

Algoritmos e Estruturas de Dados II

Árvores Trie

Prof. Tiago Eugenio de Melo

tmelo@uea.edu.br

www.tiagodemelo.info

Observações

- As palavras com a fonte Courier indicam as palavras-reservadas da linguagem de programação.

Referências

- **Projetos de Algoritmos – com implementações em Pascal e C.** Nivio Ziviani. 2ª edição. Thomson, 2005.
- A visualização da construção da Árvore Trie pode ser realizada no endereço abaixo:
 - <https://www.cs.usfca.edu/~galles/visualization/Trie.html>

Árvores TRIE

Introdução

Introdução

- Definida em 1960 por Edward Fredkin.

Introdução

- Definida em 1960 por Edward Fredkin.
- O termo surgiu da palavra *retrieval*.

Introdução

- Definida em 1960 por Edward Fredkin.
- O termo surgiu da palavra *retrieval*.
- Para distinção com *tree*, pronuncia-se *try*.

Introdução

- Definida em 1960 por Edward Fredkin.
- O termo surgiu da palavra *retrieval*.
- Para distinção com *tree*, pronuncia-se *try*.
- Cada nó contém informações sobre um ou mais símbolos do alfabeto utilizado.

Árvores Trie

Árvores Trie

- As *tries* são estruturas indicadas para suportar tarefas de tratamento léxico, tais como:

Árvores Trie

- As *tries* são estruturas indicadas para suportar tarefas de tratamento léxico, tais como:
 - Manuseamento de dicionários.

Árvores Trie

- As *tries* são estruturas indicadas para suportar tarefas de tratamento léxico, tais como:
 - Manuseamento de dicionários.
 - Pesquisas em texto de grande dimensão.

Árvores Trie

- As *tries* são estruturas indicadas para suportar tarefas de tratamento léxico, tais como:
 - Manuseamento de dicionários.
 - Pesquisas em texto de grande dimensão.
 - Construção de índices de documentos.

Árvores Trie

- As *tries* são estruturas indicadas para suportar tarefas de tratamento léxico, tais como:
 - Manuseamento de dicionários.
 - Pesquisas em texto de grande dimensão.
 - Construção de índices de documentos.
 - Expressões regulares (padrões de pesquisa).

Árvores Trie

Árvores Trie

- A ideia geral é usar parte das chaves como caminho da busca.

Árvores Trie

- A ideia geral é usar parte das chaves como caminho da busca.
- O caminho da raiz (*root*) da Trie para qualquer outro nó representa um prefixo de uma string.

Árvores Trie

- A ideia geral é usar parte das chaves como caminho da busca.
- O caminho da raiz (*root*) da Trie para qualquer outro nó representa um prefixo de uma string.
- Em Tries Compactas todos os descendentes diretos do mesmo pai são agrupados.

Árvores Trie

- A ideia geral é usar parte das chaves como caminho da busca.
- O caminho da raiz (*root*) da Trie para qualquer outro nó representa um prefixo de uma string.
- Em Tries Compactas todos os descendentes diretos do mesmo pai são agrupados.
- No último nó, o último caractere da palavra sendo procurada deverá ter associado a posição da palavra no texto.

Características

Características

- Chaves

Características

- Chaves
 - Cada chave é formada por palavras sobre um alfabeto.

Características

- Chaves
 - Cada chave é formada por palavras sobre um alfabeto.
 - Palavras possuem tamanho **variável** e **ilimitado**.

Características

- Chaves
 - Cada chave é formada por palavras sobre um alfabeto.
 - Palavras possuem tamanho **variável** e **ilimitado**.
 - Em geral, associam-se as chaves a elementos ou registros.

Características

Características

- Chaves

Características

- Chaves
 - Cada chave é formada a partir de alfabeto de símbolos.

Características

- Chaves
 - Cada chave é formada a partir de alfabeto de símbolos.
 - O alfabeto pode abranger $\{0,1\}$, $\{a, b, c, d, \dots\}$ ou $\{0, 1, 2, \dots, 9\}$ e mais o caractere nulo (branco).

Características

- Chaves
 - Cada chave é formada a partir de alfabeto de símbolos.
 - O alfabeto pode abranger $\{0,1\}$, $\{a, b, c, d, \dots\}$ ou $\{0, 1, 2, \dots, 9\}$ e mais o caractere nulo (branco).
 - Exemplos de chaves:

Características

- Chaves
 - Cada chave é formada a partir de alfabeto de símbolos.
 - O alfabeto pode abranger $\{0,1\}$, $\{a, b, c, d, \dots\}$ ou $\{0, 1, 2, \dots, 9\}$ e mais o caractere nulo (branco).
 - Exemplos de chaves:
 - AABBADA

Características

- Chaves
 - Cada chave é formada a partir de alfabeto de símbolos.
 - O alfabeto pode abranger $\{0,1\}$, $\{a, b, c, d, \dots\}$ ou $\{0, 1, 2, \dots, 9\}$ e mais o caractere nulo (branco).
 - Exemplos de chaves:
 - AABBADA
 - 1239123

Características

- Chaves
 - Cada chave é formada a partir de alfabeto de símbolos.
 - O alfabeto pode abranger $\{0,1\}$, $\{a, b, c, d, \dots\}$ ou $\{0, 1, 2, \dots, 9\}$ e mais o caractere nulo (branco).
 - Exemplos de chaves:
 - AABBADA
 - 1239123
 - Maria

Características

- Chaves

- Cada chave é formada a partir de alfabeto de símbolos.
- O alfabeto pode abranger $\{0,1\}$, $\{a, b, c, d, \dots\}$ ou $\{0, 1, 2, \dots, 9\}$ e mais o caractere nulo (branco).
- Exemplos de chaves:
 - AABBADA
 - 1239123
 - Maria
- As chaves são parcialmente partilhadas entre os elementos.

Estrutura

Estrutura

- Árvore ordenada e n-ária.

Estrutura

- Árvore ordenada e n-ária.
- As chaves são geralmente caracteres.

Estrutura

- Árvore ordenada e n-ária.
- As chaves são geralmente caracteres.
- Ao contrário das árvores de busca binária, nenhum nó armazena a chave.

Estrutura

- Árvore ordenada e n-ária.
- As chaves são geralmente caracteres.
- Ao contrário das árvores de busca binária, nenhum nó armazena a chave.
- As chaves são determinadas pela posição na árvore.

Estrutura (cont.)

Estrutura (cont.)

- Descendentes de mesmo nó possuem o mesmo prefixo.

Estrutura (cont.)

- Descendentes de mesmo nó possuem o mesmo prefixo.
- A raiz é nó (cadeia) vazio(a).

Estrutura (cont.)

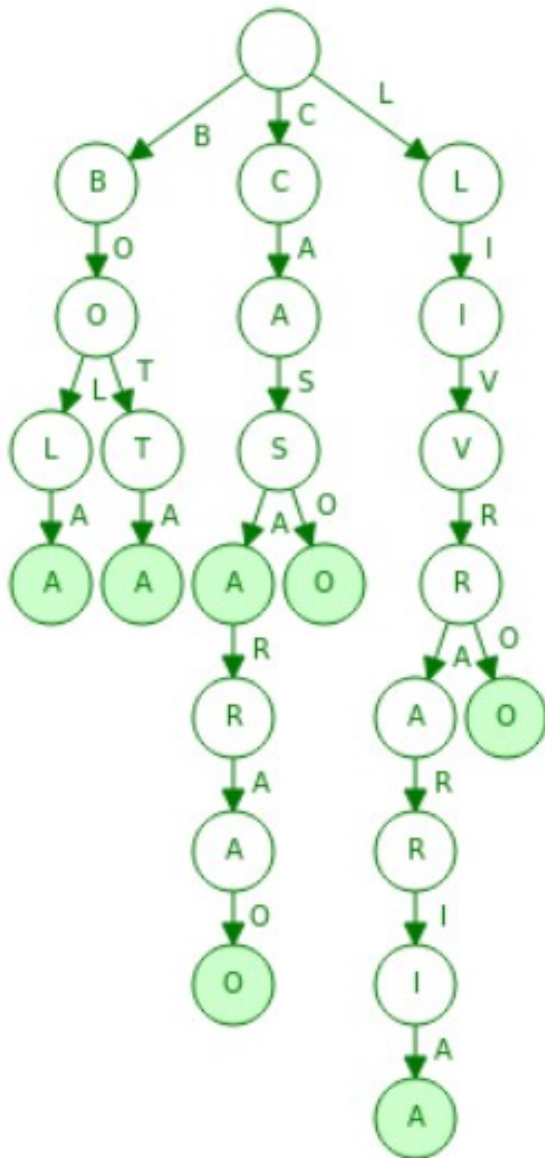
- Descendentes de mesmo nó possuem o mesmo prefixo.
- A raiz é nó (cadeia) vazio(a).
- Elementos associados a folhas ou a alguns valores de nós internos de interesse.

Estrutura (cont.)

- Descendentes de mesmo nó possuem o mesmo prefixo.
- A raiz é nó (cadeia) vazio(a).
- Elementos associados a folhas ou a alguns valores de nós internos de interesse.
- O caminho da raiz para qualquer outro nó é um prefixo de uma string.

