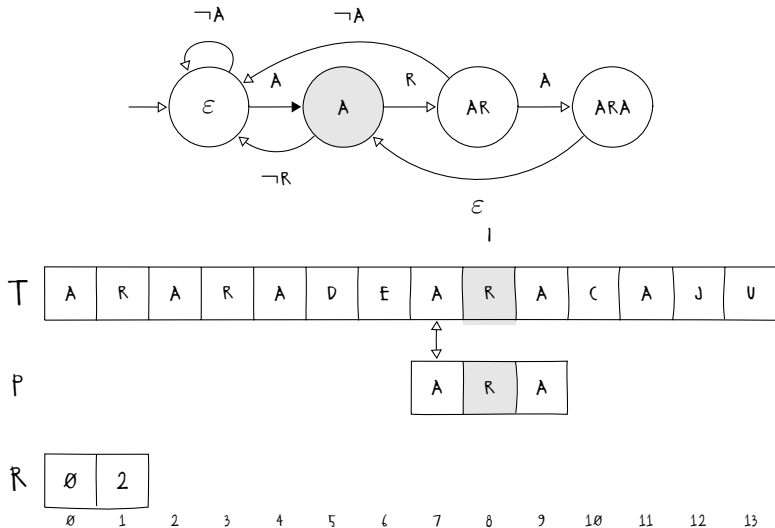
$R$

$\emptyset$	2
-------------	---

$\emptyset$    1   2   3   4   5   6   7   8   9   10   11   12   13

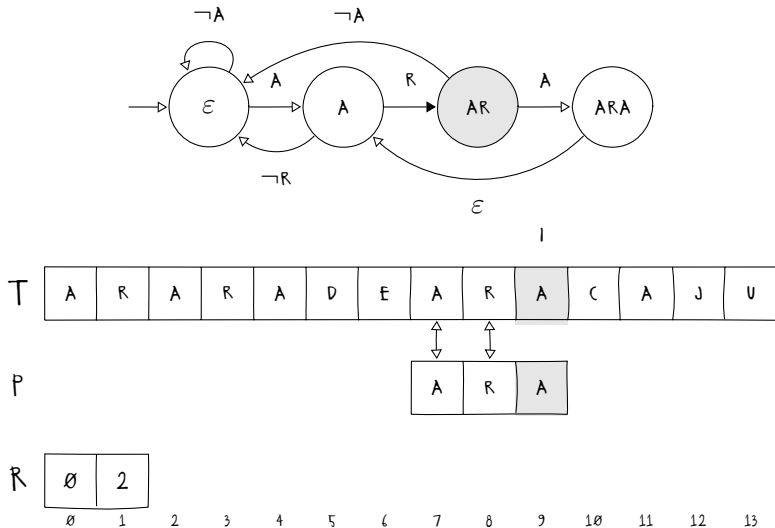
# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
- ▶ Buscando  $P = ara$  em  $T = araradearacaju$



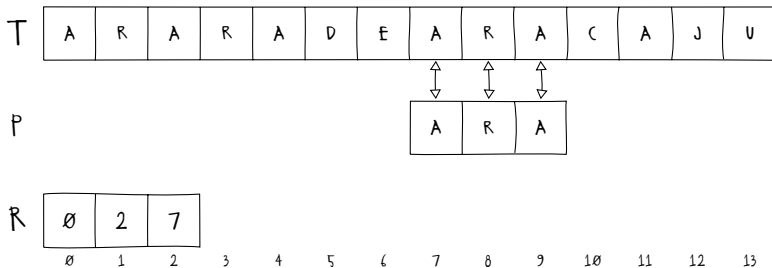
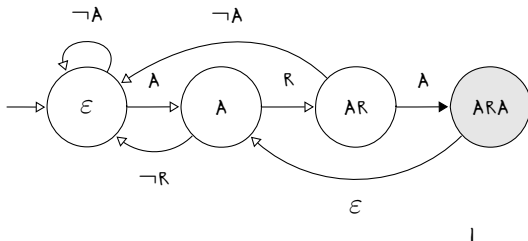
# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
- ▶ Buscando  $P = ara$  em  $T = araradearacaju$



