

# ÁRVORES TRIE

Prof. Patrícia Noll de Mattos

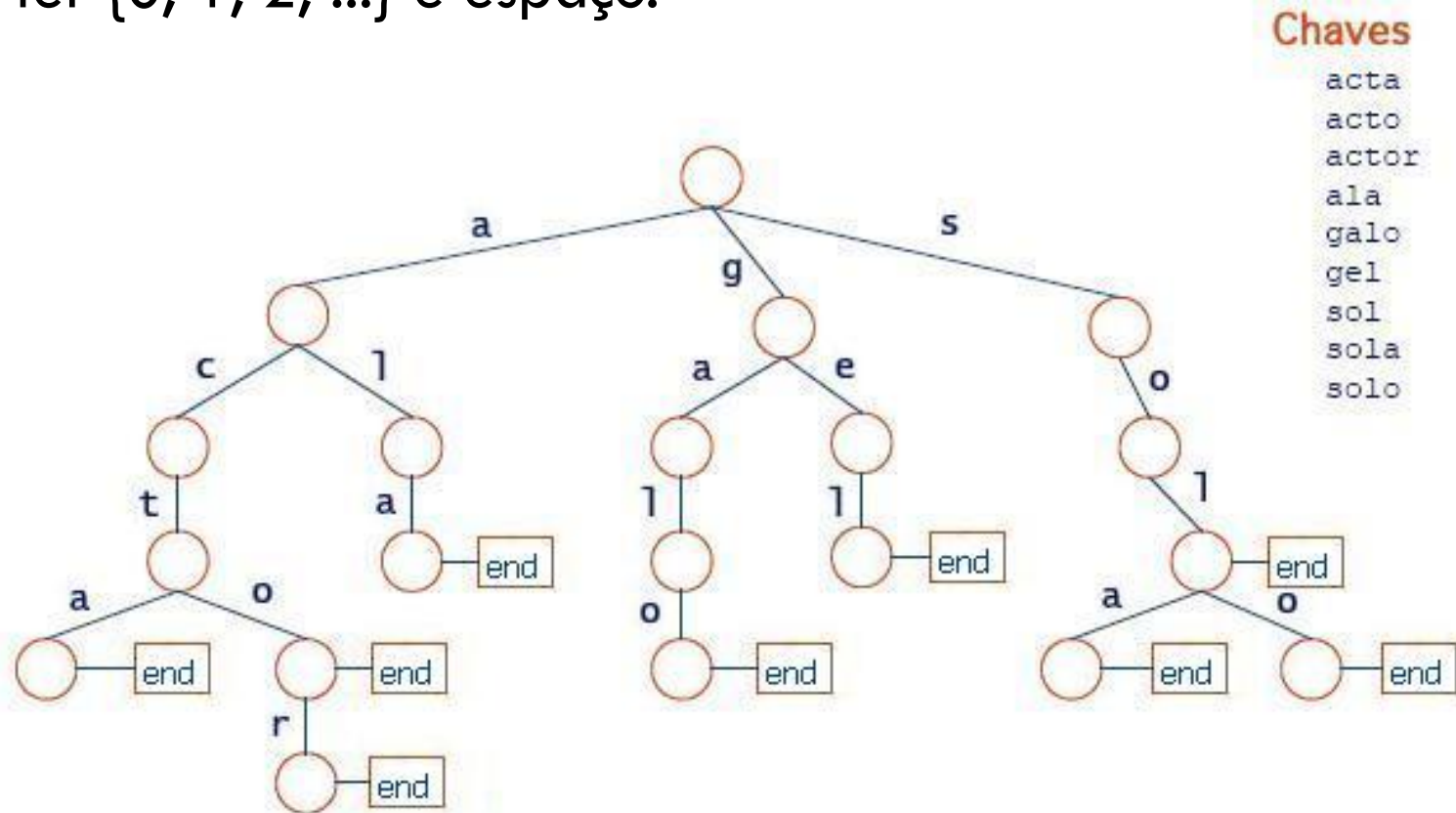
Ilustrações: Gilberto Irajá.

# Árvore Trie

- O nome “TRIE” vem de **Retrieval** (recuperação de dados);
- Pronuncia-se “TRAI” ou “TRI” para distinguir de “Tree”;
- **Idéia geral:** usar partes das CHAVES como caminho busca.

# Características:

- Árvore ordenada...
- Chave em **geral caracteres**: {A, B, C, D, ...}, porém, podemos ter {0, 1, 2, ...} e espaço.

