

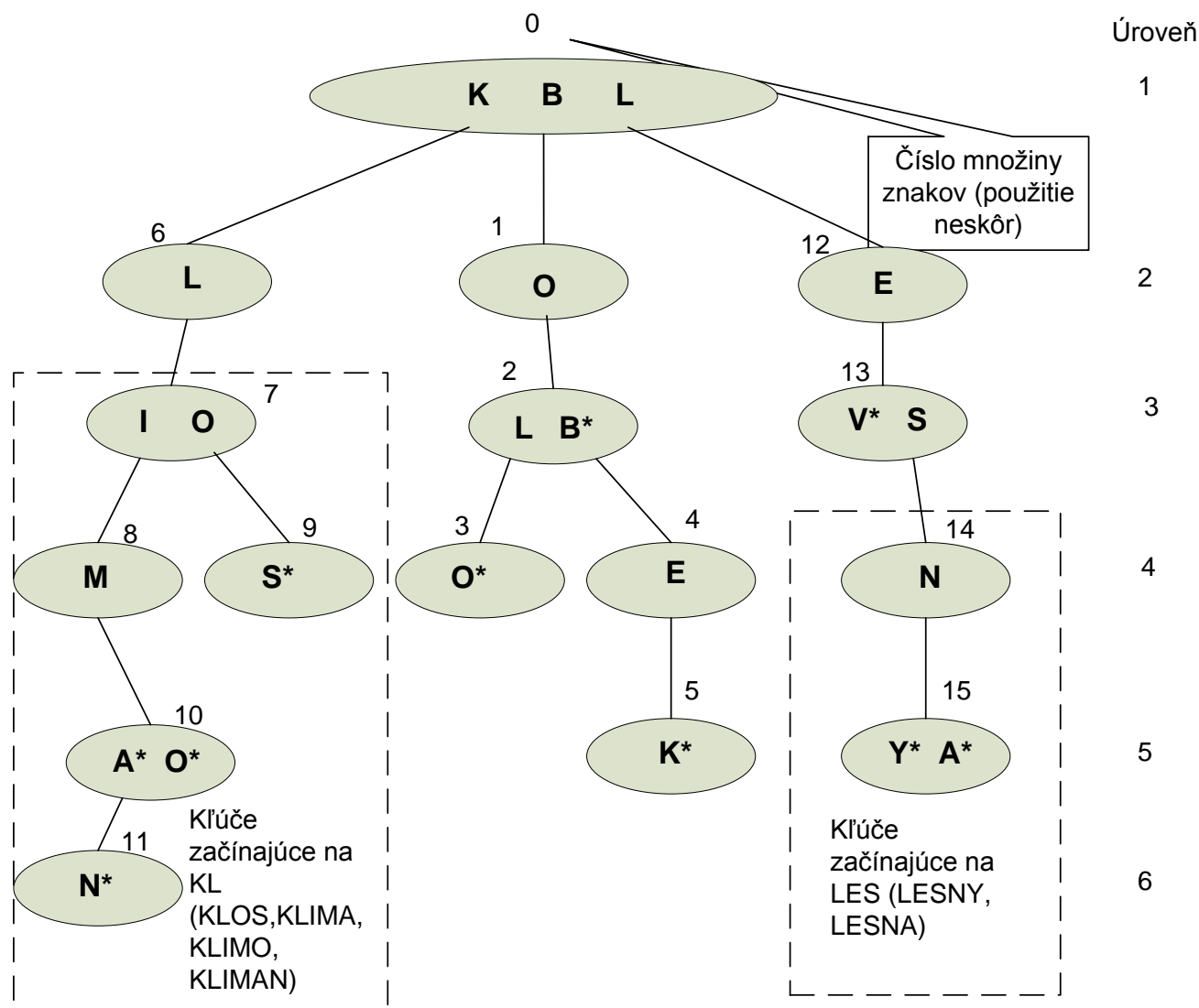
Znakový strom (trie)

- retrieval, čítaj „traj“
- pre implementáciu tabuliek
- pre tvorbu prístupových indexov k tabuľkám

Podmienka: kľúč je efektívne rozložiteľný na znaky

Abstraktná predstava pre množinu textových kľúčov:

{KLIMA, KLIMO, KLIMAN, KLOS, BOLO, BOB, BOBEK, LEV, LESNA, LESNY} :



