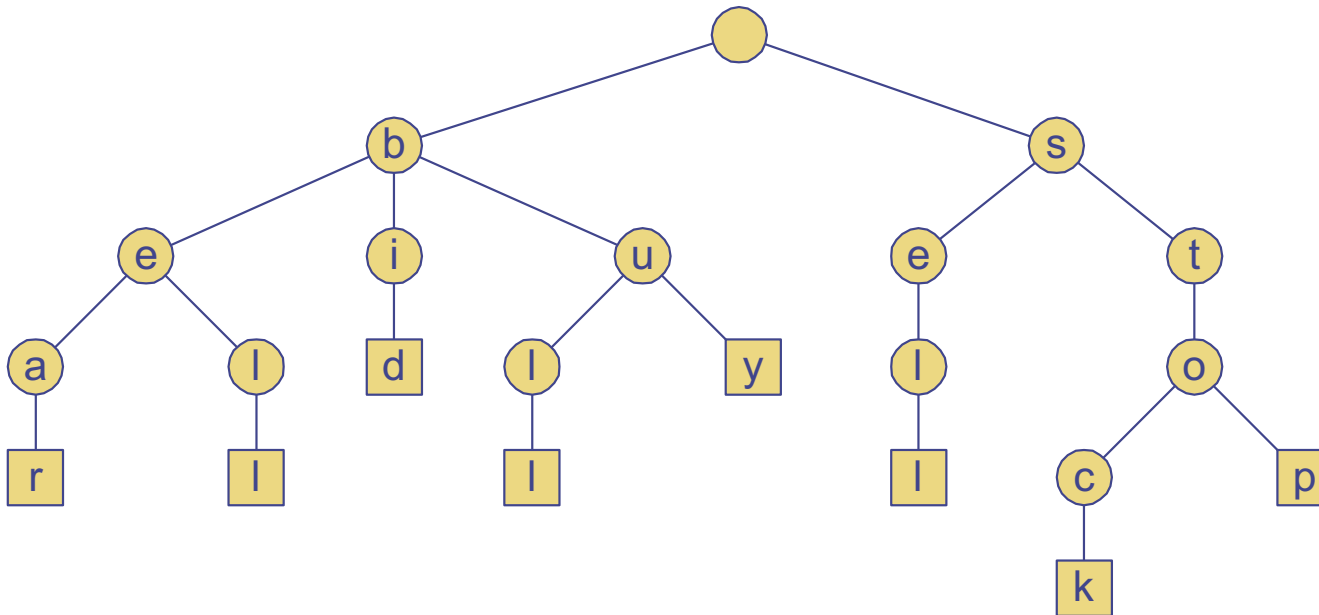


Standard Tries

- The standard trie for a set of strings S is an ordered tree such that:
 - Each node but the root is labeled with a character
 - The children of a node are alphabetically ordered
 - The paths from root to the external nodes yield the strings of S



