Decidibilidade

Esdras Lins Bispo Jr. bispojr@ufg.br

Teoria da Computação Bacharelado em Ciência da Computação

06 de junho de 2016





Plano de Aula

- Pensamento
- 2 Revisão
 - Definição de algoritmo
 - Terminologia para descrever MTs
- Oecidibilidade





Sumário

- Pensamento
- 2 Revisão
 - Definição de algoritmo
 - Terminologia para descrever MTs
- 3 Decidibilidade





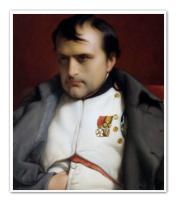
Pensamento







Pensamento



Frase

Nada é mais difícil e, portanto, tão precioso, do que ser capaz de decidir.

Quem?

Napoleão Bonaparte (1769-1821) Imperador francês.





Sumário

- Pensamento
- 2 Revisão
 - Definição de algoritmo
 - Terminologia para descrever MTs
- 3 Decidibilidade







Contribuição

Mostrou, em 1970, que não existe algoritmo para se testar se um polinômio tem raízes inteiras.

Quem?

Yuri Matijasevich (1947-) Cientista da computação e matemático russo.





Noção intuitiva de algoritmos de igual a algoritmos de máquina de Turing

FIGURA 3.22

A Tese de Church-Turing

Conclusão

Existem problemas que são algoritmicamente insolúveis.





Contexto

 $D = \{p \mid p \text{ \'e um polinômio com uma raiz inteira}\}$

Problema

O conjunto D é decidível?

Resposta

Não é decidível. Mas é Turing-reconhecível.





Problema análogo

 $D_1 = \{p \mid p \text{ \'e um polinômio sobre } x \text{ com uma raiz inteira}\}$

$\mathsf{MT}\ M_1$ que reconhece D_1

 M_1 = "A entrada é um polinômio p sobre a variável x.

• Calcule o valor de p com x substituída sucessivamente pelos valores $0, 1, -1, 2, -2, 3, -3, \dots$

Se em algum ponto o valor do polinômio resulta em 0, aceite.

Considerações

 M_1 reconhece D_1 , mas não a decide.





Resultado obtido por Matijasevich

É possíve \mid construir um decisor para D_1 . Mas não para D.

Justificativa

É possível obter um limitante para polinômios de uma única variável. Porém, Matijasevich provou ser impossível calcular tais limitantes para polinômios multivariáveis.

Limitante para polinômios de uma única variável

$$\pm k \frac{c_{max}}{c_1}$$

em que

- k é o número de termos do polinômio,
- c_{max} é o coeficiente com maior valor absoluto, e
- c₁ é o coeficiente do termo de mais alta ordem.



Terminologia para descrever MTs

Níveis de descrição

- Descrição formal: esmiúça todos os elementos da 7-upla, conforme definição;
- Descrição de implementação: descreve a forma pela qual a MT move a sua cabeça e a forma como ela armazena os dados na fita:
- Descrição de alto nível: neste nível não precisamos mencionar como a máquina administra a sua fita ou sua cabeça de leitura-escrita.





Exemplo

Seja A a linguagem consistindo em todas as cadeias representando grafos não-direcionados que são conexos. Logo:

 $A = \{\langle G \rangle | G \text{ \'e um grafo n\~ao-direcionado conexo}\}$

Descrição de alto nível

 $M = \text{``Sobre a entrada } \langle G \rangle$, a codificação de um grafo G:

- 1 Selecione o primeiro nó de G e marque-o.
- Repita o seguinte estágio até que nenhum novo nó seja marcado:
 - Para cada nó em G, marque-o se ele está ligado por uma aresta a um nó que já está marcado.
- Faça uma varredura em todos os nós de G para determinar se eles estão todos marcados. Se eles estão, aceite; caso contrário, rejeite".





Exemplo

Pergunta

Como seria a descrição de M no nível de implementação?





Sumário

- Pensamento
- 2 Revisão
 - Definição de algoritmo
 - Terminologia para descrever MTs
- Opecidibilidade





Introdução

Propósitos da Teoria da Computação

- Conhecer o poder dos algoritmos;
- Explorar os limites da solubilidade algorítmica;
- Identificar algoritmos insolúveis.





Introdução

Propósitos da Teoria da Computação

- Conhecer o poder dos algoritmos;
- Explorar os limites da solubilidade algorítmica;
- Identificar algoritmos insolúveis.

Por que devemos estudar insolubilidade?

- Relaxamento dos requisitos;
- Conhecimento das limitações dos modelos computacionais.





Linguagens Decidíveis

Exemplos de Linguagens Decidíveis

São úteis porque

- Algumas linguagens decidíveis estão associadas a aplicações;
- Algumas linguagens aparentemente triviais não são decidíveis.





