

# Unidade X: Árvores TRIE PATRICIA



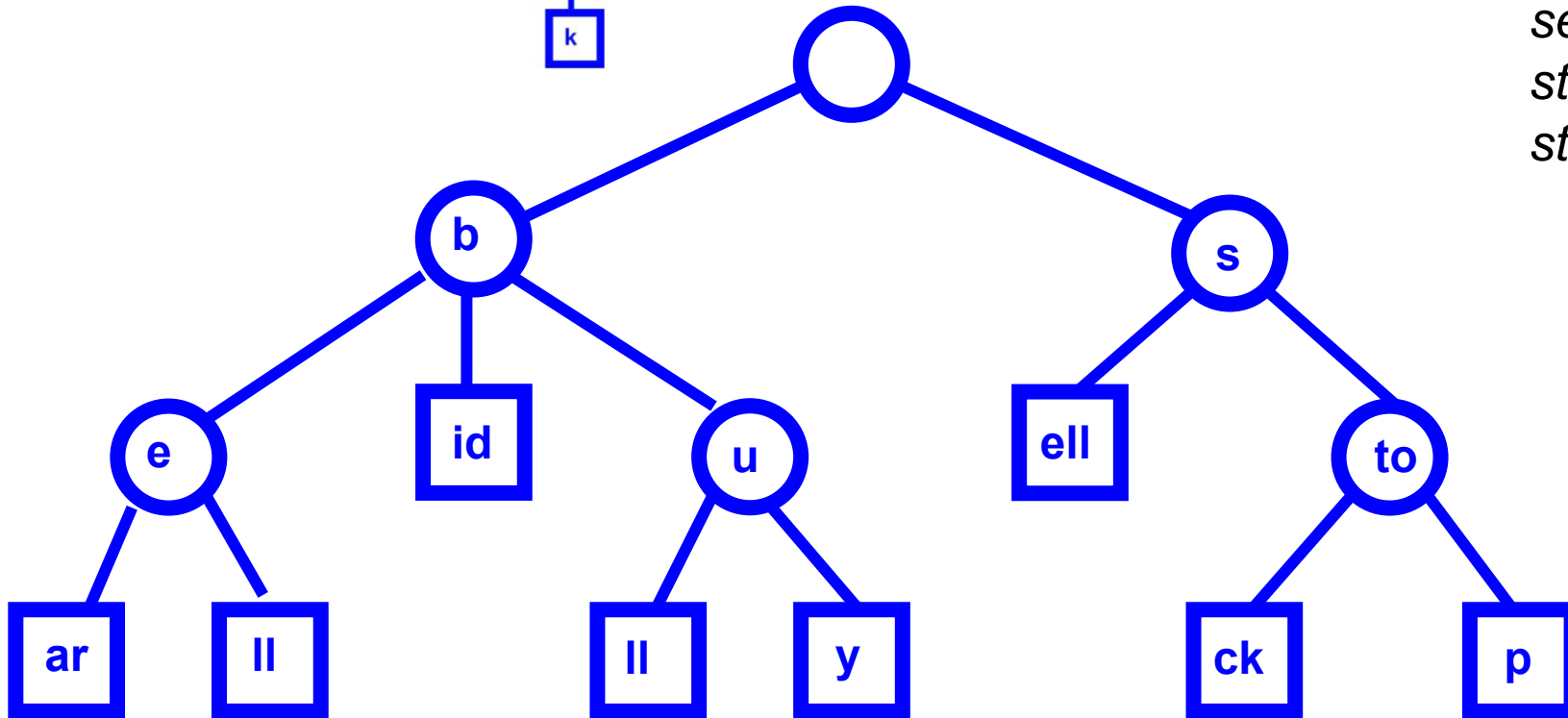
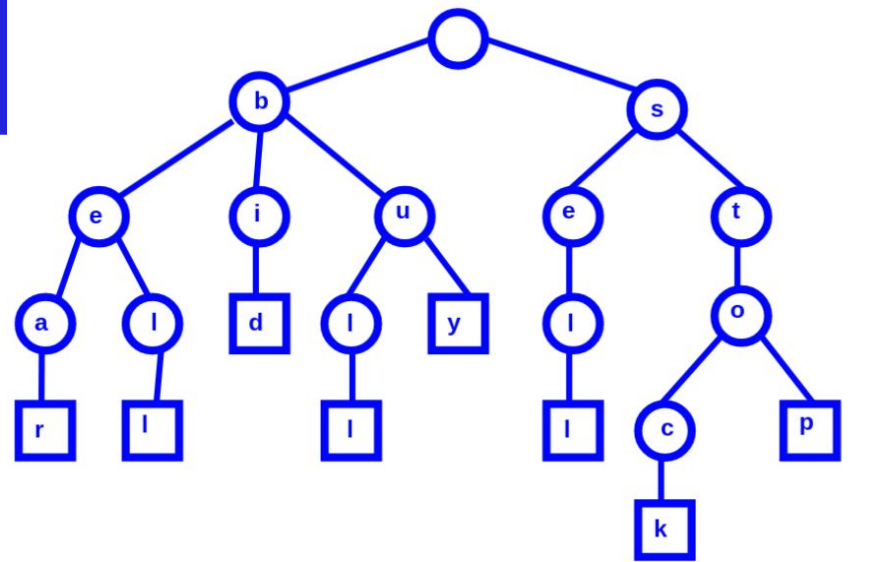
**PUC Minas**

