Reforço: Conjuntos e Diagonalização

Computabilidade e Complexidade

Márcio Nicolau

2025-08-18

Table of contents

List of Figures

Objetivos da Aula

- Revisar e solidificar a diferença entre conjuntos enumeráveis e não-enumeráveis.
- Aprofundar a compreensão da técnica de prova por diagonalização.
- Conectar explicitamente a diagonalização de Cantor com a prova da indecidibilidade do Problema da Parada.
- Reforçar por que esses conceitos são a base para os limites da computação.

Conteúdo

Revisitando o "Tamanho" dos Infinitos

No início do curso, estabelecemos uma ideia que parece paradoxal: nem todos os infinitos são do mesmo tamanho. Essa distinção é a pedra angular para entender por que existem problemas que os computadores não podem resolver. (Sipser, 2012)

> **Definições Essenciais (Revisão)**
>
> - **Conjunto Enumerável (ou Contável)**: Um conjunto é enumerável se seus elementos podem ser listados em uma sequência (ou seja, se existe uma correspondência um-para-um com os números naturais $\mathbb{N}$). Exemplos incluem os inteiros ($\mathbb{Z}$), os racionais ($\mathbb{Q}$) e, crucialmente, o conjunto de **todos os programas de computador válidos**.
> - **Conjunto Não-Enumerável (ou Incontável)**: Um conjunto infinito que é "maior" que os números naturais. É impossível listar todos os seus elementos. O exemplo clássico é o conjunto dos números reais ($\mathbb{R}$). O exemplo mais importante para nós é o conjunto de **todas as linguagens possíveis** (ou seja, todos os problemas computacionais).

A conclusão fundamental para a ciência da computação é este descompasso:

> Temos uma quantidade **enumerável** de possíveis soluções (programas), mas uma quantidade **não-enumerável** de possíveis problemas (linguagens).

Isso, por si só, já garante que devem existir problemas para os quais não há solução algorítmica. A ferramenta que nos permite provar isso rigorosamente é a diagonalização.

A Diagonalização: A Receita para Provar o Impossível

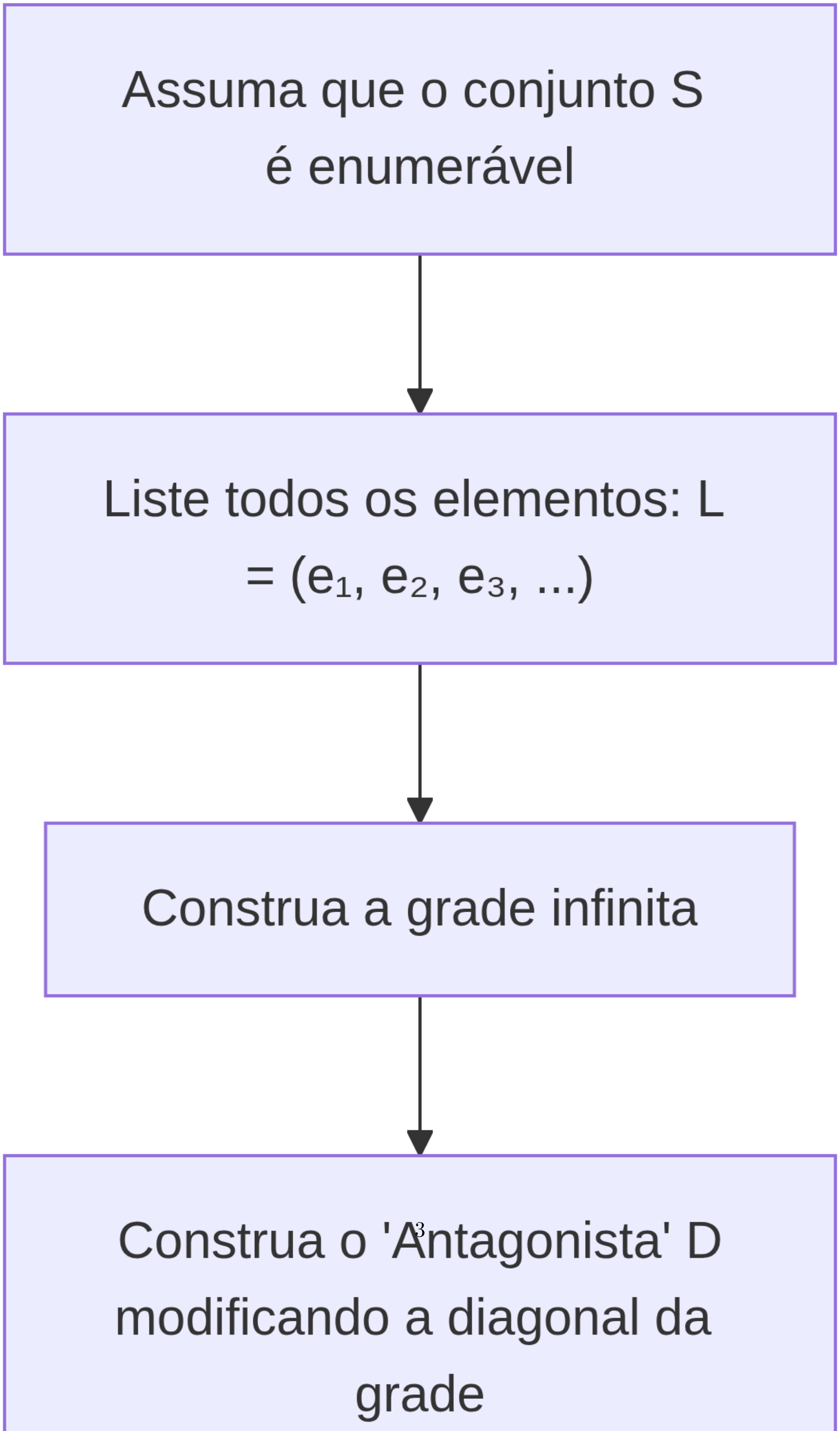
A prova por diagonalização é uma das técnicas mais elegantes e poderosas da matemática e da computação. É uma forma de prova por contradição com uma "receita" bem definida.

