

Algoritmos em Strings

R. Rossetti, A. P. Rocha, J. Pascoal Faria
FEUP, MIEIC, CAL, 2013/2014

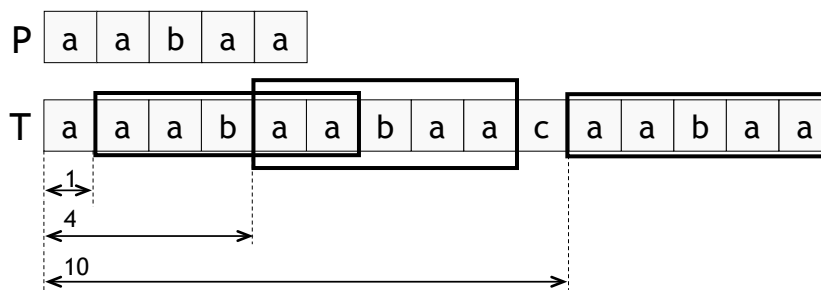
Índice

- Pesquisa exata (*string matching*)
- Pesquisa aproximada (*approximate string matching*)
- Outros problemas

Pesquisa exata (*string matching*)

Problema

- Encontrar todas as ocorrências de um padrão P num texto T
 - P e T são cadeias de caracteres
 - Ocorrências são definidas pela deslocação em relação ao início do texto
 - Ocorrências podem ser sobrepostas



Algoritmos

■ Algoritmo *naive*

- Para cada deslocamento possível, compara desde o início do padrão
- Ineficiente se o padrão for comprido: $O(|P| \cdot |T|)$

■ Algoritmo baseado em autômato finito

- Pré-processamento: gerar autômato finito correspondente ao padrão
- Permite depois analisar o texto em tempo linear $O(|T|)$, pois cada carácter só precisa de ser processado uma vez
- Mas tempo e espaço requerido pelo pré-processamento pode ser elevado: $O(|P| \cdot |\Sigma|)$, em que $|\Sigma|$ é o tamanho do alfabeto

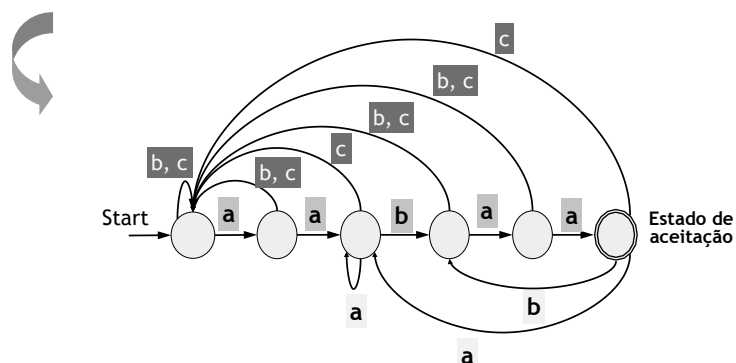
■ Algoritmo de Knuth-Morris-Pratt

- Efetua um pré-processamento do padrão em tempo $O(|P|)$, sem chegar a gerar explicitamente um autômato, seguido de processamento do texto em $O(|T|)$, dando total $O(|T| + |P|)$

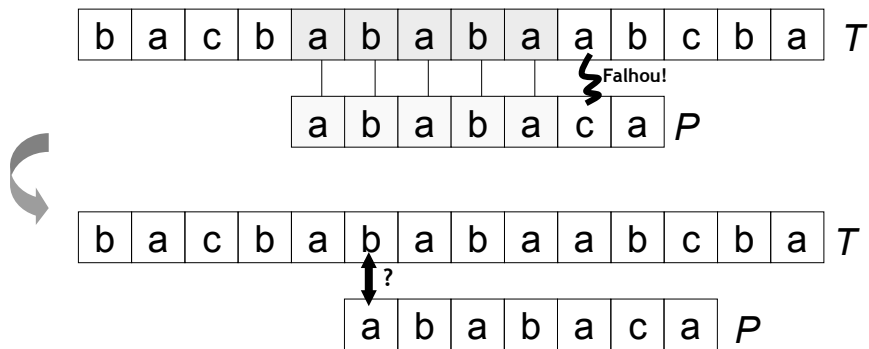
Autômato finito correspondente ao padrão

P a a b a a

$\Sigma = \{a, b, c\}$

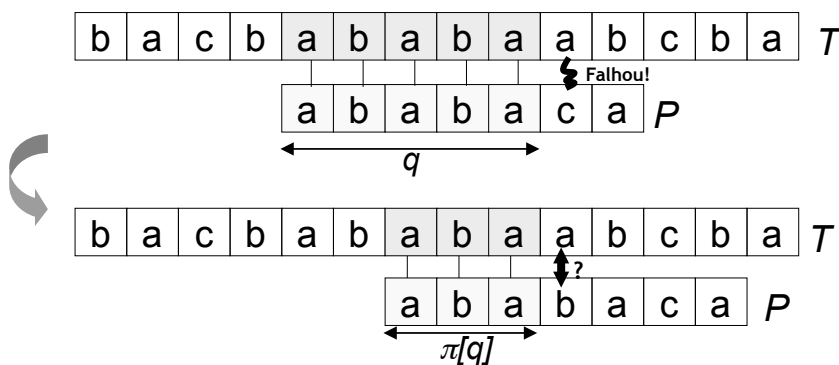


Algoritmo *naive*



Desloca-se o padrão uma casa para a direita e recomeça-se a comparação do início do padrão! Ineficiente: $O(|P| \cdot |T|)$

Algoritmo de Knuth-Morris-Pratt



Desloca-se o padrão para a direita de uma forma que permite continuar a comparação na mesma posição do texto! Evita comparações inúteis!

