

Exame de Qualificação 07/08/2014

Álvaro Freitas Moreira e Luciana S. Buriol

March 26, 2017

Teoria da Computação

Perguntas

1. (2.5 pontos) Disserte sobre a Tese de Church-Turing. Deixe claro

(a) O que ela afirma:

- *Church Turing Thesis*: If a function f is effectively calculable, then f is Turing-computable. Equivalently, if a function f is not Turing Computable, then f is not effectively computable.

(b) Qual o seu significado para a Ciência da Computação:

- Assumindo a tese de Church-Turing, se uma função f não é Turing computável, ela também não é computável em qualquer outro sentido. Isso é, se a tese de CT é verdadeira, então essa função f está estritamente além das fronteiras do que é computável. Sendo assim, a Tese de CT define os limites ou as fronteiras do que é computável e o que não é.
- O Problema da Parada é historicamente importante porque foi um dos primeiros problemas a ser provado como indecidível (ambas as provas foram publicadas dentro um curto espaço de tempo por Church e Turing). Subsequentemente, foi possível provar que outros problemas também pertencem a esta classe (problema indecidível); o método típico para provar que um problema é indecidível é a redução. Para fazer isso, é necessário mostrar que se uma solução para um novo problema for encontrada, ela poderá ser utilizada para decidir se um problema é indecidível, seguindo um processo onde as instâncias de um problema indecidível são transformadas em instâncias de um novo problema. Uma vez que já sabemos que nenhum método pode resolver o antigo problema, por consequência, nenhum método também é capaz de solucionar o novo problema. Frequentemente, o novo problema é reduzido para a solução do Problema da Parada

(c) Por que ela não pode ser provada formalmente:

- A Tese de Church-Turing se trata de uma tese e não de um teorema, uma vez que não pode ser provada. Se a afirmação indicando que uma função f é efetivamente computável também é Turing-computável fosse falsa, deveria ser esperada a existência de ao menos uma função que pode ser efetivamente computável mas que ao mesmo tempo não é Turing-computável. O fato é que esta função nunca aconteceu ou jamais foi encontrada, sugerindo - mas de nenhuma forma provando - que esta função não existe, que por sua vez significa que a afirmação inicial é verdadeira.
- O motivo para isso é que entre todos os conceitos que esta tese envolve existe um que é informal e impreciso, chamado de "efetivamente computável" (*effective computability*). A tese iguala a matematicamente precisa noção de "computável por uma máquina de Turing" com a informal, intuitiva noção de "computável efetivamente," que alude que para todos computadores reais e todas as linguagens de programação, aquelas que nós conhecemos no

presente e também aquelas que não. Isso soa muito mais como uma especulação selvagem do que o que ela realmente é: uma profunda e abrangente afirmação, colocada por dois dos pioneiros mais respeitados da ciência da computação teórica.

(d) **Por que, mesmo sem prova formal de sua verdade, ela é aceita como sendo um dos fundamentos da Teoria da Computabilidade:**

- Por muito tempo pesquisadores tentaram desenvolver um computador dito universal tentando capturar a noção elusiva de efetivamente computável. Turing sugeriu seu modelo de máquinas primitivas e Church desenvolveu um formalismo matemático simples de funções chamadas cálculo lambda. Outros métodos desenvolvidos, como é o caso das funções recursivas, também obtiveram sucesso em utilizar seus modelos para resolver diversos problemas algorítmicos que ficaram conhecidos como algoritmos efetivamente computáveis. Alguns modelos mais complexos se aproximavam até mesmo dos computadores atuais, porém, o fato crucial sobre esses modelos é que todos se mostravam equivalentes em termos de classe da algoritmos que eram capazes resolver.
- Em resumo, o fato de que diversas pessoas trabalharam com diferentes ferramentas e técnicas foram capazes de capturar o mesmo conceito serve como evidência para a profundidade da afirmação da tese de Church-Turing. Devido todas elas partirem de um mesmo conceito e terminarem com aparentemente diferentes definições, mas equivalentes, podemos utilizar isso como justificativa para assemelhar essa noção intuitiva com os resultados das outras definições precisas.

2. (2.5 pontos) Um dos mais importantes e interessantes teoremas da Teoria da Computabilidade é o Teorema de Rice. Ele afirma que toda propriedade não-trivial de funções computadas por programas é não-decidível. Uma propriedade de funções parciais é dita *trivial* se ela é verdadeira para todas as funções ou para nenhuma. Note que o Teorema se refere a *funções* computadas por programas e não à representação ou forma como a função é computada (por exemplo, o número de transições de um programa em uma Máquina de Turing ou o número de linhas de código de um programa em C são propriedades da representação ou forma, não da função computada).