> **A Receita da Diagonalização**
>
> Para provar que um conjunto `S` é não-enumerável:
>
> 1. **Assuma o Contraditório**: "Suponha que `S` *seja* enumerável."
> 2. **Liste os Elementos**: Se é enumerável, podemos criar uma lista infinita $e_1, e_2, e_3, \ldots$ que contém *todos* os elementos de `S`.
> 3. **Construa a Grade**: Organize essa lista em uma grade infinita, onde cada linha representa um elemento e cada coluna representa uma "característica" ou "parte" desse elemento.
> 4. **Construa o Antagonista**: Crie um novo elemento, o "antagonista" D, olhando para a **diagonal** da grade. A construção de D garante que sua i-ésima característica seja diferente da i-ésima característica do i-ésimo elemento da lista (e_i).
> 5. **Encontre a Contradição**: O antagonista D é, por construção, um elemento válido do conjunto `S`. No entanto, ele não pode ser igual a nenhum elemento da lista, pois difere de e_i na i-ésima posição. Isso contradiz a suposição de que a lista continha *todos* os elementos.
> 6. **Conclua**: A suposição inicial era falsa. O conjunto `S` é não-enumerável.

Diagrama: A Lógica da Diagonalização

Aplicações: De Cantor a Turing

A genialidade de Turing foi perceber que essa mesma lógica poderia ser aplicada não apenas a números, mas a **computações**.


Assuma que o conjunto S é enumerável
Liste todos os elementos: L = (e_1, e_2, e_3, ...)
Construa a grade infinita
Construa o 'Antagonista' D modificando a diagonal da grade

Aplicação de Cantor (Revisão)

- **Conjunto**: Números reais $\mathbb{R}$ entre 0 e 1.
- **Grade**: Linhas são os números reais (r_i); colunas são seus dígitos decimais.
- **Diagonal**: O i-ésimo dígito do i-ésimo número.
- **Antagonista**: Um novo número real cujos dígitos são diferentes dos dígitos da diagonal.

