Exame de Qualificação 11/12/2014

Álvaro Freitas Moreira e Luciana S. Buriol

November 3, 2016

Teoria da Computação

Material

- Turing & The Halting Problem - Computerphile
- Algorithmics: The Spirit of Computing, p. 228
- Models of Computation and Formal Languages, p. 395

Perguntas

1. **Escreva sobre a tese de Church-Turing. Em sua resposta você deve:**

   (a) **Escrever o enunciado da tese**
   Church Turing Thesis: If 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) **Explicar porque é uma tese e não um teorema (ou seja, explicar por que ela não pode ser provada)**
   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.
   The reason for this is that among the concepts it involves there is one that is informal and imprecise, namely that of "effective computability." The thesis equates the mathematically precise notion of "solvable by a Turing machine" with the informal, intuitive notion of "solvable effectively," which alludes to all real computers and all programming languages, those that we know about at present as well as those that we do not. It thus sounds more like a wild speculation than what it really is: a deep and far-reaching statement, put forward by two of the most respected pioneers of theoretical computer science.

   (c) **Escrever sobre as evidências que suportam a sua verdade (mencionar duas dessas evidências)**
   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 algoritmicos 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 que diversas pessoas trabalharem com diferentes ferramentas e técnicas serem 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.

(d) **Explicar a sua importância para a Ciência da Computação, em particular explicar porque ela é importante em provas de resultados negativos, como a indecidibilidade do Problema da Parada.**