

LICENCIATURA EM CIÊNCIAS DA COMPUTAÇÃO

COMPUTABILIDADE E COMPLEXIDADE

2. PROBLEMAS DE DECISÃO

José Carlos Costa

Dep. Matemática
Universidade do Minho
Braga, Portugal

1^o semestre 2021/2022

Um *problema de decisão* é uma coleção de questões, cada uma das quais admite como resposta *sim* ou *não*.

EXEMPLO

Seja, para cada $n \in \mathbb{N}$, $P(n)$ a questão “ n é um quadrado perfeito?”.

O problema P consiste portanto das perguntas:

$P(1)$: 1 é um quadrado perfeito?

$P(2)$: 2 é um quadrado perfeito?

$P(3)$: 3 é um quadrado perfeito?

\vdots

Este problema pode ser visto como um *predicado* $P(n)$, que envolve um único *parâmetro* (ou *variável*) n e cujo domínio é \mathbb{N} . Cada $P(n)$ diz-se uma *instância* do problema.

PROBLEMAS DECIDÍVEIS

DEFINIÇÃO

- Uma *solução* para um problema de decisão P é um algoritmo que fornece a resposta para cada instância do problema.
- Um problema que admite uma solução diz-se *solúvel* ou *decidível*. Caso contrário diz-se *insolúvel* ou *indecidível*.

Outros exemplos de problemas de decisão:

- ① Dado um número inteiro n , n é um número primo?
- ② Dados um autómato finito determinista \mathcal{A} e uma palavra w , tem-se $w \in L(\mathcal{A})$?
- ③ Dadas uma máquina de Turing \mathcal{T} e uma palavra w , tem-se $w \in L(\mathcal{T})$?
- ④ Dadas uma linguagem regular L e uma palavra w , tem-se $w \in L$?
- ⑤ Seja L uma linguagem recursiva. Dada uma palavra w , tem-se $w \in L$?
- ⑥ Dada uma palavra $w \in \{x, y\}^*$, tem-se $w \in \text{AutoAceite}$?
- ⑦ Dada uma máquina de Turing \mathcal{T} , será que \mathcal{T} aceita o seu código $c(\mathcal{T})$?
- ⑧ *Problema da paragem*: Dadas uma máquina de Turing \mathcal{T} e uma palavra w , será que \mathcal{T} pára com w ?
- ⑨ *Problema da aceitação*: Dadas uma máquina de Turing \mathcal{T} e uma palavra w , será que \mathcal{T} aceita w ?

- Dos problemas anteriores, são decidíveis o 1, o 2, o 4 e o 5.
- Note-se que o problema 6 é indecidível devido à linguagem *AutoAceite* ser *não recursiva*. Não existe um algoritmo que decida *AutoAceite*. No entanto esta linguagem é *recursivamente enumerável*.

Diz-se então que o problema 6 é *semi-decidível* (isto significa que existe uma máquina de Turing que permite responder nos casos afirmativos, ou seja, nos casos em que *w* é uma palavra de *AutoAceite*).

CODIFICAÇÃO DE PROBLEMAS

DEFINIÇÃO

Seja P um predicado de domínio D , um conjunto enumerável. Para cada $d \in D$, $P(d)$ é uma afirmação verdadeira ou falsa. Suponhamos que existe uma aplicação injetiva $e : D \rightarrow A^*$, onde A é um alfabeto. Seja

$$L_P = \{e(d) \in A^* : P(d) \text{ é verdadeira}\}.$$

A linguagem L_P é dita a **codificação** do problema P .

Diz-se que P é um problema:

- **decidível** se L_P é uma linguagem recursiva.
- **indecidível** se não é decidível.
- **semi-decidível** se L_P é uma linguagem recursivamente enumerável.

EXEMPLO

Seja $A = \{a, b\}$ e seja $P(w)$ a afirmação

“o número de ocorrências da letra a em w é ímpar”

a respeito de uma palavra $w \in A^*$.

- O problema P é decidível.

De facto, dado que w é já uma sequência (sobre o alfabeto A), pode-se considerar que w é o código de si própria, donde P pode ser codificado pela linguagem

$$L_P = \{w \in A^* : |w|_a \text{ é ímpar}\}.$$

Ora, L_P é uma linguagem recursiva, pois é decidida pela máquina de Turing apresentada na página 41 dos slides do capítulo 1. Logo, dada uma palavra $w \in A^*$, a propriedade $P(w)$ é decidível, ou seja, é possível determinar se w tem (ou não) um número ímpar de ocorrências da letra a .

PROBLEMA DA ACEITAÇÃO

DADOS: uma máquina de Turing \mathcal{T} e uma palavra w .

AFIRMAÇÃO: \mathcal{T} aceita w .

TEOREMA

O problema da aceitação é indecidível mas semi-decidível.

Demonstração: O problema da aceitação pode ser codificado pela linguagem

$$L_A = \{c(\mathcal{T})c(w) \in \{x, y\}^* : \mathcal{T} \text{ aceita } w\}.$$

Portanto, o problema da aceitação é indecidível se L_A não é recursiva.