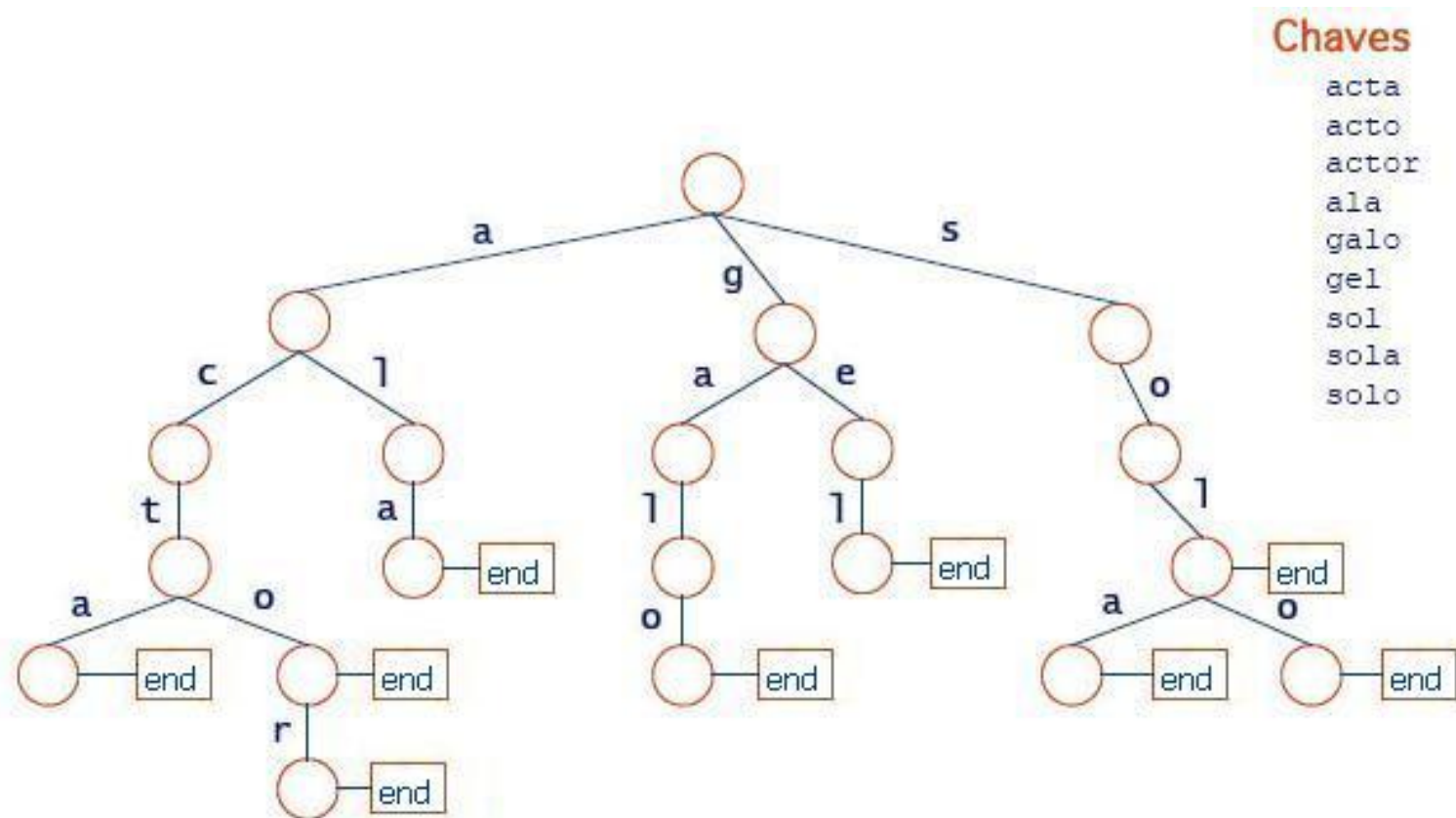


# Características...

- Nenhum **nodo contém a chave** e sim uma parte dela.
- Possui um **alfabeto** e um **dicionário** armazenado na própria estrutura.
- **Nodos internos: armazenam as letras do alfabeto, podendo ter filhos.**
- **d:** número de letras do alfabeto.
- **Nodos externos:** letras finais de cada palavra.
- Os padrões apenas serão considerados se terminarem em um nodo externo.

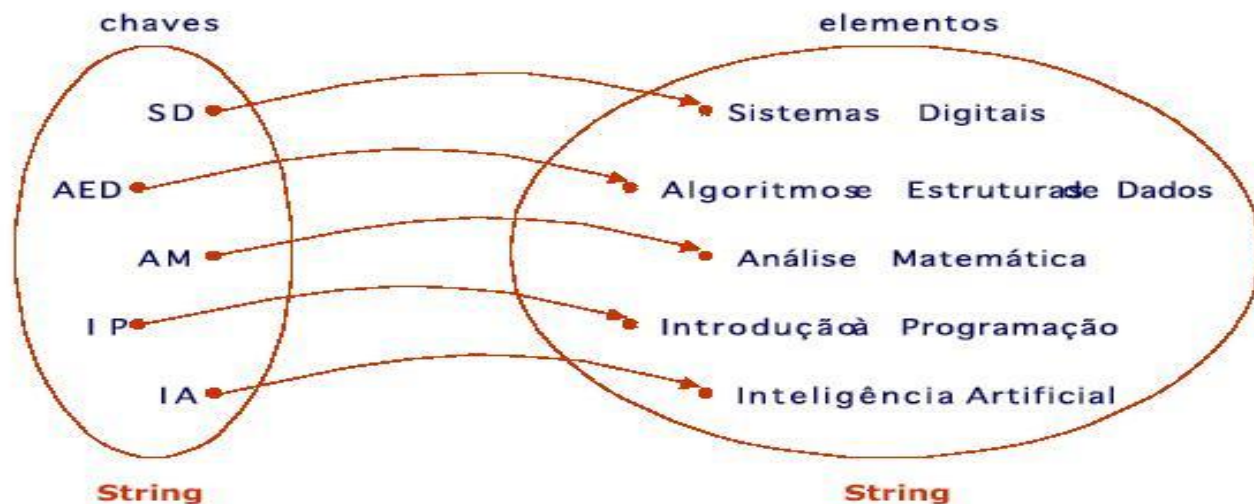
# Características...

- **Descendentes** do mesmo nodo com mesmo **prefixo..**
- **Raiz:** cadeia vazia.



# Armazenamento Externo...

- O **caminho da raiz** para qualquer outro nodo é um prefixo de uma string.
- Cada **nível** que se desce corresponde a avançar um elemento na chave.
- Em geral associam-se chaves a elementos ou registros, como na tabela hash.
- Existe um número de referência ao final da chave que aponta para o elemento

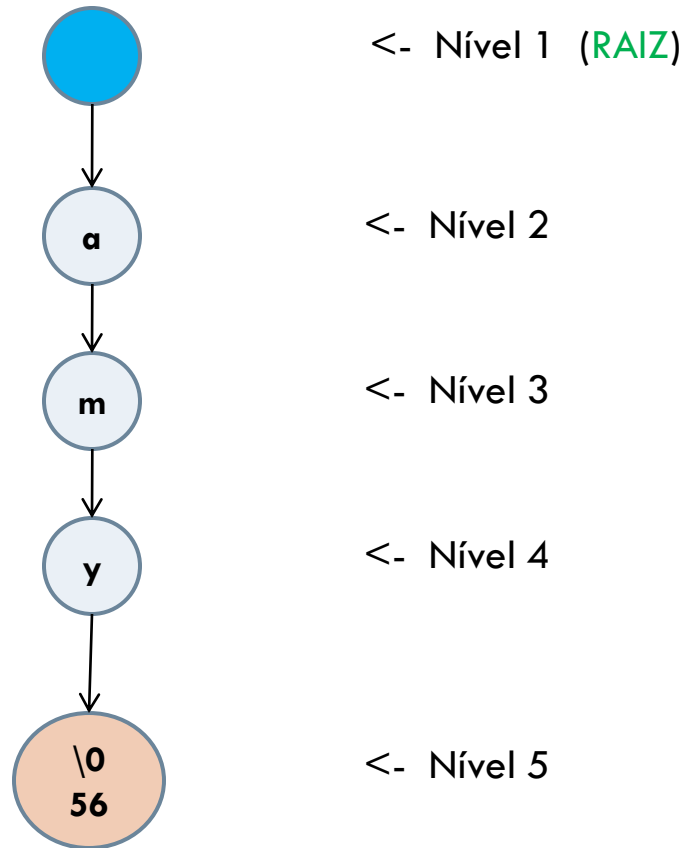


# Aplicações...

- Dicionários (telefone celular);
- Corretores Ortográficos;
- Programas para compreender Linguagem Natural;
- Auto-preenchimento:
  - ▣ browsers,
  - ▣ e-mail,
  - ▣ linguagens de programação.

