Definição de Algoritmo

Esdras Lins Bispo Jr.
bispojr@ufg.br

Teoria da Computação
Bacharelado em Ciência da Computação

10 de abril de 2018

Plano de Aula

1. Revisão
   - Máquina de Turing
   - Variantes da MT
2. Definição de algoritmo

UFG
Regional Jataí

Sumário

1. Revisão
   - Máquina de Turing
   - Variantes da MT
2. Definição de algoritmo

Problema

Problema 3.15 (a)

Mostre que a coleção de linguagens decidíveis é fechada sob a operação de união.

3.15 **(a)** Para quaisquer duas linguagens decidíveis L_1 e L_2, sejam M_1 e M_2 as MTs que as decidem. Construimos uma MT M' que decide a união de L_1 e L_2:

"Sobre a entrada w:

1. Rode M_1 sobre w. Se ela aceita, *aceite*.
2. Rode M_2 sobre w. Se ela aceita, *aceite*. Caso contrário, *rejeite*."

M' aceita w se M_1 ou M_2 a aceita. Se ambas rejeitam, M' rejeita.

MT Multifita

Definição

Uma **máquina de Turing multifita** é como uma máquina de Turing comum com várias fitas:

- cada fita tem sua própria cabeça de leitura e escrita;
- a configuração inicial consiste da cadeia de entrada aparecer sobre a fita 1, e as outras iniciar em branco;
- a função de transição permite ler, escrever e mover as cabeças em algumas ou em todas as fitas simultaneamente

$$\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{E, D, P\}^k$$

em que k é o número de fitas.

Exemplo

$\delta(q_i, a_1, \ldots, a_k) = (q_j, b_1, \ldots, b_k, P, D, \ldots, E)$

MT Multifita

Teorema

Toda máquina de Turing multifita tem uma máquina de Turing de uma única fita que lhe é equivalente.

MT Multifita

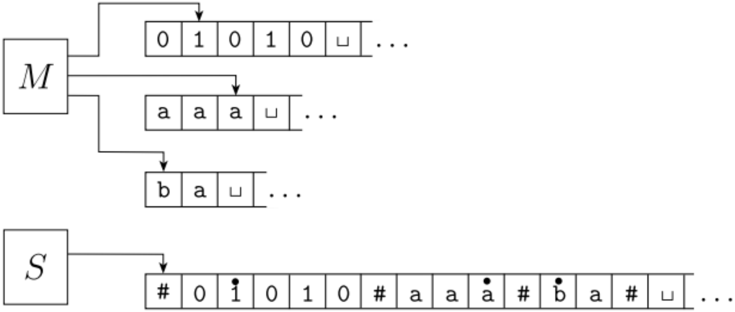

FIGURA 3.14
Representando três fitas com apenas uma

Regional Jataí

MT Multifita

$S =$ "Sobre a entrada $w = w_1 \cdots w_n$:

1. Primeiro S ponha sua fita no formato que representa todas as k fitas de M. A fita formatada contém

$$\#\dot{w}_1 w_2 \cdots w_n \#\dot{\sqcup}\#\dot{\sqcup}\# \cdots \#$$

2. Para simular um único movimento, S faz uma varredura na sua fita desde o primeiro #, que marca a extremidade esquerda, até o $(k+1)$-ésimo #, que marca a extremidade direita, de modo a determinar os símbolos sob as cabeças virtuais. Então S faz uma segunda passagem para atualizar as fitas conforme a maneira pela qual a função de transição de M estabelece.

UFG
Regional Jataí

MT Multifita

3. Se em algum ponto S move uma das cabeças virtuais sobre um **#**, essa ação significa que M moveu a cabeça correspondente para a parte previamente não-lida em branco daquela fita. Portanto, S escreve um símbolo em branco nessa célula da fita e desloca o conteúdo da fita, a partir dessa célula até o **#** mais à direita, uma posição para a direita. Então ela continua a simulação tal qual anteriormente."

MT Multifita

Teorema

Toda máquina de Turing multifita tem uma máquina de Turing de uma única fita que lhe é equivalente.

Corolário

Uma linguagem é Turing-reconhecível se e somente se alguma máquina de Turing multifita a reconhece.

MT Multifita

PROVA Uma linguagem Turing-reconhecível é reconhecida por uma máquina de Turing comum (com uma única fita), o que é um caso especial de uma máquina de Turing multifita. Isso prova uma direção desse corolário. A outra direção segue do Teorema 3.13.

