

Documentação

DCC206 Algoritmos 1

Universidade Federal de Minas Gerais 2022/02

Bernardo Reis de Almeida (2021032234)

1. Introdução

Em época de eleições, um deputado federal estabelece um plano de ação que envolve diversas propostas, imbuídas nas mais diversas áreas, a fim de se consolidar perante os votos da maior quantidade possível de eleitores. Entretanto, em meio a tantas ideias, o deputado criou diversas contradições, o que tornou necessário a filtragem e a seleção de suas propostas. Para tal, foi realizada uma enquete em sua rede social, na qual cada eleitor pode selecionar quatro propostas: duas às quais é a favor e duas às quais é contra. Com os resultados em mãos, cabe ao deputado verificar se é possível realizar essa seleção de modo a satisfazer todos os seus eleitores, dadas as preferências de cada um.

O presente trabalho prático lida com a **abstração** e com o **desenvolvimento de modelos** e de **algoritmos** que atuem a pretexto dessa problemática de cunho eleitoral. Em particular, dados o número de propostas, o número de eleitores e suas preferências - fornecidos como dados de entrada -, deve ser desenvolvido um programa que seja capaz de verificar se é possível realizar uma seleção que satisfaça todos os votantes, retornando o resultado como uma resposta binária: ser possível ou não ser possível. A essa finalidade, foi designado o uso da **linguagem C** - o compilador utilizado foi o **GCC** - e da plataforma **Unix**.

Em um primeiro momento, a abstração do problema em um modelo matemático/computacional será abordada na seção **[Modelagem]**, na qual as minuciosidades de definições e as estruturas de dados elaboradas serão detalhadas. Em seguida, na seção **[Execução & Funcionamento]**, o fluxo de trabalho do algoritmo será discutido, relevando a maneira como aquilo definido na seção anterior é efetivamente utilizado na resolução da questão. Por fim, na seção **[Considerações Finais]**, há instruções para se compilar e para se executar o programa, além de referências bibliográficas para todas as fontes de consulta utilizadas.

2. Modelagem

A **campanha eleitoral do deputado**, sendo este o principal elemento de análise para o algoritmo, foi abstraída como um **tipo abstrato de dados** - utilizando o elemento **struct**, definido na linguagem C -, cujas entidades componentes agregam suas principais características, sejam elas: o **número de propostas**, o **número de eleitores** e as **preferências de cada eleitor**.

O número de propostas e o número de eleitores foram representados simplesmente por variáveis dedicadas. As preferências de cada eleitor, por sua vez, foram organizadas em uma **matriz**, da qual as linhas correspondem a cada votante que respondeu à enquete e as colunas, às suas preferências (as duas primeiras colunas são as propostas às quais é a favor e as duas últimas, às quais é contra). Ou seja, é uma matriz de tamanho **[número de eleitores] x [quatro preferências]**. Tal escolha é justificada pelo fato de essa estrutura ser apenas um modo de se organizar as informações que serão recebidas da entrada de dados, além de que, justamente por serem fornecidas, todas as entradas da matriz serão efetivamente utilizadas.

A problemática em si, seja ela **determinar se há alguma seleção de propostas que agrade a todos os eleitores simultaneamente**, foi modelada como um problema **2-SAT** - satisfatibilidade de expressões booleanas envolvendo um produto de somas de duas variáveis. Primeiramente, define-se que um eleitor é satisfeito se ao menos uma das propostas às quais é a favor é mantida, enquanto ao menos uma das propostas às quais é contra é retirada do plano de ação do deputado. Tendo isso em mente, as quatro escolhas de cada votante podem ser

representadas por um produto booleano de dois termos, cada qual composto pela soma de duas variáveis, sendo um correspondente às propostas a favor e o outro, às propostas contra.

Eleitor 1			
<u>Manter</u>		<u>Remover</u>	
1	4	2	3
$(x1 + x4)$		$(\sim x2 + \sim x3)$	

Cada proposta é representada por uma variável x, a qual assume o **valor 1** se a proposta for **mantida** no plano de ações, e assume o **valor 0** caso contrário. Dessa forma, o valor do primeiro termo será 1 (ou seja, ele será satisfeito) se ao menos uma das duas propostas que o compõem for mantida (assumir o valor 1). Enquanto isso, o valor do segundo termo será 1 (novamente, ele será satisfeito) se ao menos uma das duas propostas for removida (o seu complemento assumir o valor 1). Sendo um produto de termos, o valor da expressão final, constituída pelas preferências do eleitor, assumirá o valor 1 somente se ambos os termos assumirem o valor 1, ou seja, se ambos forem satisfeitos, o eleitor também será. A enquete como um todo, por sua vez, pode ser pensada como o produto dos termos relativos a cada eleitor, de modo que será satisfeita somente se todos os eleitores forem.

