Máquinas de Turing – Introdução

IBM1088 Linguagens Formais e Teoria da Computação

Evandro Eduardo Seron Ruiz
evandro@usp.br

Departmento de Computação e Matemática
FFCLRP
Universidade de São Paulo

Homenagem do dia

William Shakespeare

Hell is empty and all the devils are here.

WS: Dramaturgo, Inglês, 1564–1616.

Próximas etapas

- Máquina de Turing (Esta!)
- Linguagens sensíveis ao contexto
- Linguagens Recursivamente enumeráveis
- O problema da parada
- Complexidade de tempo: classes P, NP e NP-completos

Conteúdo

Modelo genérico

- Início do século XX, na CC, é marcado pela procura por um modelo computacional suficientemente genérico
- Nesta época, computador era uma pessoa que calculava
- Busca: Modelo ideal para processamento matemático geral
- Objetivo: encontrar a 'Máquina Universal'
- Destacamos 3 matemáticos importantes:
  1. David Hilbert (1862–1943)
  2. Kurt Friedrich Gödel (1906–1978)
  3. Alan Turing (1912–1954)

Hilbert

- 1928 formulou o 'Problema da decisão'
- Problema da lógica simbólica
- Todas asserções matemáticas deveriam ser deduzidas
- Encontrar um algoritmo genérico para determinar se um dado enunciado da lógica de primeira ordem pode ser provado

Entscheidungsproblem

Existe um procedimento efetivo que demonstre, em tempo finito, que uma proposição matemática possa ou não ser provada a partir de um conjunto de axiomas?

Os 3 desejos de Hilbert

1. Consistência O conjunto de axiomas deveria ser consistemente através de provas
2. Completude Todas as proposições matemáticas deveriam ser deduzíveis através dos axiomas
3. Decidibilidade Deveria existir um procedimento claramente formulado tal que, dada qualquer proposição matemática, seja possível afirmar, em tempo finito, se aquela proposição é derivada dos axiomas.

Hilbert em LFA

- Estes três desejos podem ser formulados sintaticamente
- Se dependessem da semântica, seriam mais difíceis de atingir

Hilbert em LFA

- Estes três desejos podem ser formulados sintaticamente
- Se dependessem da semântica, seriam mais difíceis de atingir

Consistência

Diz-se de um sistema o qual nunca se pode provar ambas as proposições P e $\neg P$.

Hilbert em LFA

- Estes três desejos podem ser formulados sintaticamente
- Se dependessem da semântica, seriam mais difíceis de atingir

Consistência

Diz-se de um sistema o qual nunca se pode provar ambas as proposições P e $\neg P$.

Completude

Um sistema completo é aquele para qual sempre podemos provar P ou $\neg P$ para qualquer proposição P expressa pelo sistema.

Gödel: Primeiro teorema da incompletude, 1931

Primeiro teorema da incompletude

Em qualquer teoria axiomática verdadeira (consistente) e rica o suficiente para expressar e provar proposições aritméticas, pode-se construir uma proposição G tal que não se pode provar nem G nem ao menos $\neg G$ através de seus axiomas. Assim o sistema é incompleto.

- Gödel mostra que asserções e relações matemática podem ser codificadas como comandos aritméticos
- Pode-se codificar uma proposição G verdadeira a qual não se pode depois provar se, de fato, é verdadeira

Turing e o problema aberto

- Gödel deixou aberto o problema
- Gödel provou que um sistema aritmético consistente poderia deixar asserções aritméticas sem provas
- Mas Gödel não afirmou que não poderia existir um procedimento de decisão 'computacional' efetivo que poderia, num tempo finito, revelar se uma proposição P poderia ou não ser provada

Turing e sua máquina

- Em 1936 Turing mostrou uma notação matematicamente efetiva baseada na 'Máquina de Turing' (MT).
- Criou a 'Máquina Universal' matematicamente factível mas...
- Mostrou também que existem problemas que não podem ser computados usando-se estas máquinas
- Provou a impossibilidade de escrever um programa que determina se uma outra MT para dada uma entrada qualquer

Problema da Parada

Dadas uma descrição de um programa e uma entrada finita, decida se o programa termina de rodar ou rodará indefinidamente.

