

# Árvores Trie e Patricia

Disciplina de Algoritmos e Estrutura de  
Dados III

Prof. Marcos Antonio Schreiner

21/05/2018

# Introdução

- Seja uma busca de uma chave  $x$  em um conjunto de chaves  $S = \{s_1, \dots, s_n\}$ .
- Nas estruturas estudadas organizávamos  $S$  de modo a fazermos comparações entre  $s_i$  e  $x$  até encontrarmos  $x$ .
- As chaves  $s_i$  e  $x$  são *tratadas* como um único dado indivisível na estrutura.
- E se as chaves excederem o espaço reservado?

# Introdução

- E se as chaves excederem o espaço reservado?
  - Chaves dinâmicas.

Suponha que se deseje armazenar um texto e em seguida, tentar localizar as frases desse texto.

- Neste caso, as chaves corresponderia às frases do texto.

Neste cenário, as **Árvores Trie ou Patricia** e a Busca Digital são apropriadas.

# Busca Digital

- Na busca digital a chave é tratada como um elemento **divisível**.
- Cada chave é constituída de um **conjunto de caracteres ou dígitos**.
- Na busca, a comparação é efetuada entre os dígitos que compõem as chaves (dígito a dígito, caractere a caractere).

# CHAVES: Características

- Cada **chave** formada por **palavras** sobre um **alfabeto de símbolos**.
- **Curiosidade:** Pesquise sobre **Autômatos Finitos**.
- Palavras com tamanho **VARIÁVEL** e **ILIMITADO**.

# CHAVES: Características

Exemplos de **alfabetos**:

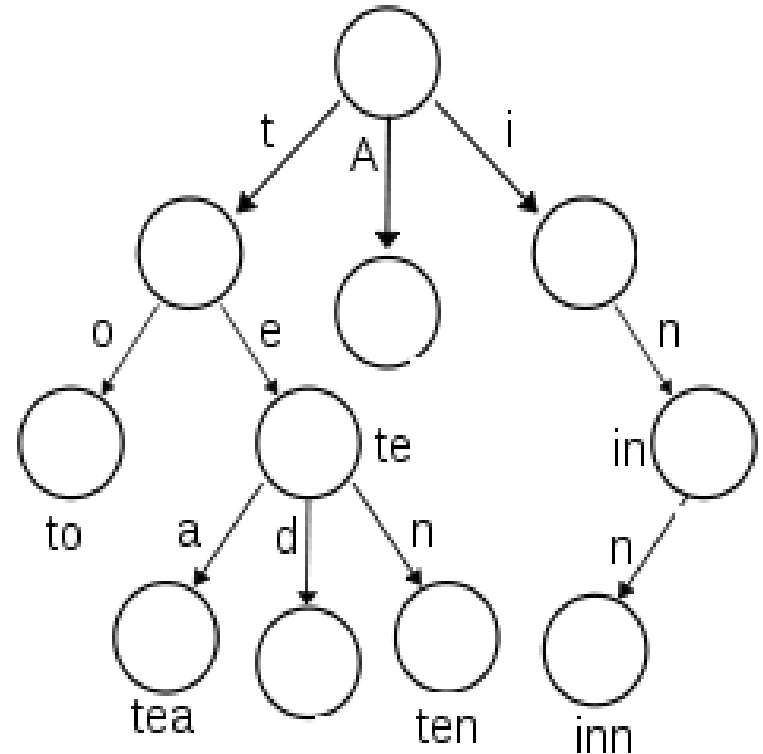
- $\{0,1\}$ ,  $\{A, B, C, \dots, Z, a, b, c, \dots, z\}$ ,  $\{0,1,2,3,4,5, \dots, 9\}$

Exemplos de **chaves**:

- 0101010100000000001010000000001010
- ABABBBABABA, Maria
- 19034717

# Introdução

- TRIE vem de RETRIEVAL  
– RECUPERAÇÃO
- É um tipo de árvore de busca.
- Ideia geral: usar **partes** das CHAVES como caminho busca



# Árvore TRIE - Características

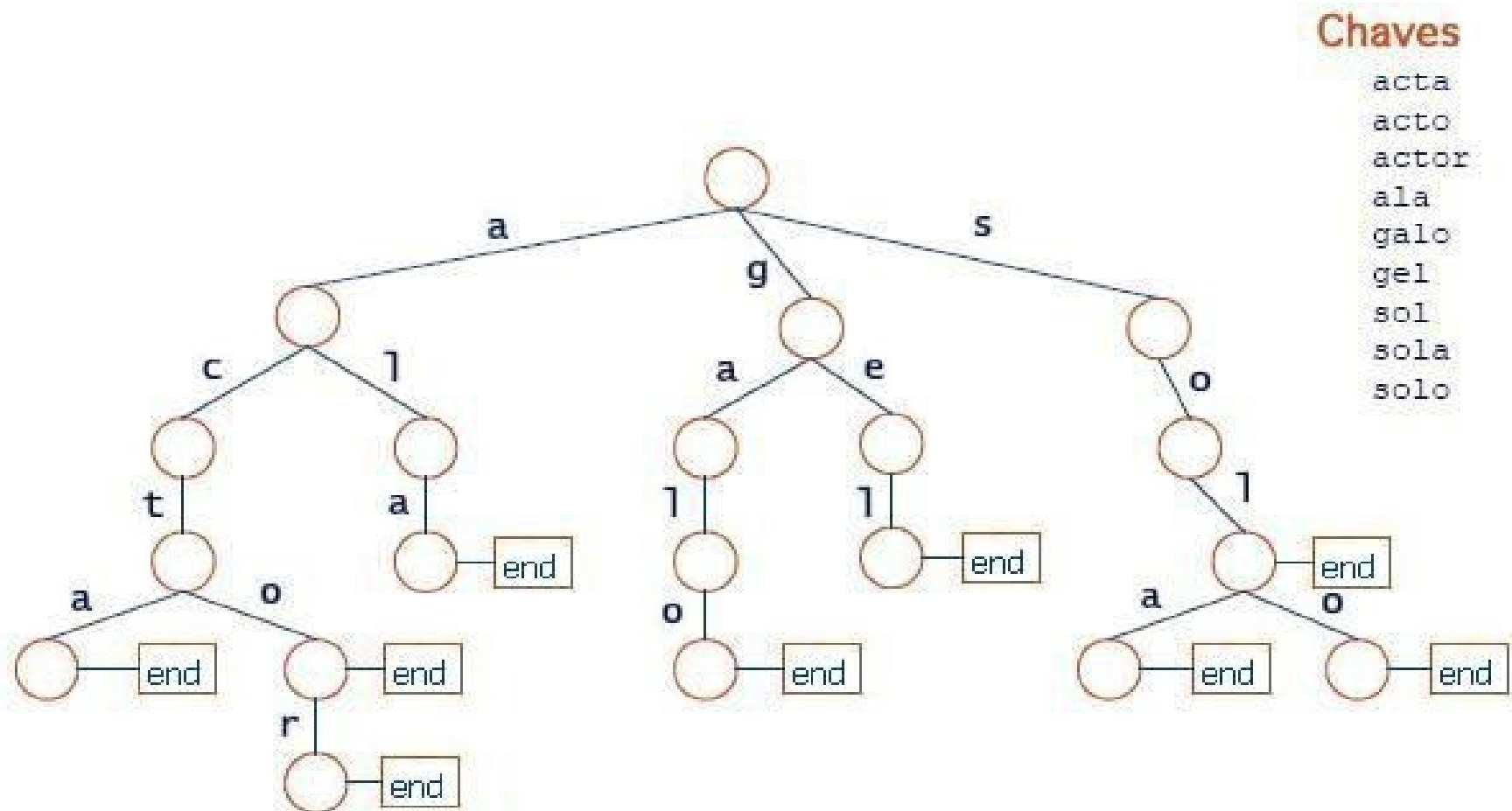
- Árvore N-ária.
- Chaves em geral são caracteres.
- Ao contrário da árvore binária de busca **nenhum nó armazena a chave.**
- Chave determinada pela posição na árvore.
- O **grau da árvore** corresponde ao tamanho do alfabeto



# Árvore TRIE - Características

- Cada nível percorrido corresponde a avançar um dígito na chave.
- O caminho da raiz para qualquer nó é um **prefixo de uma chave**.
- Descendentes do mesmo nó tem o mesmo prefixo.
- Raiz representa a chave vazia.
- Nós devem indicar quando completar uma chave.