# Exemplo

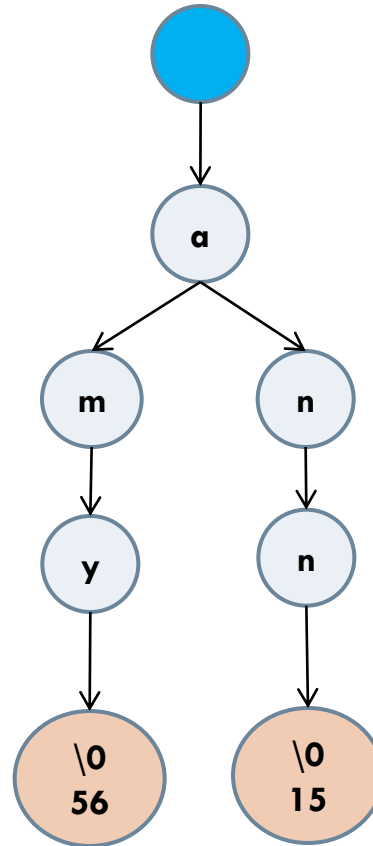
□ Inserindo amy, 56





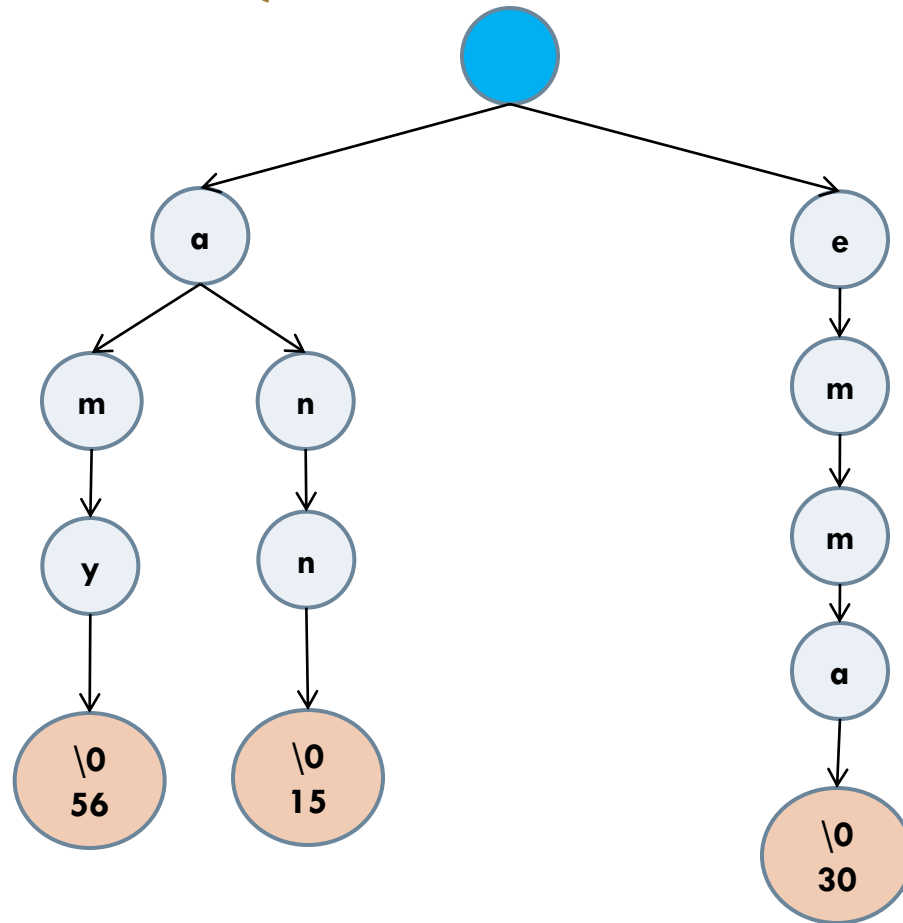
# Exemplo

□ Inserindo ann, 15



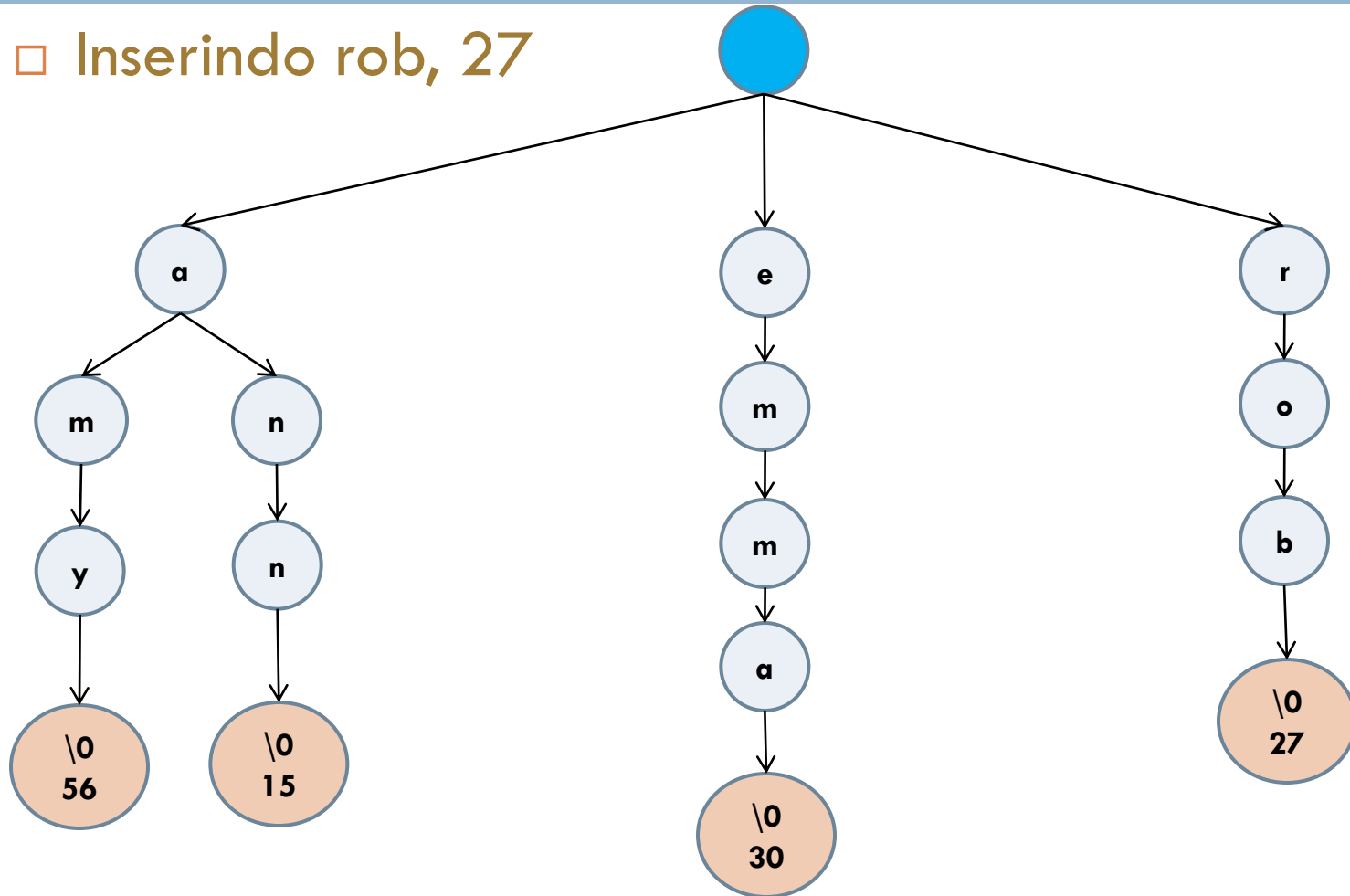
# Exemplo

□ Inserindo emma, 30



