

Decidibilidade

Universidade Federal de Campina Grande – UFCG

Centro de Engenharia Elétrica e Informática – CEEI

Departamento de Sistemas e Computação – DSC

Professor: Andrey Brito

Período: 2023.2

Estudo dirigido – Compensação de aulas

- Faça uma máquina que decide a linguagem www com $w \in \{0,1\}^*$, em passos intermediários
 - Máquina 1: recebe www e transforma em $w'w'w'$ onde w' é a palavra w com o seu último símbolo marcado com um '
 - Máquina 2: recebe $w'w'w'$ e transforma em $w\#w\#w$
 - Máquina 3: recebe $w\#w\#w$ e aceita se as 3 partes são iguais
- Faça uma máquina que compute $f(x,y) = x/2+y$, deixando o resultado no início da fita e a fita toda limpa
- Cada uma das duas máquinas contará como 3 horas-aula e precisa ser entregue em papel A4 durante a chamada no dia 30/4
 - Ou enviado digitalizado por e-mail antes e entregue em papel na aula seguinte

Decidíveis vs. Reconhecíveis

- $L_1 = \{ p \mid p \text{ é um polinômio sobre uma variável, com raízes inteiras} \}$
 - Exemplo: $4.x^3 - 2.x^2 + x - 7$
- $L_2 = \{ p \mid p \text{ é um polinômio sobre múltiplas variáveis, com raízes inteiras} \}$
 - Exemplo: $6.x^3.y.z + 3.x.y^2 - x^3 - 10$
- $L_3 = \{ \langle G \rangle \mid G \text{ é um grafo não-direcionado conexo} \}$
 - Grafo conexo se todo nó pode ser atingido a partir de qualquer nó

É importante também entender bem o problema formulado na forma de conjunto.

Um pouco sobre representação dos problemas

1. Teste de pertinência a uma linguagem

- Problema ex.: teste se AFD B aceita a entrada w
- Em forma de linguagem:
 - Seja $A_{AFD} = \{ \langle B, w \rangle \mid B \text{ é um AFD que aceita } w \}$
 - A_{AFD} é um conjunto que contém todas as possíveis descrições de autômatos, associados a cada uma das palavras que ele aceita
- O objetivo da MT é receber como entrada uma codificação de AFD e responder se ele pertence à A_{AFD}
- Mostrar que o problema é decidível é equivalente a mostrar que a linguagem é decidível

Representação dos problemas

2. Teste de vacuidade (*emptiness*)

- Problema ex.: teste se AFD B aceita alguma entrada
- Em forma de linguagem:
 - $E_{AFD} = \{ \langle A \rangle \mid A \text{ é um AFD e } L(A) = \emptyset \}$
 - E_{AFD} contém todos os autômatos (i.e., todas as descrições) que não aceitam nenhuma palavra
- Como antes, o objetivo da MT é receber como entrada uma codificação de AFD e responder se ele pertence à E_{AFD}

E_{AFD} é decidível?

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

E_{AFD} é decidível?

- $E_{\text{AFD}} = \{ \langle A \rangle \mid A \text{ é um AFD e } L(A) = \emptyset \}$
- $\langle A \rangle$ é uma codificação do AFD A
 - Qual codificação exatamente não é importante agora
- Pense no diagrama de estados do AFD...
 - O que seria preciso para que A aceitasse alguma palavra?

E_{AFD} é decidível?

- $E_{AFD} = \{ \langle A \rangle \mid A \text{ é um AFD e } L(A) = \emptyset \}$
- $T =$ “Com entrada $\langle A \rangle$, onde A é um AFD:
 - Marcar o estado inicial de A
 - Repetir até que nenhum novo estado seja marcado:
 - Marque todo estado que tenha uma transição chegando de qualquer estado já marcado.
 - Se nenhum estado final esteja marcado, aceite. Caso contrário, rejeite.”

A_{AFD} é decidível?

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

A_{AFD} é decidível?

- $A_{AFD} = \{ \langle B, w \rangle \mid B \text{ é um AFD que aceita } w \}$
- Sabemos que uma MT pode simular outra MT, mesmo uma não-determinística então deve poder simular um AFD
 - Comece com uma codificação do AFD e a entrada w
 - Marque o estado inicial
 - Olhe o primeiro símbolo de w e veja em qual estado a máquina estaria depois desse símbolo
 - A lista de estados poderiam estar em uma fita, a tabela da função de transição poderia estar em outra fita (ou em 3 fitas!) e a palavra de entrada na outra fita
 - Apague o símbolo e vá para o próximo símbolo

EQ_{AFD} é decidível?

- $EQ_{AFD} = \{ \langle A, B \rangle \mid A \text{ e } B \text{ são AFDs e } L(A) = L(B) \}$

EQ_{AFD} é decidível

- $EQ_{AFD} = \{ \langle A, B \rangle \mid A \text{ e } B \text{ são AFDs e } L(A) = L(B) \}$

- Seja

$$L(C) = (L(A) \cap \overline{L(B)}) \cup (\overline{L(A)} \cap L(B))$$

- $F =$ “Com entrada $\langle A, B \rangle$, onde A e B são AFDs:
 - Construir C para $L(C)$, como descrito.
 - Executar MT T com entrada $\langle C \rangle$.
 - Se T aceitar, aceite. Se T rejeitar, rejeite.”

