



Relatório de ED2

Autor: Eduardo Figueredo Pacheco

Data
20/06/2023

Sumário

1	Introdução	2
2	Desenvolvimento	3
2.1	Testes	5
3	Conclusão	7

1 Introdução

O código em questão implementa um reconhecedor de padrões baseado em expressões regulares utilizando um grafo direcionado. A expressão regular fornecida pelo usuário é convertida em um grafo, onde cada caractere da expressão é representado por um vértice. O reconhecimento de padrões é realizado por meio de uma busca em profundidade no grafo, verificando se uma determinada palavra é reconhecida pela expressão regular. O programa solicita ao usuário a entrada de uma expressão regular e um número de testes. Em seguida, para cada teste, o usuário deve fornecer uma palavra para verificar se é reconhecida pela expressão regular. O resultado da verificação é exibido como 'S' (sim) ou 'N' (não).

O relatório a seguir apresentará uma análise mais detalhada das estruturas e algoritmos utilizados, destacando a implementação do reconhecedor de padrões e sua aplicação prática. Além disso, serão discutidos aspectos de desempenho e possíveis melhorias a serem consideradas.

2 Desenvolvimento

No código, a classe Graph é responsável pela criação e manipulação do grafo. Ela armazena a expressão regular em um vetor de caracteres, mantém uma lista de adjacência para representar as arestas do grafo e um vetor de vetores de caracteres para armazenar os caracteres entre [] presentes na expressão regular. O método setRegex é utilizado para definir a expressão regular no grafo, e o método getRegex retorna a expressão regular do grafo. A função reconhece realiza a verificação do reconhecimento da palavra pela expressão regular, utilizando a busca em profundidade no grafo. A função criarGrafoExpressaoRegular constrói o grafo a partir da expressão regular fornecida, adicionando as arestas correspondentes aos caracteres entre [] e as demais arestas conforme as regras da expressão regular.

```
1 class Graph {
2 private:
3     char* re; // Vetor de caracteres para
        armazenar a express o regular
4
5 public:
6     vector<vector<int>> adj;
7     vector<vector<char>> letras;
8     int L;
9     int V;
10
11     Graph(int V) {
12         this->V = V;
13         adj.resize(V);
14         letras.resize(V);
15         L = 0;
16     }
17
18     void addEdge(int u, int v) {
19         adj[u].push_back(v);
20     }
21
22     void dfsR(int v, bool* visited) {
23         visited[v] = true;
24         cout << v << "orange orange";
25
26         for (int i = 0; i < adj[v].size();
```

```

27         i++) {
28             int u = adj[v][i];
29             blueif (!visited[u]) {
30                 dfsR(u, visited);
31             }
32         }
33
34     void setRegex(string regexp) {
35         int m = regexp.length();
36         re = new char[m + 1];
37         strcpy(re, regexp.c_str());
38     }
39
40     char* getRegex() {
41         bluereturn re;
42     }
43 };

```

2.1 Testes

```
1 Digite uma express o regular: (A|(B|C))*  
2 Digite o n mero de testes: 3  
3 Digite uma palavra para verificar: AAAAAA  
4 S  
5 Digite uma palavra para verificar: ABCCC  
6 S  
7 Digite uma palavra para verificar: ABCD  
8 N
```

O teste apresentado tem como objetivo verificar o reconhecimento de palavras em relação a uma expressão regular específica. A expressão regular fornecida é $(A|(B|C))^*$, o que significa que a palavra deve conter sequências de A, B ou C de tamanho arbitrário (incluindo o tamanho zero), repetidas qualquer número de vezes.

No teste, são realizadas três verificações com palavras diferentes: "AAAAA", "ABC" e "ABCD". A primeira palavra, "AAAAA", atende aos critérios da expressão regular, pois contém somente a letra "A" repetida várias vezes. Durante a execução, é possível observar que o vetor atingidos é atualizado, indicando quais estados do grafo foram alcançados. No final, o programa exibe o resultado "S", indicando que a palavra é reconhecida pela expressão regular.

A segunda palavra, "ABC", também satisfaz a expressão regular, pois possui as letras "A", "B" e "C" nessa ordem. Durante o processo de reconhecimento, o vetor atingidos é atualizado de acordo com os estados alcançados no grafo. No final, o resultado exibido é "S", confirmando que a palavra é reconhecida pela expressão regular.

Por fim, a terceira palavra, "ABCD", não atende à expressão regular, pois contém a letra "D", que não está presente na expressão regular definida. Durante a execução, é possível observar que o vetor atingidos não é atualizado para alguns estados, indicando que a palavra não alcançou o estado final do grafo. O resultado exibido é "N", indicando que a palavra não é reconhecida pela expressão regular.

Esses testes demonstram o funcionamento básico do programa ao reconhecer palavras com base em uma expressão regular pré-definida, destacando a importância de escolher a expressão regular adequada para a correspondência desejada.

```
1 Digite uma express o regular: [AEIOU][BC]
2 Digite o n mero de testes: 5
3 Digite uma palavra para verificar: AB
4 S
5 Digite uma palavra para verificar: AD
6 N
7 Digite uma palavra para verificar: AL
8 N
```

O teste apresentado tem como objetivo verificar o reconhecimento de palavras com base em uma expressão regular específica. A expressão regular fornecida é [AEIOU][BC], o que significa que a palavra deve começar com uma vogal (A, E, I, O ou U) seguida de uma consoante (B ou C).

Durante a execução do teste, são realizadas cinco verificações de palavras diferentes. A primeira palavra testada é "AB". Essa palavra atende aos critérios da expressão regular, pois começa com a vogal "A" e é seguida pela consoante "B". Ao final da execução, o programa exibe o resultado "S", indicando que a palavra é reconhecida pela expressão regular.

A segunda palavra testada é "AD". Nesse caso, a palavra começa com a vogal "A", mas é seguida pela consoante "D", que não está definida na expressão regular. Como a palavra não atende aos critérios estabelecidos, o programa exibe o resultado "N", indicando que a palavra não é reconhecida pela expressão regular.

A terceira palavra testada é "AL". Novamente, a palavra começa com a vogal "A", mas é seguida pela consoante "L", que não está definida na expressão regular. Assim como no caso anterior, o resultado exibido é "N", indicando que a palavra não é reconhecida pela expressão regular.

Esses testes demonstram como a expressão regular [AEIOU][BC] é aplicada para verificar se palavras atendem aos critérios estabelecidos. A correta correspondência entre vogais e consoantes determina se a palavra é reconhecida ou não pela expressão regular.

3 Conclusão

O código implementado apresenta um reconhecedor de padrões funcional utilizando expressões regulares e grafos direcionados. Através do uso adequado das estruturas de dados, como o grafo e os vetores de caracteres, foi possível realizar a verificação de correspondência entre palavras e expressões regulares de forma eficiente. O programa suporta múltiplos testes e exibe o resultado do reconhecimento para cada palavra em relação à expressão regular fornecida.

No entanto, é importante ressaltar que o código possui algumas limitações. Expressões regulares mais complexas, que envolvam caracteres especiais ou recursos avançados, não são tratadas pelo programa. Além disso, em casos de expressões regulares ou palavras muito longas, pode haver um aumento no consumo de memória e tempo de execução. Considerando esses aspectos, é recomendado avaliar o contexto de uso e a complexidade das expressões regulares antes de aplicar o código.