(a) **Dê pelo menos 3 exemplos de propriedades não triviais de funções computadas por programas.**

- A função f retorna 1 para pelo menos uma das suas entradas possíveis;
- A função f retorna 1 para todas as entradas possíveis;
- A função f é constante.

(b) **Disserte sobre a significância do Teorema de Rice para a prática da Ciência da Computação.**

- Uma implicação do Teorema de Rice é que não podemos afirmar nada sobre um programa nem seu *status* de parada apenas olhando para a sua representação, forma, ou ainda a linguagem aceita por ele. Impedindo a construção de métodos que automatizem a análise de comportamento de uma função. Por exemplo, não é possível decidir se um programa sempre responde **sim**, ou se o programa é equivalente a outro programa, e mesmo que se o programa irá terminar. Todas as propriedades não-triviais são indecidíveis.

3. (2.5 pontos) Um conjunto S é contável se é vazio ou se existe uma sequência s_0, s_1, s_2, \dots , com todos (e somente todos) os elementos de S . Está sequência é chamada de enumeração de S . Exemplos de conjuntos contáveis:

- O conjunto de números naturais é contável. Uma sequência óbvia mencionando todos os elementos é $0, 1, 2, \dots, n$.

- O conjunto de todos os números inteiros é contável. Uma sequência para os seus elementos é $0, 1, -1, 2, -2, 3, -3, \dots, n, -n$.
- O conjunto $\mathbb{N} \times \mathbb{N}$ é contável. Uma sequência é $(0,0), (0,1), (1,0), (0,2), (1,1), (0,3), (1,2), (2,1), (3,0), \dots, (n, n)$.

Esta terminologia também se aplica a conjuntos de funções, parciais ou totais. Sejam A e B conjuntos e seja S um conjunto não vazio cujos elementos são funções parciais de A para B . Então S é contável se existe uma sequência f_0, f_1, \dots, f_n tal que $g \in S$ se e somente se $g = f_i$, para algum i ou seja, g ocupa alguma posição i na sequência.

Prove as seguintes afirmações:

- O conjunto de todas as funções totais de \mathbb{N} para \mathbb{N} é incontável;
- O conjunto de todas as funções parciais de \mathbb{N} para \mathbb{N} é incontável;
- O conjunto de todas as funções computáveis parciais de \mathbb{N} para \mathbb{N} é contável;
- O conjunto de todas as funções computáveis totais de \mathbb{N} para \mathbb{N} é contável;
- Existe uma função não computável total \mathbb{N} para \mathbb{N} ;
- Existe uma função não computável parcial \mathbb{N} para \mathbb{N} ;

Observações:

- A prova de (a) acima pode ser feita por contradição usando o método da diagonalização.
- A prova de (b) segue imediatamente do resultado (a) acima e de teoria dos conjuntos.
- A prova de (c) faz uso da Tese de Church-Turing.
- A prova de (d) é imediata a partir da afirmação (c) e de conhecimentos básicos de funções e de teoria dos conjuntos.
- A prova de (d) e (e) também é imediata a partir dos resultados anteriores e de teoria dos conjuntos.

Ou seja, o esforço de prova está na verdade nos itens (a) e (c) acima. A prova das demais afirmações pode ser feita assumindo a verdade das afirmações anteriores usando resultados básicos da Teoria dos Conjuntos de acordo com as sugestões acima.

- Prova por contradição, assume-se que o conjunto de funções totais de \mathbb{N} para \mathbb{N} é contável. Por diagonalização é possível demonstrar que sempre existirá uma função g que não pertence ao conjunto das funções existentes, $g \in \mathbb{N} \rightarrow \mathbb{N}$, onde $g(x) = f_x(x) + 1$, para todo x , $g \neq f_i$. Ou seja, $\mathbb{N} \rightarrow \mathbb{N}$ é incontável (Harel, pg. 206).
- O conjunto de funções totais X é um subconjunto das funções parciais Y , sendo assim, se um conjunto Y tem um subconjunto X que é incontável, então Y também é incontável. Portanto, através do uso de Teoria dos Conjuntos, o conjunto de funções parciais é incontável.
- Assumindo que a Tese de Church-Turing é verdadeira, se uma função f é computável então ela é computável por uma máquina de Turing. Máquinas de Turing podem ser codificadas por números naturais, sendo assim, utilizando a definição de um conjunto contável e a Tese de CT, o conjunto de funções computáveis parciais é contável.
- Uma vez que o conjunto de funções computáveis parciais é contável, o conjunto de funções totais computáveis que por sua vez é um subconjunto do anterior, por Teoria dos Conjuntos, também é contável.
- Afirmção (a) + (d).
- Afirmção (b) + (c).