



Figura 1.3: Esquema ilustrativo da Tese de Church-Turing.

1.2 Tese de Church-Turing

A Tese de Church, ou hipótese de Church-Turing, ilustrada na Figura 1.3, não é um teorema mas sim um resultado epistêmico cuja aceitação é quase universal. O conceito intuitivo de computável é identificado com uma certa classe de funções aritméticas chamadas funções recursivas, cuja caracterização é um dos objetivos deste livro.

O leitor pode pensar que se a hipótese não pode ter sua veracidade comprovada de forma direta, então talvez ela possa ser refutada. Assim, para negar a tese basta encontrar um procedimento que não pudesse demonstradamente ser computado por uma Máquina de Turing. Até o presente momento isto não ocorreu, e ainda, em virtude do grande número de dados experimentais favoráveis, esta tese tem sido aceita pelos estudiosos do assunto. Além disto, as diversas tentativas independentes de formalizar o conceito de algoritmo resultaram em formalismos que podem ser demonstrados como equivalentes ao de Turing.

Neste sentido, a Tese de Church é uma hipótese sobre a natureza mecânica do ato de calcular, relacionando-se diretamente com o computador, e com os tipos de algoritmos eles podem executar. Assim, toda função considerada sistematizável pode ser computada por uma Máquina de Turing. Programas podem ser traduzidos em uma Máquina de Turing, bem como, qualquer Máquina de Turing pode ser traduzida para uma linguagem de programação. Conseqüentemente, qualquer linguagem de

programação de propósito geral é suficiente para expressar qualquer algoritmo, seja ele qual for.

Até o presente, ninguém foi capaz de produzir um modelo computacional que tenha maior poder que o de Turing. Trata-se de um modelo mais geral que os computadores atuais, pois não é limitado por questões restritivas como espaço de armazenamento e memória.

Uma consequência positiva da Tese de Church é que se o ser humano consegue resolver um determinado problema, de forma consciente, sem adivinhação ou mímica, então é possível construir uma máquina que também o resolva. Uma aplicação prática disto é o campo de estudo conhecido como Inteligência Artificial ¹. O lado negativo é que os problemas que não podem ser resolvidos pela Máquina de Turing, também não serão resolvidos exclusivamente pelo raciocínio humano. Desta forma, existem problemas que possuem soluções, mas que nunca serão encontradas através do raciocínio humano.

Gödel trabalhou muito tempo com questões filosóficas relativas ao contraste entre mentes e máquinas [9]. Para ele a mente humana é incapaz de mecanizar todas as intuições matemáticas, fato este que poderia ser chamado de incompletude da Matemática, intimamente relacionado com o problema da incompletude. Outro resultado afirma que, ou a mente humana consegue ultrapassar qualquer máquina ou, então, existem questões da teoria dos números que são indecidíveis para a mente humana. Entretanto, Gödel, assim como Hilbert, se recusava a admitir o comportamento absolutamente irracional, de acordo com o qual a mente seria capaz de formular questões que, pela sua própria natureza, lhe seja impossível responder.

¹O termo Inteligência Artificial é um nome brilhantemente criado por McCarthy em 1956 para chamar a atenção de investidores para aquela área de conhecimento. McCarthy disse que tratava-se da capacidade de uma máquina de realizar funções que se fossem realizadas pelo ser humano seriam consideradas inteligentes. Ora, se a máquina consegue desempenhar essa tarefa, é porque esta mesma é sistemática, e por definição nunca pode ser dita inteligente. Contudo, apesar deste cheque-mate à respeitabilidade [8] o termo é um bom nome para vender a idéias para o público leigo.

1.3 Teoremas de Gödel

Em 1931, Gödel² tornou públicos seus teoremas relacionados com a indecidibilidade e incompletude da matemática. Estes teoremas se contrapunham ao otimismo científico dominante da época que acreditava poder resumir toda a matemática, e até mesmo outras ciências naturais, em um único sistema formal livre de contradições. Seus teoremas delinearam novos contornos ao domínio da matemática e influenciam não só as ciências correlatas, como a computação, mas também outros campos do conhecimento humano, como a filosofia.

No fim do século XIX e início do século XX a matemática era profundamente influenciada pelas idéias do matemático alemão David Hilbert (1862-1943) pertencente a uma escola chamada formalista que atuou ao longo de todo o século XX. Em 1900, durante o Congresso Internacional de Matemáticos, realizado em Paris, ele propôs uma coleção de vinte e três problemas inconclusos que direcionaram grande parte da pesquisa matemática no decorrer do século XX, alguns destes problemas ainda hoje não são resolvidos, e outros o foram apenas parcialmente. Em 1920, ele sugeriu explicitamente uma linha de pesquisa na qual se defendia que os desenvolvimentos destes problemas aconteceriam a partir de uma amostra finita de proposições, devidamente selecionadas, chamadas axiomas. Além disto, tal sistema axiomático provavelmente seria consistente, ou seja, livre de contradições. A partir destes axiomas, os formalistas, seguidores de Hilbert, acreditavam ser possível desenvolver uma teoria única que contemplasse toda a matemática de forma a torná-la completa.