# Árvore TRIE - Exemplo



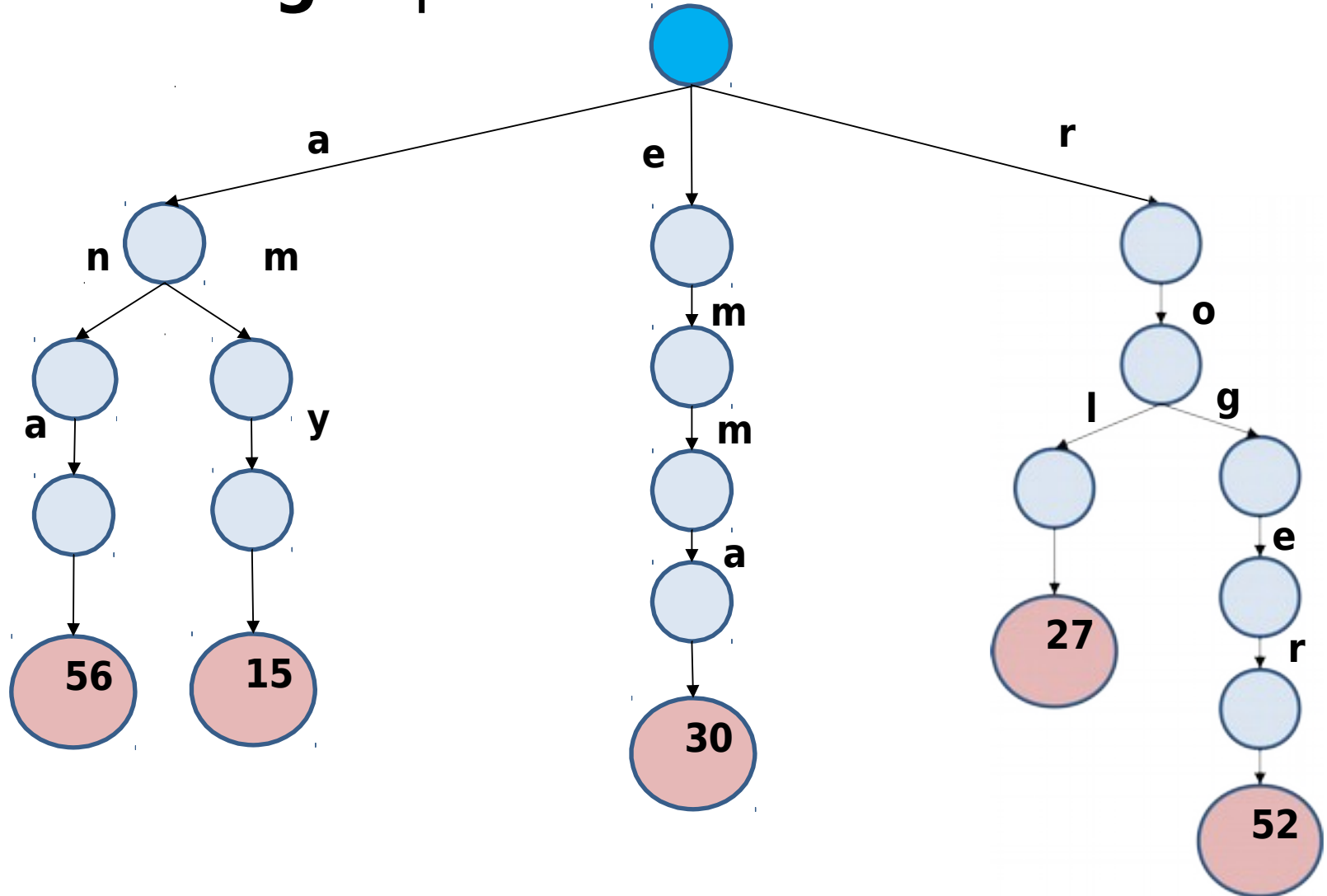
# Árvore TRIE – Buscar Chaves

Busca caractere a caractere a CHAVE.

- 1) Se o caractere não pertence a árvore a chave não pertence a TRIE;
- 2) Se o caractere pertence a árvore,  
verifique o próximo e caso todos os caracteres pertençam em sequencia a TRIE, a chave pertence a árvore.

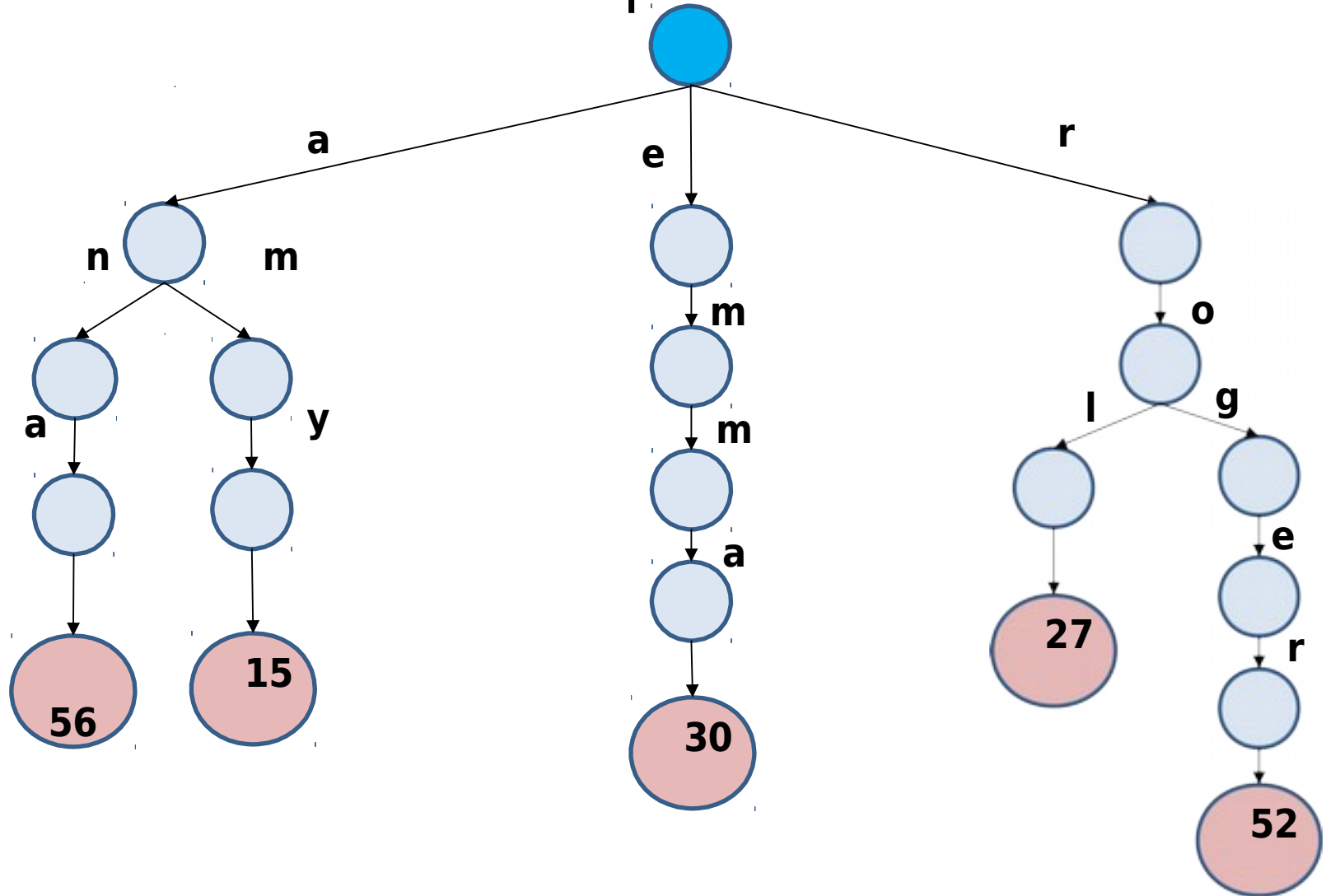
# Árvore TRIE – Buscar Chaves

- A chave **roger** pertence a árvore



# Árvore TRIE – Buscar Chaves

- A chave **roberta** não pertence a árvore



# Incluir Chaves – Sequencia de Passos

- 1) Faz-se uma **busca** pela palavra a ser inserida. Se ela já existir na TRIE nada é feito.
- 2) Caso contrário, é recuperado o último nó **n** da maior **substring** da palavra a ser inserida.
- 3) O **restante** dos caracteres da chave são adicionados na TRIE a partir do nó **n**.

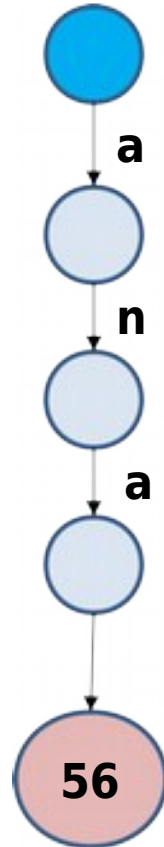
# Árvore TRIE – Incluir Chaves

Vamos inserir os seguinte conjunto de dados em uma Árvore TRIE:

- ana 56
- amy 15
- emma 30
- rol 27
- roger 52

# Árvore TRIE – Incluir Chaves

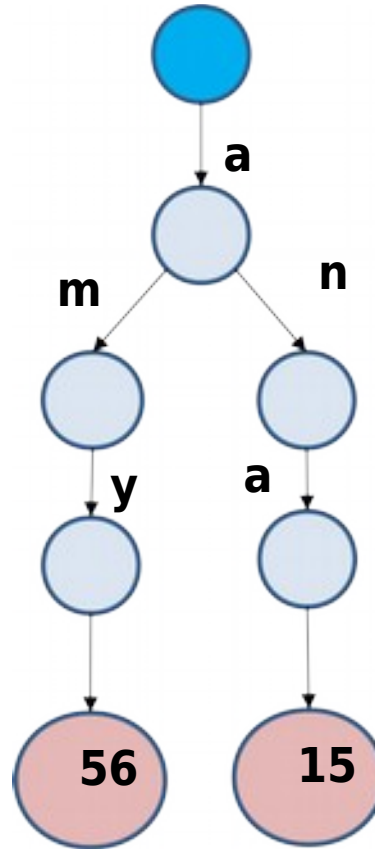
- ana 56





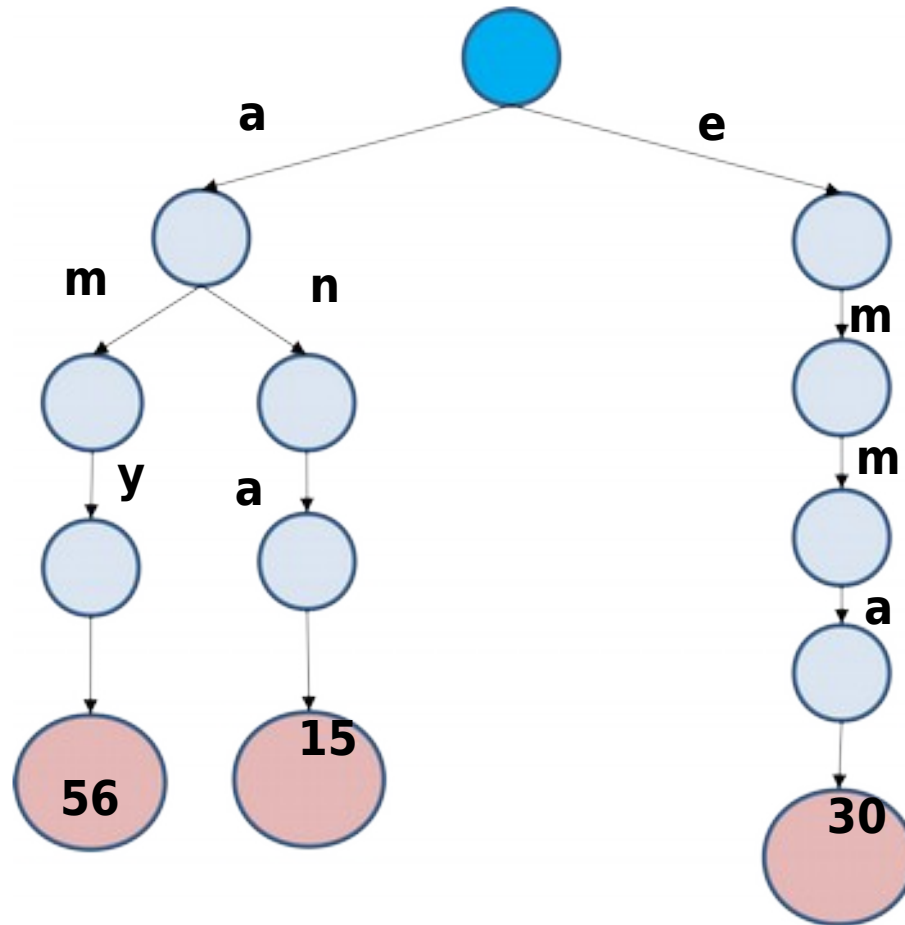
# Árvore TRIE – Incluir Chaves

- amy 15



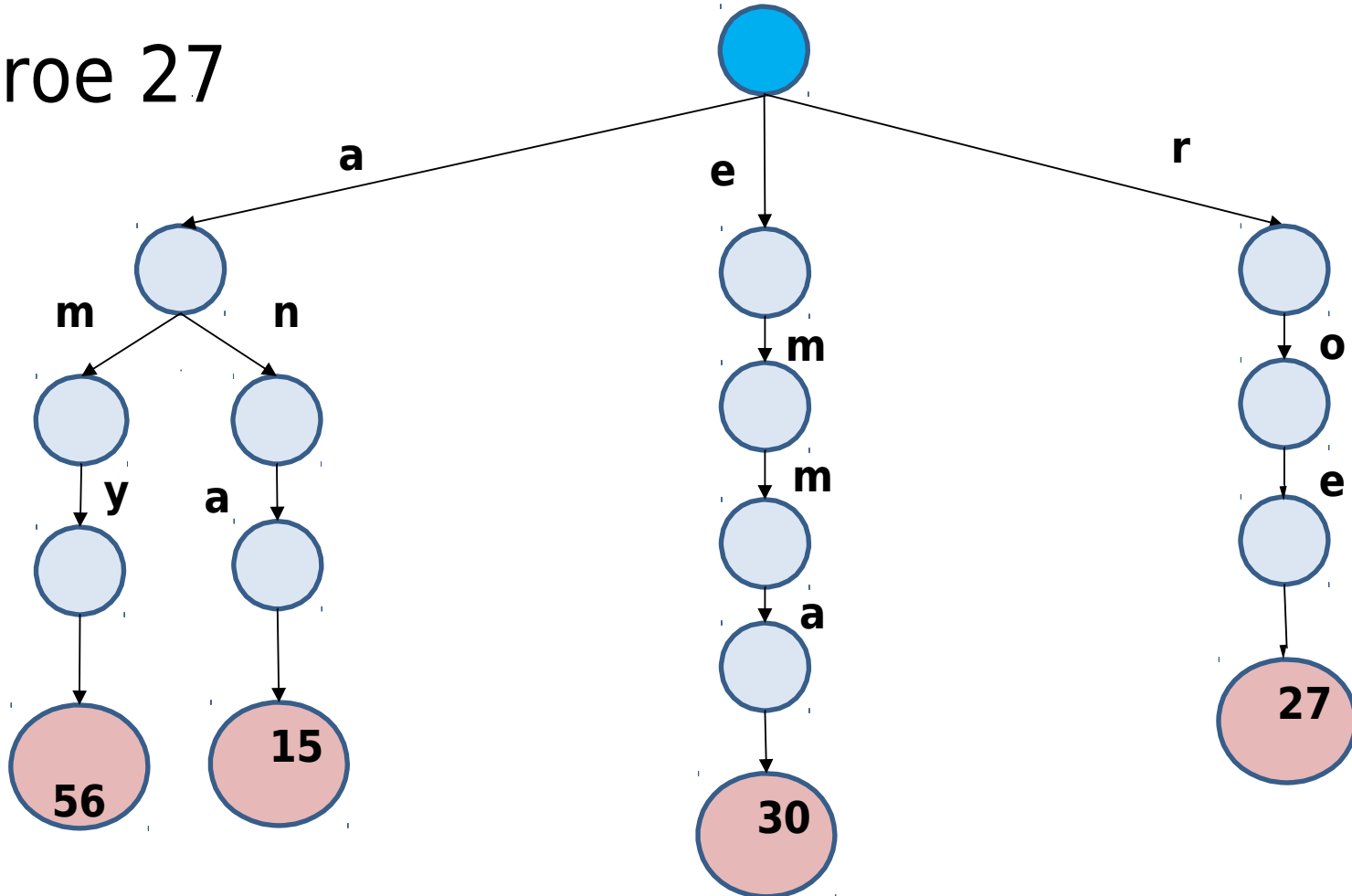
# Árvore TRIE – Incluir Chaves

- emma 30



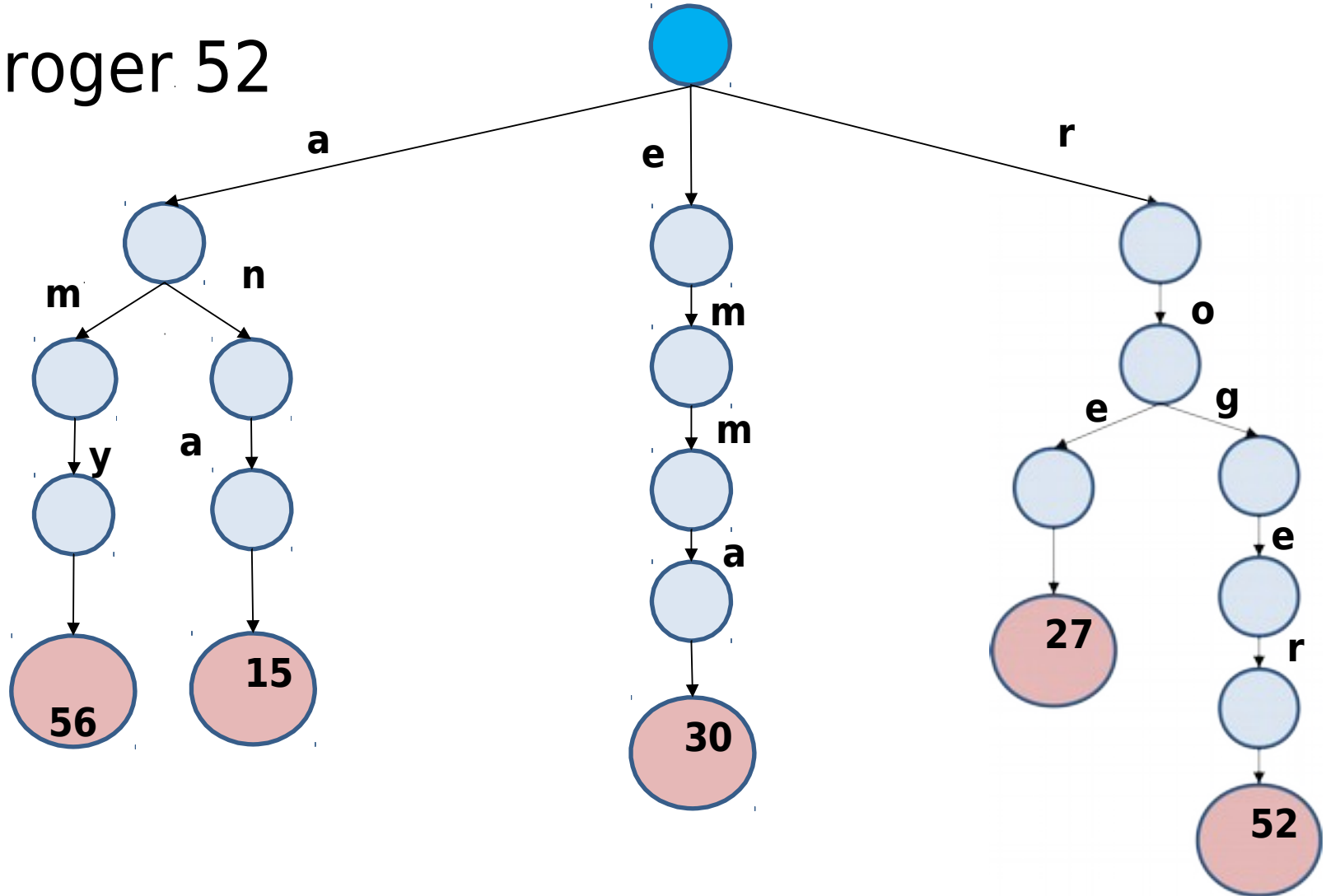
# Árvore TRIE – Incluir Chaves

- roe 27



# Árvore TRIE – Incluir Chaves

- roger 52