Suponhamos, por contradição, que L_A é recursiva e mostremos que, então, a linguagem AA é recursiva.

Seja \mathcal{T}_A uma máquina de Turing que decide L_A e seja

$$\mathcal{T}_{AA} = \mathcal{T}_1 \mathcal{T}_A$$

onde \mathcal{T}_1 é uma máquina de Turing, de alfabeto de entrada $\{x, y\}$, que transforma a fita da forma $\underline{\Delta}w$ em $\underline{\Delta}wc(w)$.

Verifiquemos que \mathcal{T}_{AA} decide AA :

- Por um lado, se $w \in AA$, então $w = c(\mathcal{T})$ para alguma máquina de Turing \mathcal{T} que aceita w . Logo, \mathcal{T}_{AA} aceita w dando como resultado $\underline{\Delta}1$ pois $wc(w) = c(\mathcal{T})c(w) \in L_A$ e \mathcal{T}_A decide L_A .
- Por outro lado, se \mathcal{T}_{AA} dá como resultado $\underline{\Delta}1$ para a entrada w , então $wc(w) \in L_A$. Isto significa que $wc(w) = c(\mathcal{T}')c(w')$ para alguma máquina de Turing \mathcal{T}' e alguma palavra w' tais que \mathcal{T}' aceita w' . Da definição da função de codificação resulta que $c(w) = c(w')$ e portanto que $w = w' = c(\mathcal{T}')$ o que mostra que $w \in AA$.

Conclui-se assim que \mathcal{T}_{AA} decide AA , o que é uma contradição pois já sabemos que AA não é recursiva. Portanto L_A não é recursiva e o problema da aceitação é indecidível.

Para mostrar que o problema da aceitação é semi-decidível basta notar que L_A é reconhecida pelas máquinas de Turing universais:

- De facto, se \mathcal{T}_U é uma máquina de Turing universal, tem-se que

\mathcal{T} aceita w se e só se \mathcal{T}_U aceita $c(\mathcal{T})c(w)$.

Ou seja, $c(\mathcal{T})c(w) \in L_A$ se e só se \mathcal{T}_U aceita $c(\mathcal{T})c(w)$. Portanto, \mathcal{T}_U aceita L_A .

Isto mostra que L_A é uma linguagem recursivamente enumerável e que o problema da aceitação é semi-decidível. \square

DEFINIÇÃO

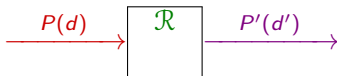
Sejam P e P' dois problemas de decisão, de domínios D e D' respetivamente.

Diz-se que P é *reduzível a* P' (ou que P *se reduz a* P'), e escreve-se $P \leq P'$, se existe uma máquina de Turing \mathcal{R} que

- transforma cada instância $P(d)$ do problema P numa instância $P'(d')$ do problema P'

de tal forma que

- $P(d)$ é verdadeira se e só se $P'(d')$ é verdadeira.



OBSERVAÇÕES

- 1 Note-se que a máquina de Turing \mathcal{R} da definição anterior aplica cada elemento $d \in D$ num elemento $d' \in D'$. Podemos portanto dizer que $P \leq P'$ se existe uma função computável $r : D \rightarrow D'$ tal que $P(d)$ é válida se e só se $P'(r(d))$ é válida.
- 2 Uma formulação alternativa da definição anterior é a seguinte.

Sejam P e P' dois problemas de decisão e sejam $L_P \subseteq A^*$ e $L_{P'} \subseteq B^*$ codificações de P e P' , respetivamente.

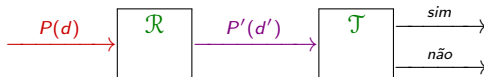
Então $P \leq P'$ se existe uma função computável $r : A^* \rightarrow B^*$ tal que $w \in L_P$ se e só se $r(w) \in L_{P'}$.

PROPOSIÇÃO

Sejam P e P' dois problemas de decisão tais que $P \leq P'$.

- a) Se P' é decidível, então P é decidível.
- b) Se P é indecidível, então P' é indecidível.
- c) Se P' é semi-decidível, então P é semi-decidível.

- O caso da alínea a) pode ser esquematizado da seguinte forma



- Como consequência da alínea b) e do facto de já sabermos que o problema da aceitação é indecidível, pode-se provar que o problema da paragem é também indecidível (ver o Exercício 2.4).
- Podemos agora apresentar mais alguns exemplos de problemas indecidíveis.

TEOREMA

Os seguintes problemas são **indecidíveis**.

- ① $\text{Aceita}_\epsilon(\mathcal{T})$: “ \mathcal{T} aceita ϵ ”.
- ② $\text{Pára}_\epsilon(\mathcal{T})$: “ \mathcal{T} pára com ϵ ”.
- ③ $\text{AceitaNada}(\mathcal{T})$: “ $L(\mathcal{T}) = \emptyset$ ”.
- ④ $\text{AceitaTudo}(\mathcal{T})$: “ $L(\mathcal{T}) = A^*$, onde A é o alfabeto de \mathcal{T} ”.