Ora, a matemática é amplamente utilizada para formalizar a modelagem de

²Kurt Gödel (1906-1978), conhecido como senhor “Por que?”, nasceu em Brno, atualmente República Tcheca. Ainda na infância sofre um ataque de febre reumática, o que lhe debilitou físico e psicologicamente tornando-o hipocondríaco durante toda a vida. Em 1939, foi incluído na lista negra nazista talvez por ser intelectual, talvez devido ao seu círculo de amizades com judeus, talvez por fazer parte do Círculo de Viena, e assim sendo fugiu para a Universidade de Princeton nos Estados Unidos da América. Não era uma pessoa muito sociável e cultivou poucas relações de amizade, entre elas a de Albert Einstein com quem trabalhou na Teoria da Relatividade. Com o passar dos anos, sua hipocondria foi piorando, até que morreu de fome no hospital de Princeton, ocasião em que se convenceu de que o estavam envenenando pela alimentação.

fenômenos da natureza. Sob esta ótica, o otimismo científico apontava para a possibilidade de modelagem de todos os problemas, inclusive daqueles relativos às ciências naturais. Além disto, Hilbert defendia também, que um problema matemático deveria admitir obrigatoriamente uma solução exata, seja ela de forma direta ou através da demonstração de sua impossibilidade de resolução. Esta característica de singularidade de uma solução passou a ser conhecida como decidibilidade de um problema. Quando se afirma que uma proposição é verdadeira ou falsa, está-se enunciando um dos princípios da lógica clássica (lógica de predicados de primeira ordem). Trata-se do princípio do terceiro excluído (*tertium non datur*) pois tal proposição não é ao mesmo tempo verdadeira e falsa, tampouco nem verdadeira, nem falsa. Informalmente tem-se: “O que é, é, o que não é, não é, e não há uma terceira opção”.

Entretanto, a decidibilidade se contrapõe à existência de paradoxos que são facilmente criados, como por exemplo, a seguinte sentença: “*Esta afirmação é falsa*”. É fácil perceber que os seguidores de Hilbert claramente enfrentaram dificuldades para construir uma matemática livre de contradições, decidível, completa e consistente. Neste sentido, torna-se importante compreender algumas definições de termos comuns a este assunto.

Definição 1.1 Um conjunto de enunciados verdadeiros, selecionados para serem axiomas de um sistema qualquer, é dito *completo* se é possível obter novamente, através de demonstrações, todos os enunciados verdadeiros do objeto de estudo que se propõe axiomatizar.

Definição 1.2 Um sistema axiomático é *consistente* se não é possível provar, a partir dos axiomas, uma contradição, isto é, um enunciado e sua negação.

Definição 1.3 Uma classe de enunciados que não podem ser demonstrados nem refutados dentro de um sistema axiomático é chamada de *indecidíveis*. Se em um sistema axiomático existe algum problema indecidível, então este sistema é incompleto.

Uma etapa importante da axiomatização da matemática proposta por Hilbert foi realizada pelo matemático italiano Giuseppe Piano (1858-1932), em 1889.

Apesar de outros matemáticos terem apresentado anteriormente³ tais sistemas de axiomatização foi Peano que conseguiu fazê-lo através de uma meta-linguagem de primeira ordem, ou seja, através de uma linguagem onde aparecem apenas predicados e proposições aplicados a objetos. Um exemplo é a definição de identidade feita através de duas propriedades de relacionamento:

- $a = a$ (reflexão); $(a = b) \rightarrow (b = a)$ (simetria); $(a = b \wedge b = c) \rightarrow a = c$ (transitividade)
- $a = b \rightarrow \varphi(a) = \varphi(b)$ (se a igual a b , b possuirá a mesma propriedade de a)

A principal característica da aritmética de Peano é a definição de sucessor. Através dela, os seguintes axiomas apresentam as propriedades fundamentais dos números naturais \mathbb{N} .

1. $\neg \underline{suc}(x) = 0$: O 0 não é sucessor de nenhum número natural;
2. $\underline{suc}(x) = \underline{suc}(y) \rightarrow x = y$: Se dois números naturais x e y têm o mesmo sucessor então eles são iguais;
3. $[p(0) \wedge (\forall x)(p(x) \rightarrow p(\underline{suc}(x)))] \rightarrow (\forall x_1), p(x_1)$: Trata-se do princípio da indução matemática (que será explorado mais profundamente na seção 2.6), isto é, se a propriedade é válida para o elemento inicial e a sentença “*se a propriedade p é verdadeira para um número x , então ela também é verdadeira para o sucessor*”.

A fim de construir um sistema formal que demonstra a existência de teoremas indecidíveis, Gödel criou uma linguagem estritamente numérica⁴ associando os sinais primitivos da aritmética de Peano a números naturais primos.

³Em 1888, Dedekind apresentou uma axiomatização similar à apresentada por Peano um ano depois. Entretanto, Ao contrário de Peano, Dedekind usa uma linguagem de segunda ordem, onde expressões fazem referência a outras expressões e não apenas objetos.

⁴O processo consiste em traduzir o enunciado da meta-linguagem de Peano para a linguagem-objeto da aritmética e por isso é notadamente denominado aritmetização [10]

O problema a seguir, conhecido como quebra-cabeça godeliano, proveniente de Raymond Smullyan [11], expressa a idéia geral do teorema de Gödel. Suponha a existência de uma máquina que imprime setenças, ou cadeias, compostas pelos símbolos

$$\neg, P, N$$

As sentenças que a máquina é capaz de imprimir estão em uma das quatro formas a seguir:

- PX
- PNX
- $\neg PX$
- $\neg PNX$

onde X é uma cadeia qualquer.

A máquina que questão segue as seguintes regras:

1. PX é verdadeiro \Leftrightarrow a máquina imprime X ;
2. PNX é verdadeiro \Leftrightarrow a máquina imprime XX ;
3. $\neg PX$ é verdadeiro \Leftrightarrow a máquina não imprime X ;
4. $\neg PNX$ é verdadeiro \Leftrightarrow a máquina não imprime XX ;
5. $\neg X$ é a negação de X . Para todas as cadeias X , ou elas são verdadeiras, ou sua negação é verdadeira;
6. Todas as sentenças impressas pela máquina são verdadeiras;
7. A máquina nunca imprime sentenças falsas.