Sumário

Problema

Problema 3.16 (b)

Mostre que a coleção de linguagens Turing-reconhecíveis é fechada sob a operação de concatenação.

Problema

Prova: Sejam duas linguagens Turing-reconhecíveis (TR) quaisquer A e B.

Problema

Prova: Sejam duas linguagens Turing-reconhecíveis (TR) quaisquer A e B. Sejam M_A e M_B as duas máquinas de Turing (MT) que reconhecem A e B, respectivamente

Problema

Prova: Sejam duas linguagens Turing-reconhecíveis (TR) quaisquer A e B. Sejam M_A e M_B as duas máquinas de Turing (MT) que reconhecem A e B, respectivamente (pois se uma linguagem é Turing-reconhecível, então uma MT a reconhece).

Problema

Prova: Sejam duas linguagens Turing-reconhecíveis (TR) quaisquer A e B. Sejam M_A e M_B as duas máquinas de Turing (MT) que reconhecem A e B, respectivamente (pois se uma linguagem é Turing-reconhecível, então uma MT a reconhece). Iremos construir uma MT não-determinística (MTN) M_{aux}, a partir de M_A e M_B, que reconhece $A \circ B$.

Problema

Prova: Sejam duas linguagens Turing-reconhecíveis (TR) quaisquer A e B. Sejam M_A e M_B as duas máquinas de Turing (MT) que reconhecem A e B, respectivamente (pois se uma linguagem é Turing-reconhecível, então uma MT a reconhece). Iremos construir uma MT não-determinística (MTN) M_{aux}, a partir de M_A e M_B, que reconhece $A \circ B$. A descrição de M_{aux} é dada a seguir:

Problema

Prova: Sejam duas linguagens Turing-reconhecíveis (TR) quaisquer A e B. Sejam M_A e M_B as duas máquinas de Turing (MT) que reconhecem A e B, respectivamente (pois se uma linguagem é Turing-reconhecível, então uma MT a reconhece). Iremos construir uma MT não-determinística (MTN) M_{aux}, a partir de M_A e M_B, que reconhece $A \circ B$. A descrição de M_{aux} é dada a seguir:
M_{aux} = "Sobre a entrada ω, faça:

UFG
Regional Jataí

Problema

Prova: Sejam duas linguagens Turing-reconhecíveis (TR) quaisquer A e B. Sejam M_A e M_B as duas máquinas de Turing (MT) que reconhecem A e B, respectivamente (pois se uma linguagem é Turing-reconhecível, então uma MT a reconhece). Iremos construir uma MT não-determinística (MTN) M_{aux}, a partir de M_A e M_B, que reconhece $A \circ B$. A descrição de M_{aux} é dada a seguir:
M_{aux} = "Sobre a entrada ω, faça:

1. Corte, não deterministicamente, ω em duas cadeias (i.e. $\omega = \omega_1 \circ \omega_2$).

Problema

Prova: Sejam duas linguagens Turing-reconhecíveis (TR) quaisquer A e B. Sejam M_A e M_B as duas máquinas de Turing (MT) que reconhecem A e B, respectivamente (pois se uma linguagem é Turing-reconhecível, então uma MT a reconhece). Iremos construir uma MT não-determinística (MTN) M_{aux}, a partir de M_A e M_B, que reconhece $A \circ B$. A descrição de M_{aux} é dada a seguir:
M_{aux} = "Sobre a entrada ω, faça:

1. Corte, não deterministicamente, ω em duas cadeias (i.e. $\omega = \omega_1 \circ \omega_2$).
2. Rode M_A sobre ω_1. Se M_A rejeita, *rejeite*.

Problema

Prova: Sejam duas linguagens Turing-reconhecíveis (TR) quaisquer A e B. Sejam M_A e M_B as duas máquinas de Turing (MT) que reconhecem A e B, respectivamente (pois se uma linguagem é Turing-reconhecível, então uma MT a reconhece). Iremos construir uma MT não-determinística (MTN) M_{aux}, a partir de M_A e M_B, que reconhece $A \circ B$. A descrição de M_{aux} é dada a seguir:
M_{aux} = "Sobre a entrada ω, faça:

1. Corte, não deterministicamente, ω em duas cadeias (i.e. $\omega = \omega_1 \circ \omega_2$).
2. Rode M_A sobre ω_1. Se M_A rejeita, *rejeite*.
3. Rode M_B sobre ω_2. Se M_B aceita, *aceite*.

