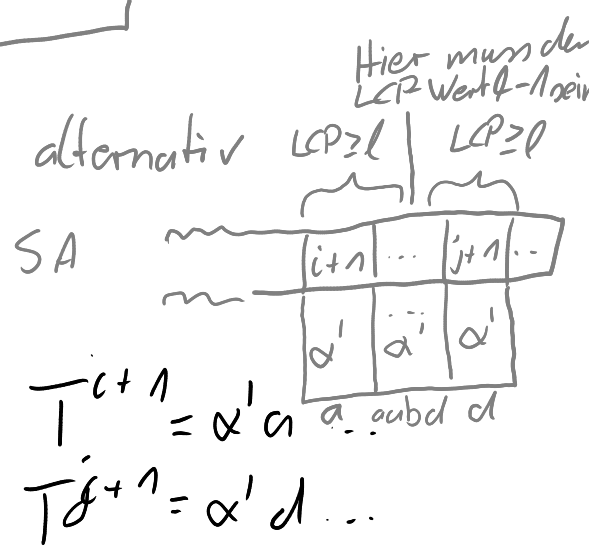
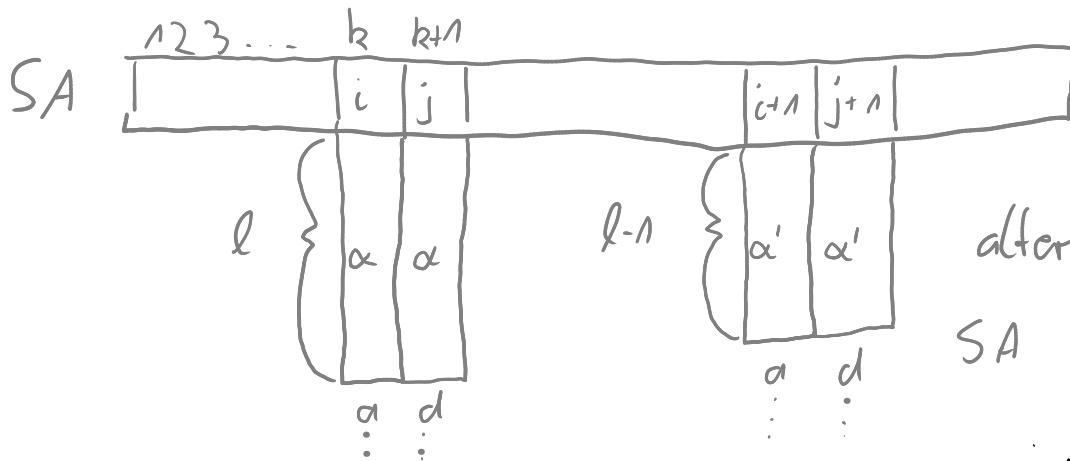


a) $\exists i \in [1, n]$ mit $H[i] = l > 0 \Rightarrow \exists j \in [1, n]$ mit $H[j] = l-1$

Seien T^i und T^j zwei Suffixe mit LCP-Wert $l > 0$.



