UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE CRATEÚS

O Problema da Parada

Teoria da Computação

Professor: Rennan Dantas

Universidade Federal do Ceará
Campus de Crateús

09 de maio de 2022

[0]Slides 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.

O Problema da Parada

O princípio da diagonalização

Seja R uma relação binária em um conjunto A e seja $D = \{a : a \in A \text{ e } (a, a) \notin R\}$. Para cada $a \in A$, seja $R_a = \{b : b \in A \text{ e } (a, b) \in R\}$. Então D é distinto de cada R_a.

Exemplo

Considere a relação $R =$
$\{(a, b), (a, d), (b, b), (b, c), (c, c), (d, b), (d, c), (d, e), (e, e), (e, f), (f, a), (f, c), (f, d), (f, e)\}$

O Problema da Parada

Exemplo

	a	b	c	d	e	f
a		×		×		
b		×	×			
c			×			
d		×	×		×	×
e					×	×
f	×		×	×	×	

Figura: Fonte: LEWIS, Harry R.; PAPADIMITRIOU, Christos H. Elements of the Theory of Computation. ACM SIGACT News, v. 29, n. 3, p. 62-78, 1998.

O Problema da Parada

Exemplo

A sequência presente na diagona dessa matriz é:

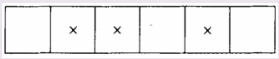

Figura: Fonte: LEWIS, Harry R.; PAPADIMITRIOU, Christos H. Elements of the Theory of Computation. ACM SIGACT News, v. 29, n. 3, p. 62-78, 1998.

O complemento é:

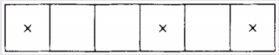

Figura: Fonte: LEWIS, Harry R.; PAPADIMITRIOU, Christos H. Elements of the Theory of Computation. ACM SIGACT News, v. 29, n. 3, p. 62-78, 1998.

O Problema da Parada

Teorema

O conjunto $2^{\mathbb{N}}$ é incontável.

Ideia da demonstração

- Suponha, por absurdo, que $2^{\mathbb{N}}$ seja contável
- Logo podemos enumerar: $2^{\mathbb{N}} = \{R_0, R_1, R_2...\}$
- Agora considere o conjunto $D = \{n \in \mathbb{N} : n \notin R_n\}$
- Como D é um conjunto de números naturais, ele deve pertencer a $2^{\mathbb{N}}$
- Contudo D não pertence à enumeração citada
- Absurdo! Logo, $2^{\mathbb{N}}$ é incontável

O Problema da Parada

Introdução

- Suponha que você escreva um programa que recebe como entrada qualquer programa P (na mesma linguagem) e uma entrada X desse programa
- O seu programa consegue determinar corretamente se o programa P para para a entrada X (retorna "sim") ou se P nunca pararia (retorna "não")
- O seu programa chama-se halts(P,X)
- O seu programa é capaz de, por exemplo, detectar bugs que possam fazer com que um programa nunca se encerre

O Problema da Parada

Diagonal

diagonal(X)
a: se halts(X,X) para então goto a, caso contrário pare

- Observe como diagonal(X) funcional:
  - Se o programa halts decide que o programa X pararia se recebesse ele mesmo como entrada, então diagonal(X) entra em loop
  - caso contrário, diagonal(X) para
- Diagonal(diagonal) para se e somente se halts(diagonal, diagonal) retorna "não"
- Em outras palavras, Diagonal(diagonal) para se e somente se ele não para
- Absurdo! Ou seja, o programa halts não pode existir
- Não pode existir programa/algoritmo para resolver o que o programa halts resolveria

O Problema da Parada

Linguagens não recursivas

- Estamos prontos para definir uma linguagem que não é recursiva e provar que ela não é
- Seja $H = \{\text{“M”“w”: máquina de Turing } M \text{ que para para a entrada } w\}$
- Note primeiro que H é recursivamente enumerável: ela é precisamente a linguagem semi-decidida pela máquina de Turing universal
- Se H é recursiva, então toda linguagem recursivamente enumerável é recursiva

Linguagens não recursivas

- Se H fosse recursiva, então H_1 também seria:
  $H_1 = \{$"M": máquina de Turing M para com a entrada "M"$\}$
- Portanto é suficiente provar que H_1 não é recursiva
- Se H_1 fosse recursiva, então $\overline{H_1}$ também seria já que a classe das linguagens recursivas é fechada para o complemento
  $\overline{H_1} = \{w$: se w não é o código de uma máquina de Turing ou se é o código "M" de uma máquina de Turing que não para para a entrada "'M'$\}$
- $\overline{H_1}$ é a linguagem diagonal, o análogo ao programa diagonal

O Problema da Parada

Linguagens não recursivas

- $\overline{H_1}$ não pode ser sequer recursivamente enumerável
- Suponha, por absurdo, que M^* fosse uma máquina de Turing semi-decide $\overline{H_1}$
- M^* está em $\overline{H_1}$?
- Por definição de $\overline{H_1}$, "M^*" $\in \overline{H_1}$ se e somente se M^* não aceita a entrada "M^*"
- Mas M^* supostamente semidecide $\overline{H_1}$, então "M^*" $\in \overline{H_1}$ se e somente se M^* aceita "M^*"
- Absurdo!

O Problema da Parada

Teorema

A linguagem H não é recursiva; portanto, a classe das linguagens recursivas é um subconjunto estrito da classe das linguagens recursivamente enumeráveis.

H_1 é recursivamente enumerável

Teorema

A classe das linguagens recursivamente enumeráveis não é fechada sobre o complemento.

Próxima aula

O que vem por aí?

- Problemas sem solução com máquinas de Turing
- Linguagens recursivas e linguagens recursivamente enumeráveis e suas propriedades

UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE CRATEÚS

O Problema da Parada

Teoria da Computação

Professor: Rennan Dantas

Universidade Federal do Ceará
Campus de Crateús

09 de maio de 2022

[0]Slides 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.