Tese de Church-Turing e Modelos Equivalentes

Computabilidade e Complexidade

Márcio Nicolau

2025-09-01

Table of contents

List of Figures

Introdução: A Robustez da Computabilidade

Na aula anterior, formalizamos a noção de "algoritmo" usando a Máquina de Turing (TM). Definimos que uma linguagem é **decidível** se existe uma TM que para em todas as entradas, aceitando ou rejeitando. Mas isso nos deixa com uma pergunta fundamental: o modelo da Máquina de Turing é especial?

E se usássemos um modelo mais poderoso, como uma TM com várias fitas? Ou um modelo não-determinístico que pode explorar múltiplos caminhos de uma vez? Ou, mais pragmaticamente, e se simplesmente usássemos uma linguagem de programação moderna como Python? A nossa definição de "o que é computável" mudaria?

A resposta, surpreendentemente, é **não**. A classe de problemas que podem ser resolvidos por algoritmos é notavelmente robusta e não depende do modelo computacional específico que escolhemos (desde que ele seja razoável).

O objetivo desta aula é:

> **Entender a Tese de Church-Turing e a equivalência entre diferentes modelos de computação.**

A Tese de Church-Turing

Após Alan Turing e Alonzo Church, de forma independente, proporem seus modelos formais de computação (Máquinas de Turing e Cálculo Lambda, respectivamente), os cientistas da computação notaram que todos os modelos "razoáveis" de computação propostos pareciam ter exatamente o mesmo poder. Isso levou à formulação de uma das ideias mais centrais da ciência da computação.

! A Tese de Church-Turing

A noção intuitiva de "algoritmo" é equivalente em poder à noção formal de uma Máquina de Turing. Ou seja, qualquer função que pode ser efetivamente calculada por um procedimento algorítmico pode ser calculada por uma Máquina de Turing. (Baseado em (Sipser, 2012, p. 182))

É crucial entender a natureza desta afirmação. Ela não é um teorema matemático que pode ser provado. É uma **tese** (ou hipótese) que conecta um conceito informal e intuitivo (o que sentimos que é um "algoritmo") com um modelo matemático formal (a TM).

A evidência a seu favor é empírica e avassaladora: 1. **Convergência:** Todo modelo formal de computação já proposto (Cálculo Lambda, Funções Recursivas, etc.) provou ser equivalente em poder à Máquina de Turing. 2. **Robustez:** Como veremos, variantes aparentemente mais poderosas da Máquina de Turing ainda são equivalentes ao modelo padrão. 3. **Simulação:** Ninguém jamais encontrou um problema que acreditamos ser solucionável por um algoritmo para o qual não se pudesse construir uma Máquina de Turing.

i A Implicação Mais Profunda

A Tese de Church-Turing nos dá a confiança de que os limites que encontramos para as Máquinas de Turing são limites fundamentais da própria computação. Se provarmos que um problema é indecidível para uma TM, podemos afirmar com segurança que **nenhum computador**, presente ou futuro, e nenhuma linguagem de programação, por mais avançada que seja, será capaz de resolvê-lo.

Equivalência de Modelos: Variantes de TMs

Para fortalecer a Tese de Church-Turing, vamos examinar algumas variantes de TMs que parecem ser mais poderosas, mas que, na verdade, não são. Dizemos que dois modelos são **equivalentes** se um pode simular o outro, ou seja, se eles decidem e reconhecem a mesma classe de linguagens.

Máquinas de Turing Multi-fitas

Uma TM multi-fitas possui várias fitas de armazenamento, cada uma com seu próprio cabeçote de leitura/escrita. Em cada passo, a máquina pode ler os símbolos sob todos os cabeçotes e, com base nisso, escrever novos símbolos e mover cada cabeçote de forma independente.

Intuição: Parece muito mais poderoso e prático, como ter vários "rascunhos" ou registradores.

Equivalência: Uma TM de fita única padrão pode simular uma TM multi-fitas. A ideia é codificar o conteúdo de todas as k fitas em uma única fita, usando um caractere especial `#` como separador. Para simular as posições dos cabeçotes, usamos versões "marcadas" dos símbolos do alfabeto.