Instituto de Ciências Exatas e Informática  
Departamento de Ciência da Computação

- Significa *Practical Algorithm to Retrieve Information Coded in Alphanumeric*
- Elimina os nós redundantes fazendo com que todos os nós (exceto a raiz) tenham pelo menos dois filhos
- Na *trie-padrão*, a existência de nós com apenas um filho representa ineficiência em termos de espaço

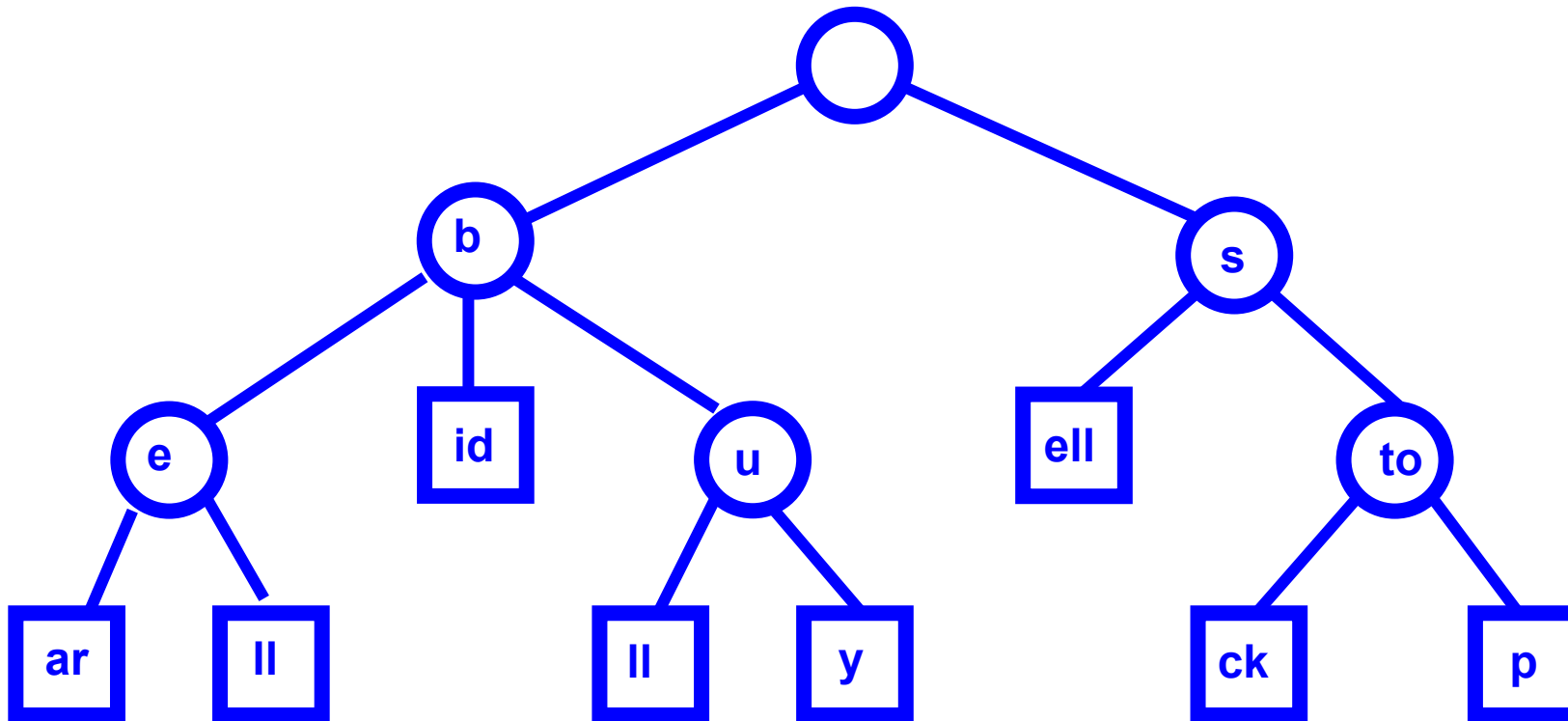
# Exemplo de *Trie Patricia*

- bear* - urso
- bell* - sino
- bid* - oferta
- bull* - touro
- buy* - compra
- sell* - vende
- stock* - ação
- stop* - parar



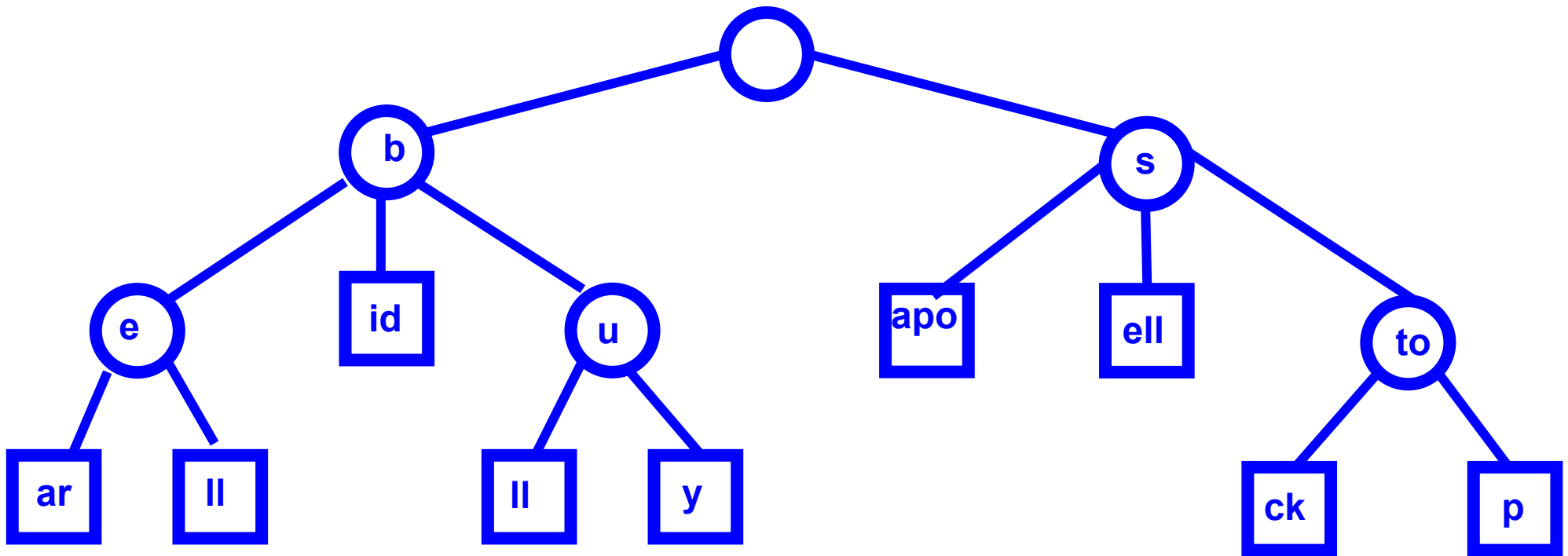
## Exercício Resolvido (1)

