Exame de Qualificação 06/09/2015

Álvaro Freitas Moreira e Luciana S. Buriol

March 27, 2017

Teoria da Computação

Perguntas

1. **Considere os seguintes problemas de decisão:**

   - ***HamCycle*: Dado um grafo $G = (V, E)$, ele contém ciclos cujos vértices são visitados exatamente uma vez?**
   - ***HamPath*: Dado um grafo $G = (V, E)$ e dois vértices distintos s e t, ambos em V, existe caminho simples em G que começa em s, termina em t cujos vértices são visitados exatamente uma vez?**
   - ***EulerCycle*: Dado um grafo $G = (V, E)$, ele contém ciclo cujas arestas são visitadas exatamente uma vez?**

   Sabendo que:

   - ***HamCycle* é NP-Completo;**
   - ***HamCycle* é redutível em tempo polinomial a *HamPath*;**
   - ***EulerCycle* pode ser resolvido em tempo polinomial.**

   Responda se as seguintes afirmações são verdadeiras ou falsas? Justifique suas respostas.

   (a) ***EulerCycle* é redutível em tempo polinomial a *HamCycle*.**

   - *Verdadeiro.* Dados que HamCycle é NP-Completo, e todos os problemas em NP podem ser reduzidos a ele em tempo polinomial.

   (b) ***EulerCycle* é redutível em tempo polinomial a *HamPath*.**

   - *Verdadeiro.* Como EulerCycle pode ser resolvido em tempo polinomial, então ele pode ser reduzido para qualquer problema NP-Completo em tempo polinomial, inclusive HamPath.

   (c) ***HamPath* é redutível em tempo polinomial a *EulerCycle*.**

   - *Falso.* Sabe-se que EulerCycle pode ser resolvido em tempo polinomial, e que ele pode ser reduzido a outros problemas NP-Completos. Isso quer dizer que está em NP. Porém, nenhuma das afirmações anteriores demonstra que ele está em NP-Hard, condição necessária para que $HamPath \leqslant_p EulerCycle$ seja verdadeira.

2. **Mostre que *HamCycle* é redutível em tempo polinomial para *HamPath* (ambos os problemas são definidos no enunciado da questão anterior).**

- Uma redução é uma função ou algoritmo que executa em tempo polinomial, transformando a solução de um problema para outro. Se alguém, com uma entrada X para o problema *HamCycle* deseja um sim ou não como resposta, a funçao de redução irá trnasformar X em uma entrada para o problema *HamPath*. A função de redução transforma a saída, tal que a reposta do segundo problema é precisamente a mesma resposta do primeiro problema. This is very weird, I'm not sure if the transformation occurs in the output or in the input for reduction... Needs some review.
- Afirmação de que *HamCycle* é redutível a *HamPath*. Prova: Dado um grafo G, para toda aresta (u, v) executar *HamPath* com entrada $(G\{(u, v)\}, u, v)$. Assim caso exista um caminho simples entre os dois vértices u e $v \in G$, *HamPath* irá retornar sim. A resposta para o problema *HamCycle* é sim precisamente quando a resposta para a segunda questão for sim também.

3. **Dada uma prova da não existência da Máquina de Turing para decidir o Problema da Parada. explique o papel da Tese de Church-Turing na prova de que o Problema da Parada é indecidível.**

   - A Tese de Church-Turing afirma que se uma função f é efetivamente computável ela também é Turing-computável. Da mesma forma, toda função Turing-computável é efetivamente computável.
   - Para provar que o problema da parada é indecidível, a partir da tese de Church-Turing, basta demonstrar que não existe uma máquina de Turing capaz de computar um função com tal propriedade.
   - Pelo *Teorema de Rice* podemos demonstrar que não existe uma MT capaz de decidir se um algoritmo é capaz de computar uma função f, com uma determinada propriedade não trivial P. Uma implicação do teorema é que não podemos afirmar nada sobre um programa nem sobre o *status* de parada apenas olhando para seu código ou a linguagem aceita por ele. Isso impede que métodos sejam construídos para analisar de forma automática o comportamento de uma função. Por exemplo, não é decidível se um programa sempre responde com *sim*, ou se um programa é equivalente a outro programa, nem mesmo se ele sempre irá parar. Eles são todos indecidíveis.
   - Se não existe uma Máquina de Turing capaz de computar uma resposta para o problema, pela tese de Church-Turing podemos dizer que ele é indecidível. A Tese de CT não é provada formalmente, devido a definição de efetivamente computável ser informal, mas ainda assim é aceita pela comunidade científica. Sendo assim, o problema da parada está acima dos limites do que é computável.
   - Está é a prova utilizando o Teorema de Rice, está correta, mas acho que o ideal seria apresentar a outra forma. Se possível depois vou adicionar.