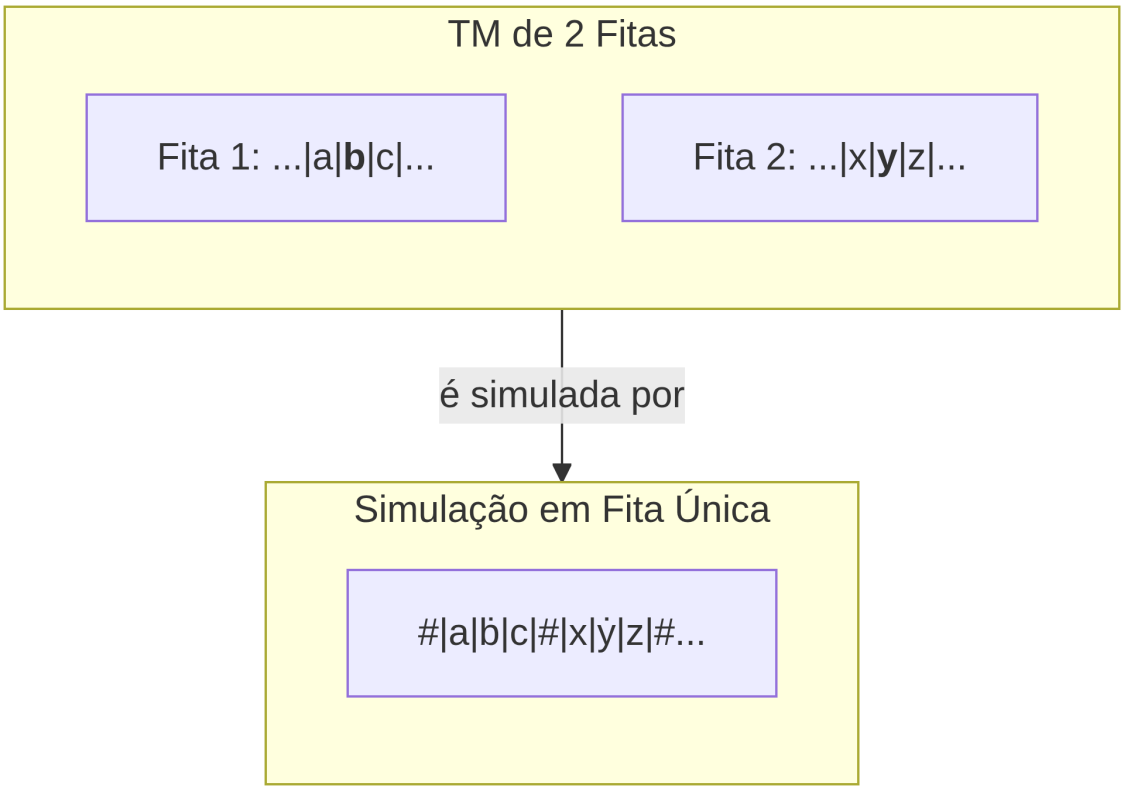

Figure 1: Simulação de uma TM de 2 fitas em uma TM de fita única.

Para simular um único passo da máquina multi-fitas, a máquina de fita única precisa varrer toda a sua

fita para encontrar os símbolos marcados (as posições dos cabeçotes), decidir a ação, e depois varrer a fita novamente para atualizar os símbolos e mover as marcações. É um processo mais lento, mas **possível**.

Máquinas de Turing Não-Determinísticas (NTMs)

Em uma TM padrão (determinística), cada configuração da máquina leva a exatamente uma próxima configuração. Em uma TM Não-Determinística (NTM), a função de transição pode especificar **múltiplas** próximas configurações possíveis.

$\delta : Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\})$

A NTM "explora" todos esses caminhos computacionais simultaneamente. Ela aceita uma entrada se **pelo menos um** dos caminhos leva ao estado de aceitação.

Intuição: Parece um salto massivo de poder, como uma forma de computação paralela massiva.

Equivalência: Uma TM determinística (DTM) pode simular uma NTM. A DTM simula a NTM explorando a árvore de computações da NTM de forma sistemática, por exemplo, usando uma **busca em largura (breadth-first search)**. A fita da DTM armazena uma fila de todas as configurações possíveis da NTM em cada nível da árvore.

A simulação é exponencialmente mais lenta, mas novamente, **possível**. Isso significa que NTMs e DTMs decidem a mesma classe de linguagens. A enorme diferença de velocidade entre elas é a base para uma das maiores questões abertas da computação: o problema **P vs. NP**.

Outros Modelos de Computação

A robustez da Tese de Church-Turing é reforçada pela descoberta de que muitos outros modelos, desenvolvidos com propósitos diferentes, também são equivalentes à TM.

Qualquer um desses modelos poderia ter sido usado como nossa definição formal de "algoritmo", e toda a teoria da computabilidade que desenvolveremos seria a mesma. A Máquina de Turing é frequentemente preferida por ser a mais "mecânica" e fácil de visualizar como um computador.

Verificação de Aprendizagem

1: Conceitual

Explique a Tese de Church-Turing com suas próprias palavras. Por que ela é considerada uma "tese" e não um "teorema"?

2: Simulação e Equivalência

Uma variante da Máquina de Turing é a "TM com fita duplamente infinita", onde a fita se estende infinitamente para a esquerda e para a direita. Descreva, em alto nível, como uma TM padrão (com fita infinita apenas para a direita) poderia simular esta variante.