A_{AFND} é decidível

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

A_{AFND} é decidível

- $A_{AFND} = \{ \langle B, w \rangle \mid B \text{ é um AFND que aceita } w \}$
- $N =$ “Com entrada $\langle B, w \rangle$, onde B é um AFND e w uma palavra:
 - **Converta B para um AFD equivalente C**
 - Execute a MT M com entrada $\langle C, w \rangle$
 - (M é a máquina que simula AFD)
 - Se M aceita a entrada, N aceita.
 - Se M rejeita, N também rejeita.”

Converta B para o AFD equivalente C?

- Lembre se uma tarefa pode ser especificada em passos finitos, pode ser feita por uma MT...
- Gere todas as combinações de estados
 - Ok, finito!
- Percorra os estados e para cada um calcule a nova transição com base na tabela original
 - Ok, finito!
- Calcule qual é o estado inicial e quais são os estados finais
 - Ok, finito!

Indecidibilidade

Indecidibilidade

- Existem problemas indecidíveis?
- A_{MT} não é decidível
 - Dado um programa e uma entrada, simular aquela entrada
 - $A_{MT} = \{ \langle M, w \rangle \mid M \text{ é uma MT e } M \text{ aceita } w \}$
 - Mais algo, verificação de software: verificar propriedades interessantes de “programas” não é decidível
- O que faz a máquina que tem como linguagem A_{MT} ?
 - Lembre do testador de A_{AFD}

Máquina Universal

- A_{MT} : dado uma descrição de máquina e uma entrada, dizer se a máquina reconhece ou não
 - Entrada $\langle M, w \rangle$, onde M é a descrição de uma MT
 - E w é uma palavra a ser “processada” por essa MT
- Uma MT U é chamada máquina de Turing Universal se ela recebe $\langle M, w \rangle$ como entrada e simula a execução de w em M

Problema da parada (A_{MT} é decidível?)

- (Uma versão dele)
- A_{MT} é reconhecível
 - $U = \langle M, w \rangle$ com entrada $\langle M, w \rangle$ onde M é uma descrição e w uma palavra:
 - Simule M com a palavra w
 - Se M entraria no estado de aceitação, aceite. Se M entraria no estado de rejeição, rejeite.”

Problema da parada

- Discutimos como um A_{AFD} pode ser simulado e que uma MT pode simular/executar outra MT (não-determinismo, fitas)
- $A_{MT} = \{ \langle M, w \rangle \mid M \text{ é uma MT e } M \text{ aceita } w \}$
 - O que a máquina que tem A_{MT} como linguagem faz?
 - Será que essa linguagem é decidível?
- Assuma que é decidível, H é uma MT que decide A_{MT}
 - $H(\langle M, w \rangle)$ aceita se M aceitaria
 - $H(\langle M, w \rangle)$ rejeita se M não aceita

Problema da parada

- Seja D uma MT que usa H
 - D recebe uma MT M como entrada e chama H passando M e a própria descrição de M: $\langle M, \langle M \rangle \rangle$
 - Quando D recebe o resultado de H, se H aceitou, D rejeita e vice-versa
- O que acontece se a entrada de uma máquina for a própria descrição dela?
 - Um programa em Python que conta e diz se o um arquivo de entrada tem um número par de bytes poderia analisar seu próprio código?

Dito de outra forma

- Considere primeiro a máquina H , como já discutimos: ela recebe dois parâmetros, o código/projeto de máquina e uma string de teste, H tem que parar sempre
- Considere 3 máquinas
 - D_1 recebe uma entrada e duplica (recebe um elemento e retorna dois)
 - D_2 recebe dois elementos e chama $H(\text{arg1}, \text{arg2})$
 - D_3 recebe um booleano e retorna a negação
- Agora assuma que essas 3 máquinas são combinadas fazendo uma só

Dito de outra forma

- Considere primeiro a máquina H , como já discutimos: ela recebe dois parâmetros, o código/projeto de máquina e uma string de teste, H tem que parar sempre
- Considere 3 máquinas
 - D_1 recebe uma entrada e duplica (recebe um elemento e retorna dois)
 - D_2 recebe dois elementos e chama $H(\text{arg1}, \text{arg2})$
 - D_3 recebe um booleano e retorna a negação
- Agora assuma que essas 3 máquinas são combinadas fazendo uma só
- Pegue o projeto dessa máquina e passe para ela mesma, o que acontece?

Indecidibilidade

- O que aconteceria se executássemos D com sua própria descrição?
 - D aceitaria D se “D rejeitasse $\langle D \rangle$ ”
 - D rejeitaria D se “D aceitasse $\langle D \rangle$ ”
 - Contradição...
- Não pode ser possível construir uma tal máquina, logo não é possível construir H
 - Qual seria uma possível tentativa de execução de H?
 - H tenta ver o que D faria quando recebesse a entrada $\langle D \rangle$
 - D chamaria H passando D, $\langle D \rangle$, e H por sua vez simularia essa nova entrada...

De outra forma...

- Não necessariamente H simula a máquina
 - Poderia fazer uma “análise”
 - Mesmo assim H não pode existir
- A_{MT} não é decidível, mas é reconhecível...