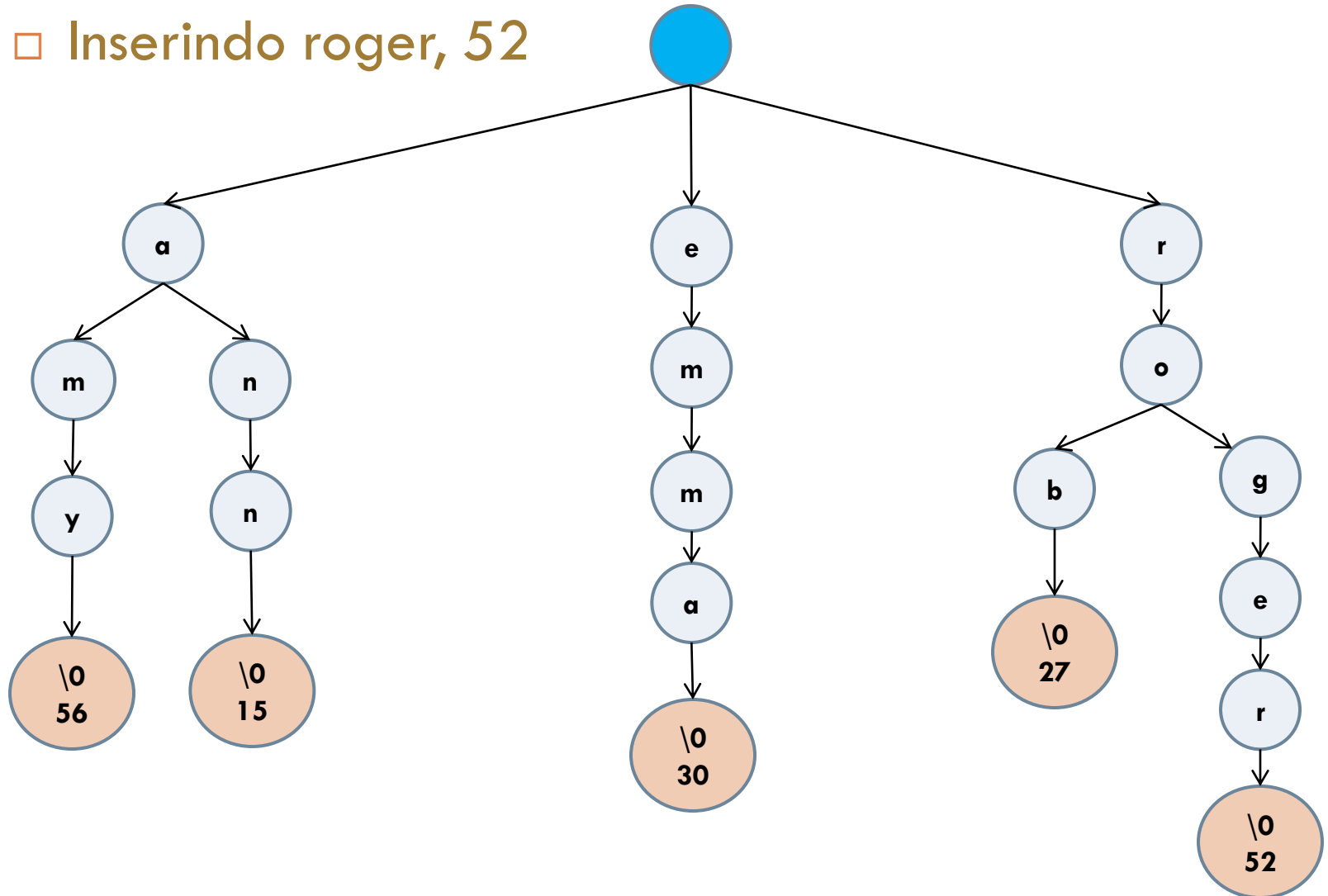
# Exemplo

□ Inserindo rob, 27



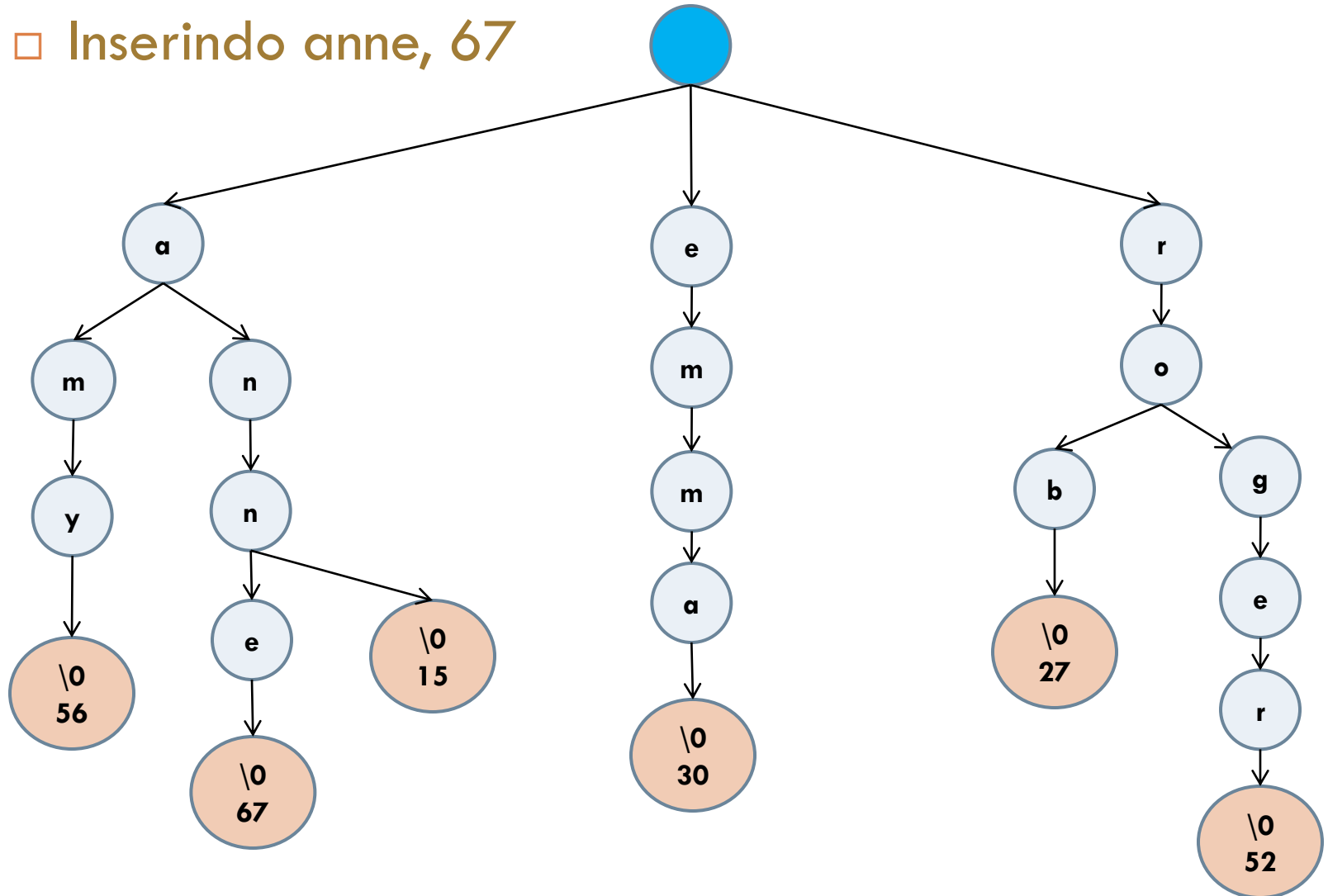
# Exemplo

□ Inserindo roger, 52



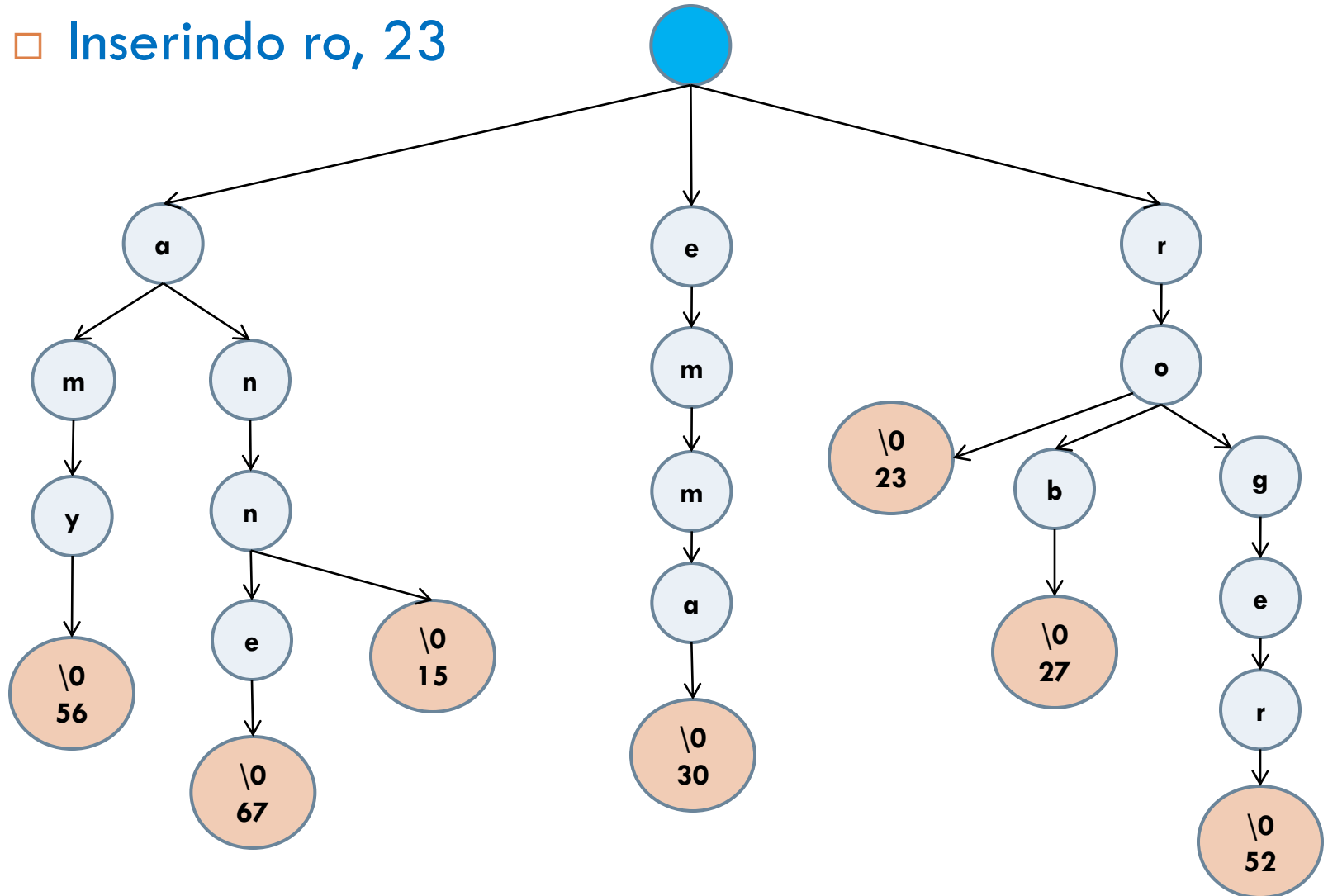
# Exemplo

□ Inserindo anne, 67



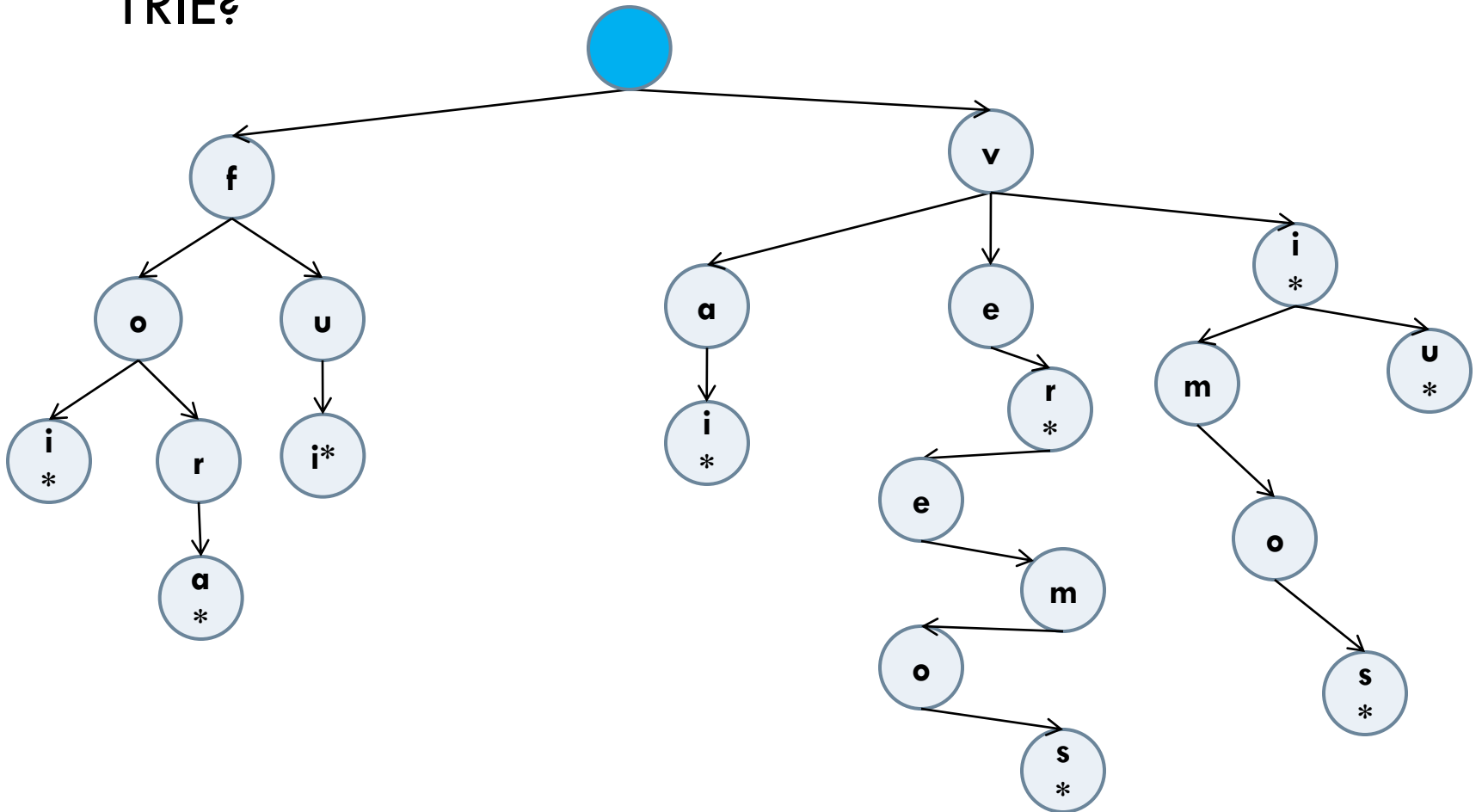