# Busca em cadeias

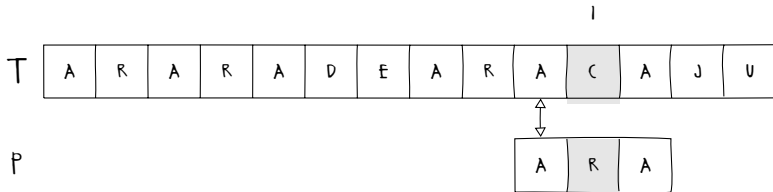
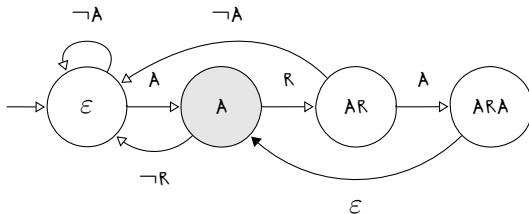
- ▶ Knuth-Morris-Pratt (KMP)
- ▶ Buscando  $P = ara$  em  $T = araradearacaju$





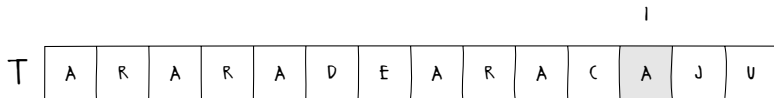
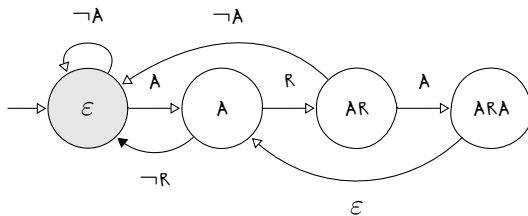
# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
  - ▶ Buscando  $P = ara$  em  $T = araradearacaju$



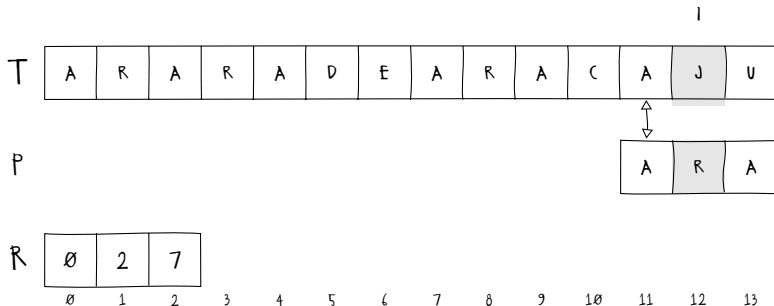
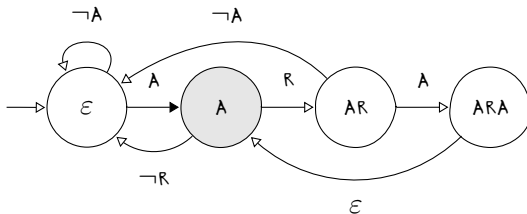
# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
  - ▶ Buscando  $P = ara$  em  $T = araradearacaju$



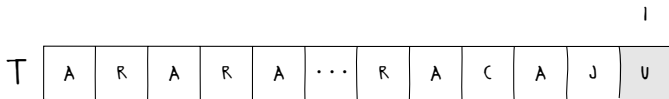
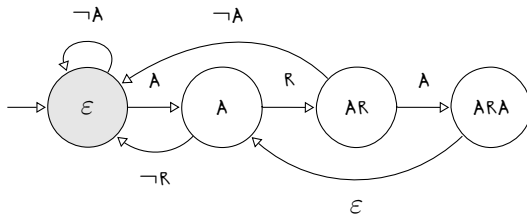
# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
- ▶ Buscando  $P = ara$  em  $T = araradearacaju$



# Busca em cadeias

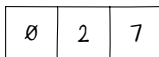
- ▶ Knuth-Morris-Pratt (KMP)
- ▶ Buscando  $P = ara$  em  $T = araradearacaju$