Linguagens Decidíveis

Exemplos de Linguagens Decidíveis

São úteis porque

- Algumas linguagens decidíveis estão associadas a aplicações;
- Algumas linguagens aparentemente triviais não são decidíveis.

Problema da aceitação

Dado um modelo computacional MC e uma cadeia de entrada ω , identificar se MC aceita ω .





Problema da aceitação para AFDs

Dado um AFD B e uma cadeia de entrada ω , identificar se B aceita ω .





Problema da aceitação para AFDs

Dado um AFD B e uma cadeia de entrada ω , identificar se B aceita ω .

Problema

 $A_{AFD} = \{\langle B, \omega \rangle \mid B \text{ \'e um AFD que aceita a cadeia de entrada } \omega \}$





Problema da aceitação para AFDs

Dado um AFD B e uma cadeia de entrada ω , identificar se B aceita ω .

Problema

 $A_{AFD} = \{\langle B, \omega \rangle \mid B \text{ \'e um AFD que aceita a cadeia de entrada } \omega \}$

Estratégia de Resolução

Resolver o problema da aceitação para AFDs é decidir se $\omega \in \mathcal{A}_{AFD}$.





Teorema 4.1

 A_{AFD} é uma linguagem decidível.





Teorema 4.1

A_{AFD} é uma linguagem decidível.

Ideia da Prova

M= "Sobre a entrada $\langle B,\omega\rangle$, em que B é um AFD, e ω , uma cadeia:

- **1** Simule B sobre a entrada ω ;
- Se a simulação termina em um estado de aceitação, aceite. Senão, rejeite."





Detalhes de implementação

- A entrada $\langle B, \omega \rangle$ representa um AFD e uma cadeia;
 - Uma representação razoável de B seria uma lista de seus cinco componentes: $Q, \Sigma, \delta, q_0 \in F$;
 - M simula B de forma que M aceita se B estiver em um estado final, e rejeita, caso contrário.





Problema da aceitação para AFNs

Dado um AFN B e uma cadeia de entrada ω , identificar se B aceita ω .





Problema da aceitação para AFNs

Dado um AFN B e uma cadeia de entrada ω , identificar se B aceita ω .

Problema

 $A_{AFN} = \{\langle B, \omega \rangle \mid B \text{ \'e um AFN que aceita a cadeia de entrada } \omega \}$





Problema da aceitação para AFNs

Dado um AFN B e uma cadeia de entrada ω , identificar se B aceita ω .

Problema

 $A_{AFN} = \{\langle B, \omega \rangle \mid B \text{ \'e um AFN que aceita a cadeia de entrada } \omega\}$

Estratégia de Resolução

Decidir se $\langle B, \omega \rangle \in A_{AFN}$.





Teorema 4.2

A_{AFN} é uma linguagem decidível.





Teorema 4.2

A_{AFN} é uma linguagem decidível.

Prova

N= "Sobre a entrada $\langle B,\omega\rangle$, em que B é um AFN, e ω , uma cadeia:

- Converta AFN B para um AFD equivalente C, usando o procedimento para essa conversão dado no Teorema 1.39;
- 2 Rode a MT M do Teorema 4.1 sobre a cadeia $\langle C, \omega \rangle$;
- 3 Se M aceita, aceite. Caso contrário, rejeite."





Problema da Vacuidade de uma Linguagem

Descrição

Dada uma linguagem L, identificar se $L = \emptyset$.

Problema aplicado a AFDs

$$V_{AFD} = \{ \langle A \rangle \mid A \text{ \'e um AFD e } L(A) = \emptyset \}$$

Estratégia de Resolução

Decidir se $\langle A \rangle \in V_{AFD}$.





Problema da Vacuidade de uma Linguagem

Teorema 4.4

 V_{AFD} é uma linguagem decidível.





Problema da Vacuidade de uma Linguagem

Teorema 4.4

 V_{AFD} é uma linguagem decidível.

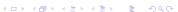
Prova

A seguinte MT T decide V_{AFD} .

T = "Sobre a entrada $\langle A \rangle$, em que A é uma AFD:

- Marque o estado inicial de A;
- Repita até que nenhum estado novo venha a ser marcado;
 - Marque qualquer estado que tenha uma transição chegando nele a partir de qualquer estado que já está marcado.
- Se nenhum estado final estiver marcado, aceite. Caso contrário, rejeite."





Bônus (0,5 pt)

Desafio

- Problema 4.12:
 - Seja $A = \{\langle R, S \rangle \mid R \in S \text{ são expressões regulares e L}(R) \subseteq L(S)\}$. Mostre que A é decidível.
- Candidaturas até amanhã (07 de junho, 09h30);
- Apresentação e resposta por escrito → Segunda (14 de junho, 11h30);
- 20 minutos de apresentação.

Candidato

???







Decidibilidade

Esdras Lins Bispo Jr. bispojr@ufg.br

Teoria da Computação Bacharelado em Ciência da Computação

06 de junho de 2016