Eleitor 1	Eleitor 2	(...)	Eleitor n
$(x1 + x4)(\sim x2 + \sim x3)$	$(x5 + x6)(\sim x1 + \sim x2)$		$(x4 + x1)(\sim x7 + \sim x5)$

Enquete modelada como uma expressão booleana

$$(x1 + x4)(\sim x2 + \sim x3)(\dots)(x5 + x6)(\sim x1 + \sim x2)(\dots)(x4 + x1)(\sim x7 + \sim x5)$$

Um detalhe importante de se notar é que cada eleitor também pode escolher não utilizar algum de seus votos (o que é representado com o valor 0, considerando que a numeração das propostas começa do valor 1). No caso, o modelo de suas preferências assumiria a forma:

Eleitor 1			
<u>Manter</u>		<u>Remover</u>	
1	0	2	0
$(x1)$		$(\sim x2)$	

Tal situação foi tratada com uma manipulação algébrica da expressão booleana correspondente, de modo a transformá-la em um produto de termos com duas variáveis. Em particular, observa-se que a expressão $(x1)$ é equivalente à expressão $(x1 + xn)(x1 + \sim xn)$, de modo que as preferências do eleitor enunciado acima possam ser escritas na forma:

Eleitor 1			
<u>Manter</u>		<u>Remover</u>	
1	0	2	0
$(x1 + xn)(x1 + \sim xn)$		$(\sim x2 + xn)(\sim x2 + \sim xn)$	

...sendo xn uma variável qualquer cujo valor não importa (veja que qualquer que seja o valor assumido por xn , a expressão acima ainda é equivalente a $(x1)(x2)$). Em termos de implementação, além das variáveis referentes a cada uma das propostas, uma variável extra foi incluída e utilizada para tratar todas as aparições deste caso.

A questão da **satisfação dessa expressão booleana**, por sua vez, foi modelada por meio de um **grafo**. Em específico, é possível notar que, dado o produto de termos relativo a um eleitor, sua satisfação pode ser representada por implicações lógicas que garantem a manutenção da

integridade da expressão como um todo dada uma designação de valores às variáveis. Por exemplo, seja $(x_1 + x_4)$ o termo relativo às propostas às quais um eleitor qualquer é favorável. Dada uma designação de valores a essas variáveis, se x_1 assumir o valor 0 (ou seja, se $\sim x_1$ for 1), x_4 necessariamente deve ser 1 para que o termo seja satisfeito. O análogo vale para o caso em que x_4 assume o valor 0.

Eleitor 1			
<u>Manter</u>		<u>Remover</u>	
1	4	2	3
$(x_1 + x_4)$		$(\sim x_2 + \sim x_3)$	
$\sim x_1 \rightarrow x_4$		$x_2 \rightarrow \sim x_3$	
$\sim x_4 \rightarrow x_1$		$x_3 \rightarrow \sim x_2$	

A partir desse conceito, um **grafo direcionado não-valorado** é construído de tal modo que, para cada proposta x , há dois vértices que a representam: x e $\sim x$. Entre dois vértices quaisquer há uma aresta dirigida se e somente se a expressão booleana que modela a enquete contém um termo que origine uma implicação lógica entre as variáveis a eles correspondentes, sendo a aresta direcionada no sentido da implicação.

Uma **designação de vértices e arestas**, representando os resultados de uma instância da enquete, **é satisfatível se não há um conjunto de implicações lógicas** - o que se traduz para um caminharmento no grafo - **que origine um absurdo**, isto é, uma expressão lógica que não pode ser verdadeira qualquer que seja a designação de valor às variáveis. Em particular a este problema, tal situação ocorrerá se existirem duas implicações - dois caminhos - nas formas $x \rightarrow (\dots) \rightarrow \sim x$ e $\sim x \rightarrow (\dots) \rightarrow x$ (veja que a única forma de satisfazer a primeira é se x assumir o valor 0, mas isso invalida a segunda, de modo que não haja designação de valores que valide as implicações).

