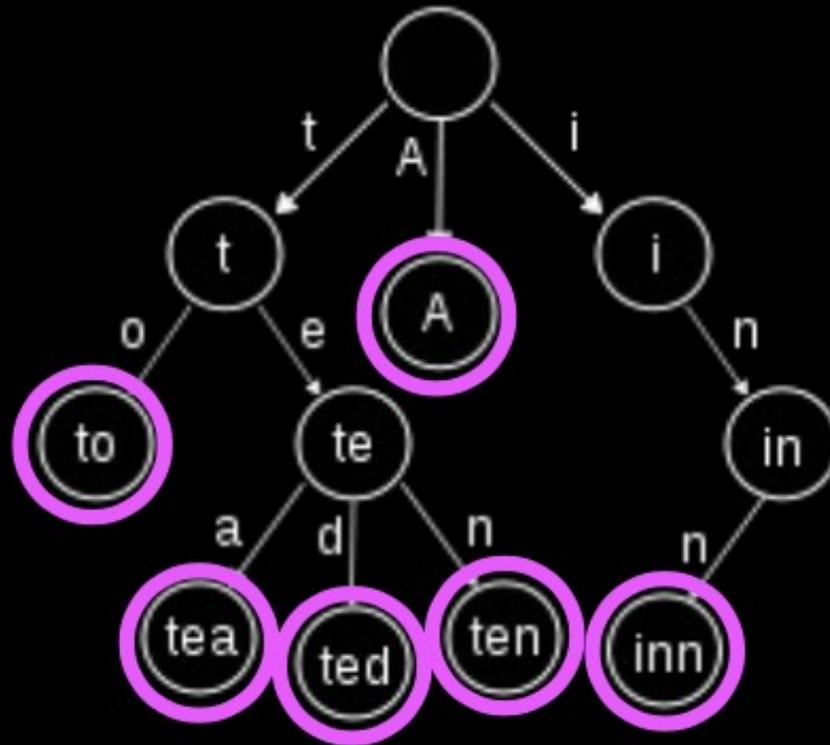


Árvores digitais

Palavras/Chaves:

- A
- to
- tea
- ted
- ten
- inn



Palavras com tamanho variável e ilimitado

Árvores digitais

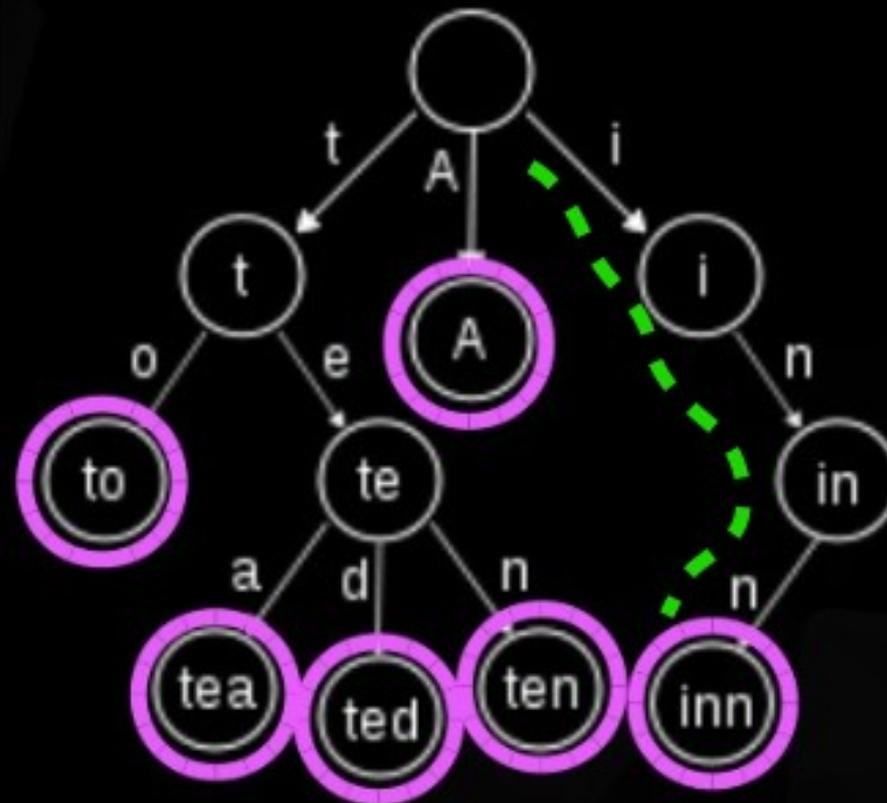
Uma árvore TRIE (ATRIE) é uma estrutura do tipo árvore ordenada que permite a recuperação

A ATRIE armazena um array associativo em que as chaves são normalmente cadeias de caracteres

A chave é uma sequência de símbolos pertencentes a um alfabeto.

As chaves são armazenadas nas folhas da árvore, os nós internos são parte do caminho de busca.

Árvores digitais

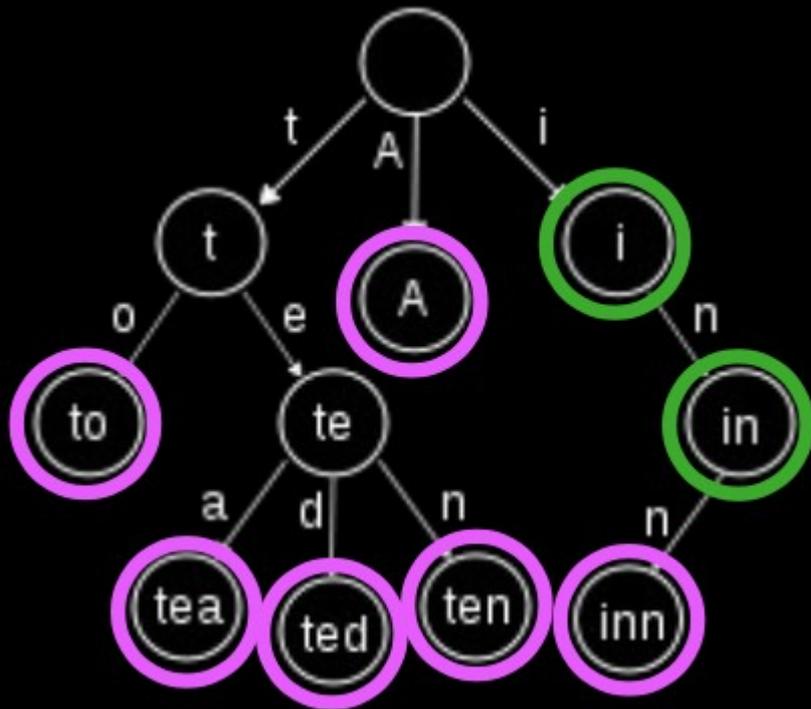


TRIE originado de '*Information reTRIEval*'



Claude Shannon, John McCarthy, Ed Fredkin and Joseph Weizenbaum (1966) [6]