Conseqüências da MT

- Dado o famoso 'Problema da Parada' das MT
- Turing mostrou que o Problema da Decisão proposto por Hilbert não tinha solução

Máquina de Turing

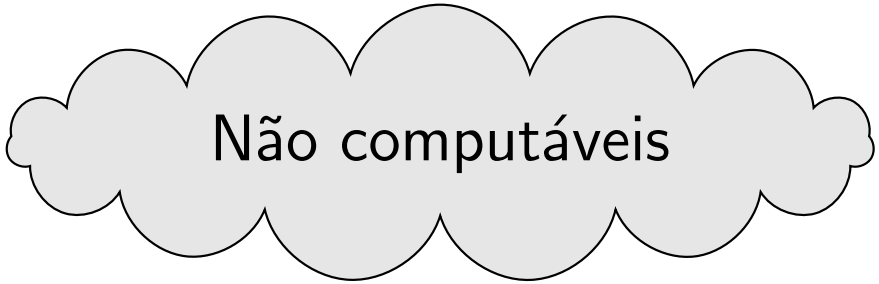
- Formalização de um algoritmo, uma função computável
- Seqüência finita de instruções que podem ser realizadas mecanicamente em tempo finito

Hipótese de Church, Alonzo (1936)

Qualquer função computável pode ser realizada pela Máquina de Turing

Segundo esta hipótese, a MT é o dispositivo de computação mais genérico.

recordando. . . Hierarquia de Chomsky



Recursivamente enumeráveis	Máquina de Turing
Sensíveis ao contexto	Autômato linearmente limitado
Livres do contexto	Autômato com pilha
Regulares	Autômato de estado finito
Gramáticas	**Máquinas**

Linguagens Enumeráveis Recursivamente

- Linguagens do Tipo 0
- Linguagens Recursivamente Enumeráveis
- Para estas a MT para ou aceita uma palavra que pertença a uma linguagem mas...
- Parar e rejeitar uma cadeia que não pertença a linguagem

Linguagens recursivas – Dedidíveis

São as linguagens para as quais a MT sempre para, aceitando ou não a palavra

Este tipo de linguagem não foi definido no momento da criação da Hierarquia de Chomsky

Volta ao Problema da Parada

Parada e Recursividade

O Problema da Parada é recursivamente enumerável mas não é recursivo

Linguagens sensíveis ao contexto

- Linguagens do Tipo 1
- Subconjunto das linguagens recursivamente enumeráveis
- Incluem praticamente todas as linguagens que podem ser decididas através da MT
- Decidíveis = MT sempre para, aceitando ou não

Existe?