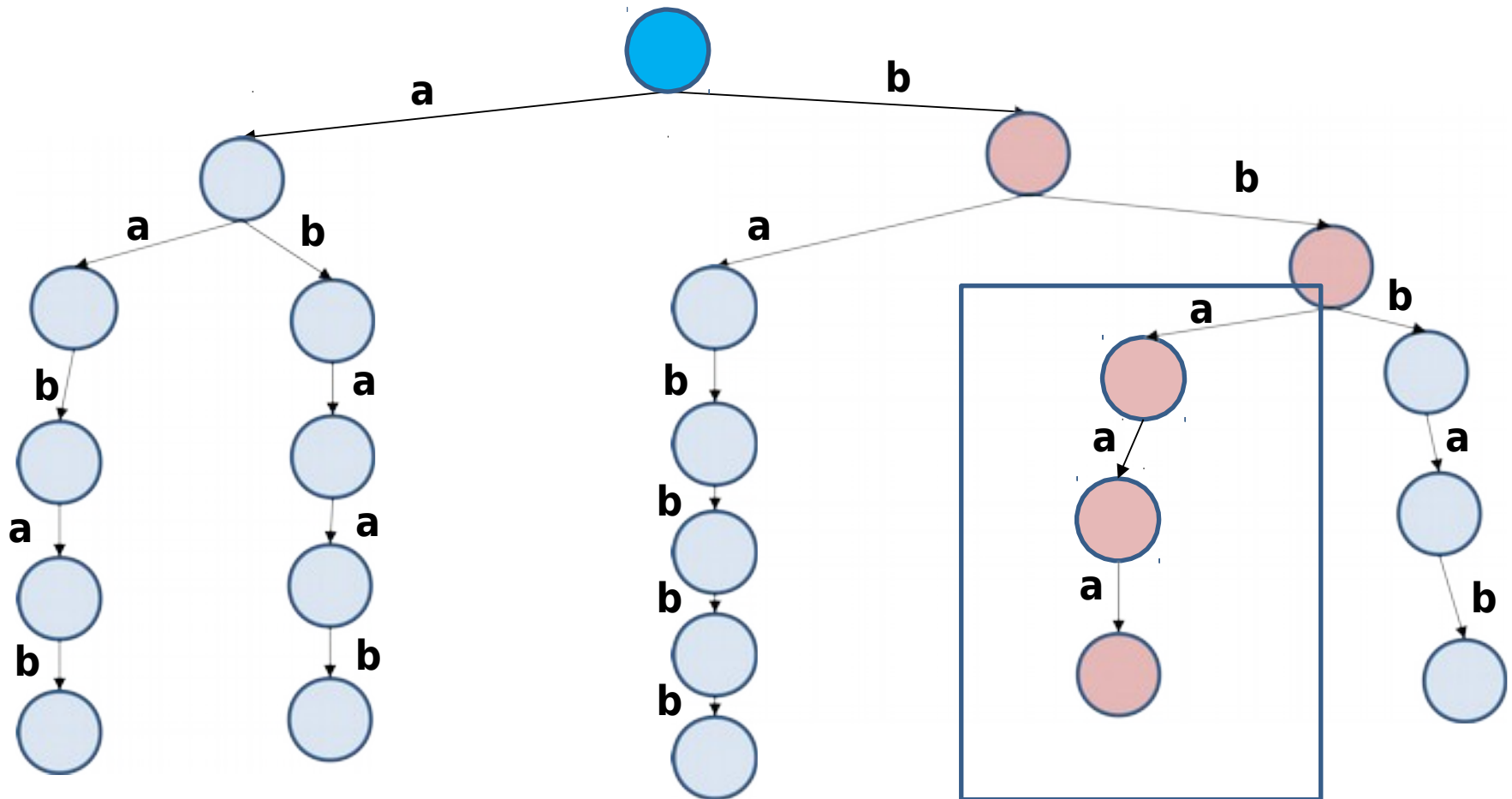


# Árvore TRIE – Remover Chave

- 1) Busca-se a chave a ser removida.
- 2) A partir da folha (bottom-up), são removidos todos os nós que tem apenas um filho.

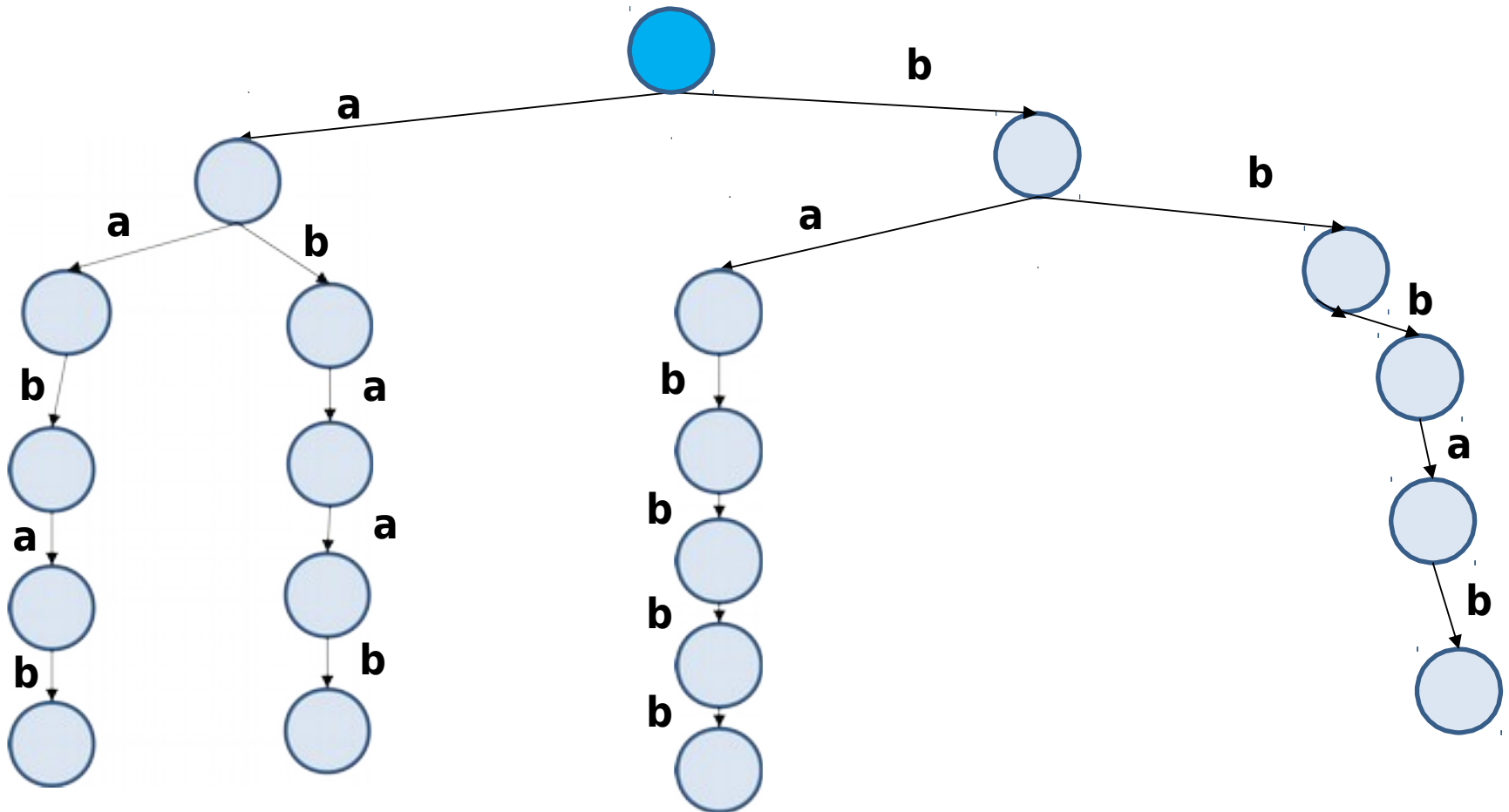
# Árvore TRIE – Remover Chave

## Remoção da chave **bbaaa**



# Árvore TRIE – Remover Chave

Remoção da chave **bbaaa**



# Aplicações para árvore TRIE

Solução do problema de busca de chaves semelhantes.

