
Fortgeschrittene Algorithmen

Randomisierte Algorithmen

```
findingA_LV(array A, n)
begin
  repeat
    Randomly select one element out of n elements.
  until 'a' is found
end
```

```
findingA_MC(array A, n, k)
begin
  i := 0
  repeat
    Randomly select one element out of n elements.
    i := i + 1
  until i = k or 'a' is found
end
```

Programmieraufgaben

- Pi Raten
 - Textvergleich
 - Bildvergleich
 - Vermessung der Fläche eines Landes
- ➔ Vorstellung

Zusammenfassung - Randomisierung

- Las Vegas
 - Liefert immer ein richtiges Ergebnis
 - Variiert mit der Wahl des Zufallsereignisses / Zufallszahl

- Monte Carlo
 - Können auch (manchmal) falsche Ergebnisse liefern
 - Einseitiger Fehler:
 - Wenn die Wahrscheinlichkeit für eine der beiden Antwortmöglichkeiten (Ja/Nein) gleich Null ist und für die andere größer Null
 - Beidseitiger Fehler:
 - Wenn für die beiden Antwortmöglichkeiten (Ja/Nein) die Wahrscheinlichkeit für eine falsche Antwort größer Null ist

String-Matching

- Ziel:
 - Finden (aller) Vorkommen eines Musters in einem Text
- Beispiele → Tafel

String-Matching (etwas formaler)

String-Matching-Problem

- Alphabet: $\Sigma = \{0, 1\}$ oder $\Sigma = \{a, b, \dots, z\}$
- Text in einem Feld $T[1..n]$
- Textmuster in einem Feld $P[1..m]$
- $m \leq n$
- Felder P und T werden (häufig) als aus Zeichen bestehende Strings bezeichnet

a	b	c	a	b	a	a	b	c	a	b	a	c
			a	b	a	a						

String-Matching (etwas formaler)

String-Matching-Problem

- P kommt mit der Verschiebung s in T vor:
- Das Muster P tritt im Text an der Position s+1 beginnend auf
- Wenn für $1 \leq j \leq m$ gilt: $T[s + j] = P[j]$

a	b	c	a	b	a	a	b	c	a	b	a	c
			a	b	a	a						

- Wenn P mit einer Verschiebung s in T vorkommt:

➔ s ist eine gültige Verschiebung

Sonst: s ist eine ungültige Verschiebung

Naiver String-Matcher

```
n = T.Länge
m = P.Länge
For s = 0 to n-m
    if (P[1..m] == T[s+1..s+m])
        print „gültige Verschiebung: “ s
```

Viel Spaß beim Coden

Naiver String-Matcher

```
n = T.Länge
m = P.Länge
For s = 0 to n-m
    if (P[1..m] == T[s+1..s+m])
        print „gültige Verschiebung: “ s
```

- Laufzeit im schlechtesten Fall?

Naiver String-Matcher

$n = T.\text{Länge}$

$m = P.\text{Länge}$

For $s = 0$ to $n-m$

 if $(P[1..m] == T[s+1..s+m])$

 print „gültige Verschiebung: “ s

- Laufzeit im schlechtesten Fall?

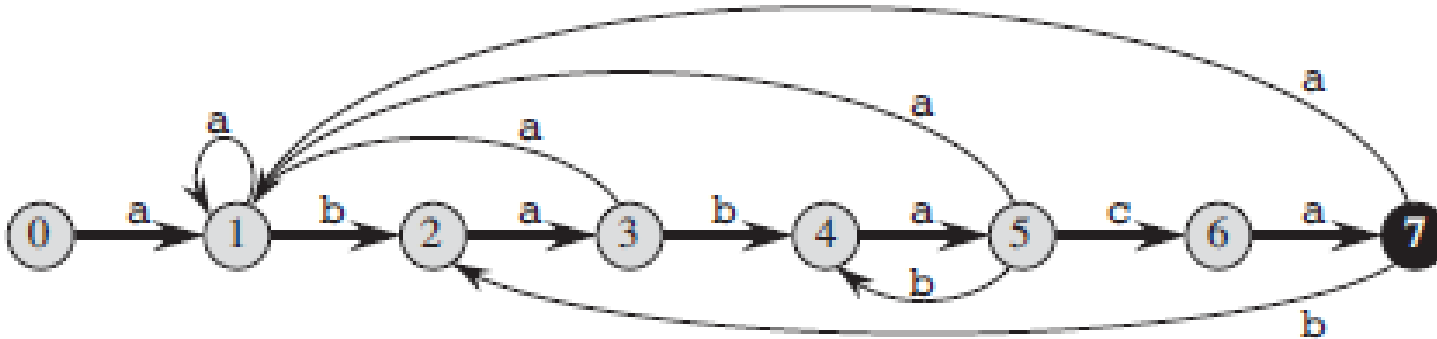
$T: a^n$

$P: a^m$

→ $O((n-m+1) \cdot m)$

Keine Vorverarbeitung: Laufzeit = Matchingzeit

String-Matching Automat



String-Matching Automat

`n = T.Länge`

`q = 0`

`For i = 1 to n`

`q = $\delta(q, T[i])$`

`if (q == m)`

`print „gültige`

`Verschiebung: “ i - m`

■ Laufzeit?

