Licenciatura em Ciências da Computação

COMPUTABILIDADE E COMPLEXIDADE

2. Problemas de decisão

José Carlos Costa

Dep. Matemática
Universidade do Minho
Braga, Portugal

1º semestre 2025/2026

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

Exemplo 1

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 $\mathscr{A}$ e uma palavra w, tem-se $w \in L(\mathscr{A})$?
3. Dadas uma linguagem regular L e uma palavra w, tem-se $w \in L$?
4. Seja L uma linguagem recursiva. Dada uma palavra w, tem-se $w \in L$?
5. Dada uma palavra $w \in \{x, y\}^*$, tem-se $w \in AutoAceite$?
6. Dada uma máquina de Turing $\mathscr{T}$, será que $\mathscr{T}$ aceita o seu código $c(\mathscr{T})$?
7. *Problema da paragem:* Dadas uma máquina de Turing $\mathscr{T}$ e uma palavra w, será que $\mathscr{T}$ pára com w?
8. *Problema da aceitação:* Dadas uma máquina de Turing $\mathscr{T}$ e uma palavra w, será que $\mathscr{T}$ aceita w?

- Dos problemas anteriores, apenas são decidíveis os quatro primeiros.
- Note-se que o problema 5 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 5 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 $\mathscr{e} : D \longrightarrow A^*$, onde A é um alfabeto. Seja

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

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 2

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 no Problema 1.6 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 $\mathscr{T}$ e uma palavra w.
Afirmação: $\mathscr{T}$ aceita w.

Teorema

O problema da aceitação é indecidível mas semi-decidível.

Demonstração: O problema da aceitação pode ser codificado pela linguagem

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

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 $\underline{\Delta} w c(w)$.

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 $\mathscr{T}_{AA}$ 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 $\mathscr{T}_U$ é uma máquina de Turing universal, tem-se que

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

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

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

Definição

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

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

$$\xrightarrow{\;P(d)\;} \boxed{\mathscr{R}} \xrightarrow{\;P'(d')\;}$$

Observações

1. Note-se que a máquina de Turing $\mathscr{R}$ 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 $r : D \longrightarrow D'$ 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 $r : A^* \longrightarrow B^*$ tal que $w \in L_P$ se e só se $r(w) \in L_{P'}$.

Proposição

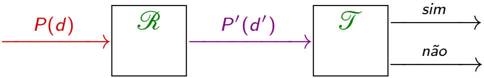
Sejam P e P' dois problemas de decisão tais que $P \leq P'$.

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).
- Pode-se ainda obter outros exemplos de problemas indecidíveis.

Teorema

Os seguintes problemas são indecidíveis.

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

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 $\mathscr{T}$ e uma palavra w, seja

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

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

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

Como a função $(\mathscr{T}, w) \mapsto \mathscr{T}_w$ 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 $\mathscr{T}$ e uma palavra w, tem-se que

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

Logo Paragem$\leq$ Pára_ϵ, donde se conclui que Pára_ϵ é indecidível.

3. Mostremos que Aceita_ϵ se reduz a ¬AceitaNada. Dada uma MT $\mathscr{T}$, seja

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

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

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

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

4. Note-se que

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

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

Corolário

Os problemas

1. $\neg$AceitaNada($\mathscr{T}$): "$L(\mathscr{T}) \neq \emptyset$";
2. $\neg$AceitaTudo($\mathscr{T}$): "$L(\mathscr{T}) \neq A^*$, onde A é o alfabeto de $\mathscr{T}$";

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*.

Exemplo 3

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 $\mathscr{T}$, será que $\mathscr{T}$ escreve algum símbolo não branco quando é iniciada com a fita vazia?

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

Em n passos, ou $\mathscr{T}$ atinge uma configuração de paragem ou $\mathscr{T}$ 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 $\mathscr{T}$ parar em menos de n passos.
- No 2º caso, $\mathscr{T}$ 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 $\mathscr{T}$ irá atingir uma configuração de paragem).

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