MI-SP-SP-6 (MI-EVY) (přednášky 2, 3 a 4)

Úplné indexování textu.

Lyndon decomposition
- primitivní slovo & osobě nejmenší lexikografické rotace

cbcbbaabaacaaaaca - |c|bc|b|b|aabaac|aaac|a

LZ - decomposition - používáno v LZ kompresi

Border Array - obsahuje délku nejdelšího borderu na každé pozici
- pokud se další nerovnají, skáču na $i = \beta(current)$
  → pokud nenajdu pokračování, zapisuji 0

$[\beta[1] = 0]$

Suffix array - nové symboly $\$ < \Sigma < \#$

string T = # abaabab $

i	-1	0	1	2	3	4	5	6	7
T[i]	#	a	b	a	a	b	a	b	$
SA[i]	7	2	5	0	3	6	1	4	-1
LCP[i]	0	0	1	2	3	0	1	2	0

SA[i]: starting pozice lexikograficky seřazených suffixů

LCP[i]: nejdelší společné ~~slovo~~ prefixy

LCP Search x = aba - hledaný string

algoritmus: $d = -1$, $f = 7$

$i = \lfloor (d+f)/2 \rfloor = 3$ - půlící pole

$L3 = T[SA[3] \dots 13] = T[3 \dots 6]$ - abab

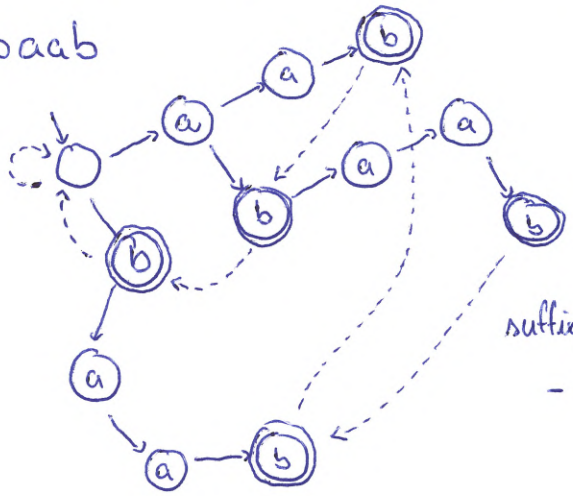
$l = LCP(L3, x) = 3$ → match - našli jsme

- přes LCP kolem pozice 3 mohou mít ostatní výskyty
  - musí být $\geq len(x)$
- pokud bych nematchnul, porovnávám $L3[l]$ s $x[l]$
  → pokud je $x[l]$ menší, pokračuji v nižší půlce (tj. $f = i$)
  → v opačném případě ve vyšší ($d = i$)

Suffix Trie - strom všech možných sufixů

- strom pro každý možný morfém, label hrany vždy jeden symbol

T = abaab

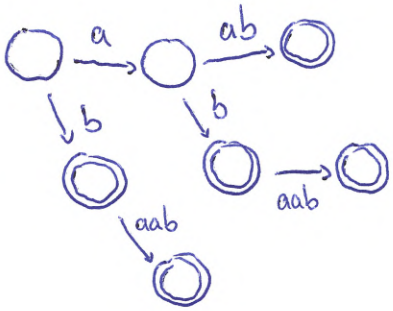


- uzle s větvením : fork
- koncové uzle vždy na konci sufixu

sufixové linky

- směřují do uzlu se sufixem o 1 kratším

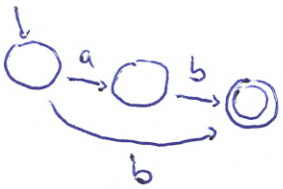
Suffix Tree - trie with nonfinal nodes of degree 1 deleted



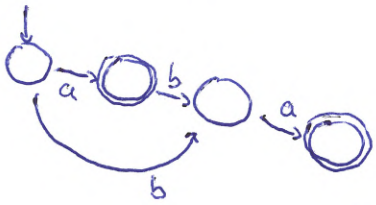
Suffix automaton - minimální automat přijímající všechny sufixy

a:

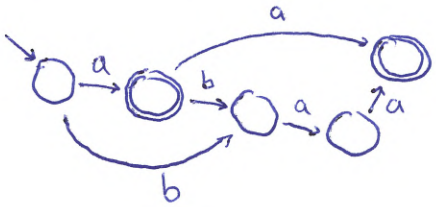
ab:



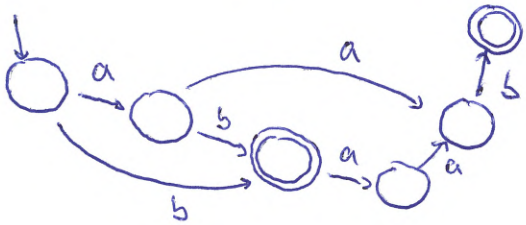
aba:



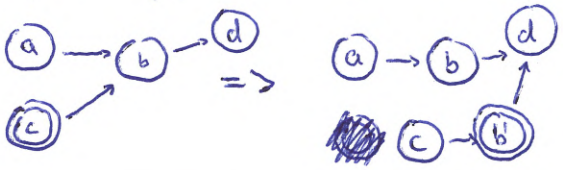
abaa :

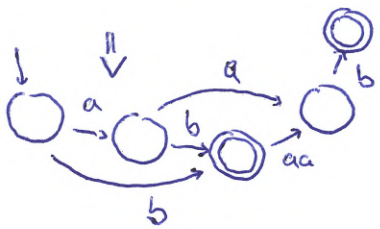


abaab :



- v každé iteraci „vynoří koncový stav" ze počátečního stavu
- pokud se vejde k každé příchozí hraně koncový stav → dc
- v opačném případě ji nutný fork





Compact suffix automaton

[implementace]