Árvore Trie

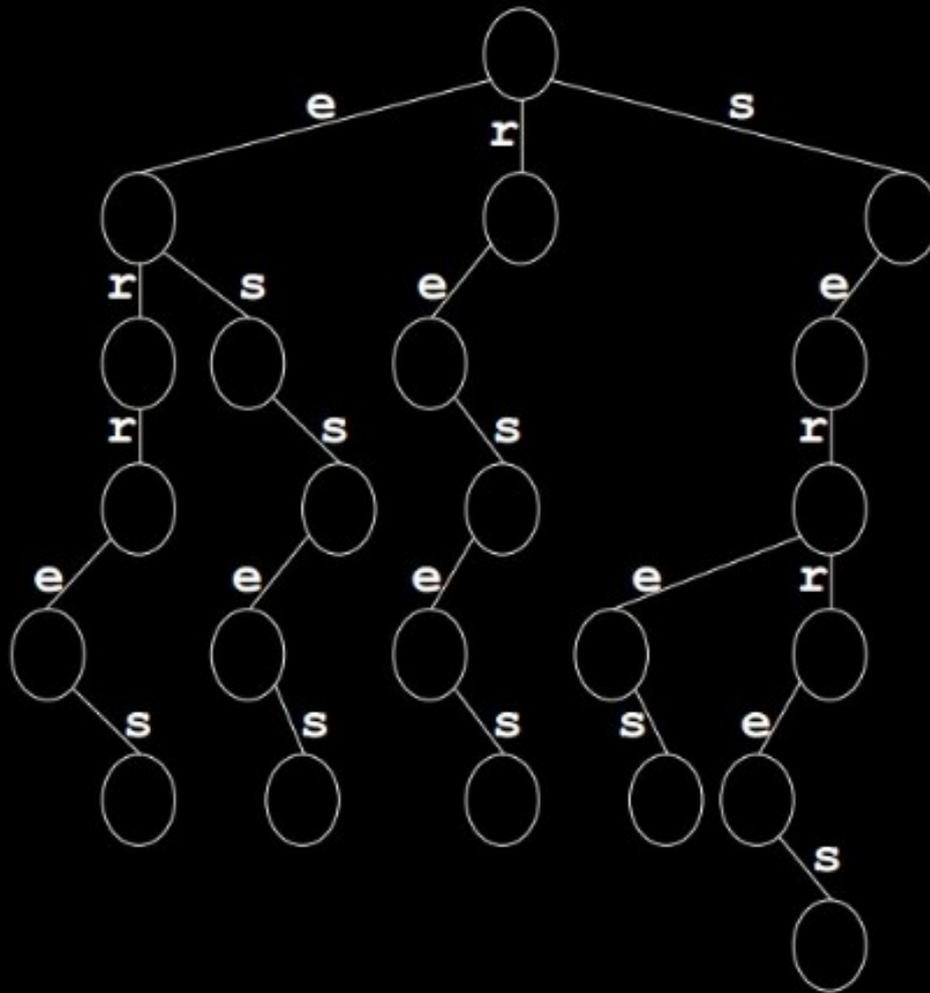


A busca se inicia na raiz e continua com a subárvore associado ao símbolo / caractere procurado até chegar a uma folha (ou nó interno)

Exemplo de Árvore Trie

Alfabeto:

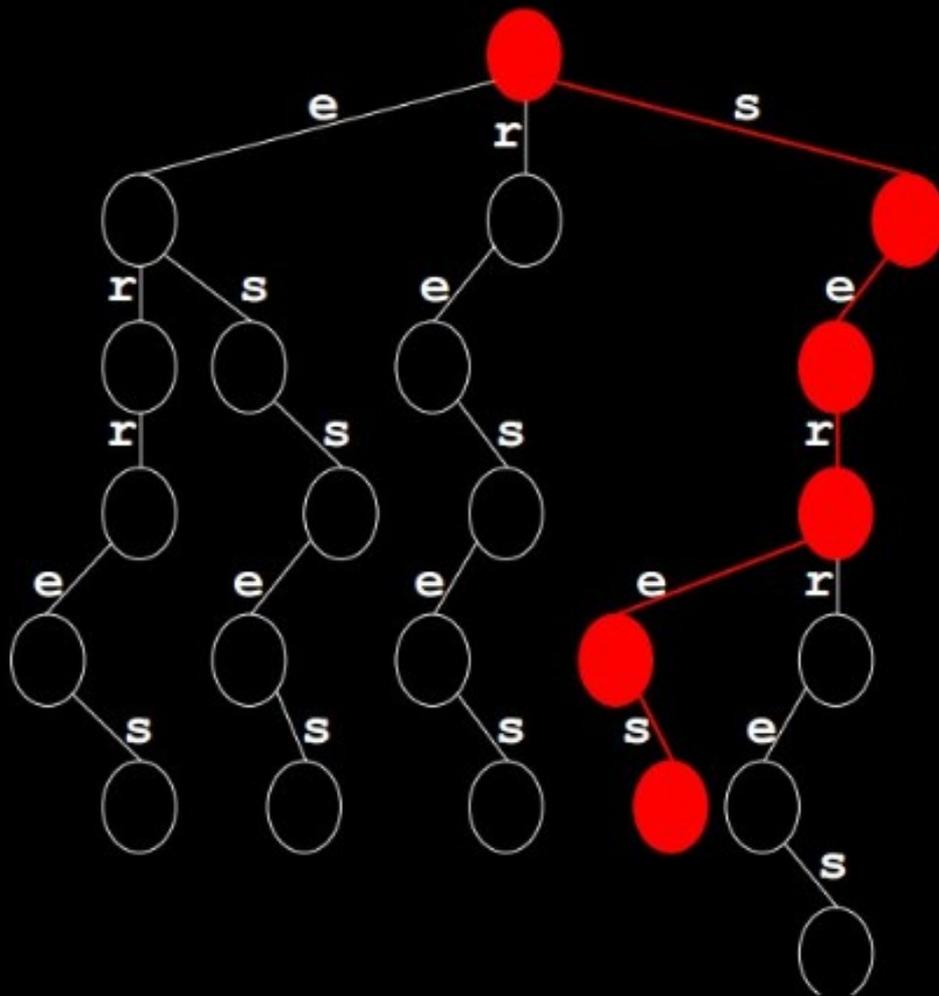
{e, r, s}



Exemplo de Árvore Trie

Alfabeto:

