Problemas decidíveis

Luís Felipe

UFF

19 de Junho de 2023

Luís Felipe
19/06/23

A tese de Church-Turing

- Além dos modelos baseados na Máquina de Turing, outros pesquisadores desenvolveram outros modelos formais de computação.
  - Alonzo Church: λ-cálculo
  - Stephen Kleene: Funções μ-Recursivas
- Estes modelos são equivalentes entre si e equivalentes ao modelo da Máquina de Turing
  - A classe de funções computáveis definida por qualquer desses modelos é a mesma classe das funções computáveis por uma Máquina de Turing.
  - Isto sugere que Máquinas de Turing são efetivamente um limite superior natural para o que pode ser computado.
- Tese de Church-Turing (1936): Um problema de decisão, ou uma função, é algoritmicamente computável se, e somente se, existe uma MT que a decide, ou a computa.

Luís Felipe
19/06/23

Noção de Algoritmo

- Algoritmo: sequência finita de passos para realizar uma tarefa
- Receita de bolo; passo-a-passo
- O décimo problema de Hilbert (1900) dizia respeito a algoritmos.
  - Conceber um algoritmo que decide se um polinômio tem raiz inteira.
  - Da formulação, Hilbert parecia assumir que tal algoritmo existia.
  - Será que Hilbert estava certo?
  - Hoje sabemos que Hilbert estava errado e que esse problema é Insolúvel
  - A tese de Church-Turing definiu precisamente a noção de algoritmo e mais tarde (1970), foi mostrado que não existe algoritmo para testar se um polinômio tem raízes inteiras.

Luís Felipe

19/06/23

Décimo problema de Hilbert

- $D = \{p \mid p \text{ é um polinômio com raiz inteira}\}$
- O décimo problema de Hilbert questiona se D é decidível. Como dito, não é.
- Mas é Turing-reconhecível.
  - Ou seja, existe MT que a reconhece e não é decisor
- Como seria a MT que reconhece D?
  - Pensemos num caso mais simples (p só com x de variável): $D_1 = \{p \mid p \text{ é um polinômio em } x \text{ com uma raiz inteira}\}$
- $M_1 =$ Entrada é um polinômio em x
  1. Calcule o valor de p substituindo x por $0, 1, -1, 2, -2, \ldots$. Se p resultar em 0 para alguma substituição, aceite.
- Note que se p tiver raiz inteira, em algum momento a M_1 para aceitando. Caso contrário, vai rodar para sempre.
- Uma máquina M para D pode ser construída similarmente.

Luís Felipe

19/06/23

Linguagens Decidíveis - LR

- Problema 1: testar se um AFD específico aceita uma determinada cadeia.
  - Seja $\langle B, w \rangle$ uma string que contém a codificação de um AFD B e uma cadeia w.
  - Considere a linguagem $A_{AFD} = \{\langle B, w \rangle \mid B \text{ é um AFD que aceita a cadeia de entrada } w\}$
  - Esta linguagem contém todas as codificações de todos os AFDs juntamente com cadeias que os AFDs aceitam.
  - O problema de testar se um AFD aceita uma entrada w é o mesmo que o problema de se testar se $\langle B, w \rangle$ pertence à linguagem A_{AFD}.

- Vamos mostrar que A_{AFD} é decidível.

Luís Felipe
19/06/23

Linguagens Decidíveis - LR

Teorema. A_{AFD} é decidível.

Obs.: Ao tomar um par $\langle B, w \rangle$, saberei se ele pertence ou não a A_{AFD}.
Ideia: Simular os autômatos em uma máquina de Turing M.
M: Sobre a entrada $\langle B, w \rangle$, onde B é um AFD, e w, uma cadeia:

1. Simule B sobre a entrada w
2. Se a simulação termina em um estado de aceitação, aceite. Se ela termina em um estado de não-aceitação, rejeite.

- Primeiro M verifica se a entrada é de fato um AFD e uma cadeia. Se não for, rejeita.
- M realiza a simulação de B mantendo o registro do estado atual de B e da posição atual de B na entrada w escrevendo na fita.
- O estado inicial de B é q_0 e os estados e posição são atualizados conforme a função de transição.

Luís Felipe

19/06/23

Linguagens Decidíveis - LR

Considere $A_{AFN} = \{\langle B, w\rangle \mid B \text{ é um AFN que aceita } w\}$

Teorema. A_{AFN} é uma linguagem decidível.

Ideia: Construir uma MT N que converta o AFN em um AFD e use M (MT do slide anterior) como subrotina.

Luís Felipe

19/06/23

Linguagens Decidíveis - LR

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

Teorema. V_{AFD} é uma linguagem decidível.

Ideia: Construir uma MT T que utiliza um algoritmo de marcação de estados.

T: Sobre a entrada $\langle A \rangle$, onde A é um AFD:

1. Marque o estado inicial de A
2. Repita até que nenhum estado novo possa ser marcado:
   2.1 Marque todo estado que tenha seta entrando a partir de um estado já marcado.
3. Se nenhum estado de aceitação foi marcado, aceite. Senão, rejeite.

Luís Felipe

19/06/23

Linguagens Decidíveis - LLC

Considere
$A_{GLC} = \{\langle G, w\rangle \mid G \text{ é uma GLC que gera a cadeia } w\}$

Teorema. A_{GLC} é uma linguagem decidível.

Ideia I: Simular G em uma MT S que passe por todas as derivações de G e verifique que em alguma delas o produto final é w.

Isso não funciona para um decisor. Por que não funciona? Quantidade infinita de derivações (testando todas as derivações em 1 etapa, todas as derivações em 2 etapas, ...) a ser testada conduziria a uma MT reconhecedora e não decisora.

Luís Felipe

19/06/23

Como obter um decisor?

Ideia 2: Converter a GLC para a FNC, pois na FNC, qualquer derivação de w teria $2n - 1$ passos, onde $|w| = n$.
S = Sobre a entrada $\langle G, w \rangle$, onde G é uma GLC, e w é uma cadeia:

1. Converta G para a FNC
2. Liste todas as derivações com $2n - 1$ passos, onde $|w| = n$, exceto se $n = 0$. Neste caso, liste todas as derivações com 1 passo.
3. Se alguma derivação gera w, aceite. Senão, rejeite.

Luís Felipe
19/06/23

Linguagens Decidíveis - LLC

- Considere $V_{GLC} = \{\langle G \rangle \mid G \text{ é uma GLC } L(G) = \emptyset\}$

Teorema. V_{GLC} é uma linguagem decidível.

Ideia: Verificar se a variável inicial é capaz de gerar uma cadeia de terminais. Para verificar isso, o algoritmo verifica algo mais geral: ele determina, para cada variável, se ela pode gerar uma cadeia de terminais. O registro dessa informação é dado por uma marca sobre a variável.
R = Sobre a entrada $\langle G \rangle$, onde G é uma GLC:

1. Marque todos os símbolos terminais de G
2. Repita até que nenhuma nova variável possa ser marcada:
   2.1 Marque toda variável A tal que G tenha uma regra $A \to U_1 U_2 \ldots U_k$, e todo símbolo $U_i, i = \{1, \ldots, k\}$ já tenha sido marcado.
3. Se a variável inicial está marcada marcada, rejeite. Senão, aceite.

Luís Felipe

19/06/23

Linguagens Decidíveis - LLC

Teorema. Toda LLC é decidível.

Ideia: Toda LLC tem uma GLC, G, associada. Vamos usar a máquina S que simula G e tudo que S aceita, a nova MT M_G também aceita; tudo que S rejeita, M_G rejeita.

Luís Felipe

19/06/23