$P$



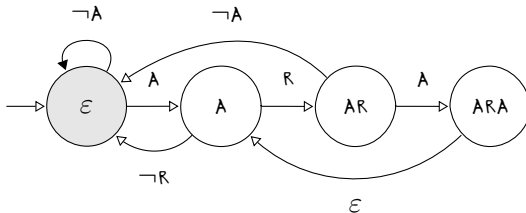
$R$



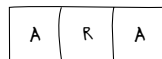
0 1 2 3 4 ... 8 9 10 11 12 13

# Busca em cadeias

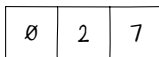
- ▶ Knuth-Morris-Pratt (KMP)
- ▶ Buscando  $P = ara$  em  $T = araradearacaju$



P



R



0 1 2 3 4 ... 8 9 10 11 12 13

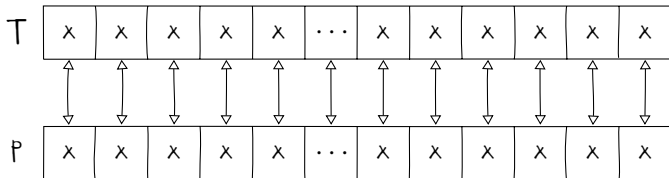
# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
  - ▶ Por que a busca em cadeias é  $O(n)$  e não  $O(n \times m)$ ?

# Busca em cadeias

## ► Knuth-Morris-Pratt (KMP)

- Por que a busca em cadeias é  $O(n)$  e não  $O(n \times m)$ ?



SÓ OCORRE UMA TRANSIÇÃO NO AUTÔMATO POR COMPARAÇÃO

# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)

- ▶ Por que a busca em cadeias é  $O(n)$  e não  $O(n \times m)$ ?

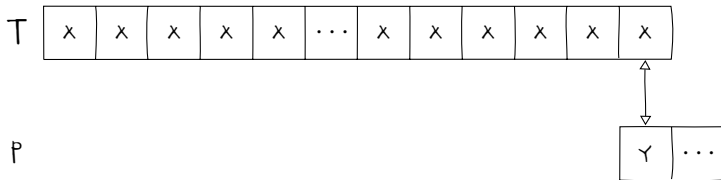


SÓ OCORRE UMA TRANSIÇÃO NO AUTÔMATO POR COMPARAÇÃO



# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
  - ▶ Por que a busca em cadeias é  $O(n)$  e não  $O(n \times m)$ ?

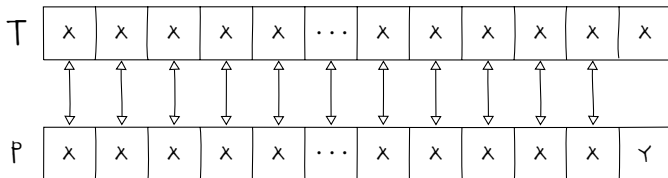


SÓ OCORRE UMA TRANSIÇÃO NO AUTÔMATO POR COMPARAÇÃO

# Busca em cadeias

## ► Knuth-Morris-Pratt (KMP)

- Por que a busca em cadeias é  $O(n)$  e não  $O(n \times m)$ ?



PODEM OCORRER ATÉ  $N$  TRANSIÇÕES PARA A CADEIA COMPLETA,  
OU SEJA, SÓ RETORNA A QUANTIDADE DE PASSOS QUE AVANÇOU

# Busca em cadeias

- ▶ Knuth-Morris-Pratt (KMP)
  - ▶ Análise de complexidade
    - ▶ Espaço  $O(n + m)$
    - ▶ Tempo  $\Theta(m) + \Theta(n) = \Theta(n + m)$

# Exemplo

- ▶ Aplique os algoritmos de busca em cadeias para encontrar o padrão 111000 na sequência binária  
10111000110111100010101100011100001101101111
  - ▶ Execute passo a passo a busca na cadeia
  - ▶ Descreva seu princípio de funcionamento e as vantagens com relação aos algoritmos já vistos