Em termos de implementação, esse grafo foi definido por meio de uma **matriz de tamanho $[2 * (\text{número de propostas} + 1)] \times [2 * (\text{número de propostas} + 1)]$** , tal que a entrada $[i][j]$ assumirá o valor 1 se há uma aresta direcionada de i a j no grafo, assumindo o valor 0 caso contrário. Os primeiros $(\text{número de propostas} + 1)$ vértices são referentes às variáveis x das propostas, enquanto os últimos $(\text{número de propostas} + 1)$ são referentes às respectivas negações, sejam elas as variáveis $\sim x$. Essa decisão é justificada pelo fato de o número máximo de propostas estabelecido pela definição do problema não ser grande o suficiente para a alocação de uma matriz em memória ser um problema relevante, além de que o grafo é dirigido e, portanto, não há redundância simétrica (a entrada $[i][j]$ indica uma aresta no sentido contrário da entrada $[j][i]$). Essa estrutura de dados também permite um acesso direto às informações referentes a um par de vértices, o que agiliza os algoritmos que os processam. Uma observação pertinente é que o valor 1 está sendo somado ao número de propostas pois o vértice referente ao índice 0 da matriz (também vale para sua negação) será utilizado para o tratamento dos casos em que um voto não é utilizado pelo eleitor, como já definido. Essa representação do grafo - a matriz - também é incluída como um componente do tipo abstrato de dados que modela a campanha eleitoral do deputado.

3. Execução & Funcionamento

O programa começa seu funcionamento inicializando uma variável do tipo abstrato de dados que modela a campanha, a qual será a base do processamento de todas as informações envolvidas sob o problema. Em seguida, as variáveis referentes a uma instância da enquete eleitoral são lidas da entrada de dados e utilizadas para o preenchimento da estrutura de dados mencionada. Nesse momento, o número de eleitores, o número de propostas, a matriz de preferências e o grafo que modela a enquete são todos declarados, inicializados e populados com os dados fornecidos ao programa.

A **verificação da satisfatibilidade** das implicações lógicas é feita por meio do algoritmo de pesquisa em largura **BFS (breadth-first search)**, o qual, a partir de um vértice de origem,

percorre as arestas - no sentido da largura do grafo - de modo a alcançar todos os demais vértices aos quais há um caminho partindo do primeiro. Em particular, como definido anteriormente, a expressão booleana que modela a enquete não pode ser satisfeita se existirem duas implicações nas formas $x \rightarrow (...) \rightarrow \neg x$ e $\neg x \rightarrow (...) \rightarrow x$, o que, em termos de grafos, é análogo a existir um caminho de x a $\neg x$ e um caminho de $\neg x$ a x . Assim, para cada vértice x , referente a uma proposta, é executado um procedimento que realiza duas buscas: uma para verificar se $\neg x$ é alcançável de x e outra para verificar se x é alcançável de $\neg x$. Caso ambos os caminhos sejam encontrados, é retornado o resultado de que o problema de satisfação dos eleitores não pode ser resolvido. Caso contrário, é verificada e retornada a possibilidade de resolução dessa questão.

Por fim, é realizada apenas a limpeza das variáveis alocadas para a estrutura de dados que representa a instância processada do problema, e a execução do programa é finalizada.

4. Considerações Finais

Instruções para compilação e para execução

1. Extraia o conteúdo do arquivo .zip disponibilizado.
2. Acesse o diretório `</tp1>` por meio do comando `<$ cd tp1>`.
3. Execute no terminal o comando `<$ make all>`. Esse comando deverá gerar um executável denominado **tp01**.
4. A execução do programa é realizada por meio do comando `<$./tp01 < input.txt>`, sendo **input.txt** o arquivo de texto contendo os dados de entrada.
5. A saída será retornada pela saída padrão de dados.

Referências Bibliográficas

Stack Exchange, Inc.. *Stack Overflow* [Online]. Disponível em: <https://stackoverflow.com/> (Acessado em: 07/10/2022).

GeeksforGeeks. *GeeksforGeeks* [Online]. Disponível em: <https://www.geeksforgeeks.org/> (Acessado em: 07/10/2022).