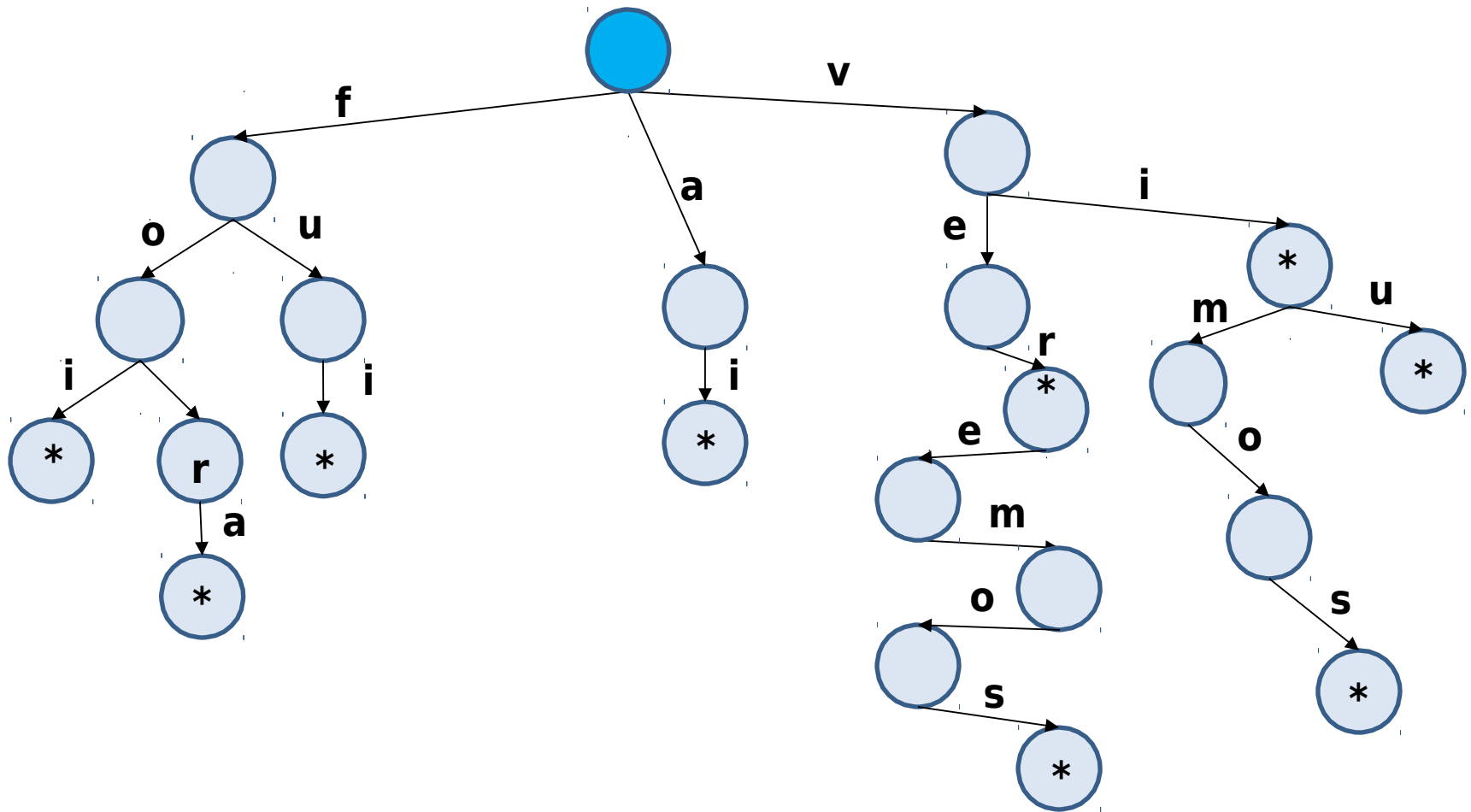
Por exemplo:

- Corretor ortográfico (word);
- Correção de chaves de busca (google);



# Árvore TRIE – Exercícios

1) Quais chaves/palavras estão representadas nesta TRIE?



# Árvore TRIE – Exercícios

2) Crie uma árvore TRIE com a seguinte frase:

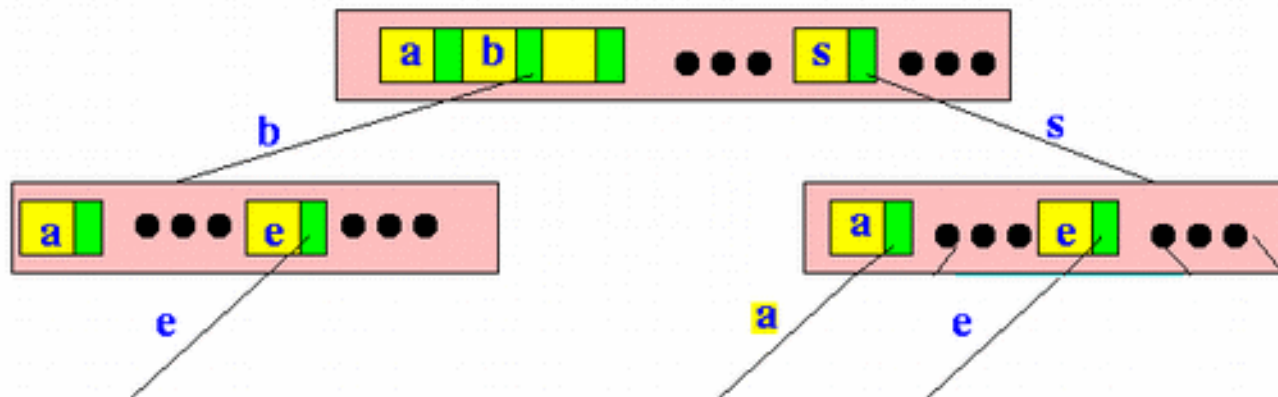
- O rato roeu a roupa do rei de Roma e a rainha de raiva roeu o resto. O rei então ficou furioso e brigou com a rainha.

3) Remova as seguintes chaves da árvore do exercício 2:

- rei, rainha, roeu, rato, então, furioso.

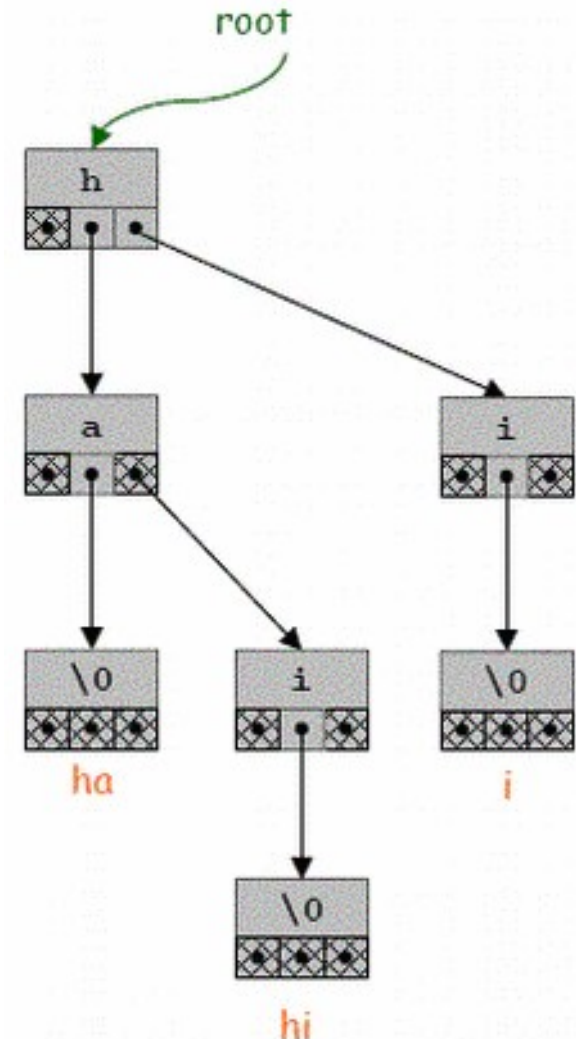
# Árvore TRIE – Implementação

- A implementação mais simples: **R-WAY**
  - Cada nó contém todos os valores do alfabeto mais 1 símbolo especial para determinar se é uma chave.
- Há desperdício de espaço.