UFG
Regional Jataí

Problema

Prova: Sejam duas linguagens Turing-reconhecíveis (TR) quaisquer A e B. Sejam M_A e M_B as duas máquinas de Turing (MT) que reconhecem A e B, respectivamente (pois se uma linguagem é Turing-reconhecível, então uma MT a reconhece). Iremos construir uma MT não-determinística (MTN) M_{aux}, a partir de M_A e M_B, que reconhece $A \circ B$. A descrição de M_{aux} é dada a seguir:

M_{aux} = "Sobre a entrada ω, faça:

1. Corte, não deterministicamente, ω em duas cadeias (i.e. $\omega = \omega_1 \circ \omega_2$).
2. Rode M_A sobre ω_1. Se M_A rejeita, *rejeite*.
3. Rode M_B sobre ω_2. Se M_B aceita, *aceite*.
4. *Rejeite*".

UFG
Regional Jataí

Problema

Prova: Sejam duas linguagens Turing-reconhecíveis (TR) quaisquer A e B. Sejam M_A e M_B as duas máquinas de Turing (MT) que reconhecem A e B, respectivamente (pois se uma linguagem é Turing-reconhecível, então uma MT a reconhece). Iremos construir uma MT não-determinística (MTN) M_{aux}, a partir de M_A e M_B, que reconhece $A \circ B$. A descrição de M_{aux} é dada a seguir:
M_{aux} = "Sobre a entrada ω, faça:

1. Corte, não deterministicamente, ω em duas cadeias (i.e. $\omega = \omega_1 \circ \omega_2$).
2. Rode M_A sobre ω_1. Se M_A rejeita, *rejeite*.
3. Rode M_B sobre ω_2. Se M_B aceita, *aceite*.
4. *Rejeite*".

Como é possível construir M_{aux}, então $A \circ B$ é TR

UFG
Regional Jataí

Problema

Prova: Sejam duas linguagens Turing-reconhecíveis (TR) quaisquer A e B. Sejam M_A e M_B as duas máquinas de Turing (MT) que reconhecem A e B, respectivamente (pois se uma linguagem é Turing-reconhecível, então uma MT a reconhece). Iremos construir uma MT não-determinística (MTN) M_{aux}, a partir de M_A e M_B, que reconhece $A \circ B$. A descrição de M_{aux} é dada a seguir:
M_{aux} = "Sobre a entrada ω, faça:

1. Corte, não deterministicamente, ω em duas cadeias (i.e. $\omega = \omega_1 \circ \omega_2$).
2. Rode M_A sobre ω_1. Se M_A rejeita, *rejeite*.
3. Rode M_B sobre ω_2. Se M_B aceita, *aceite*.
4. *Rejeite*".

Como é possível construir M_{aux}, então $A \circ B$ é TR (pois toda MTN tem uma MT equivalente).

Problema

Prova: Sejam duas linguagens Turing-reconhecíveis (TR) quaisquer A e B. Sejam M_A e M_B as duas máquinas de Turing (MT) que reconhecem A e B, respectivamente (pois se uma linguagem é Turing-reconhecível, então uma MT a reconhece). Iremos construir uma MT não-determinística (MTN) M_{aux}, a partir de M_A e M_B, que reconhece $A \circ B$. A descrição de M_{aux} é dada a seguir:
M_{aux} = "Sobre a entrada ω, faça:

1. Corte, não deterministicamente, ω em duas cadeias (i.e. $\omega = \omega_1 \circ \omega_2$).
2. Rode M_A sobre ω_1. Se M_A rejeita, *rejeite*.
3. Rode M_B sobre ω_2. Se M_B aceita, *aceite*.
4. *Rejeite*".

Como é possível construir M_{aux}, então $A \circ B$ é TR (pois toda MTN tem uma MT equivalente). Logo, a classe de linguagens Turing-reconhecíveis é fechada sob a operação de concatenação ∎

Definição de algoritmo

Contribuição

Apresentou uma noção do que seria um algoritmo no Congresso Internacional de Matemáticos em Paris, no ano de 1900.

Quem?

David Hilbert (1862-1943)
Matemático alemão.

Polinômio

Definições

Um **polinômio** é uma soma de termos. Um **termo** é um produto de variáveis e uma constante chamada de **coeficiente**.

Polinômio

Definições

Um **polinômio** é uma soma de termos. Um **termo** é um produto de variáveis e uma constante chamada de **coeficiente**.