# Exemplo

□ Inserindo ro, 23



## Exercício/Exemplo

- ❑ Quais chaves/palavras estão representadas neste TRIE?



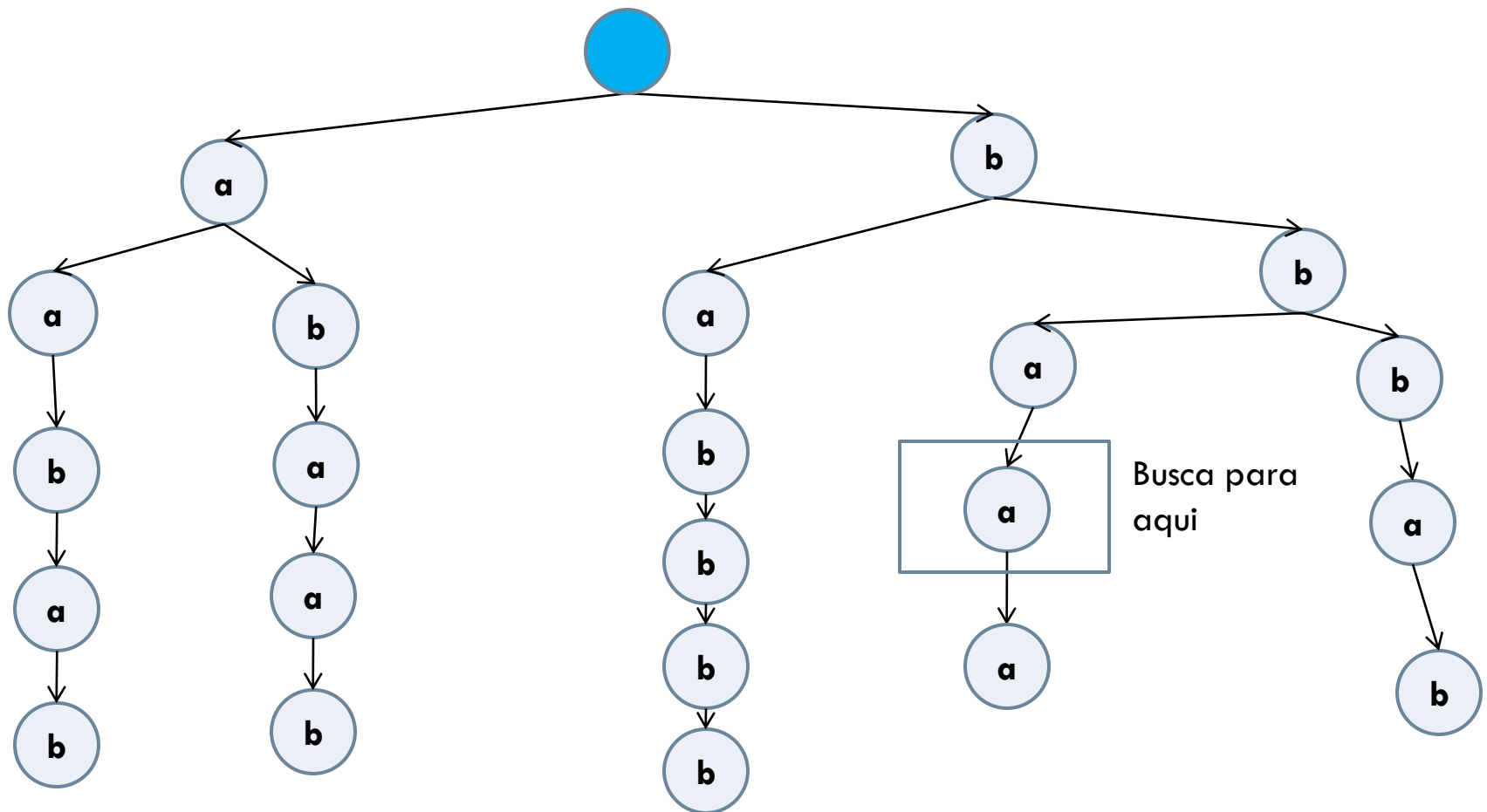
# Operação de Inserção

- Faz-se uma busca pela chave a ser inserida. Se ela **já existir** no TRIE, então, finaliza-se a operação;
- Caso contrário, é recuperado o nodo até onde acontece a **maior substring da palavra** a ser inserida;
- O restante dos seus caracteres são adicionados no TRIE a partir daquele nodo.