state	input		
	a	b	c
0	1	0	0
1	1	2	0
2	3	0	0
3	1	4	0
4	5	0	0
5	1	4	6
6	7	0	0
7	1	2	0

String-Matching Automat erstellen

$\delta(q, T[i])$: Definieren / Berechnen

Notation: $P_n = P[1..n], P_0 = \varepsilon$

$T \sqsupset M \rightarrow T$ ist Präfix von M

Definition (Hilfsfunktion) :

$$\sigma(x) = \max\{k: P_k \sqsupset x\}$$

Wir definieren:

$$\delta(q, a) = \sigma(P_q a)$$

String-Matching Automat erstellen

$$\delta(q, x) = \sigma(P_q x)$$

Beispiel: $P = \text{anax}$

$$\delta(0, a) = \sigma(P_0 a) = \sigma(a) = \max\{k: P_k \sqsupseteq a\} = 1$$

String-Matching Automat erstellen

$m = P.\text{Länge}$

For $q = 0$ to m

For jedes Zeichen $a \in \Sigma$

$k = \min(m+1, q+2)$

repeat

$k = k - 1$

until $P_k \sqsupset P_q a$

$\delta(q, a) = k$

Return δ

String-Matching Automat erstellen

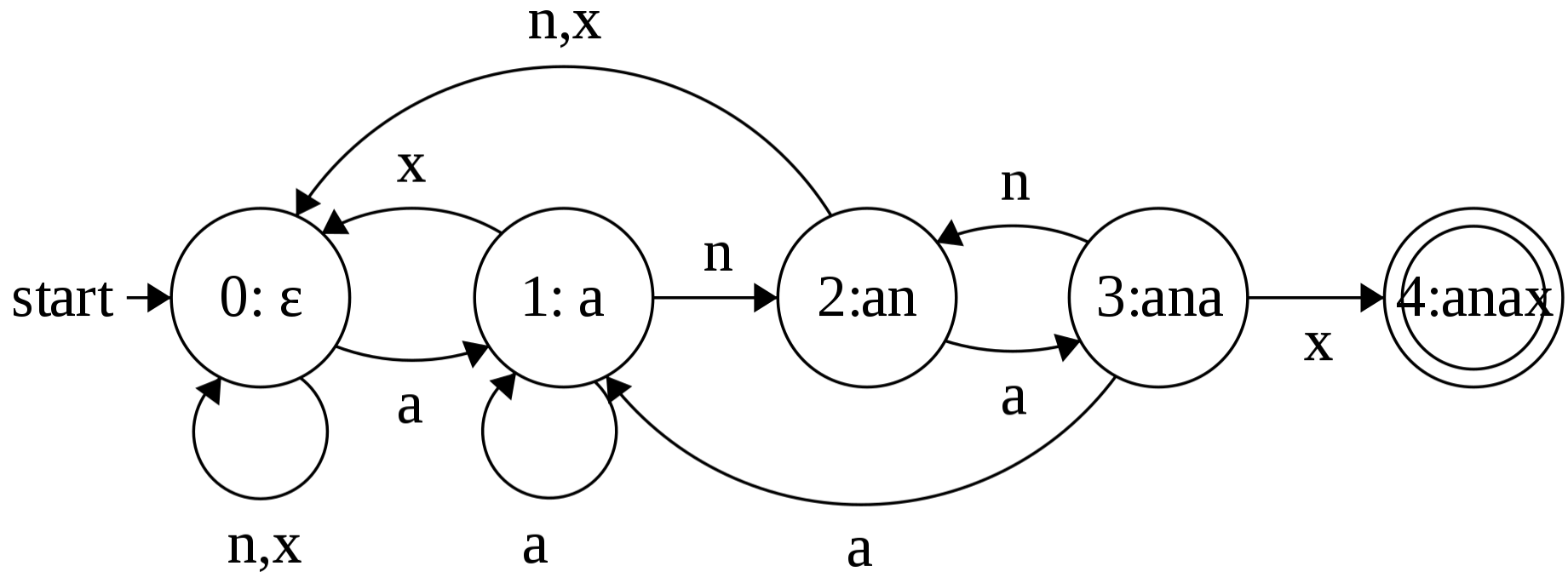
```
m = P.Länge
For q = 0 to m
  For jedes Zeichen a  $\in \Sigma$ 
    k=min(m+1, q+2)
    repeat
      k=k - 1
    until  $P_k \supset P_q a$ 
     $\delta(q, a) = k$ 
Return  $\delta$ 
```

Erstellen wir einen Automaten für „anax“

String-Matching Automat erstellen

Erstellen wir einen Automaten für „anax“

String-Matching Automat erstellen



String-Matching Automat erstellen

```
m = P.Länge
For q = 0 to m
    For jedes Zeichen a  $\in \Sigma$ 
        k=min(m+1, q+2)
        repeat
            k=k - 1
        until  $P_k \supset P_q a$ 
         $\delta(q, a) = k$ 
Return  $\delta$ 
```

Viel Spaß beim Coden