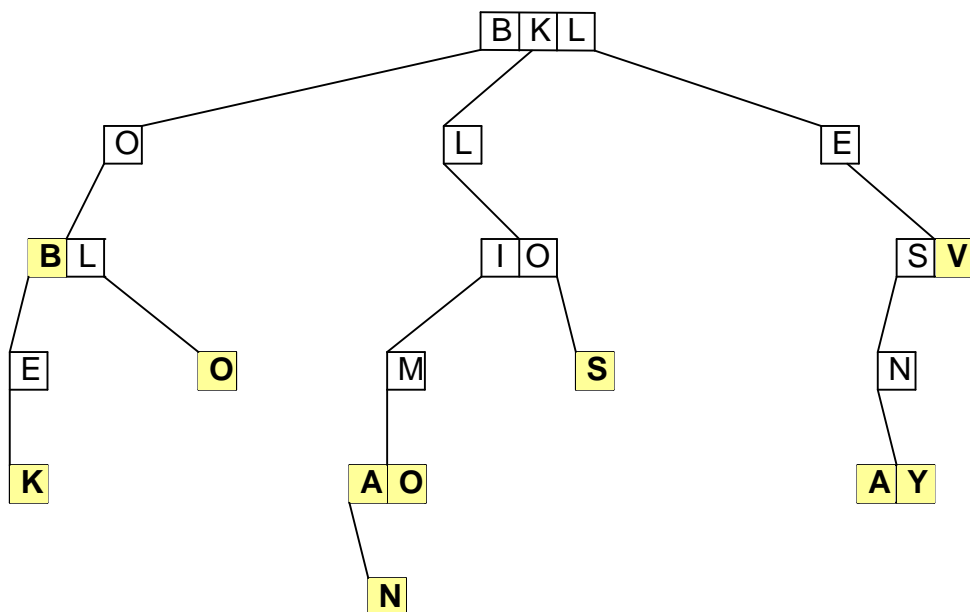
Na i-tej úrovni sú i-té písmená všetkých kľúčov

* je terminátor kľúča (počet hviezdičiek ... počet kľúčov v štruktúre)

Na mieste hviezdičky si predstavujeme link na informačnú časť záznamu

Možné implementácie:

a.) Explicitný viaccestný strom množín implementovaných jednorozmerným dynamickým poľom



b.) Dvojrozmerné (riedke) pole:

- implicitné pole množín implementovaných poľom

č. množiny	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y	Z
0		1									6	12													
1															2										
2		-4										3													
3															-1										
4					5																				
5											-1														
6												7													
7									8						9										
8												10													
9													10						-1						
10	-11														-1										
11														-1											
12					13																				
13																			14			-1			
14														15											
15	-1																							-1	
16																									
17																									
18																									
19																									
20																									

Záporná hodnota: a) terminátor (-1)

b) terminátor a číslo množiny ďalšieho znaku

(rôzne od -1)

Šedým vyznačené: vyhľadanie "BOBEK".

Materiál slúži výlučne pre študentov FRI ŽU, nie je dovolené ho upravovať, prípadne ďalej šíriť.