- Insira as palavras sapo e sapato na árvore abaixo



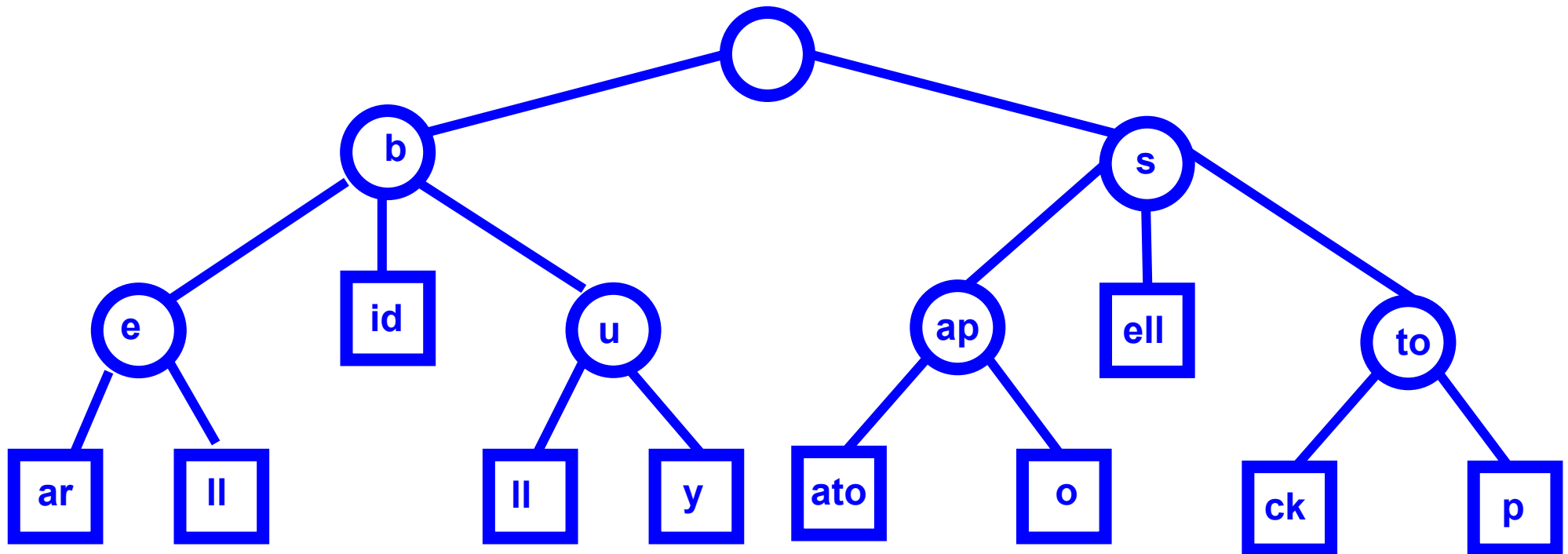
## Exercício Resolvido (1)

- Insira as palavras **sapo** e sapato na árvore abaixo



# Exercício Resolvido (1)

- Insira as palavras sapo e sapato na árvore abaixo



# Propriedades das *Trie Patricia*

- Nós rotulados por *substrings* das cadeias de caracteres da coleção S
- Temos  $\Theta(s)$  nós, onde  $s$  é o número de cadeias da coleção
- O número de nós é proporcional ao número de cadeias da coleção S; não ao comprimento das mesmas
- Todo nó interno tem entre 2 e  $d$  filhos