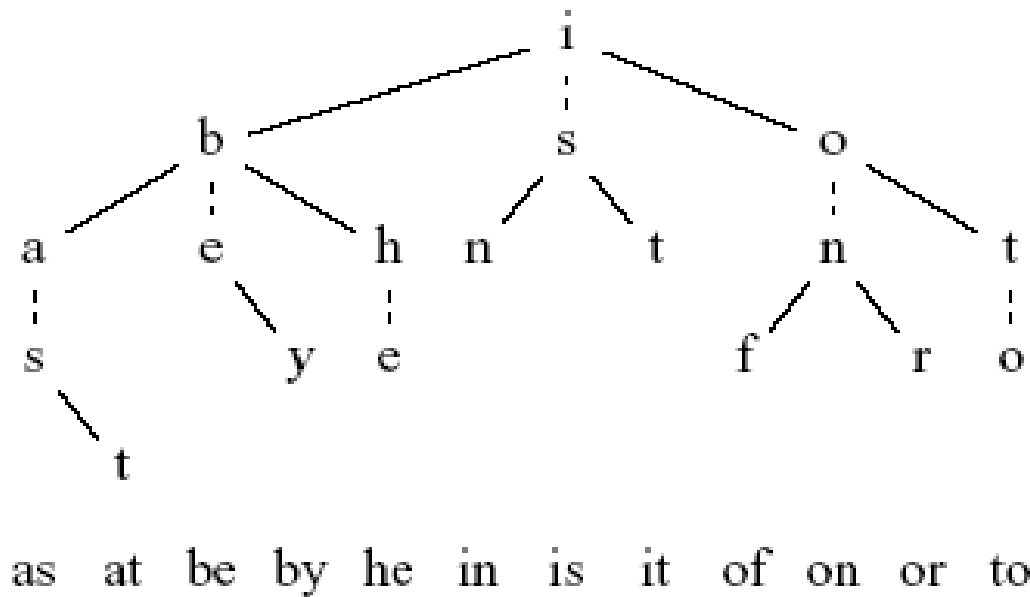
# Árvore TRIE – Implementação

- **TST**- Ternary Search Tree (Árvore de busca composta por três partes)
- Resolve o problema de desperdício de espaço.
- Cada nó aloca três ponteiros
  - **Centro:** caractere seguinte.
  - **Filho da esquerda:** caractere alternativo menor.
  - **Filho da direita:** caractere alternativo maior.
- Incluir root



# Árvore TRIE – Implementação - TST

- Para a árvore ficar balanceada as chaves devem estar ordenadas.
- Insere as chaves a partir da chave central (semelhante a busca binária)



# Árvore PATRICIA

**P** ractical  
**A** lgorithm  
**T** o  
**R** etrieve  
**I** nformation  
**C** oded  
**I** n  
**A** lphanumeric

Algoritmo prático para recuperar informações alfanuméricas

# Árvore PATRICIA

---

- Definida em 1968 por Donald R. Morrison
- Também conhecida como TRIE Compactada Binária.
- **Caminhos que possuem nós com apenas 1 filho são agrupados em uma única aresta**

# Árvore PATRICIA

---

- Os NÓS contém:
  - **y**: caractere que deve ser comparado
  - **x**: quantidade de posições que se deve avançar na chave para efetuar a comparação.
- Se menor ou igual a **y** avança a esquerda, se maior que **y** avança a direita.
- As chaves válidas encontram-se nas folhas da árvore.

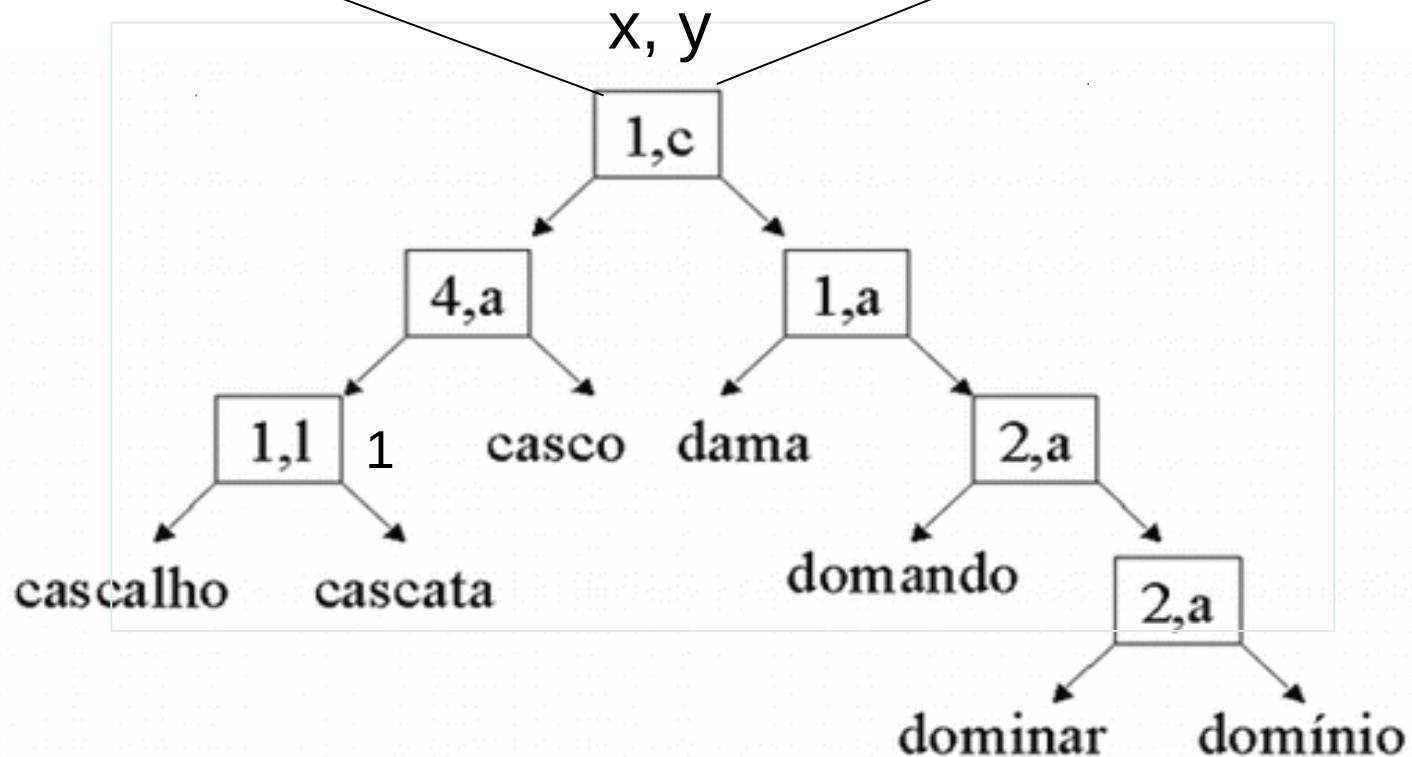


# Árvore PATRICIA

Exemplo de Representação:

Avance x dígitos

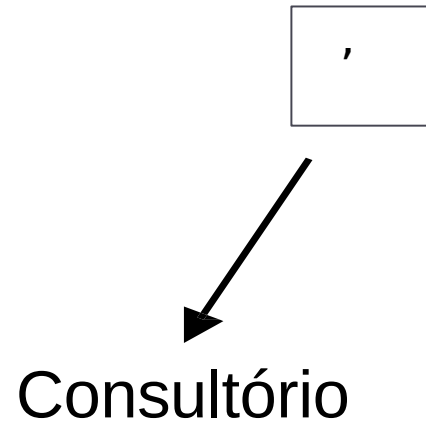
Compare com y



# Árvore PATRICIA – Inserir Chave

---

**Palavra 1:** Consultório



# Árvore PATRICIA – Inserir Chave

Palavra 1: Consultório

Palavra 2: Consultar

Consultório, Consulta

Encontrada Diferença  
no 8º Caractere



8,a

$X = 8$

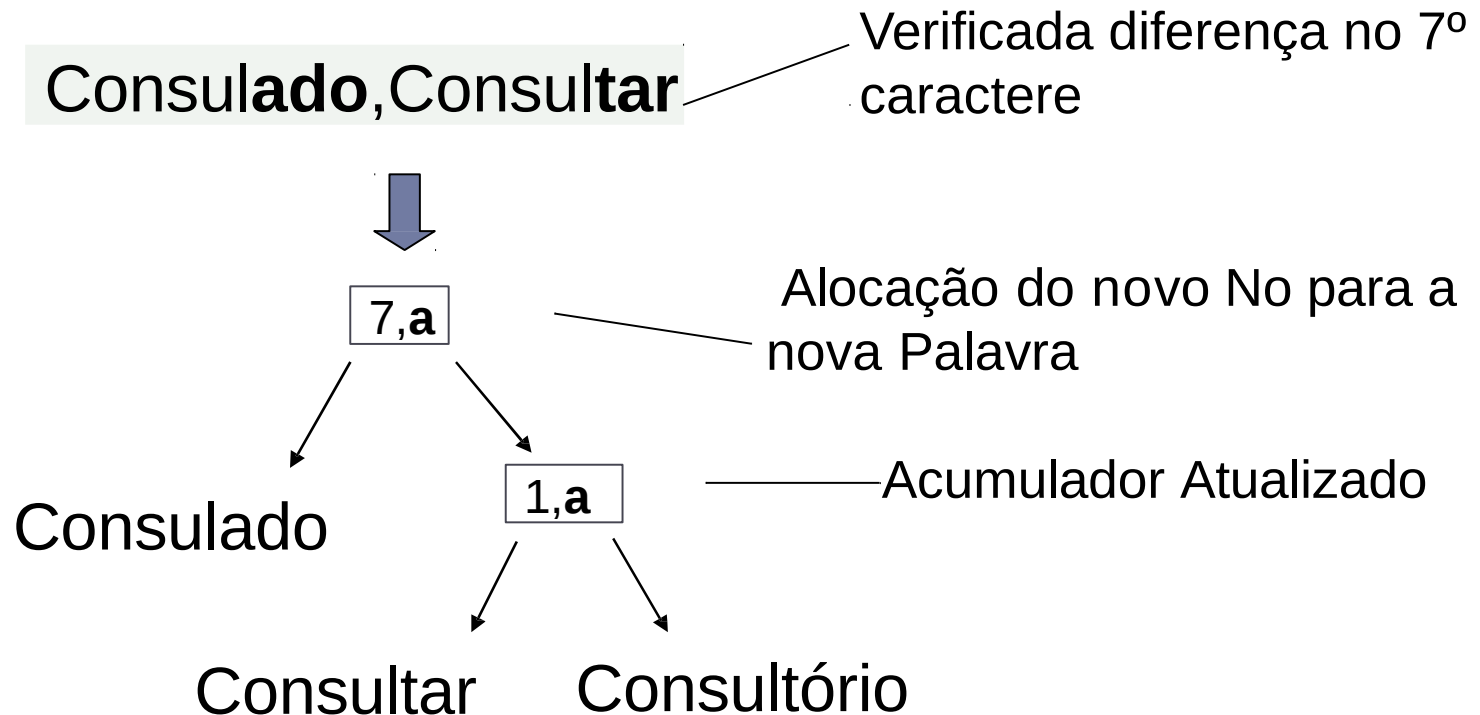
$Y = a \ (a < o)$

Consultar

Consultório

# Árvore PATRICIA – Inserir Chave

## Palavra 3: Consulado



# Árvore PATRICIA – Inserir Chave

## Palavra 4: Consultado

Consultado,Consultado

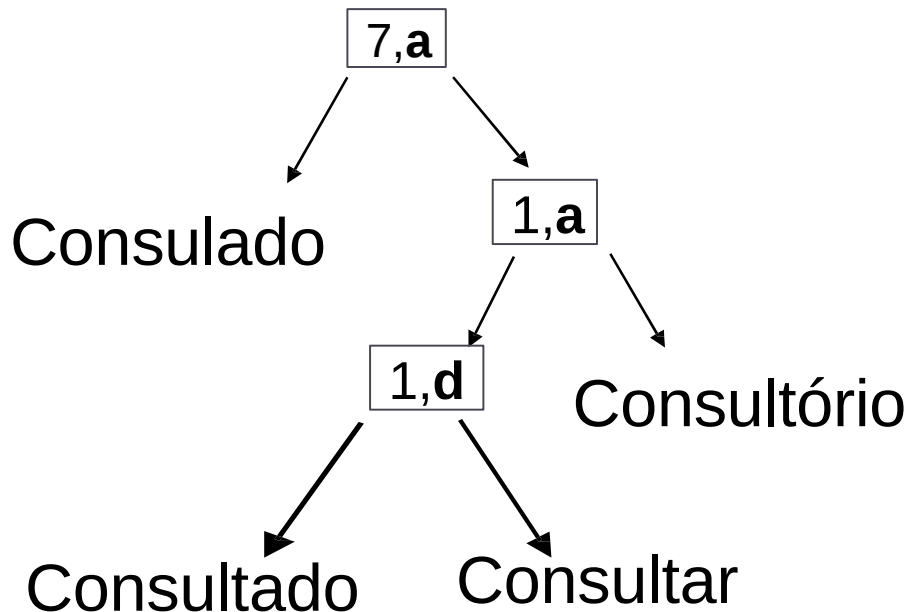
Consultado,a

Consultado,Consultar

t > a no 7º caractere

a = a no 8º caractere (direita)

d < r no 9º caractere



Alocação do novo No para distinguir consultado de consultar

# Árvore PATRICIA – Buscar Chave

## Sequencia de passos:

Comparar o caractere na posição **x** com o caractere **y**.

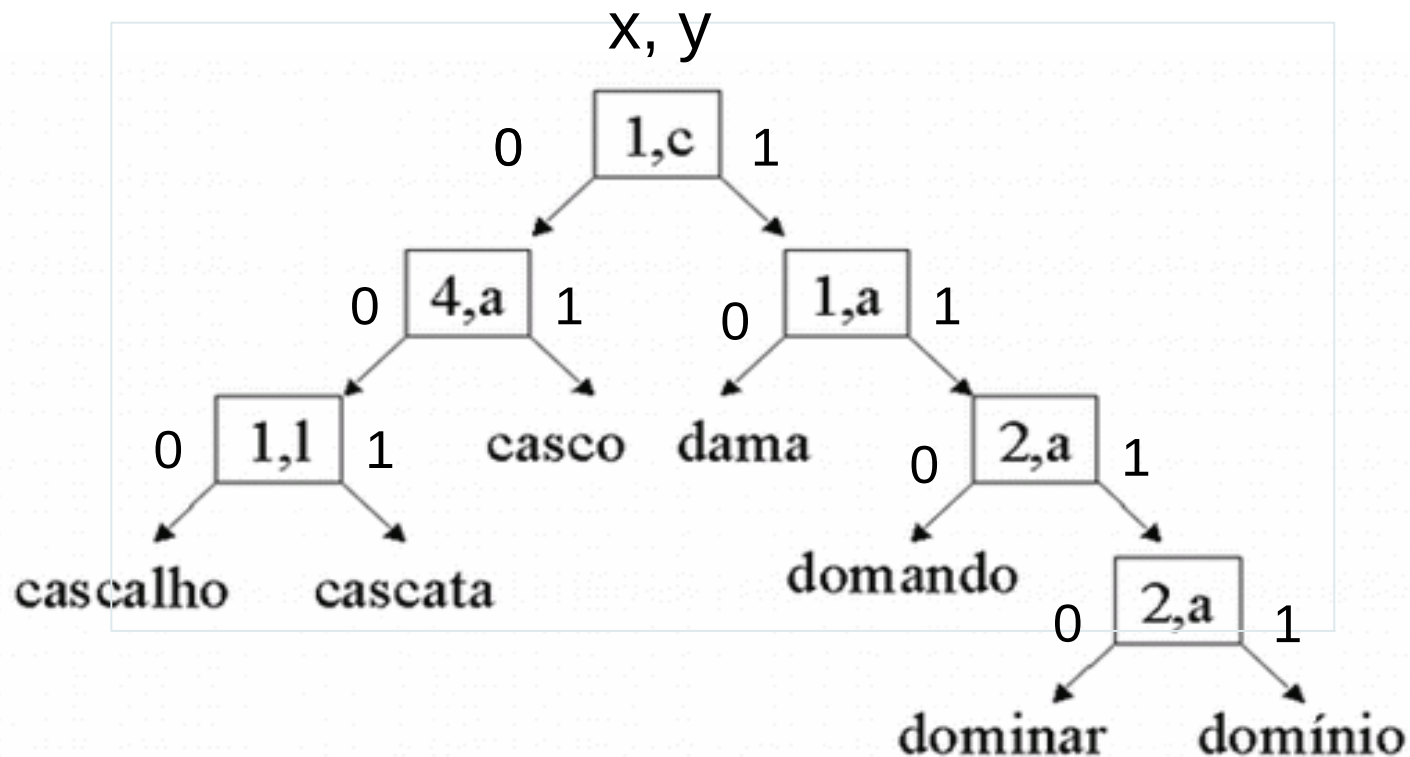
- 1) Se menor ou igual: Segue ramo a esquerda;
- 2) Se maior: Segue ramo a direita