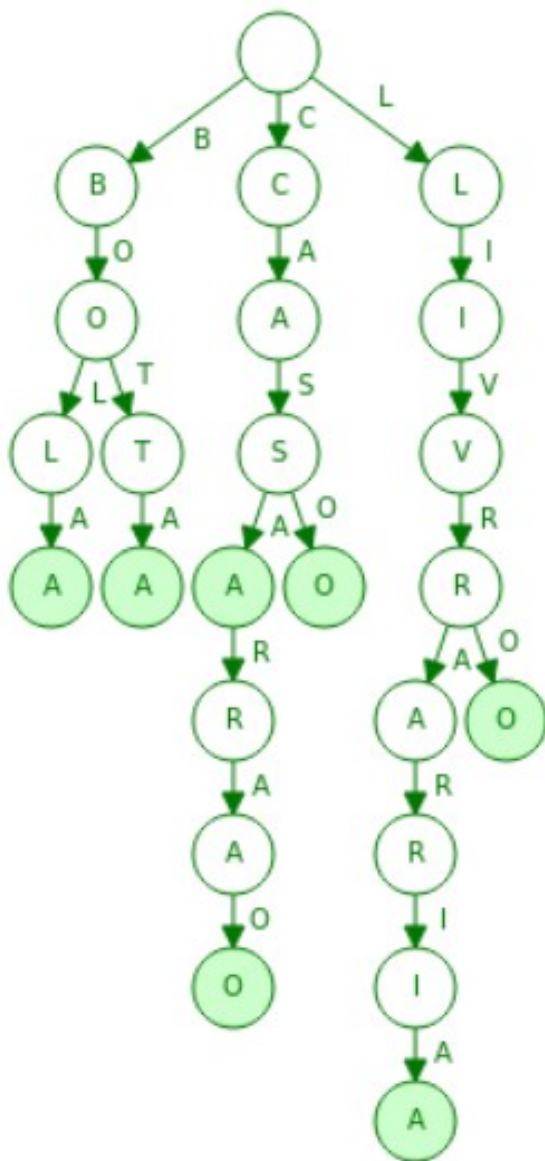
Exemplo

Exemplo

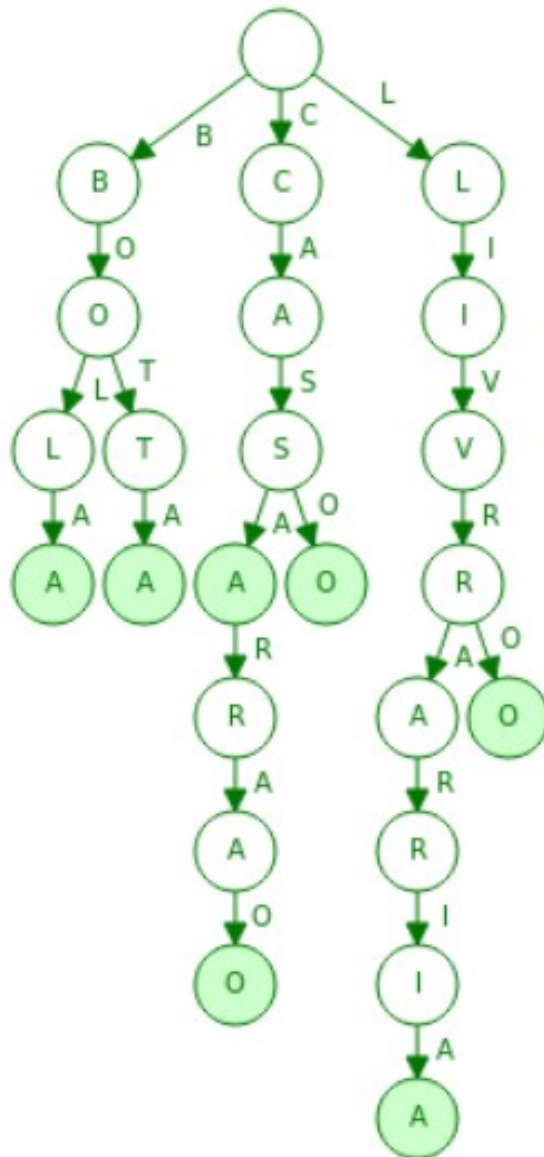


Exemplo

Chaves

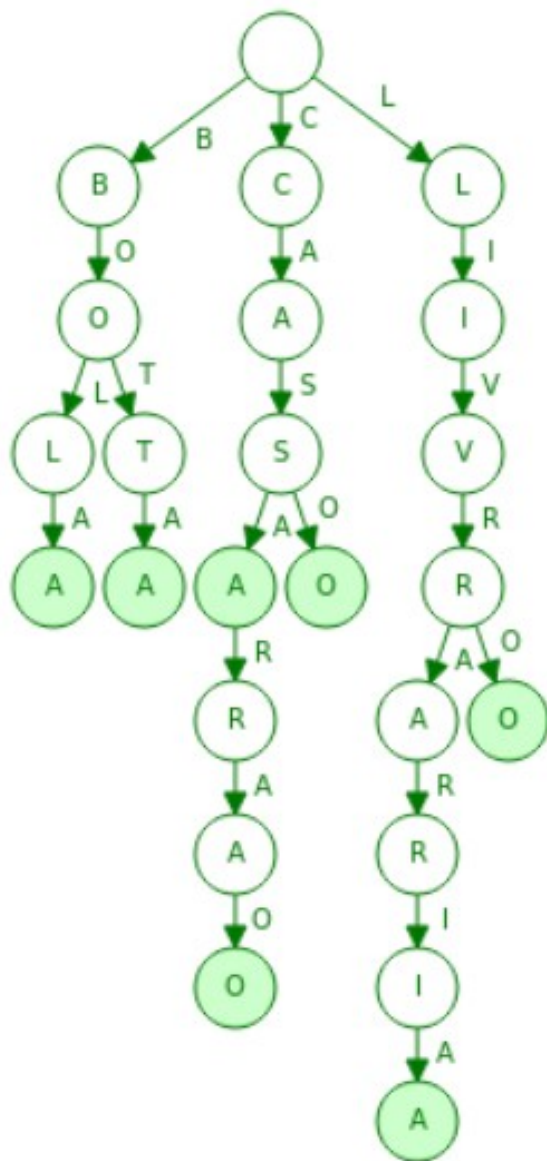


Exemplo



Chaves
- BOLA

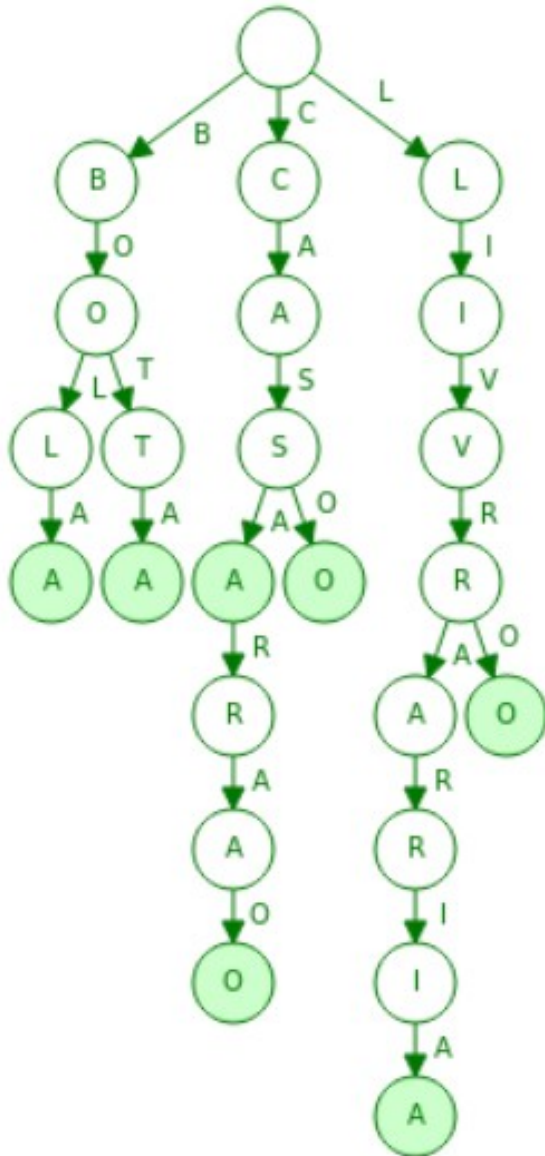
Exemplo



Chaves

- BOLA
- BOTA

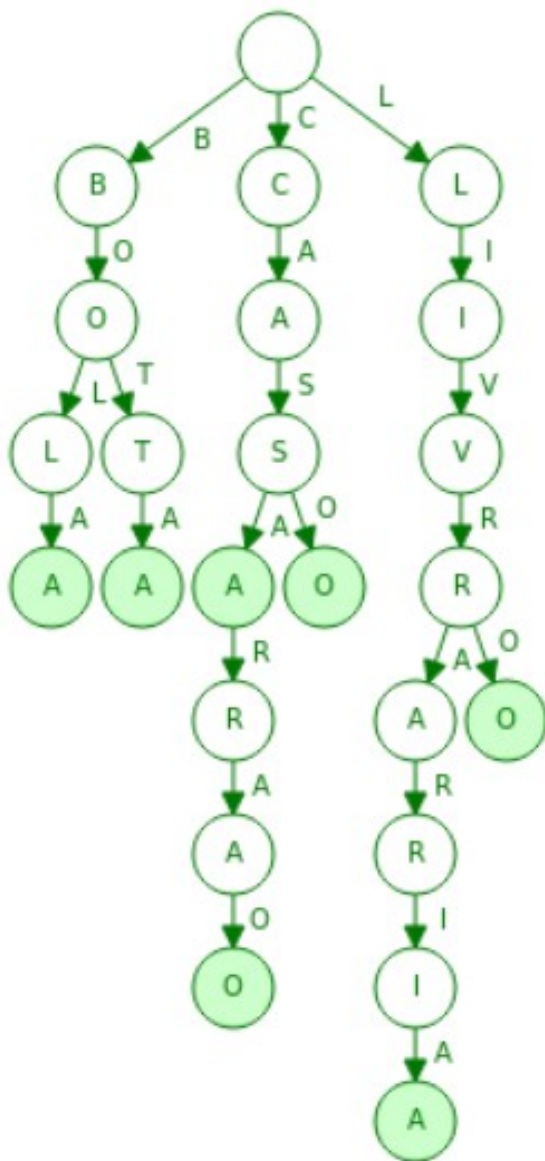
Exemplo



Chaves

- BOLA
- BOTA
- CASA

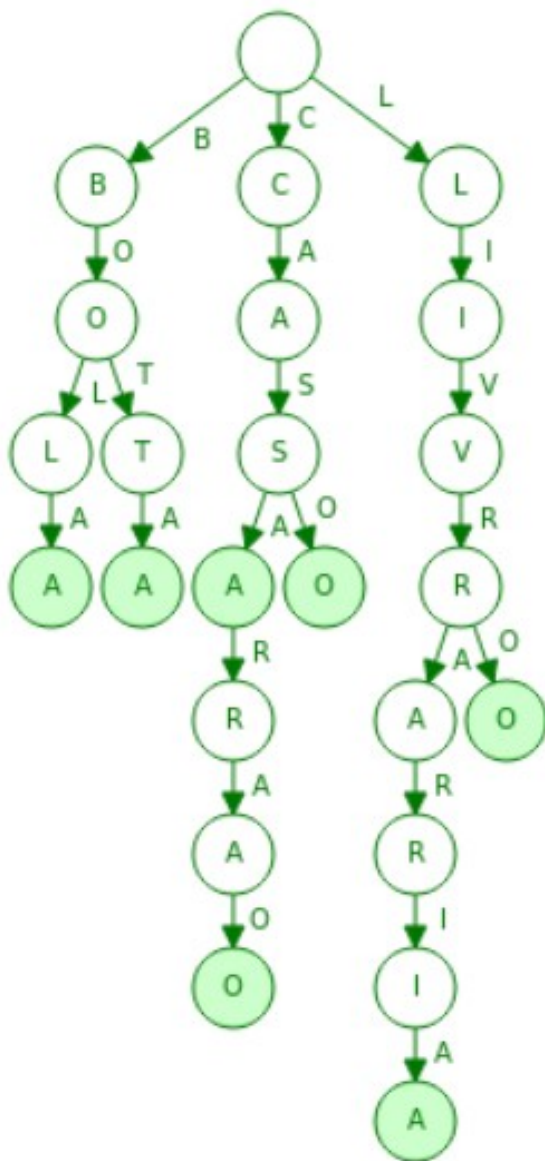
Exemplo



Chaves

- BOLA
- BOTA
- CASA
- CASARAO

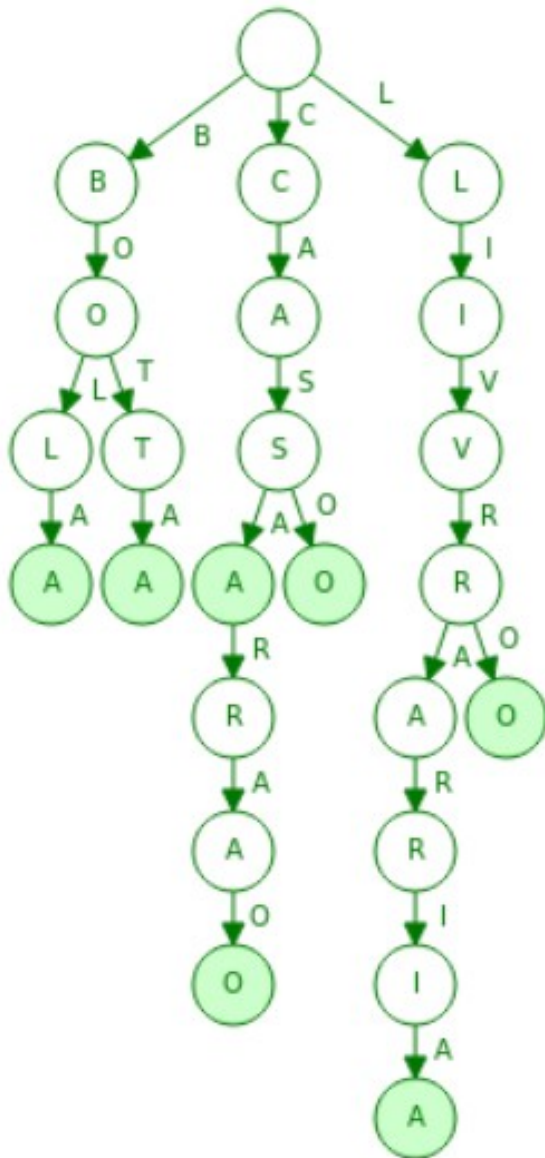
Exemplo



Chaves

- BOLA
- BOTA
- CASA
- CASARAO
- CASO

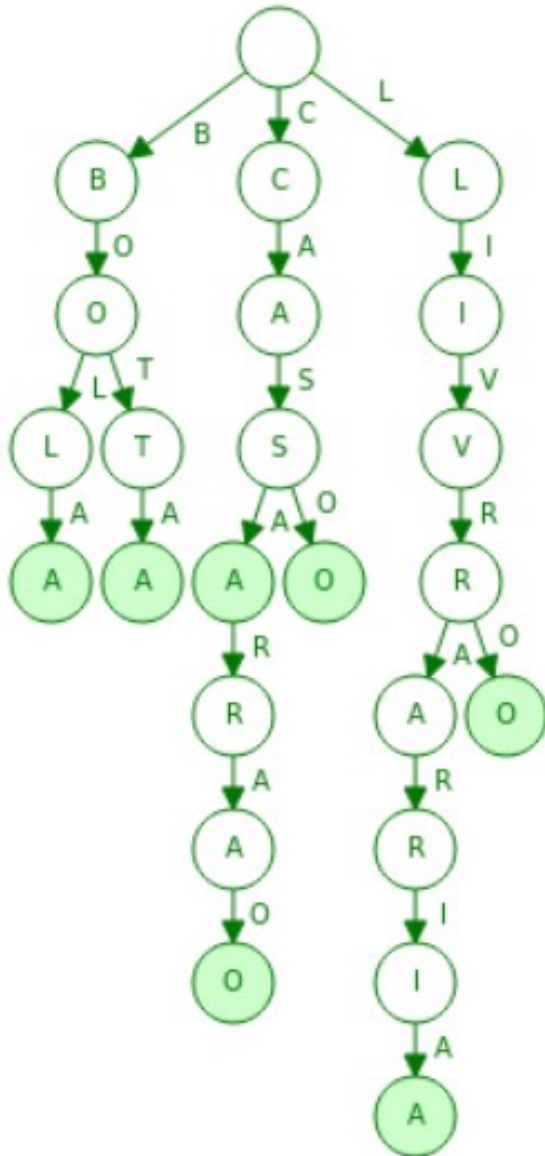
Exemplo



Chaves

- BOLA
- BOTA
- CASA
- CASARAO
- CASO
- LIVRARIA

Exemplo

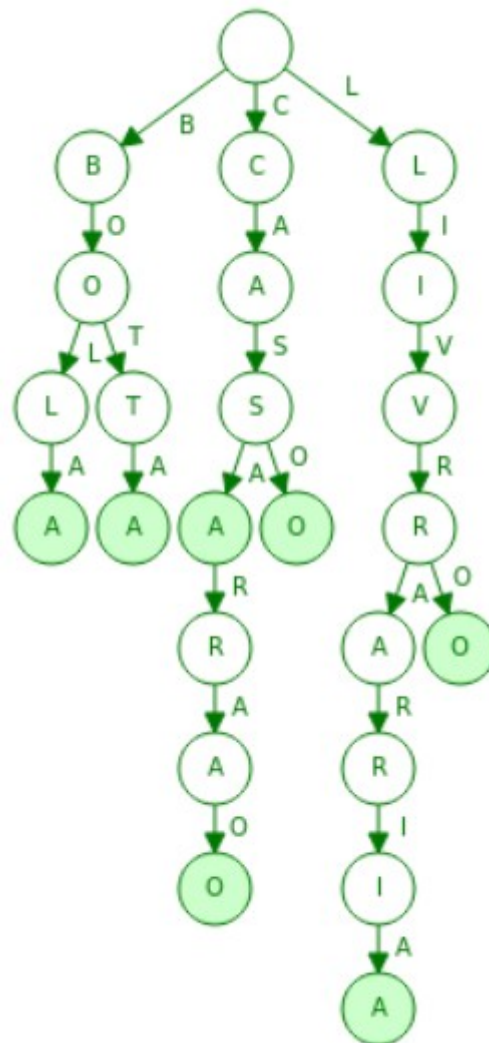


Chaves

- BOLA
- BOTA
- CASA
- CASARAO
- CASO
- LIVRARIA
- LIVRO

Exemplo

- Árvore ordenada e n-ária.

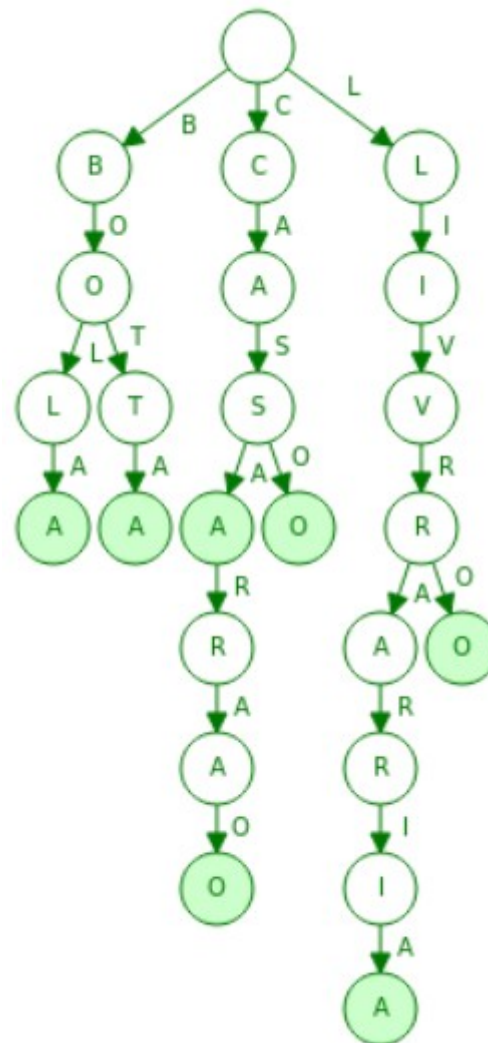


Chaves

- BOLA
- BOTA
- CASA
- CASARAO
- CASO
- LIVRARIA
- LIVRO

Exemplo

- As chaves são geralmente caracteres.

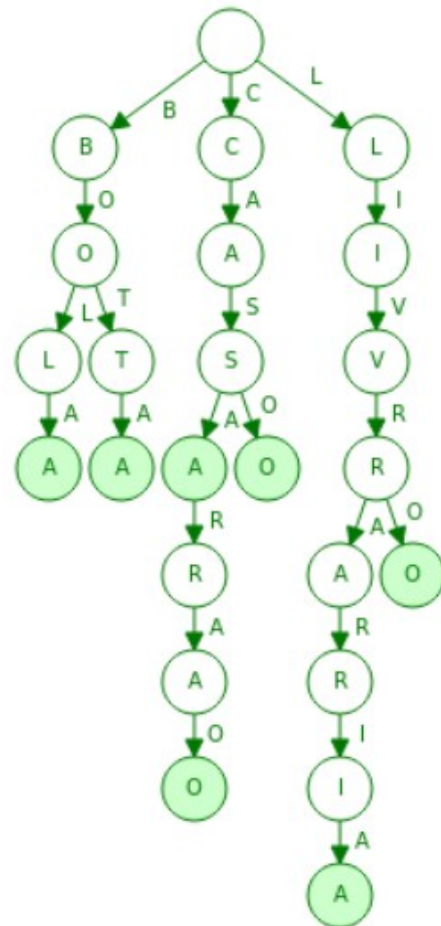


Chaves

- BOLA
- BOTA
- CASA
- CASARAO
- CASO
- LIVRARIA
- LIVRO

Exemplo

- Ao contrário das árvores de busca binária, nenhum nó armazena a chave.

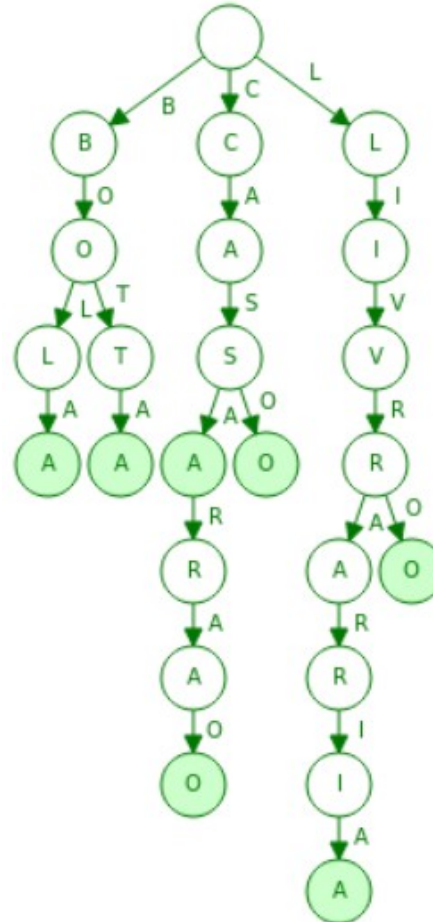


Chaves

- BOLA
- BOTAS
- CASAS
- CASARAO
- CASO
- LIVRARIA
- LIVRO

Exemplo

- As chaves são determinadas pela posição na árvore.

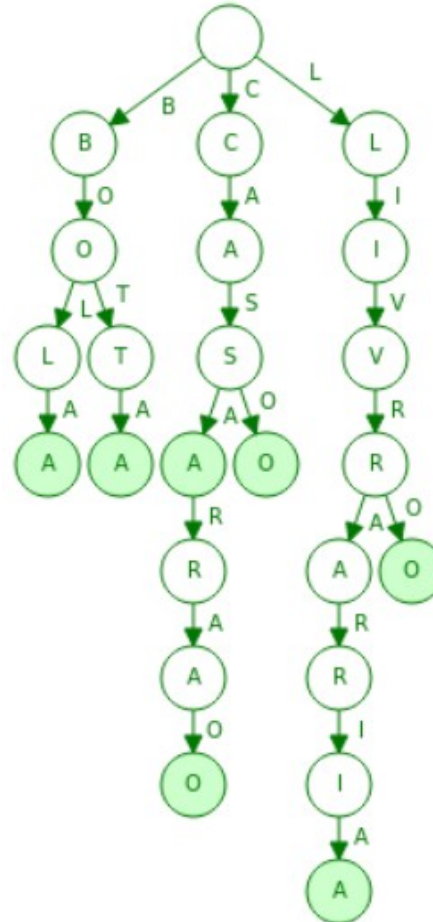


Chaves

- BOLA
- BOTA
- CASA
- CASARAO
- CASO
- LIVRARIA
- LIVRO

Exemplo

- Descendentes de mesmo nó possuem o mesmo prefixo.

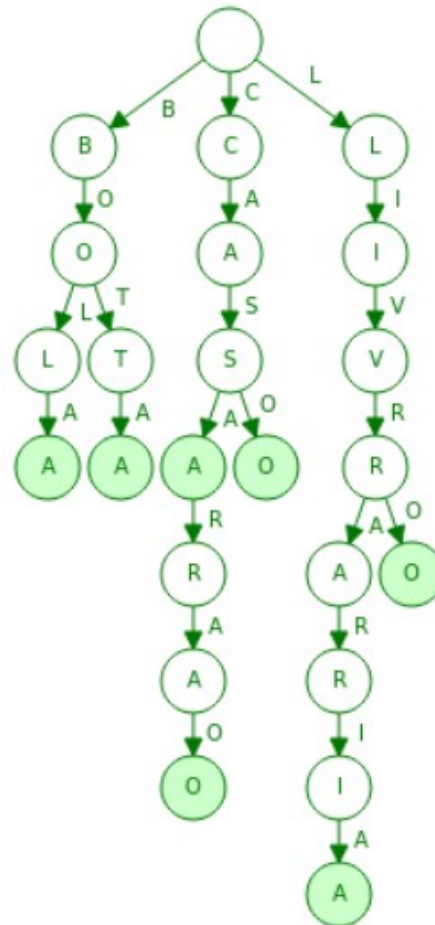


Chaves

- BOLA
- BOTA
- CASA
- CASARAO
- CASO
- LIVRARIA
- LIVRO

Exemplo

- A raiz é nó (cadeia) vazio(a).

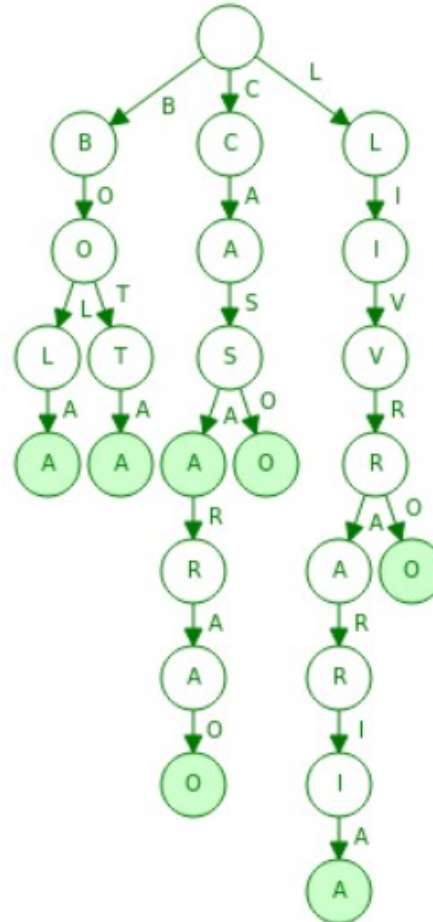


Chaves

- BOLA
- BOTA
- CASA
- CASARAO
- CASO
- LIVRARIA
- LIVRO

Exemplo

- Elementos são associados a folhas ou a alguns valores de nós internos de interesse.

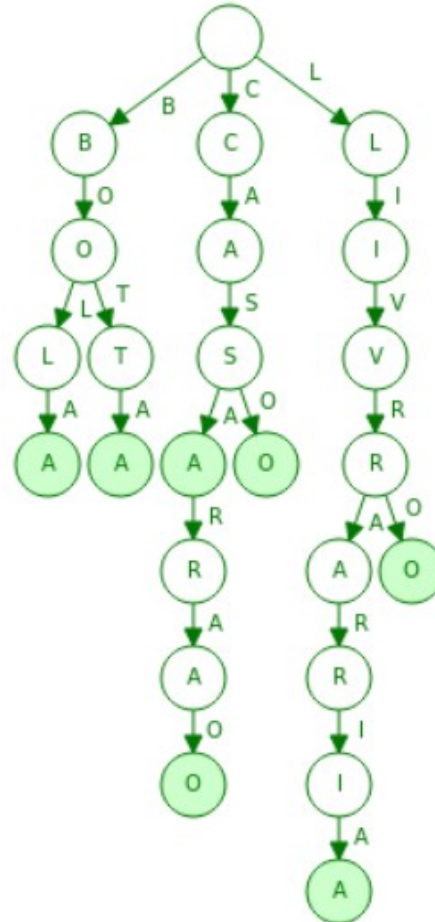


Chaves

- BOLA
- BOTA
- CASA
- CASARAO
- CASO
- LIVRARIA
- LIVRO

Exemplo

- O caminho da raiz para qualquer outro nó é um prefixo de uma string.



Chaves

- BOLA
- BOTA
- CASA
- CASARAO
- CASO
- LIVRARIA
- LIVRO

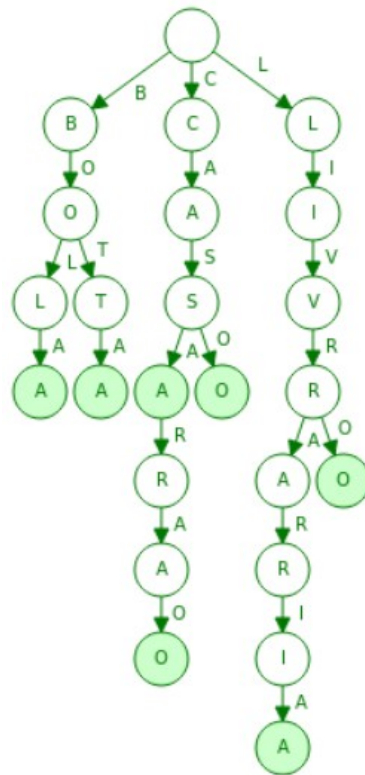
Exemplo

Exemplo

- O grau corresponde ao tamanho do alfabeto.

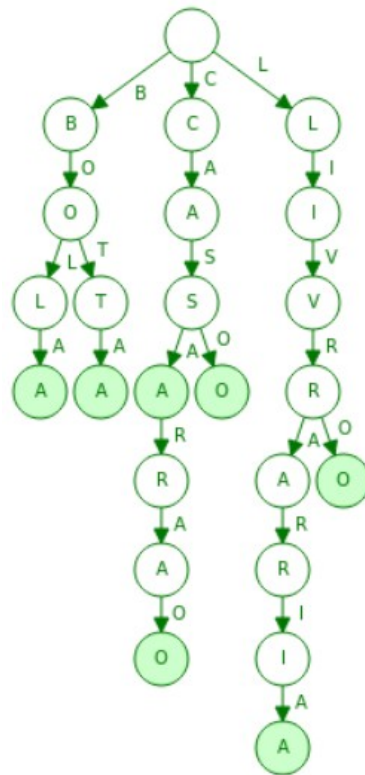
Exemplo

- O grau corresponde ao tamanho do alfabeto.



Exemplo

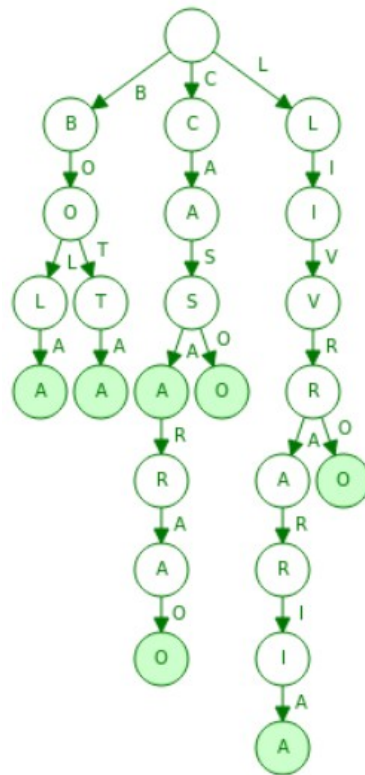
- O grau corresponde ao tamanho do alfabeto.



Chaves

Exemplo

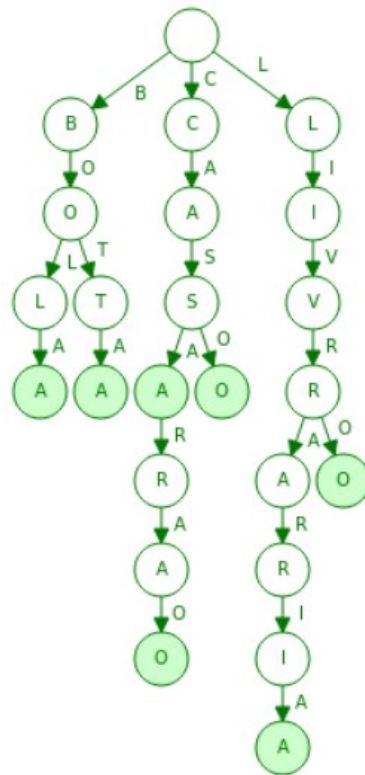
- O grau corresponde ao tamanho do alfabeto.



Chaves
- BOLA

Exemplo

- O grau corresponde ao tamanho do alfabeto.

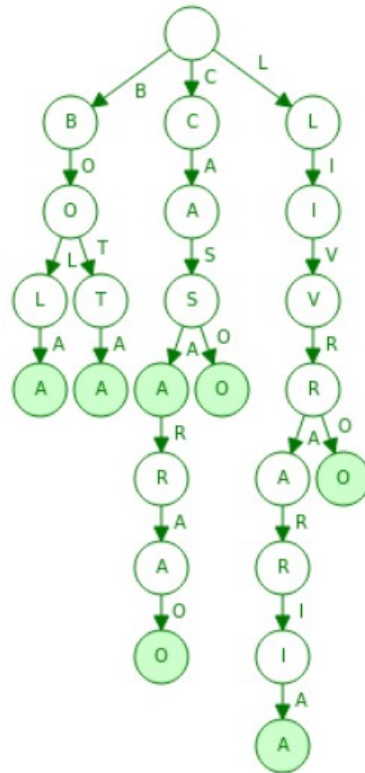


Chaves

- BOLA
- BOTA

Exemplo

- O grau corresponde ao tamanho do alfabeto.

