

Trabalho Prático 1

Plano de Campanha

Vitor Emanuel Ferreira Vital - 2021032072

Departamento de Ciência da Computação - Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte - MG - Brazil

vitorvital@dcc.ufmg.br

1. Introdução

Esta documentação tem como objetivo apresentar a implementação e modelagem da resolução do Trabalho Prático 1 da disciplina de Algoritmos 1. Em tal prática, os principais temas explorados, solucionados na linguagem C++, são:

- Modelagem de problemas reais por meio de grafos
- Utilização de estruturas de dados estudados na disciplina
- Utilização do 2-SAT

O problema a ser tratado lida com uma campanha política, no qual, dado um conjunto de propostas, o deputado gostaria de saber de seus seguidores quais dessas propostas eles desejam manter e quais desejam remover. Com o objetivo de tentar agradar todos os seguidores, visto que a quantidade inviabiliza atender as necessidades de todos de forma completa, o deputado deve remover pelo menos uma das duas propostas que seus seguidores não querem e deve continuar com pelo menos uma das duas propostas que cada um deseja manter.

A documentação explicita os métodos e modelagens utilizados para resolução do problema em questão.

2. Modelagem

Conforme explicitado, cada um dos seguidores de determinado deputado deve escolher duas propostas que ele deseja manter e duas propostas que ele deseja remover. Dentre todas as escolhas realizadas, o deputado deve verificar se é possível agradar a todos seus seguidores, ou seja, satisfazer as seguintes questões:

1. Pelo menos uma proposta que o seguidor deseja manter deve continuar no plano de campanha
2. Pelo menos uma proposta que o seguidor deseja remover deve ser retirada do plano de campanha

Com isso em mãos podemos fazer a modelagem seguindo o problema da satisfatibilidade. Nesse caso, dado uma fórmula booleana na forma normal conjuntiva, deseja-se verificar se existe alguma entrada binária que a torne verdadeira.

Para o caso da campanha política em questão utiliza-se a forma 2-SAT, que é uma versão do problema de satisfatibilidade no qual cada cláusula tem exatamente dois literais. A definição da cláusula seguirá o seguinte formato:

- x caso a proposta x seja votada para ser mantida.
- $\neg x$ caso a proposta x seja votada para ser retirada.

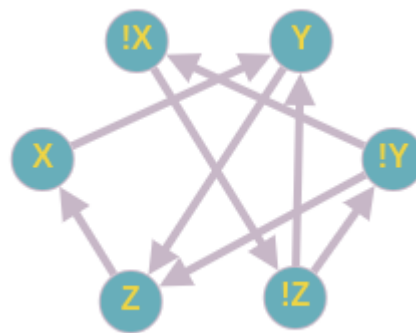
Desta forma, caso o seguidor deseje manter a proposta 2 e 3, removendo a 1 e 4, teríamos as seguintes cláusulas: $(x_2 + x_3)(!x_1 + !x_4)$ - Assim uma proposta removida recebe 0 e uma mantida recebe 1.

Modelado o problema, podemos converter as cláusulas da expressão booleana formada da seguinte forma:

1. Para cada variável x_i , adiciona-se dois nós o x_i e o $!x_i$.
2. Para cada cláusula $C=(x_i+y_i)$ adiciona-se uma aresta de $!x_i$ para y_i e uma outra aresta de $!y_i$ para x_i

Ambas arestas representam cenários no qual as cláusulas são verdadeiras.

Um exemplo de conversão de expressão booleana $S = (!x + y)(!y + z)(x + !z)(z + y)$ para um problema 2SAT gera o grafo abaixo.



Grafo para solucionar o problema 2-SAT

Assim, podemos fazer as seguintes conclusões que tornam uma fórmula booleana não satisfazível:

- Existe um caminho de x para $!x$ no grafo e
- Existe um caminho de $!x$ para x

Porém, para satisfazer o problema para cada variável x do grafo fazemos as seguintes operações:

- Rodamos o algoritmo BFS testando se existe um caminho de x para $!x$
- Rodamos o algoritmo BFS testando se existe um caminho de $!x$ para x

Algoritmo BFS: resolve o problema de encontrar um caminho entre dois nós s e t . A partir de um nó s de partida ele percorre todos os vizinhos e os vizinhos dos vizinhos até encontrar o ponto desejado t .

Se ambas conclusões forem falsas então o grafo possui solução e, portanto, retornando ao problema inicial: existe uma combinação de retirada e manutenção das propostas colocadas a julgamento pelos seus seguidores que satisfazem a condição inicial, podendo-se agradar a todos os seguidores no seu plano de campanha.

Entretanto, ainda a possibilidade de inserção do elemento zero dificulta o processo do 2-SAT, isso porque além de criar uma cláusula com apenas um literal, permite a inserção de dois elementos nulos. Porém para esses dois casos existem soluções que foram utilizadas de forma a retirar seus danos.

1. Caso um elemento zero seja adicionado em conjunto com uma proposta, necessariamente a condição dessa proposta deve ser satisfeita (retirada ou mantida). Para isso, na modelagem do problema todos os vértices apontam para esse novo vértice que deve ser necessariamente satisfeito, isso porque há uma dependência direta das outras cláusulas para com essa.
2. Caso dois elementos zeros sejam adicionados, o seguidor abdicou de realizar sua operação de retirada ou manutenção e, portanto, ele está por padrão satisfeito com as propostas. Logo, não há necessidade de adição dela no gráfico.

Ao fim do problema concluímos que podemos modelar esse problema como um 2-SAT, pois precisamos encontrar uma combinação de mantimento e retirada das propostas, assim obtemos uma modelagem de resolução fácil no qual podemos verificar a validade do problema solucionando-o.