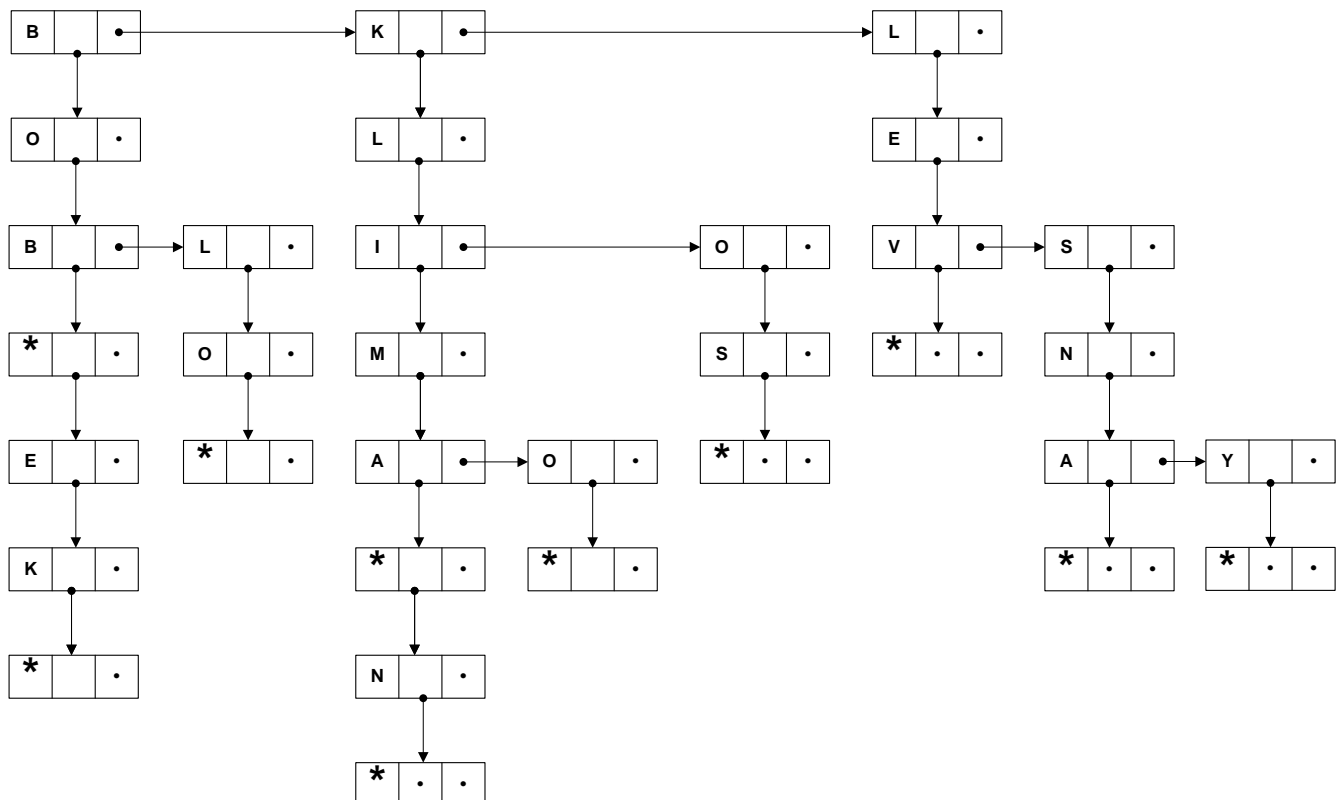
Vlastnosti, ktoré platia pre implementáciu a aj b:

Nech d je maximálna dĺžka kľúča v znakoch, s počet množín, n počet kľúčov, k počet znakov abecedy.

- výška stromu množín je d
- najhoršia zložitosť *Nájdí, Vlož, Zruš* je $O(d)$ - nezávisí na veľkosti tabuľky
- maximálna spotreba pamäti je $(2^d - 1) * k$ a to nezávisle na n , teda implementujúce polia množín typicky pamäťovo nevyužitú.

c.) De la Briandais - strom (binárny znakový strom)

- transformácia viaccestného stromu na binárny
- prvky v každej množine sú zreťazené



Rádové zložitosti v binárnom znakovom strome ťažko odhadnuteľné. Zostáva nezávislosť na n , zložitosť hľadania v množine sa zvyšuje z $O(1)$ v poli na $O(k)$ v zozname maximálnej dĺžky k . Pamäťová zložitosť podstatne lepšia.

Možnosť optimalizácie štruktúry, ak poznáme rozdelenie početnosti prístupu k jednotlivým kľúčom:

- usporiadame zoznamy v poradí klesajúcej početnosti alebo
- implementujeme samoorganizáciu stromu:
 - heuristika “nájdený na začiatok” alebo
 - heuristika “nájdený doľava” alebo
 - sledovanie početnosti a periodická reorganizácia.