# Exercício

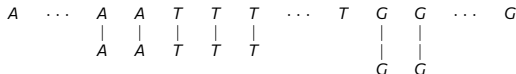
- ▶ A empresa de biotecnologia Poxim Tech possui um sistema de diagnóstico para doenças genéticas, comparando a sequência de DNA com genes conhecidos
  - ▶ A sequência de DNA é composta somente pelos símbolos *A*, *C*, *G* e *T* para codificação dos genes
  - ▶ Uma doença genética pode possuir vários genes associados, com sequências de tamanho entre 100 até 1000 símbolos, denotada exclusivamente por letras maiúsculas e números com tamanho entre 4 e 8 caracteres
  - ▶ Para tratar os efeitos da mutação nos genes que alteram sua codificação, é feita a busca por combinações que possuam o tamanho mínimo de subcadeia, com pelo menos 90% de compatibilidade para manifestação da doença
  - ▶ No diagnóstico será calculada a probabilidade de manifestação da doença, de acordo com a quantidade de genes detectados na sequência de DNA

# Exercício

- ▶ Diagnóstico da doença CTRLF4 com genes *AATTTGGCCC* e *GGGGGGGGGG*
  - ▶ DNA: *AAAAAAAAAATTTTTTTTTTTGGGGGGGGG*
  - ▶ Tamanho da subcadeia: 3

# Exercício

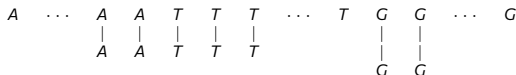
- ▶ Diagnóstico da doença CTRLF4 com genes *AATTTGGCCC* e *GGGGGGGGGG*
  - ▶ DNA: *AAAAAAAAAATTTTTTTTTTTGGGGGGGGGG*
  - ▶ Tamanho da subcadeia: 3



*AATTTGGCCC* : 5 combinações = 50%

# Exercício

- ▶ Diagnóstico da doença CTRLF4 com genes *AATTTGGCCC* e *GGGGGGGGGG*
  - ▶ DNA: *AAAAAAAAAATTTTTTTTTTTGGGGGGGGGG*
  - ▶ Tamanho da subcadeia: 3



*AATTTGGCCC* : 5 combinações = 50%

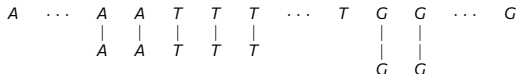


*GGGGGGGGGG* : 9 combinações = 90%



# Exercício

- ▶ Diagnóstico da doença CTRLF4 com genes *AATTTGGCCC* e *GGGGGGGGGG*
  - ▶ DNA: *AAAAAAAAAATTTTTTTTTTTGGGGGGGGGG*
  - ▶ Tamanho da subcadeia: 3



*AATTTGGCCC* : 5 combinações = 50%



*GGGGGGGGGG* : 9 combinações = 90%

Chance de 50% de ocorrência da doença CTRLF4

# Exercício

## ► Formato do arquivo de entrada

- $[ \#Tamanho\ da\ subcadeia ]$
- $[ B_0 \dots B_{N-1} ]$
- $[ \#Número\ de\ doenças ]$
- $[ Código_0 ] [ \#Genes_0 ] [ G_{0_0} ] \dots [ G_{0_{i-1}} ]$
- $\vdots$
- $[ Código_{M-1} ] [ \#Genes_{M-1} ] [ G_{M-1_0} ] \dots [ G_{M-1_{j-1}} ]$

```
1 3
2 AAAATTTTCGTTAAATTTGAACATAGGGATA
3 4
4 ABCDE 3 AAA AAT AAAG
5 XY1WZ2AB 1 TTTTTTGGGG
6 H1N1 4 ACTG AACCGGTT AATAAT AAAAAAAGA
7 HUEBR 1 CATAGGGATT
```

# Exercício

- ▶ Formato do arquivo de saída
  - ▶ É feita a ordenação estável em ordem decrescente dos resultados, utilizando como critério de ordenação a probabilidade de ocorrência da doença e fazendo o arredondamento dos percentuais para fins de comparação

1	XY1WZ2AB → 100%
2	HUEBR → 100%
3	ABCDE → 67%
4	H1N1 → 25%