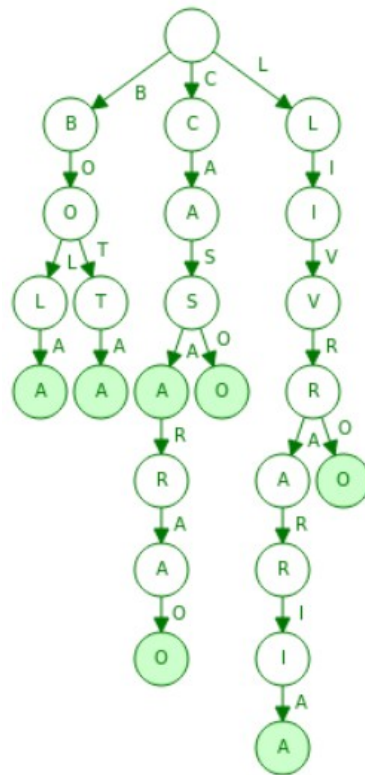


Chaves

- BOLA
- BOTA
- CASA

Exemplo

- O grau corresponde ao tamanho do alfabeto.

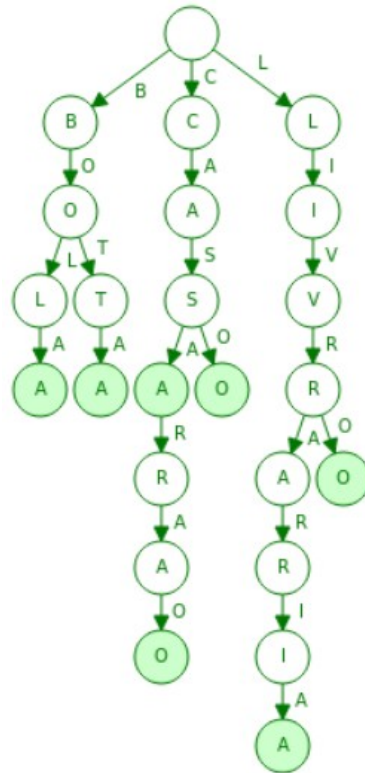


Chaves

- BOLA
- BOTA
- CASA
- CASARAO

Exemplo

- O grau corresponde ao tamanho do alfabeto.

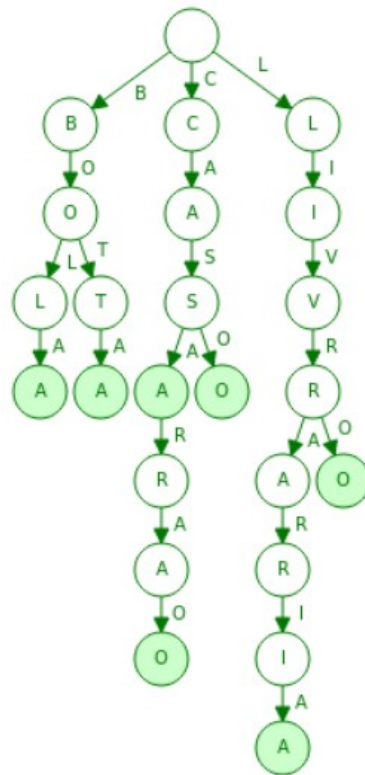


Chaves

- BOLA
- BOTÁ
- CASA
- CASARAO
- CASO

Exemplo

- O grau corresponde ao tamanho do alfabeto.

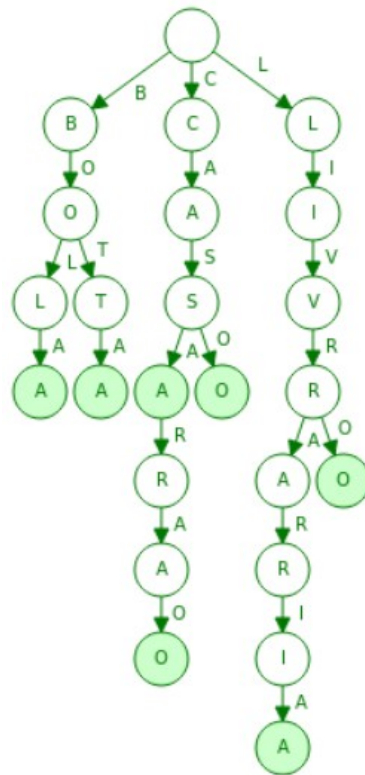


Chaves

- BOLA
- BOTA
- CASA
- CASARAO
- CASO
- LIVRARIA

Exemplo

- O grau corresponde ao tamanho do alfabeto.

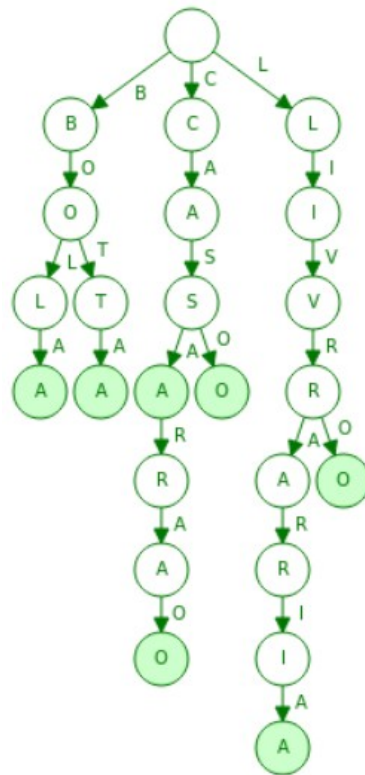


Chaves

- BOLA
- BOTA
- CASA
- CASARAO
- CASO
- LIVRARIA
- LIVRO

Exemplo

- O grau corresponde ao tamanho do alfabeto.
- Cada nível que se desce corresponde a avançar um elemento na chave.



Chaves

- BOLA
- BOTA
- CASA
- CASARAO
- CASO
- LIVRARIA
- LIVRO

Construção de uma árvore Trie

Construção de uma árvore Trie

- Considere que queremos construir uma árvore Trie com os seguintes elementos:

Construção de uma árvore Trie

- Considere que queremos construir uma árvore Trie com os seguintes elementos:
 - BOLA

Construção de uma árvore Trie

- Considere que queremos construir uma árvore Trie com os seguintes elementos:
 - BOLA
 - BOTA

Construção de uma árvore Trie

- Considere que queremos construir uma árvore Trie com os seguintes elementos:
 - BOLA
 - BOTA
 - CASA

Construção de uma árvore Trie

- Considere que queremos construir uma árvore Trie com os seguintes elementos:
 - BOLA
 - BOTA
 - CASA
 - CASARAO

Construção de uma árvore Trie

- Considere que queremos construir uma árvore Trie com os seguintes elementos:
 - BOLA
 - BOTA
 - CASA
 - CASARAO
 - CASO

Construção de uma árvore Trie

- Considere que queremos construir uma árvore Trie com os seguintes elementos:
 - BOLA
 - BOTA
 - CASA
 - CASARAO
 - CASO
 - LIVRARIA

Construção de uma árvore Trie

- Considere que queremos construir uma árvore Trie com os seguintes elementos:
 - BOLA
 - BOTA
 - CASA
 - CASARAO
 - CASO
 - LIVRARIA
 - LIVRO

