Trabalho de implementação II

Gilson Trombetta Magro

Generated by Doxygen 1.8.13

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Namespace Documentation

3.1 functions Namespace Reference

Namespace que engloba as funções do trabalho.

Functions

- bool validateWord (string word)

  Função que valida uma palavra encontrada.
- Trie * buildTrie (string filename)

  Função que instancia e constroi uma árvore Trie a partir de um arquivo dicionário.

3.1.1 Detailed Description

Namespace que engloba as funções do trabalho.

O namespace functions engloba as funções buildTrie e validateWord, utilizadas no trabalho.

3.1.2 Function Documentation

3.1.2.1 buildTrie()

```
Trie* functions::buildTrie (
            string filename )
```

Função que instancia e constroi uma árvore Trie a partir de um arquivo dicionário.

Parameters

filename	uma string que representa o caminho para o arquivo dicionário.

Returns

um ponteiro para uma árvore Trie, construída a partir do dicionário especificado.

Essa função percorre o arquivo dicionário, buscando encontrar palavras-chave entre colchetes, seguidas de suas definições, como em um dicionário. Ao encontrar uma palavra, ela é validada pela função validateWord, e caso seja válida, sua posição e o tamanho da sua linha são inseridos na Trie.

3.1.2.2 validateWord()

```
bool functions::validateWord (
            string word )
```

Função que valida uma palavra encontrada.

Parameters

word	uma string para ser verificada.

Returns

um boolean: true caso a palavra seja válida, false caso contrário.

Para o caso de dicionários que não seguem o padrão de que palavras-chave, entre colchetes, devem conter somente caracteres de 'a' a 'z' (97 a 122 em ASCII), se faz necessário validar que as palavras encontradas pela função buildTrie realmente cumpram essa especificação.

3.2 structures Namespace Reference

Namespace que engloba as estruturas do trabalho.

Classes

- struct Data

  Struct que encapsula o retorno do método find da classe Trie.
- class Trie

  Classe que implementa a árvore de prefixos "retrieval".

3.2.1 Detailed Description

Namespace que engloba as estruturas do trabalho.

O namespace structures engloba a struct Data e a classe Trie.

Chapter 4

Class Documentation

4.1 structures::Data Struct Reference

Struct que encapsula o retorno do método find da classe Trie.

Public Member Functions

- **Data** (unsigned long position_, unsigned long length_, bool found_)

Public Attributes

- unsigned long **position**
- unsigned long **length**
- bool **found**

4.1.1 Detailed Description

Struct que encapsula o retorno do método find da classe Trie.

Contém os campos position, length e found. O atributo position representa a posição da palavra no dicionário. Length representa o comprimento da linha em que a palavra se encontra, e found é um valor booleano que determina se a palavra foi encontrada na árvore ou não.

The documentation for this struct was generated from the following file:

- trie.cpp

4.2 structures::Trie Class Reference

Classe que implementa a árvore de prefixos "retrieval".

Public Member Functions

- void insert (const char *word, unsigned long position, unsigned long length)

  Método que insere palavras na árvore.
- Data find (const char *word)

  Método que busca palavras na árvore.

4.2.1 Detailed Description

Classe que implementa a árvore de prefixos "retrieval".

A árvore Trie é implementada de forma a cada nó possuir até 26 filhos (um para cada letra do alfabeto). O nó raiz é usado como um nó sentinela, que apenas faz a ligação com os outros nós da árvore, e não representa nenhum caracter. Percorrer a árvore numa certa ordem de filhos (0 a 25) equivale a percorrer as letras ('a' a 'z') de uma palavra. Desse modo, é possível indexar o final de palavras como os nós mais inferiores da árvore. Quando uma palavra é inserida na árvore, um caminho de nós é criado para representar aquela palavra. No último nó desse caminho, são guardadas as informações acerca da posição e do tamanho da linha da palavra no dicionário.

4.2.2 Member Function Documentation

4.2.2.1 find()

```
Data structures::Trie::find (
            const char * word )  [inline]
```

Método que busca palavras na árvore.

Checa se o nó raiz existe, e se existir, chama o método find do nó raiz. Cada nó, por sua vez, chama o método find em um de seus filhos, reduzindo a palavra a cada chamada. Se a palavra chegar ao tamanho zero, o último nó do caminho foi encontrado, e portanto a palavra foi encontrada na árvore. Nesse caso, retorna-se a posição e o comprimento armazenados naquele nó. Se não houverem filhos suficientes para continuar no caminho especificado pela palavra, então a palavra não está presente na árvore.

4.2.2.2 insert()

```
void structures::Trie::insert (
            const char * word,
            unsigned long position,
            unsigned long length )  [inline]
```

Método que insere palavras na árvore.

Parameters

word	Uma sequência de caracteres que representa a palavra a ser inserida.
position	A posição da palavra no dicionário.
length	O tamanho da linha em que a palavra se encontra.

Verifica se a palavra não é vazia. Se for, lança uma exceção; caso contrário, chama o método insert do nó raiz. O método find do nó navega pela árvore, criando filhos quando necessário, para que ao final se tenha um caminho de nós que representa a palavra que foi inserida. Consiste em diminuir a palavra em um caracter a cada chamada, e passar a palavra reduzida para um filho, que executa o mesmo processo até a palavra ter tamanho zero. Quando não há mais caracteres na palavra, significa que aquele é o último nó do caminho, e as informações de posição e comprimento devem ser armazenadas ali.

The documentation for this class was generated from the following file:

- trie.cpp

Index