Este tipo de construção é um caso de auto-referência: *o programa imprime sentenças que definem o que o programa pode e não pode imprimir*, portanto o programa descreve seu próprio comportamento. Seja um program deste tipo, que

imprime apenas sentenças verdadeiras sobre si mesmo. É possível que o programa seja capaz de imprimir todas as sentenças verdadeiras?

Considere a combinação $\neg PN \neg PN$, e suponha que a máquina imprime esta cadeia. Utilizando a regra 2, e assumindo $X = \neg PN$, tem-se que é verdadeira a sentença $\neg PN \neg PN \equiv \neg PNX \Leftrightarrow \neg XX$. Desfazendo a substituição, obtém-se $\neg XX \equiv \neg \neg PN \neg PN \equiv PN \neg PN$ é verdadeiro porque a hipótese era que a máquina imprime a sentença $\neg PN \neg PN$. Além disto, pela regra 5, a negação de $PN \neg PN$, isto é, $\neg PN \neg PN$ é falsa. Ora, pela regra 7, a máquina nunca imprime sentenças falsas, então ela não deveria imprimir $\neg PN \neg PN$. Assim, a suposição de que a máquina imprime $\neg PN \neg PN$ está incorreta.

Desta forma, supõem-se então que a máquina não imprime $\neg PN \neg PN$. Entretanto, se essa impressão não ocorre, e tomando a regra 4, tem-se que $\neg PNX \equiv \text{verdadeiro}$. Por substituição $X = \neg PN$, tem-se $\neg PN \neg PN \equiv \text{verdadeiro}$ também. Assim, $\neg PN \neg PN$ é verdadeira, mas a máquina não imprime a sentença. Assim, a suposição de que a máquina não imprime $\neg PN \neg PN$ está incorreta. Logo, tem-se duas hipóteses:

- a sentença é imprimível, mas ela é falsa, o que configura uma contradição, pois o programa só imprime sentenças verdadeiras;
- a sentença é verdadeira, mas ela não imprimível, o que fere o propósito da máquina.

A analogia entre este enigma e o teorema da incompletude está no paralelo entre um programa que imprime cadeias verdadeira e um sistema formal que gera teoremas verdadeiros.

O artigo original de Gödel pode ser encontrado em [12]. Na primeira seção daquele artigo, o esboço da demonstração indica claramente a consequência final do teorema e o caminho a ser tomado para chegar a ela. No entanto, são necessárias dezenas de definições e teoremas para completar rigorosamente a prova. À luz da Tese de Church, pode-se dispensar o arcabouço formal e apresentar apenas a genialidade da idéia global.

O princípio fundamental é o de auto-referência: Gödel parte do sistema formal para a aritmética, definido em Principia Mathematica (PM) [13], e “reescreve-o” utilizando a linguagem dos números naturais. Este processo de *aritmética* permite que os conceitos e as propriedades daquele sistema formal sejam transformados em conceitos e propriedades sobre os números naturais.

Em suma, a aritmética de Gödel é um mapeamento entre os elementos do sistema formal e os números naturais. A seguir, apresenta-se uma pequena variação do mapeamento original utilizado por Gödel, encontrado em Nagel and Newman [14]. Inicialmente, tem-se os símbolos do sistema formal, na tabela a seguir. A correspondência é feita com os 7 primeiros números ímpares.

símbolo	significado	número
\neg	negação	1
\rightarrow	se-então	3
\exists	existe	5
0	zero	7
<u>suc</u>	função sucessor	9
(abre parêntese - separador	11
)	fecha parêntese - separador	13

Considera-se também a existência de 3 tipos de variáveis: (1) x_1, x_2, \dots que podem ser substituídas por números; (2) $\alpha_1, \alpha_2, \dots$ que podem ser substituídas por fórmulas; e (3) P_1, P_2, \dots que podem ser substituídas por predicados. A numeração das variáveis é como mostra a tabela a seguir.

variável	número	exemplo
x_i	i-ésimo primo > 13	17, 19, 23, 29, ...
α_i	i-ésimo primo > 13 ao quadrado	$17^2, 19^2, 23^2, 29^2, \dots$
P_i	i-ésimo primo > 13 ao cubo	$17^3, 19^3, 23^3, 29^3, \dots$

O número de Gödel de uma expressão de k símbolos é o produto dos k

primeiros primos elevados aos números de Gödel de seus constituintes. Por exemplo:

$$\begin{array}{ccc} \neg & & \alpha_1 \\ \downarrow & & \downarrow \\ 1 & & 17^2 \\ 2^1 & \times & 3^{17^2} \end{array}$$

ou ainda, o exemplo:

$$\begin{array}{ccccccc} P_1 & (& \underline{suc} & (& x_1 &) &) \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\ 17^3 & 11 & 9 & 11 & 17 & 13 & 13 \\ 2^{17^3} & \times & 3^{11} & \times & 5^9 & \times & 7^{11} & \times & 11^{17} & \times & 13^{13} & \times & 17^{13} \end{array}$$

O número de Gödel de uma sequência de k expressões é o produto dos k primeiros primos elevados aos números de Gödel de cada expressão. Por exemplo, se $n = 2^1 \times 3^{17^2}$ e $m = 2^{17^3} \times 3^{11} \times 5^9 \times 7^{11} \times 11^{17} \times 13^{13} \times 17^{13}$ então o número de Gödel da sequência $\neg\alpha_1, P_1(s(x_1))$ é $2^n \times 3^m$.

A aritmetização das seqüências de fórmulas pode ser aplicada a qualquer seqüência finita de fórmulas, especialmente nas demonstrações axiomáticas, cujas definições são obtidas por indução. Embora Gödel objetivasse demonstrar a relação de consistência através da aritmética de Peano, Guerreiro [10] ressalta que tal método permite que qualquer linguagem seja reproduzida de maneira semelhante, especialmente as linguagens de programação.

