



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS DE CRATEÚS

Problemas indecidíveis com Máquinas de Turing

Teoria da Computação

Professor: Rennan Dantas

Universidade Federal do Ceará
Campus de Crateús

09 de maio de 2023

⁰Slides baseados no livro LEWIS, Harry R.; PAPADIMITRIOU, Christos H. Elements of the Theory of Computation. ACM SIGACT News, v. 29, n. 3, p. 62-78, 1998.

Introdução

- Provamos alguns resultados importantes nas últimas aulas
- Vejamos, no nível intuitivo, alguns dos impactos desses resultados à luz da tese de Church-Turing
- Como H não é recursiva, concluímos que não existe algoritmo que decide, para uma máquina de Turing M e uma string de entrada w arbitrárias, se M aceita w ou não
- Problemas para os quais não existem algoritmos são chamados de **indecidíveis** ou **insolúveis**
- O mais famoso e fundamental dos problemas indecidíveis é o **Problema da Parada para Máquinas de Turing**

Reduções

- Uma vez que tenhamos estabelecido que o Problema da Parada é indecidível, a indecidibilidade de uma grande variedade de problemas segue
- Os resultados são provados por **reduções**: mostraremos em cada caso que se uma linguagem L_2 fosse recursiva, então alguma linguagem L_1 , já reconhecidamente não recursiva, seria recursiva

Definição

Sejam $L_1, L_2 \subseteq \Sigma^*$ linguagens. Uma **redução** de L_1 para L_2 é uma função recursiva $\tau : \Sigma^* \rightarrow \Sigma^*$ tal que $x \in L_1$ se e somente se $\tau(x) \in L_2$

Problemas indecidíveis com Máquinas de Turing

Observação importante

Para mostrar que uma linguagem L_2 não é recursiva, devemos identificar uma linguagem L_1 que já sabemos não ser recursiva e então reduzimos L_1 para L_2

Teorema

Se L_1 não é recursiva e existe uma redução de L_1 para L_2 , então L_2 também não é recursiva

Prova

- Suponha que L_2 seja recursiva
- Seja M_2 a máquina de Turing que decide L_2
- Seja T a máquina de Turing que computa a redução τ
- Então a máquina de Turing TM_2 decidiria L_1
- Contradição

Problemas indecidíveis

Os seguintes problemas são indecidíveis com máquinas de Turing:

- (a) Dada uma máquina de Turing M e uma entrada w , M para para a entrada w ?
- (b) Dada uma máquina de Turing M , M para com a fita vazia?
- (c) Dada uma máquina de Turing M , existe alguma string para a qual M para?
- (d) Dada uma máquina de Turing M , M para para todas as entradas?
- (e) Dadas duas máquinas de Turing M_1 e M_2 , elas param para as mesmas strings de entrada?
- (f) Dada uma máquina de Turing M , a linguagem que M semidecide é regular? É livre de contexto? É recursiva?
- (g) Existe uma certa máquina de Turing M para o qual o seguinte problema é indecidível: dado w , M para para w ?

Problemas indecidíveis com Máquinas de Turing

Demonstração - Problema (a)

Aula passada

Demonstração - Problema (b)

- Descrevemos uma redução de H para a linguagem $L = \{ "M" : M \text{ para para } e \}$
- Dada uma descrição de uma máquina de Turing M e uma entrada x , nossa redução simplesmente constrói a descrição de uma máquina de Turing M_w que opera da seguinte forma:
 - M_w , quando iniciada com uma fita vazia, escreve w na sua fita e então começa a simular M
 - Em outras palavras, se $w = a_1 \dots a_n$, então M_w é simplesmente a máquina $Ra_1Ra_2R \dots Ra_nL \sqcup M$
- É fácil ver que a função τ que mapeia $"M"$ $"w"$ para $"M_w"$ é recursiva

Demonstração - Problema (c)

- Podemos reduzir a linguagem L , mostrada ser não recursiva, para a linguagem $L' = \{ "M" : M \text{ para para alguma entrada} \}$
- Dada a representação de qualquer máquina de Turing M , nossa redução constrói a representação de uma máquina de Turing M' que apaga qualquer entrada que é dada e então simula M para a string vazia
- Claramente, M' para para alguma string se e somente se ela para para todas as strings, se e somente se M para para a string vazia

Demonstração - Problema (d)

- O argumento do problema (c) funciona para o problema (d)
- M' é construída de um jeito que aceita alguma entrada se e somente se aceita qualquer entrada

Demonstração - Problema (e)

- Podemos reduzir o problema (d) para esse
- Dada a descrição de uma máquina M , nossa redução constrói a string $\tau("M") = "M" "y"$, onde $"y"$ é a descrição da máquina que aceita qualquer entrada imediatamente
- Claramente, as duas máquinas M e y aceitam as mesmas entradas se e somente se M aceita todas as entradas

Demonstração - Problema (f)

- Reduzimos o problema (b) para o presente
- Mostramos como modificar qualquer máquinas de Turing M para obter a máquina de Turing M' tal que M' para para strings em H ou para nenhuma string, dependendo se M para para a string vazia ou não
- Como não existe algoritmo para determinar se M para para a string vazia, não pode existir nenhuma para determinar se $L(M)$ é \emptyset (que é regular, livre de contexto e recursiva) ou H (que não é nenhuma das três)
- Primeiro, M' salva sua string de entrada e inicia o que M faria para a string vazia
- Quando e se M parar, M' restaura a sua entrada e executa a operação da máquina universal U para aquela entrada
- Então M' ou para para nenhuma entrada, porque ela nunca termina a simulação de M para a entrada e , ou então ela para precisamente para as strings em H

Demonstração - Problema (g)

A máquina M_0 para a qual a declaração faz referência é precisamente a máquina universal U

O que vem por aí?

- Linguagens recursivas e linguagens recursivamente enumeráveis e suas propriedades



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS DE CRATEÚS

Problemas indecidíveis com Máquinas de Turing

Teoria da Computação

Professor: Rennan Dantas

Universidade Federal do Ceará
Campus de Crateús

09 de maio de 2023

⁰Slides baseados no livro LEWIS, Harry R.; PAPADIMITRIOU, Christos H. Elements of the Theory of Computation. ACM SIGACT News, v. 29, n. 3, p. 62-78, 1998.