Construção de uma árvore Trie

Construção de uma árvore Trie

- BOLA

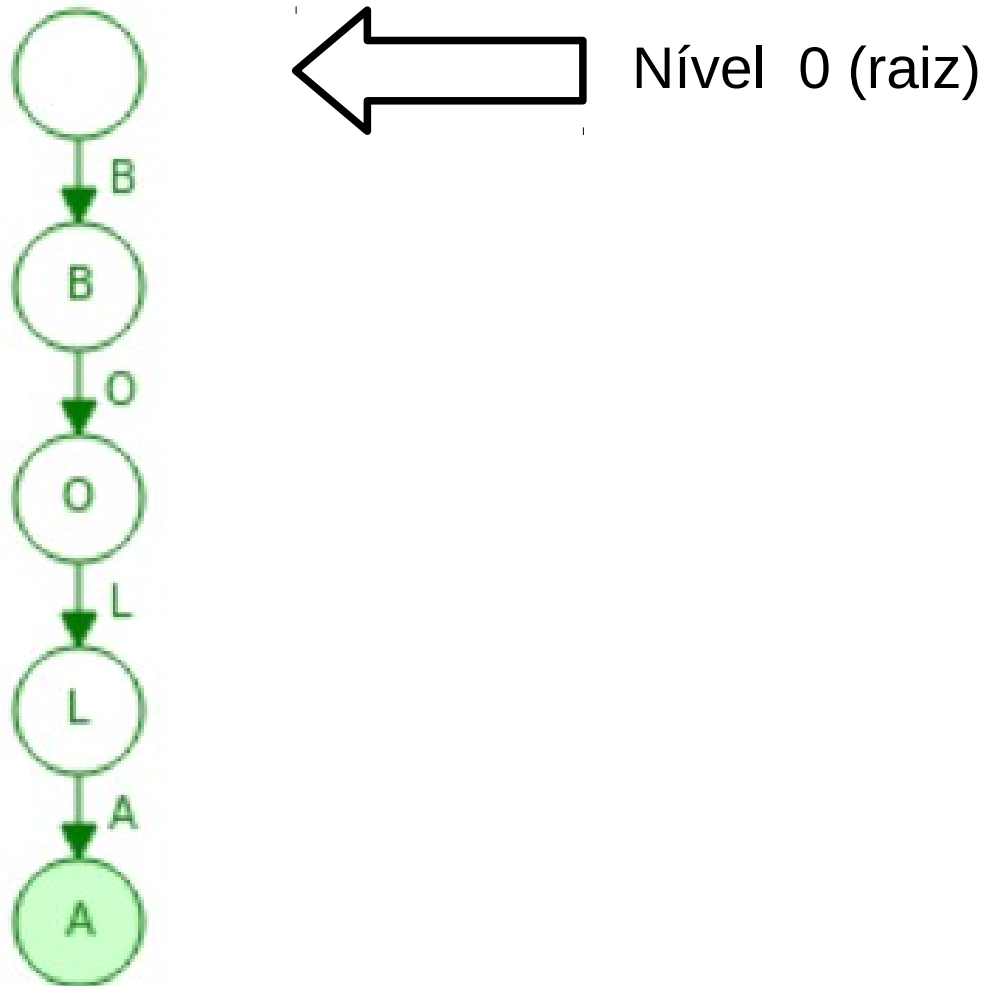
Construção de uma árvore Trie

- BOLA



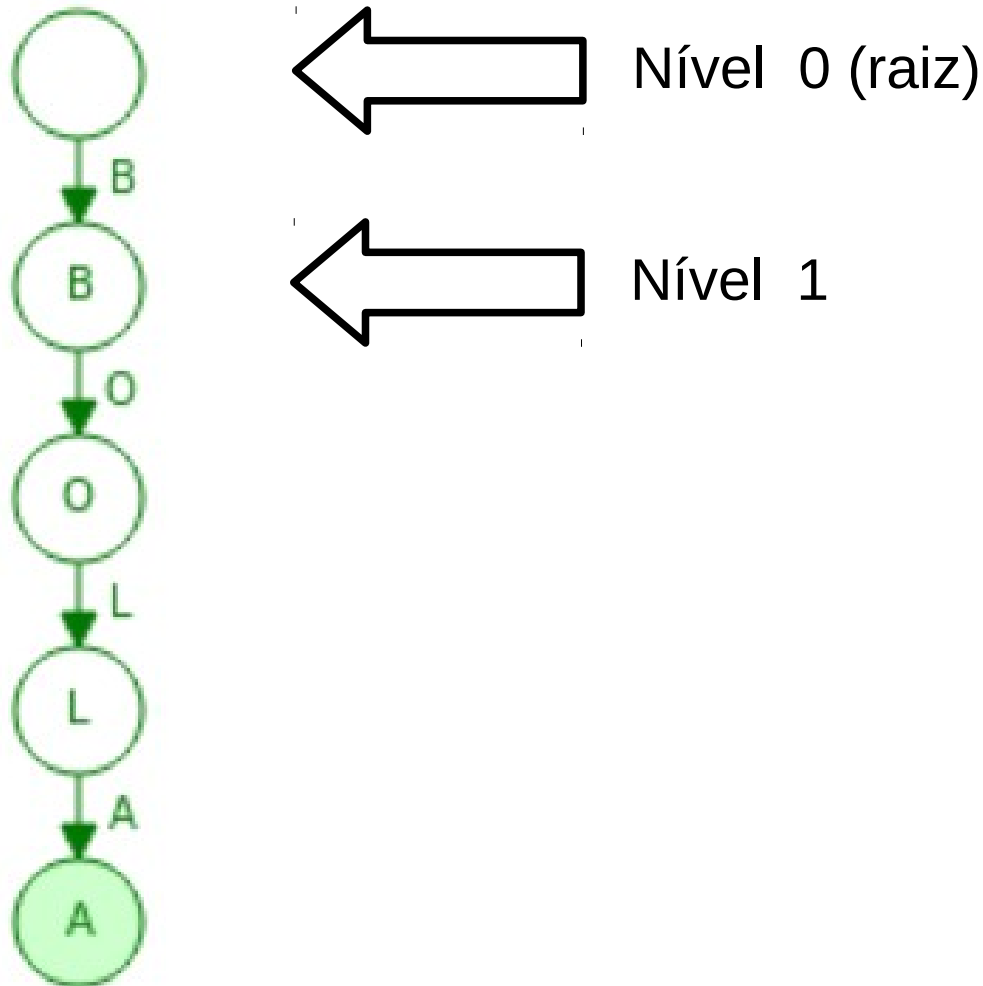
Construção de uma árvore Trie

- BOLA



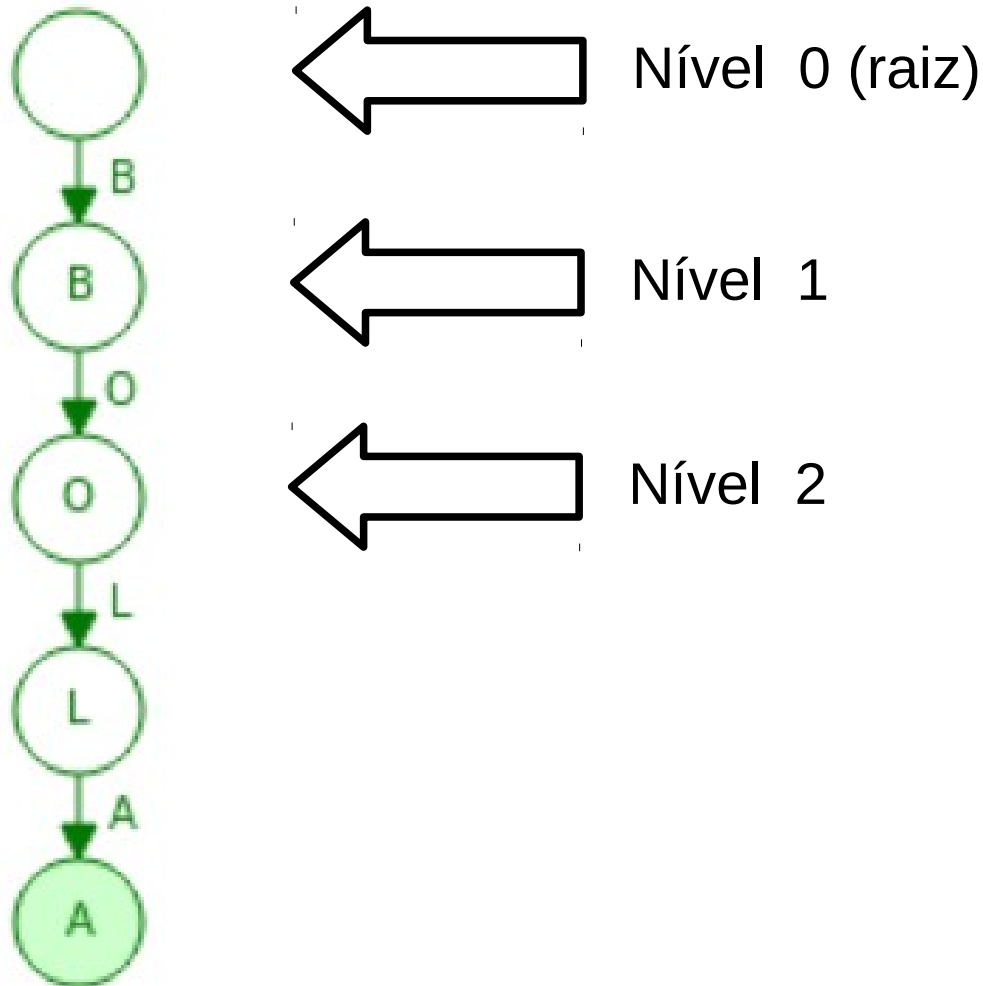
Construção de uma árvore Trie

- BOLA



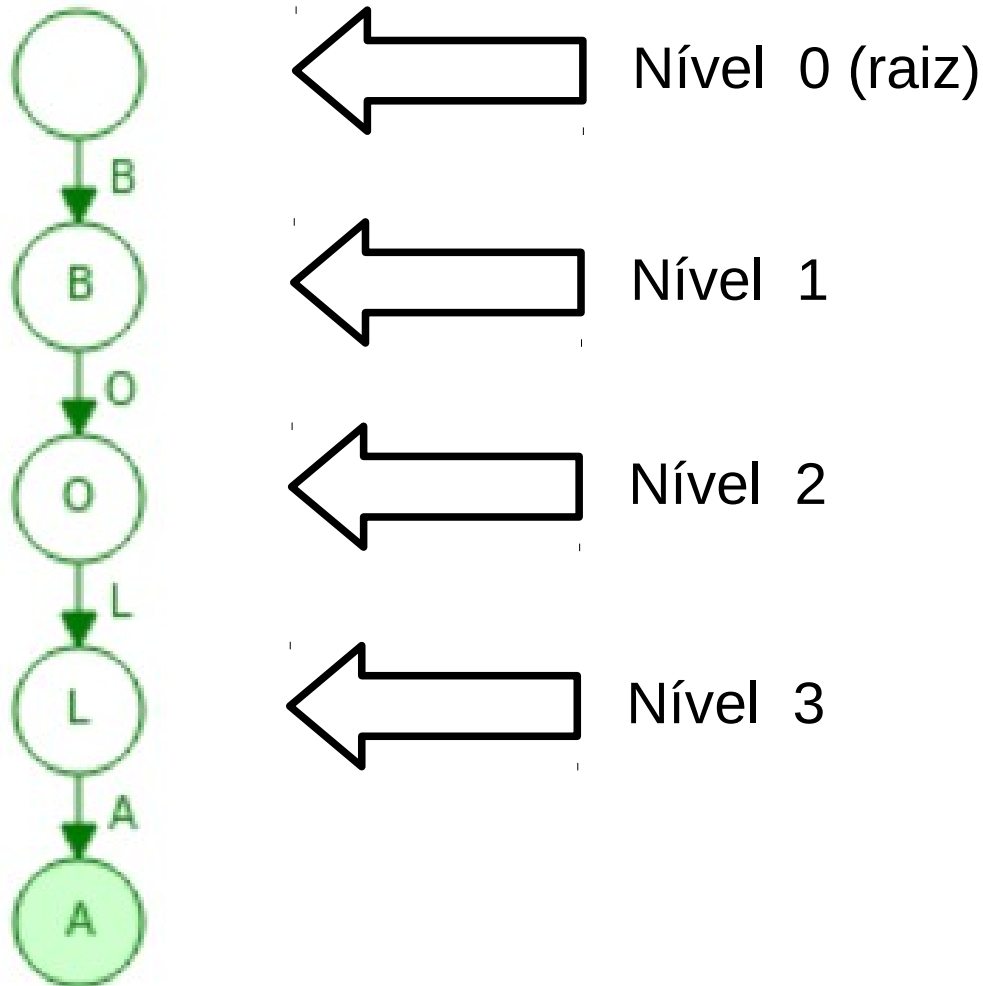
Construção de uma árvore Trie

- BOLA



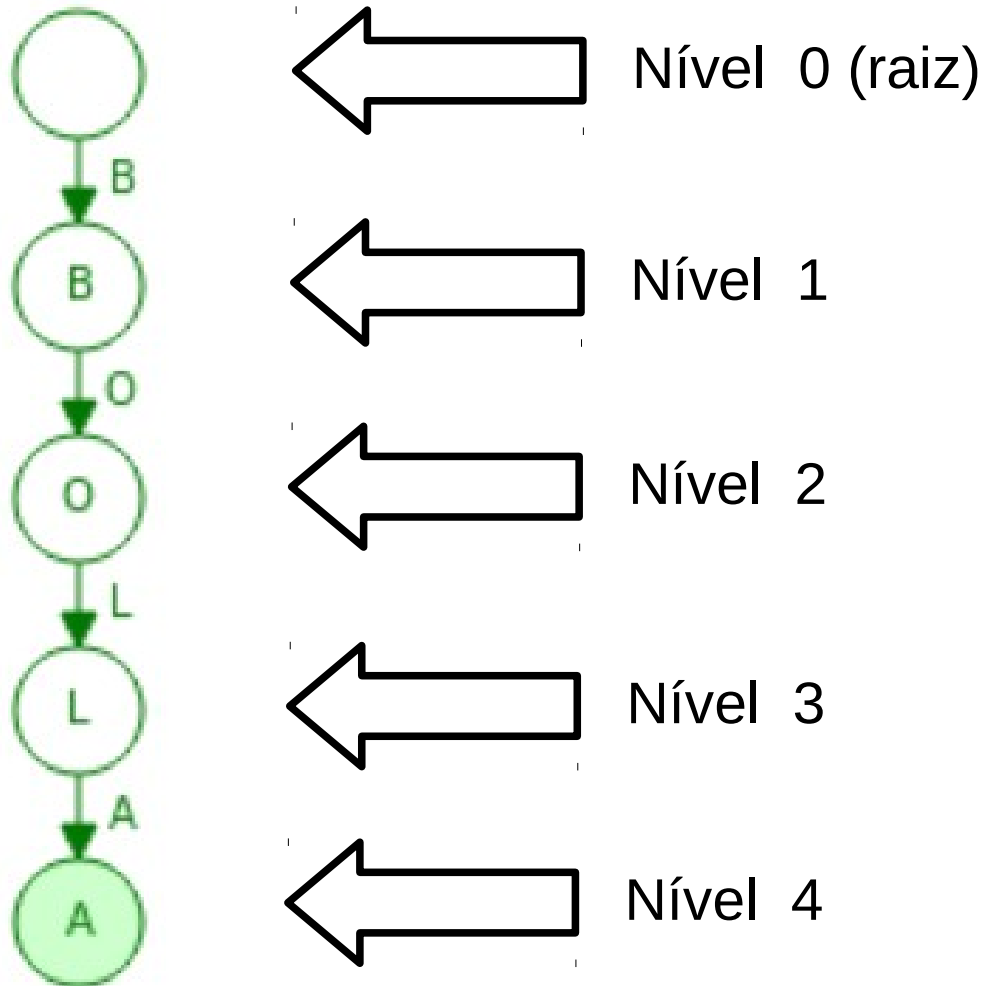
Construção de uma árvore Trie

- BOLA



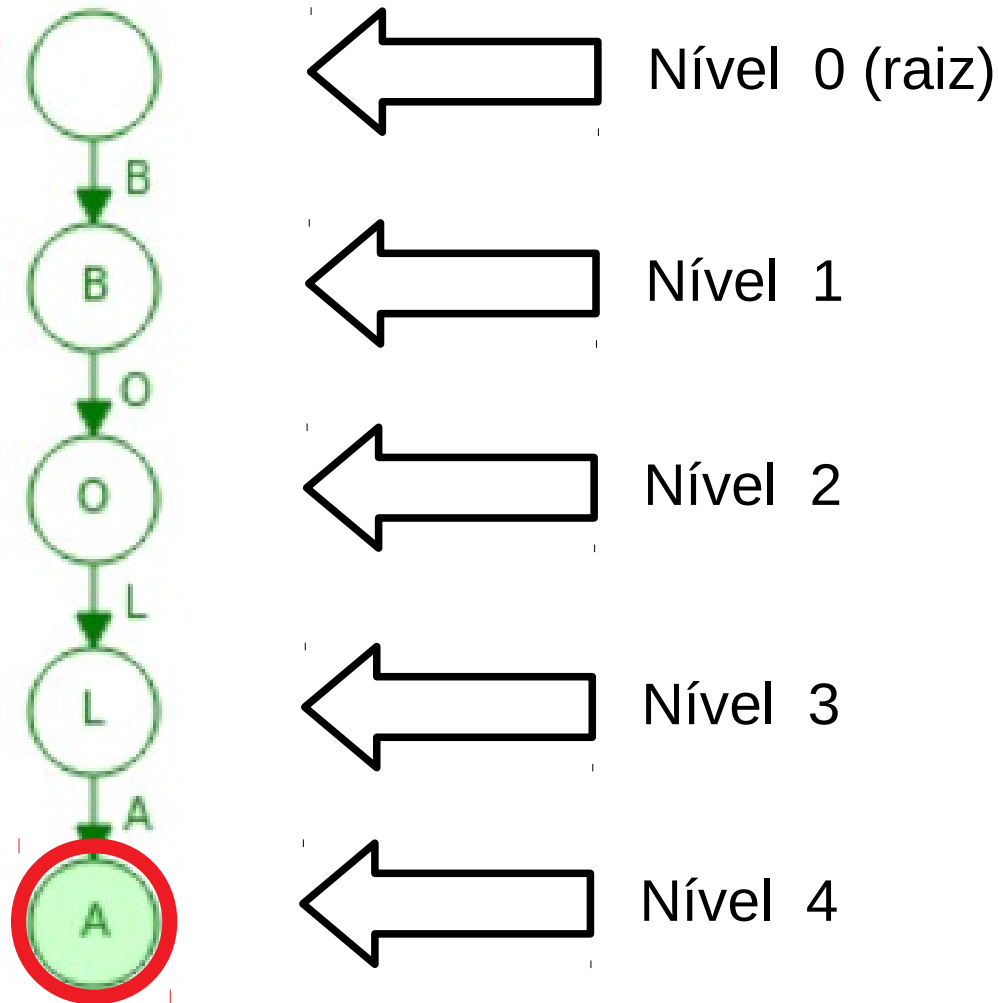
Construção de uma árvore Trie

- BOLA



Construção de uma árvore Trie

- BOLA



Construção de uma árvore Trie

Construção de uma árvore Trie

- BOTA

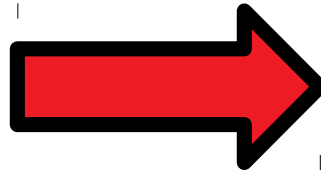
Construção de uma árvore Trie

- BOTA



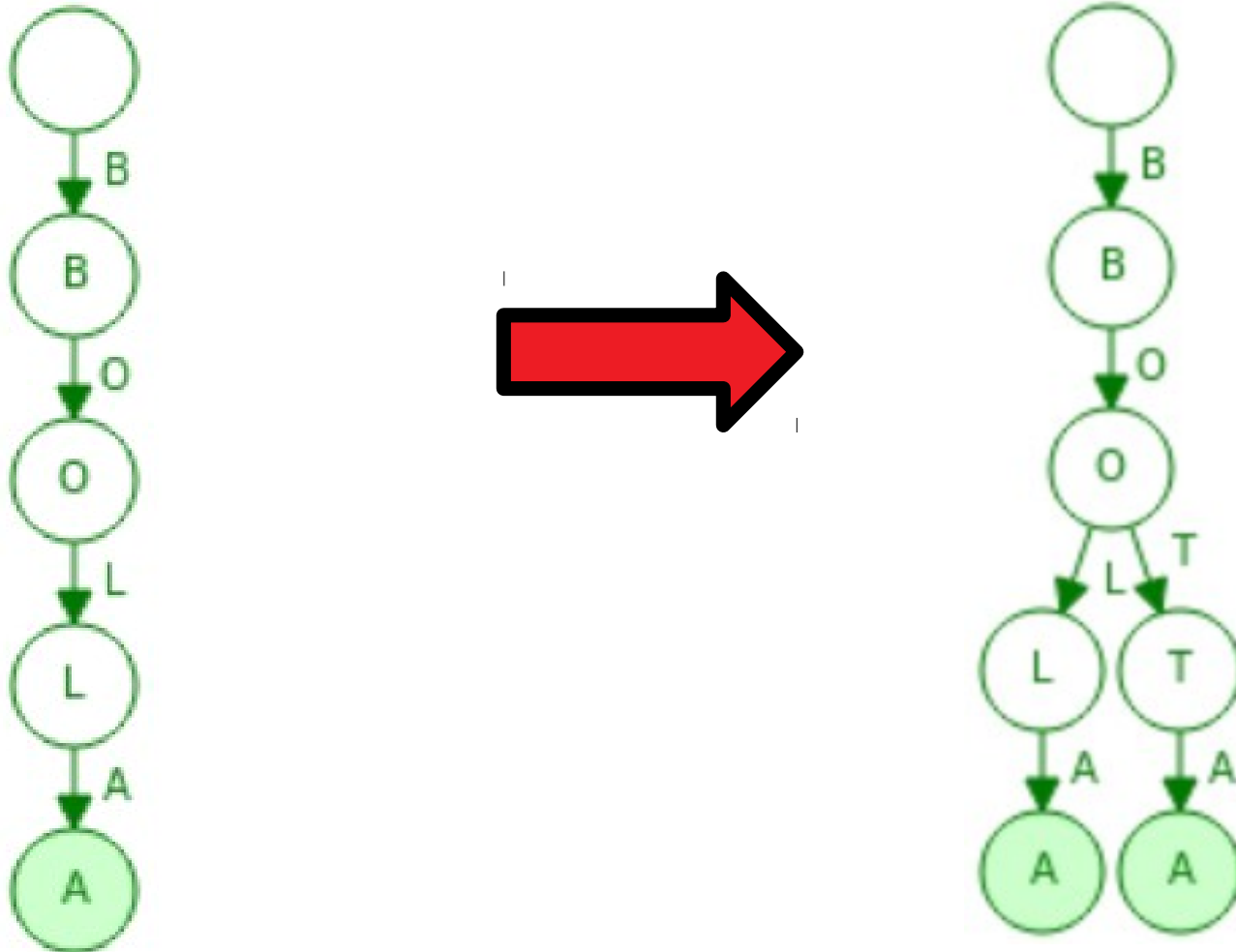
Construção de uma árvore Trie

- BOTA



Construção de uma árvore Trie

- BOTA



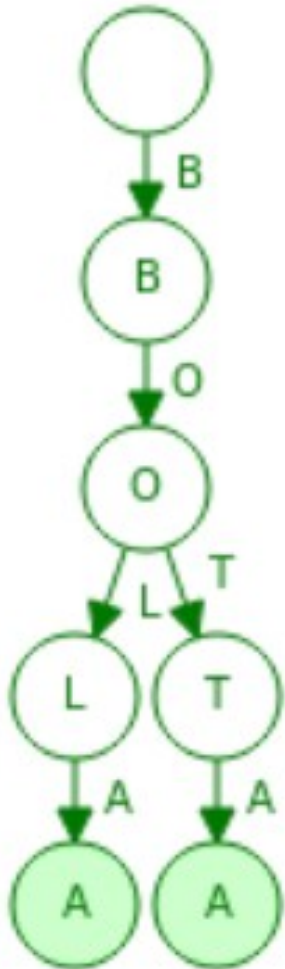
Construção de uma árvore Trie

Construção de uma árvore Trie

- CASA

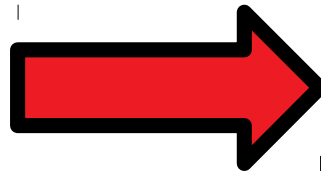
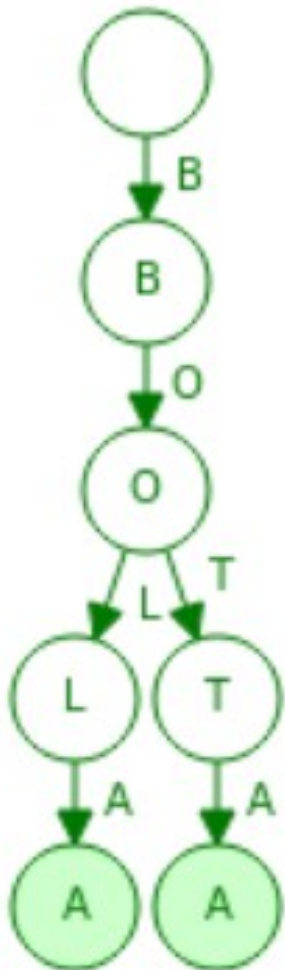
Construção de uma árvore Trie

- CASA



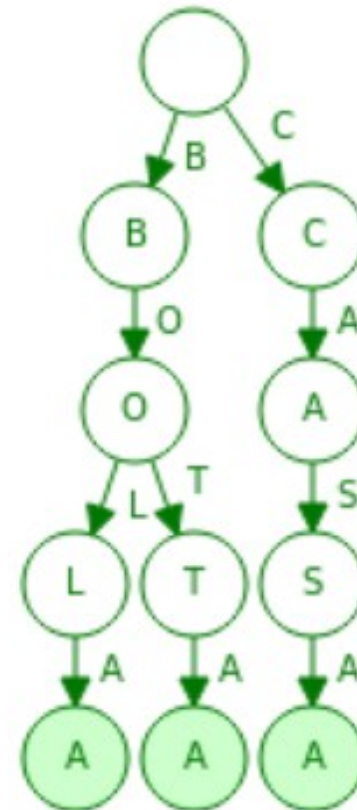
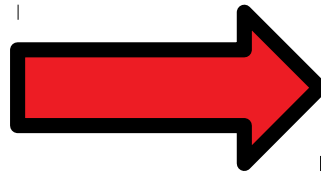
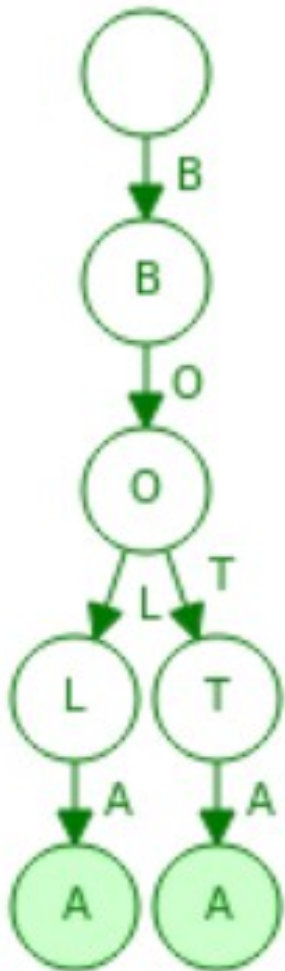
Construção de uma árvore Trie

- CASA



Construção de uma árvore Trie

- CASA



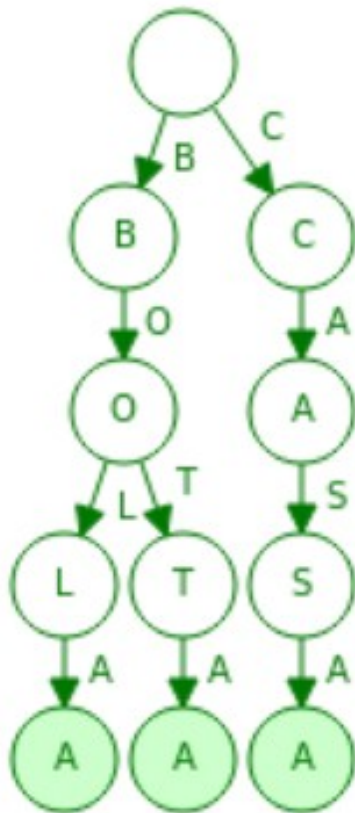
Construção de uma árvore Trie

Construção de uma árvore Trie

- CASARAO

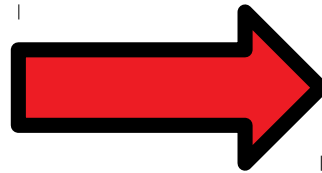
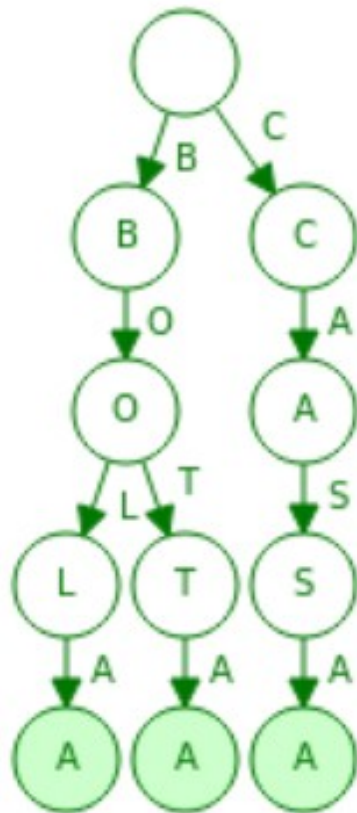
Construção de uma árvore Trie

- CASARAO



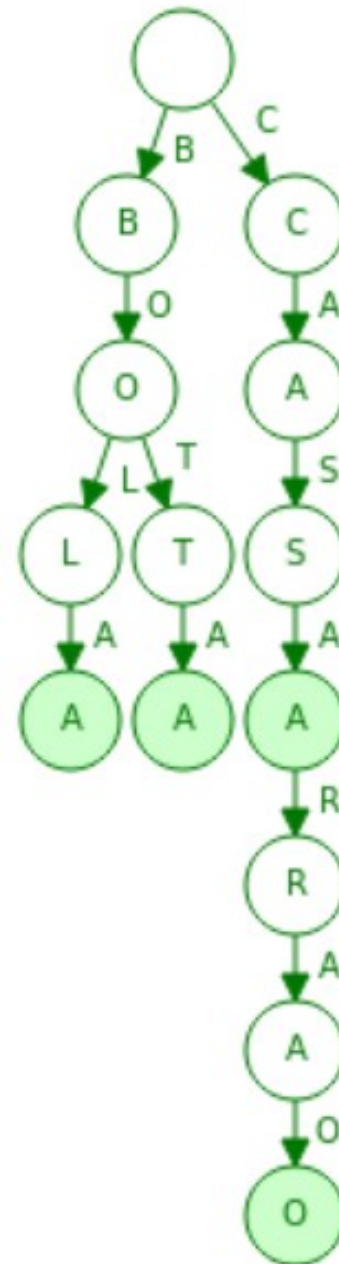
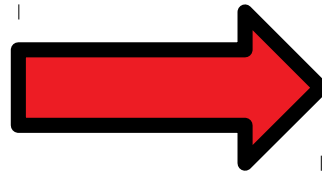
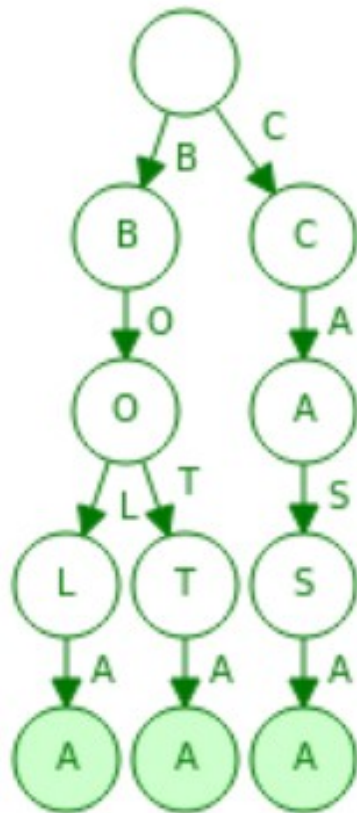
Construção de uma árvore Trie

- CASARAO



Construção de uma árvore Trie

- CASARAO



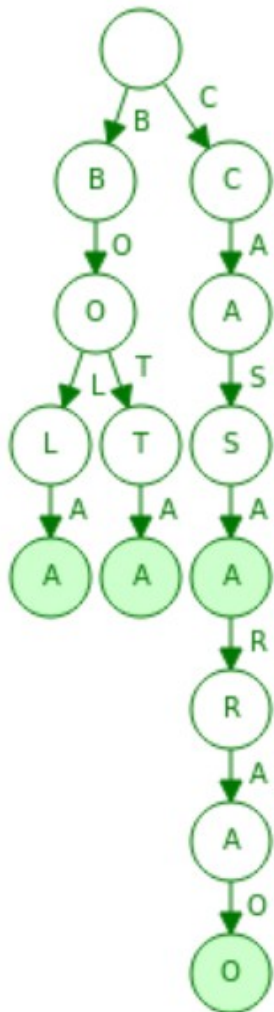
Construção de uma árvore Trie

Construção de uma árvore Trie

- CASO

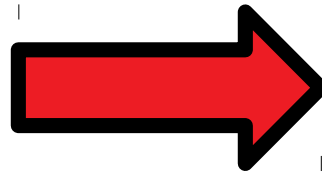
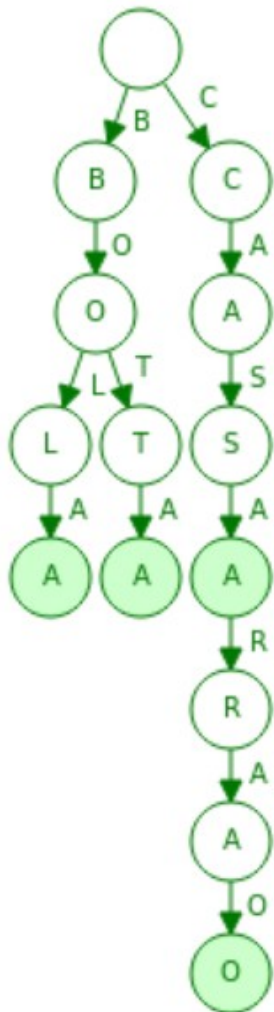
Construção de uma árvore Trie

- CASO



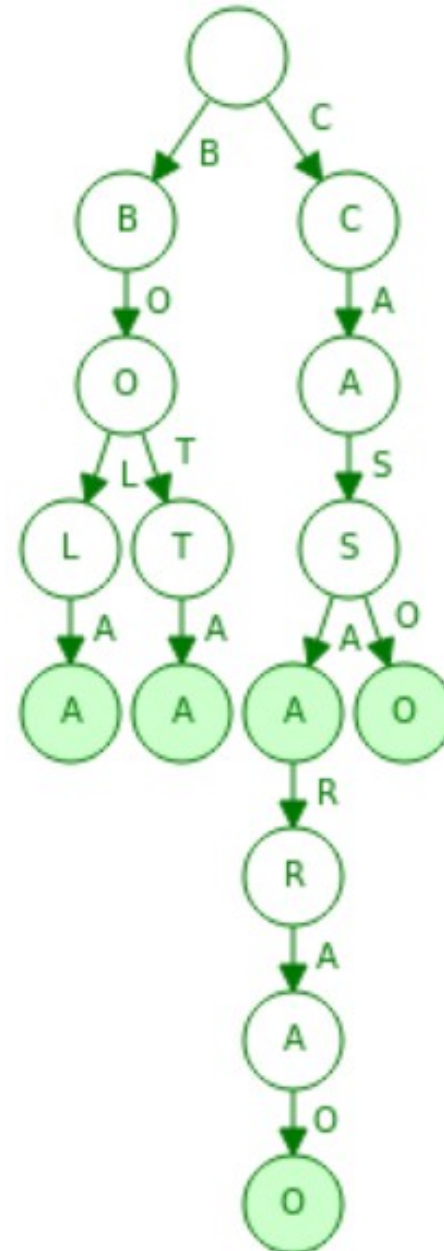
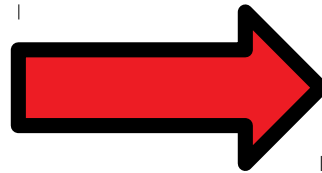
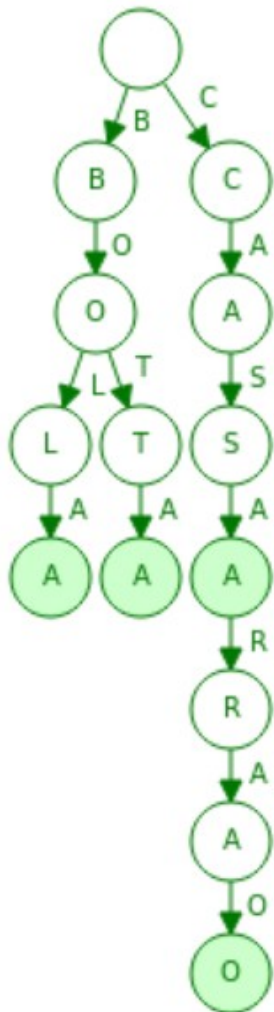
Construção de uma árvore Trie