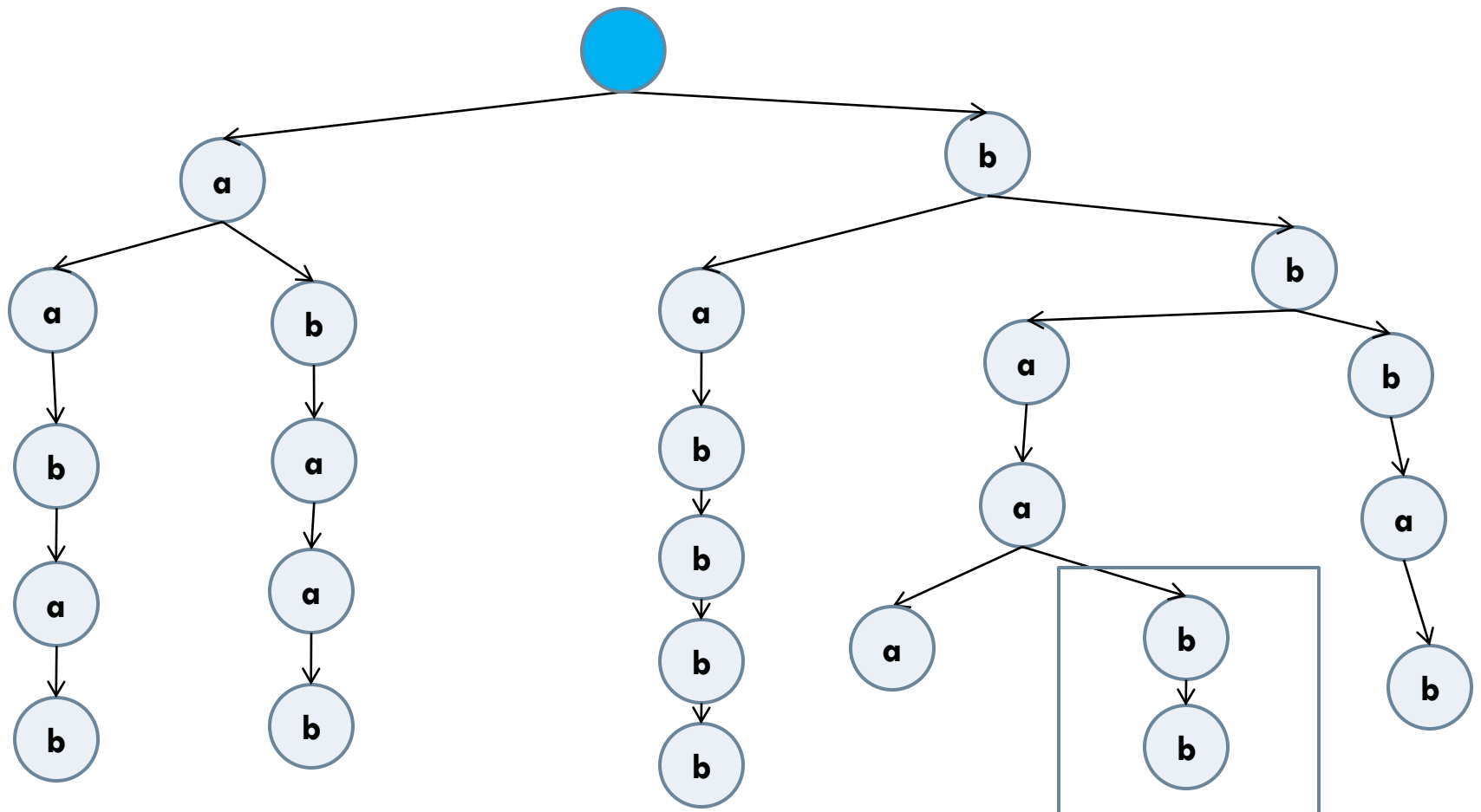
# Operação de Inserção

□ Inserção de bbaabb



# Operação de Inserção

□ Inserção de bbaabb



# É Membro?

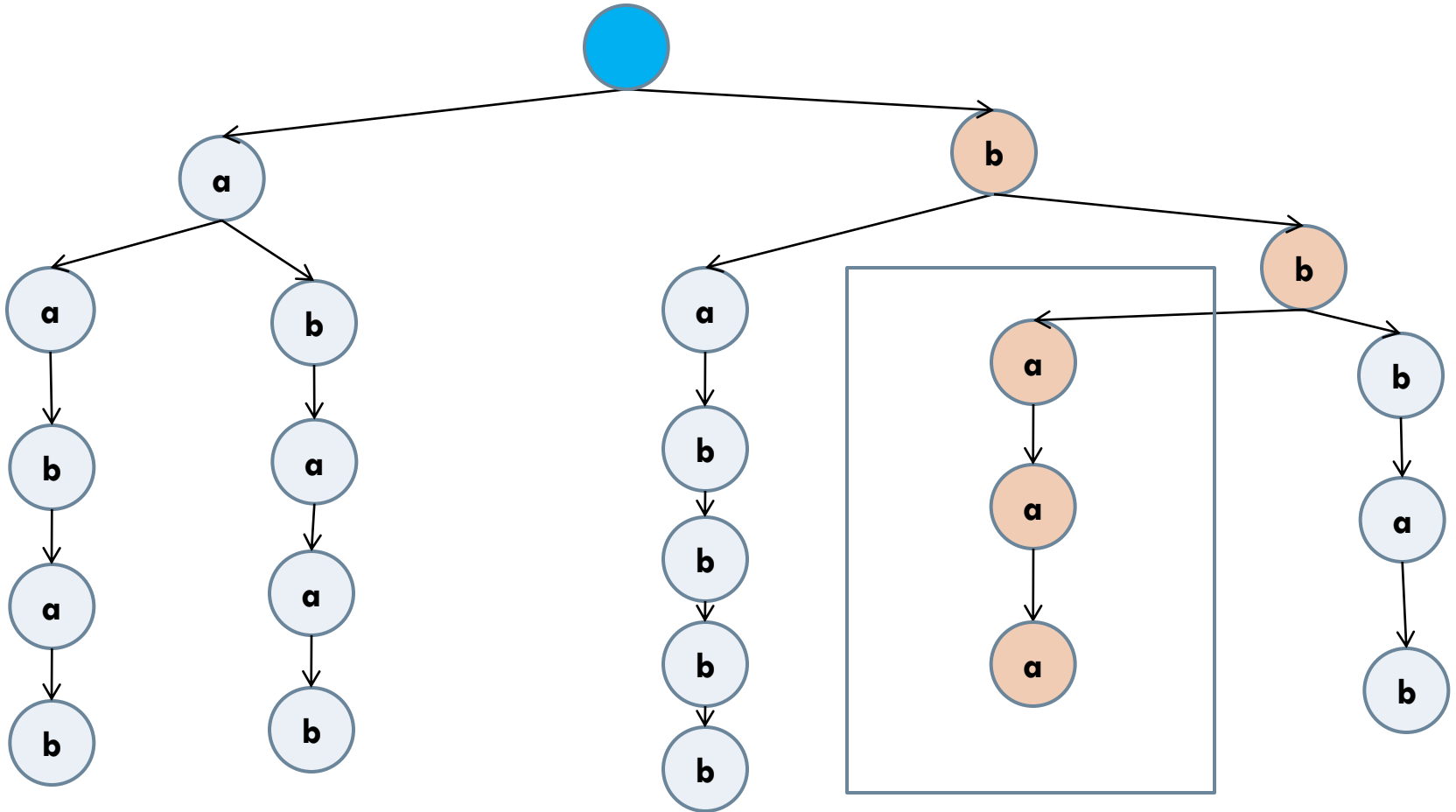
1. Busca no nível superior o nodo que confere com o primeiro caractere (corrente) da chave;
2. Se nenhum, retorna FALSO;  
Senão
3. Se o caractere que confere é `\0`  
Retorna VERDADEIRO;  
Senão
4. Move para o subTRIE que confere com esse caractere;
5. Avança para o próximo caractere na chave;
6. Vá para o passo 1.