Demonstração:

- ① Para mostrar que Aceita_ϵ é **indecidível** basta provar que o **problema da aceitação** se **reduz** a Aceita_ϵ . Dadas uma máquina de Turing \mathcal{T} e uma palavra w , seja

$$\mathcal{T}_w = \text{Escreve}_w \mathcal{T}$$

onde Escreve_w é a MT que, começando com a palavra vazia termina com $\underline{\Delta}w$. Então

$$\mathcal{T} \text{ aceita } w \Leftrightarrow \mathcal{T}_w \text{ aceita } \epsilon.$$

Como a função $(\mathcal{T}, w) \mapsto \mathcal{T}_w$ é computável, deduz-se que $\text{Aceitação} \leq \text{Aceita}_\epsilon$ e conclui-se que Aceita_ϵ é **indecidível**.

- 2 Dado que já sabemos que o problema da paragem é indecidível, basta provar que esse problema se reduz a Pára_ϵ . Ora, dadas uma MT \mathcal{T} e uma palavra w , tem-se que

$$\mathcal{T} \text{ pára com } w \Leftrightarrow \mathcal{T}_w \text{ pára com } \epsilon.$$

Logo $\text{Paragem} \leq \text{Pára}_\epsilon$, donde se conclui que Pára_ϵ é indecidível.

- 3 Mostremos que Aceita_ϵ se reduz a $\neg \text{AceitaNada}$. Dada uma MT \mathcal{T} , seja

$$\mathcal{T}_\emptyset = \text{ApagaFita } \mathcal{T}$$

onde ApagaFita é a MT que transforma $\underline{\Delta}x$ em $\underline{\Delta}$. Então

$$\mathcal{T} \text{ aceita } \epsilon \Leftrightarrow \mathcal{T}_\emptyset \text{ aceita alguma palavra.}$$

Como a função $\mathcal{T} \mapsto \mathcal{T}_\emptyset$ é computável, deduz-se que $\text{Aceita}_\epsilon \leq \neg \text{AceitaNada}$. Como Aceita_ϵ é indecidível, $\neg \text{AceitaNada}$ também o é. Logo AceitaNada também é indecidível.

- ① Note-se que

$$\mathcal{T} \text{ aceita } \epsilon \Leftrightarrow \mathcal{T}_{\emptyset} \text{ aceita } A^*.$$

Logo $\text{Aceita}_{\epsilon} \leq \text{AceitaTudo}$, donde AceitaTudo é indecidível. \square

COROLÁRIO

Os problemas

- ① $\neg \text{AceitaNada}(\mathcal{T})$: “ $L(\mathcal{T}) \neq \emptyset$ ”;
- ② $\neg \text{AceitaTudo}(\mathcal{T})$: “ $L(\mathcal{T}) \neq A^*$, onde A é o alfabeto de \mathcal{T} ”;

são indecidíveis.

DEFINIÇÃO

Um problema P , de domínio D , diz-se *trivial* se:

ou

- $P(d)$ é uma afirmação verdadeira para todo o $d \in D$,

ou

- $P(d)$ é uma afirmação falsa para todo o $d \in D$.

Em particular, se P é um problema sobre linguagens recursivamente enumeráveis, ou seja, se

$$D = \{L : L \text{ é uma linguagem recursivamente enumerável}\},$$

então P é *trivial* se todas as linguagens de D satisfazem P ou nenhuma o satisfaz.

TEOREMA [RICE, 1953]

Se P é uma propriedade não trivial sobre linguagens recursivamente enumeráveis, então P é indecidível.

EXEMPLOS

Os problemas seguintes, sobre uma linguagem recursivamente enumerável $L \subseteq A^*$, são indecidíveis.

- 1 $\epsilon \in L$.
- 2 $L = \emptyset$.
- 3 $L = A^*$.

Note-se que o Teorema de Rice estabelece que qualquer problema envolvendo a linguagem aceite por uma máquina de Turing, ou é trivial, ou é indecidível.

No entanto, nem todos os problemas sobre máquinas de Turing são indecidíveis, como o mostra o próximo resultado.

TEOREMA

O seguinte problema é **decidível**.

Escreve Não Branco: Dada uma máquina de Turing \mathcal{T} , será que \mathcal{T} escreve algum símbolo não branco quando é iniciada com a fita vazia?

Demonstração: Suponhamos que \mathcal{T} tem n estados.

Em n passos, ou \mathcal{T} atinge uma configuração de paragem ou \mathcal{T} passa duas vezes no mesmo estado q . Se até essa altura nenhum símbolo branco foi escrito, então nunca o será.

- Isto é evidente no 1º caso, ou seja, no caso de \mathcal{T} parar em menos de n passos.
- No 2º caso, \mathcal{T} repete sempre os mesmos passos entre duas passagens por q entrando em ciclo (a não ser que na segunda passagem por q o cursor esteja situado à esquerda da posição que ocupava na primeira passagem, caso em que \mathcal{T} irá atingir uma configuração de paragem).

Portanto, um algoritmo de decisão para este problema é executar \mathcal{T} durante n passos, se não parar antes: \mathcal{T} escreve um símbolo não branco se e só se o faz durante esse período.