- CASO



Construção de uma árvore Trie

- CASO



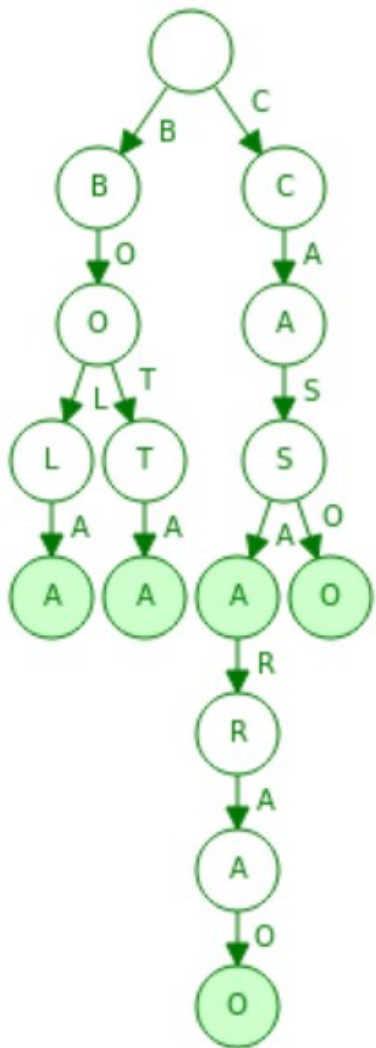
Construção de uma árvore Trie

Construção de uma árvore Trie

- LIVRARIA

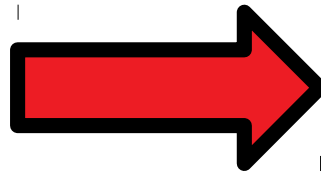
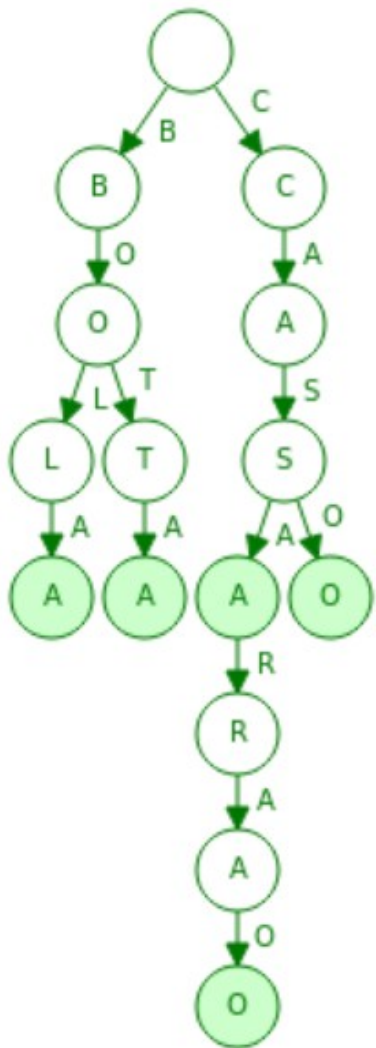
Construção de uma árvore Trie

- LIVRARIA



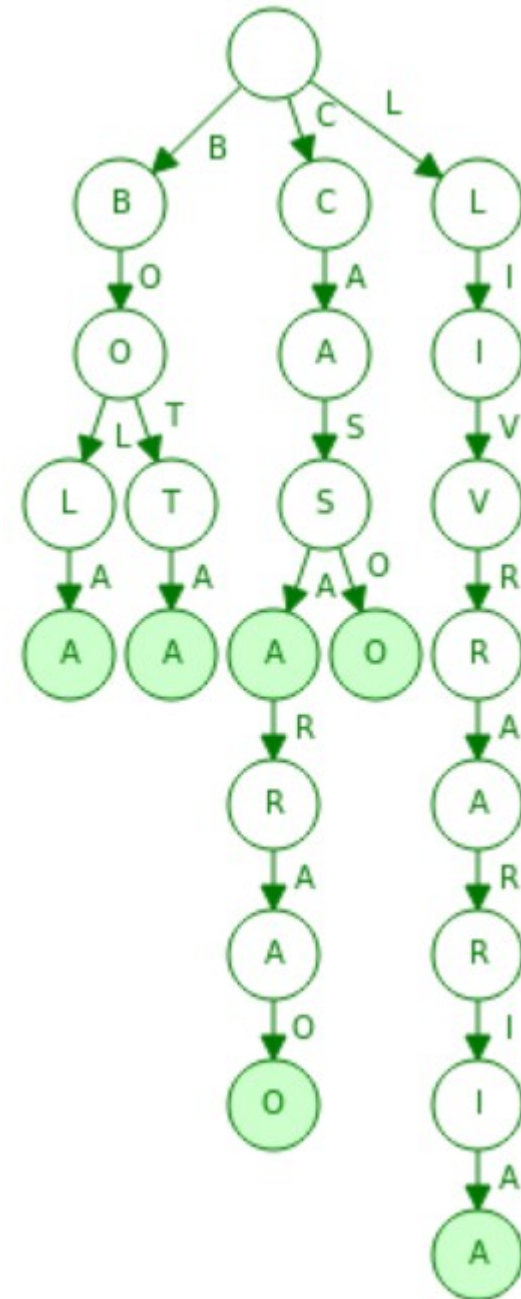
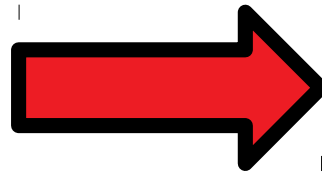
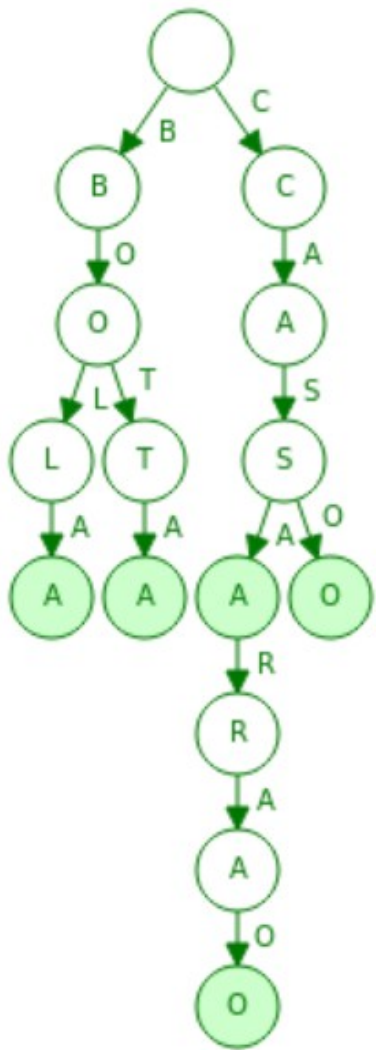
Construção de uma árvore Trie

- LIVRARIA



Construção de uma árvore Trie

- LIVRARIA



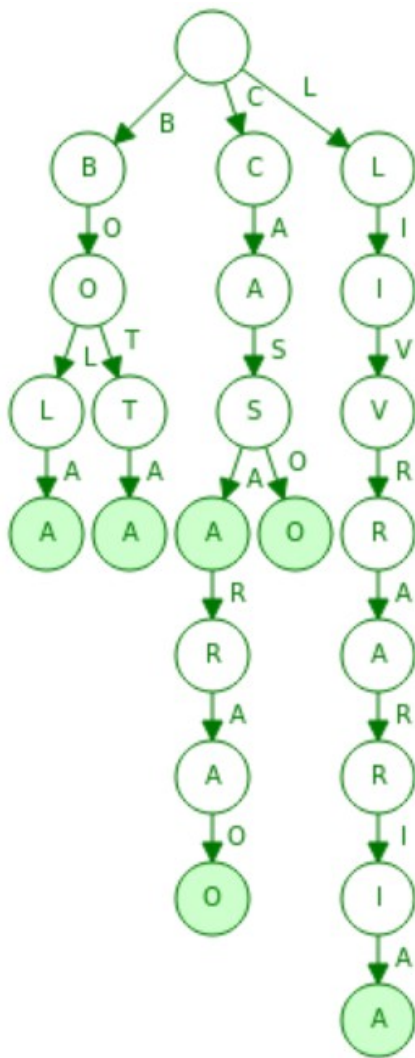
Construção de uma árvore Trie

Construção de uma árvore Trie

- LIVRO

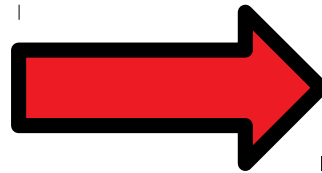
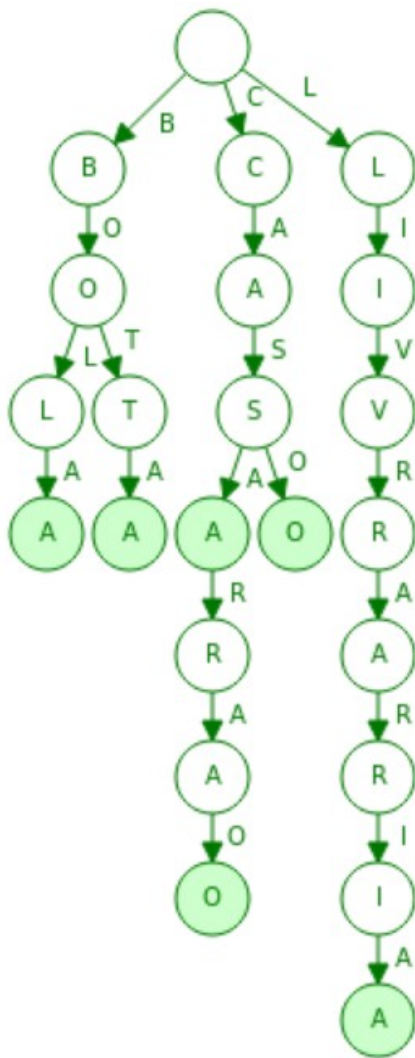
Construção de uma árvore Trie

- LIVRO



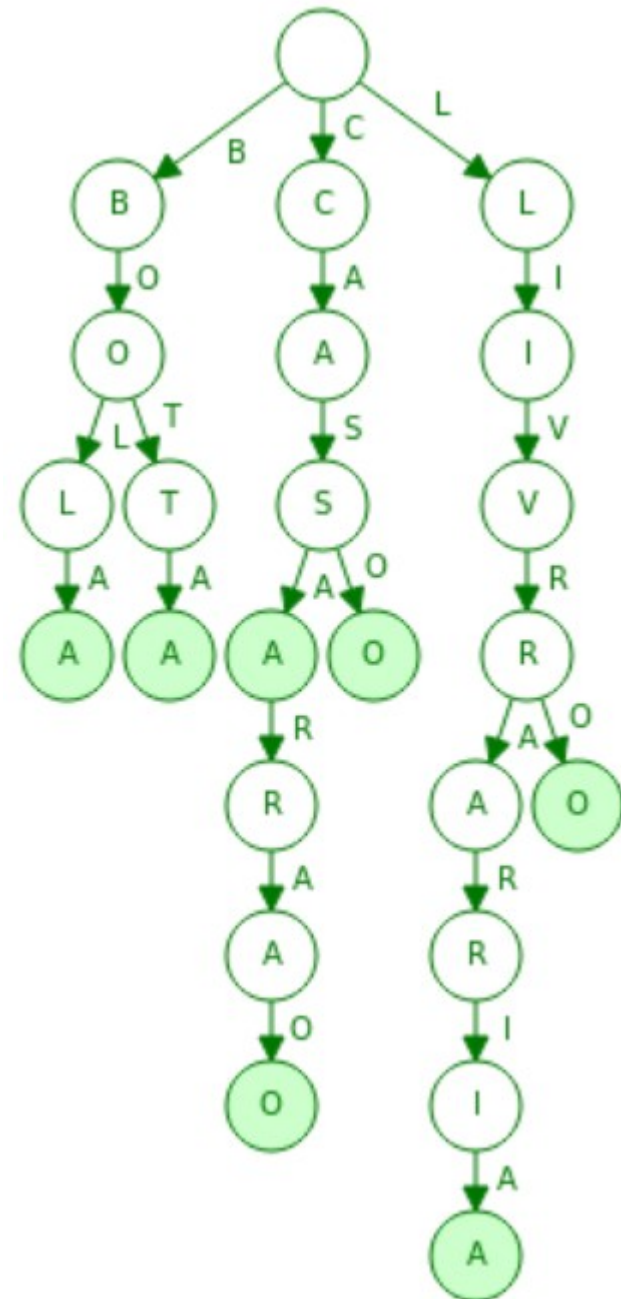
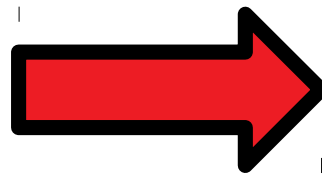
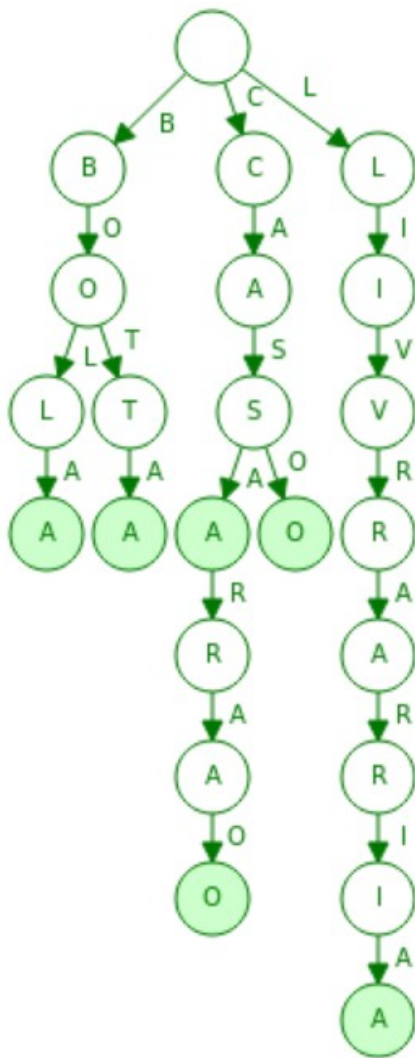
Construção de uma árvore Trie

- LIVRO



Construção de uma árvore Trie

- LIVRO



Operações

Operações

Operações

- Inserção

Operações

- Inserção
- Verifica membro

Operações

- Inserção
- Verifica membro
- Remoção

Inserção

Inserção

- Faz-se uma busca pela palavra a ser inserida.

Inserção

- Faz-se uma busca pela palavra a ser inserida.
- Se ela já estiver na Trie, então nada é feito.

Inserção

- Faz-se uma busca pela palavra a ser inserida.
- Se ela já estiver na Trie, então nada é feito.
- Caso contrário, é recuperado (pesquisado) o nó até onde acontece a maior substring da palavra a ser inserida.

Inserção

- Faz-se uma busca pela palavra a ser inserida.
- Se ela já estiver na Trie, então nada é feito.
- Caso contrário, é recuperado (pesquisado) o nó até onde acontece a maior substring da palavra a ser inserida.
- O restante dos seus caracteres são adicionados na Trie a partir daquele nó.

Inserção

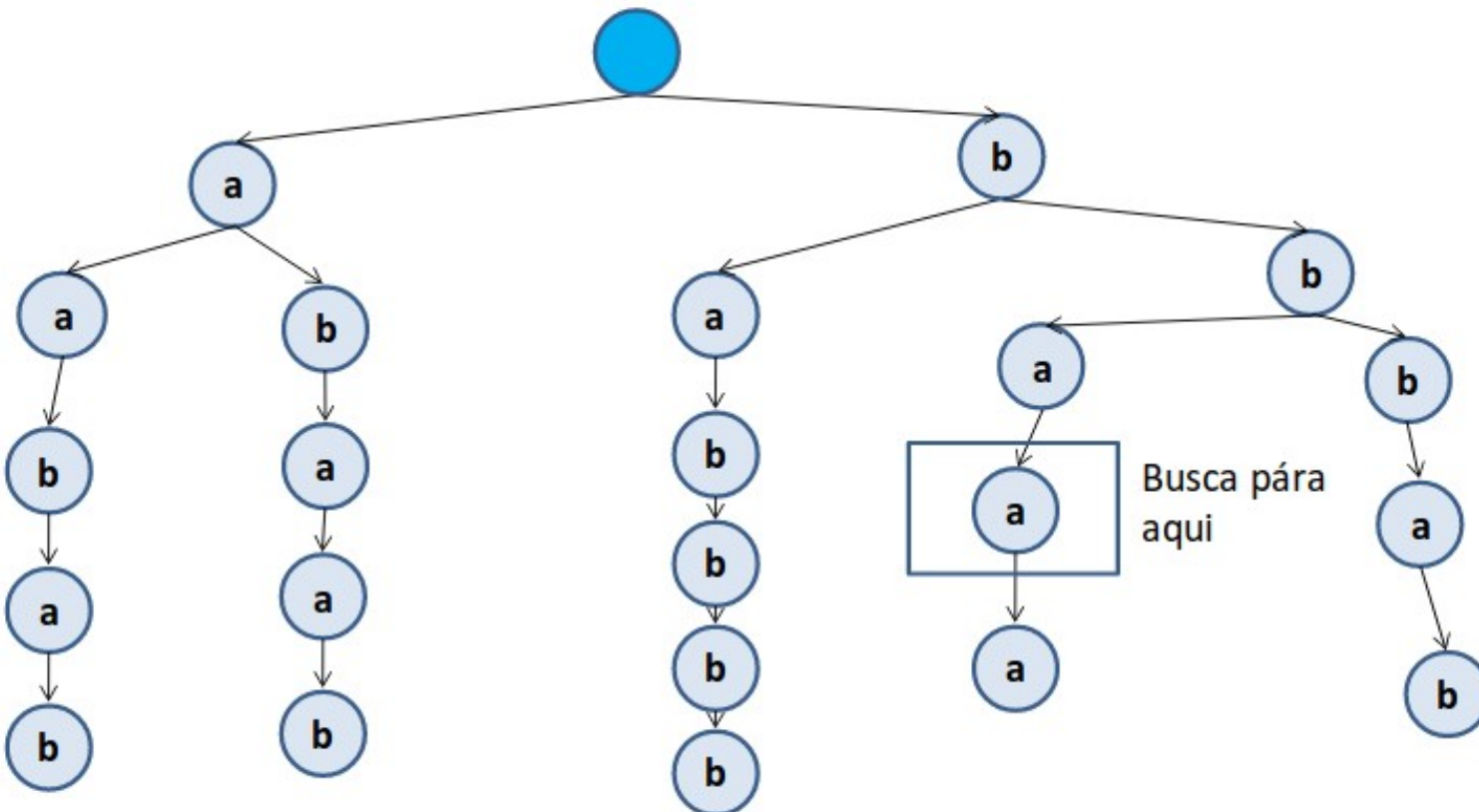
- Exemplo

Inserção

- Exemplo
 - Inserção bbaabb

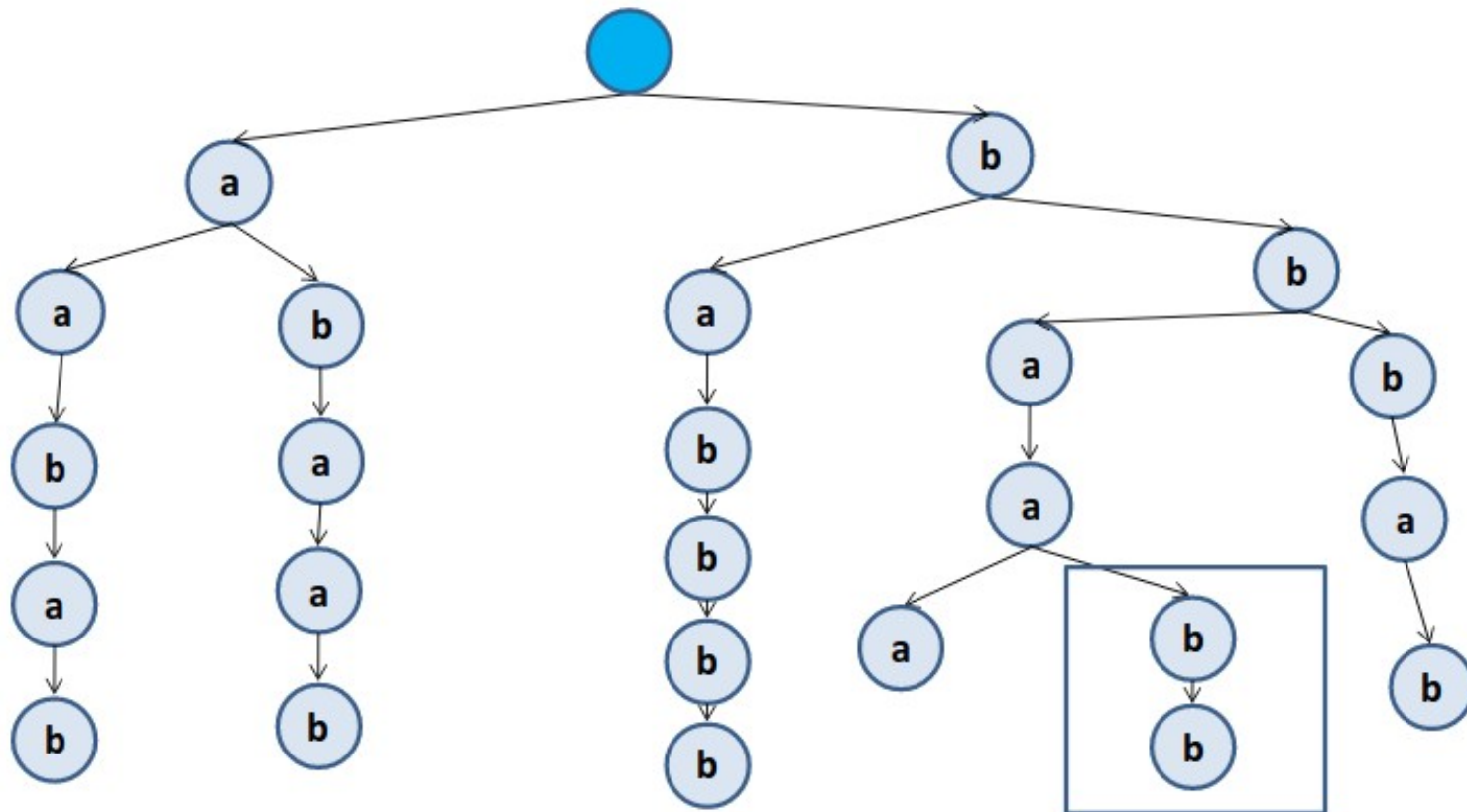
Inserção

- Exemplo
 - Inserção bbaabb



Inserção

- Exemplo
 - Inserção bbaabb



Verificação de membro

Verificação de membro

- Procedimento para verificar se determinada palavra é membro da Trie:

Verificação de membro

- Procedimento para verificar se determinada palavra é membro da Trie:
 1. Busca no nível superior o nó que confere com o primeiro caractere da chave.