# Estrutura de Dados das *Trie Patricia*

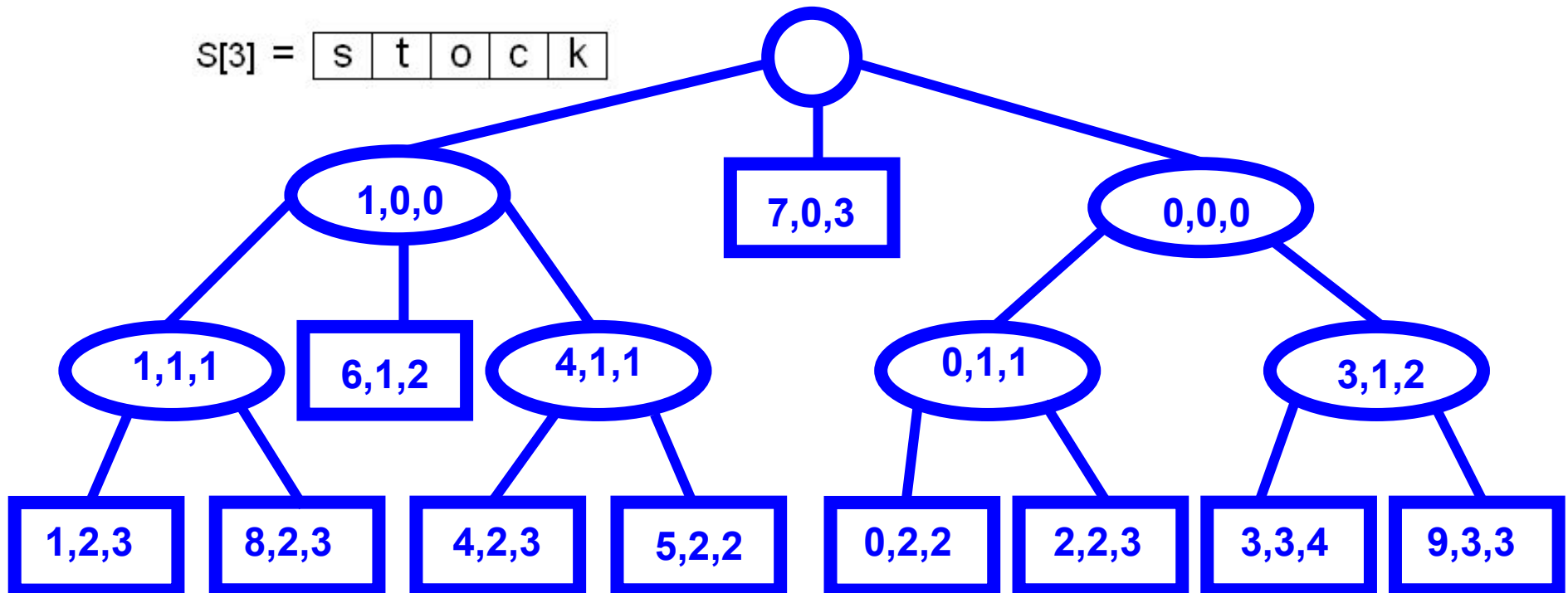
- Cada nó armazena uma tripla de inteiros  $(i, j, k)$ , indicando o rótulo do nó de tal forma que  $S[i][j...k]$ , onde:
  - A coleção de cadeias  $S$  será  $S[0], S[1], \dots, S[s-1]$
  - $j$  e  $k$  representam, respectivamente, a primeira e última (inclusive) posições da cadeia  $S[i]$  que correspondem ao rótulo corrente