Além do fato que um número de Gödel representar apenas um símbolo ou fórmula, outra característica importante desse método é o fato de que qualquer número de Gödel pode ser revertido novamente na fórmula ou símbolo original, bastando para isso decompor esse número em números primos.

Como será observado a seguir, a tabela de substituição proposta por Gödel em [15] é suficiente para tradução de axiomas da aritmética de Peano. Mais tarde, Stephen Cole Kleene, um dos alunos de Gödel em seu curso apresentado na Universidade de Princeton em 1934, desenvolveria a teoria das funções recursivas (objeto de estudo no Capítulo 3) propondo um conjunto de funções iniciais e um conjunto de construtores de funções que podem ser encadeados, traduzindo assim qualquer função computável [16].

A primeira vista, pode parecer que a associação é insuficiente para um algoritmo completo ou mesmo para traduzir os axiomas da aritmética de Peano. Até mesmo por isso, foi acrescentada a associação com primos maiores que 13 para obter os exemplos já expostos. Gödel sobrepõe estas dificuldades através do conceito de funções recursivas primitivas (ver Capítulo 3) onde, uma vez definidas funções muito simples e estabelecendo algumas regras, é possível derivar qualquer função que possua um número finito de etapas.

Kleene amplia esse resultado, afirmando que uma função (ou programa) é computável se puder ser obtida a partir de algumas funções básicas e da definição de alguns construtores, ou seja, ele define o que se pode processar por um computador. Embora esse resultado seja eminentemente teórico [16], o trabalho necessário para verificar a computabilidade de funções é muito simples.

Como estes modelos de computação serão estudados nos próximos capítulos, o foco no momento será o processo de Gödel. Para reconhecer uma expressão, dado seu número de Gödel n , basta observar certas regras, como:

1. se $n \leq 13$ é ímpar, a expressão é um símbolo;
2. se $n > 13$ é ímpar e é potência de um número primo, a expressão é uma variável;
3. se n é par, decompondo-se n em fatores primos:
 - (a) se for o produto de primos sucessivos com expoentes ímpares, então potencialmente é uma fórmula;
 - (b) se for o produto de primos sucessivos com expoentes pares, então potencialmente é uma seqüência de mais de uma fórmula.

Outras condições devem ser observadas para que um natural seja número de Gödel para uma fórmula. Por exemplo, $30 = 2^1 \times 3^1 \times 5^1$ corresponde a $\neg\neg\neg$, que não é uma fórmula no sistema formal pretendido. De qualquer modo, o leitor pode imaginar como construir um programa que enumere/reconheça fórmulas de um sistema formal com os símbolos apresentados, segundo esta aritmetização.

Para aritmetizar o conceito de derivável por Modus Ponens, observe o exemplo:

fórmula	número
$P_2(x_1) \rightarrow P_1(x_1)$	$\frac{2^{19^3} \times 3^{11} \times 5^{17} \times 7^{13} \times 11^3 \times 13^{17^3} \times 17^{11} \times 19^{19} \times 23^{13}}{2^{19^3} \times 3^{11} \times 5^{17} \times 7^{13}}$
$P_2(x_1)$	$2^{19^3} \times 3^{11} \times 5^{17} \times 7^{13}$
$P_1(x_1)$	$2^{17^3} \times 3^{11} \times 5^{19} \times 7^{13}$

O número da segunda premissa é o produto dos 4 fatores primos iniciais da primeira premissa. Os expoentes da conclusão são os mesmos dos fatores primos da primeira premissa, após o fator 11^3 correspondente ao símbolo \rightarrow . Novamente, o leitor deve imaginar como construir um programa que aplique Modus Ponens sobre dois números de Gödel. Este programa computa a função característica da relação *DerivávelPorMP*(x, y, z).

Para aritmetizar a regra de substituição de uma variável por um termo, observe o exemplo:

fórmula	número
$P_1(x_1)$	$2^{17^3} \times 3^{11} \times 5^{17} \times 7^{13}$
$\underline{suc}(0)$	$2^9 \times 3^{11} \times 5^7 \times 7^{13}$
$P_1(\underline{suc}(0))$	$2^{17^3} \times 3^{11} \times 5^9 \times 7^{11} \times 11^7 \times 13^{13} \times 17^{13}$

Há, de fato, uma substituição do fator 5^{17} correspondente ao x_1 , com a inserção dos expoentes do número de $\underline{suc}(0)$ aplicados aos primos a partir do 5, e os expoentes dos fatores à direita do 5^{17} foram deslocados para primos após o 13. O programa assim descrito computa a função característica da relação *DerivávelPorSUB*(x, y).

Extrapolando estas observações, Gödel criou predicados aritméticos para conceitos tais como *SerFórmula*, *SerAxioma*, *SerTeorema*, todos eles conjuntos recursivos. Para chegar ao ponto crucial do teorema, foram utilizados, especialmente:

1. a função $sub(y, 17, y)$, que calcula o número de Gödel da fórmula obtida a partir da fórmula de número y ($P_1(x_1)$), pela substituição da variável x_1 por

y novamente. Por exemplo:

$$\begin{array}{ccc} \text{sub}(2^{17^3} \times 3^{11} \times 5^{17} \times 7^{13} & , & 17 & , & 2^{17^3} \times 3^{11} \times 5^{19} \times 7^{13} &) \\ \downarrow & & \downarrow & & \downarrow & \\ P_1(x_1) & & x_1 & & 2^{17^3} \times 3^{11} \times 5^{17} \times 7^{13} & \end{array}$$

é o número de Gödel de $P_1(2^{17^3} \times 3^{11} \times 5^{17} \times 7^{13})$, isto é, $P_1(P_1(x_1))$.

2. o predicado $\acute{E}ProvaPara(y, z)$ que representa que a seqüência de fórmulas de número de Gödel y é uma dedução para a fórmula de número de Gödel z .