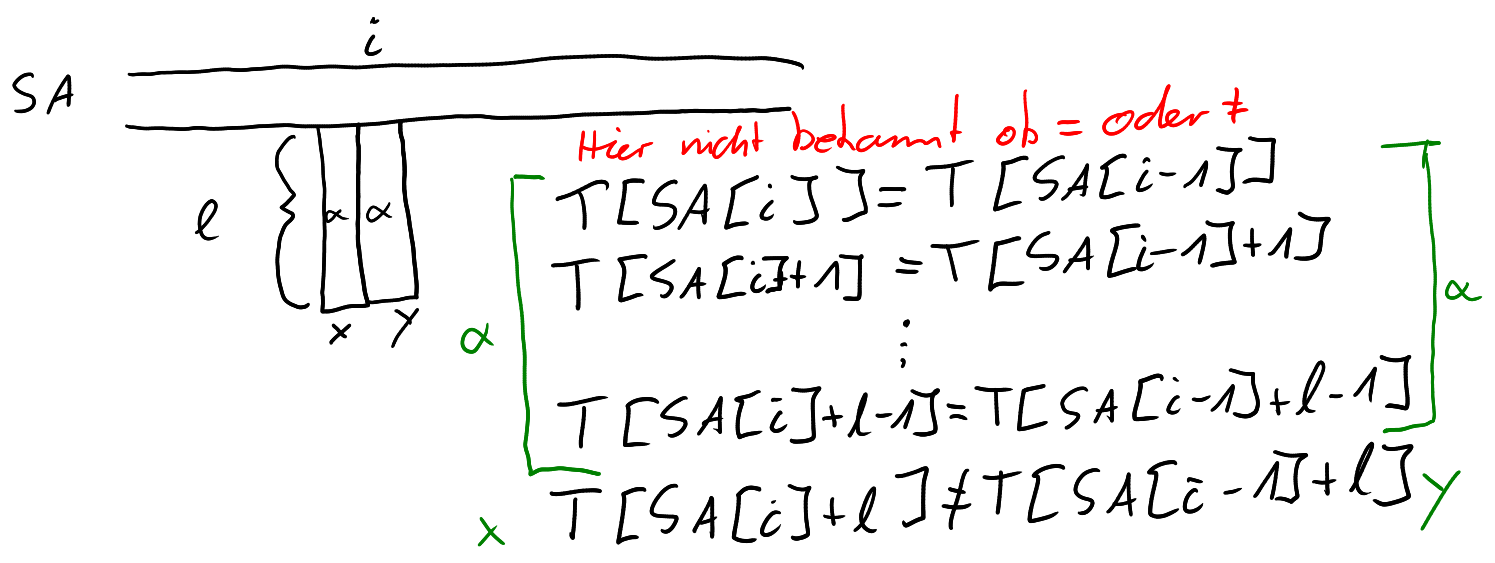
$T^i = \alpha a \dots$ und $T^j = \alpha d \dots \rightarrow$

$T^{i+1} = \alpha' a \dots$
 $T^{j+1} = \alpha' d \dots$

- Diese haben eine LCP von $l-1$.
- Auch wenn andere Suffixe im SA zwischen T^{i+1} und T^{j+1} liegen müssen diese das Präfix α' haben und mindestens eines der Suffixe hat ein Zeichen $\neq a$ oder d das auf α' folgt (aus der Mengen T^{i+1}, T^{j+1} und aller lex. dazwischenliegend Suffixe)

(b)

$\exists i \in [1, n]$ mit $H[i] = l \Leftrightarrow \exists i' \in [1, n]$ mit $H^R[i'] = l$



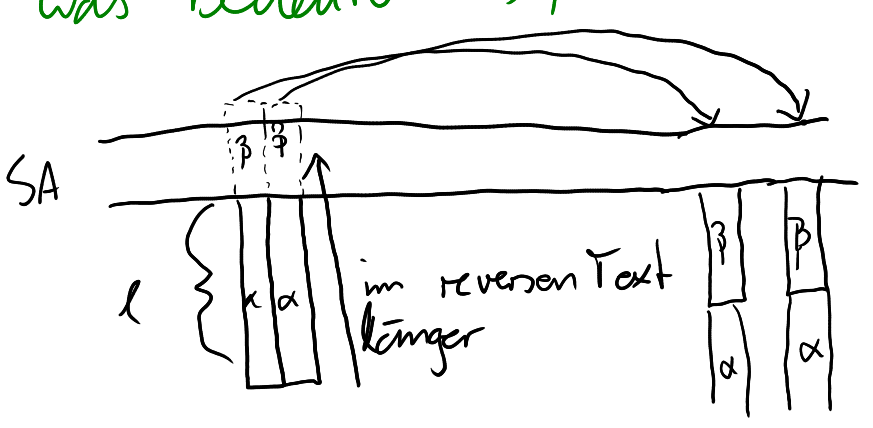
Im reversen Text existiert α in umgekehrter Reihenfolge α^R . Wir wissen aber nichts über die Zeichen vor α (und damit nach α^R).