# Exercício Resolvido (2)

- Na figura, identifique a substring representada por cada nó

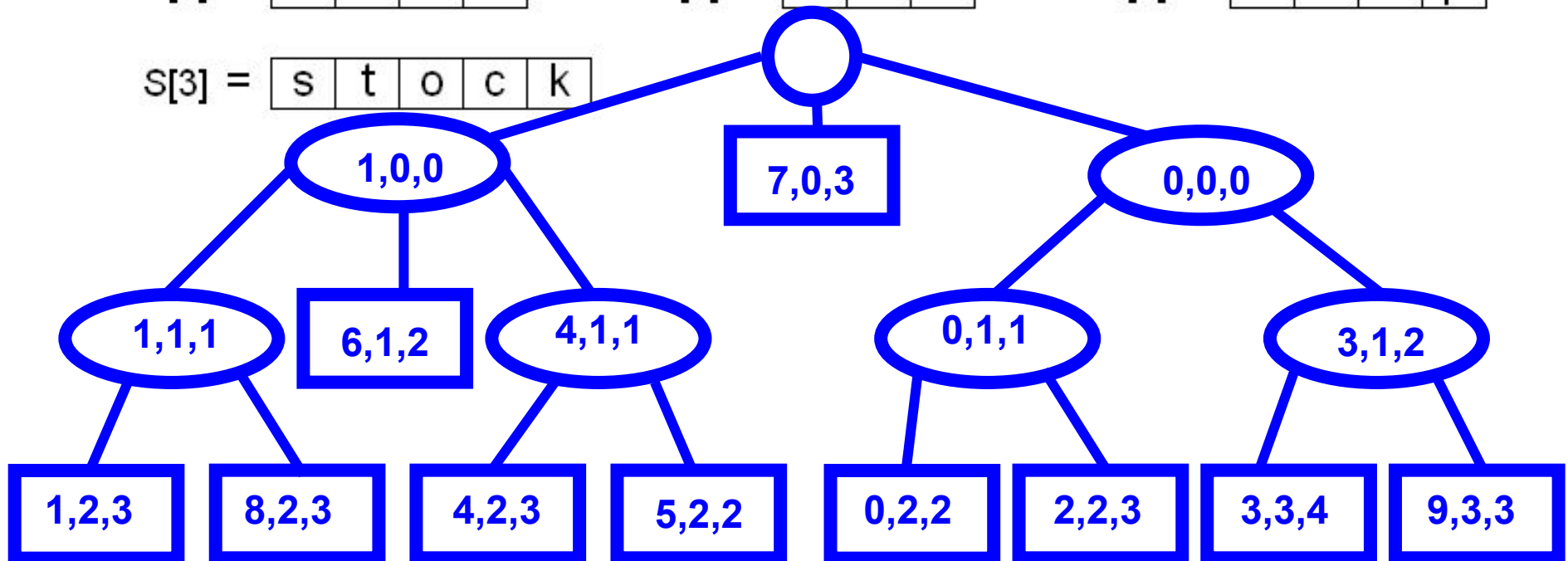
	0 1 2 3 4		0 1 2 3		0 1 2 3											
s[0] =	<table border="1"><tr><td>s</td><td>e</td><td>e</td></tr></table>	s	e	e	s[4] =	<table border="1"><tr><td>b</td><td>u</td><td>l</td><td>l</td></tr></table>	b	u	l	l	s[7] =	<table border="1"><tr><td>h</td><td>e</td><td>a</td><td>r</td></tr></table>	h	e	a	r
s	e	e														
b	u	l	l													
h	e	a	r													
s[1] =	<table border="1"><tr><td>b</td><td>e</td><td>a</td><td>r</td></tr></table>	b	e	a	r	s[5] =	<table border="1"><tr><td>b</td><td>u</td><td>y</td></tr></table>	b	u	y	s[8] =	<table border="1"><tr><td>b</td><td>e</td><td>l</td><td>l</td></tr></table>	b	e	l	l
b	e	a	r													
b	u	y														
b	e	l	l													
s[2] =	<table border="1"><tr><td>s</td><td>e</td><td>l</td><td>l</td></tr></table>	s	e	l	l	s[6] =	<table border="1"><tr><td>b</td><td>i</td><td>d</td></tr></table>	b	i	d	s[9] =	<table border="1"><tr><td>s</td><td>t</td><td>o</td><td>p</td></tr></table>	s	t	o	p
s	e	l	l													
b	i	d														
s	t	o	p													
s[3] =	<table border="1"><tr><td>s</td><td>t</td><td>o</td><td>c</td><td>k</td></tr></table>	s	t	o	c	k										
s	t	o	c	k												



# Exercício Resolvido (3)

- Insira as palavras sapo e sapato na árvore abaixo

0 1 2 3 4	0 1 2 3	0 1 2 3											
S[0] = <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>s</td><td>e</td><td>e</td></tr></table>	s	e	e	S[4] = <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>b</td><td>u</td><td>l</td><td>l</td></tr></table>	b	u	l	l	S[7] = <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>h</td><td>e</td><td>a</td><td>r</td></tr></table>	h	e	a	r
s	e	e											
b	u	l	l										
h	e	a	r										
S[1] = <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>b</td><td>e</td><td>a</td><td>r</td></tr></table>	b	e	a	r	S[5] = <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>b</td><td>u</td><td>y</td></tr></table>	b	u	y	S[8] = <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>b</td><td>e</td><td>l</td><td>l</td></tr></table>	b	e	l	l
b	e	a	r										
b	u	y											
b	e	l	l										
S[2] = <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>s</td><td>e</td><td>l</td><td>l</td></tr></table>	s	e	l	l	S[6] = <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>b</td><td>i</td><td>d</td></tr></table>	b	i	d	S[9] = <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>s</td><td>t</td><td>o</td><td>p</td></tr></table>	s	t	o	p
s	e	l	l										
b	i	d											
s	t	o	p										
S[3] = <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td>s</td><td>t</td><td>o</td><td>c</td><td>k</td></tr></table>	s	t	o	c	k								
s	t	o	c	k									



- Na prática, a *trie-patricia* tem ganhos em termos de espaço mesmo considerando o armazenamento da coleção S

# Exercício Resolvido (4)

- Implemente a classe nó da árvore trie-patricia

# Exercício Resolvido (5)

- Implemente a árvore *trie-patricia*: construtor, pesquisar, inserir e mostrar