Considere então a fórmula:

$$\neg \exists x_2 \acute{E}ProvaPara(x_2, x_3) \cdots (G_0)$$

que diz, essencialmente, “não existe uma dedução para a fórmula de número de Gödel x_3 ”. Substituindo x_3 por $\text{sub}(x_1, 17, x_1)$ temos

$$\neg \exists x_2 \acute{E}ProvaPara(x_2, \text{sub}(x_1, 17, x_1)) \cdots (G_1)$$

que diz, essencialmente, “não existe uma dedução para a fórmula de número de Gödel $\text{sub}(x_1, 17, x_1)$ ”. Em seguida, seja n o número de Gödel de (G_1) . Portanto, o valor de $\text{sub}(n, 17, n)$ é o número de Gödel de

$$\neg \exists x_2 \acute{E}ProvaPara(x_2, \text{sub}(n, 17, n)) \cdots (G)$$

que diz, essencialmente, “não existe uma dedução para (G) ”.

Analisando o conteúdo da fórmula (G) , temos duas possibilidades:

1. se existe uma dedução no sistema formal da aritmética para (G) , então é possível deduzir uma contradição dentro do sistema;
2. se não existe uma dedução para (G) então (G) é verdadeira, porém não é possível deduzir todas as verdades aritméticas dentro do sistema, já que (G) não é dedutível.

Diante desta situação, Gödel se viu numa situação delicada em que era necessário fazer uma escolha: (1) ou que existem enunciados verdadeiros que não podem ser demonstrados; (2) ou que existem demonstrações podem provar a afirmação

e a negação de um mesmo enunciado. Sob esta ótica, Gödel foi obrigado a postular os Teoremas da Incompletude e da Consistência, descritos a seguir.

Teorema 1.1 [Teorema da Incompletude] Todo sistema axiomático consistente e recursivo para a aritmética possui enunciados indecidíveis. Em particular, se os axiomas do sistema são enunciados verdadeiros, pode existir um enunciado verdadeiro que não é demonstrável dentro deste sistema.

Teorema 1.2 [Teorema da Consistência] Um enunciado que expressa a consistência de um sistema axiomático recursivo para a aritmética não é demonstrável dentro deste sistema.

Após este trabalho, vários matemáticos e filósofos avançaram sobre os teoremas ampliando suas implicações para matemática (Alfred Tarski, 1902-1983; Paul Cohen, 1934-2007; John Rosse, 1907-1989), para a informática (Johann Von Neumann, 1903-1952; Alonzo Church, 1903-1995; Alan Turing, 1912-1954; Joseph Bernard Kruskal, 1928-2010) e para a filosofia (Hao Wang, 1921-1995), tais como:

- Nenhum programa de computador pode demonstrar todas as proposições verdadeiras da matemática. Uma vez que a recursividade corresponde a um dos paradigmas da Computabilidade (Tese de Church-Turing) é natural que a Ciência da Computação assuma que existem enunciados que não podem ser provados ou refutados, e ainda, que o princípio de bivalência do valor verdade deve ser rejeitado;
- Nenhuma programa de computador pode ser, ao mesmo tempo, não contraditório e completo;
- Não existe programa capaz de verificar, de antemão, a presença de laços infinitos em um programa, isto é, não existe um algoritmo que pode ser aplicado a qualquer programa arbitrário, com uma entrada, para decidir se este programa para ou não com a dada entrada.

A imediata recepção pelos matemáticos da época à apresentação dos teore-

mas de Gödel ⁵ foi cautelosa. Essa cautela se justificava pela divergência não só aos resultados matemáticos apresentados na mesma época, mas também ao direcionamento “hilbertiano” que os trabalhos tomavam em geral. Um claro exemplo pode ser encontrado na citação que Hilbert realizou apenas dois dias depois no mesmo congresso (ainda ignorante aos resultados alcançados por Gödel): “Para o matemático, não existe *ignorabinus* (ignoraremos) e, na minha opinião, o mesmo vale para as ciências naturais. A verdadeira razão porque ainda não conseguimos encontrar um problema insolúvel reside, segundo creio, no fato deles não existirem. (...)”, conforme Guerreiro [10].

Essa multiplicidade de implicações que torna o trabalho de Gödel atrativo a matemáticos e leigos. Como se pode verificar em Kubrusly [17] e Guerreiro [10] seus teoremas inspiram as mais diversas conclusões, enveredando até a teologia, área abordada matematicamente por Gödel no fim de sua vida. Contudo, é a “derrocada da pretensa segurança” [10] da matemática o resultado geral que mais se sobressai para ambos os grupos. O desenvolvimento da matemática e de ciências correlatas como a informática, traduzem, de certa forma, o desenvolvimento do próprio pensamento humano e, por isso, da própria humanidade. As demonstrações de Gödel em 1931 marcaram um momento de inflexão da matemática moderna, indo mais longe que mera demonstração da falibilidade da disciplina: Gödel expande o domínio da matemática promovendo uma nova abordagem através de sua lógica e sua repercussão é claramente visível nos dias de hoje através da informática.

1.4 Conclusões e leituras recomendadas

Sem dúvida, o mais conhecido e respeitado dos especialistas em Gödel é Smullyan, cuja bibliografia sobre o tema de indecidibilidade é bastante extensa. Entre seus livros encontram-se textos extremamente profundos como [11] e [18]; e livros lúdicos, que ilustram o conceito através de enigmas e charadas, tais como [19], [20]. Goldstein [21] também oferece um material didático e profundo sobre a

⁵Congresso sobre teoria do conhecimento nas ciências exatas (1931) realizado na cidade bohemia de Königsberg, hoje Kaliningrado, anexada a Rússia após a II Guerra Mundial.

prova de Gödel. O texto de Hofstadter [22] contém discussões sobre auto-referência, comparando os trabalhos de Gödel na matemática, Escher nas artes plásticas e Bach na música.