Exemplo: Termo

$6 \cdot x \cdot x \cdot y \cdot z \cdot z \cdot z = 6x^2yz^3$

Polinômio

Definições

Um **polinômio** é uma soma de termos. Um **termo** é um produto de variáveis e uma constante chamada de **coeficiente**.

Exemplo: Termo

$6 \cdot x \cdot x \cdot y \cdot z \cdot z \cdot z = 6x^2yz^3$

Exemplo: Polinômio

$6x^2yz^3 + 3xy^2 - 10$

UFG
Regional Jataí

Polinômio

Definições

Uma **raiz** de um polinômio é uma atribuição de valores às suas variáveis de modo que o valor do mesmo seja 0. Chamamos de **raiz inteira** aquela em todos os valores atribuídos são valores inteiros.

Polinômio

Definições

Uma **raiz** de um polinômio é uma atribuição de valores às suas variáveis de modo que o valor do mesmo seja 0. Chamamos de **raiz inteira** aquela em todos os valores atribuídos são valores inteiros.

Exemplo: Raiz

O polinômio $6x^3yz^2 + 3xy^2 - x^3 - 10$ tem uma raiz em $x = 5, y = 3$ e $z = 0$.

Polinômio

Definições

Uma **raiz** de um polinômio é uma atribuição de valores às suas variáveis de modo que o valor do mesmo seja 0. Chamamos de **raiz inteira** aquela em todos os valores atribuídos são valores inteiros.

Exemplo: Raiz

O polinômio $6x^3yz^2 + 3xy^2 - x^3 - 10$ tem uma raiz em $x = 5, y = 3$ e $z = 0$.

Exemplo: Raiz Inteira

A raiz do exemplo acima é uma raiz inteira.

UFG
Regional Jataí

Polinômio

Problema apresentado por Hilbert

É possível conceber um algoritmo que teste se um polinômio tem uma raiz inteira ou não?

Polinômio

Problema apresentado por Hilbert

É possível conceber um algoritmo que teste se um polinômio tem uma raiz inteira ou não?

Expressão utilizada por Hilbert

"Um processo com o qual ela possa ser determinada por um número finito de operações".

UFG
Regional Jataí

Polinômio

Problema apresentado por Hilbert

É possível conceber um algoritmo que teste se um polinômio tem uma raiz inteira ou não?

Expressão utilizada por Hilbert

"Um processo com o qual ela possa ser determinada por um número finito de operações".

Curioso

Não existe algoritmo que execute esta tarefa.

Definição de algoritmo

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.

UFG
Regional Jataí

Definição de Algoritmo

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

FIGURA 3.22
A Tese de Church–Turing

Definição de Algoritmo

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

FIGURA 3.22
A Tese de Church–Turing

Conclusão

Existem problemas que são algoritmicamente insolúveis.

Definição de Algoritmo

Contexto

$D = \{p \mid p$ é um polinômio com uma raiz inteira$\}$

Definição de Algoritmo

Contexto

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

Problema

O conjunto D é decidível?

Definição de Algoritmo

Contexto

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

Problema

O conjunto D é decidível?

Resposta

Não é decidível. Mas é Turing-reconhecível.

UFG
Regional Jataí

Definição de Algoritmo

Problema análogo

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

Definição de Algoritmo

Problema análogo

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

MT M_1 que reconhece D_1

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

1. Calcule o valor de p com x substituída sucessivamente pelos valores $0, 1, -1, 2, -2, 3, -3, \ldots$
   Se em algum ponto o valor do polinômio resulta em 0, *aceite*.

UFG
Regional Jataí

Definição de Algoritmo

Problema análogo

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

MT M_1 que reconhece D_1

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

1. Calcule o valor de p com x substituída sucessivamente pelos valores $0, 1, -1, 2, -2, 3, -3, \ldots$
   Se em algum ponto o valor do polinômio resulta em 0, *aceite*.

Considerações

M_1 reconhece D_1, mas não a decide.

Definição de Algoritmo

Resultado obtido por Matijasevich

É possível construir um decisor para D_1. Mas não para D.

Definição de Algoritmo

Resultado obtido por Matijasevich

É possível 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.

UFG
Regional Jataí

Definição de Algoritmo

Resultado obtido por Matijasevich

É possível 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
- c_1 é o coeficiente do termo de mais alta ordem.

Definição de Algoritmo

Esdras Lins Bispo Jr.
bispojr@ufg.br

Teoria da Computação
Bacharelado em Ciência da Computação

10 de abril de 2018