A pergunta de Turing

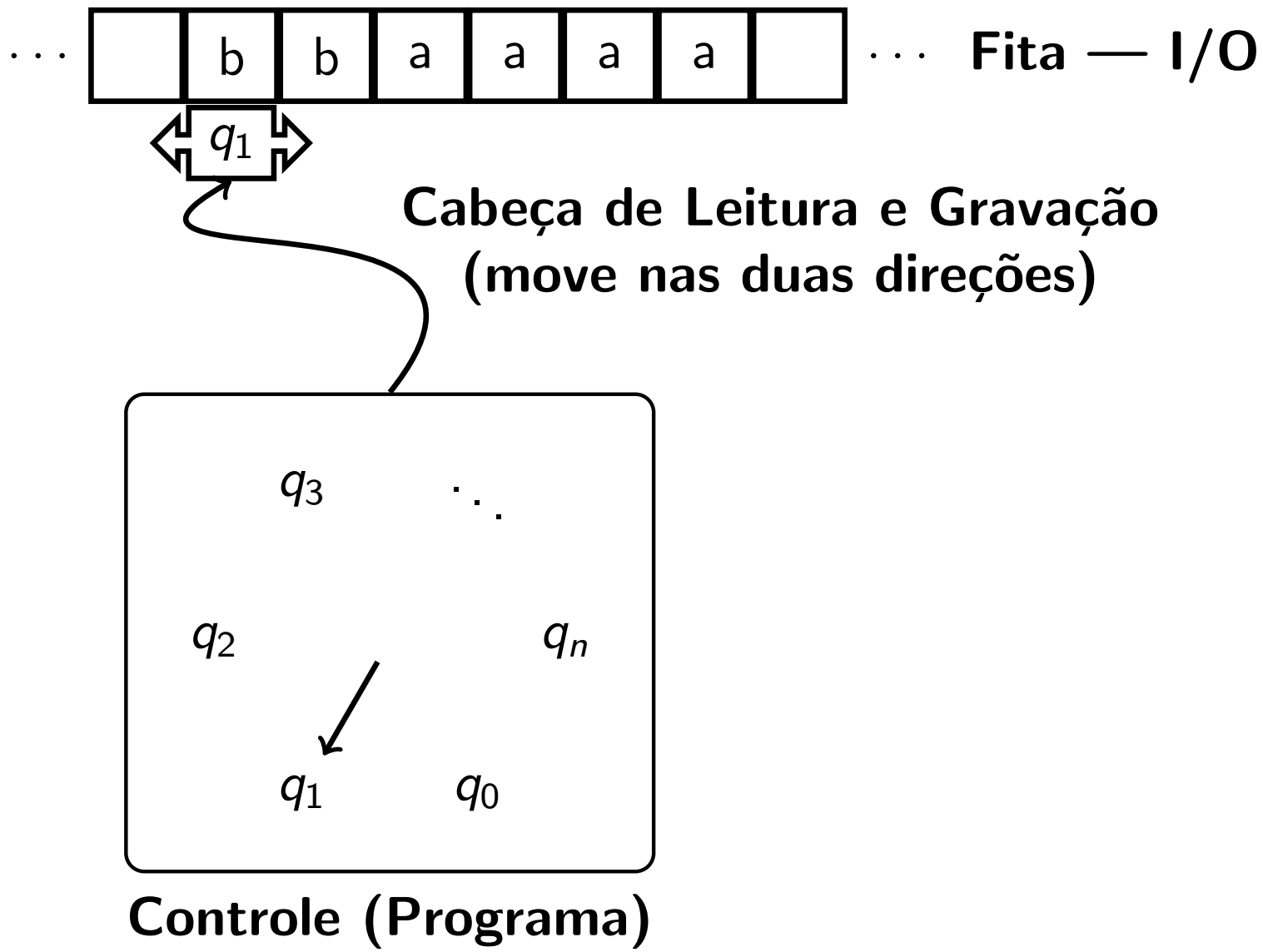
Existe uma máquina (hipotética) que, dada uma pergunta ela responde se existe a resposta?

- Tese de Turing, em termos simples
- É quase que uma reformulação do problema de Hilbert & Gödel

O modelo da MT

- Fita ilimitada: memória. Funciona como I/O
- Fita tem a cadeia de entrada e depois espaços em branco
- Controle lê e escreve sobre a fita
- Controle tem 2 estados:
  - Aceita a cadeia
  - Rejeita a cadeia
- Controle processa a fita até terminar a palavra
- Veja esquema a seguir...

Esquema da Máquina de Turing



Exemplo

Problema

Analisar uma MT M_1 para testar a pertinência da linguagem

$$L_1 = \{w\#w \mid w \in \{0,1\}^*\}$$

Condições:

- A entrada (palavra) pode ser muito longa
- Seja $s \in L_1$, $s = 011000\#011000$

Algoritmo para verificar $w\#w$

1. Faça uma varredura na cadeia de entrada para se assegurar que contenha somente um '#'. Caso contrário, rejeite
2. Faça um zigue-zague na fita para corresponder as posições nos dois lados de '#'. Marque-os a medida que correspondam
3. Depois de correspondidos todos os símbolos à esquerda de '#' verifique se ainda há algum símbolo sem correspondente. Caso haja, rejeite a cadeia, caso contrário aceite-a

Estratégia

Percorrer a fita movendo a cabeça para frente e para trás e analisar os símbolos das posições correspondentes.
Marcar os símbolos que são verificados

```
  |-------------|
0 1 1 0 0 0 # 0 1 1 0 0 0
x 1 1 0 0 0 # x 1 1 0 0 0
    |-------------|
x 1 1 0 0 0 # x 1 1 0 0 0
x x 1 0 0 0 # x x 1 0 0 0
      |-------------|
x x 1 0 0 0 # x x 1 0 0 0
x x x 0 0 0 # x x x 0 0 0
```

e assim segue. . .