↳ Es gibt einen LCP-Wert von mindestens l in H^R

↳ Nach (a) bedeutet das, dass es auch jeden kleineren LCP-Wert in H^R gibt.

• Die Rückrichtung funktioniert analog ($T^{RR} = T$)

Was bedeutet es, wenn der Wert größer ist

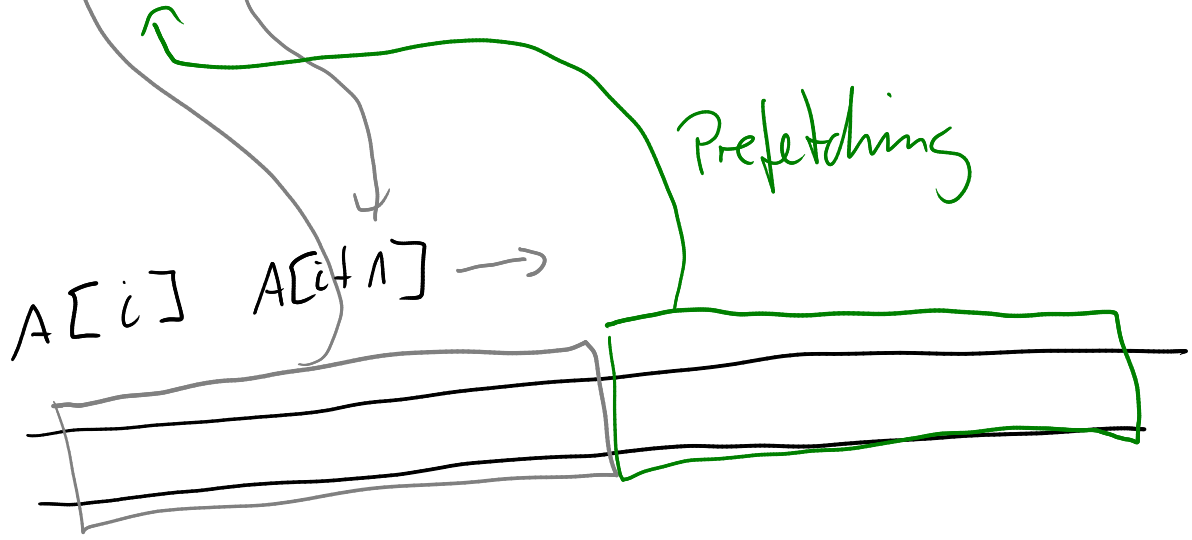
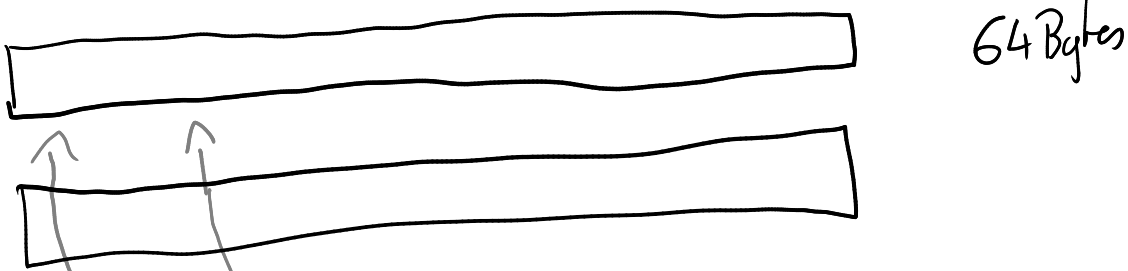


Cache-Miss

L1
L2
L2

↓ langsamer & größer

Cache-Lines



Infos zur Latenz

<https://gist.github.com/hellerbarde/2843375>

?-Operator

$A ? B : D$

↔ syntaktisch

if A
then B
else D

„unconditional branch“

- „kann Sprung sparen“
- ↳ „bessere Anordnung des Assembler-Codes“

What every Programmer should know about memory