MO417 – Complexidade de Algoritmos

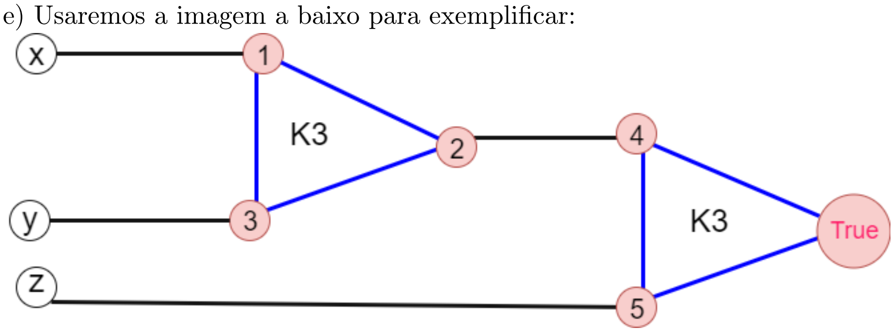
Nathana Facion - RA 191079 - Lista 11

19/06/2017

Questão 1.

d) Seja c uma 3-coloração de um grafo. Como c é uma 3-coloração então as suas adjacências devem ter cores diferentes. Seja v_i , vj e v_{red} os vértices envolvidos na coloração. Assumimos que v_{red} é o vértice já pintado de vermelho. Seguindo as arestas literais de v_{red} temos 2 vértices adjacentes v_i e v_j, como eles formamos um triângulo, ou K3, como preferir. Por meio disso sabemos que as 3 cores devem aparecer. Como já temos um vértice colorido c(RED), então os restantes devem ser c(TRUE) e c(FALSE). Considere v_i igual a c(TRUE), então v_j é c(FALSE), similar ocorre se v_i for c(FALSE). Então temos uma 3-coloração.

e) Usaremos a imagem a baixo para exemplificar:



Como temos dois K3, sabemos por meio da prova de (d) que cada um dos K3 serão coloridos por c(RED) , c(TRUE) e c(FALSE). Seja x,y,z vértices do grafo acima.

Como estamos tentando checar que é 3-coloração se e somente se pelo menos um de x,y,z são c(TRUE), vamos negar uma das direções do if

($\Rightarrow$)Primeiro vamos demonstrar que todos os vértices citados não podem ter a coloração c(FALSE). Por contradição, considere que são todas c(FALSE), então V_1 e V_3 deve ser coloridos com c(TRUE) e c(RED), independente de qual dos dois vértice terá qual cor (desde que as cores não se repitam). Então V_2, por ser um K3, com certeza será c(FALSE).

Agora vamos observar z, como sua coloração é c(FALSE), então V_4 e V_5 teriam que ser c(RED) e c(TRUE). Entretanto a cor(TRUE) está sendo usada no K3, então temos c(FALSE) e c(RED) para o triângulo, ao atribuirmos c(RED) para o v_5, teremos problema com v_4, pois o mesmo terá a mesma cor de v_2.

($\Leftarrow$) Vamos demonstar agora, por meio de todos os casos, que se eu tenho um literal que é colorido c(TRUE), então com certeza temos que o grafo é 3-coloração.

Caso	x	y	z	V_1	V_2	V_3	V_4	V_5
1	c(TRUE)	c(FALSE)	c(FALSE)	c(FALSE)	c(RED)	c(TRUE)	c(FALSE)	c(RED)
2	c(FALSE)	c(FALSE)	c(TRUE)	c(RED)	c(FALSE)	c(TRUE)	c(RED)	c(FALSE)
3	c(TRUE)	c(TRUE)	c(FALSE)	c(FALSE)	c(TRUE)	c(RED)	c(FALSE)	c(RED)
4	c(TRUE)	c(FALSE)	c(TRUE)	c(FALSE)	c(RED)	c(TRUE)	c(FALSE)	c(RED)
5	c(TRUE)	c(TRUE)	c(TRUE)	c(FALSE)	c(TRUE)	c(RED)	c(FALSE)	c(RED)

Considere que no Caso 1, foi omitido o caso que y é c(TRUE) , x e z são c(FALSE), pois é o mesmo caso. No Caso 4, foi omitido o caso que y e z são c(TRUE) e x é c(FALSE), pelo mesmo motivo.

Logo, em cada caso onde pelo menos uma das entradas é TRUE, existe uma atribuição de cores para os outros vértices que valida uma 3-coloração, portanto está provado. ☐

f) Para provar que 3-coloração está em NP-Completo, precisamos mostrar que o mesmo está em NP e NP-Difícil.

NP: Para ser NP temos que ter uma certificação. Podemos verificar por meio do algoritmo abaixo:

Algoritmo

Algorithm 1 Verifica Certificado

function VERIFICA-3COLORAVEL($< G >, E$)
  $S \leftarrow 0$
  for $(u, v) \in E$ **do**
    if $(u, v) \notin E(G)$ or $c(u) == c(v)$ **then**
      devolva não
    $S \leftarrow S + 1$
  if $|G.E| == S$ **then**
    devolva sim
  devolva não

Como podemos ver, a verificação pode ser feita em tempo polinomial, ou melhor dizendo, em tempo linear.Portanto a parte de NP está provado.

Agora temos a parte referente a NP-difícil. Para conseguir demonstrar isso temos de reduzir 3-CNF-SAT para o problema 3-coloração. Isso precisa ser feito pois sabemos que 3-CNF-SAT é NP-difícil, então se reduzirmos 3-coloração de forma polinomial para 3 coloração, estará provado.

Prova: Na letra (d) demonstramos que 3-coloração, contem uma atribuição c(TRUE). Podemos complementar esse argumento na letra (e), para cada cláusula, sabemos que uma 3-coloração deve existir uma atribuição TRUE para um literal. Demonstramos também que a 3-coloração é possível se e somente se um literal é TRUE. Em termos de complexidade de algoritmo, seja n as variaveis e m a clausula, sua complexidade é de $O(n + m)$, portanto polinomial.

Agora, considere que temos uma instância de 3-CNF-SAT. Como podemos conferir no exercício anterior, cada um dos vértices literais e sua negação devem ter cor c(TRUE) e c(FALSE), e apenas um deles pode ser c(TRUE), por causa das arestas literais.

Logo, podemos perceber que uma coloração só será válida se atribuirmos a cor c(TRUE) para um dos vértices variáveis da cláusula. Então, conseguiremos uma atribuição, que faz com que pelo menos uma das cláusulas de entrada seja c(TRUE), se atribuirmos para cada variável que tem c(x_i)=c(TRUE) como TRUE e cada variável que tem c($\bar{x}_i$)=c(TRUE) para FALSE. Esta é uma atribuição que satisfaz a fórmula.

Concluimos então que conseguimos reduzir um 3-CNF-SAT para uma instância 3-coloração, provando portanto que como 3-coloração é NP e NP-Difícil então é por consequência NP-Completo. $\square$