Deslocamento é determinado por uma função $\pi[q]$ calculada numa fase de pré-processamento do padrão!

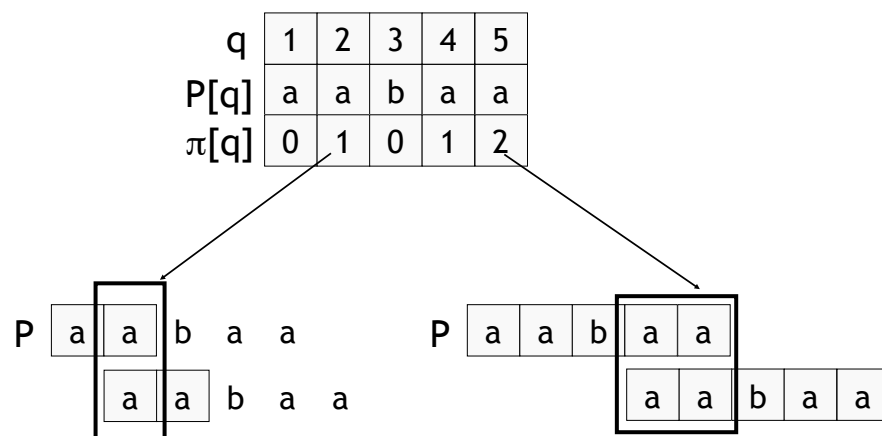
Pré-processamento do padrão

- Compara-se o padrão com deslocações do mesmo, para determinar a **função prefixo**

$$\pi[q] = \max \{k: 0 \leq k < q \text{ e } P[1..k] = P[(q-k+1)..q] \}$$

- $q = 1, \dots, |P|$
- $P[i..j]$ - substring entre índices i e j
- Índices a começar em 1
- $\pi[q]$ é o comprimento do maior prefixo de P que é um sufixo próprio do prefixo de P de comprimento q

Pré-processamento do padrão



Pseudo-código

KMP-MATCHER(T, P)

```
1   $n \leftarrow \text{length}[T]$ 
2   $m \leftarrow \text{length}[P]$ 
3   $\pi \leftarrow \text{COMPUTE-PREFIX-FUNCTION}(P)$ 
4   $q \leftarrow 0$  ▷ Number of characters matched.
5  for  $i \leftarrow 1$  to  $n$  ▷ Scan the text from left to right.
6      do while  $q > 0$  and  $P[q + 1] \neq T[i]$ 
7          do  $q \leftarrow \pi[q]$  ▷ Next character does not match.
8      if  $P[q + 1] = T[i]$ 
9          then  $q \leftarrow q + 1$  ▷ Next character matches.
10     if  $q = m$  ▷ Is all of  $P$  matched?
11         then print "Pattern occurs with shift"  $i - m$ 
12          $q \leftarrow \pi[q]$  ▷ Look for the next match.
```

Pseudo-código

COMPUTE-PREFIX-FUNCTION(P)

```
1   $m \leftarrow \text{length}[P]$ 
2   $\pi[1] \leftarrow 0$ 
3   $k \leftarrow 0$ 
4  for  $q \leftarrow 2$  to  $m$ 
5      do while  $k > 0$  and  $P[k + 1] \neq P[q]$ 
6          do  $k \leftarrow \pi[k]$ 
7      if  $P[k + 1] = P[q]$ 
8          then  $k \leftarrow k + 1$ 
9       $\pi[q] \leftarrow k$ 
10 return  $\pi$ 
```

Eficiência do algoritmo Knuth-Morris-Pratt *

- KMP-MATCHER (sem incluir COMPUTE-PREFIX-FUNCTION) *
 - Eficiência depende do nº de iterações do ciclo “while” interno
 - Dado que $0 \leq \pi[q] < q$, cada vez que a instrução 7 é executada, o valor de q é decrementado de pelo menos 1, sem nunca chegar a ser negativo
 - Dado que o valor de q começa em 0 e só é incrementado no máximo n vezes (+1 de cada vez, na linha 9), o nº máximo de vezes que pode ser decrementado (nas linhas 7 e 12) é também n
 - ⇒ Nº máximo de iterações do ciclo “while” interno (no conjunto de todas as iterações do ciclo “for” externo) é n
 - ⇒ Tempo de execução da rotina é $O(n)$, i.e., $O(|T|)$
- COMPUTE-PREFIX-FUNCTION
 - Seguindo o mesmo raciocínio, tempo de execução é $O(m)$, i.e., $O(|P|)$
- Total: $O(n+m)$, isto é, $O(|T| + |P|)$