

Tipos Especiais de Tries: Árvore Patricia

Mathaus C. Huber

¹Universidade Federal de Pelotas (UFPEL) – Discente do Curso Superior de Ciência da Computação
R. Gomes Carneiro, 1 - Centro – 96075-630 – Pelotas – RS – Brazil

mchuber@inf.ufpel.edu.br

Abstract. *This article describes the research project developed in the discipline of Algorithms and Data Structure II, given to the fourth semester of the Computer Science course, at the Federal University of Pelotas, which refers to the analysis of a special type of trie, in this case specifically, the Patricia Tree was chosen, an algorithm to perform searches in trees with the nodes keys represented in binary. Aiming to elucidate its main characteristics, applications and a brief account of its history.*

Resumo. *Este artigo descreve o projeto de pesquisa desenvolvida na disciplina de Algoritmos e Estrutura de Dados II, conferida ao quarto semestre do curso de Ciência da Computação, da Universidade Federal de Pelotas, no qual se refere a análise de um tipo especial de trie, no caso em específico, foi escolhido a Árvore Patricia, um algoritmo para realização de buscas em árvores com as chaves dos nós representadas em binário. Almejando elucidar as suas principais características, aplicações e, um breve relato de sua história.*

1. Introdução

O significado do seu acrônimo é autoexplicativo, onde PATRICIA significa Practical Algorithm To Retrieve Information Coded In Alphanumeric, ou melhor traduzido, Algoritmo Prático para Recuperar Informações Codificadas em Alfanumérico. Esse método é particularmente útil para tratamento de chaves de tamanho variável extremamente longas, tais como títulos e frases.

O algoritmo para construção da árvore Patricia é baseado no método de pesquisa digital, mas sem apresentar o inconveniente das tries, ou seja, ela é construída a partir da árvore binária de prefixo. O problema de caminhos de uma só direção é eliminado por meio de uma solução simples e elegante: cada nó interno da árvore contém o índice do caractere a ser testado para decidir qual subárvore seguir

A árvore Patricia é considerada um tipo especial de Trie pois, é uma representação compacta de uma Trie, onde os nós que teriam apenas um filho são agrupados nos seus antecessores. É comum também que muitos nós das tries tenham apenas um nó descendente, Isto faz com que elas tenham um custo elevado de espaço.

1.1. Tries

Antes de abordarmos a fundo o conceito e as características do algoritmo da Árvore Patricia, considerada um tipo especial de Trie, é necessário elucidar os conceitos

da estrutura de dados, ou árvore de prefixos, Trie. Como já visto anteriormente na disciplina, Trie vem de Retrieval, que significa recuperação e, trata-se de um tipo de árvore de busca, na qual a ideia geral é usar parte das chaves como caminhos de busca. Dependendo da aplicação, a Árvore trie pode substituir uma tabela hash, e tem algumas vantagens em relação a Árvores de busca binária, algumas delas são:

- A busca é mais rápida;
- Uma árvore Trie requer menos espaço quando contém um grande número de cadeias curtas, porque as chaves não são armazenadas de forma explícita e os nós das chaves iniciais comuns são compartilhados.

2. Características

2.1. Inserção

Inserir uma string em uma árvore Patricia é similar a pesquisar por essa string até o ponto onde a busca é encerrada, pois a string não é encontrada na árvore. Se a busca é encerrada em uma aresta, um novo nó é criado nessa aresta. Esse nó armazena a posição do caractere que distingue a chave destino daquela aresta e a chave que se deseja inserir, e tem como filhos o nó que estava na extremidade seguinte da aresta e um novo nó com a parte restante da nova chave. Se a busca for encerrada em um nó, então um nó filho é criado e o restante da nova chave é usado como rótulo para aresta entre os dois.

Etapas:

1. Se a subárvore atual for vazia, é criado um nó de informação com a chave X (isto ocorre somente na inserção da primeira chave) e o algoritmo termina.
2. Se a subárvore atual for simplesmente um nó de informação, os bits da chave X são comparados, a partir do bit de índice imediatamente após o último índice da sequência de índices consecutivos do caminho de pesquisa, com os bits correspondentes da chave X deste nó de informação, até encontrar um índice *i* cujos bits sejam diferentes
3. Se a raiz da subárvore atual for um nó de desvio, deve-se prosseguir para a subárvore indicada pelo bit da chave X de índice dado pelo nó atual, de forma recursiva
4. Criar um nó de desvio e um nó de informação: o primeiro contendo o índice *i* e o segundo a chave X. A seguir, o nó de desvio é ligado ao de informação pelo ponteiro de subárvore esquerda ou direita, dependendo se o bit de índice *i* da chave X seja 0 ou 1, respectivamente
5. O caminho de inserção é percorrido novamente de baixo para cima, subindo com o par de nós criados no passo 4 até chegar a um nó de desvio cujo índice seja menor que o índice *i* determinado no passo 2: este é o ponto de inserção e o par de nós é inserido

2.2. Busca

A busca por uma string em uma árvore Patricia é similar a busca em uma Trie, com a diferença de que ao chegar em um nó, é comparado apenas um caractere, contra a comparação de substrings inteiras que acontece na Trie.

Etapas:

1. Comparar o caractere na posição x com o caractere y.
2. Se menor ou igual: Segue ramo a esquerda;
3. Se maior: Segue ramo a direita
4. Repete isso até chegar numa folha

2.3. Remoção

Remover uma string de uma árvore Patricia é o oposto da operação de inserção. Primeiro, localiza-se a folha correspondente a string e remove-se ela da árvore. Como o pai terá apenas um filho, os nós pai e irmão do nó removido são agrupados em um único nó.

Etapas:

1. Primeiro busca-se e apaga-se a palavra da árvore
2. Soma-se o valor do campo avançar do nó pai a todos os nós filhos

3. Aplicações

A implementação do algoritmo da Árvore Patricia tem várias aplicações, especialmente no ramo de banco de dados, uma exemplificação seria o ScaleDB, que utiliza Patricia-Tries, e é uma estrutura baseada na árvore Patricia, porém otimizado para o acesso em disco. É muito utilizado também em Busca em redes P2P e em Ganho de eficiência com documentos XML, onde, a estrutura de indexação usada em XML pode ser substituída por uma estrutura do tipo Patricia para ganho de performance em tempo de execução das operações.

4. Conclusão

O objetivo de se utilizar esse tipo especial de Trie é, justamente, contornar as desvantagens e tornar os métodos de inserção, remoção e busca ainda mais eficientes. As Árvores Patricia buscam tornar mais compactas as Tries e com isso ganham em velocidade, por outro lado, não possuem escalabilidade adequada quando se têm muitas chaves distintas para um mesmo caractere.

5. Referências

EDELKAMP, S. "Patricia tree", in **Dictionary of Algorithms and Data Structures**, Vreda Pieterse and Paul E. Black, 2010.

MORRISON, D. R. PATRICIA - **Practical Algorithm to Retrieve Information Coded in Alphanumeric**. Journal of the Association of Computing Machinery. 1968

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

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

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