



Teilwortsuche: Text $a_0 a_1 a_2 \dots a_{m-1} \in \Sigma^*$
 Muster $b_0 b_1 \dots b_{n-1}$

Kommt das Muster im Text vor? An welchen Stellen?

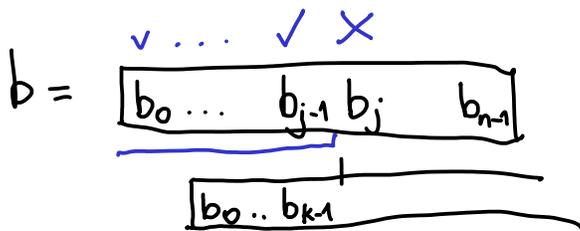
$a = 0110100110010110$
 $\quad \checkmark \checkmark \times \checkmark \checkmark \checkmark \times$
 $\quad \quad \times \times \quad \times$
 $b = 01010$

naiver Algorithmus
 Laufzeit $O(mn)$.
 $\quad \quad \quad \uparrow$
 $\quad \quad \quad m-n+1$

$a =$ kein steifes beinstellen

$b =$ einstein
 $f(7) = 2$
 $\quad \quad \quad \begin{array}{c} \text{einstein} \\ \underline{01} \quad (2) \quad 3 \end{array}$

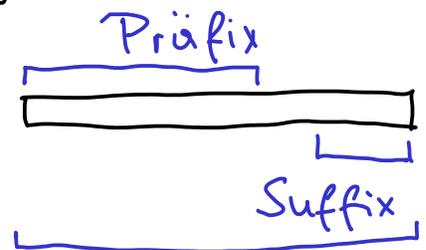
Wenn HIER der erste Unterschied festgestellt wird, dann kann man das Muster um 5 Positionen weiter schieben.



Verschiebefunktion $f(j) :=$
 Länge k des längsten Präfixes
 $b_0 \dots b_{k-1}$ ($0 \leq k < j$)

der mit einem Suffix von $b_0 \dots b_{j-1}$ übereinstimmt:
 $(1 \leq j \leq n)$

$b_{j-k} b_{j-k+1} \dots b_{j-1} = b_0 b_1 \dots b_{k-1}$



- Bei b_j wurde der Unterschied gefunden.
 ($j=n$: Das vollständige Muster wurde gefunden)
- $k=f(j)$: Als nächstes wird b_k verglichen (mit der gleichen Stelle im Text)

a = keinsteifes beinstellen

b =

einstein
j=6
k=1

einstein

j = 0 1 2 3 4 5 6 7 8
f[j] = - 0 0 0 0 0 1 2 3

Algorithmus von Knuth, Morris und Pratt (1977)

i = 0; j = 0

while True:

if a[i] == b[j]:

i += 1

j += 1

if j == n: return i - n

if i == m: return None

else:

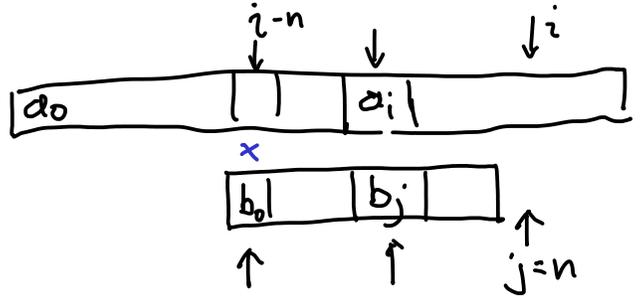
if j > 0:

j = f[j]

else:

i = i + 1

if i == m: return None



i ↑

(2i - j) ↑
≤ 2m

(i - j) ↑

Laufzeit O(m)

Berechnung der Verschiebefunktion

f(j) - 0 0 0 1 2 3 4 0 0 1 2 1 2 3 4 5 6 7 5 6 0 0 0 1 2
eineinerleieeineineinszwei

~~eineinerleieeineineinszwei~~

f(j+1) = f(j) + 1

eineinerleieeineineinszwei

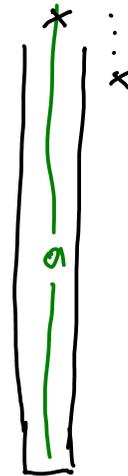
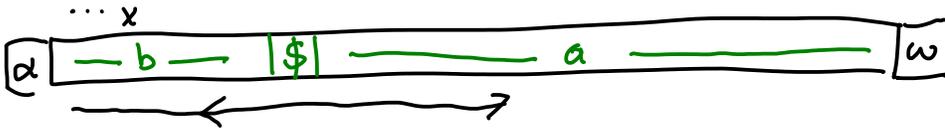
eineinerleieeineineinszwei

Laufzeit O(m)

eineinerleieeineineinszwei

Das Teilwortproblem kann in O(m+n) Zeit gelöst werden.

Deterministische Zwei-Wege-Kellerautomaten



SATZ: Das Ergebnis eines 2Wege-det. KA kann in linearer Zeit berechnet werden.

⇒ Es gibt einen linearzeitalgorithmus für das Teilwortproblem.

Teilwortsuche naiver Algorithmus Definition der Verschiebefunktion Algorithmus von Knuth-Morris-Pratt Hintergrund und Entstehung des Algorithmus Algorithmus von Boyer und Moore

- Speichere für jeden Buchstaben $x \in \Sigma$ die rechteste Position im Muster, wo x vorkommt.
- Vergleiche von rechts nach links.
- Bei einem Unterschied: Bring den Buchstaben x im Text und bringe ihn mit der rechten Position von x im Muster zur Deckung, falls das eine Rechtsverschiebung bewirkt. Andernfalls schiebe um 1 nach rechts.

Laufzeit im besten Fall : $O\left(\frac{m}{n}\right) + O(|\Sigma|)$

$\ominus((m-n) \cdot n)$

- Kombination mit Ideen der Verschiebefunktion $\Rightarrow O(m+n)$ auch im schlimmsten Fall