Especulações sobre o significado e as consequências do Teorema da Incompletude, tanto em matemática como em filosofia são abundantes. Algumas destas consequências são aceitas com restrições sérias, por parte da comunidade científica, e o próprio Gödel em trabalhos que só foram publicados após sua morte em Janeiro de 1978 [23], discute exaustivamente questões que variam, desde o conceito de inteligência até a possibilidade (na qual ele acreditava) da existência de Deus. Dentre as obras que abordam o legado lógico-filosófico de Gödel recomenda-se fortemente a coleção [24]. O texto *Reflections on Kurt Gödel* [25] contém uma excelente biografia comentada pelo grande lógico Hao Wang.

1.5 Exercícios

1.1 Explique o que se entende por atividade inteligente e atividade computável.

1.2 Qual é a proposta da Tese de Church-Turing?

1.3 Calcule o número de Gödel associado à expressão $P_2(x_1) \rightarrow \alpha_1$.

1.4 Determine a expressão associado ao número de Gödel 37.968.750.

Referências Bibliográficas

- [1] HOPCROFT, J. E., *Theory os Machines and Compuatations*, chapter An n log n algorithm for minimizing states in a finite automaton, Academic Press, pp. 189–196, 1971.
- [2] MONTEIRO, A. A., PAULO, J. D. S., *Aritmética Racional*. Lisboa, Livraria Avelar Machado, 1945.
- [3] IFRAH, G., *Os Números: a história de uma grande invenção*. São Paulo, Globo S.A., 2005.
- [4] DEDEKIND, R., *Was sind und was sollen die Zahlen*, v. 3, *Gesammelte Mathematische Werke*. New York, Chelsea Publishing Company, 1969. pp. 335-391.
- [5] GÖDEL, K., *Collected Works*, v. 2, *Gesammelte Mathematische Werke*. Oxford, Oxford University Press, 1990.
- [6] ISRAEL, D., “Reflections on Gödel’s and Gandy’s Reflection on Turing’s Thesis”, *Minds and Machines*, v. 12, n. 2, pp. 181–201, 2002.
- [7] SOBRINHO, J. Z., “Aspectos da Tese de Church-Turing”, *Revista Matemática Universitária - USP*, v. 1, n. 6, pp. 1–23, 1987.
- [8] MCDERMOTT, D., “Artificial Intelligence Meets Natural Stupidity”, *SI-GART Newsletter*, v. 57, pp. 4–9, April 1976.
- [9] SETTI, M. D. O. G., *O Processo de Discretiza çã o do Raciocínio Matemático na Tradu çã o para o Raciocínio Computacional*, Report, Universidade Federal do Paraná, 2009.

- [10] GUERREIRO, G., “A Vanguarda Matemática e os Limites da Razão”, *Scientific American Brasil*, v. 5, n. 12, pp. 39–56, 2007. Coleção Gênios da Ciência.
- [11] SMULLYAN, R., *Gödel’s Incompleteness Theorems*. Oxford University Press, 1992.
- [12] GÖDEL, K., *The Undecidable*, chapter On Formally Undecidable Propositions of Principia Mathematica and Related Systems, New York, Raven Press, pp. 5–38, 1965.
- [13] WHITEHEAD, A. N., RUSSELL, B., *Principia Mathematica*. Londres, Cambridge University Press, 1913.
- [14] NAGEL, E., NEWMAN, J. R., *Gödel’s Proof*. USA, Routledge, 1989.
- [15] GÖDEL, K., “Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme - On Formally Undecidable Propositions of Principia Mathematica and Related Systems”, *Kurt Gödel: Collected Works*, v. 1, n. 1, pp. 144–195, 1986. Tradução para o inglês por Martin Hirzel. www.research.ibm.com/people/h/hirzel/papers/canon00-goedel.pdf [capturado em 13 de agosto de 2011].
- [16] MELO, A. C. V. D., SILVA, F. S. C. D., *Modelos Clássicos de Computação*, Coleção Schaum. São Paulo, Thomson, 2006.
- [17] KUBRUSLY, R. D. S., *Uma viagem informal ao Teorema de Gödel ou (O preço da matemática é o eterno matemático)*, Report, Universidade Federal do Rio de Janeiro, 2007. IM/UFRJ.
- [18] SMULLYAN, R., *Recursion Theory for Metamathematics*. Oxford University Press, 1993.
- [19] SMULLYAN, R., *What’s the Name of This Book*. Penguin Books, 1978.
- [20] SMULLYAN, R., *Forever Undecided: A Puzzle Guide to Gödel*. Oxford University Press, 1987.
- [21] GOLDSTEIN, R., *Incompleteness: The Proof and Paradox of Kurt Gödel*. W. W. Norton Company, Inc., 2005.