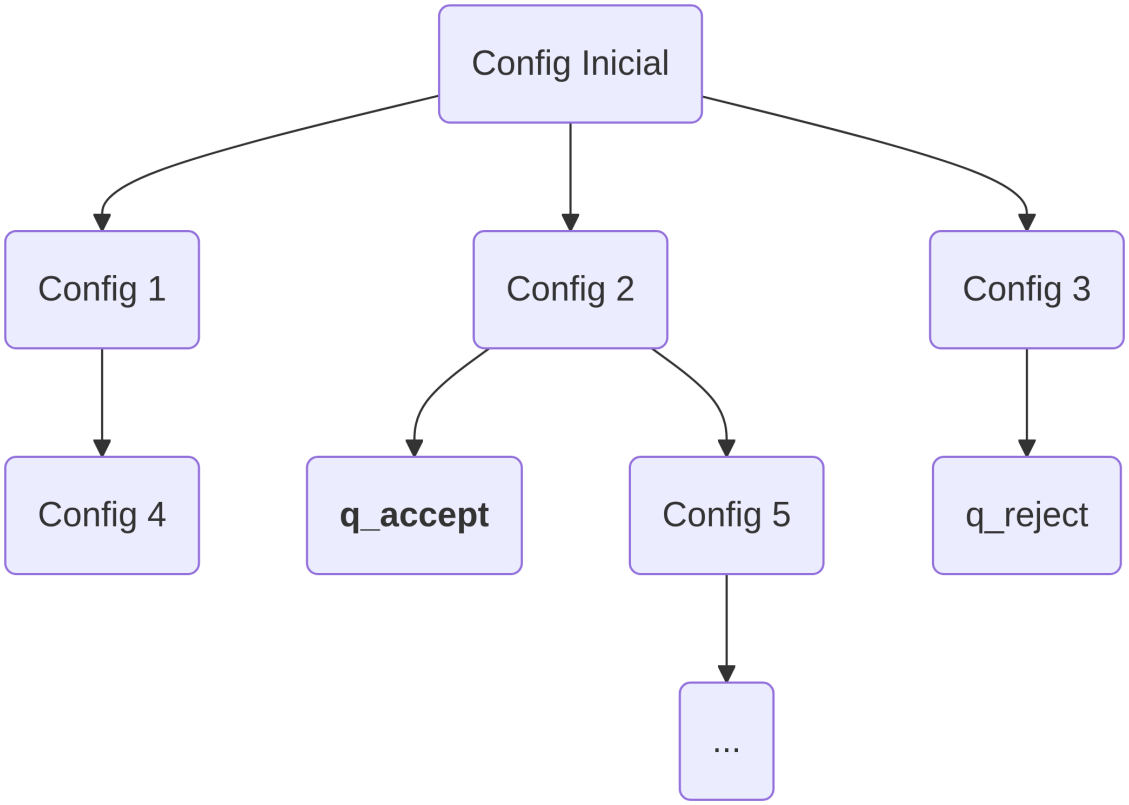

Figure 2: Árvore de computação de uma NTM. A DTM explora esta árvore nível por nível.

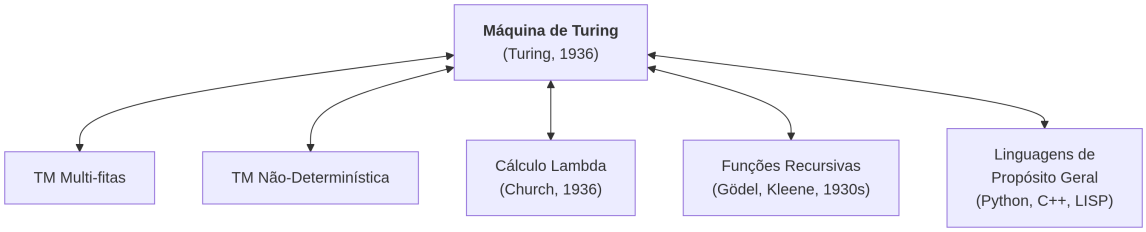

Figure 3: Nuvem de equivalência de modelos computacionais.

> **Dica**
>
> Pense em como você poderia "dobrar" a fita duplamente infinita em uma fita de sentido único. Você pode precisar de um marcador especial para o "ponto de dobra".

3: Conexão com Python

Você escreve a seguinte função em Python:

```python
def decide_linguagem_L(w: str) -> bool:
    # ... um código complexo aqui ...
    # No entanto, é garantido que este código não tem loops infinitos.
    # Ele sempre retorna True ou False para qualquer string 'w'.
    return len(w) % 2 == 0 # Exemplo simples de uma computação que para
```

Com base na Tese de Church-Turing, o que a existência desta função nos permite concluir sobre a linguagem $L = \{w \mid \text{decide\_linguagem\_L(w) retorna True}\}$? A linguagem L é decidível? Justifique sua resposta.

Referências Bibliográficas

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