{e, r, s}



erre
erres
es
esse
esses
se
ser
serre
re
res
rese
reses
serres
seres

Exemplos de alfabetos e chaves

Alfabetos

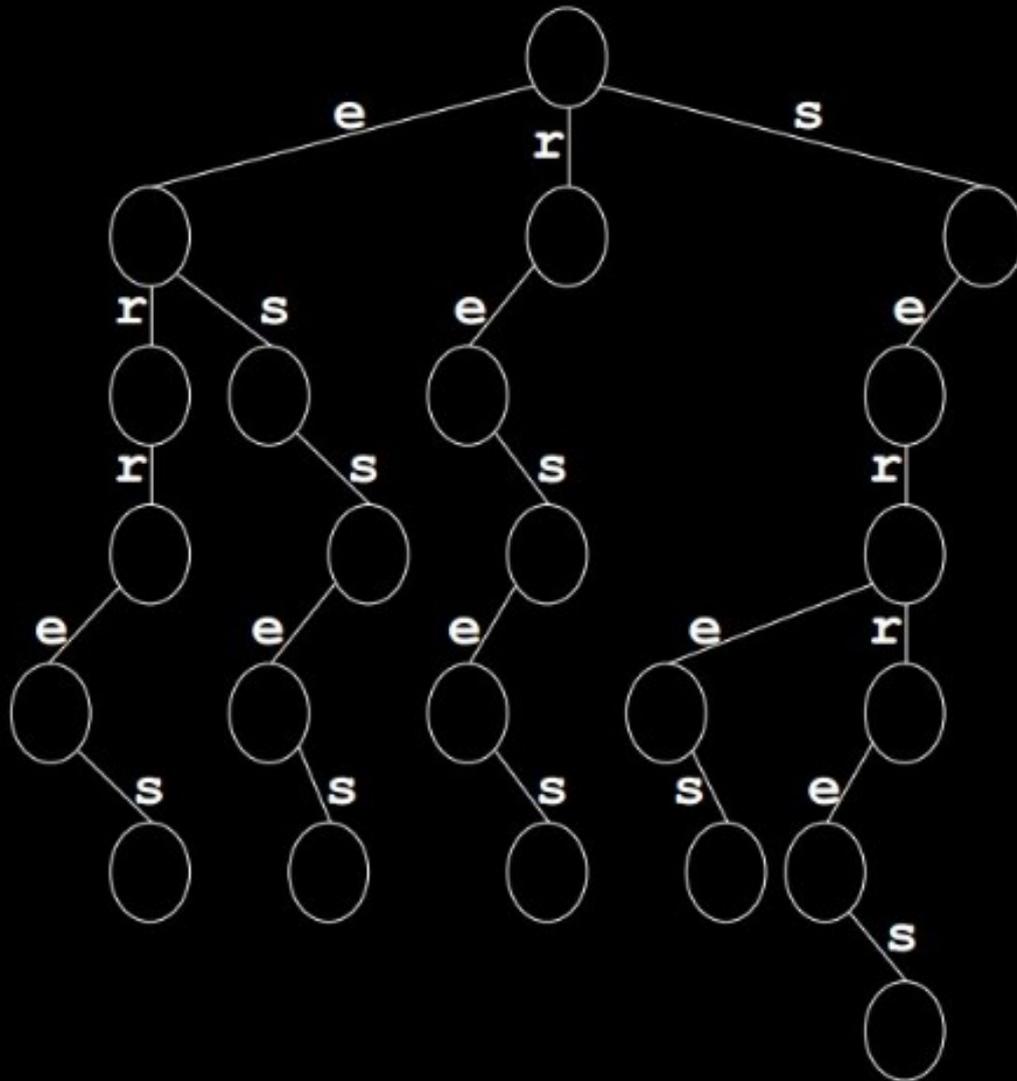
$\{0,1\}$, $\{A, B, C, D, E, \dots, Z\}$, $\{0,1,2,3,4,5, \dots, 9\}$

Chaves

ABABBBABABA 19034717 Maria
01010101000000000101000000001010

A chave é determinada pela posição na árvore.

Árvore Trie (árvore n-ária)



Árvore ternária;

Alfabeto:

$\{e, r, s\} \quad e < r < s$

$m = 3$

$S = \{erre, erres, es, esse, esses, se, ser, serre, re, res, rese, reses, serres, seres\}$

