

## Sistema Axiomático

É um sistema lógico-matemático o qual consiste de um conjunto de axiomas estabelecidos como verdadeiros sem requererem demonstrações e de teoremas que são derivados desses axiomas. Toda os ramos da matemática podem ser axiomatizados, de forma a se tornarem sistemas formais. Certos axiomas, com é o caso do axioma das paralelas (a saber, por um ponto exterior à uma reta dada passa uma e só uma reta paralela àquela), são tidos como verdadeiros em sistemas axiomáticos – por exemplo a geometria plana Euclidiana – e são tidos como falsos em outros sistemas axiomáticos – como é o caso das geometrias não-Euclidianas, Hiperbólicas, etc. Ao assumirmos uma flexibilidade de linguagem, podemos dizer que: uma verdade matemática não é absoluta em si, mas é absoluta num sistema axiomático onde ela está inserida.

Duas grandes questões, levantadas por grandes filósofos e matemáticos de todas as épocas, são: a de saber se podemos garantir a consistência de um sistema formal, utilizando-se somente dele próprio; e a de saber se toda proposição num sistema formal pode ser demonstrada verdadeira ou falsa, ou seja, será que todas as perguntas tem resposta? Um sistema onde toda a pergunta possui resposta é um sistema completo. Por consistência, queremos dizer: livre de contradições, ou seja, um sistema em que nenhuma proposição pode ser demonstrada verdadeira e falsa simultaneamente. Esta questão foi respondida pelo matemático K. Gödel no princípio do século passado, da seguinte forma surpreendente:

Informalmente, o Primeiro Teorema da Incompletude de Gödel diz que todo sistema axiomático consistente (e com um certo grau de complexidade que inclua pelo menos os números naturais) encerra proposições (ou perguntas) que não são falsas nem verdadeiras (ou sem resposta).

O Segundo Teorema da Incompletude de Gödel diz que todo sistema axiomático (com um certo grau de complexidade que inclua pelo menos os números naturais) pode demonstrar sua própria consistência é inconsistente.

É um raciocínio engenhoso no qual Gödel mostra como podemos representar por números afirmações (ou teoremas) sobre números e, a seguir aplica de forma engenhosa e correta uma versão válida de um paradoxo matemático bastante conhecido. (o paradoxo do Barbeiro, ou do Mentiroso)

PARADOXO DO BARBEIRO. Um homem de Sevilha é barbeado pelo Barbeiro de Sevilha se e somente se o homem não barbeia a si mesmo. Pergunta: o Barbeiro de Sevilha barbeia a si próprio?

PARADOXO DO MENTIROSO. “Eu estou mentindo ao dizer esta frase!”, disse eu. Pergunta: eu menti?

UMA DEMONSTRAÇÃO INFORMAL. Consideremos um computador idealizado, sem limite predeterminado de memória. Neste computador vamos executar programas (em uma linguagem também idealizada). Um programa é descrito por um inteiro (obtido considerando-o como a expansão de um número na base  $N$ , onde  $N$  é maior do que o número de símbolos permitidos; nos computadores de verdade poderíamos tomar  $N = 256$ ). Alguns programas recebem um ou mais inteiros como entrada. Alguns programas terminam sua execução em tempo finito e outros nunca param.

Será possível decidir algoritmicamente olhando para um programa  $P$  e para sua entrada  $n$  se a execução vai parar algum dia? Ou seja, será que existe um programa  $X$  que recebe como entradas  $[P]$  (o número do programa  $P$ ) e  $n$  e responde "pára" ou "não pára", sempre após um tempo finito? A resposta é não.

A razão é que se existisse  $X$  poderíamos facilmente construir um outro programa  $Q$  que, recebendo como entrada  $[P]$ , pára se e somente se  $P$ , recebendo como entrada  $[P]$ , não pára. Basta a  $Q$  rodar  $X$  com entradas  $[P]$  e  $[P]$  e aguardar a resposta; se a resposta for "não pára",  $Q$  pára; se a resposta for "pára",  $Q$  joga-se deliberadamente em loop infinito. O paradoxo do barbeiro aparece se fazemos  $Q$  receber  $[Q]$ : a execução pára se e somente se a execução não pára...

Voltando ao teorema de Gödel, não é difícil convencer-se que a afirmação "a execução do programa  $P$  com entrada  $n$  para depois de tempo finito" pode ser traduzida para a aritmética. Se for verdadeira, a frase será um teorema: basta exibir a execução completa. É portanto impossível escrever um programa que decida em tempo finito se frases deste tipo são teoremas. Com mais forte razão, é impossível escrever um programa que decida se uma proposição aritmética é um teorema.