Verificação de membro

- Procedimento para verificar se determinada palavra é membro da Trie:
 1. Busca no nível superior o nó que confere com o primeiro caractere da chave.
 2. Se não houver, retorna FALSO

Verificação de membro

- Procedimento para verificar se determinada palavra é membro da Trie:
 1. Busca no nível superior o nó que confere com o primeiro caractere da chave.
 2. Se não houver, retorna FALSO
 3. Senão

Verificação de membro

- Procedimento para verificar se determinada palavra é membro da Trie:
 1. Busca no nível superior o nó que confere com o primeiro caractere da chave.
 2. Se não houver, retorna FALSO
 3. Senão
 4. Se o caractere que confere é '\0'

Verificação de membro

- Procedimento para verificar se determinada palavra é membro da Trie:
 1. Busca no nível superior o nó que confere com o primeiro caractere da chave.
 2. Se não houver, retorna FALSO
 3. Senão
 4. Se o caractere que confere é '\0'
 5. Retorna VERDADEIRO

Verificação de membro

- Procedimento para verificar se determinada palavra é membro da Trie:
 1. Busca no nível superior o nó que confere com o primeiro caractere da chave.
 2. Se não houver, retorna FALSO
 3. Senão
 4. Se o caractere que confere é '\0'
 5. Retorna VERDADEIRO
 6. Senão

Verificação de membro

- Procedimento para verificar se determinada palavra é membro da Trie:
 1. Busca no nível superior o nó que confere com o primeiro caractere da chave.
 2. Se não houver, retorna FALSO
 3. Senão
 4. Se o caractere que confere é '\0'
 5. Retorna VERDADEIRO
 6. Senão
 7. Move para a subTrie que confere com esse caractere

Verificação de membro

- Procedimento para verificar se determinada palavra é membro da Trie:
 1. Busca no nível superior o nó que confere com o primeiro caractere da chave.
 2. Se não houver, retorna FALSO
 3. Senão
 4. Se o caractere que confere é '\0'
 5. Retorna VERDADEIRO
 6. Senão
 7. Move para a subTrie que confere com esse caractere
 8. Avança para o próximo caractere na chave

Verificação de membro

- Procedimento para verificar se determinada palavra é membro da Trie:
 1. Busca no nível superior o nó que confere com o primeiro caractere da chave.
 2. Se não houver, retorna FALSO
 3. Senão
 4. Se o caractere que confere é '\0'
 5. Retorna VERDADEIRO
 6. Senão
 7. Move para a subTrie que confere com esse caractere
 8. Avança para o próximo caractere na chave
 9. Vá para o passo 1

Remoção

Remoção

- Busca-se o nó (elemento) que representa o final da palavra que será removida.

Remoção

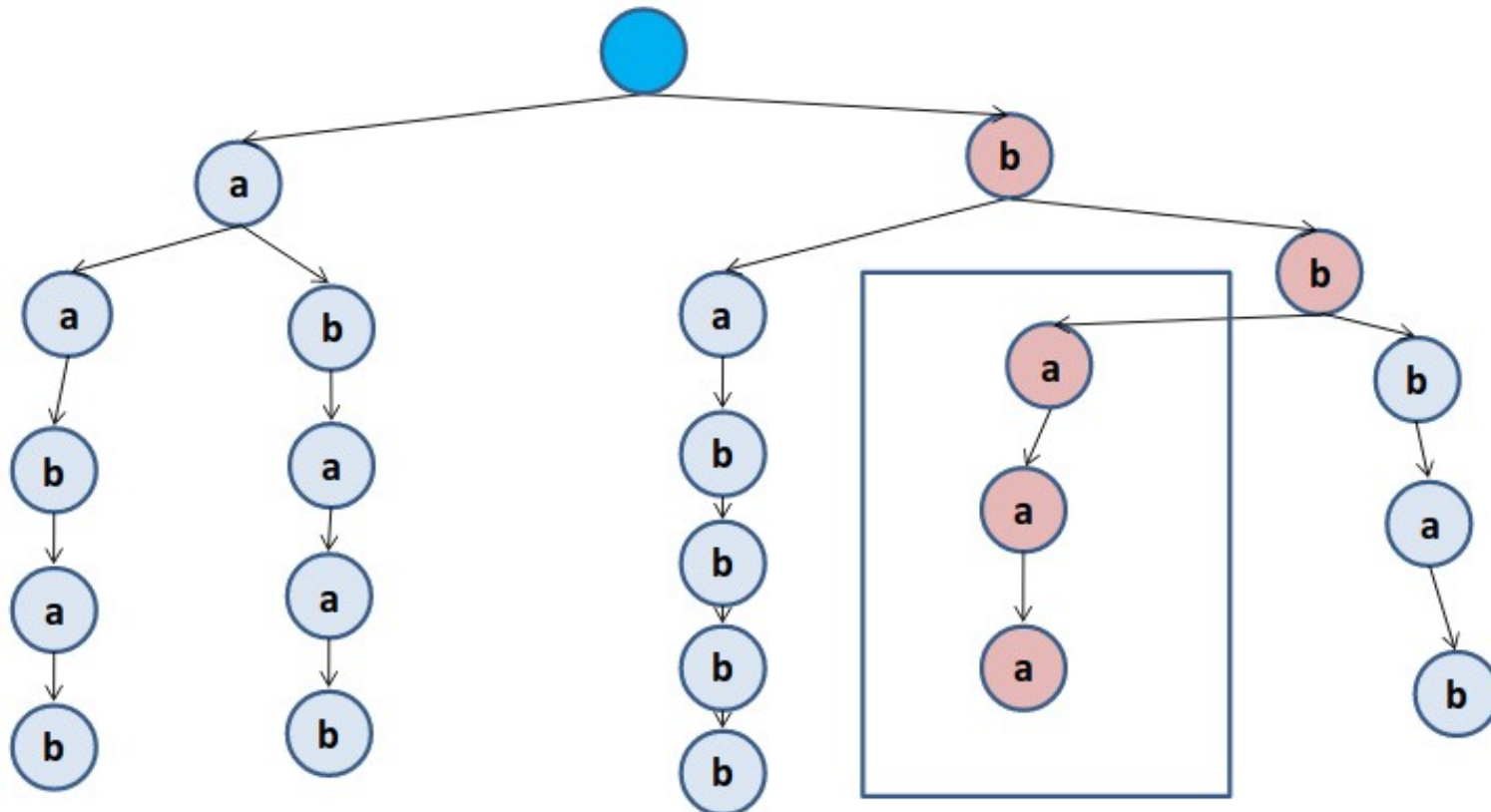
- Busca-se o nó (elemento) que representa o final da palavra que será removida.
- São removidos os nós que possuem apenas um filho pelo caminho ascendente.

Remoção

- Busca-se o nó (elemento) que representa o final da palavra que será removida.
- São removidos os nós que possuem apenas um filho pelo caminho ascendente.
- A remoção é concluída quando se encontra um nó com mais de um filho.

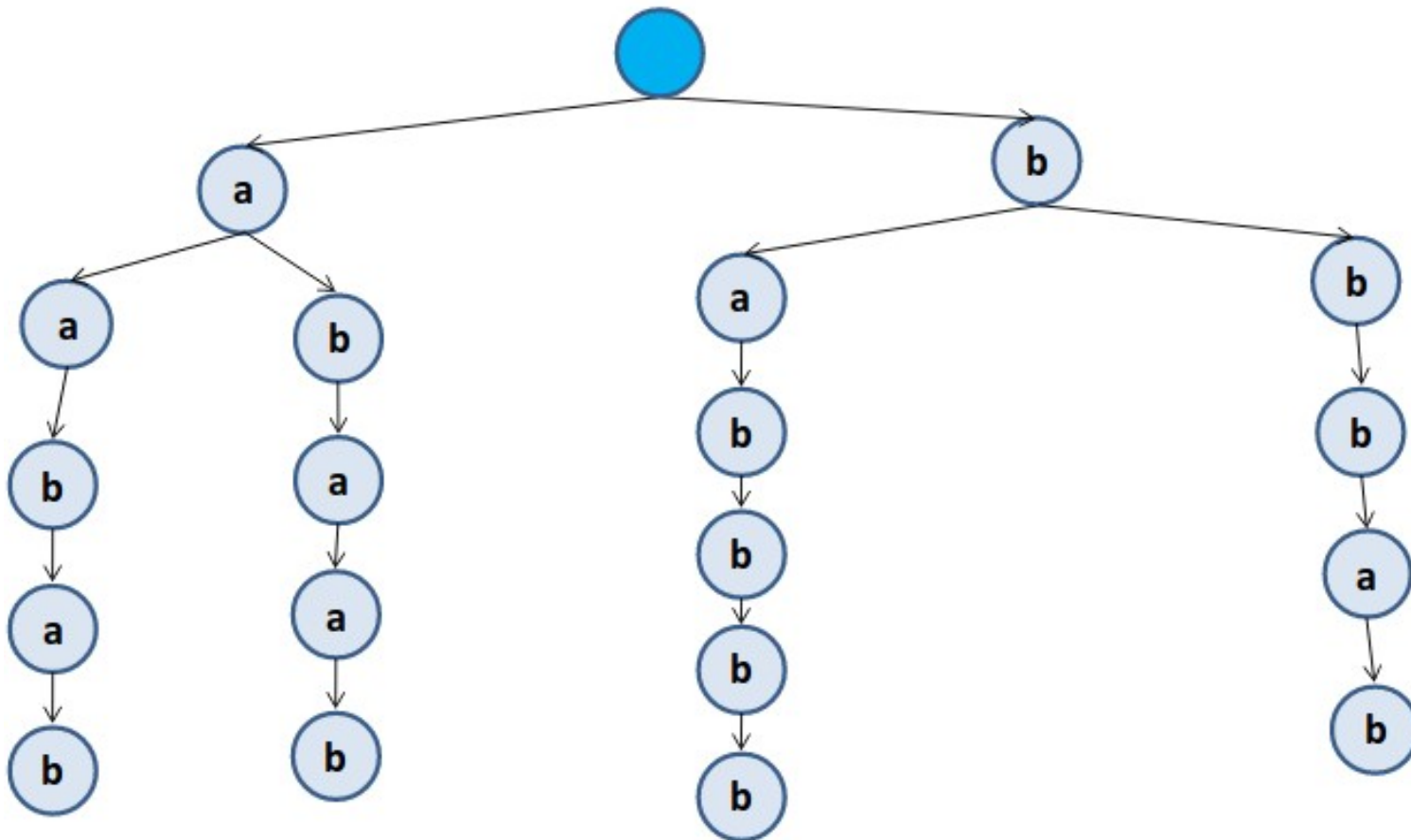
Remoção

- Exemplo:
 - Remoção 'bbaaa'



Remoção

- Exemplo:
 - Remoção 'bbaaa'



Operações em Tries

Operações em Tries

- Complexidade

Operações em Tries

- Complexidade
 - A altura da árvore é igual ao comprimento da chave mais longa.

Operações em Tries

- Complexidade
 - 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.

Operações em Tries

- Complexidade
 - 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 é $O(A.K)$** , onde A é o tamanho do alfabeto e K é o tamanho da chave.

Operações em Tries

- Complexidade
 - 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 é $O(A.K)$** , onde A é o tamanho do alfabeto e K é o tamanho da chave.
 - A utilização de uma Trie só compensa se o acesso aos componentes individuais das chaves for bastante rápido.

Operações em Tries

- Complexidade

- 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 é $O(A.K)$** , onde A é o tamanho do alfabeto e K é o tamanho da chave.
- A utilização de uma Trie só compensa se o acesso aos componentes individuais das chaves for bastante rápido.
- Quanto maior a estrutura, mais eficiente será o uso do espaço.

Operações em Tries

- Complexidade
 - 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 é $O(A.K)$** , onde A é o tamanho do alfabeto e K é o tamanho da chave.
 - A utilização de uma Trie só compensa se o acesso aos componentes individuais das chaves for bastante rápido.
 - Quanto maior a estrutura, mais eficiente será o uso do espaço.
 - Para enfrentar o desperdício de espaço com estruturas pequenas foram criadas outras espécies de árvores, tais como as árvores Patrícias.

Tipos de Tries

Tipos de Tries

Tipos de Tries

- Existem muitas variantes e tipos de Tries

Tipos de Tries

- Existem muitas variantes e tipos de Tries
 - R-Way

Tipos de Tries

- Existem muitas variantes e tipos de Tries
 - R-Way
 - DST (*Digital Search Tree*)

Tipos de Tries

- Existem muitas variantes e tipos de Tries
 - R-Way
 - DST (*Digital Search Tree*)
 - Suffix Tree

Tipos de Tries

- Existem muitas variantes e tipos de Tries
 - R-Way
 - DST (*Digital Search Tree*)
 - Suffix Tree
 - Patrícia Tree

Tipos de Tries

- Existem muitas variantes e tipos de Tries
 - R-Way
 - DST (*Digital Search Tree*)
 - Suffix Tree
 - Patrícia Tree
 - DAWG (*Directed Acyclic Word Graph*)

Tipos de Tries

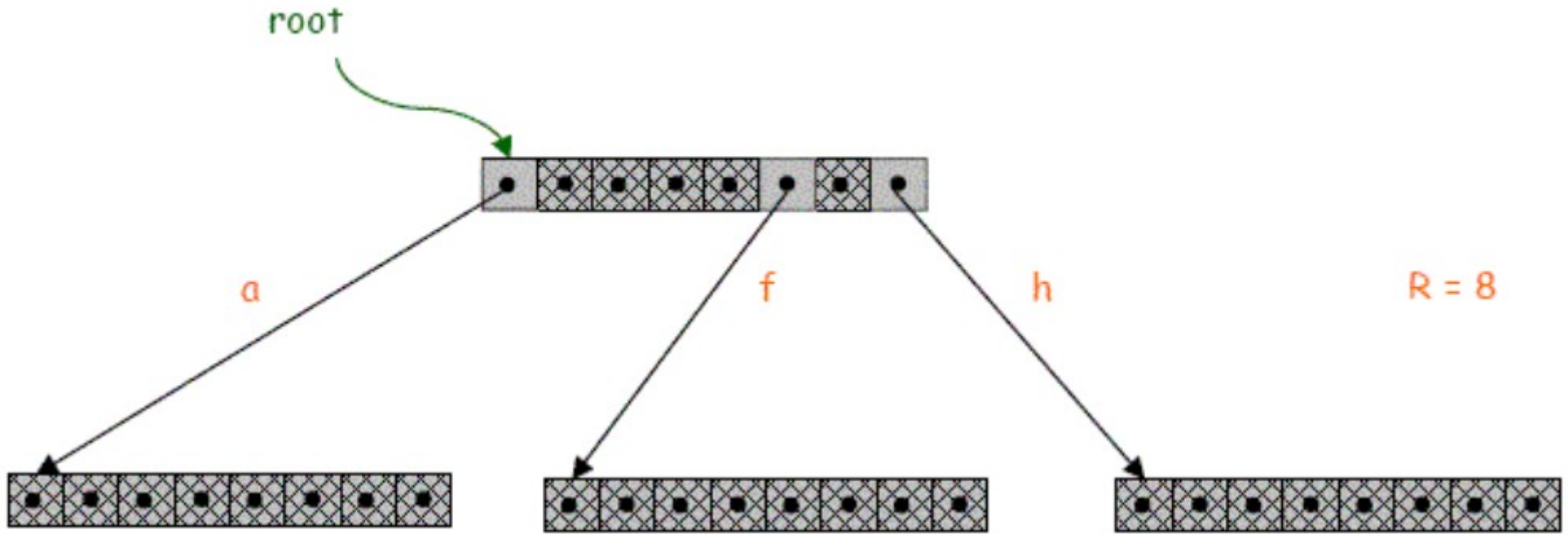
- Existem muitas variantes e tipos de Tries
 - R-Way
 - DST (*Digital Search Tree*)
 - Suffix Tree
 - Patrícia Tree
 - DAWG (*Directed Acyclic Word Graph*)
 - TST

Tipos de Tries

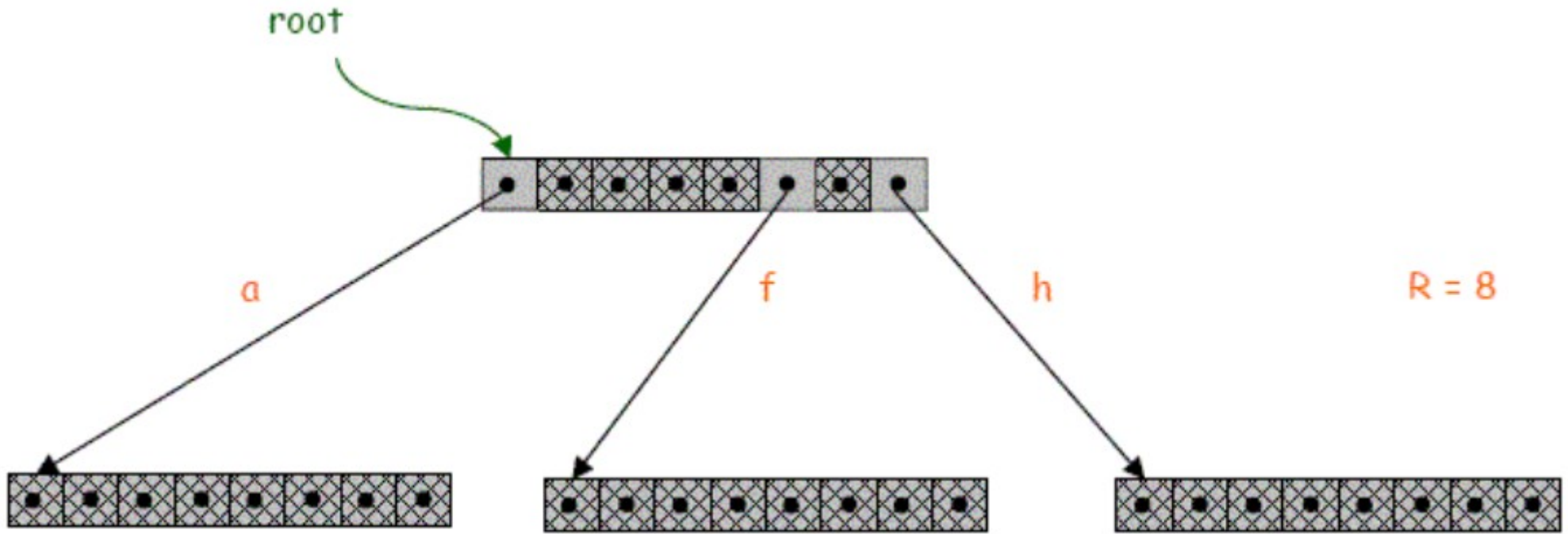
- Existem muitas variantes e tipos de Tries
 - R-Way
 - DST (*Digital Search Tree*)
 - Suffix Tree
 - Patrícia Tree
 - DAWG (*Directed Acyclic Word Graph*)
 - TST
 - ...