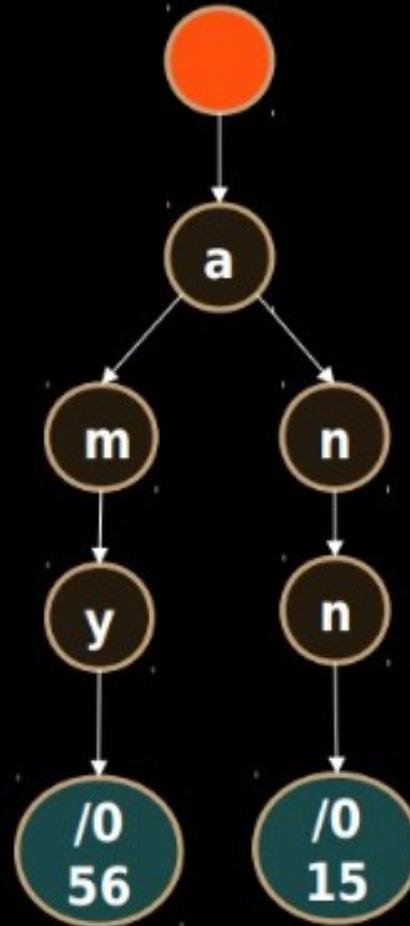
Montando uma árvore Trie

- amy 56



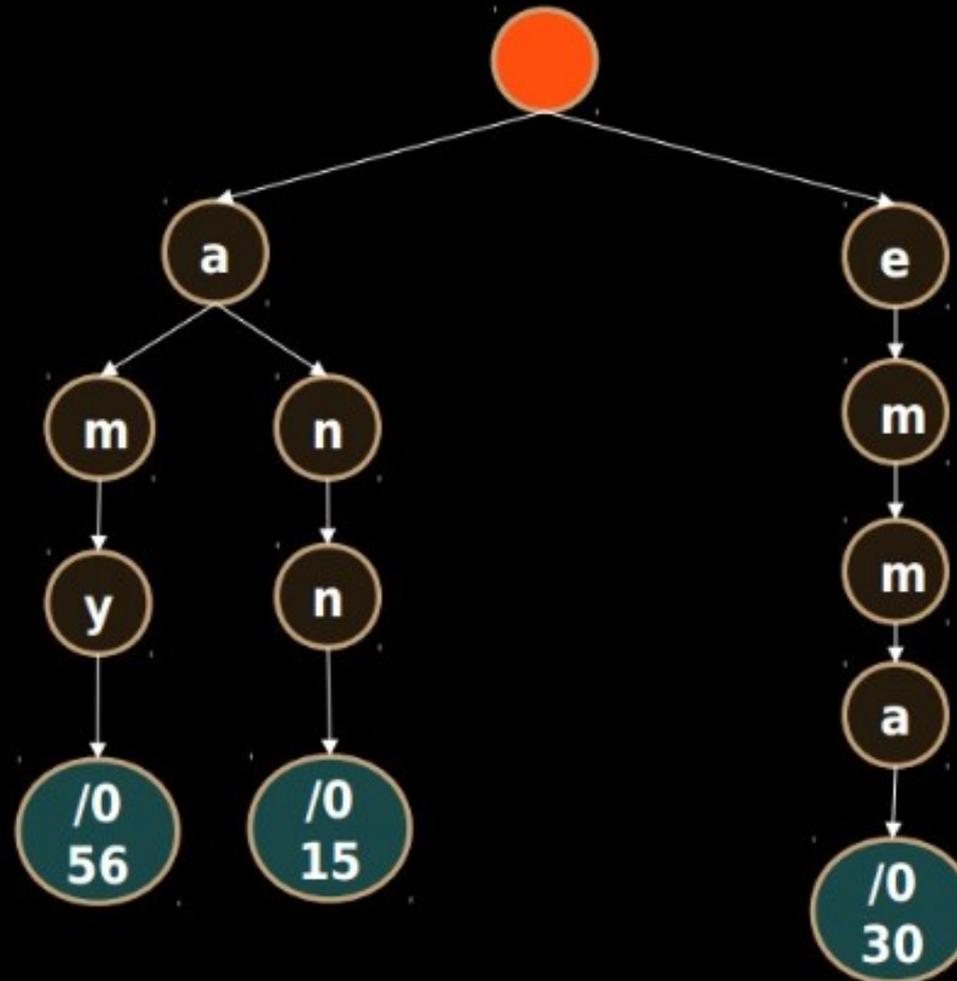
Montando uma árvore Trie

- ann 15



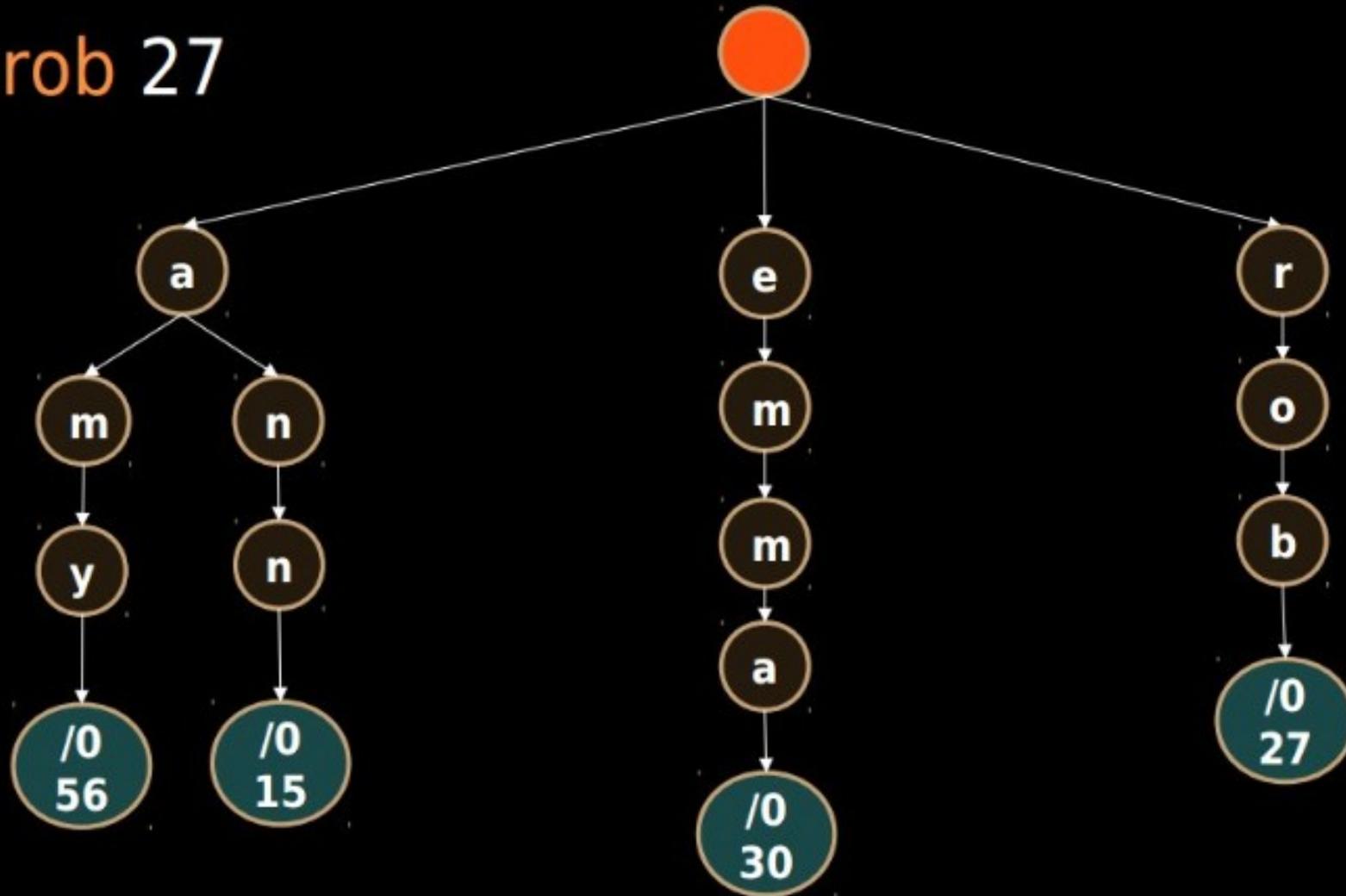
Montando uma árvore Trie

- emma 30



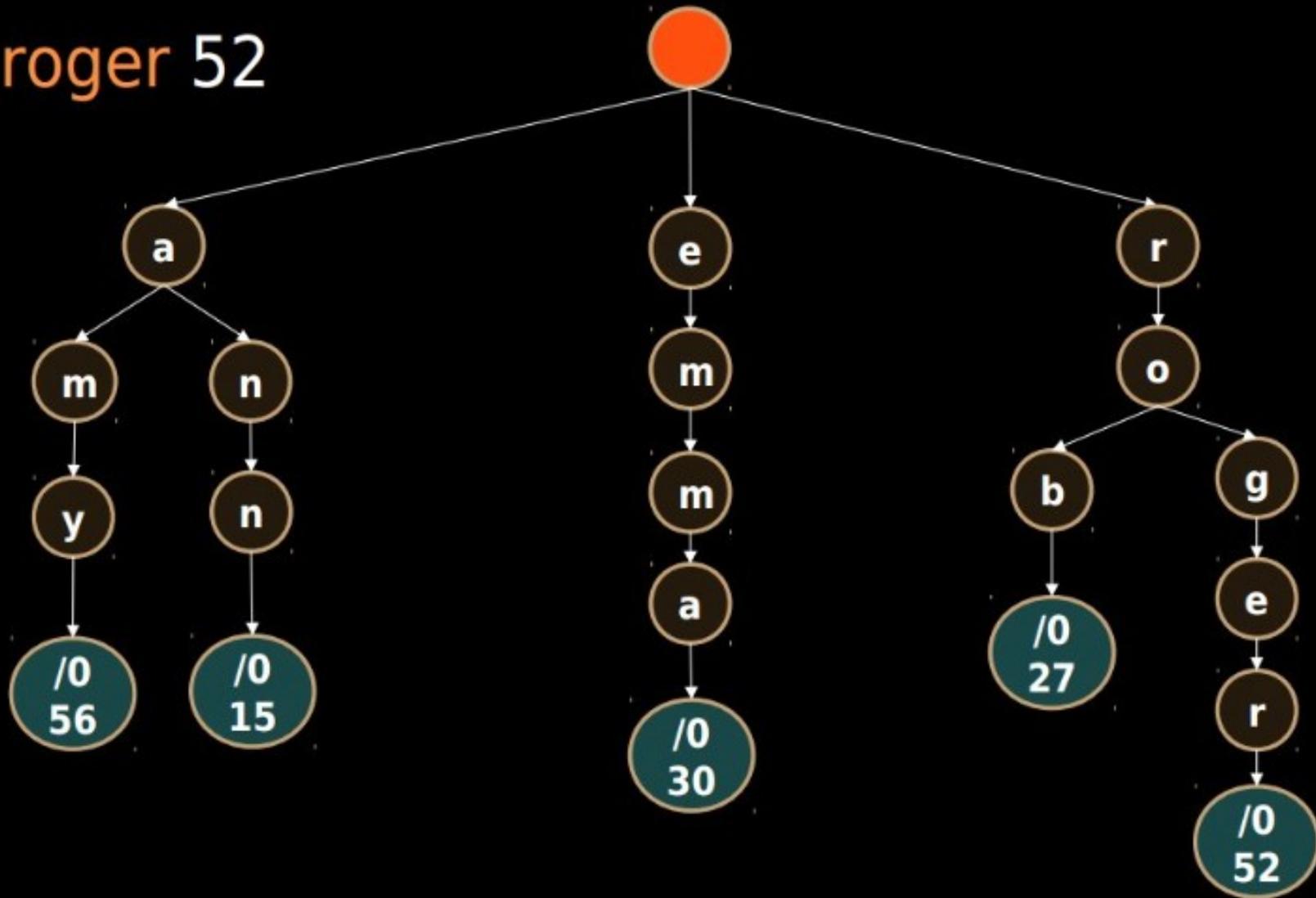
Montando uma árvore Trie

- rob 27