- [22] HOFSTADTER, D. R., *Gödel, Escher e Bach: an Eternal Golden Braid*. Nova Iorque, Basic Books, 1979.
- [23] Rodríguez-Consuegra, F. A. (ed.), *Kurt Gödel - Unpublished Philosophical Essays*. Berlin, Birkhäuser Verlag, 1995.
- [24] Feferman, S., et al. (eds.), *Kurt Gödel - Collected Works*, v. I, II, III. New York, Oxford University Press, 1986.
- [25] WANG, H., *Reflections on Kurt Gödel*. Cambridge, Massachusetts, The MIT Press, 1988.
- [26] SUPPES, P., *Axiomatic Set Theory*. New York, Dover, 1972.
- [27] LIPSCHUTZ, S., *Teoria Elementar dos Conjuntos*, Cole çã o Schaum. São o Paulo, McGraw-Hill do Brasil, 1990.
- [28] SUPPES, P., *Axiomatic Theory Set*. USA, Van Nostrand Company Inc., 1960.
- [29] MELLO, F. L. D., CARVALHO, R. L. D., “Knowledge Geometry”, *Journal of Information and Knowledge Management*, v. 14, pp. 1550028, 2015.
- [30] STOLL, R. R., *Set Theory and Logic*. USA, Dover Publications Inc., 1961.
- [31] MIRAGLIA, F., *Teoria dos Conjuntos: um mínimo*. Brasil, Edusp - Editora da Universidade de São Paulo, 1992.
- [32] HALMOS, P., *Teoria Ingênua dos Conjuntos*. Brasil, Edusp - Editora da Universidade de São Paulo, 1970.
- [33] GRATZER, G., *Universal Algebra*. USA, Van Nostrand Company Inc., 1968.
- [34] COHN, P. M., *Universal Algebra*. USA, Harper and Row, 1965.
- [35] GALLIER, J. H., *Logic for Computer Science*. USA, John Wiley and Sons Inc., 1987.
- [36] PREPARATTA, F. P., YEH, R. T., *Introduction to Discreate Structures for Computer Science and Engineering*. USA, Addison-Wesley Publishing Company, 1973.

- [37] SHOENFIELD, J. R., *Degrees of Unsolvability*. North-Holland Publishing Company, 1971.
- [38] BRAINERD, W. S., LANDWEBER, L. H., *Theory of Computation*. USA, John Wiley and Sons, 1974.
- [39] EILENBERG, S., ELGOT, C., *Recursiveness*. New York: Academic Press, 1970.
- [40] CARVALHO, R. L. D., OLIVEIRA, C. M. G. M. D., *Modelos de Computação e Sistemas Formais*, 11^a Escola de Computação. Rio de Janeiro, Universidade Federal do Rio de Janeiro, 1998.
- [41] BRAINERD, W. S., LANDWEBER, L. H., *Theory of Computation*. John Wiley & Sons, 1974.
- [42] KLEENE, S. C., *Introduction to Metamathematics*. D. Van Nostrand Company, Inc., 1952.
- [43] Rogers Jr., H., *Theory of Recursive Functions and Effective Computability*. USA, McGraw-Hill Book Company, 1967.
- [44] DAVIS, M., *Computability and Unsolvability*. New York, Dover, 1983.
- [45] BOOLOS, G. S., JEFFREY, R. C., *Computability and Logic*. Cambridge University Press, 1974.
- [46] MARTIN DAVIS, R. S., WEYUKER, E. J., *Computability Complexity and Languages*. New York, Academic Press, 1994.
- [47] MALLOZZI, J. S., LILLO, N. J. D., *Computability with Pascal*. New Jersey, Prentice-Hall, Inc., 1984.
- [48] TURING, A. M., *The Undecidable*, chapter On Computable Numbers, with an Application to the Entscheidungsproblem, New York, Raven Press, pp. 115–151, 1965.
- [49] ELGOT, C. C., ROBINSON, A., “Random-access stored-program machines, an approach to programming languages”, *Journal of the ACM*, v. 11, pp. 365–399, 1964.

- [50] PREPARATTA, F. P., YEH, R. T., *Introduction to Discrete Structures for Computer Science and Engineering*. Addison-Wesley Publishing Company, 1973.
- [51] HENNIE, F., *Introduction to Computability*. Addison-Wesley Publishing Company, 1977.
- [52] NELSON, R. J., *Introduction to Automata*. USA, Jonh Wiley & Sons, Inc., 1968.
- [53] CARVALHO, R. L. D., *Máquinas, Programas e Algoritmos*, 2^a Escola de Computação. Campinas, Universidade Estadual de Campinas, 1981.
- [54] MENDELSON, E., *Introduction to Mathematical Logic*, Cole Mathematics Series. 3 ed. The Wadsworth and Brooks, 1987.
- [55] MANIN, Y. I., *A Course in Mathematical Logic*, Graduate Texts in Mathematics 53. 1 ed. Springer-Verlag, 1977.
- [56] HOMER, S., SELMAN, A. L., *Computability and Complexity Theory*, Texts in Computer Science. 2 ed. Springer, 2011.
- [57] CUTLAND, N. J., *Computability: An introduction to recursive function theory*. Cambridge University Press, 1980.
- [58] SIPSER, M., *Introduction to The Theory of Computation*, Course Technology Series. 2 ed. Thomson, 2006.
- [59] WALTER CARNIELLI, R. L. E., *Computability: computable functions, logic and the foundations of mathematics*. Belmont, Wadsworth and Brooks, 1989.
- [60] TARSKI, A., *Logic, semantics, metamathematics*, chapter Fundamental concepts of the methodology of the deductive sciences, London, Oxford at the Clarendon Press, pp. 60–109, 1969.
- [61] ADAM YOUNG, M. Y., *Malicious Cryptography: Exposing Cryptovirology*. John Wiley and Sons Inc., 2004.