Formalização da MT

Formalização da Máquina de Turing

Uma MT é uma 7-upla $M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$, para Q, Σ e Γ conjuntos finitos

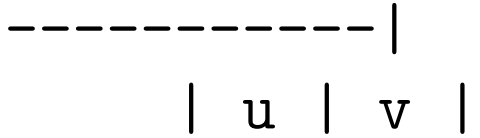
1. Q o conjunto dos estados
2. Σ o alfabeto de entrada (sem $\sqcup$)
3. Γ é o alfabeto da fita ($\sqcup \in \Gamma$ e $\Sigma \subseteq \Gamma$)
4. $\delta : Q \times \Gamma \rightarrow Q \times \{E, D\}$ é a função de transição
5. q_0 é o estado inicial
6. $q_A \in Q$ é o estado de aceitação
7. $q_R \in Q$ é o estado de rejeição, $q_A \neq q_R$

Detalhes

- A função de transição δ é o 'cérebro' da MT, o controle
- A MT não retrocede ao início da cadeia de entrada
- Durante a aplicação de δ sucessivos, ou seja, a função programa, estas variáveis podem mudar:
  - o estado atual
  - o conteúdo da fita
  - a posição da cabeça de I/O

Configuração

- É o conjunto de 3 dados, instâncias
- Dados:
  - q o estado
  - μ e v cadeias em Γ
- Descrevemos a configuração $\mu q v$
- Para q o estado atual e μv o conteúdo da fita, e
- A cabeça de I/O está sobre v



Ou ainda...

Dados: a cadeia: 1 0 1 1 q_x 0 1 1 1

Esta situação pode ser representada por:

```
--------|
1 0 1 1 0 1 1 1 _ ...
|--u--|---v-------|
```

Reafirmando, dado μqv a cabeça de I/O está sobre o início de v.

Movimento para esquerda

- De $\mu a \mathbf{q_i} b v$ para $\mu \mathbf{q_j} a c v$
- É equivalente a transição

$$\delta(q_i, b) = (q_j, c, E)$$

- A MT está no estado q_i sobre o símbolo b
- A MT escreve o símbolo c sobre b
- E vai para o estado q_j
- A cabeça move-se para a esquerda (E)

Movimento para a direita

- De $\mu a \mathbf{q_i} b v$ para $\mu a c \mathbf{q_j} v$
- É equivalente a transição

$$\delta(q_i, b) = (q_j, c, D)$$

- A MT está no estado q_i sobre o símbolo b
- A MT escreve o símbolo c sobre b
- E vai para o estado q_j
- A cabeça move-se para a direita (D)

Caso especial: extremidade da fita

- As MT com fita limitada não permitem acesso à esquerda do símbolo delimitador de início de cadeia '$<$'
- Nem acesso ao símbolo delimitador de final de cadeia '$>$'
- Para as MT sem limitação no tamanho da fita veremos os casos especiais mais adiante

Aceitação ou rejeição

A MT para quando está nos estados aceita ou rejeita.

$$\delta : Q' \times \Gamma \rightarrow Q \times \Gamma \times \{E, D\}$$

para

$$Q' = Q - \{q_A \text{ ou } q_R\}$$

A coleção de cadeias que M aceita é a linguagem M, ou ainda, a linguagem reconhecida por M.

Turing-reconhecível

Linguagens recursivamente enumeráveis

Chama-se de uma linguagem Turing-reconhecível se alguma MT a reconhece. Ou ainda...
São as Linguagens recursivamente enumeráveis.

Nas linguagens recursivamente enumeráveis a MT pode parar num estado não final ou entrar em loop.

Turing-decidível

Linguagens recursivas

Chama-se de uma linguagem Turing-decidível, ou simplesmente dedidível, se alguma MT a decide. Ou ainda...
São as Linguagens recursivas.

Turing-decidível

Linguagens recursivas

Chama-se de uma linguagem Turing-decidível, ou simplesmente dedidível, se alguma MT a decide. Ou ainda...
São as Linguagens recursivas.

Toda linguagem decidível é Turing-reconhecível. O inverso não se aplica.

Decidível: o autômato para qualquer que seja a cadeia de entrada.
Para porque aceitou ou porque rejeitou a cadeia.
Nas linguagens recursivas a MT para em qualquer estado não final se não aceita a cadeia.

Finalizando...

Caros estudantes,

Leiam os livros texto e façam os exercícios sugeridos.