Repete isso até chegar numa folha

# Árvore PATRICIA – Buscar Chave

**Buscar as palavras:**

Cascata, domínio, carro



# Árvore PATRICIA – Remover Chave

## Sequencia de Passos:

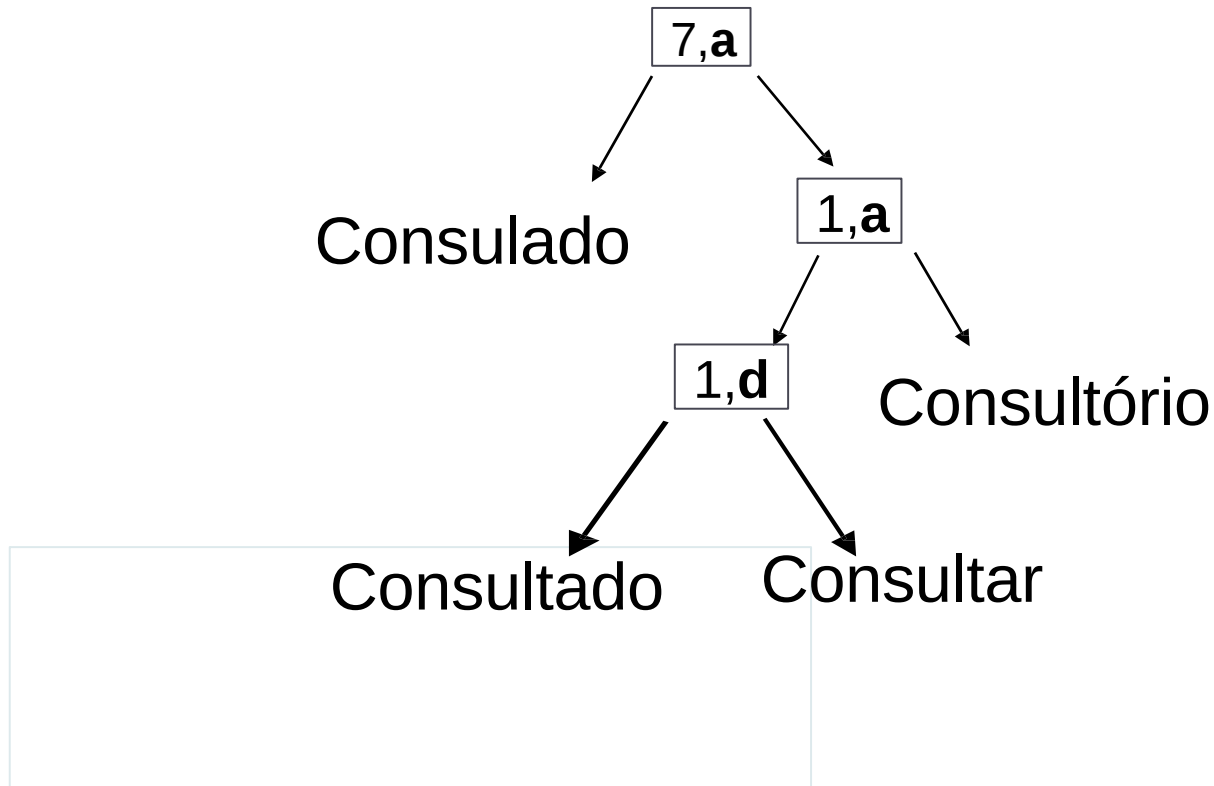
- 1) Buscar e apagar a chave da Árvore;
- 2) O pai da chave deve ser apagado;
- 3) Soma-se o valor do **Campo Avançar** do Nó Pai a Todos os nós Filhos.





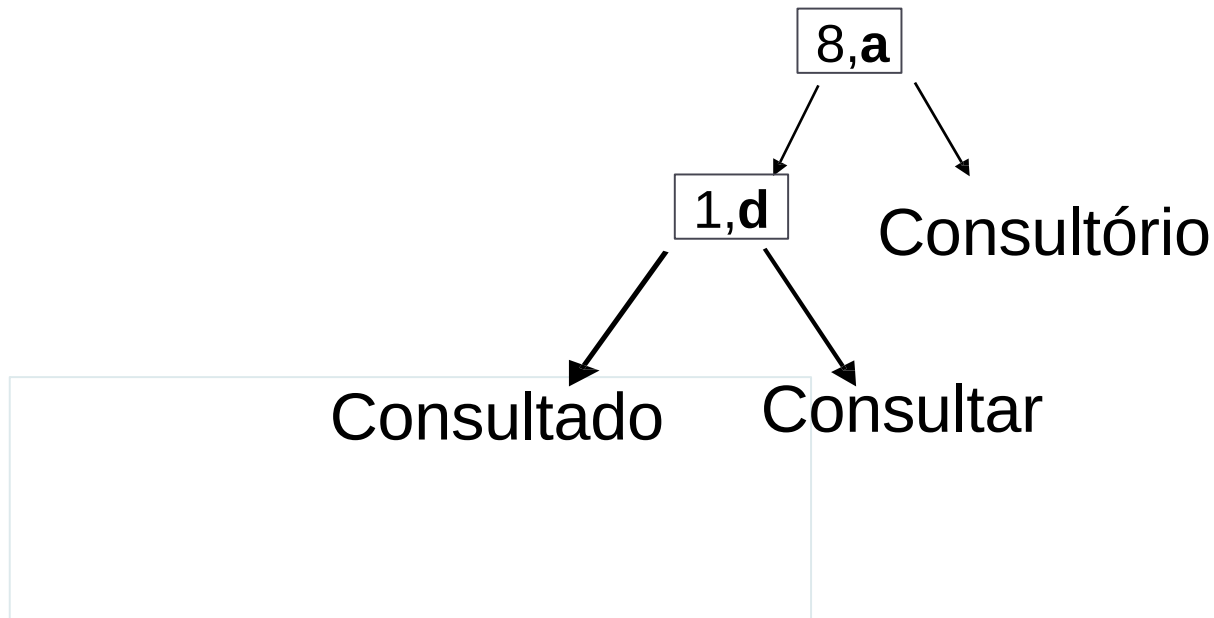
# Árvore PATRICIA – Remove Chave

Remover Consultado



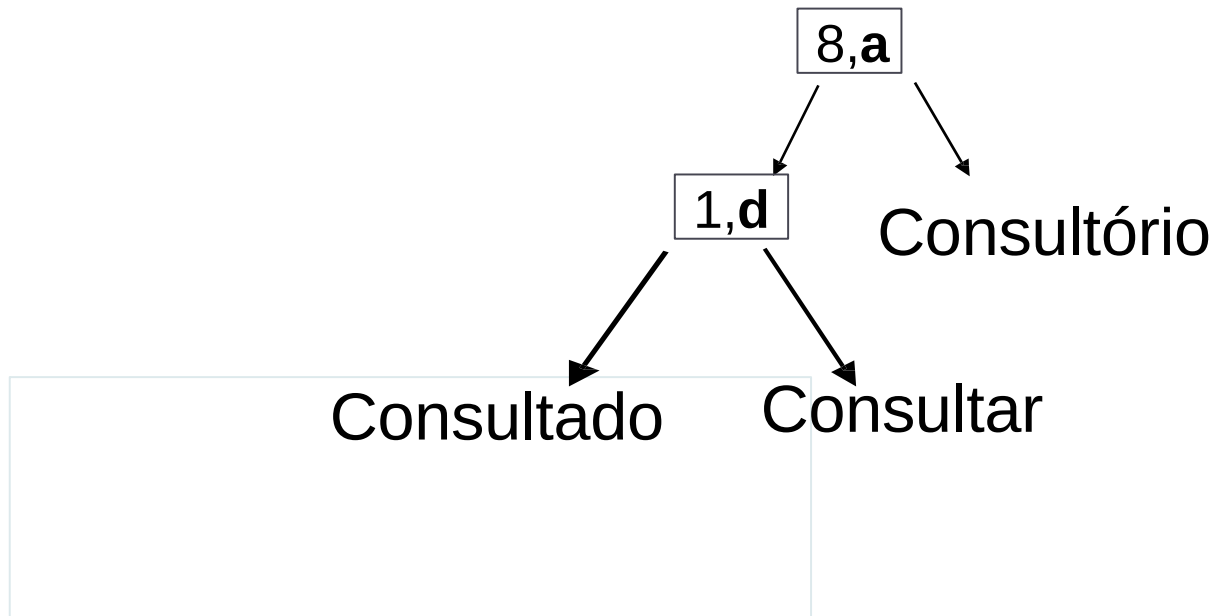
# Árvore PATRICIA – Remove Chave

Remove Consultado



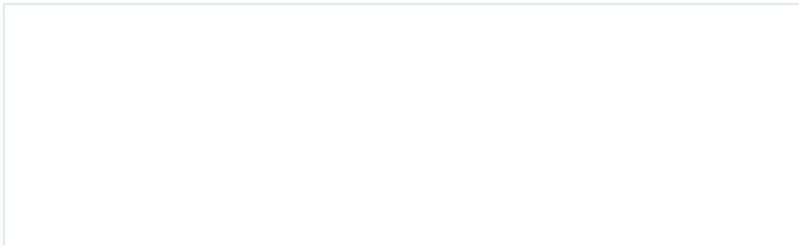
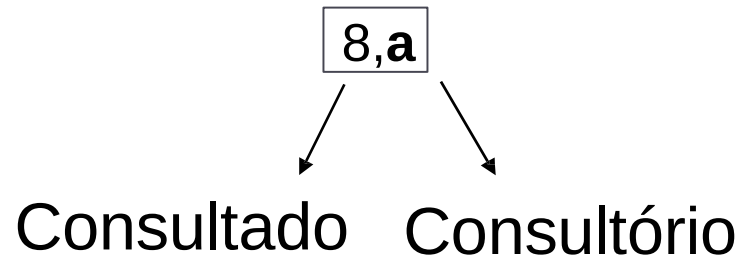
# Árvore PATRICIA – Remove Chave

Remove Consular



# Árvore PATRICIA – Remove Chave

Remove Consular



# Exercícios

- 4) Represente a árvore de exercício 2 na formato R-Way e no formato TST.
- 5) Crie uma árvore PATRICIA com a seguinte frase:
- O rato roeu a roupa do rei de Roma e a rainha de raiva roeu o resto. O rei então ficou furioso e brigou com a rainha.
- 6) Remova as seguintes chaves da árvore do exercício 5:
- rei, rainha, roeu, rato, então, furioso.

# Referências

SZWARCFITER, J. L., MARKENZON, L. Estruturas de Dados e seus Algoritmos. 3a ed. Rio de Janeiro: LTC, 2010.

TENENBAUM, A. M., LANGSAM, Y., AUGENSTEIN, M. J. Estruturas de Dados Usando C. São Paulo: Makron, 1995.

LEISERSON, C. E., RIVEST, R. L., CORMEN, T. H., STEIN, C. Algoritmos – Teoria e prática. Rio de Janeiro: Campus, 2002.

<http://www.inf.ufrgs.br/~cagmachado/INF01124/t3.htm>