$S = \{\text{bear, bell, bid, bell, buy, sell, stock, shop}\}$

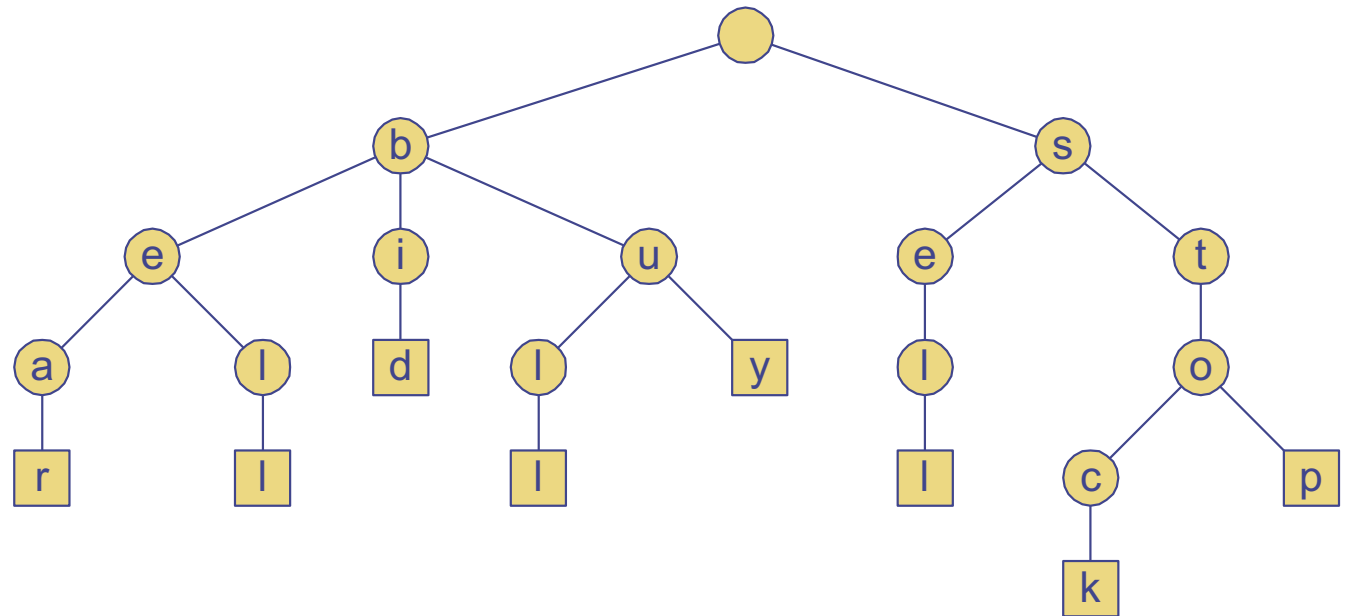
Analysis of Standard Tries

- A standard trie uses $O(n)$ space and supports searches, insertions and deletions in time $O(dm)$ where:

n : total size of all strings in S

m : size of the string being searched/inserted/deleted

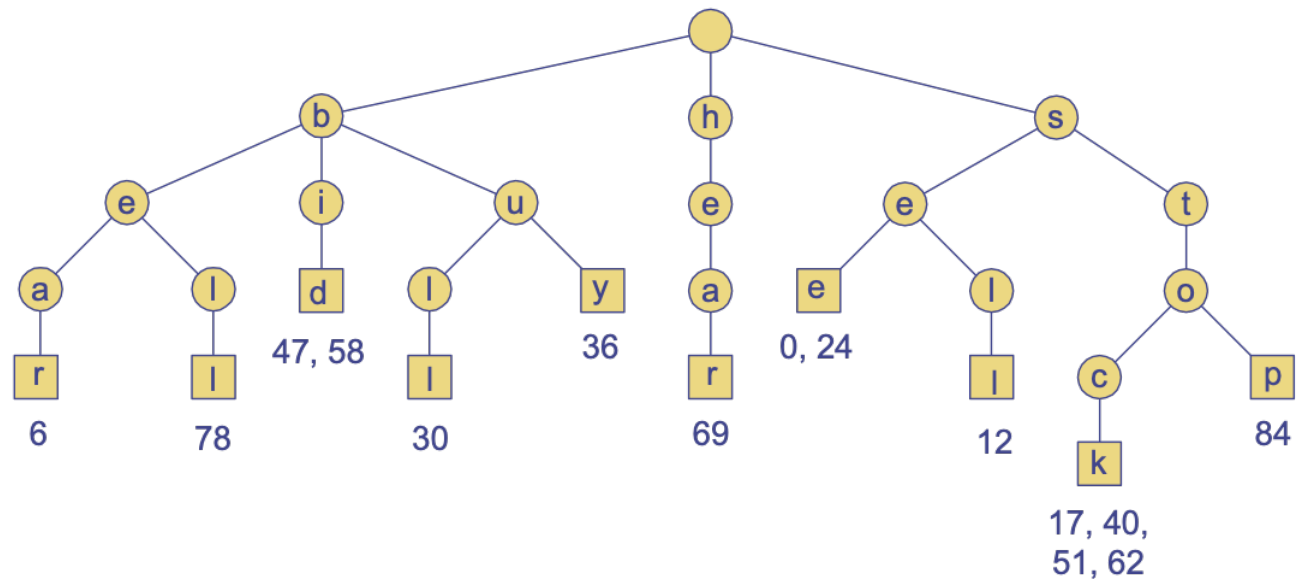
d : size of the alphabet



Word Matching with Trie

- We insert words of the text into a trie
- Each leaf stores the indices of the occurrences of the associated words in the text

s	e	e		a		b	e	a	r	?		s	e	l	l		s	t	o	c	k	!	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
s	e	e		a		b	u	l	l	?		b	u	y		s	t	o	c	k	!		
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	
b	i	d		s	t	o	c	k	!		b	i	d		s	t	o	c	k	!			
47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68		
h	e	a	r		t	h	e		b	e	l	l	?		s	t	o	p	!				
69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88				