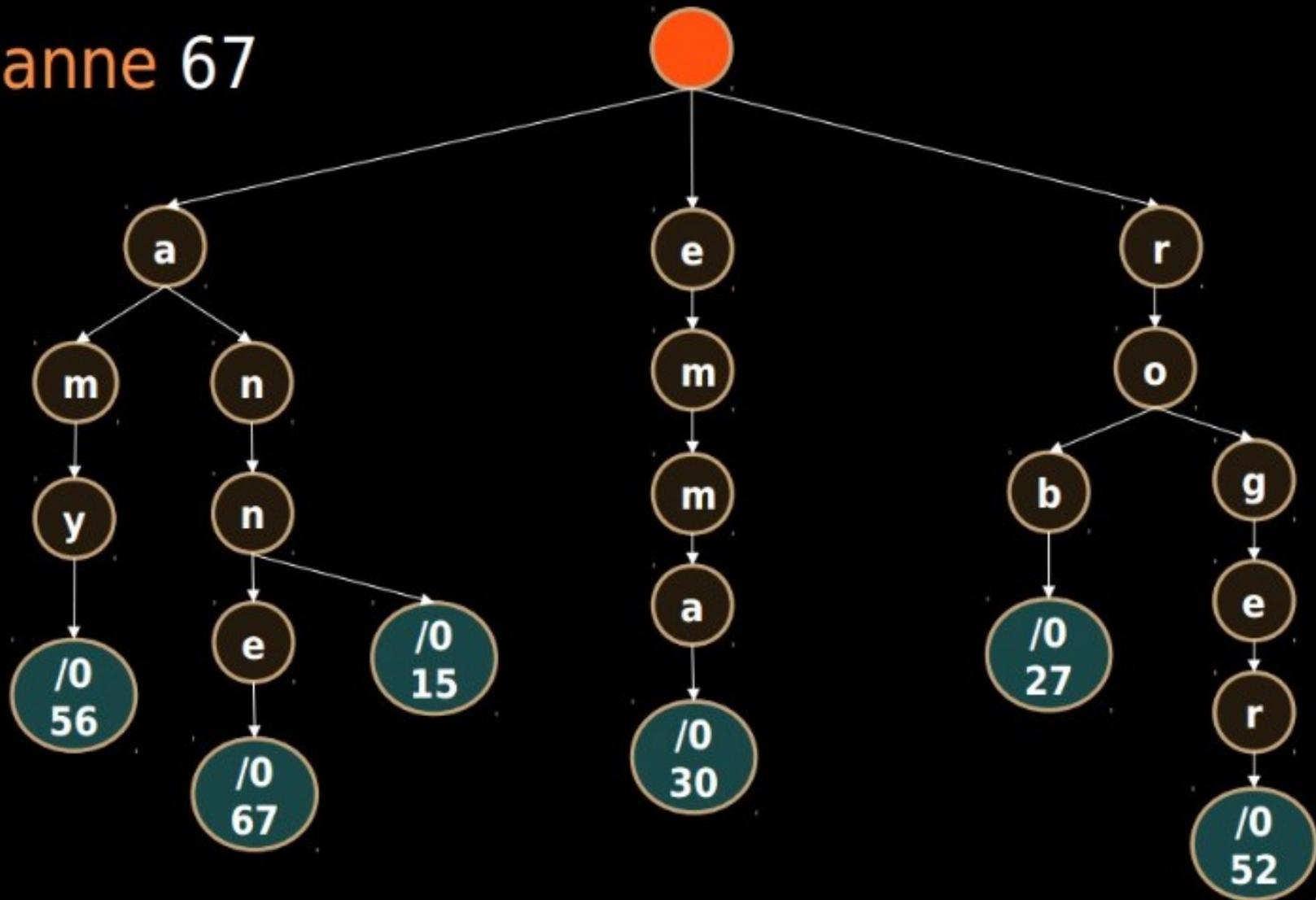
Montando uma árvore Trie

- roger 52



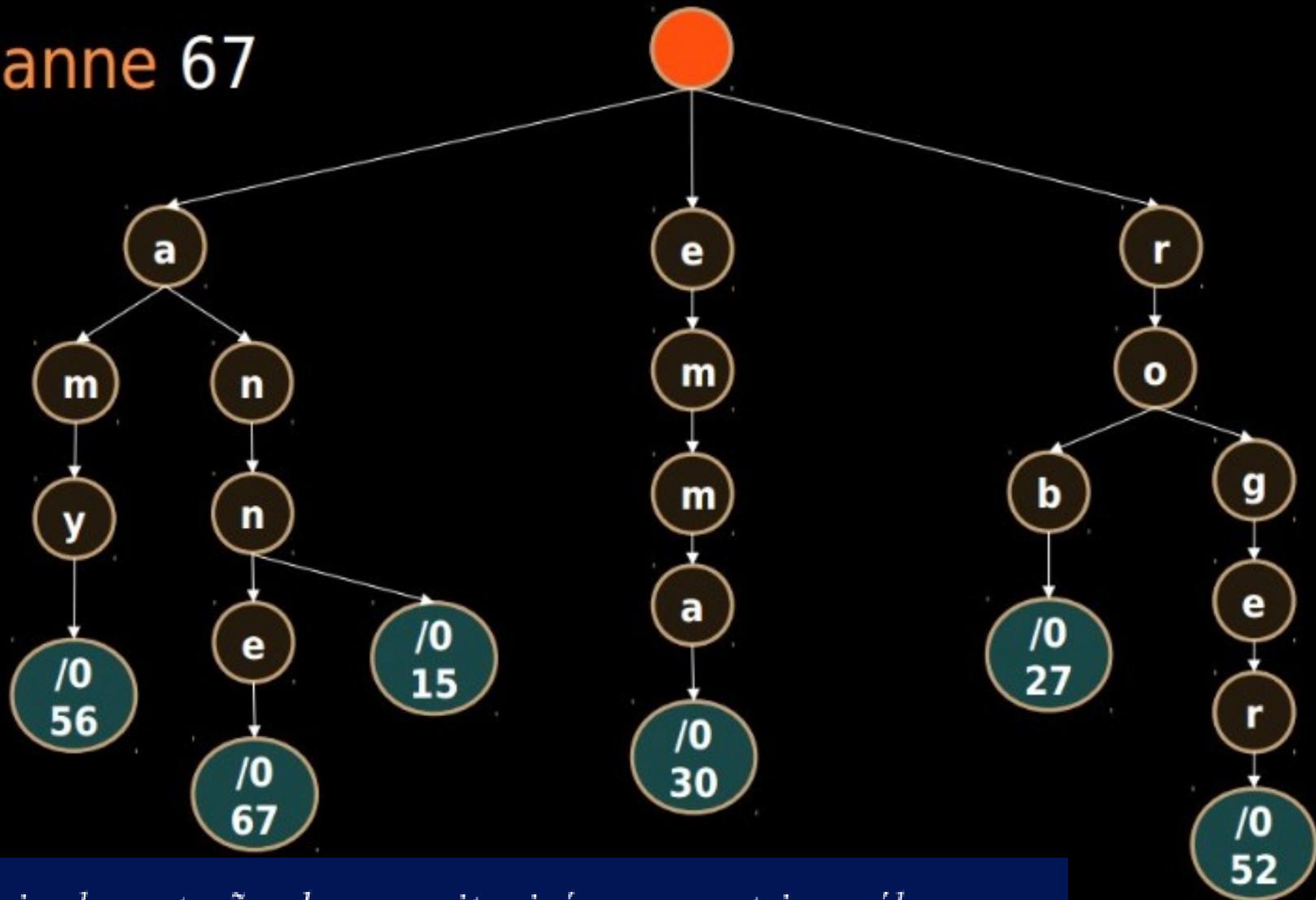
Montando uma árvore Trie

- ane 67



Montando uma árvore Trie

- anne 67



Na implementação: deve se evitar inúmeros ponteiros nulos.

Implementando uma ATRIE

Implementação mais simples: **R-way**

A árvore contém dois tipos de nós:

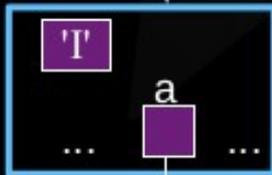
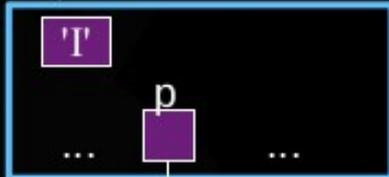
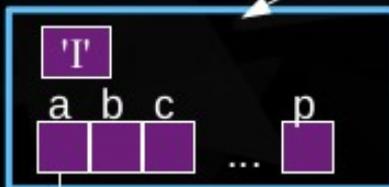
Nó interno: 'I'

Nó de informação / palavra: 'P'

Cada nó contém todos os valores do alfabeto. (há desperdício de espaço)

Considere uma ATRIE para armazenar chaves do alfabeto
{a,b,c,d,...,y,z} (27 letras)

