MATA51 - Teoria da Computação
Docente: Laís Salvador
Discentes: Adrielle Andrade Carvalho, Fernando Franco de Lacerda Neto, João Lucas Lima de Melo e Thiago Vieira Souza Andrade

Atividade - História da Computação, Modelos de Computação e Teorema da Incompletude de Godel

1. O problema da parada é um problema indecidível, pois ele constitui uma linguagem RE não recursiva. Será que o problema da parada pode ser decidido com o lambda-cálculo? Justifique sua resposta.

Há uma equivalência entre lambda-cálculo e máquinas de turing de tal forma que o uso de cálculo-lambda como linguagem de programação é Turing-computável. É possível, por exemplo, codificar estados e símbolos de uma MT utilizando números de forma a traduzir uma máquina em termos lambda.

Uma vez que não é possível decidir o problema da parada por máquinas de Turing, e há uma equivalência entre o modelo e lambda-cálculo, o problema da parada não pode ser decidido com lambda-cálculo.

2. Se eu desenvolvo um algoritmo para um dado problema X, é possível desenvonver uma função mi-recursiva para esse mesmo problema? Por que?

Como é possível desenvolver um algoritmo para um problema X, isso implica que X é um problema computável. Sendo um problema computável, há uma máquina de Turing MT que pode realizar a computação. O modelo de máquinas de Turing possui mesmo poder computacional que funções mi-recursivas, de forma que um modelo pode ser usado para representar o outro. Portanto, uma vez que há uma MT para X, há também uma função mi-recursiva para X.

3. Enuncia e explique os teoremas da incompletude de Godel.

   1º Teorema da Incompletude: Existem nas linguagens, a partir de uma certa complexidade das mesmas, afirmações expressáveis em termos dos números naturais que não podem ser provadas a partir dos axiomas e, mesmo que aumentemos o número de axiomas independentes, sempre existirão afirmações que não poderão ser provadas nessas linguagens.

   O teorema pode ser melhor entendido pela demonstração de Godel, que codifica fórmulas lógicas em números, usando propriedades de números primos. Dessa forma, enunciados da meta-teoria sobre demonstrabilidade são traduzios em enunciados simples da matemática elementar.

   Godel introduziu uma fórmula G que afirma a sua idemonstrabilidade quando decodificada na meta-linguagem, com forte conexão com paradoxos semânticos. A verdade de G é conhecida na meta-teoria, mas não é possível demonstrá-la na teoria (a linguagem do objeto).

   Podemos ainda usar como exemplo para entendimento do teorema a expressão “os axiomas não se contradizem entre si”. A expressão pode ser formulada em termos aritméticos sobre a

base de números primos, ainda que seja formalmente indecidível. Dessa forma, a expressão não pode ser refutada nem provada sobre a base de outros axiomas.

2º Teorema da Incompletude: Nenhum dos sistemas aos quais se aplica o primeiro Teorema da Incompletude pode ser suficientemente poderoso para demonstrar sua própria consistência.

É possível usar o mesmo exemplo da expressão “os axiomas não se contradizem entre si”, explicado no parágrafo anterior, para entender a segundo Teorema da Incompletude. O exemplo implica que qualquer prova de consistência de uma teoria razoavelmente complexa, como a aritmética, necessita de linguagens mais poderosas que as da teoria considerada. Isso é observável no caso da aritmética, por exemplo.

4. Explique o impacto dos teoremas da incompletude de Godel (1931) sobre o programa de Hilbert (1928).

Hilbert imaginava um programa para uma fundamentação da Matemática através de um processo de sucessivas fundações, onde a consistência de uma teoria matemática complexa seria derivada de uma mais simples.

Godel, no entanto, demonstrou as limitações do formalismo através dos seus Teoremas da Incompletude por mostrar que a derivação de teoremas não poderia nunca ser completamente mecanizada por um algoritmo.

5. Como relacionar os teoremas da incompletude de Godel (1931) com os trabalhos posteriores de Turing, Kleene e Church sobre modelos de computação?

Para fazer a demonstração de seus teoremas, Godel definiu uma classe de funções capaz de fazer manipulação simbólica através de operações aritméticas. Essa classe de funções corresponderia às funções computáveis, usadas nos trabalhos de Turing, Kleene e Church.

6. Por que a Tese de Church é uma tese científica, mas não é matemática?

A Tese de Church não foi formalmente comprovada. No entanto, a tese apresenta um modelo suficientemente robusto, eficiente e aceito na comunidade científica, evidenciado pela a proliferação de formalismos nunca mais expressivos que o da máquina de Turing.

Referências

VIEIRA, Newton José. Linguagens e Máquinas: Uma Introdução aos Fundamentos da Computação. Belo Horizonte, 2004.

C.V. D'Alkaine. Os trabalhos de Godel e ad denominadas ciências exatas. Em homenagem ao centenário do nascimento de Kurt Godel. São Paulo: Revista Brasileira de Ensino de Física, 2006.

TASSO DOURADO. DOC Teocomp - História da Teoria da computação. YouTube, 6 de abril de 2021. Disponível em: <https://www.youtube.com/watch?v=a7Ixq4fHoVc>.

SIPSER, Michael. Uma Introdução à Teoria da Computação. PWS Publishing Company, 1997.