R-Way Trie

R-Way Trie

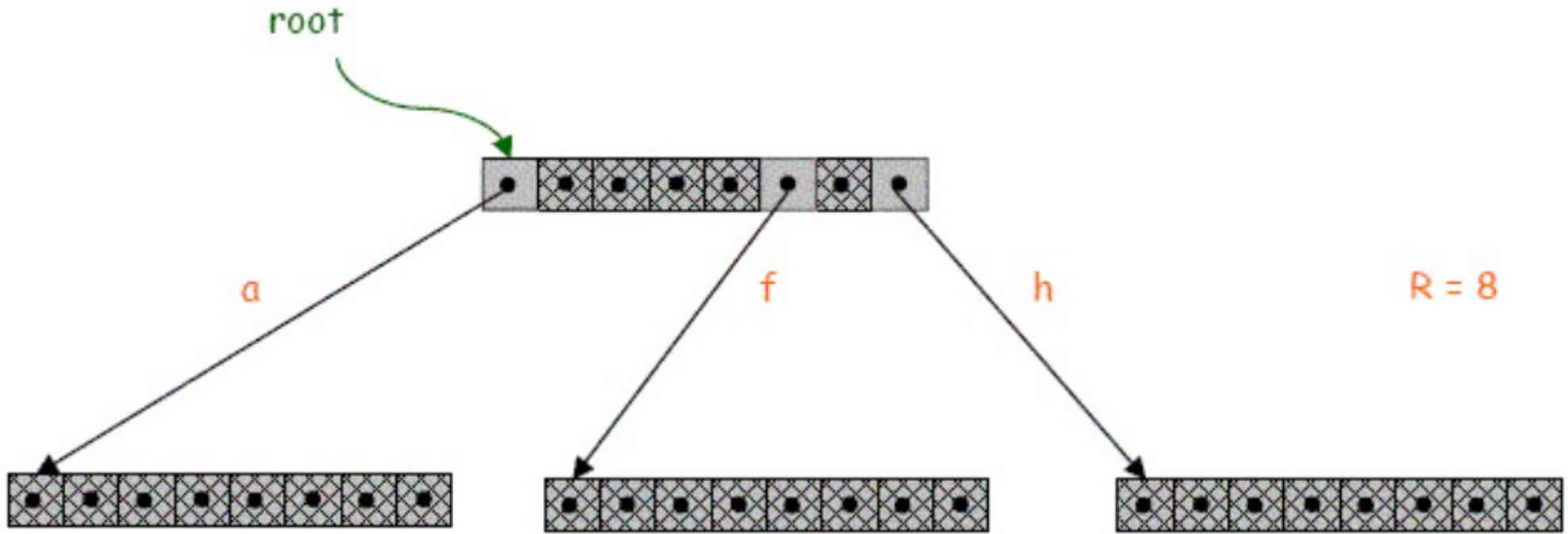


R-Way Trie



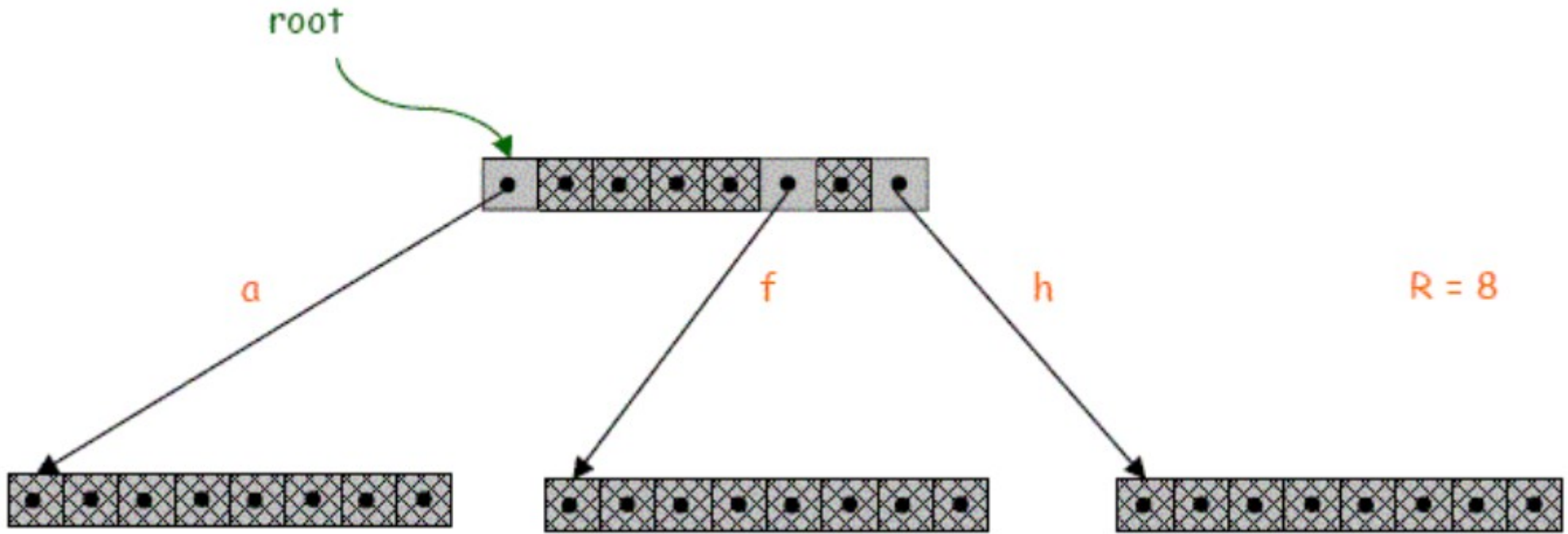
- Cada nó aloca espaço para todo o alfabeto.

R-Way Trie



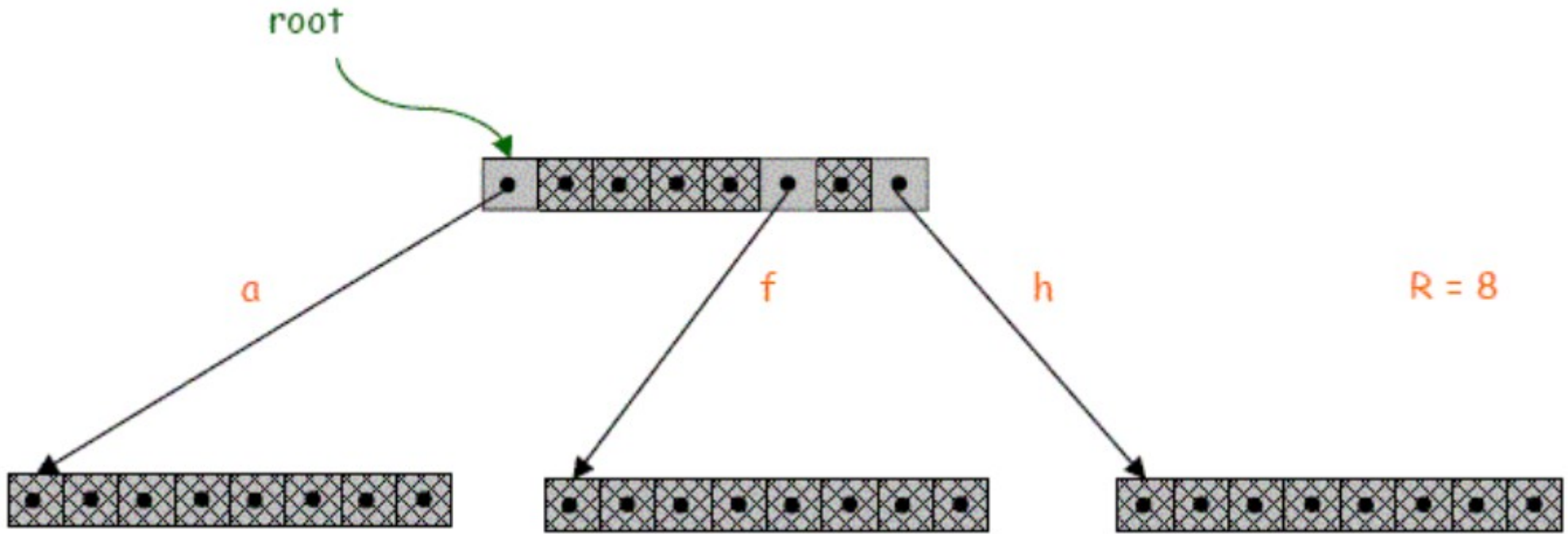
- Cada nó aloca espaço para todo o alfabeto.
- Há desperdício de espaço.

R-Way Trie



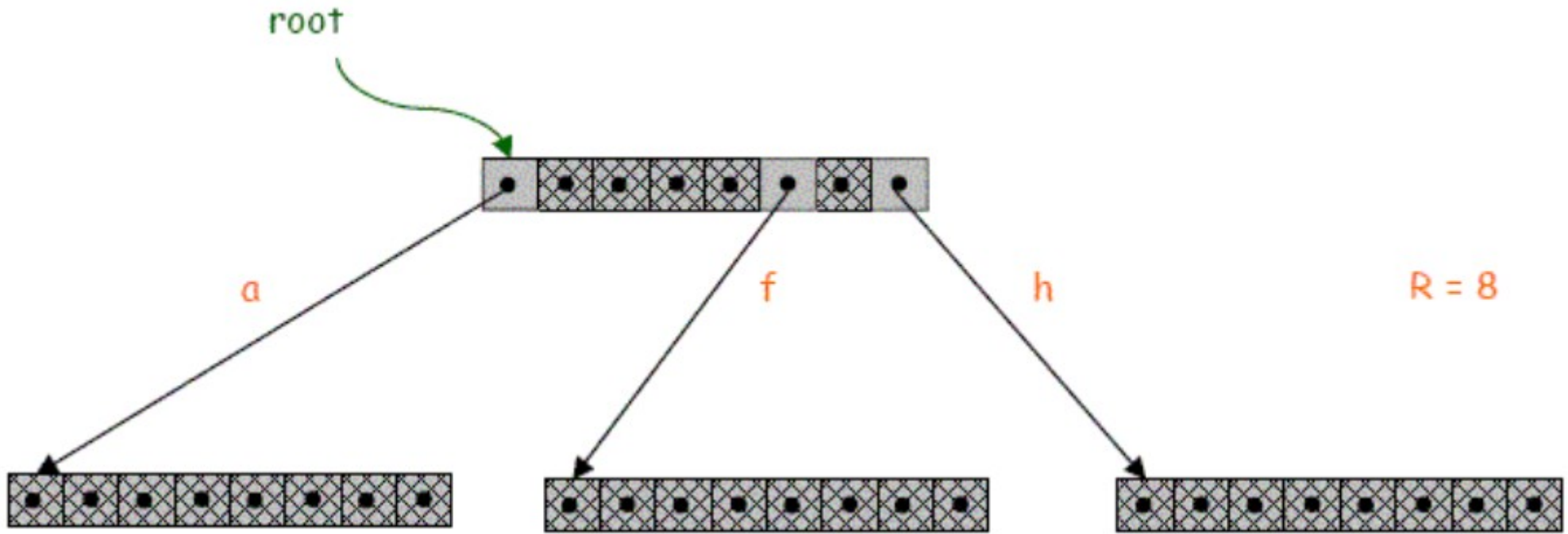
- Cada nó aloca espaço para todo o alfabeto.
- Há desperdício de espaço.
- Descendentes diretos de um pai ficam num só.

R-Way Trie



- Cada nó aloca espaço para todo o alfabeto.
- Há desperdício de espaço.
- Descendentes diretos de um pai ficam num só.

R-Way Trie



- Cada nó aloca espaço para todo o alfabeto.
- Há desperdício de espaço.
- Descendentes diretos de um pai ficam num só.

TST – *Ternary Search Tree*

TST – *Ternary Search Tree*

- Cada nó aloca três ponteiros:

TST – *Ternary Search Tree*

- Cada nó aloca três ponteiros:
 - Centro: caractere seguinte.

TST – *Ternary Search Tree*

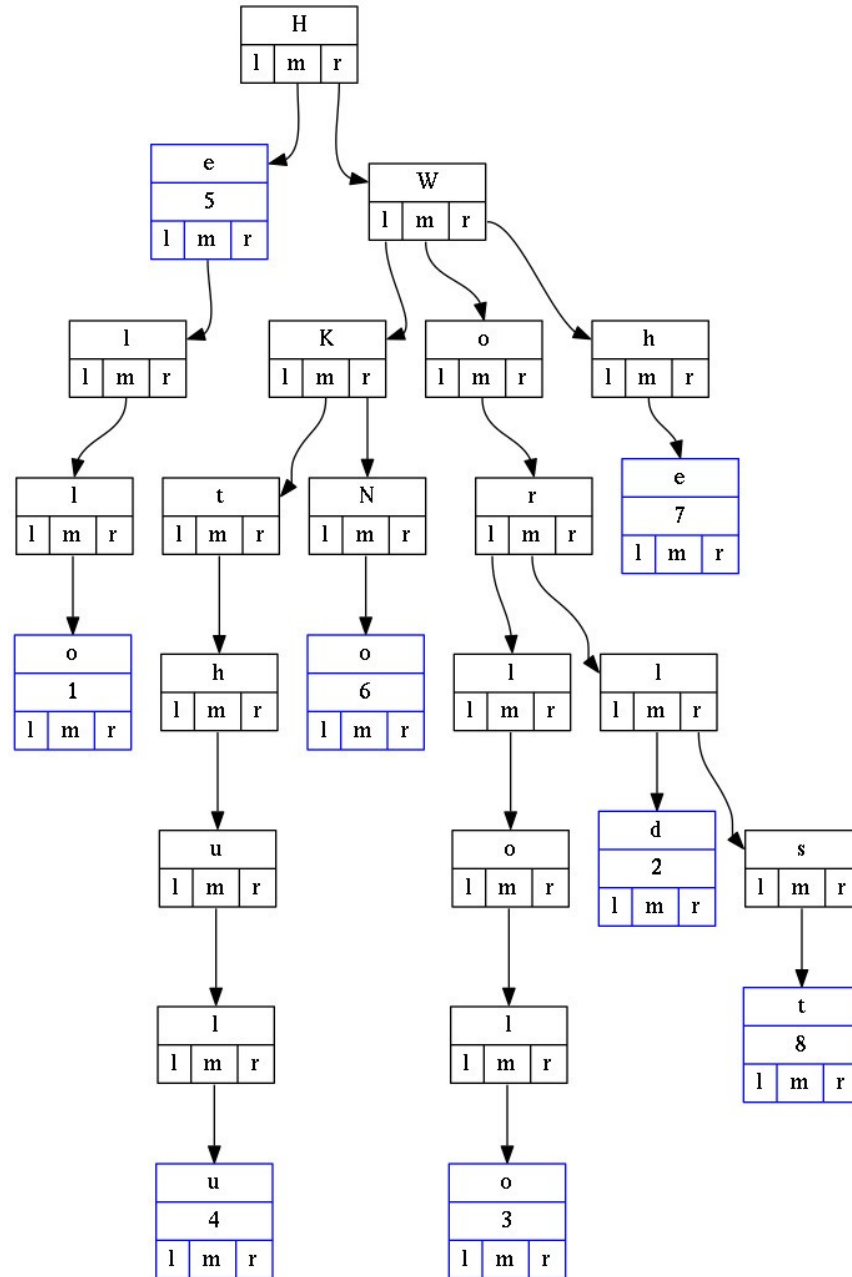
- Cada nó aloca três ponteiros:
 - Centro: caractere seguinte.
 - Filhos da esquerda e direita: caracteres alternativos.

TST – *Ternary Search Tree*

- Cada nó aloca três ponteiros:
 - Centro: caractere seguinte.
 - Filhos da esquerda e direita: caracteres alternativos.
 - Tem desempenho melhor no que se refere ao espaço.

TST – Ternary Search Tree

- Exemplo



Árvore Patrícia

Árvore Patrícia

- É comumente conhecida como **RADIX**.

Árvore Patrícia

- É comumente conhecida como **RADIX**.
- É uma representação compacta de uma Trie, onde os nós que teriam apenas um filho são agrupados nos seus antecessores.

Árvore Patrícia

- É comumente conhecida como **RADIX**.
- É uma representação compacta de uma Trie, onde os nós que teriam apenas um filho são agrupados nos seus antecessores.
- É comum que muitos nós das tries tenham apenas um nó descendente.

Árvore Patrícia

- É comumente conhecida como **RADIX**.
- É uma representação compacta de uma Trie, onde os nós que teriam apenas um filho são agrupados nos seus antecessores.
- É comum que muitos nós das tries tenham apenas um nó descendente.
- Isto faz com que as tries tenham um custo elevado de espaço.

Construção de uma árvore Patrícia

Construção de uma árvore Patrícia

- Considere que queremos construir uma árvore Patrícia com os seguintes elementos:

Construção de uma árvore Patrícia

- Considere que queremos construir uma árvore Patrícia com os seguintes elementos:
 - PA

Construção de uma árvore Patrícia

- Considere que queremos construir uma árvore Patrícia com os seguintes elementos:
 - PA
 - PATA

Construção de uma árvore Patrícia

- Considere que queremos construir uma árvore Patrícia com os seguintes elementos:
 - PA
 - PATA
 - PASTA

Construção de uma árvore Patrícia

- Considere que queremos construir uma árvore Patrícia com os seguintes elementos:
 - PA
 - PATA
 - PASTA
 - PASSA

Construção de uma árvore Patrícia

- Considere que queremos construir uma árvore Patrícia com os seguintes elementos:
 - PA
 - PATA
 - PASTA
 - PASSA
 - PAPA

Construção de uma árvore Patrícia

- PA



Construção de uma árvore Patrícia

Construção de uma árvore Patrícia

- PATA

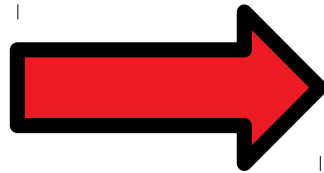
Construção de uma árvore Patrícia

- PATA



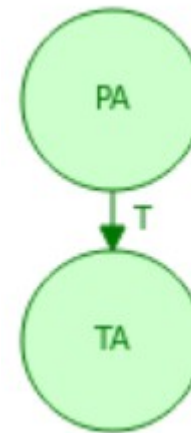
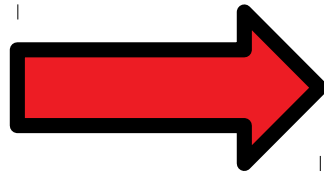
Construção de uma árvore Patrícia

- PATA



Construção de uma árvore Patrícia

- PATA



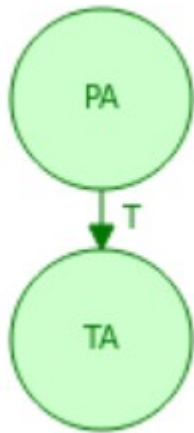
Construção de uma árvore Patrícia

Construção de uma árvore Patrícia

- PASTA

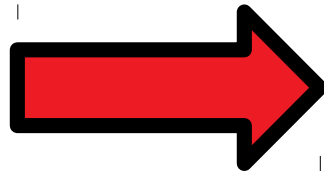
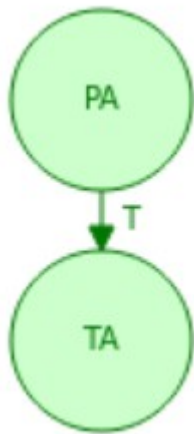
Construção de uma árvore Patrícia

- PASTA



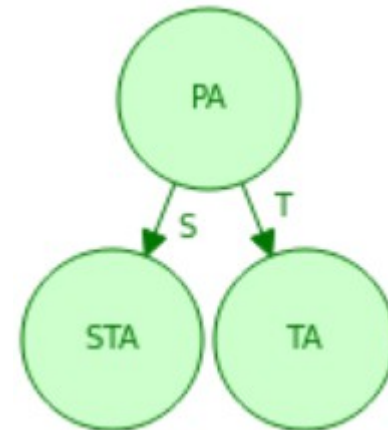
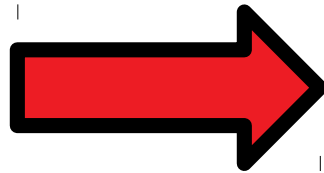
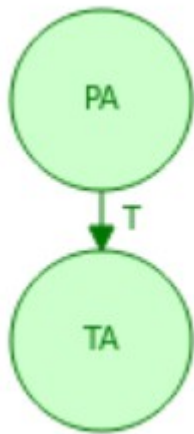
Construção de uma árvore Patrícia

- PASTA



Construção de uma árvore Patrícia

- PASTA



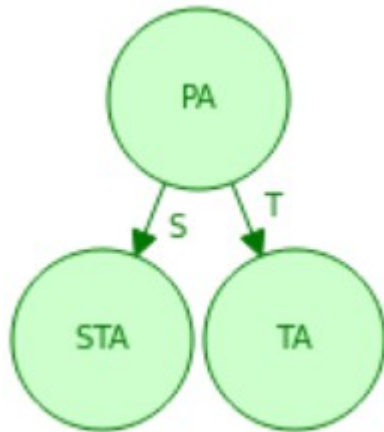
Construção de uma árvore Patrícia

Construção de uma árvore Patrícia

- PASSA

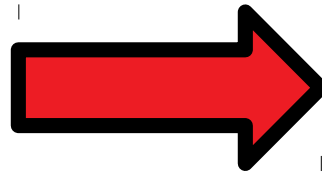
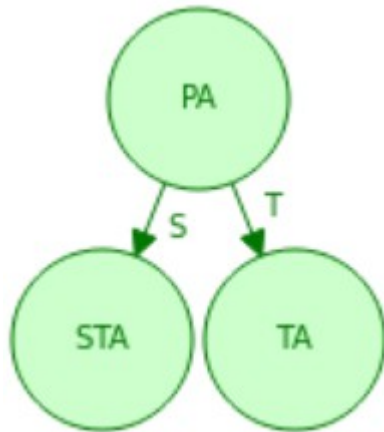
Construção de uma árvore Patrícia

- PASSA



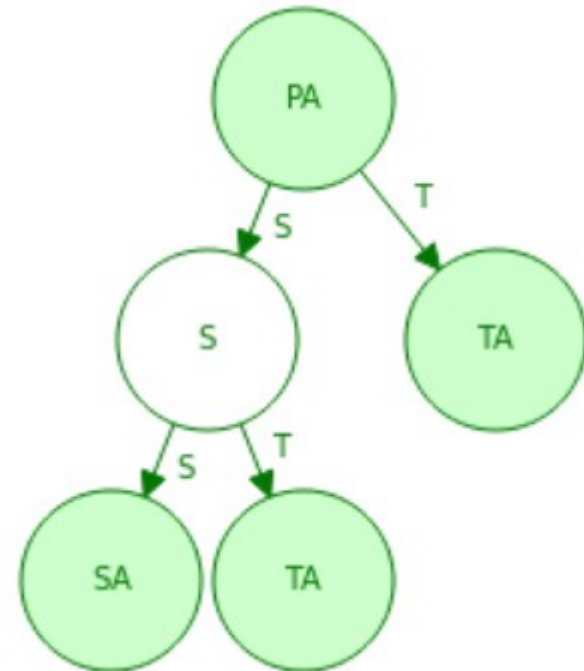
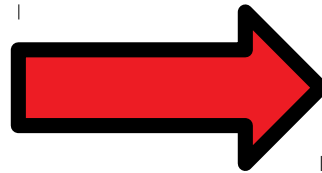
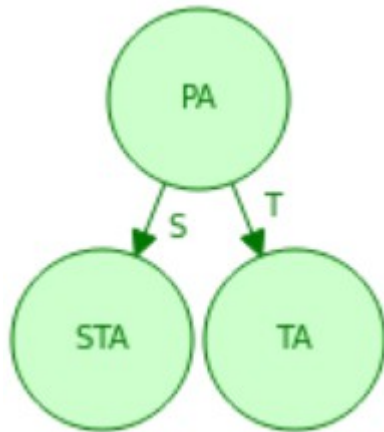
Construção de uma árvore Patrícia