Mas se toda proposição aritmética ( $A$ ) fosse um teorema ou a negação de um teorema, seria muito fácil escrever um programa que decidisse qual das duas possibilidades é a correta: basta procurar sistematicamente demonstrações de ( $A$ ) e de ( $\neg A$ ). Assim, existem proposições que não são nem teoremas

nem a negação de teoremas, i.e., que não podem ser nem provadas nem refutadas.

Métodos de Demonstração

I – REDUÇÃO AO ABSURDO. Consideremos um sistema consistente (=nenhuma preposição pode ser verdadeira e falsa ao mesmo tempo), onde queremos demonstrar que dadas uma proposição P podemos demonstrar outra proposição Q, ou seja, queremos verificar o teorema  $P \Rightarrow Q$ . O método consiste em supor como verdadeiras as afirmações P e  $\sim Q$  (a negação de Q), e a partir daí mostrar que podemos encontrar proposições contraditórias no sistema, o que não é verdade *a priori*, portanto quando valer a hipótese P não se poderá negar a hipótese Q e portanto teremos demonstrado teorema  $P \Rightarrow Q$ .

OBS. A própria justificação acima consiste no uso de um método de redução ao absurdo, portanto não pode ser encarado como uma prova.

COMENTÁRIOS. O matemático G. Hardy disse: o método “*Reductio ad absurdum*”, que Euclides amava tanto, é uma das maiores armas. É uma estratégia maior que qualquer estratégia de xadrez: um jogador de xadrez pode oferecer o sacrifício de um peão, mas o matemático oferece o jogo inteiro.

PROBLEMA. Existem infinitos números primos.  
 RESOLUÇÃO. Suponhamos (essa é a hipótese de absurdo) que existem finitos números primos, sejam eles  $p_1, p_2, \dots, p_n$ . Considere o número  $K = p_1 p_2 \dots p_n + 1$ , é fácil de ver que nenhum primo  $p_1, p_2, \dots, p_n$  divide K, portanto o próprio K é um número primos, mas ele é maior que todos os números primos  $p_1, p_2, \dots, p_n$ , portanto existe outro número primo, isso é um absurdo, ou seja, temos uma contradição, que é a seguinte: nós chegamos a conclusão que se existem exatamente n primos distintos, então existe pelo menos um primo além desses n, ou seja, existem exatamente n primos distintos e também não existem exatamente n primos distintos. Supondo que a aritmética dos inteiros não encerra contradições, a hipótese de que existem finitos números primos não pode ser verdadeira, portanto precisa ser falsa, ou seja, existem infinitos números primos.

II – CONTRA-POSIÇÃO. Num sistema consistente, demonstrar um teorema  $P \Rightarrow Q$  é o mesmo que demonstrar o teorema  $\sim Q \Rightarrow \sim P$ . Utilizaremos uma redução ao

absurdo para provar isso. Por algum meio, conseguimos demonstrar o teorema  $\sim Q \Rightarrow \sim P$ , vamos demonstrar agora o teorema  $P \Rightarrow Q$ . Suponhamos por hipótese que vale P e também vale  $\sim Q$ , pelo que demonstramos deve valer também  $\sim P$ , por causa da implicação  $\sim Q \Rightarrow \sim P$ . Portanto valem P e sua negação  $\sim P$ , uma contradição ao fato de o sistema não conter contradições. Logo quando vale P valerá também Q, ou seja, vale o teorema  $P \Rightarrow Q$ .

PROBLEMA. Se a fração  $\frac{a}{b}$  é irredutível então a fração  $\frac{a+b}{b}$  é irredutível.

RESOLUÇÃO. Vamos demonstrar a contra-posição dessa afirmação. Se a fração  $\frac{a+b}{b}$  pode ser simplificada, então também pode a fração  $\frac{a}{b}$ . Ora, se um primo p divide a + b e b então ele divide a e b, portanto a fração  $\frac{a}{b}$  pode ser simplificada.

Bibliografia

Site Math World:  
[www.mathworld.com/](http://www.mathworld.com/)

Arquivo da Lista OBM-I:  
<http://www.mat.puc-rio.br/~nicolau/olimp/>

This document was created with Win2PDF available at <http://www.daneprairie.com>.  
The unregistered version of Win2PDF is for evaluation or non-commercial use only.