- [62] BONFANTE, G., KACZMAREK, M., MARION, J.-Y., *A Classification of Viruses through Recursion Theorems*, volume 4497 of Lecture Notes in Computer Science. 2 ed. CiE 2007, 2007.
- [63] MACHTEY, M., YOUNG, P., *An Introduction to the General Theory of Algorithms*. New York, North Holland, 1978.
- [64] ROBINSON, J. A., “A machine oriented logic based on the resolution principle”, *J. Assoc. Comput.*, v. 12, pp. 23–41, 1965.
- [65] ROBINSON, J. A., “Automatic deduction with hyper-resolution”, *Internat. J. Comput. Math*, v. 1, pp. 227–234, 1965.
- [66] GILMORE, P. C., “A proof method for quantification theory: Its justification and realization”, *IBM Journal of Research and Development*, v. 4, n. 1, pp. 28–35, 1960.
- [67] DAVIS, M., PUTNAM, H., “A computing procedure for quantification theory”, *Journal of the ACM*, v. 7, n. 3, pp. 201–215, 1960.
- [68] CHANG, C.-L., LEE, R. C.-T., *Symbolic Logic and Mechanical Theorem Proving*. Academic Press Inc, 1973.
- [69] KLEENE, S. C., *Mathematical Logic*. Wiley, 1967.
- [70] HILBERT, D., ACKERMANN, W., *Prinmciples of Mathematical Logic*. Chelsea, 1950.
- [71] MCCAWLEY, J. D., *Everything That Linguists Have Always Wanted To Know About Logic*. 2 ed. The University of Chicago Press, 1993.
- [72] SUPPES, P., *Introduction to Logic*. D. van Nostrand, 1966.
- [73] RUSSELL, B., *A Filosofia do Atomismo Lógico*, *Lógica e Conhecimento*. 1 ed. Abril Cultural, 1974. (Os Pensadores, 42).
- [74] RUSSELL, B., *Significado e Verdade*. 1 ed. Zahar, 1978.
- [75] POPPER, K. R., *A Lógica da Pesquisa Científica*. 2 ed. Cultrix, 1974.

- [76] LAKATOS, I., , MUSGRAVE, A., *A Crítica e o Desenvolvimento do Conhecimento*. 1 ed. EDUSP, Cultrix, 1979. Tradução: M. O. Caiado.
- [77] WANG, H., *From Mathematics to Philosophy*. 1 ed. Cultrix, 1974.
- [78] GREEN, C. C., *The Application of Theorem Proving to Question-Answering Systems*. Ph.D. dissertation, Stanford, June 1969. AI Project MEMO AI-96.
- [79] LOVELAND, D. W., *Automated Theorem Proving: a Logical Basis*. 1 ed. North Holland, 1978.
- [80] HUGHES, G. E., LONDEY, D. G., *The Elements of Formal Logic*. USA, Methuen and Co Ltd, 1965.
- [81] BOOK, R. V., OTTO, F., *String-Rewriting Systems*. USA, Springer-Verlag, 1993.
- [82] HOPCROFT, J. E., ULLMAN, J. D., *Introduction to Automata Theory, Language and Computation*. USA, Addison-Wesley Publishing Company, 1979.
- [83] HOPCROFT, J. E., ULLMAN, J. D., *Formal Languages and their Relation to Automata*. USA, Addison-Wesley Publishing Company, 1969.
- [84] CURRY, H. B., *Foundations of Mathematical Logic*. New York, Academic Press, 1977.
- [85] SMULLYAN, R., *Theory of Formal Systems*. USA, Princeton, 1961.
- [86] BERSTEL, J., BOASSON, L., CARTON, O., *et al.*, *Handbook of Automata: from Mathematics to Applications*, chapter Minimization of automata, European Mathematical Society, pp. 189–196, 2010.
- [87] BEAL, M. P., CROCHEMORE, M., “Minimizing incomplete automata”, *Workshop on Finite State Methods and Natural Language Processing*, , september 2008. Ispra.
- [88] VALMARI, A., LEHTINEN, P., “Efficient minimization of DFAs with partial transition”, *Proc. 25th Symp. Theoretical Aspects of Comp. Sci.*, v. 08001, pp. 645–656, 2008. S. Albers and P. Weil, editors.

- [89] PAPADONIKOLAKIS, M., BOUGANIS, C.-S., CONSTANTINIDES, G.,
“Performance comparison of GPU and FPGA architectures for the SVM training problem”, *IEEE International Conference on FieldProgrammable Technology*, pp. 388–391, 2009.
- [90] MU, S., WANG, C., LIU, M., *et al.*, “Evaluating the potential of graphics processors for high performance embedded computing”, *Proc. IEEE Design, Automation and Test in Europe Conference and Exhibition*, pp. 709–714, 2011.
- [91] KAI HWANG, F. A. B., *Computer Architecture and Parallel Processing*. McGraw-Hill, 1984.
- [92] AGARWAL, P., KRISHNAN, S., MUSTAFA, N., *et al.*, *Algorithms - ESA 2003, Lecture Notes in Computer Science*, chapter Streaming Geometric Optimization Using Graphics Hardware, Springer Berlin-Heidelberg, pp. 115–151, 2003.
- [93] TANENBAUM, A. S., *Organização Estruturada de Computadores*. 3 ed. Prentice Hall do Brasil, 1997.
- [94] BACKUS, J., “Can Programming be Liberated from von Neumann Style? A functional style and its algebra of program”, *ACM Turing Award Lecture, Communications of the ACM*, v. 21, n. 8, pp. 613–641, 1978.
- [95] OWENS, J. D., LUEBKE, D., GOVINDARAJU, N., *et al.*, “A Survey of General-Purpose Computing on Graphics Hardware”, *Eurographics 2005, State of the Art Reports*, pp. 21–51, 2005.
- [96] GUSTAFSON, J. L., “Reevaluating Amdahl’s law”, *Communications of the ACM*, v. 5, n. 31, pp. 532, 1988.
- [97] HANDLER, W., *Parallel Processing Systems, an advanced course*, chapter Innovative computer architecture - how to increase parallelism but not complexity, Cambridge University Press, pp. 1–41, 1982.
- [98] LOBUR, J., NULL, L., *The Essentials of Computer Organization And Architecture*. Jones and Bartlett Pub, 2006.

- [99] LEWIS, H. R., PAPADIMITRIOU, C. H., *Elements of the Theory of Computation*. 2 ed. New York, Prentice-Hall, 1998.
- [100] DUNNE, P., *Computability Theory: Concepts and Applications*. Ellis Horwood, 1991.
- [101] AARONSON, S., “NP-complete Problems and Physical Reality”, *ACM SIGACT News*, , march 2005. Complexity Theory Column 46.