- PASSA



Construção de uma árvore Patrícia

- PASSA



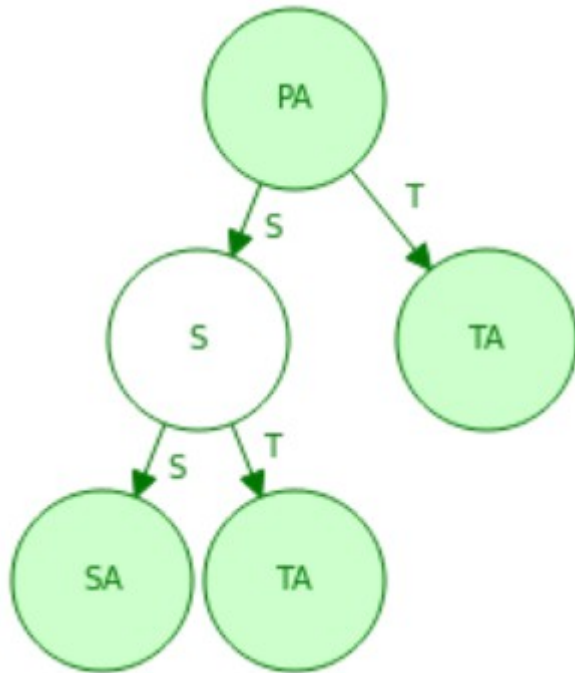
Construção de uma árvore Patrícia

Construção de uma árvore Patrícia

- PAPA

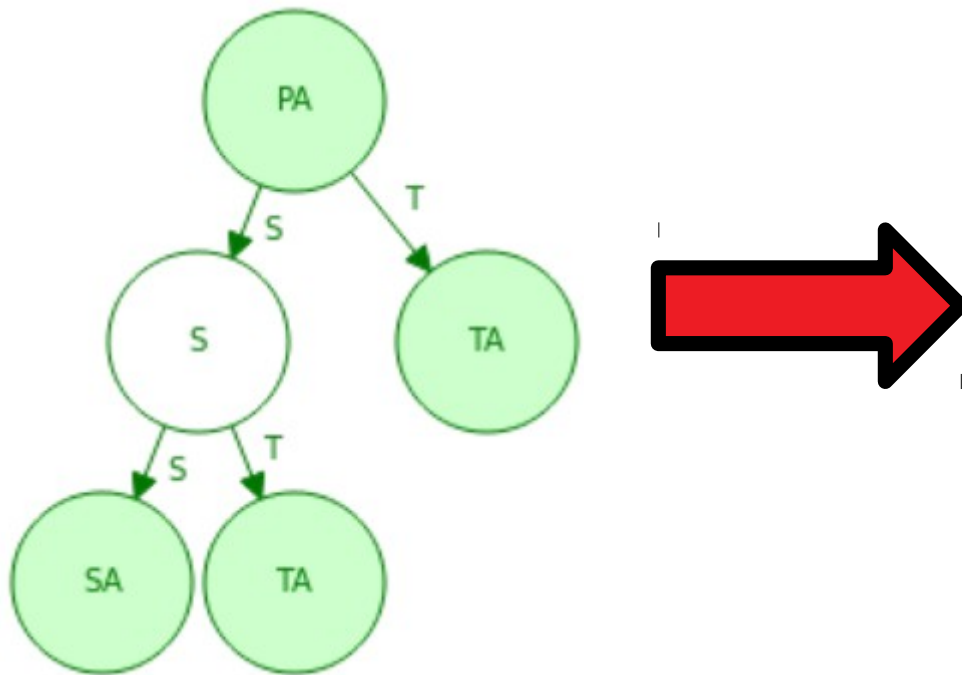
Construção de uma árvore Patrícia

- PAPA



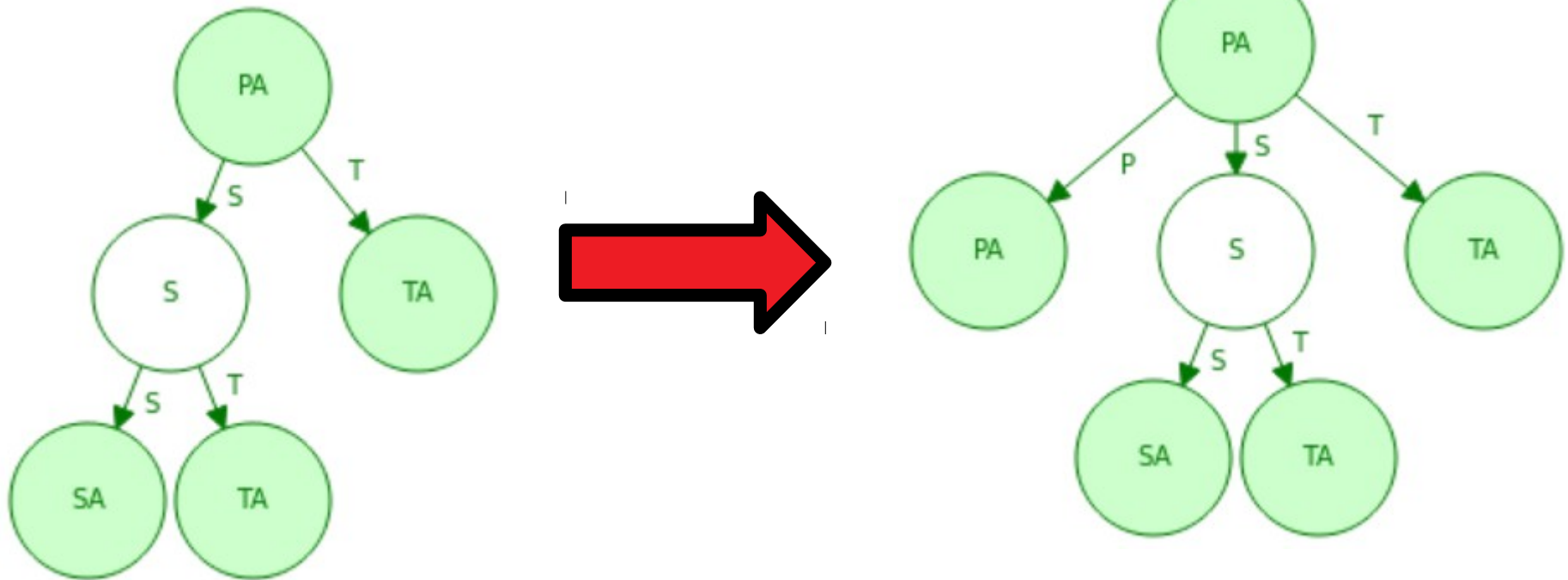
Construção de uma árvore Patrícia

- PAPA



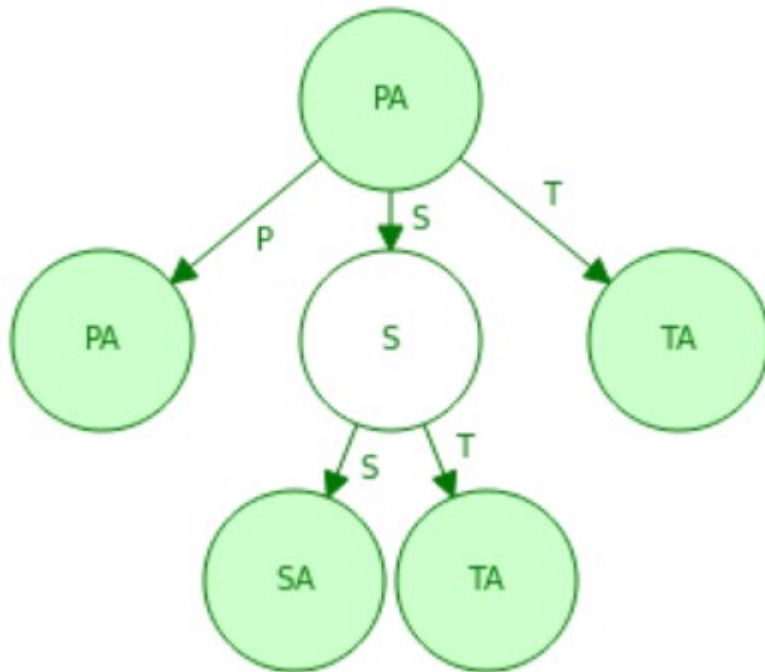
Construção de uma árvore Patrícia

- PAPA



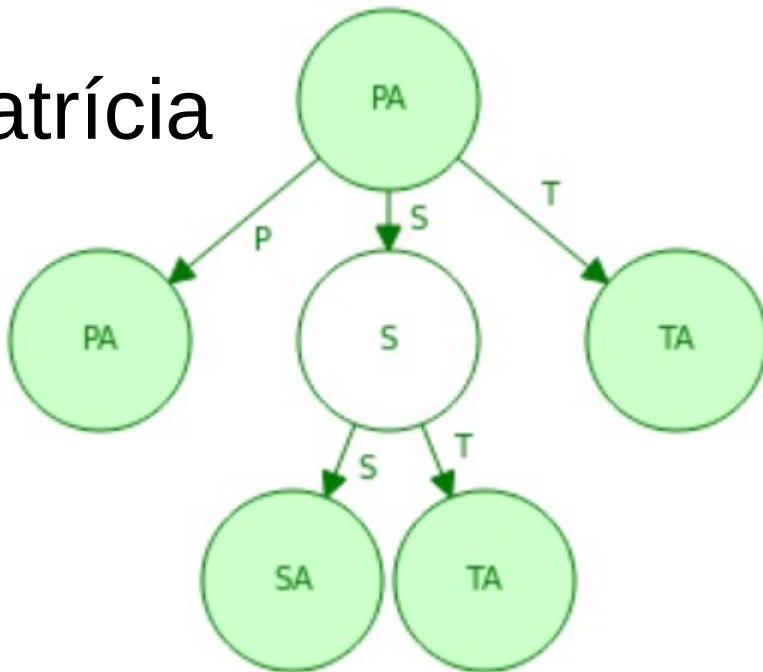
Trie versus Patrícia

Trie versus Patrícia



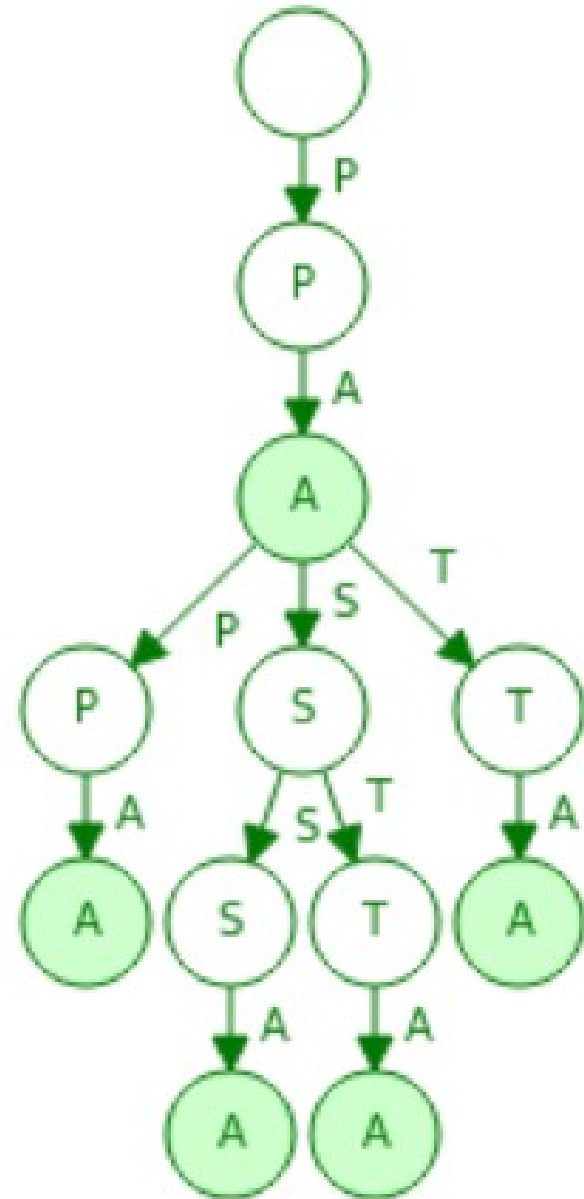
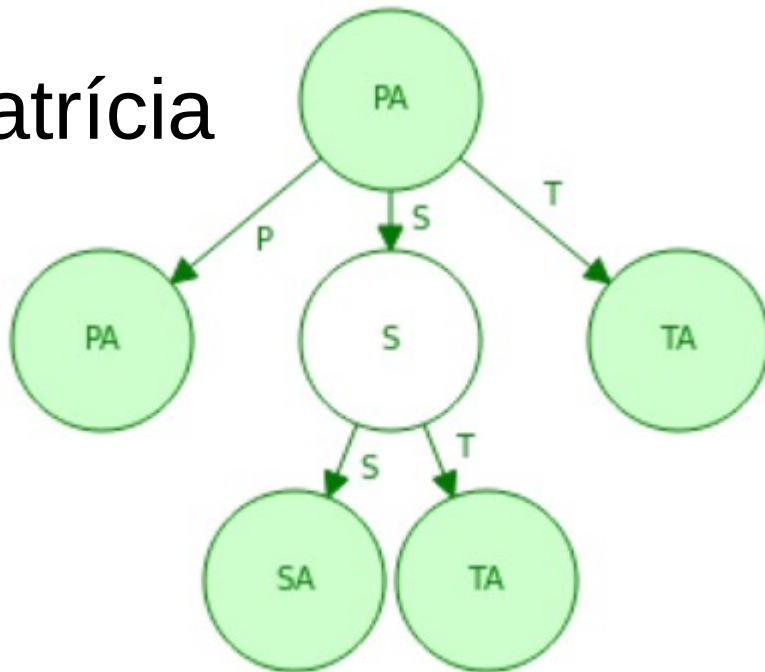
Trie versus Patrícia

Patrícia



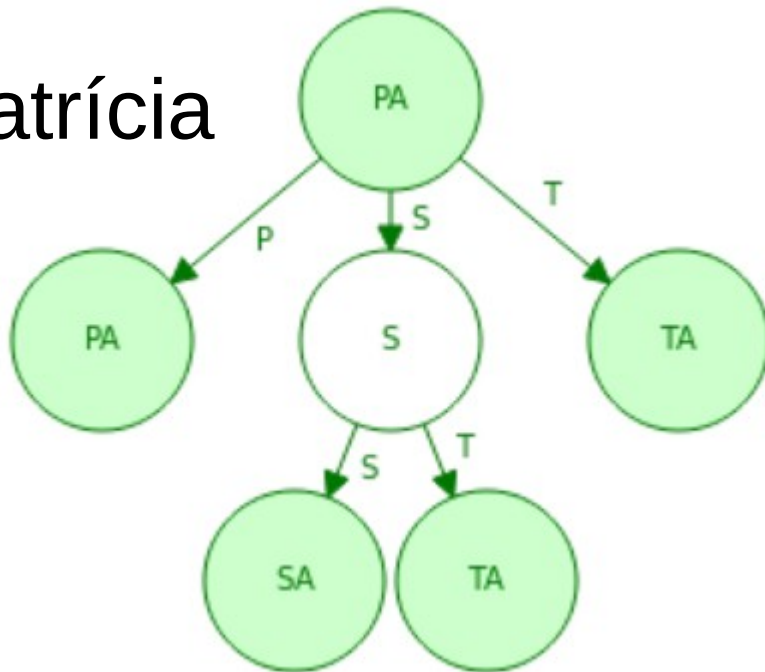
Trie versus Patrícia

Patrícia

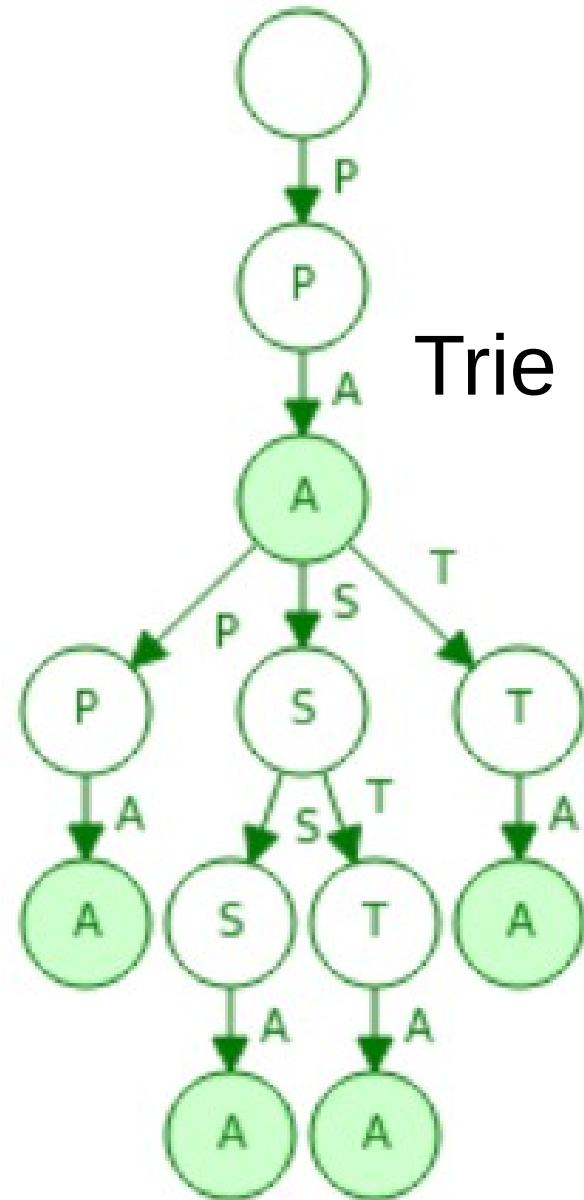


Trie versus Patrícia

Patrícia



Trie



Árvore Patrícia

Árvore Patrícia

- Proposta por Donald Morrison em 1968.

Árvore Patrícia

- Proposta por Donald Morrison em 1968.
- O nome é um acrônimo de **P**ractical **A**lgorithm **T**o **R**etrieve **I**nformation **C**oded in **A**lphanumeric.

Árvore Patrícia

- Proposta por Donald Morrison em 1968.
- O nome é um acrônimo de **P**ractical **A**lgorithm **T**o **R**etrieve **I**nformation **C**oded in **A**lphanumeric.
- Esse método é particularmente útil para tratamento de chaves de tamanho variável extremamente longas, tais como títulos e frases.

Aplicações das Tries

Aplicações das Tries

Aplicações das Tries

- Dicionários

Aplicações das Tries

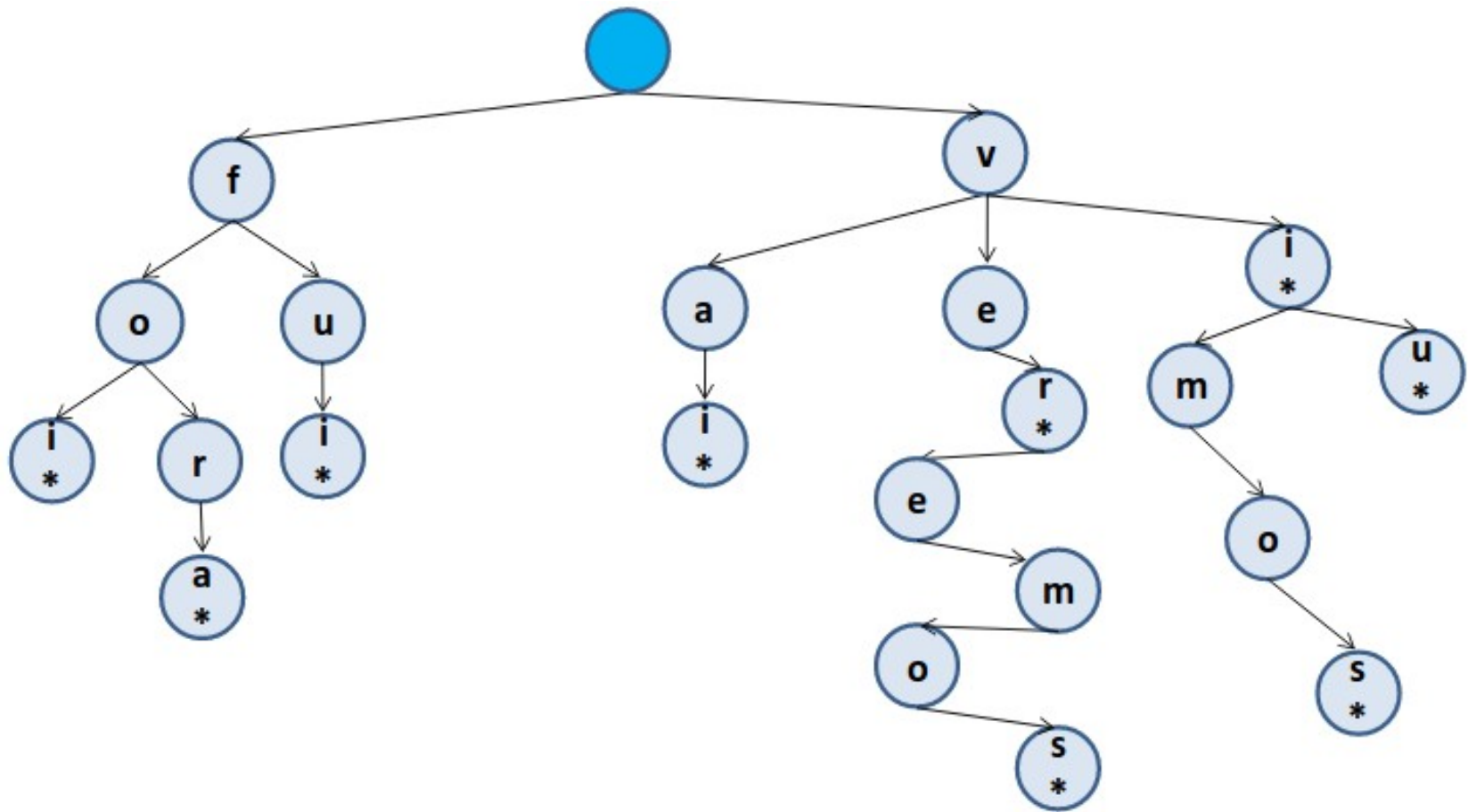
- Dicionários
- Corretores ortográficos

Aplicações das Tries

- Dicionários
- Corretores ortográficos
- Programas para Processamento de Linguagem Natural (NLP – *Natural Language Processing*).

Exercícios

- Quais chaves/palavras estão representadas nesta Trie?



Exercícios

- Insira as chaves abaixo em uma árvore Patricia:
 - A 000100
 - B 010100
 - C 000010
 - D 100100
 - E 001001
 - F 001100
 - G 101000
 - H 101010

Exercícios

- Crie uma árvore Trie com o seguinte texto:
 - “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.”
- Remova as seguintes chaves abaixo e escreva o resultado final da árvore:
 - rei, rainha, roeu, rato, então, furioso.