Aplicação de Turing (A Prova do Problema da Parada)

- **Conjunto**: Todas as Máquinas de Turing.
- **Grade**: As linhas são todas as TMs (M_i). As colunas também são todas as TMs (M_j). A célula (i, j) contém o resultado da execução de M_i na entrada $\langle M_j \rangle$.
- **Diagonal**: A célula (i, i), que representa o que a máquina M_i faz quando recebe sua **própria descrição** como entrada.
- **Antagonista**: A máquina D que, ao receber $\langle M_i \rangle$, primeiro simula o comportamento da diagonal (M_i em $\langle M_i \rangle$) e depois faz **o oposto**.

Diagrama: O Paralelismo entre Cantor e Turing

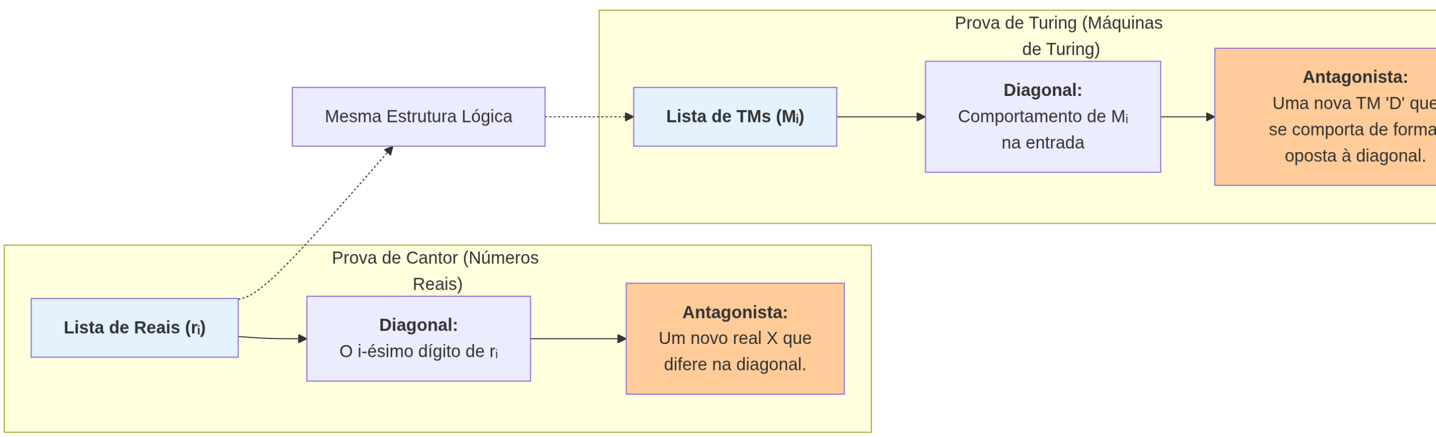


Figure 2: A mesma lógica de diagonalização usada em dois domínios diferentes.

Quando nos perguntamos "o que a máquina antagonista D faz com sua própria descrição $\langle D \rangle$?", estamos forçando-a a olhar para sua própria posição na diagonal, onde, por construção, ela deve fazer o oposto de si mesma, levando à contradição.

Exercícios de Verificação

Atividade Prática de Reforço

1. **Conceitual**: Explique com suas próprias palavras por que o conjunto de todas as strings finitas compostas pelos caracteres ‘a’, ‘b’ e ‘c’ é **enumerável**. Qual seria uma estratégia para listá-las sem deixar nenhuma de fora?
2. **Aplicação da Técnica**: Considere o conjunto `F` de todas as funções de $\mathbb{N}$ para $\mathbb{N}$ (ou seja, funções que recebem um número natural e retornam um número natural). Esboce um argumento de diagonalização para provar que este conjunto `F` é **não-enumerável**.
3. **Conexão Final**: Na prova da indecidibilidade de A_{TM}, a máquina antagonista D usa um suposto decisor H como sub-rotina. Qual é o papel da **diagonal** nessa prova específica? O que a consulta `H(<M>, <M>)` representa em termos da “grade” de Turing?

Referências Bibliográficas

SIPSER, Michael. **Introdução à Teoria da Computação**. 3. ed. São Paulo, Brasil: Cengage Learning, 2012.