# Remoção

- Busca-se o **nodo** que representa o final da palavra a ser removida;
- São removidos os nodos que possuem apenas **um filho** pelo caminho ascendente;
- A remoção é concluída quando se encontra um nodo com **mais de um** filho.

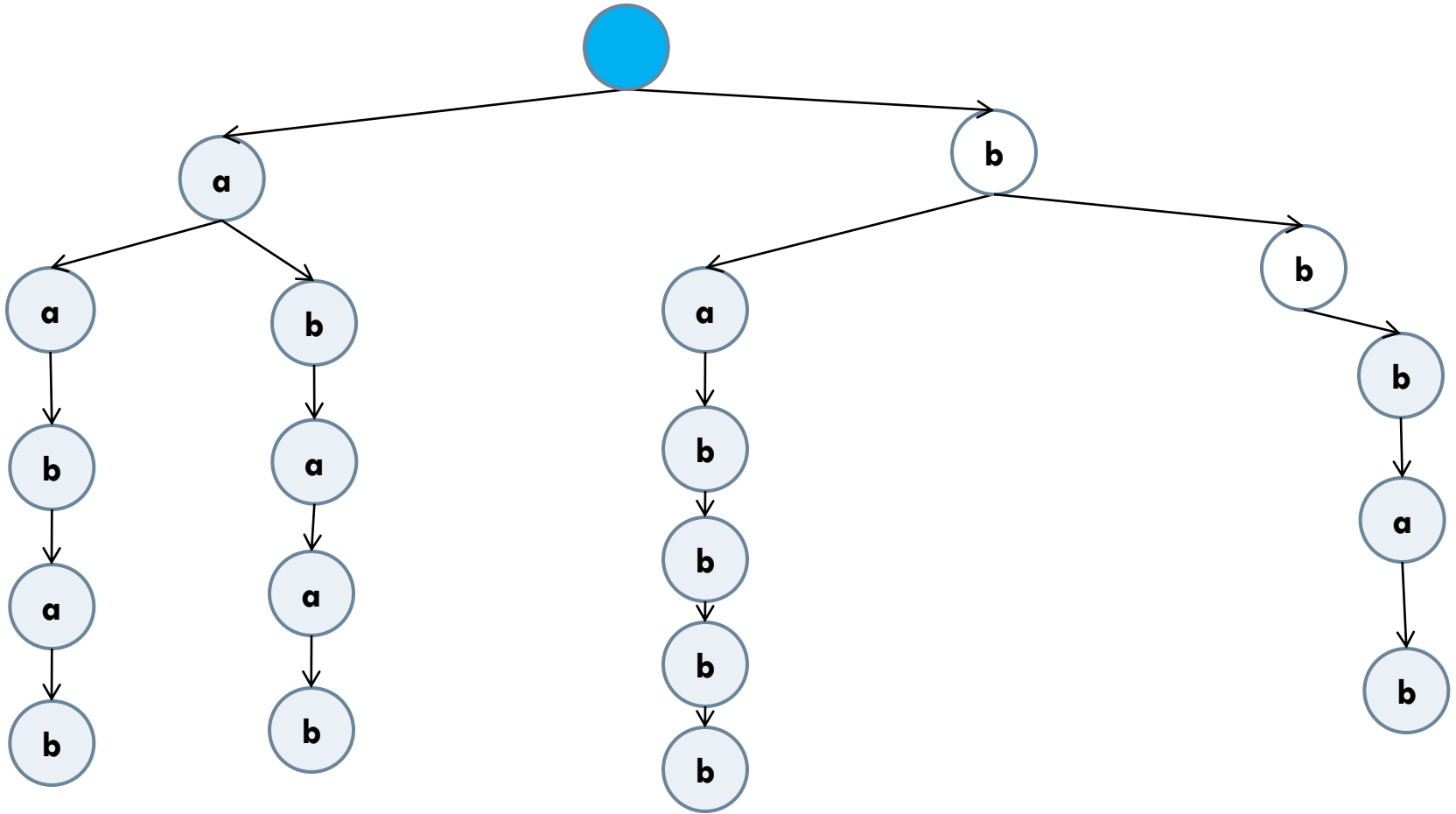
# Remoção

- Remoção de bbaaa



# Remoção

- Remoção de bbaaa



# Complexidade – Análise de Algoritmo

- A **altura da árvore** é igual ao comprimento da chave mais longa;
- O tempo de execução das operações não depende do número de elementos da árvore;
- Complexidade no pior caso é  $O(AK)$ 
  - A = tamanho do alfabeto
  - K = tamanho da chave

# Tarefa Trie

1 - Demonstre a árvore TRIE (passo a passo) para as seguintes chaves, bem como seus significado:

MACA -> Utilizada para socorrer pessoas

MACHO -> Gênero encontrado na natureza

MATO -> Local com árvores e animais

BALA -> Doce conhecido entre crianças e “professores”

BANANA -> Fruta tropical

BALELA -> No sentido de enrolação

BALEIA -> Animal encontra no mar



# Tarefa Trie

2 – Como seria um **corretor ortográfico** ao procurar a chave MALA e MATA, referente o TRIE anterior?

3 – Qual seria o resultado utilizando uma árvore TRIE de **auto preenchimento** baseado no TRIE do exercício 1? A partir das digitações abaixo.

B

BA

BAL

BALE

BALEL

# Referências Bibliográficas

- CORMEN, Thomas H. et al. **Introduction to algorithms**. 3. ed. Cambridge: MIT, 2009. xix. 1 292 p.
- LUZZARDI, P. R. G. **Lâminas de Estruturas de Dados**. Universidade Católica de Pelotas.