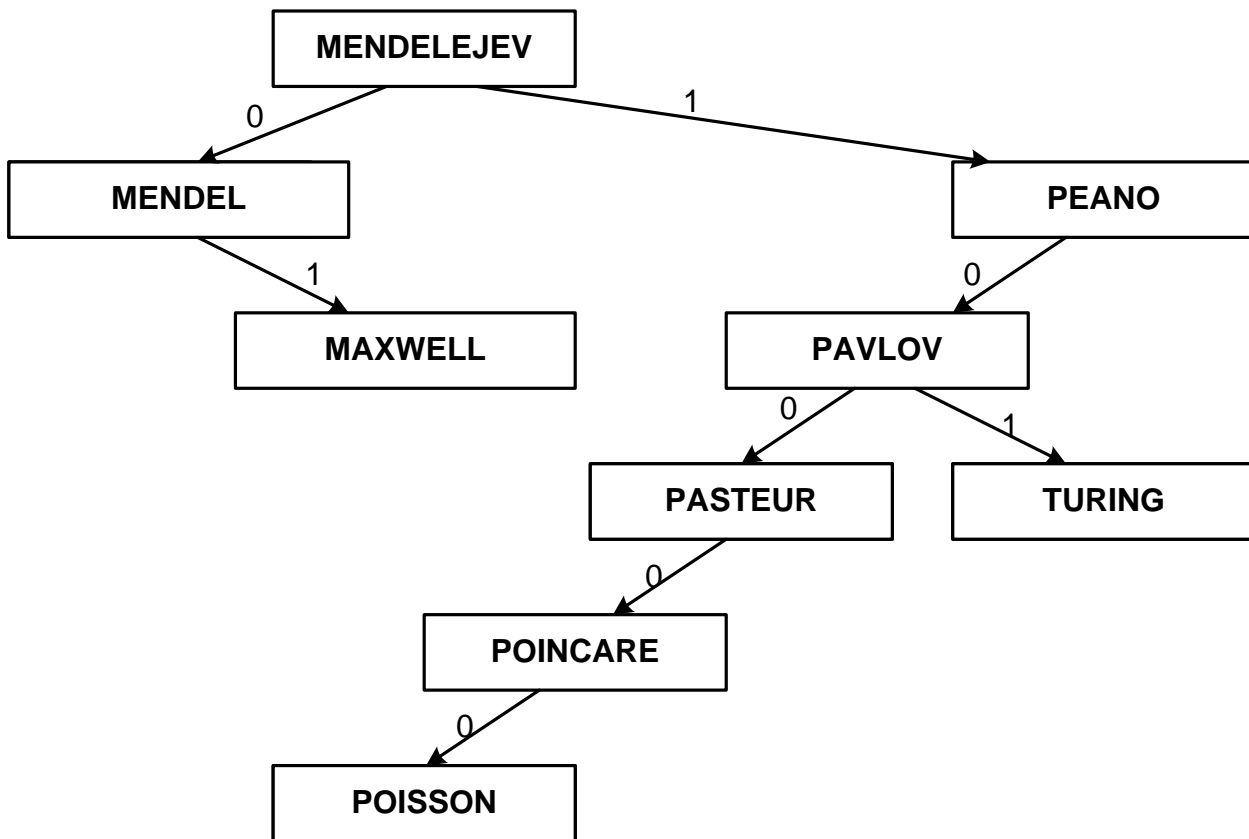
Možnosť anticipovaného vyhľadania (najmä) v optimalizovanej štruktúre pri interaktívnom prístupe:

- po zadaní každého znaku sa “ponúkne” najpravdepodobnejšie hľadaný kľúč (znaky zo začiatkov zoznamov). Napríklad:
- po zadaní K : KLIMA; po zadaní KLO: KLOS

Digitálny znakový strom

- modifikácia znakového stromu, keď kľúč sa chápe ako bitový reťazec, teda znakový reťazec s dĺžkou abecedy $k = 2$
- je to binárny strom, v ktorom sa vetví na základe hodnoty bitu (0 - vľavo, 1 - vpravo)
- uzol stromu obsahuje dva smerníky a jeden z kľúčov začínajúcich na bitový reťazec vytvorený cestou od koreňa po tento uzol.

V nasledujúcom príklade sa predpokladá ASCII kód, v ktorom: A=00001, ... M=01101, ..., P=10000, ... , T=10100, ..., Z=11010



V prípade, že všetky bity z kľúča boli vyčerpané a v strome sa neprišlo až k listu je potrebné vo vrchole ukladať zoznam hodnôt (na obrázku je v tomto zozname vždy jedna hodnota).

Možnosť skonštruovania optimálneho (z hľadiska rýchlosti prístupu) digitálneho stromu za podmienok:

- vopred je známa množina kľúčov,
- vopred je známa frekvencia prístupu k jednotlivým kľúčom.

Potom: umiestnenie kľúčov volíme tak, že do koreňa (úroveň 0) dáme kľúč s maximálnou frekvenciou a na najnižšiu úroveň kľúč s minimálnou frekvenciou prístupu. Teda vľavo od koreňa je kľúč s najvyššou frekvenciou z tých, ktoré začínajú nulou (okrem kľúča v koreni), analogicky vpravo atď smerom dole.

Vkladanie: štandardné - vloží sa na list, kde skončilo neúspešné hľadanie.

Odobratie: odoberaný prvok sa nahradí ľubovoľným listom z jeho podstromu a potom sa tento list zruší.

Časové zložitosti: rovnaké, ako pri náhodnom (nevyváženom) binárnom vyhľadávacom strome (BVS).