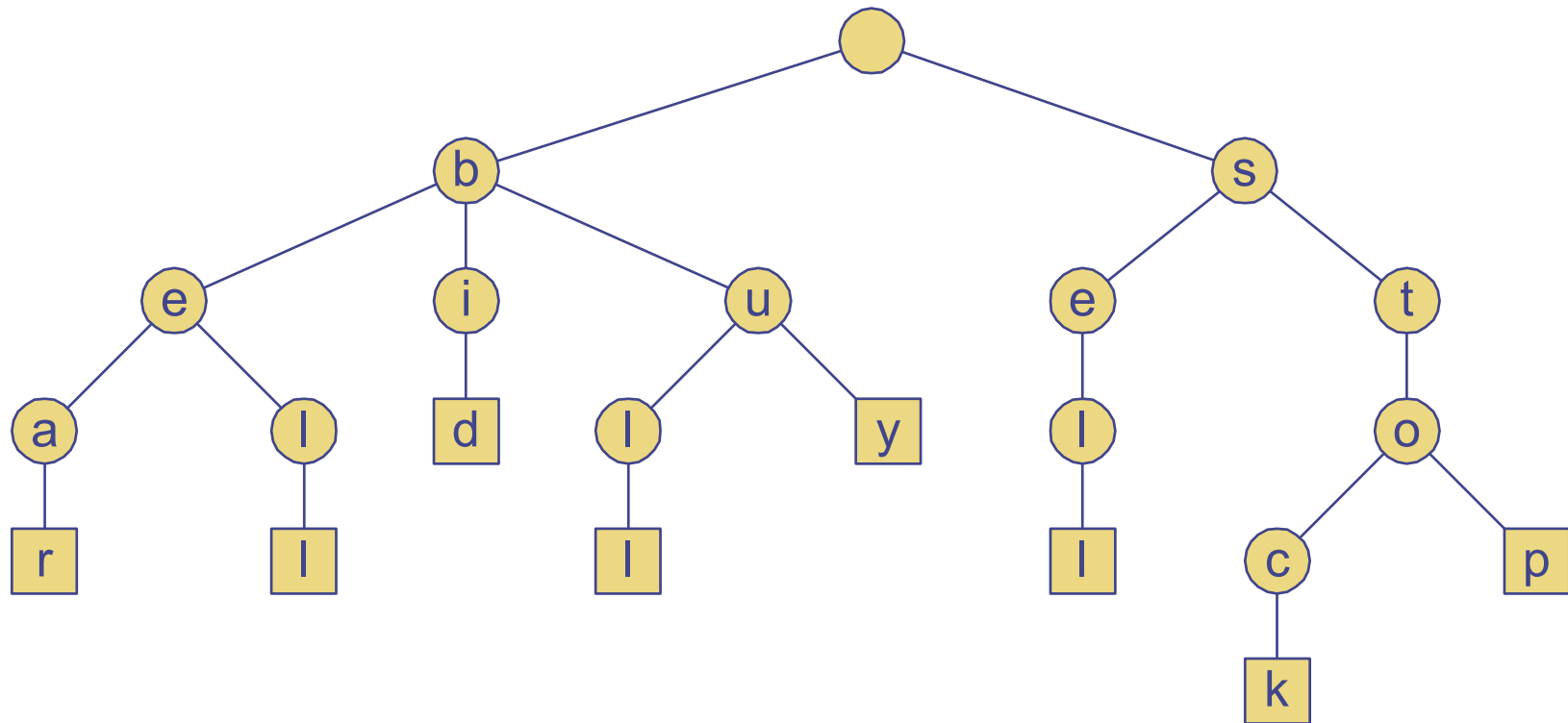


Standard Trie Generation

- To construct a standard trie, we can use an incremental algorithm which inserts strings from S one at a time.
- To insert string X in the current trie T :
 - First, trace the path associated with X in T .
 - The search will stop at an internal node before reaching the end of X .
 - Create a new chain of node descendents to store the remaining characters.

Standard Trie Generation

$S = \{\text{bear, bell, bid, bell, buy, sell, stock, shop}\}$



More Tries

- There are other types of tries such as compressed tries, suffix tries, etc. which are optimized for space or handling different pattern matching problem.
- More information in [Goodman] Chapter 12, Section 12.5.