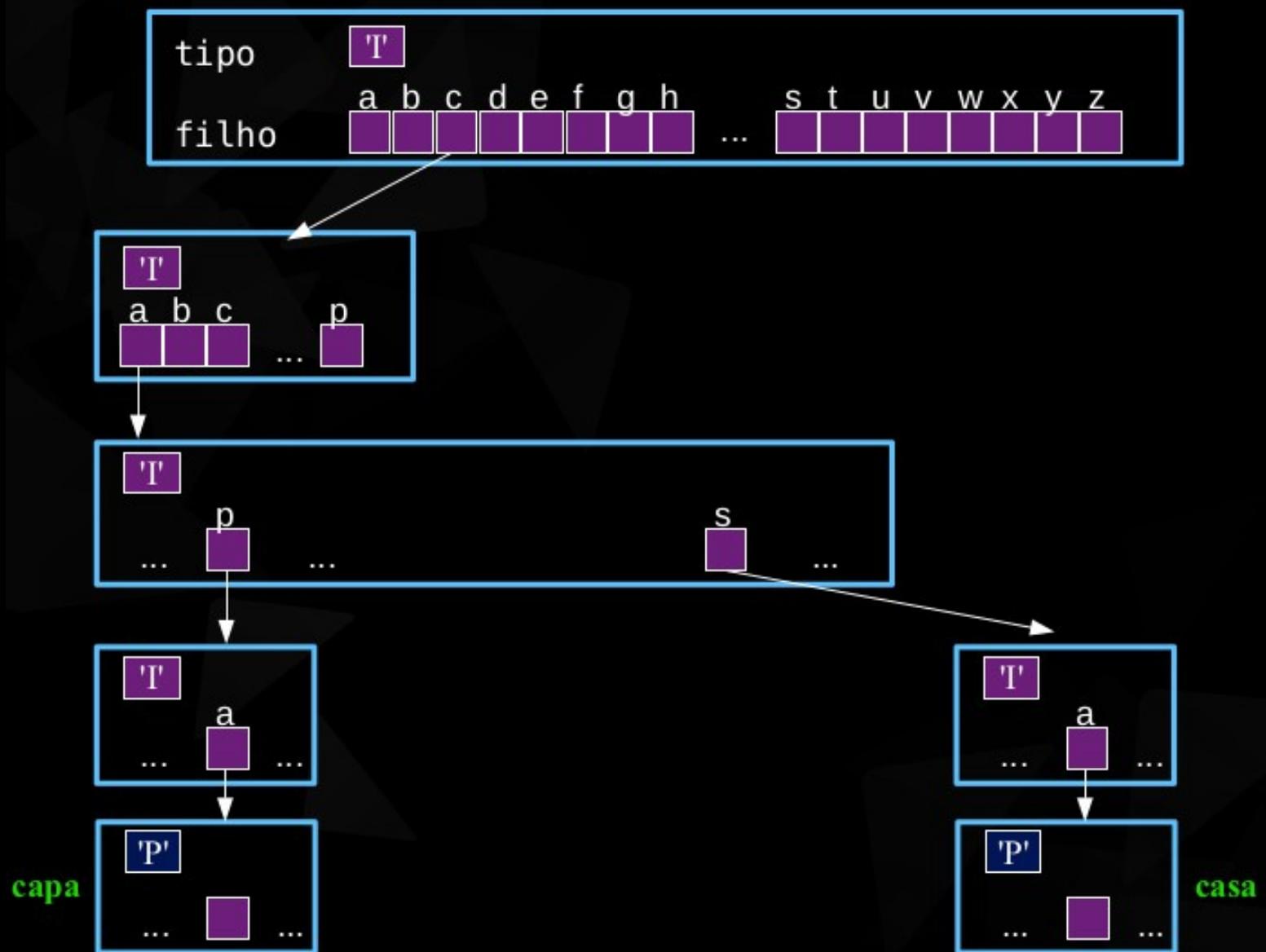
Implementando uma ATRIE



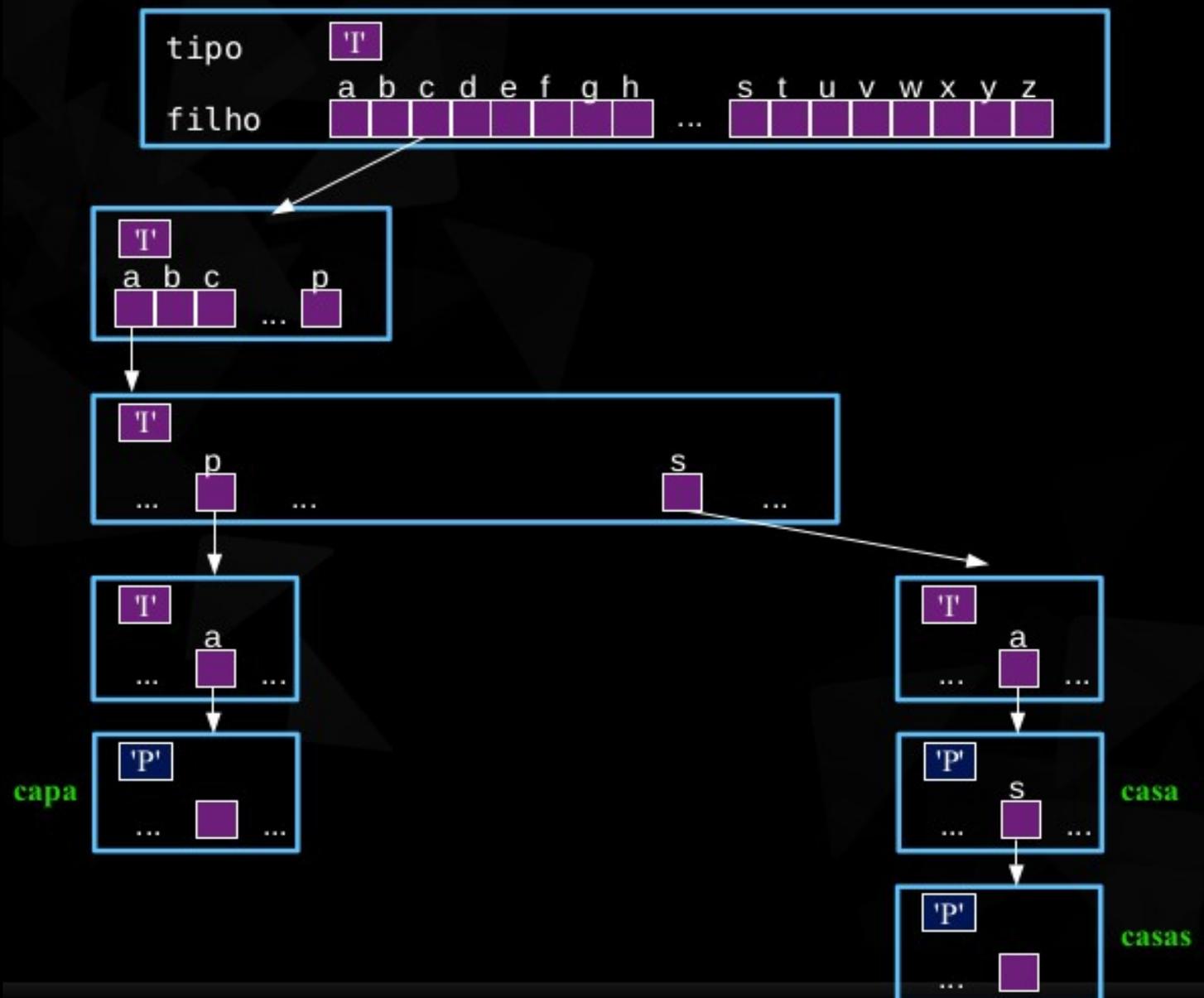
capa



Implementando uma ATRIE

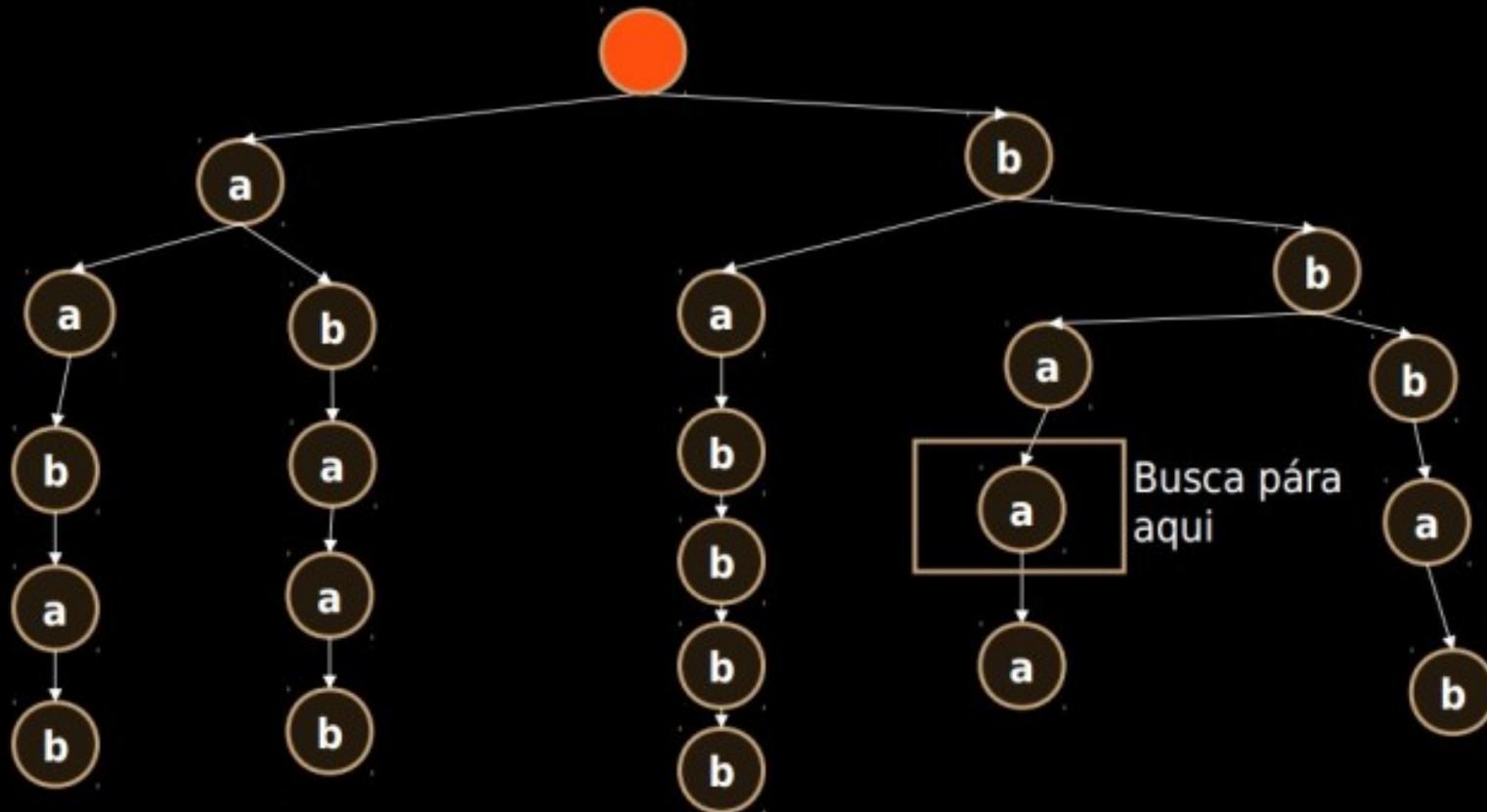


Implementando uma ATRIE



Inserção em uma ATRIE

Inserção : bbaabb



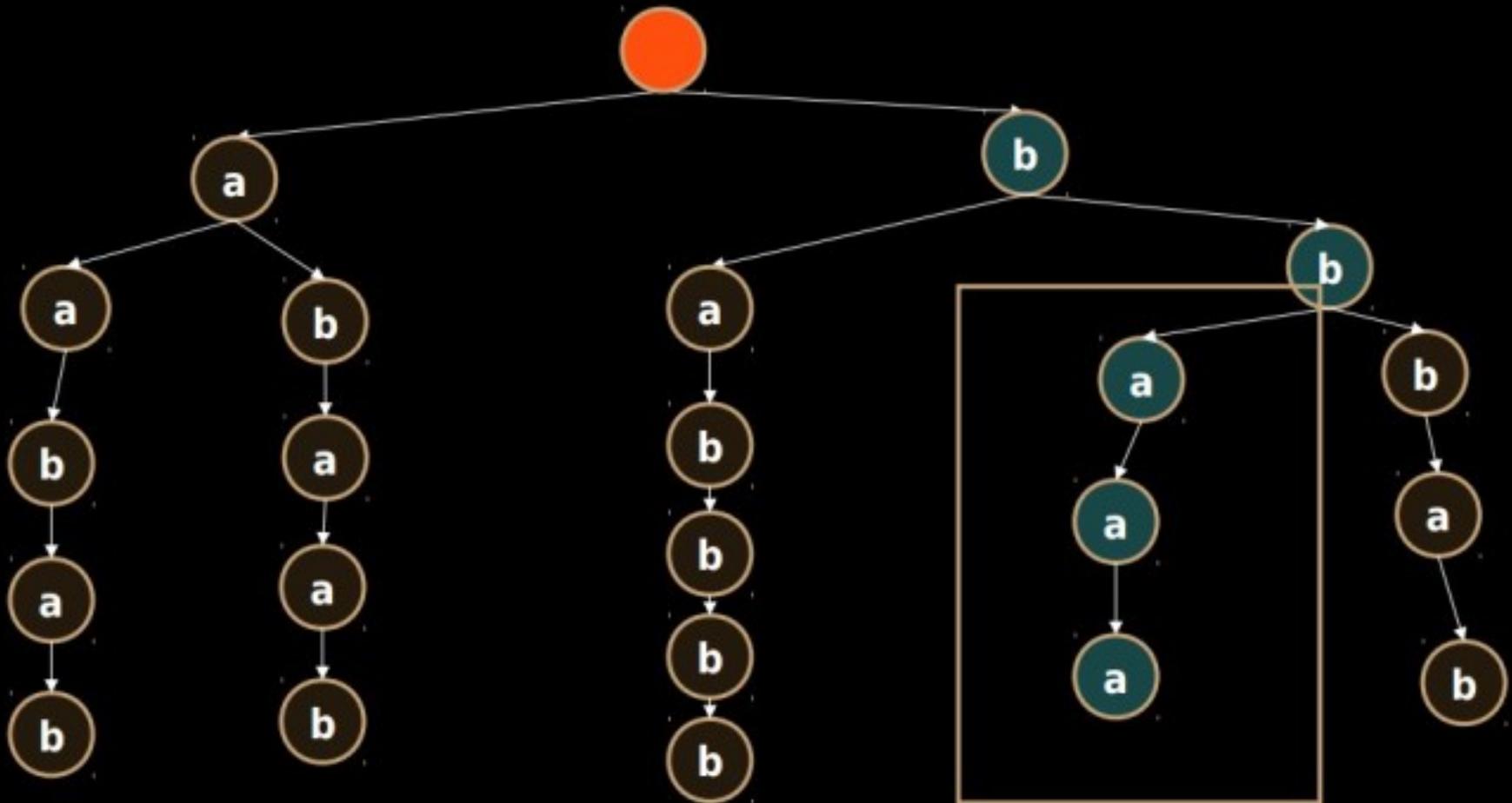
Remoção em uma ATRIE

- 1) Busca-se o nó que representa o final da palavra a ser Removida.
- 2) São removidos os nós que possuem apenas um filho pelo caminho ascendente.
- 3) A remoção é concluída quando se encontra um nó com mais de um filho.



Remoção em uma ATRIE

Remoção : **bb**aaa



Complexidade Computacional

Busca, Inserção e Remoção:

Para uma palavra p de tamanho $|p|$: $O(|p|)$
Independente do número total de palavras.

Uso de memória:

Ineficiente para armazenar todas as palavras.



Aplicações

Busca por aproximação de correspondência:
Onde podem ser localizadas dados que são semelhantes a
uma chave informada.

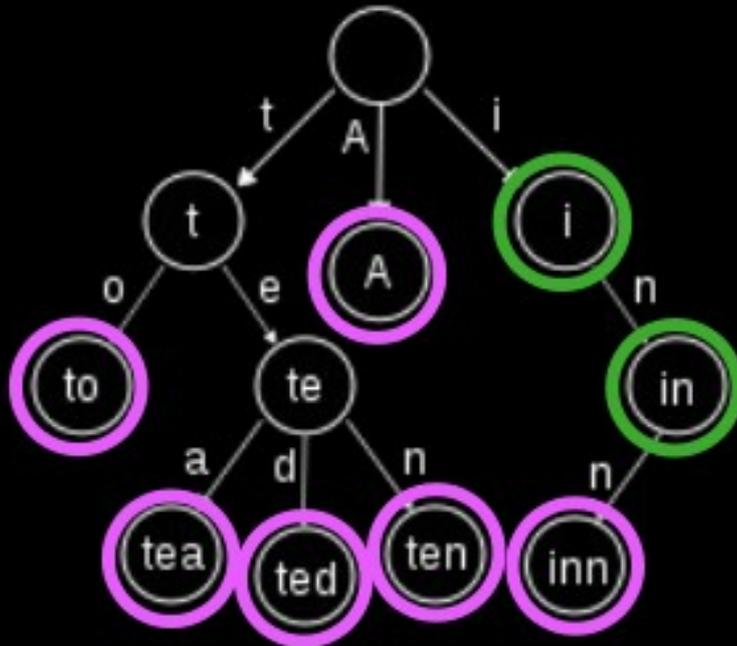
- 1) Corretor Ortográfico
- 2) Auto-preenchimento



Aplicações

Corretor ortográfico:

Nesse tipo de programas, as palavras são comparadas com um dicionário armazenado em arquivo, e se não são encontradas, indica-se as opções de correção.

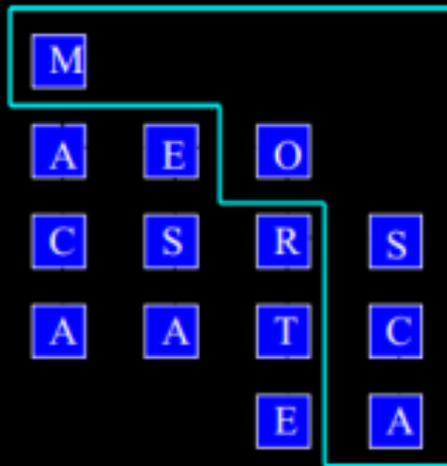


Palavra a ser comparada/procurada: **tex**

Aplicações

Auto-preenchimento:

Nesse tipo de programas, a medida que vai digitando são exibidas as opções possíveis de palavras já utilizadas.



M O S C A

O algoritmo compara a existência de correspondência na estrutura.

A cada caractere digitado, são apresentadas as opções de preenchimento, e no momento em que só existir um caminho possível a ser seguido na TRIE ocorre o preenchimento automático

AULA 18

ESTRUTURA DE DADOS

Árvores Trie
